



## പ്രധാന പദ്ധതിക്കുള്ള അടിസ്ഥാനങ്ങൾ

- പരിഥാവ് ആർജിക്കേണ്ട ശൈലികൾ
- ഉപയോകതാവിനാൽ നിർവ്വചിക്കപ്പെടുന്ന (user defined) ഡാറ്റാ ഇനത്തിന്റെ ആവശ്യകതയും സ്ട്രക്ചർ ഉപയോഗിച്ച് വിവിധ ഇനം ഡാറ്റാ ഏകോപിപ്പിക്കുന്നതിന്റെ സാധ്യതയും തിരിച്ചറിയുക.
- സ്ട്രക്ചർ ഡാറ്റാതരം നിർമ്മിച്ച് അതിലെ അംഗങ്ങളെ റഹർ ചെയ്യുക.
- അടിസ്ഥാന ഡാറ്റകളെയും ഗ്രൂപ്പ് ഡാറ്റകളെയും പ്രതിനിധികരിക്കുന്നതിൽ നേരുഡിയ് സ്ട്രക്ചറിന്റെ ഉപയോഗം മനസ്സിലാക്കുക.
- ദൈനന്ദിന ജീവിതത്തിലെ പ്രശ്നങ്ങൾ പരിഹരിക്കാൻ സ്ട്രക്ചർ ഉപയോഗിച്ചുള്ള C++ പ്രോഗ്രാമുകൾ തയാറാക്കുക.
- പോയിന്റർ എന്ന ആശയവും, &, \* എന്നീ ഓപ്പറേറ്ററുകളുടെ ഉപയോഗവും വിവരിക്കുക.
- രണ്ട് തരം മെമ്മറി നൈക്കിവയ്ക്കലുകൾ താരതമ്യം ചെയ്യുക. ഡെയനാമിക് ഓപ്പറേറ്ററുകളായ new, delete എന്നീ ഓപ്പറേറ്ററുകളുടെ ഉപയോഗം മനസ്സിലാക്കുക.
- പോയിന്ററിലെ ഓപ്പറേഷനുകൾ ഉദാഹരണം സഹിതം വിവരിക്കുകയും . അതിന്റെ ഒരുപ്പുട്ട് നിർവ്വചിക്കുകയും ചെയ്യുക.
- പോയിന്ററും അറൈയും തമ്മിലുള്ള ബന്ധം സമർപ്പിക്കുക.
- സ്ട്രീജുകൾ കൈകാര്യം ചെയ്യുന്നതിൽ പോയിന്ററിന്റെ ഉപയോഗം അണിയുക.
- സയം സൂചിത (self referential) സ്ട്രക്ചർ റിന്റെ ആശയം വിവരിക്കുക.

വിവിധ പ്രശ്നങ്ങൾ നിർബാരണം ചെയ്യാനുള്ള C++ പ്രോഗ്രാമുകൾ പരിചയപ്പെടിട്ടുണ്ടാണ്. മിക്കവാറും എല്ലാ പ്രശ്നങ്ങളിലും ഡാറ്റ പ്രോസസ്സിൽ ആവശ്യമായി വരാറുണ്ട്. അടിസ്ഥാന ഡാറ്റ വിഭാഗത്തിലുള്ള പൂർണ്ണസംവൃക്തശ്രീ (integers), ഭംഗം സംവദകൾ (fractional numbers) കാരക്ടറുകൾ, സ്ട്രീംഗുകൾ എന്നിവയെക്കുറിച്ച് നാം കഴിഞ്ഞ വർഷം മനസ്സിലാക്കി. ഈ ഡാറ്റകളെ സൂചിപ്പിക്കുന്നതിന് വേണ്ട ഡിജിറ്റൽ ഉപയോഗിക്കുകയും ഇത്തരം വേണ്ട ഡിജിറ്റൽ C++ ലെ അടിസ്ഥാന ഡാറ്റയുള്ള ഉപയോഗചും സ്ഥാപിക്കുകയും ചെയ്തു. എല്ലാ ഡാറ്റയും അടിസ്ഥാന വിഭാഗത്തിൽപ്പെടുന്നില്ല എന്നു നമുക്കറിയാം. ചിലത് വിവിധതരം അടിസ്ഥാന ഡാറ്റ വിഭാഗങ്ങളുടെ ഏകോപനത്തിലും ദേശവാസികൾ നിർമ്മിക്കുന്നത്. എല്ലാത്തരം ഡാറ്റകളെയും ഉൾക്കൊള്ളുന്ന തരത്തിൽ ഡാറ്റ കെടപ്പുകൾ നിർണ്ണയിക്കാൻ ഒരു പ്രോഗ്രാമിൽ ഭാഷയ്ക്കും സാധിക്കില്ല. അതുകൊണ്ടുതന്നെ പ്രോഗ്രാമിൽ ഭാഷകളിൽ ഉപയോകതാവിന്റെ ആവശ്യമനുസരിച്ച് പുതിയ ഡാറ്റായുള്ള അടിസ്ഥാനങ്ങൾ നിർമ്മിക്കാനുള്ള സൗകര്യം അനുവദിച്ചിട്ടുണ്ട്. ഈ അധ്യായത്തിൽ നമ്മൾ അതുരത്തിലുള്ള ഉപയോകതു നിർവ്വചിത ഡാറ്റായുള്ള (user-defined data type) ആയ സ്ട്രക്ചർ (structure) എന്ന ഡാറ്റ കെപ്പിനെ കുറിച്ച് ചർച്ച ചെയ്യുന്നു. മറ്റാരു തരം വേറിയബിളായ പോയിന്ററിനെ (pointer) കുറിച്ചും ഈ അധ്യായത്തിൽ പരിചയപ്പെടാം. പോയിന്റർ എന്ന ആശയം C, C++ എന്നീ ഭാഷകളിലെ ഒരു പ്രത്യേകതയാണ്. ഈ മെമ്മറി സ്ഥാനങ്ങളെ നേരിട്ട്



അതിന്റെ വിലാസത്തിലും (Address) പ്രാപ്യമാക്കുകയും അതുവഴി പ്രോഗ്രാം നടപ്പിലാക്കൽ (Execution) വേഗത്തിലാക്കുകയും ചെയ്യുന്നു. ഈ ആശയം നന്ദായി മനസ്സിലാക്കിയാൽ നമുക്ക് സിസ്റ്റം ലൈവ്സ് (system level) പ്രോഗ്രാമുകളും ഡാറ്റാ സ്റ്റെക്ചർ ആപ്ലിക്കേഷൻുകളും എല്ലാപ്പുതിൽ തയാറാക്കാം.



കമ്പ്യൂട്ടർ ഹാർഡ്‌വെയറുകളെ നിയന്ത്രിക്കുന്ന പ്രോഗ്രാമുകളെ സിസ്റ്റം ലൈവ്സ് പ്രോഗ്രാം എന്നു വിളിക്കാം. സ്റ്റാക്സ്, ക്യൂ, ലിക്സ് തുടങ്ങിയ ഡാറ്റാ രൂപങ്ങളെ നിർമ്മിക്കാനും ഉപയോഗിക്കാനുമുള്ള പ്രോഗ്രാമുകൾ ചെയ്യാം മുകളെ ഡാറ്റാ സ്റ്റെക്ചർ ആപ്ലിക്കേഷൻ എന്ന് വിളിക്കാം.

നമ്മൾ കഴിഞ്ഞ വർഷം ഗ്രൂപ്പ് കമ്പയിലർക്കുള്ള കളക്ഷൻ (GCC) ഉൾക്കൊള്ളുന്ന ജിനി IDE ആണ് C++ പ്രോഗ്രാം നിർമ്മിക്കാൻ ഉപയോഗിച്ചത്. ഈ അധ്യായത്തിൽ C++ പ്രോഗ്രാമുകൾ പൂർണ്ണമായും GCC ഉപയോഗിച്ച് ചെയ്യാവുന്ന തരത്തിലാണ് അവതരിപ്പിച്ചിട്ടുള്ളത്.

## 1.1 സ്റ്റെക്ചർകൾ

വിദ്യാർഥികൾ, തൊഴിലാളികൾ, ഉദ്യോഗസ്ഥർ തുടങ്ങിയവർ അവരവരുടെ സ്ഥാപനങ്ങൾ നൽകുന്ന തിരിച്ചറിയൽ കാർഡ് (Identity card) യരിക്കാറുണ്ട്. ചിത്രം 1.1 കാണിക്കുന്നത് ഒരു കൂട്ടിയുടെ തിരിച്ചറിയൽ കാർഡ് ആണ്. പട്ടിക 1.1 ലെ ഒന്നാമത്തെ കോളം ത്തിൽ കാർഡിൽ (പ്രിൻ്റ് ചെയ്തിരിക്കുന്ന ചില ഡാറ്റകളാണ് കാണിച്ചിരിക്കുന്നത്. പട്ടികയുടെ രണ്ടാമത്തെ കോളം ഈ ഡാറ്റകൾക്കുന്നേയാജ്യമായ C++ ഡാറ്റാഖനങ്ങൾ ഉപയോഗിച്ച് പൂരിപ്പിക്കുക.

### Govt.HSS Thykkunnam,Kottayam

Student ID: 12345  
Student Name: Sneha S.Raj  
Date of Birth: 20-02-1997  
Blood group: O+ve  
Address: Sneha Nilayam  
Gandhi Nagar  
Chemmanavattom  
Pin: 685 531



ചിത്രം 1.1: കൂട്ടിയുടെ തിരിച്ചറിയൽ കാർഡ്

Data	C++ data type
12345 Sneha S. Raj 20/02/1997 O+ve Snehanilayam, Gandhi Nagar, Chemmanavattom, Pin 685 531	

പട്ടിക 1.1: ഡാറ്റയും C++ ഡാറ്റാ രേഖപ്പെടുത്തുന്നതും

നിങ്ങൾ അഡ്മിഷൻ നമ്പറിന് short അല്ലെങ്കിൽ int ഡാറ്റാടെപ്പും, പേര് (Sneha S. Raj), രക്തഗ്രൂപ്പ് (blood group (O +ve)), വിലാസം (അധ്യാള്സ്) തുടങ്ങിയവയ്ക്ക് char അനൈയും ഉപയോഗിച്ചിട്ടുണ്ടാവും. ചിലപ്പോൾ നിങ്ങൾക്ക് ജനന തീയതിക്കും (Date of birth) വിലാസത്തിനും എറ്റവും യോജിച്ച ഡാറ്റാഖനം കൂടിത്താൻ കഴിഞ്ഞില്ല എന്നു വന്നേക്കാം. നമുക്ക് 20/02/1997 എന്ന തീയതി പരിഗണിക്കാം. ഈത് ഒരു സംയോജിത (compound) ഡാറ്റാ ഖനമാണ് എന്ന് വിശകലനം ചെയ്താൽ മനസ്സിലാവും. ഈതിൽ

ദിവസസംഖ്യ (20), മാസസംഖ്യ (02), വർഷസംഖ്യ (1997) എന്നിവ കൂടിച്ചേർന്നിൽക്കുന്നു. മാസസംഖ്യകൾ പകരം മാസത്തിന്റെ പേര് ഉപയോഗിക്കാറുണ്ട്. ഇതുപോലെ വിലാസം എന്നത്, വീടുനമ്പൾ, വീടുപേര്, സ്ഥലം, ജില്ല, സംസ്ഥാനം, പിൻകോഡ് എന്നിവയുടെ ഒരു സംയോജിത രൂപമാണ്. ഈ കാർഡിലെ മുഴുവൻ വിവരങ്ങളെയും വേണ്ട മെക്കിൽ ഒരു വലിയ സംയോജിത ഡാറ്റാഓഫുൾമായി (grouped datatype) പരിഗണിക്കാം. ഇത്തരം സംയോജിത ഡാറ്റാഓഫുൾമായി (user defined data type) പരിഗണിക്കാം എന്നു വിളിക്കുന്നു. C++ തു ഇത്തരം പുതിയ ഡാറ്റാഓഫുൾമൾ നിർവ്വചിക്കാനുള്ള സൗകര്യമുണ്ട്.

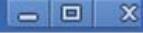
**സ്റ്റ്രക്ചർ:** - യുക്തിപരമായി പരസ്പരം ബന്ധപ്പെട്ടു നിൽക്കുന്ന ഒരു കൂട്ടം ഡാറ്റാ ഇന അങ്ങേ പ്രതിനിധാനം ചെയ്യാൻ C++ തു ഉപയോഗിക്കുന്ന ഫോട്ടുകൂടിയ ഉപയോക്കത്യും നിർവ്വചിത ഡാറ്റാഓഫുൾമാണ് സ്റ്റ്രക്ചർ. ഈ ഡാറ്റകൾ ചിലപ്പോൾ വ്യത്യസ്ത അങ്ങളായ ഡാറ്റാഓഫുൾമായിൽ ഉള്ളവ ആയിരിക്കും. 11-ാം ക്ലാസിൽ നമ്മൾ പഠിച്ച അറിയേണ്ടിലുള്ള ഡാറ്റയുടെ ഏകോപിതരുപമാണ്. പക്ഷേ ഒരു പേരിലുടെ അറിയപ്പെടുന്ന പല തരത്തിലുള്ള ഡാറ്റയുടെ ഏകോപിത രൂപമാണ് സ്റ്റ്രക്ചർ. C++ തു സ്റ്റ്രക്ചർ നിർവ്വചിക്കുന്നതും ഉപയോഗിക്കുന്നതും നമുക്ക് ചർച്ച ചെയ്യാം.

### 1.1.1 സ്റ്റ്രക്ചർ നിർവ്വചനം

കമ്പ്യൂട്ടർ പ്രോഗ്രാമിലുടെ പ്രശ്നങ്ങൾ നിർബന്ധമാം ചെയ്യുന്നോൾ മേൽ വിവരിച്ച പ്രകാരം ഏകോപിത ഡാറ്റാഓഫുൾമായി ഉപയോഗിക്കേണ്ടിവരും. ആ സമയത്ത് ഏകോപിത ഡാറ്റാഓഫുൾമായ പേര് നൽകുന്നതോടൊപ്പം, അതിൽ അടങ്കിയിരിക്കുന്ന ഡാറ്റാഓഫുൾമാൾ കണ്ണെത്തി നിർവ്വചിക്കേണ്ടതുണ്ട്. താഴെ കൊടുത്തിരിക്കുന്ന വാക്യാലടന ഉപയോഗിച്ച് സ്റ്റ്രക്ചർജിനെ നിർവ്വചിക്കാം:

```
struct structure_tag
{
    data_type variable1;
    data_type variable2;
    .....
    .....
    data_type variableN;
};
```

മുകളിൽ ചേർത്ത വാക്യാലടനയിൽ **struct** എന്നത് സ്റ്റ്രക്ചർ നിർവ്വചിക്കാനുള്ള കീവേൾ ആണ്. **structure\_tag** സ്റ്റ്രക്ചർജിന്റെ പേര് നിർണ്ണയിക്കുന്ന ഫൈലിലൂടെ ആണ്. **variable1**, **variable2**, ..., **variableN** എന്നിവ സ്റ്റ്രക്ചറിലെ അടിസ്ഥാന ഡാറ്റാ ഓഫുൾമായുടെ പേര് നിർണ്ണയിക്കുന്ന ഫൈലിലൂടെയുള്ളാണ്. സ്റ്റ്രക്ചർജിന്റെ പേര് സൂചിപ്പിക്കാൻ ഉപയോഗിക്കുന്ന ഫൈലിലൂടെയുള്ള ആയ **structure\_tag** ഒരു പുതിയ ഉപയോക്കത്യും നിർവ്വചിത ഡാറ്റാഓഫുൾമാണ്. ഈ ഡാറ്റാഓഫുൾമായും മറ്റ്



തൊരു ഡാറ്റ ഇനത്തെയും പോലെ നിശ്ചിത മെമ്മറി ആവശ്യമാണ്. ഈത് പുതിയ വേരിയബിളുകൾ പ്രവ്യാഹിക്കാൻ ചെയ്യാൻ ഉപയോഗിക്കുന്നു. ഈ ഡാറ്റാഇനത്തിലുള്ള വേരിയബിളുകൾ ഫണ്ട്ഷൻ ആർഗിഫെൽഡ്രായോ, ഫണ്ട്ഷനിൽ നിന്നും റിട്ടേൺ ചെയ്യുന്ന വേരിയബിളായോ ഉപയോഗിക്കാം. സ്ട്രക്ചർ നിർവ്വചനത്തിൽ ബോയ്ക്കറ്റിനുള്ളിൽ ({} ) കാണുന്ന വേരിയബിളുകളെ സ്ട്രക്ചറിൽ മെമ്പർ വേരിയബിളുകൾ എന്നു പറയുന്നു. ഈത്തരം മെമ്പർ വേരിയബിളിൽ ഡാറ്റാഇനം അടിസ്ഥാന ഡാറ്റാഇനമോ ഉപയോക്കു നിർവ്വചിത ഡാറ്റാഇനമോ ആവാം. മെമ്പർ വേരിയബിളുകളുടെ വലുപ്പങ്ങളുടെ തുകയാണ് സ്ട്രക്ചർ വേരിയബിളിൽ ആകെ വലുപ്പം.

ഈ നമുക്ക് എഴുപ്പെട്ടിട്ടി കാർഡിലെ ജനനത്തീയതിയായ 20/02/1997 പ്രതിനിധീകരിക്കുന്ന ഒരു സ്ട്രക്ചർ തയാറാക്കി നോക്കാം. ഈ തീയതിയിലൽ മുന്ന് പുർണ്ണ സംവ്യൂക്തി ഉണ്ട്. ആദ്യത്തെത്ത് ദിവസസംവ്യൂഹം രണ്ടാമത്തെത്ത് മാസസംവ്യൂഹം മുമ്പാം മത്തേതെത് വർഷസംവ്യൂഹമാണ്. ചുവടെചേർത്ത രീതിയിൽ നമുക്ക് ഒരു സ്ട്രക്ചർ എടുത്ത തയാറാക്കാം.

```
struct date
{
    int dd;
    int mm;
    int yy;
};
```

ഈവിടെ date എന്നത് സ്ട്രക്ചർ ടാഗിനെ (സ്ട്രക്ചർിൽ പേരിനെ) സൂചിപ്പിക്കുന്നു. dd, mm, yy എന്നീ int കെപ്പ് വേരിയബിളുകൾ യഥാക്രമം ദിവസം, മാസം, വർഷം എന്നിവയെ സൂചിപ്പിക്കുന്നു. ഈ സ്ട്രക്ചർിലെ അംഗങ്ങളാണ്. മാസസംവ്യൂഹം പകരം മാസത്തിൽ പേര് ചേർക്കണമെങ്കിൽ സ്ട്രക്ചർ നിർവ്വചനത്തിൽ താഴെ കൊടുത്ത രീതിയിൽ മാറ്റം വരുത്തണം.

```
struct strdate
{
    int day;
    char month[10]; // മാസത്തിൽ പേര് സ്ട്രീങ്ങാണ്
    int year;
};
```

പ്രോഗ്രാമുകൾ നിർമ്മിക്കുന്ന സമയത്ത് ചില ഡാറ്റകൾ പരസ്പരബന്ധം പുലർത്തുന്ന തായി കാണാം. ഈ ഡാറ്റകളെ ഏകോപിപ്പിച്ച് കുറച്ചുകൂടി ഒരുക്കമുള്ള രൂപത്തിലേക്ക് മാറ്റാൻ സ്ട്രക്ചറുകൾ ഉപയോഗിക്കാം. ഉദാഹരണത്തിന് അധികം നമ്പർ, പേര്, ശുപ്പ്, ഫൈല് തുടങ്ങിയ വിവരങ്ങൾ കൂടിയുമായി ബന്ധപ്പെട്ടിരിക്കുന്നു. താഴെ കൊടുത്ത രീതിയിൽ ഒരു സ്ട്രക്ചർ നിർമ്മിക്കാം.

```
struct student
{
    int adm_no;
```

```

char name[20];
char group[10];
float fee;
}

```

ഈ നിങ്ങൾ സ്വന്തമായി സ്ട്രക്ചർ നിർവ്വചിക്കാൻ ശ്രമിക്കു.



'address' ഉം 'blood group' ഉം ആകട്ട പുതിയ സ്ട്രക്ചറു കൾ. സ്ലാഷ് ഗ്രൂപ്പിൽ, ഗ്രൂപ്പിൽപ്പേരും, **rh** വിലയും ഉണ്ടായിരിക്കണം.

നമുക്ക് പെരും

തൊഴിലാളിയുടെ വിവരങ്ങളിൽ **employee code, name, gender, designation, salary** എന്നിവ പ്രധാനപ്പെട്ടവയാണ് എന്ന് നമു കരിയാമല്ലോ. ഈ ഉൾപ്പെടുത്തി അനുയോജ്യമായ ഒരു സ്ട്രക്ചർ നിർവ്വചിക്കുക.

ഈവരെ നമ്മൾ ചർച്ച ചെയ്തത് സ്ട്രക്ചർ നിർവ്വചിക്കുന്ന രീതിയാണ്. ഈ സ്ട്രക്ചർ വേരിയബിളുകൾ എങ്ങനെ പ്രവ്യാപിക്കാമെന്നും അതിൽ ഡാറ്റ എങ്ങനെ ശേഖരിക്കാ മെന്നും മനസ്സിലാക്കാം.

### 1.1.2 വേരിയബിൾ പ്രവ്യാപനവും (Declaration) മെമ്മറി നീക്കിവയ്ക്കലും (Allocation)

അടിസ്ഥാന ഡാറ്റാളുന്നങ്ങളെ പോലെ തന്നെ സ്ട്രക്ചർ ഡാറ്റാളുന്നതിനും വേരിയബി ഭൂകൾ ഉണ്ടെങ്കിൽ മാത്രമേ ഡാറ്റ ശേഖരിക്കാൻ കഴിയു. താഴെ കൊടുത്തിരിക്കുന്ന മാതൃ കയിലാണ് വേരിയബിളുകൾ പ്രവ്യാപിക്കുന്നത്.

```
struct structure_tag var1, var2, ..., varN;
```

OR

```
structure_tag var1, var2, ..., varN;
```

ഈ വാക്യാലഘടനയിൽ **structure\_tag** എന്നത് സ്ട്രക്ചർ ഡാറ്റ പേരും **var1, var2, ..., varN** എന്നിവ സ്ട്രക്ചർ വേരിയബിളുകളുമാണ്. നമുക്ക് **date, strdate** എന്നീ സ്ട്രക്ചറുകളിൽ വേരിയബിളുകൾ പ്രവ്യാപിക്കാം.

```
date dob, today; OR struct date dob, today;
```

```
strdate adm_date, join_date;
```

വേരിയബിൾ പ്രവ്യാപനം നടക്കുമ്പോൾ അതിൻ്റെ വലുപ്പം അനുസരിച്ച് മെമ്മറി നീക്കി വയ്ക്കുന്നു എന്ന് നമുക്കരിയാം. ഒരു സ്ട്രക്ചർ വേരിയബിളുകൾ വലുപ്പം എന്തായി രിക്കും? സ്ട്രക്ചർ ഒരു ഉപയോക്തൃ നിർവ്വചിത ഡാറ്റാളുന്നമായതിനാൽ അതിൻ്റെ വലുപ്പം, നിർവ്വചനം അനുസരിച്ചായിരിക്കും. മേൽ സൂചിപ്പിച്ച വേരിയബിൾ പ്രവ്യാപനത്തിൽ, സ്ട്രക്ചർ **date** വേരിയബിളുകളായ **dob** യും **today** യും 12 byte വീതം വലുപ്പമുള്ള വയായിരിക്കും (മുന്ന് **int** ഡാറ്റാളുന്നതിലുള്ള അംഗങ്ങൾ, **GCC** തിൽ **int** എൻ്റെ വലുപ്പം 4 byte ആയതിനാൽ  $3 \times 4 = 12$  bytes). സ്ട്രക്ചർ **strdate** ലെ വേരിയബി താഴെ **join\_date** എൻ്റെ മെമ്മറി ഘടന ചിത്രത്തിൽ ചേർക്കുന്നു.



join_date											
day		month [10]									
← 4 Bytes →											→ 4 Bytes ←

ചിത്രം 1.2: സ്റ്റെറ്റ് വേരിയബിളിന്റെ മെമ്മറി നീക്കിവയ്ക്കൽ



GCC യിൽ int കെപ്പിന്റെ വലുപ്പം 4 bytes ഉം Turbo IDE യിൽ 2 bytes ഉം ആണ്. ചിത്രം 1.2 ലെ day, year എന്നീ അംഗങ്ങൾ 4 bytes വിത്തം ഉപയോഗിക്കുന്നു. GCC ഉപയോഗിക്കുന്നതിനാൽ പിലപ്പോൾ നമ്മൾക്ക് ഇത്തരം മെമ്മറി ആവശ്യമില്ല എങ്കിൽ short ഡാറ്റാളും ഉപയോഗിക്കുകയാണ് നല്ലത്.

join\_date എന്ന വേരിയബിളിൽ day, month, year എന്നീ മൂന്ന് അംഗങ്ങളാണ് ഉള്ളത്. ഈ തമാക്രമം 4 bytes, 10 bytes, 4 bytes മെമ്മറി ഉപയോഗിക്കുന്നു. അങ്ങനെ ഈ വേരിയബിളിന്റെ ആകെ മെമ്മറി ഉപയോഗം 18 bytes ആണ്.



ഈ നമ്മൾ മൂന്ന് നിർവ്വചിച്ച സ്റ്റെറ്റ് student ന്റെ മെമ്മറി വലുപ്പം കണ്ണുപിടിക്കുക. കൂടാതെ student സ്റ്റെറ്റ് വേരിയബിൾ പ്രവ്യാഹരിക്കുക. ഈ വേരിയബിളിന്റെ മെമ്മറി നീക്കി വയ്ക്കുന്നതിന്റെ ചിത്രം വരയ്ക്കുക.

#### നമുക്ക് ചെയ്യാം

താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ ഒരു സ്റ്റെറ്റ് വേരിയബിളിനെ അതിന്റെ നിർവ്വചനത്തിന്റെ കൂടുതൽനേര പ്രവ്യാഹരിക്കാം.

```
struct complex
{
    short real;
    short imaginary;
} c1, c2;
```

complex എന്നു പേരുള്ള സ്റ്റെറ്റ് കോംപ്ലക്സ് നമ്പറിനെ പ്രതിനിധാനം ചെയ്യുന്നു. c1 ഉം c2 ഉം ഈ സ്റ്റെറ്റ് തരത്തിലുള്ള രണ്ട് വേരിയബിളുകളാണ്. സ്റ്റെറ്റ് നിർവ്വചനത്തിന്റെ കൂടുതൽ വേരിയബിൾ പ്രവ്യാഹരിക്കുന്നേരം വേണമെങ്കിൽ സ്റ്റെറ്റ് ടാഗ് ഓഫിവാക്കുകയും ചെയ്യാം. ചുവടെ കൊടുത്ത ഉദാഹരണം നോക്കുക.

```
struct
{
    int a, b, c;
} eqn_1, eqn_2;
```

ഈ നിർവ്വചനത്തിനു ഒരു പരിമിതിയുണ്ട്. സ്റ്റെറ്റ് ടാഗ് ഇല്ലാത്തതിനാൽ വീണ്ടും പുതിയ വേരിയബിളുകൾ നിർമ്മിക്കാൻ ഈ നിർവ്വചനം പര്യാപ്തമല്ല. സ്റ്റെറ്റ് ടാഗ് അംഗങ്ങൾ ഒരേ തരമാണെങ്കിൽ ഒറ്റവർ പ്രവ്യാഹരിക്കുമെന്നും ആവശ്യമുള്ളു.

## വേരിയബിളിന് പ്രാരംഭ വിലനൽകൽ (Initialisation)

വേരിയബിളുകൾ പ്രവ്യാഹിക്കുന്നേം അതിലേക്ക് ചില വിലകൾ നൽകാവുന്ന താണ്. ഈ സ്ട്രക്ചർ വേരിയബിളിന്റെ കാര്യത്തിലും ശ്രദ്ധിയാണ്. സ്ട്രക്ചർ വേരിയബിളിന് താഴെ കൊടുത്ത രീതിയിൽ പ്രാരംഭവില നൽകാം.

```
structure_tag variable={value1, value2, ..., valueN};
```

ഉദാഹരണമായി student സ്ട്രക്ചറിന് താഴെ കൊടുത്ത രീതിയിൽ പ്രാരംഭവില നൽകാം.

```
student s={3452, "Vaishakh", "Science", 270.00};
```

സ്ട്രക്ചർ അംഗങ്ങളായ വേരിയബിളുകൾക്ക് നിർവ്വചനത്തിൽ നൽകിയിരിക്കുന്ന അതേ ക്രമത്തിലാണ് ബ്രായ്ക്ക്രീൽ കൊടുത്ത ഡാറ്റകൾ നൽകിയിരിക്കുന്നത്. അതുകൊണ്ടു തന്നെ ഡാറ്റയുടെ ക്രമത്തിലും തരത്തിലും ശ്രദ്ധിക്കേണ്ടതാണ്. മുകളിൽ കൊടുത്ത C++ വാചകം 38 ബെബ്രെസ്റ്റ് മെമ്മറി s എന്ന വേരിയബിളിനു വേണ്ടി ഉപയോഗിക്കുന്നു. 3452, "Vaishakh", "Science", 270.00 എന്നീ ഡാറ്റ adm\_no, name, group, fee എന്നീ വേരിയബിളുകൾക്ക് തമാക്രമം നൽകപ്പെടുന്നു (Assign ചെയ്യപ്പെടുന്നു).

എല്ലാ അംഗങ്ങൾക്കും വിലകൾ നൽകിയില്ലെങ്കിൽ ആദ്യം മുതലാളുള്ള അംഗങ്ങൾക്ക് ക്രമത്തിൽ വിലകൾ നൽകപ്പെടും, അവസാനം ബാക്കി വരുന്ന അംഗങ്ങൾക്ക് സംഖ്യയാണെങ്കിൽ 0 (zero) യും, സ്ട്രീറ്റ് ആണെങ്കിൽ '\0' (സൾ ക്യാരക്ടറും) നൽകപ്പെടുകയും ചെയ്യുന്നു.

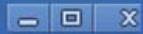
ഒരു സ്ട്രക്ചർ വേരിയബിളിലേക്ക് മറ്റാരു സ്ട്രക്ചർ വേരിയബിളിന്റെ ഡാറ്റ നൽകാൻ കഴിയും. അങ്ങനെ ചെയ്യാമെങ്കിൽ രണ്ട് വേരിയബിളുകളും ഒരേ സ്ട്രക്ചർ ദേശപ്പെടുത്താൻ കഴിയില്ലെങ്കിലും ഉദാഹരണം താഴെ ചേർക്കുന്നു.

```
student st = s;
```

ഈ വാചകം വേരിയബിൾ st ലേക്ക് വേരിയബിൾ 's' എൽ ഡാറ്റ ശേഖരിക്കുന്നു.

<b>s</b>				
adm_no	name	group	fee	
3452	Vaishakh	Science	270.00	
<b>st</b>				
adm_no	name	group	fee	
3452	Vaishakh	Science	270.00	

ചിത്രം 1.3: സ്ട്രക്ചർ അഥവാ മെമ്മറ്



ഒരു സ്റ്റ്രക്ക്‌ചർ നിർവ്വചിക്കുന്നോൾ അതിനുകൂടുതുള്ള അംഗങ്ങളായ വേറിയബിള്ളുകൾക്ക് പ്രാരംഭ വില നൽകാൻ പാടില്ല. കാരണം, സ്റ്റ്രക്ക്‌ചർ നിർവ്വചനം നടക്കുന്നോൾ മെമ്മറി നീക്കിവരയ്ക്കപ്പെടുന്നില്ല. അതായത്, സ്റ്റ്രക്ക്‌ചർ നിർവ്വചനത്തിനെ ഒരു വീടിന്റെ സ്റ്റൂപിൾ്സ് (പ്ലാൻ) ആയി കണക്കാക്കാം. സ്റ്റൂനിൽ മുൻയുടെ പേരും രൂപവും കാണാമെങ്കിലും മുൻ കളിൽ ഒന്നും ശേഖരിക്കാൻ കഴിയില്ല. കാരണം, അവയ്ക്ക് ധമാർമ സമലം ഇല്ല എന്നതുതന്നെ. ഒരു വീടിന്റെ ആകെ സമലം എന്നത് എല്ലാ മുൻകളുടെയും സമലത്തിന്റെ ആകെത്തുകയാണ്. അതുപോലെ സ്റ്റ്രക്ക്‌ചർ വേറിയബിള്ളിന്റെ മെമ്മറി വലുപ്പം എന്നത് അതിന്റെ അംഗങ്ങളുടെ വലുപ്പത്തിന്റെ തുകയാണ്. ഈ സ്റ്റൂൺക്കാം ഒരേ റിതിയിലുള്ള കൂടെ വീടുകൾ നിർമ്മിക്കാം. വീടുകൾക്ക് വ്യത്യസ്തമായ പേരും സമലവും ഉണ്ടാക്കാം. ഇതുപോലെ സ്റ്റ്രക്ക്‌ചർ നിർവ്വചനം എന്നത് സ്റ്റൂപിൾ്സ് സ്റ്റൂപിൾ്സ് സ്റ്റൂപിൾ്സ് വേറിയ ബിൾ എന്നത് ഇതിന്റെ പുർത്തീകരണവുമാണ്. ഒരു സ്റ്റൂപിൾ്സ് ഉപയോഗിച്ച് അനേകം വീടുകൾ നിർമ്മിക്കുന്നതുപോലെ സ്റ്റ്രക്ക്‌ചർ നിർവ്വചനം ഉപയോഗിച്ച് വ്യത്യസ്തമായ പേരിൽ അനേകം വേറിയബിള്ളുകൾ പ്രവൃംപിക്കാം. ഓരോന്നിനും വ്യത്യസ്തമായ മെമ്മറി സമലവും ഉണ്ടായിരിക്കും.

### 1.1.3 സ്റ്റ്രക്ക്‌ചർ അംഗങ്ങളെ ഉപയോഗിക്കൽ

അര (array) എന്നത് ഡാറ്റയുടെ കൂടുമാണ്. ഒരു അരയിലെ അംഗങ്ങളെ സബ്സ്ക്രിപ്റ്റുപയോഗിച്ച് ഒരുക്കുകൾ ശേഖരിക്കാനും എടുക്കാനും കഴിയും. ഇതുപോലെ സ്റ്റ്രക്ക്‌ചർ അംഗങ്ങളെ ഉപയോഗിക്കാൻ വേണ്ടി പിരീഡ് ചിഹ്നം അമവാ ഡോറ്റ് ഓപ്പറേറ്റർ (.) ഉപയോഗിക്കുന്നു. ഈത് ഉപയോഗിക്കുന്നതിന്റെ വാക്യാലടന ചുവടെ കൊടുത്തിരിക്കുന്നു.

```
structure_variable.element_name
```

പ്രോഗ്രാമുകളിൽ സ്റ്റ്രക്ക്‌ചർ ഡാറ്റ ഉപയോഗിക്കണമെങ്കിൽ മുകളിൽ കൊടുത്തപോലെ സ്റ്റ്രക്ക്‌ചർ അംഗങ്ങളെ സൂചിപ്പിക്കണം. ചില ഉദാഹരണങ്ങൾ ചുവടെ കൊടുത്തിരിക്കുന്നു.

```
today.dd = 10;
strcpy(adm_date.month, "June");
cin >> s1.adm_no;
cout << c1.real + c2.real;
```

എന്നാൽ c1+c2 എന്ന പ്രയോഗം സാധ്യമല്ല. കാരണം '+' എന്ന ഓപ്പറേറ്റർ സംഖ്യകളുടെ കൂടെ മാത്രമേ ഉപയോഗിക്കാൻ പാടുള്ളൂ.

ഈ ഒരു രണ്ട് സ്റ്റ്രക്ക്‌ചർ വേറിയബിള്ളിലെ വിലനൽക്കൽ ഓപ്പറേഷൻ (Assignment) രസകരമായ ചില കാര്യങ്ങൾ മനസ്സിലാക്കാം.

ചുവടെ ചേർത്തിരിക്കുന്ന രണ്ട് സ്റ്റ്രക്ക്‌ചറ്റുകൾ പതിഗണിക്കുക.

<pre>struct test_1 {     int a;     float b; }t1={3, 2.5};</pre>	<pre>struct test_2 {     int a;     float b; }t2;</pre>
--	---

## 1. സ്റ്റെക്ചറും പോയിസ്റ്റും

മുകളിൽ കൊടുത്ത രണ്ട് സ്റ്റെക്ചറുകളിലെയും അംഗങ്ങൾ ഒരേ എന്നവും ഒരേ തരവും, ഒരേ പേരുള്ളവയുമാണ്. test\_1 ലെ വേരിയബിളായ t1 ലെ അംഗങ്ങളായ a യും b യും 3, 2.5 എന്നീ പ്രാരംഭ വിലകൾ യഥാക്രമം നൽകിയിട്ടുണ്ട്. പക്ഷെ t2=t1; എന്ന വിലനൽകൽ നടത്തുകയാണെങ്കിൽ അത് തെറ്റാണ് എന്ന സന്ദേശം നമുക്ക് ലഭിക്കും. കാരണം t1 ഉം t2 ഉം രണ്ട് തരത്തിലുള്ള സ്റ്റെക്ചർ വേരിയബിളുകളാണ്. പക്ഷെ നമ്മൾക്ക് t1 സ്റ്റെക്ചറിലെ വില t2 സ്റ്റെക്ചറിൽ നൽകണമെങ്കിൽ ചുവടെ ചേർത്ത രീതി ഉപയോഗിക്കാം.

```
t2.a = t1.a;      t2.b = t1.b;
```

ഈ സാധ്യമാവാൻ കാരണം, int ദേപ്പിലുള്ള വേരിയബിളുകളാണ് ഇവിടെ വിലനൽകൽ പ്രസ്താവനയിൽ (Assignment statement) ഉപയോഗിച്ചിരിക്കുന്നത്.

ഈ നമുക്ക് ഇതുവരെ പറിച്ച് കാരുജങ്ങൾ ഉപയോഗിച്ചുള്ള ഒരു പ്രോഗ്രാം പരിചയപ്പെട്ടാം. ഇവിടെ student സ്റ്റെക്ചറിൽ രജിസ്ട്രേഷൻ നമ്പർ, പേര്, കുട്ടിയുടെ CE മാർക്ക് (തുടർ മുല്യനിർണ്ണയം), PE മാർക്ക് (പ്രാക്ടിക്കൽ മുല്യനിർണ്ണയം), TE (ദേശ മുല്യനിർണ്ണയം) എന്നിവ അംഗങ്ങളായി ചേർത്തിട്ടുണ്ട്. സ്റ്റെക്ചറിലെ ഈ അംഗങ്ങളുടെ വിലകൾ ഇൻപുട്ടായി നൽകിയാൽ പ്രോഗ്രാം തുടർച്ചയും സമഗ്രവുമായ മുല്യനിർണ്ണയ തിരിക്കേണ്ട ഭാഗമായുള്ളത് ആകെ മാർക്ക് കണക്കുപിടിച്ച് എല്ലാ വിവരങ്ങളും പ്രദർശിപ്പിക്കും.

### Program 1.1: കുട്ടിയുടെ ആകെ മാർക്ക് കാണാൻ

```
#include <iostream>
#include <cstdio>           //gets() ഫലങ്ങൾ ഉപയോഗിക്കാൻ
using namespace std;
struct student //സ്റ്റെക്ചർ നിർവ്വചനം തുടങ്ങുന്നു
{
    int reg_no; // രജിസ്ട്രേഷൻ നമ്പർ 32767 ലും കുടാം, അതുകൊണ്ട് int ഉപയോഗിക്കുന്നു.
    char name[20];
    short ce; //int 4 ബെബ്രൂൾ ഉപയോഗിക്കും എന്നാൽ ce സ്കോറിന് ചെറിയ സംഖ്യ മതി
    short pe;
    short te;
};                                //സ്റ്റെക്ചർ നിർവ്വചനത്തിന്റെ അവസാനം
int main()
{
    student s;                  //സ്റ്റെക്ചർ വേരിയബിളിന്റെ പ്രവൃത്തി
    int tot_score;
    cout<<"Enter register number: ";
    cin>>s.reg_no;
    fflush(stdin); //കൈമോർഡ് സഹാ ശുന്നമാക്കാൻ
    cout<<"Enter name: ";
    gets(s.name);
```



```
cout<<"Enter scores in CE, PE and TE: ";
cin>>s.ce>>s.pe>>s.te;
tot_score=s.ce+s.pe+s.te;
cout<<"\nRegister Number: "<<s.reg_no;
cout<<"\nName of Student: "<<s.name;
cout<<"\nCE Score: "<<s.ce<<"\nPE Score: "<<s.pe
    <<"\nTE Score: "<<s.te;
cout<<"\nTotal Score      : "<<tot_score;
return 0;
}
```

പ്രോഗ്രാം 1.1 എൻ ഒന്റ്‌പുട്ട് താഴെ ചേർക്കുന്നു:

### ഒന്റ്‌പുട്ട്:

```
Enter register number: 23545
Enter name: Deepika Vijay
Enter scores in CE, PE and TE: 19   38   54
```

```
Register Number: 23545
Name of Student: Deepika Vijay
CE Score: 19 PE Score: 38     TE Score: 54
Total Score      : 111
```

പ്രോഗ്രാം 1.1 ലെ സ്ട്രക്ചർ നിർവ്വചിച്ചിരിക്കുന്നത് main() ഫലങ്ങൾ പുറത്താണ്. ഇതുവേണമെങ്കിൽ main() ഫലങ്ങൾ അകത്തും ചെയ്യാം. നിർവ്വചനത്തിന്റെ സ്ഥാനം സ്ട്രക്ചർിന്റെ പരിധിയും (scope) ആയുണ്ടാം (life) നിർണ്ണയിക്കുന്നു. നമ്മൾ 11-ാം ക്ലാസിലെ 10-ാം അധ്യായത്തിൽ പഠിച്ച ലോകത്തെ, ഗ്രോബൽ ഡാറ്റയുടെ സാധ്യതകൾ അതായത് സ്ട്രക്ചർ നിർവ്വചനം main() എൻ ഉള്ളിലായാൽ ആ സ്ട്രക്ചർ പിന്നീട് main() എൻ ഉള്ളിൽ മാത്രമേ വേരിയബിള്ളുകൾ നിർമ്മിച്ച് ഉപയോഗിക്കാൻ കഴിയു. എന്നാൽ സ്ട്രക്ചർ നിർവ്വചനം main() ന് പുറത്താണെങ്കിൽ ആ സ്ട്രക്ചർ പിന്നീട് ഫലങ്ങൾ പുറത്താണെങ്കിൽ ഉപയോഗിക്കാം.



പ്രോഗ്രാം 1.1 fflush() എന്ന ഒരു ഫലങ്ങൾ gets() ഉപയോഗിക്കുന്നതിന് മുമ്പായി ഉപയോഗിച്ചിട്ടുണ്ട്. ഇതിനു കാരണം പ്രോഗ്രാമിൽ നമ്മൾ gets() ന് ശേഷം മറ്റാരു ഇൻപുട്ട് നൽകുകയാണെങ്കിൽ, തൊട്ടുമുന്ന് gets() ലെ സ്ട്രീംഗിന്റെ റീഡിംഗ് അവസാനിപ്പിക്കാൻ നാം നൽകിയ '\n' ക്രാക്കർ (ന്യൂലൈൻ) ഇൻപുട്ട് ബഹിരിൽ തന്നെ തണ്ടിനിൽക്കും. ഇതായിരിക്കും അടുത്ത ഇൻപുട്ട് ഡാറ്റയായി സീകരിക്കപ്പെടുക. ഈ പ്രസ്താവനം ഒഴിവാക്കാൻ fflush() ഫലങ്ങനിലും ഇൻപുട്ട് ബഹിരിൽ ശുന്നമാക്കിയ ശേഷം മാത്രമേ അടുത്ത ഇൻപുട്ട് നൽകാൻ പട്ടാളം.

## 1. സ്റ്റ്രക്ചറും പോയിസ്റ്റും

പ്രോഗ്രാം 1.1 തെ ഒരു സ്റ്റ്രക്ചർ വേരിയബിൾ മാത്രമെ ഉപയോഗിച്ചിട്ടുള്ളു. അതിനാൽ പ്രോഗ്രാമിൽ ഒരു കൂട്ടിയുടെ മാർക്ക് വിവരങ്ങൾ മാത്രമെ കൈകാര്യം ചെയ്യാൻ കഴിയും. കൂടുതൽ കൂട്ടികളുടെ വിവരങ്ങൾ കൈകാര്യം ചെയ്യാൻ നമ്മൾ സ്റ്റ്രക്ചർ വേരിയബിൾ ആരെ (array) ഉപയോഗിക്കേണ്ടതായി വരും. നമുക്ക് മറ്റാരു ഉദാഹരണത്തിലും സ്റ്റ്രക്ചറുകളുടെ ആരെ പരിചയപ്പെട്ടാം.

പ്രോഗ്രാം 1.2 തെ ഒരു കൂട്ടം വിൽപ്പനകാരുടെ വിവരങ്ങൾ സീക്രിക്കൗൺ. സെയിൽസ്മാൻ കോഡ്, പേര്, 12 മാസത്തെ സെയിൽസ് ഡാറ്റ എന്നിവയാണ് സ്റ്റ്രക്ചറിലെ അടിസ്ഥാന വിവരങ്ങൾ. ഈ പ്രോഗ്രാം നൽകിയ വിവരങ്ങളുടെ കൂടെ, ഓരോ വിൽപ്പനകാരരെയും ശരാശരി വിൽപ്പന എത്രയാണെന്നും പ്രദർശിപ്പിക്കുന്നു. ഈ പ്രോഗ്രാമിൽ ഉപയോഗിക്കുന്ന സ്റ്റ്രക്ചറിൽ ഒരു ഫോളിറ്റിംഗ് പോയിസ്റ്റ് ആരെ ഉപയോഗിച്ചിട്ടുണ്ട്.

### Program 1.2: വിൽപ്പനകാരരെ ശരാശരി വിൽപ്പന കാണാൻ

```
#include <iostream>
#include <cstdio>
#include <iomanip> //setw() എന്ന ഫെംഷൻ ഉപയോഗിക്കാൻ
using namespace std;
struct sales_data
{
    int code;
    char name[15];
    float amt[12]; //12 മാസത്തെ വിൽപ്പന സംഖ്യ സൂക്ഷിക്കാൻ ഉള്ള ആരെ
    float avg;
};
int main()
{
    sales_data s[20]; //സ്റ്റ്രക്ചറിൽ ആരെ
    short n,i,j; //short മെണ്ണി ഉപയോഗം കുറയ്ക്കുന്നു
    float sum;
    cout<<"Enter the number of salesmen: ";
    cin>>n;
    for(i=0; i<n; i++)
    {
        cout<<"Enter details of Salesman "<<i+1;
        cout<<"\nSalesman Code: ";
        cin>>s[i].code;
        fflush(stdin);
        cout<<"Name: ";
        gets(s[i].name);
        cout<<"Amount of sales in 12 months: ";
        for(sum=0,j=0; j<12; j++)
            sum+=s[i].amt[j];
        s[i].avg = sum/12;
    }
}
```



```
{  
    cin>>s[i].amt[j];  
    sum=sum+s[i].amt[j];  
}  
s[i].avg=sum/12;  
}  
cout<<"\t\tDetails of Sales\n";  
cout<<"Code\t\tName\t\tAverage Sales\n";  
for(i=0;i<n;i++)  
{  
    cout<<setw(4)<<s[i].code<<setw(15)<<s[i].name;  
    for (j=0;j<12;j++)  
        cout<<setw(4)<<s[i].amt[j];  
    cout<<s[i].avg<<'\n';  
}  
return 0;  
}
```

നിങ്ങൾ ഈ പ്രോഗ്രാം ലാബിൽ ചെയ്ത നോക്കി അതിന്റെ ഒരുപുട്ട് നിരീക്ഷിക്കുക. ഈ പ്രോഗ്രാമിൽ സ്ട്രക്ചർ അംഗമായി ഫ്ളോട്ടിംഗ് പോയിന്റ് അറ വേരിയബിൾ ഉപയോഗിച്ചിട്ടുണ്ട്. int ഡാറ്റാടേപ്പ് 4 ബൈറ്റ് ഉപയോഗിക്കുന്നതിനാൽ n, i and j എന്നീ വേരിയബിളുകൾ short ഡാറ്റ ടേപ്പുപയോഗിച്ചാണ് ഈ പ്രോഗ്രാമിൽ പ്രവ്യാഹിച്ചിരിക്കുന്നത്.

#### 1.1.4 നെസ്റ്റഡ് സ്ട്രക്ചർ

ഒരു സ്ട്രക്ചർ അംഗം മറ്റാരു സ്ട്രക്ചർ വേരിയബിൾ ആവാം. ഈ അനേകണ്ണുള്ള സ്ട്രക്ചർ ഒന്നെല്ലാം നെസ്റ്റഡ് സ്ട്രക്ചർ (nested structure) എന്ന് വിളിക്കുന്നു. ഈ ആശയം വളരെ ശക്തമായ ഡാറ്റ സ്ട്രക്ചറുകൾ നിർമ്മിക്കാൻ സഹായകമാണ്. നമ്മൾക്ക് പ്രവേശന തീയതി (Admission date) കുടി student സ്ട്രക്ചറിൽ ഉൾപ്പെട്ടു തന്നെമെങ്കിൽ, പട്ടിക 1.2 ത്ത് നൽകിയിരിക്കുന്ന നിർവ്വചനങ്ങളിൽ (Definition A, Definition B) ഏതെങ്കിലും ഒന്ന് ഉപയോഗിക്കാം.



പട്ടിക 1.2 : രണ്ട് തരം സ്ട്രക്ചറുകൾ

പട്ടിക 1.2 ലെ ഡൈഫോൻഡ് A യിൽ രണ്ട് സ്ട്രക്ചറുകളും വെവ്വേറെ നിർവ്വചിച്ചിട്ടുണ്ട്. student എന്ന സ്ട്രക്ചർ dt\_adm എന്ന സ്ട്രക്ചർ വേരിയബിളീനെ അംഗമാക്കിയിട്ടുണ്ട്. ഈത് date തരത്തിലുള്ള സ്ട്രക്ചർ വേരിയബിളാണ്.

അകത്തുള്ള സ്ട്രക്ചർ സെറ്റിംഗ് നടത്തുന്നതിന് മുമ്പ് നിർവ്വചിക്കണം എന്നത് ഇവിടെ നമ്മൾ ഉറപ്പ് വരുത്തേണ്ടതാണ്. എന്നാൽ ഡൈഫോൻഡ് - B യിൽ സ്ട്രക്ചർ date തന്നെ സ്ട്രക്ചർ student ന് ഉള്ളിൽ നിർവ്വചിച്ചിരിക്കുന്നു. ഈ തരത്തിൽ ഉപയോഗിച്ചാൽ സ്ട്രക്ചർ date രേഖ സ്കോപ്പ് സ്ട്രക്ചർ student ന് ഉള്ളിൽ മാത്രമായി ചുരുങ്ങും. അതിനാൽ പുറത്തെവിടെയും date സ്ട്രക്ചർ ടെസ്റ്റിൽ വേരിയബിളുകൾ പ്രവ്യാഹിക്കാൻ സാധ്യമല്ല. വേണമെങ്കിൽ അകത്തുള്ള സ്ട്രക്ചർ നിർവ്വചനത്തിലെ ടാഗിൾ പേര് ഒഴിവാക്കാം. ഒരു നെറ്റുഡ് സ്ട്രക്ചർ വേരിയബിളിന് പ്രാരംഭവിലെ നൽകുന്നത് ചുവടെ ചേർക്കുന്നു.

```
student s = {4325, "Vishal", {10, 11, 1997}, 575};  
cout<<s.adm_no<<s.name;  
cout<<s.dt_adm.day<<"/"<<s.dt_adm.month<<"/"<<s.dt_adm.year;
```

ഈനർത്ഥന സ്ട്രക്ചർ എലിമെന്റുകളെ ഉപയോഗിക്കാനുള്ള രീതി ചുവടെ ചേർക്കുന്നു.

`outer_structure_varaiable.inner_structure_variable.element`



നമ്മക്ക് ചെയ്യാം

എംപ്ലോയിക്കോഡ്, പേര്, പ്രവേശനത്തീയതി, ജോലി, ശമ്പളം എന്നീ വിവരങ്ങൾ അടങ്ങിയ employee എന്ന സ്ട്രക്ചർ നിർമ്മിക്കുക.

employee ടെസ്റ്റിൽ ഉള്ള വേരിയബിളിൾ മെമ്മറി ഘടനവരച്ച് അതിൾ വലുപ്പം കണ്ണു പിടിക്കുക.

## അരൈയും സ്ട്രക്ചറും തമിലുള്ള താരതമ്യം

ഒരു പേരിലുണ്ട് അനേകം ഡാറ്റയെ സൂചിപ്പിക്കാനുപയോഗിക്കുന്ന ഡാറ്റാ തരങ്ങളാണ് സ്ട്രക്ചറും, അരൈയും. എന്നാൽ അവ ചില കാര്യങ്ങളിൽ വ്യത്യസ്തമാണ്. അവയെ കുറിച്ചുള്ള താരതമ്യം ചുവടെ പട്ടികയിൽ നൽകിയിരിക്കുന്നു.

അരൈ	സ്ട്രക്ചർ
<ol style="list-style-type: none"> <li>ഇത് ഒരു രൂപീകൃത ഡാറ്റാഇന്റെ ഫോം ആണ് (Derived data type)</li> <li>ഒരു മന്ത്രം ഡാറ്റയുടെ കൂട്ടമാണ്.</li> <li>അരൈയിലെ അംഗങ്ങളെ സൂചിപ്പിക്കുന്നത് സബ്സ്ക്രൈറ്റ് ഉപയോഗിച്ചാണ്.</li> <li>അരൈയിലെ ഒരു അംഗം മറ്റൊരു അരൈയെ പ്രതിനിധാനം ചെയ്യുന്നോൾ മൾട്ടി ഡയമൻഷണൽ അരെ രൂപീകരിക്കപ്പെട്ടു നും.</li> <li>സ്ട്രക്ചറിന്റെ അരെ നിർമ്മിക്കൽ സാധ്യമാണ്.</li> </ol>	<ol style="list-style-type: none"> <li>ഇത് ഒരു ഉപയോക്തയുടെ നിർവ്വചിത്ര ഡാറ്റാഇന്റെ ഫോം.</li> <li>വിവിധതരം ഡാറ്റകളുടെ കൂട്ടമാണ്.</li> <li>സ്ട്രക്ചറിലെ അംഗങ്ങളെ സൂചിപ്പിക്കുന്നത് ഡോട്ട് ഓപ്പറേറ്റ് (.) ഉപയോഗിച്ചാണ്.</li> <li>സ്ട്രക്ചറിലെ ഒരു അംഗം മറ്റൊരു സ്ട്രക്ചറിനെ പ്രതിനിധാനം ചെയ്യുന്നു എങ്കിൽ അത് നേരം സബ്സ്ക്രൈറ്റ് സ്ട്രക്ചറിനെ രൂപീകരിക്കുന്നു.</li> <li>സ്ട്രക്ചറിലെ അംഗമായി അരെ ഉപയോഗിക്കാം.</li> </ol>

പട്ടിക 1.3: അരൈയും സ്ട്രക്ചറും തമിലുള്ള താരതമ്യം

### നിങ്ങളുടെ പുരോഗതി അറിയാം

- സ്ട്രക്ചർ എന്നാൽ എന്ത്?
- സ്ട്രക്ചർ വിവിധതരം ഡാറ്റകളെ ഒരു യൂണിറ്റായി എക്കോപ്പിപ്പിക്കുന്നു - ശരി/തെറ്റ്.
- സ്ട്രക്ചർ അംഗത്വത്തെ ലഭിക്കുന്നതിന് താഴെ കൊടുത്തവയിൽ എത്രാണ് ശരി.
  - struct.element
  - structure\_tag.element
  - structure\_variable.element
  - structure\_tag.structure\_variable
- നേരുള്ള സ്ട്രക്ചർ എന്നാൽ എന്ത്? ഉദാഹരണമെഴുതുക.
- അരൈയ്ക്ക് സബ്സ്ക്രൈപ്റ്റ് ആബേജിക്കിൽ സ്ട്രക്ചറിന് \_\_\_\_\_ ആണ്.

### 1.2 പോയിന്റുകൾ

നമ്മൾ അധ്യാർഥസ്ഥ കമ്പ്യൂട്ടിംഗ് എന്ന വിഷയത്തിൽ ഒരു അഭ്യന്തരമെന്തെന്ന് പേപ്പർ തയാറാക്കുകയാണ് എന്ന് കരുതുക. നമുക്ക് അനുഭ്യവാജ്ഞാനിക്കൾ നാം പരിശോധിക്കേണ്ടതായി വരും. തീർച്ചയായും നമ്മൾ ലൈബ്രേറിയിൽ പരതി നോക്കും. നമ്മൾക്ക് പുസ്തകം കണ്ണു പിടിക്കാൻ കഴിഞ്ഞില്ലെങ്കിൽ ലൈബ്രേറിയനോ, കമ്പ്യൂട്ടർ

## 1. സ്റ്റടക്കരും പോയിര്ന്നും

സയൻസ് അധ്യാപകനോ നമ്മളെ ഇക്കാര്യത്തിൽ സഹായിക്കും. ഇവിടെ നമുക്ക് ലൈബ്രേറിയൻ്റെ/അധ്യാപകൻ്റെ പങ്ക് താഴെന്ന് പരിശോധിക്കാം. അദ്ദേഹം എല്ലായ്പ്പോഴും ഒരു റഫറൻസാണ്. ലൈബ്രേറിയിലെവിടേയോ ഉള്ള പുസ്തകം കണ്ടെത്തി നമുക്ക് തരാൻ അദ്ദേഹം സഹായിക്കുന്നു.



ചിത്രം 1.4: റഫറൻസിന്റെ ഉദാഹരണം

മുകളിൽ കൊടുത്ത ഉദാഹരണത്തിലെ ലൈബ്രേറിയൻ്റെ/അധ്യാപകൻ്റെ പങ്കാണ് പോയിര്ന്നർ വേരിയബിളിനും ഉള്ളത്. പോയിര്ന്നർ കമ്പ്യൂട്ടർ മെമ്മറിയിലേക്ക് ഉള്ള ഒരു റഫറൻസ് ആണ്. താഴെ കൊടുത്ത വാചകം പരിശീലിക്കുക. ചിത്രം 1.4 : റഫറൻസിന്റെ ഉദാഹരണം.

```
int num=25;
```

ഈ ഒരു പ്രാരംഭവിലന്തികൾ പ്രസ്താവന യാണെന്ന് നമുക്കറിയാം. ഇതിൽ നിന്ന് എന്ന വേരിയബിൾ പ്രവൃത്തിക്കുന്നതോടൊപ്പം

അതിലേക്ക് 25 എന്ന സംഖ്യ ശേഖരിക്കുകയും ചെയ്യുന്നു. ഇവിടെയുള്ള മെമ്മറി നീക്കി വയ്ക്കലിന്റെ ചിത്രം 1.5 ത്ത് കാണിച്ചിരിക്കുന്നു. ചിത്രത്തിൽ നിന്നും നമുക്ക് വേരിയബിളിന്റെ മുന്ന് സാഡാവ ഘടകങ്ങളെ മനസ്സിലാക്കാം - പേര്, വിലാസം (Address), ഉള്ളടക്കം (Content) എന്നിവയാണ് ആ

ഘടകങ്ങൾ. ഇവിടെ വേരിയബിളിന്റെ പേര് num, num അതിന്റെ ഉള്ളടക്കം 25. num ഒരു int ടെപ്പ് വേരിയബിളിയാണ്. ഇതിനു വേണ്ടി 4 ബൈറ്റ് 25 1001 1002 1003 1004

മെമ്മറിയാണ് ഉപയോഗിക്കപ്പെടുന്നത് (GCCയിൽ). ചിത്രം 1.5: മെമ്മറി നീക്കിവയ്ക്കൽ RAM ലെ ഓരോ സെല്ലിലും 1 ബൈറ്റ് മെമ്മറിയാണ് ഉപയോഗിക്കുന്നത്. ഓരോ സെല്ലിനും വ്യത്യസ്തമായ അടയാളം ഉണ്ട്. ഓനിലാഡികം സെല്ലുകൾ ചേർന്ന് ഒരു സ്റ്റാറ്റിക്ക് ലോക്കേഷൻ ആയി ഉപയോഗിക്കുന്നോൾ അതിനെ മെമ്മറി വേഡ് എന്നു പറയുന്നു. 1001, 1002, 1003, 1004 എന്നീ മെമ്മറി സ്ഥാനങ്ങൾ (Locations) വേരിയബിൾ ഉപയോഗിക്കുന്നു. ഇതിൽ തുടക്കത്തിലെ സെല്ലിന്റെ വിലാസമായ (ബേസ് അടയാള്) 1001 ആണ് നമ്മൾ num എന്ന വേരിയബിളിന്റെ വിലാസമായി പരിശീലിക്കുന്നത്. 11-ാം ഫ്ലാറ്റിൽ നമ്മൾ പരിച്ച് ഒരു കാര്യം ഓർമ്മയുണ്ടോ, ഒരു വേരിയബിളിന് L വില, R വില എന്നിങ്ങനെ രണ്ട് വിലകൾ ഉണ്ട്. ഇതിൽ L വില വിലാസത്തെയും, R വില ഉള്ളടക്കത്തെയും പ്രതിനിധിക്കാം. ചിത്രം 1.5 ത്ത് num എന്ന വേരിയബിളിനു L വില 1001 ഉം R വില 25 ഉം ആണ്.

നമുക്ക് ഒരു വേരിയബിളിന്റെ വിലാസം (L-value) മറ്റാരു വേരിയബിളിൽ ശേഖരിക്കണം എന്ന് കരുതുക. ഇതിനുപയോഗിക്കുന്ന വേരിയബിളിനെ പോയിര്ന്നർ വേരിയബിൾ എന്നു പറയുന്നു. ഇപ്രകാരം നമുക്ക് പോയിര്ന്നിനെ (Pointer) മെമ്മറി വിലാസം സുക്ഷിക്കുന്ന ഒരു വേരിയബിളായി നിർവ്വചിക്കാം. പോയിര്ന്നർ ഒരു അടിസ്ഥാന ആശയമാണ്. കാരണം ഈ മെമ്മറി വിലാസത്തിനെ അതിന്റെ ധാരാർമ്മ രൂപത്തിൽ (Atomic) ഉൾക്കൊള്ളുന്നു. അതുകൊണ്ട് പോയിര്ന്നർ എന്നാൽ മെമ്മറി സ്ഥാനത്തെക്ക്-അതിന്റെ ഉള്ളടക്കത്തിലേക്ക് - ചൂണ്ടുന്ന ഒരു വേരിയബിളായി നമുക്ക് പറയാം.



നമൾ ഒരു വേരിയബിളിവേൾ മെമ്മറി വിലാസം പ്രദർശിപ്പിച്ചു നോക്കിയാൽ ഫൈലുകൾക്കും സംഖ്യകളും 0x7ffe69878784, 0x7ffe69878785 തുടങ്ങിയ വരയായിരിക്കും ലഭിക്കുക. ഇതിനുകാരണം ഈക്കാലത്ത് മെമ്മറി വിലാസം സൂചിപ്പിക്കപ്പെടുന്നത് ഫൈലുകൾക്കും സംഖ്യകളും എന്നതാണ്. എന്നാൽ ഈ പുന്തക്കത്തിൽ സാകരുത്തിനായി മെമ്മറി വിലാസങ്ങൾ 1001, 1002 തുടങ്ങിയ ഡെസിമൽ പുർണ്ണസംഖ്യകളുപയോഗിച്ച് രേഖപ്പെടുത്തിയിരിക്കുന്നു (ചിത്രം 1.5 ശ്രദ്ധിക്കുക).



**1964** ലെ പോയിന്ററിൽ കണ്ടു പിടുത്തം അറിയപ്പെടുന്നത് ഫൈലോഫീൽ ഫോസർ (ജനിച്ചത് 1937) എന്ന സോഫ്റ്റ് വെയർ എഞ്ചിനീയറുടെ (ഇദ്ദേഹം ഒരു സിസ്റ്റം എഞ്ചിനീയറും കംപ്യൂട്ടർ ആർക്കിടെക്ചർമായിരുന്നു) പേരിലാണ്. 2000 തീയിൽ, ഫോസർ ഈ കണ്ടുപിടുത്തത്തിൽ പേരിൽ IEEE യുടെ കംപ്യൂട്ടർ പത്രികയർ അവാർഡ് നേടുകയുണ്ടായി.



പ്രോഗ്രാം നിർദ്ദേശങ്ങളും, വേരിയബിളിവേൾ വിലകളും ശേഖരിക്കാനാണ് കംപ്യൂട്ടർ അതിരേൾ മെമ്മറി ഉപയോഗിക്കുന്നത് എന്ന് നിങ്ങൾക്കരിയാം. മെമ്മറി എന്നത് ചിത്രം 1.6 കാണിച്ചതുപോലെ ദ്രോഡേജ് സെല്ലുകളുടെ തുടർച്ചയായ ഒരു ശേഖരമാണ്, സാധാരണമായി ഓരോ സെല്ലിനും ഒരു ബൈറ്റ് വലുപ്പമാണ് ഉള്ളത്. ഓരോ സെല്ലിനും അതിന്റെതായ മെമ്മറി വിലാസം ഉണ്ടായിരിക്കും. വിലാസങ്ങൾ പുജ്യത്തിൽ തുടങ്ങി തുടർച്ചയായി നന്ദി ചെയ്തിരിക്കും. അവസാനത്തെ മെമ്മറിസെല്ലിവേൾ വിലാസം മെമ്മറിയുടെ വലുപ്പത്തെ ആശയിച്ചിരിക്കും.  $64 \text{ K} (64 \times 1024 = 65536 \text{ Bytes})$  മെമ്മറി വലുപ്പം ഉള്ള ഒരു കംപ്യൂട്ടറിലേ മെമ്മറിയുടെ അവസാനത്തെ വിലാസം  $65,535$  ആയിരിക്കും.

Memory Cell	Address
	0
	1
	2
	3
	4
:	:
:	:
:	:
:	:
	65535

ചിത്രം 1.6: മെമ്മറിയുടെ പദ്ധതം

നമൾ ഒരു വേരിയബിളിവേൾ പ്രവ്യാഹിക്കുന്നോൾ അഥവാ വേരിയബിളിവേൾ വേണ്ടി മെമ്മറിയിലെവിടെയോ ഒരു സ്ഥലം സൃഷ്ടിക്കപ്പെടുന്നു. ഈ സ്ഥലത്താണ് അതിരേൾ R വില (ഉള്ളടക്കം) ശേഖരിക്കപ്പെടുന്നത്. ഓരോ വേരിയബിളിനും അതിന്റെതായ വിലാസം ഉണ്ടായിരിക്കും. ഈക്കാലത്ത് RAMരേൾ വലുപ്പം GBയിലും, മെമ്മറി സ്ഥാനത്തിരേൾ വിലാസം ഫൈലുകൾക്കും നന്ദി ചെയ്യുന്നതു അഥവാ രേഖപ്പെടുത്തുന്നതു. ഇതിനുകാരണം ഡെസിമൽ നന്ദി നിലനിൽക്കുന്നതിനും കുറഞ്ഞ അക്കങ്ങൾ ഉപയോഗിച്ച് വലിയ സംഖ്യകൾ രേഖപ്പെടുത്തൽ കഴിയും എന്നതാണ്.

### 1.2.1 പോയിന്റർ വേരിയബിളിവേൾ പ്രവ്യാഹനം (Declaration)

പോയിന്റർ ഒരു രൂപീകൃത ഡാറ്റാതരമാണ് (Derived data type). അതുകൊണ്ടു തന്നെ പ്രോഗ്രാമിൽ ഉപയോഗിക്കുന്നതിന് മുമ്പ് പോയിന്റർ വേരിയബിളികളെ പ്രവ്യാഹിക്കേണ്ടതായിട്ടുണ്ട്. താഴെ കൊടുത്ത വാക്കും പോയിന്റർ വേരിയബിൾ പ്രവ്യാഹിക്കുന്നത്.

```
data_type * variable;
```

## 1. സ്ട്രക്ചറും പോയിന്ററും

ഇവിടെ ഡാറാതരം എന്നത് അടിസ്ഥാന ഡാറാതരമോ, ഉപയോകത്യ നിർവചിത ഡാറാതരമോ ആവാം. variable എന്നാൽ ഐഡിഎസ്റ്റിഫയർ ആണ്. ഡാറാതരത്തിനും വേറയബിളിനും ഇടയിലുള്ള asterisk (\*) ചിഹ്നം ശ്രദ്ധിക്കുക. ഈത് പോയിന്റർ വേറയബിളിന്റെ പ്രവ്യാപനത്തിലെ പ്രത്യേകതയാണ്. ചുവടെ ചേർത്തവ പോയിന്റർ വേറയബിൾ പ്രവ്യാപനത്തിന്റെ ഉദാഹരണങ്ങളാണ്.

```
int *ptr1;
float *ptr2;
struct student *ptr3;
```

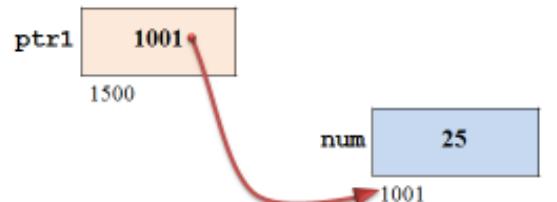
സാധാരണ പോലെ പോയിന്റർ വേറയബിളുകൾക്കും മെമ്മറി നീക്കിവയ്ക്കൽ നടക്കുന്നു. പോയിന്ററിന് ആവശ്യമായ മെമ്മറി സ്ഥാനത്തിന്റെ വലുപ്പം ഡാറാതരത്തിന് അനുസരിച്ചാണെന്നു നിങ്ങൾ വിചാരിക്കുന്നുണ്ടോ? നമുക്കുണ്ടാം മെമ്മറി വിലാസത്തിന്റെ തരം അണ്ണബൈസിൽ ഇൻഡിക്ഷൻ ആണ്. ഇതിനർമ്മം എല്ലാ പോയിന്ററും അണ്ണബൈസിൽ ഇൻഡിക്ഷൻ തരത്തിൽ പ്രവ്യാപിക്കണം എന്നല്ല. പിന്നെ എന്നാണ് പോയിന്ററിന്റെ ഡാറാതരം തീരുമാനിക്കുന്നതിലെ മാനദണ്ഡം? ഒരു പോയിന്റർ വേറയബിളിന്റെ ഡാറാതരം അത് പോയിന്റ് ചെയ്യുന്ന ഡാറായുടെ തരം തന്നെയാണ്. മുകളിൽ കൊടുത്ത ഉദാഹരണത്തിൽ ptr1 എന്ന വേറയബിളിൽ ഒരു ഇൻഡിക്ഷൻ മെമ്മറി സ്ഥാനത്തിന്റെ വിലാസമാണ് സൂക്ഷിക്കപ്പെടുന്നത്. ഈത് പോലെ ptr2 ഒരു എല്ലാട്ടിൽ പോയിന്റ് ലോക്കേഷൻ്റെയും ptr3 student തരത്തിലുള്ള സ്ട്രക്ചർറിന്റെയും വിലാസം ഉൾക്കൊള്ളുന്നു. കൂടുതൽ മെമ്മറി സ്ഥാനങ്ങൾ ഉള്ള ഡാറാതരങ്ങളെ പോയിന്റ് ചെയ്യുന്നോൾ അതിന്റെ തുടക്കത്തിലെ ലോക്കേഷൻ്റെ വിലാസമാണ് (Base Address) പോയിന്റർ വേറയബിളിൽ ശേഖരിക്കപ്പെടുന്നത്. അപ്പോൾ ഒരു പോയിന്റർ വേറയബിളിന്റെ മെമ്മറി വലുപ്പം എന്നായിരിക്കും? ഈ വലുപ്പം കമ്പ്യൂട്ടറിലെ വിലാസം സന്ദർഭത്തിനുസരിച്ചായിരിക്കും (Addressing scheme). സാധാരണയായി C++ ത്ത് ഒരു പോയിന്റർ വേറയബിളിന്റെ മെമ്മറി വലുപ്പം 2 മുതൽ 4 ബെബ്രെ വരെ ആയിരിക്കും. ഒരു പ്രോഗ്രാമറെ സംബന്ധിച്ചിടതോളം പ്രശ്നങ്ങൾ നിർജ്ജാരണം ചെയ്യുന്നോൾ പോയിന്ററിന്റെ വലുപ്പം അറിയണമെന്ന് നിർബന്ധമില്ല.

### 1.2.2 &, \* എന്നീ ഔദ്യോഗികൾ

ഒരിക്കൽ ഒരു പോയിന്റർ വേറയബിൾ പ്രവ്യാപിച്ചാൽ അതെ തരത്തിലുള്ള മെമ്മറി ലോക്കേഷൻ്റെ വിലാസം മാത്രമേ അതിൽ ശേഖരിക്കാൻ കഴിയു. സാധാരണ ഡാറാതരം C++ ത്ത് ഒരു വേറയബിളിന്റെ പേര് നൽകിയാൽ

അതിന്റെ R വില മാത്രമേ സൂചിപ്പിക്കപ്പെടുന്നുള്ളൂ. അപ്പോൾ എങ്ങനെയാണ് L വില അമ്പവാ വിലാസം ലഭിക്കുന്നത്? C++ ത്ത് ഇതിനുവേണ്ടി അഡ്യസ്സ് ഓപ്പറേറ്റർ (&) ഉപയോഗിക്കുന്നു. ഉദാഹരണമായി num ഒരു ഇൻഡിക്ഷൻ വേറയബിളാണെങ്കിൽ അതിന്റെ വിലാസം താഴെ കൊടുത്ത രീതിയിൽ ptr1 എന്ന പോയിന്ററിൽ ശേഖരിക്കാം.

```
ptr1 = &num;
```



പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുന്ന സമയത്ത് ഈ വാചകം `ptr1` എന്ന പോയിന്റർ വേരിയബിളിനും സ്ഥാപിച്ചാൽ മെമ്മറി സ്ഥാനത്തിനും ഇടയിൽ ഒരു ബന്ധം സ്ഥാപിക്കുന്നു. ഈ ചിത്രം 1.7 തീ കാണിച്ചിരിക്കുന്നു.

പോയിന്റർ ഒരു റഫറൻസ് ആണെന്നു നാം മനസ്സിലാക്കി. അതുകൊണ്ട് മെമ്മറിയിൽ എവിടെ ഡാറ്റ ശേഖരിച്ചാലും അത് സൂചിപ്പിക്കാൻ (refer) പോയിന്റർ വേരിയബിളിനു കഴിയും. പോയിന്റർ വേരിയബിളിന്റെ ഡൈറക്ഷൻസിലൂടെ നമുക്ക് ഡാറ്റ ലഭ്യമാവും. ഈ കൂടുതലും ഇൻഡിക്കേഷൻ ഓപ്പറേറ്റർ (indirection operator) അഥവാ (\*) ഉപയോഗിക്കുന്നു. ഇതിനെ ഡൈറക്ഷൻസിലൂടെ ഓപ്പറേറ്റർ (dereference operator) എന്നും പറയുന്നു. താഴെ കൊടുത്ത C++ പ്രസ്താവന പ്രോഗ്രാം എന്ന പോയിന്റർ വേരിയബിൾ ചുണ്ടുന്ന ലോക്കേഷൻിലെ വില പ്രദർശിപ്പിക്കുന്നു.

```
cout << *ptr1;
```

ഈ പ്രസ്താവന `cout << num;` എന്ന പ്രസ്താവനയ്ക്ക് തുല്യമാണെന്ന് വ്യക്തമാണല്ലോ.

\* ഓപ്പറേറ്റർ ഒരു പോയിന്റർ ലോക്കേഷൻിലെ വില തിരിച്ച് തരുന്നതു കൊണ്ട് ‘വാല്യു അറ്റ് ഓപ്പറേറ്റർ’ എന്നും അറിയപ്പെടുന്നു.

& ഓപ്പറേറ്ററും \* ഓപ്പറേറ്ററും യുനി ഓപ്പറേറ്ററുകളുമാണ്. എല്ലാ വേരിയബിളിനും മെമ്മറി സ്ഥാനവും വിലാസവും ഉള്ളടക്കാത്ത & ഓപ്പറേറ്റർ എല്ലാ വേരിയബിളിന്റെയും കൂടെ ഉപയോഗിക്കാം എന്നാൽ \* ഓപ്പറേറ്റർ പോയിന്റർ വേരിയബിളിന്റെ കൂടെ മാത്രമേ ഉപയോഗിക്കാൻ കഴിയുകയുള്ളൂ.

ചിത്രം 1.7 തീ കൊടുത്ത വേരിയബിൾ പരിശോധിക്കു. ചുവടെ കൊടുത്ത ഉദാഹരണങ്ങൾ &, \* ഓപ്പറേറ്ററുകളുടെ ഉപയോഗം കാണിക്കുന്നു.

```
cout<< &num; // 1001 num ഏ വിലാസം പ്രദർശിപ്പിക്കുന്നു
cout<< ptr1; // 1001 ptr1 ഏ ഉള്ളടക്കം പ്രദർശിപ്പിക്കുന്നു
cout<< num; // 25 num ഏ ഉള്ളടക്കം പ്രദർശിപ്പിക്കുന്നു
cout<< *ptr1; /* 25 ptr1 പോയിന്റ് ചെയ്യുന്ന സ്ഥാനത്തെ വില പ്രദർശിപ്പിക്കുന്നു
cout<< &ptr1; // 1500 (ptr1 ഏ വിലാസം പ്രദർശിപ്പിക്കുന്നു.
cout<< *num; // Error!! num എന്നത് പോയിന്റർ വേരിയബിളില്ല.
```

അവസാനത്തെ പ്രസ്താവന തെറ്റാണ്. കാരണം സ്ഥാപിച്ചാൽ പോയിന്റർ വേരിയബിളില്ല. അതിലുള്ള 25 എന്ന വില ഒരു മെമ്മറിവിലാസവും അല്ല. \* ഓപ്പറേറ്റർ പോയിന്റർ വേരിയബിളിനോടു കൂടി മാത്രമേ ഉപയോഗിക്കാൻ പാടുള്ളൂ.

### 1.3 മെമ്മറി നീക്കിവയ്ക്കലിന്റെ (Allocation) രീതികൾ

വേരിയബിൾ ഡൈറക്ഷൻ ഫ്ലോറേംഗ്ലിലുടെയാണ്, മെമ്മറി അലോക്കേഷൻ തുടങ്ങുന്നത് എന്ന നമുക്കൾക്കാം. പ്രോഗ്രാം റാമിൽ ലോഡ് ചെയ്യുന്നോൾ ആവശ്യമുള്ള മെമ്മറിയും നീക്കിവയ്ക്കപ്പെടുന്നു. മെമ്മറി നീക്കിവയ്ക്കലിന് ശേഷമാണ് പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കൽ ആരംഭിക്കുന്നത്. മെമ്മറി നീക്കി വയ്ക്കുന്നതിൽ അളവ് പ്രോഗ്രാമിൽ ഉപയോഗിച്ചിരിക്കുന്ന വേരിയബിളുകളുടെ എല്ലാത്തയ്ക്കും അനുസരിച്ചാണ്. ഈ അളവ് സ്റ്റിറ്റിംഗ് (static) ആണ്. ഈ അളവും പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുന്ന സമയത്ത്

കൂടുകയോ കുറയുകയോ ഇല്ല. ഈഅനെ പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുന്നതിന് മുമ്പ് നടക്കുന്ന മെമ്മറി നീക്കിവയ്ക്കലിനെ സ്റ്റാറ്റിക് മെമ്മറി അലോക്കേഷൻ (Static Memory Allocation) എന്ന് പറയുന്നു. ഈ നീക്കിവയ്ക്കൽ നടക്കുന്നത് പ്രോഗ്രാമിലെ വേരിയബിൾ പ്രവൃത്തം പ്രസ്താവനയ്ക്കുന്നുണ്ടാണ്. ഈ മറ്റാരു തരം മെമ്മറി അലോക്കേഷൻ ആയ ഡെയനാമിക് മെമ്മറി അലോക്കേഷൻ (Dynamic Memory Allocation) പതിചയഘൈപ്പ് ചൊം. ഈവിടെ മെമ്മറി നീക്കിവയ്ക്കൽ നടക്കുന്നത് പ്രോഗ്രാം പ്രവർത്തിക്കുന്നുണ്ടാണ്. ഈ നീക്കിവയ്ക്കൽ നടക്കുന്നത് new എന്ന ഒരു ഓപ്പറേറ്റർ മുവേദ്യാണ്. ഈഅനെ നീക്കിവച്ച് മെമ്മറി സ്വതന്ത്രമാക്കാൻ (Deallocate) delete എന്ന ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു.

### 1.3.1 ഡെയനാമിക് ഓപ്പറേറ്റുകൾ - `new, delete`

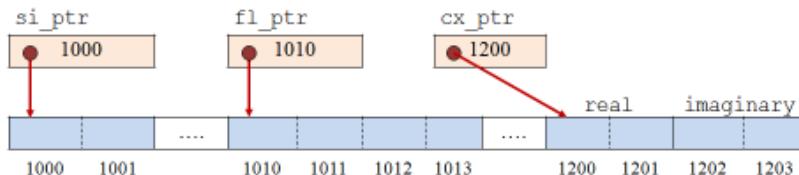
C++ ലെ നാലു എന്ന കീവോൾ ഒരു ഓപ്പറേറ്റർ ആണ്. ഈത് ഡെയനാമിക് മെമ്മറി അലോക്കേഷനുവേണ്ടി ഉപയോഗിക്കുന്നു. ഈ നീക്കിവയ്ക്കൽ നടക്കുന്നത് പ്രോഗ്രാം പ്രവർത്തിപ്പിക്കുന്ന സമയത്താണ് (ഇൻ ദെമിലാണ്). ഈത് ഒരു യുനി ഓപ്പറേറ്റർ ആണ്. അടിസ്ഥാന ഡാറ്റാ തരത്തിലെയോ ഉപയോഗത്തുനിർവ്വചിത്ര ഡാറ്റാ തരത്തിലേയോ ഒരു ഓപ്പറേറ്ററാണ് ഈതിൽ ഉപയോഗിക്കുന്നത്. new എന്ന ഓപ്പറേറ്ററും ഈ ഓപ്പറേറ്ററും കൂടിച്ചേരുന്ന് ഒരു പദ്ധത്യാഗം (Expression) ഉണ്ടാക്കുകയും. ആ പദ്ധത്യാഗം ഒരു വില തിരിച്ച് നൽകുകയും ചെയ്യുന്നു. ഒരു മെമ്മറി സ്ഥാനത്തിൽ വിലാ സമാം ആ വില. മെമ്മറി സ്ഥാനത്തിൽ വലുപ്പം ഓപ്പറേറ്ററിൽ ഡെയനാമിക് അലോക്കേഷൻ താഴെ കൊടുത്ത വാക്യം ഉപയോഗിക്കുന്നു.

```
pointer_variable = new data_type;
```

പോയിറ്റർ വേരിയബിൾ ഉപയോഗിക്കുന്നത് new ഓപ്പറേറ്റർ തിരിച്ച് തരുന്ന മെമ്മറി വിലാസം സൂക്ഷിക്കാൻ വേണ്ടിയാണ്. അതുകൊണ്ട് new ഓപ്പറേറ്ററിന് ശേഷം നൽകിയ ഡാറ്റാതരത്തിൽ തന്നെ പോയിറ്റർ വേരിയബിൾ നേരത്തെ പ്രവൃത്തിപ്പിരിക്കണം.

```
short * si_ptr;
float * fl_ptr;
struct complex * cx_ptr;
si_ptr = new short;
fl_ptr = new float;
cx_ptr = new complex;
```

ഇവയുടെ മെമ്മറി നീക്കിവയ്ക്കൽ ചിത്രം 1.8 തുടർന്ന് കാണിച്ചിരിക്കുന്നു.



ചിത്രം 1.8: ഡെയനാമിക് മെമ്മറി അലോക്കേഷൻ ഉടൻ

ചിത്രം 1.8 തുടർന്ന് കാണിച്ചിരിക്കുന്നത് ശ്രദ്ധിക്കുക. ഈവിടെ 2 ബെബർ മെമ്മറി സ്ഥാനം short ദെപ്പ് ഡാറ്റയ്ക്ക് നീക്കിവയ്ക്കുന്നു. ഈതിൽ വിലാസം 1000 ആണ്, ഈതുപോലെ 1010 എന്ന വിലാസത്തിൽ 4 ബെബർ മെമ്മറി float തരം ഡാറ്റയ്ക്ക് നീക്കിവയ്ക്കുന്നു.

ഈ സുക്ഷിക്കുന്നത് f1\_ptr എന്ന പോയിന്റർ വേരിയബിളിലാണ്. മുമ്പ് നാം പരിച്ച complex എന്ന സ്ട്രക്ചർ 2 short തരത്തിലുള്ള അംഗങ്ങൾ ഉൾക്കൊള്ളുന്നു. അതുകൊണ്ട് 1200 എന്ന വിലാസത്തിൽ തുടങ്ങുന്ന 4 ബൈറ്റ് മെമ്മറി ലൊക്കേഷൻ cx\_ptr എന്ന complex സ്ട്രക്ചർ തരത്തിലുള്ള പോയിന്റർ വേരിയബിൾ സൂചിപ്പിക്കുന്നു. (short real നും short imaginary ത്രക്കും 2 ബൈറ്റ് വീതം ആകെ 4 ബൈറ്റ്). ദൈനന്ദിനിക് മെമ്മറി അലോക്കേഷൻ നടത്തിയ സഹായങ്ങൾ സാധാരണ വേരിയബിളുകൾക്ക് സൂചിപ്പിക്കാൻ കഴിയില്ല. ഈ സൂചിപ്പിക്കാൻ പോയിന്റ് വേരിയബിൾ തന്നെ ഉപയോഗിക്കണം. താഴെ കൊടുത്ത ഉദാഹരണം ശ്രദ്ധിക്കുക.

```
*si_ptr = 247;
cin >> *f1_ptr;
```

നമുക്ക് സ്ട്രക്ചർ പോയിന്റർ വേരിയബിളായ cx\_ptr ഉണ്ട്. ഈ സൂചിപ്പിക്കുന്ന ഡാറ്റ മുകളിൽ കൊടുത്ത റീതിയിൽ ഉപയോഗിക്കാൻ കഴിയില്ല. ഇതിന്റെ ഉപയോഗരീതി ഈ അധ്യായത്തിൽത്തന്നെ നമുക്ക് പിന്നീട് മനസ്സിലാക്കാം.

നമുക്ക് സ്ലാറ്റിക് മെമ്മറി അലോക്കേഷൻ വേരിയബിളുകൾക്ക് പ്രാരംഭവിലെ നൽകുന്ന നടപ്പോലെ ദൈനന്ദിനിക് മെമ്മറി അലോക്കേഷനിലും വേരിയബിളുകൾക്ക് പ്രാരംഭവിലെ നൽകും. ഈ ചുവടെ ചേർത്ത വാക്യാലടക്ക ഉപയോഗിക്കുന്നു.

```
pointer_variable = new data_type(value);
```

ചുവടെ കൊടുത്തവ ദൈനന്ദിനിക് മെമ്മറി അലോക്കേഷൻ കുടുക്കുള്ള പ്രാരംഭവിലെ നൽകവിന് ഉദാഹരണങ്ങളാണ്.

```
si_ptr = new short(0);
f1_ptr = new float(3.14);

cx_ptr ഒരു കാര്യത്തിൽ ഈ തരത്തിലുള്ള പ്രാരംഭവിലെ നൽകൽ സാധ്യമല്ല.
ഒരിക്കൽ നമ്മൾ new ഓപ്പറേറ്റർ ഉപയോഗിച്ച് ദൈനന്ദിനിക് മെമ്മറി അലോക്കേഷൻ നടത്തിയാൽ അത് സ്വതന്ത്രമാകുന്നും ഒരു താഴീടുണ്ട് അമവാ ഡീ അലോക്കേറ്റ് ചെയ്യേണ്ടതായിട്ടുണ്ട്. C++ ലെ ഇതിനുവേണ്ടി delete ഓപ്പറേറ്റിനെ ഉപയോഗിക്കുന്നു.
സ്ലാറ്റിക് മെമ്മറി അലോക്കേഷൻ കാര്യത്തിൽ വേരിയബിളിന്റെ സ്കോപ്പും, ലൈഫ് റെംബും നോക്കി ഓപ്പറേറ്റിംഗ് സിസ്റ്റം തന്നെ മെമ്മറി നൈറ്റീവ്യ്ക്കലും, സ്വതന്ത്രമാകലും നടത്തും. എന്നാൽ ദൈനന്ദിനിക് മെമ്മറി അലോക്കേഷനുമായി ബന്ധപ്പെട്ട മെമ്മറി സ്വതന്ത്രമാകാൻ ഫ്രോഗ്രാമിൽ തന്നെ delete ഓപ്പറേറ്റർ ഉപയോഗിച്ചുള്ള നിർദ്ദേശങ്ങൾ നൽകേണ്ടതാണ്. delete ഓപ്പറേറ്റിന്റെ ഉപയോഗം താഴെ നൽകിയിരിക്കുന്നു.
```

```
delete pointer_variable;
```

ചുവടെ ഉദാഹരണങ്ങൾ ചേർക്കുന്നു.

```
delete si_ptr;
delete f1_ptr, cx_ptr;
```

### 1.3.2 മെമ്മറി ലീക്ക് (Memory leak)

new ഓപ്പറേറ്റർ ഉപയോഗിച്ച് നീക്കി വച്ച മെമ്മറി ബ്ലോക്ക് delete ഓപ്പറേറ്റർ ഉപയോഗിച്ച് സ്വതന്ത്രമാക്കുന്നില്ല എങ്കിൽ അതിനെ ഓർഫൻ സ്ലോക്ക് എന്ന് പറയുന്നു. ഈ ഉപയോഗിക്കാതെ ബാക്കി വരുന്ന മെമ്മറി സ്ലോക്ക് ആണ്. പക്ഷെ ഈ വിജ്ഞാം

## 1. സ്റ്റടക്കപ്പും പോയിര്ന്നും

ഡാറ്റ സൂക്ഷിക്കാൻവേണ്ടി നീക്കിവയ്ക്കാൻ കഴിയില്ല. ഫ്രോഗ്രാമിന്റെ ഓരോ പ്രവർത്തി പ്ലിക്ക് ലില്ലും ഇത്തരത്തിലുള്ള ബ്ലോക്കുകൾ സൃഷ്ടിക്കപ്പെടുകയും മെമ്മറിയുടെ ഉപയോഗിക്കുത്തക ഭാഗം തുടർച്ചയായി കുറയുകയും ചെയ്യും. ഈങ്ങെന മെമ്മറി നഷ്ടപ്പെടുന്നതിനെ മെമ്മറി ലീക്ക് എന്ന് പറയുന്നു.

താഴെ പറയുന്നവയാണ് മെമ്മറി ലീക്കിന്റെ കാരണങ്ങൾ.

- ദൈനന്ദിനിക് അലോക്കേഷൻ നടത്തിയ മെമ്മറി (new ഓപ്പറേറ്റ് ഉപയോഗിച്ച്) സത്രന്തമാക്കാൻ മരന്നു പോകുന്നത്.
- ഫ്രോഗ്രാമിങ്ങിലെ ലോജിക്കൽ പിശവ് മുലം ‘delete’ നിർദ്ദേശം പ്രവർത്തിപ്ലിക്കാ തത്ത്.
- നിലവിൽ ഒരു മെമ്മറി ബ്ലോക്കിനെ പോയിര്ന്ന് ചെയ്യുന്ന പോയിര്ന്നിലേയ്ക്ക് new ഓപ്പറേറ്റിലുടെ ഒരു പുതിയ മെമ്മറി വിലാസം ശേഖരിക്കുന്നത്.

new ഓപ്പറേറ്റ് ഉപയോഗിച്ച് നടത്തിയ മെമ്മറി അലോക്കേഷൻ delete ഓപ്പറേറ്റ് ഉപയോഗിച്ച് സത്രന്തമാക്കുക എന്നതാണ് മെമ്മറി ലീക്കിനുള്ള പരിഹാരം. ദൈനന്ദിനിക് അലോക്കേഷൻ കാര്യത്തിൽ മാത്രമാണ് മെമ്മറി ലീക്ക് ഉണ്ടാക്കുന്നത്. സൂഡിക് അലോക്കേഷനിൽ മെമ്മറി നീക്കിവയ്ക്കലും സത്രന്തമാക്കലും ഓപ്പറേറ്റിംഗ് സിസ്റ്റം നേരിട്ട് നടപ്പിലാക്കുന്നു. ഈതിന് പ്രത്യേക നിർദ്ദേശത്തിന്റെ ആവശ്യം ഇല്ല. അതുകൊണ്ട് സൂഡിക് മെമ്മറി അലോക്കേഷനിൽ മെമ്മറി ലീക്കിനുള്ള സാധ്യതയില്ല.



**ഇപ്പോൾ നമുക്ക് സൂഡിക് മെമ്മറി അലോക്കേഷനും ദൈനന്ദിനിക് മെമ്മറി അലോക്കേഷനും താരതമ്യം ചെയ്യാം. പട്ടിക 1.4 തും ഈ താരതമ്യം കൊടുത്തിരിക്കുന്നു. ചില ഭാഗങ്ങൾ നിങ്ങൾക്ക് പുറിപ്പിക്കാൻ വേണ്ടി നമുക്ക് ചെയ്യാം വിട്ടിരിക്കുന്നു.**

സൂഡിക് മെമ്മറി അലോക്കേഷൻ	ദൈനന്ദിനിക് മെമ്മറി അലോക്കേഷൻ
i. ഫ്രോഗ്രാം പ്രവർത്തിപ്ലിക്കുന്നതിന് മുമ്പ് നടക്കുന്നു.	
ii.	new ഓപ്പറേറ്റ് ആവശ്യമുണ്ട്.
iii.	പോയിര്ന്നിൽ നിർബന്ധമാണ്.
iv. ഡാറ്റയെ വേരിയബിള്ക്കൽ ഉപയോഗിച്ച് സൃച്ചിപ്പിക്കുന്നു.	
v. ഡീ-അലോക്കേഷൻ വേണ്ടി പ്രത്യേകം ബ്ലോക്ക് ആവശ്യമില്ല.	

പട്ടിക 1.4: സൂഡിക് അലോക്കേഷനും ദൈനന്ദിനിക് അലോക്കേഷനും തമ്മിലുള്ള താരതമ്യം

### നിങ്ങളുടെ പുരോഗതി അറിയാം



1. പോയിന്റർ എന്നാൽ എന്ത്?
2. ഒരു പോയിന്ററിന്റെ ഡാറ്റാതരം നിർണ്ണയിക്കുന്നതിന്റെ മാനദണ്ഡമെന്ത്?
3. `mkd` ഒരു ഇൻഡിക്യൂട്ടർ വേരിയബിളാസൈക്കിൽ, ഈ വേരിയബിളിന്റെ വിലാസം ഒരു പോയിന്റർ വേരിയബിളിൽ ശേഖരിക്കാനുള്ള C++ വാചകം എഴുതുക.
4. `ptr` ഒരു പോയിന്റർ വേരിയബിൾ ആണ്. ഇതുപയോഗിച്ച് ഒരു ഇൻഡിക്കേറ്റർ വേണ്ടി മെമ്മറി നീകിൽ വച്ച് അതിൽ 12 എന്ന സംഖ്യ പ്രാരംഭിക്കാതെ ചേർക്കുന്നതിനുള്ള C++ വാചകം എഴുതുക.
5. ചുവടെ ചേർത്ത വാചകങ്ങൾ പരിഗണിക്കുക: `int *p, a=5; p=&a; cout<<*p+a;` ഇതിന്റെ ഒരു പുതിയ എന്താണ്?

## 1.4 പോയിന്ററിലെ ഓപ്പൺക്ഷേമങ്ങൾ

പോയിന്ററുമായി ബന്ധപ്പെട്ട ഇൻഡായറികഷൻ (\*) ഓപ്പറേറ്റർ, അഡ്രസ്സ് ഓഫ് (ഡി) ഓപ്പറേറ്റർ എന്നിവ നമ്മൾ ചർച്ച ചെയ്ത കഴിഞ്ഞല്ലോ. 11-ാം ക്ലാസ്സിൽ അതിന്തമാറ്റിക്, റിലേഷൻസിൽ, ലോജിക്കൽ എന്നീ ഓപ്പറേറ്ററുകളും പരിചയപ്പെട്ടു. ഈ ഭാഗത്ത് പോയിന്റർ വേരിയബിളിൽ ഉപയോഗിക്കുന്ന ഓപ്പൺക്ഷേമങ്ങളാണ് നാം പരിചയപ്പെട്ടാണ് പോകുന്നത്.

### 1.4.1 പോയിന്ററിലെ അരിതമാറ്റിക് ഓപ്പൺക്ഷേമങ്ങൾ

മെമ്മറി വിലാസം എന്നത് ഒരു സംഖ്യയാണെന്ന് നമുക്ക് അറിയാം. അതിനാൽ ചില അരിതമാറ്റിക് ഓപ്പൺക്ഷേമങ്ങളും നമുക്ക് പോയിന്ററിൽ വേരിയബിളിൽ പ്രയോഗിക്കാം. ഭാഗം 1.3.1 ത്ര ഉപയോഗിച്ചു `si_ptr`, `fl_ptr` എന്നീ പോയിന്റർ വേരിയബിളുകൾ പരിഗണിക്കുക. (ചിത്രം 1.8 കാണുക). ഈ ചുവടെ ചേർത്ത പ്രസ്താവനകൾ പരിശോധിക്കുക.

```
cout << si_ptr + 1;
cout << fl_ptr + 1;
```

എന്തായിരിക്കും ഒരു പുതിയ എന്ത്? 1001 ഉം 1011 ആണ് ഒരു പുതിയ എന്നതുനുണ്ടോ.

പോയിന്ററിനോട് 1 കൂടുന്നത് `int` തരത്തിലുള്ള വേരിയബിളിനോടോ `float` തരത്തിലുള്ള വേരിയബിളിനോടോ 1 കൂടുന്നത് പോലെ അല്ല. നമ്മൾ ഒരു `short int` പോയിന്ററിനോട് 1 കൂടുന്നോൾ, ആ പ്രയോഗം തൊട്ടട്ടുത്ത മെമ്മറി സ്ഥാനത്തിന്റെ വിലാസമായി മാറുന്നു. അതായത് 1000, 1001 എന്നീ ലോക്കേഷനുകൾ `short int` വേരിയബിൾ ഉപയോഗിക്കുന്ന ലോക്കേഷനുകളാണ്. 1002 എന്നത് അടുത്ത `short int` എന്റെ ബേസ് അഡ്രസ്സായിരിക്കും. അതായത് നമ്മൾ `short int` പോയിന്റർ വേരിയബിളിനോട് ഒരു കൂടുന്നോൾ `short int` എന്റെ യഥാർത്ഥ വലുപ്പം (2) ആണ് മെമ്മറി അഡ്രസ്സിനോട് കൂടുന്നത്. ഈ ഒരു പോയിന്റർ വേരിയബിളിനോട് 1 കൂടുന്നോൾ അതിന്റെ വലുപ്പമായ 4 ആണ് വിലാസത്തോട് കൂടുന്നത്. അതുകൊണ്ട് `fl_ptr+1`

## 1. സ്റ്റ്രക്കച്ചറും പോയിസ്റ്ററും

എന്ന പദ്ധത്യാഗം നൽകുന്നത് 1014 എന്ന വിലാസമാണ്. അതുകൊണ്ട്  $si\_ptr + 4$  എന്ന പദ്ധത്യാഗം 1008 ( $1000 + 4 \times 2$ ) എന്ന വിലാസം നൽകുമെന്ന് വ്യക്തമാണ്. ഈ പോലെ  $f1\_ptr + 3$  നൽകുന്നത് 1022 ( $1010 + 3 \times 4$ ) എന്ന വിലാസമാണ്. ഈ പോലെ പോയിസ്റ്റർ വേരിയബിളുകളിൽ വ്യവകലനവും നടത്താം. മറ്റ് അരിതതമാറ്റിക് ഓപ്പറേറ്റ് ഷനുകൾ ഒന്നും തന്ന പോയിസ്റ്റർ വേരിയബിളിനു മുകളിൽ പ്രവർത്തിക്കുകയില്ല. ഈ നടത്തിയില്ലെങ്കിൽ ഓപ്പറേഷനുകൾ പ്രായോഗികമായി ചിലപ്പോഴും തെറ്റായേക്കാം. കാരണം



ചുവടെ കൊടുത്തിരിക്കുന്ന അരിതതമാറ്റിക് പദ്ധത്യാഗങ്ങൾ നൽകുന്ന വിലകൾ കണ്ടുപിടിക്കുക.

- നമുക്ക് ചെയ്യാം
- a)  $si\_ptr + 10$
  - b)  $f1\_ptr + 7$
  - c)  $si\_ptr - 5$
  - d)  $f1\_ptr - 10$

ഈ ഓപ്പറേഷനിലൂടെ സുചിപ്പിക്കുന്ന മെമ്മറി സ്ഥാനങ്ങൾ ആക്സസ്സെററ്റ് പ്രകാരം ഉപയോഗിക്കാൻ അനുവാദമില്ലാത്തവയായിരിക്കും.

ചുവടെ ചേർത്തിരിക്കുന്ന ഉദാഹരണങ്ങൾ പോയിസ്റ്റർ ഓപ്പറേഷനുകൾ വിശദീകരിക്കുന്നു.

```
int *ptr1, *ptr2; // ഒങ്ക് പോയിസ്റ്റർ വേരിയബിളുകളുടെ പ്രവ്യാപനം
ptr1 = new int(5); /* ദൈനാമിക് മെമ്മറി അലോക്കേഷൻ (വിലാസം 1000
ആണന്നിരിക്കുന്നത്), 5 എന്ന സംഖ്യ പ്രാരംഭിലയായ്
ചേർന്നിരിക്കുന്നു 5*/
ptr2 = ptr1 + 1; /* ptr2 തൊടുത്ത ഇൻജർ സ്ഥാനമായ 1004 ലേക്ക്
പോയിസ്റ്റ് ചെയ്യാം */
++ptr2; // ptr2 = ptr2 + 1 എന്ന ഓപ്പറേഷൻ തുല്യം
cout << ptr1; // 1000 പ്രദർശിപ്പിക്കുന്നു
cout << *ptr1; // 5 പ്രദർശിപ്പിക്കുന്നു
cout << ptr2; // 1004 പ്രദർശിപ്പിക്കുന്നു
cin >> *ptr2; /* ഒരു ഇൻജർ രീഡ് ചെയ്ത് (12 ആണന്നിരിക്കുന്ന)
ഇതിനെ 1004 എന്ന ലോക്കേഷനിൽ ശേഖരിക്കുന്നു */
cout << *ptr1 + 1; // 6 (5 + 1) പ്രദർശിപ്പിക്കുന്നു
cout << *(ptr1 + 2); // 12 പ്രദർശിപ്പിക്കുന്നു, (1004 ലെ വില)
ptr1--;
// ptr1 = ptr1 - 1 എന്ന പ്രസ്താവനയ്ക്ക്
തുല്യം
```

ഈ നമുക്ക് പോയിസ്റ്ററിന്റെ ഓപ്പറേഷനുകൾ വിവരിക്കുന്ന ഒരു പ്രോഗ്രാം എഴുതാം. പ്രോഗ്രാം 1.3 ഒരു കൂട്ടം കൂട്ടികളുടെ ശരാഗരി ഉയരം കാണാനുള്ളതാണ്.

### Program 1.3: കൂട്ടികളുടെ ശരാഗരി ഉയരം കാണാൻ



```
#include <iostream>
using namespace std;
int main()
{
    int *ht_ptr, n, s=0;
    float avg_ht;
    ht_ptr = new int;           // ഡെയറാമിക് മെമ്മറി അലോക്കേഷൻ
    cout<<"Enter the number of students: ";
    cin>>n;
    for (int i=0; i<n; i++)
    {
        cout<<"Enter the height of student "<<i+1<< " - ";
        cin>>* (ht_ptr+i); //to get the address of the next location
        s = s + * (ht_ptr+i);
    }
    avg_ht = (float)s/n;
    cout<<"Average height of students in the class = "<<avg_ht;
    return 0;
}
```

പ്രോഗ്രാം 1.3 ലെ ഡെയറാമിക് അലോക്കേഷൻിലൂടെ ഒരു ഇൻഡിജർ മെമ്മറിസ്ഥാനം നിർമ്മിച്ച് അതിന്റെ വിലാസം ht\_ptr എന്ന പോയിന്റർ വേരിയസിളിൽ ശേഖരിക്കുന്നു. ലൂപ്പിന്റെ ബോധി ആദ്യത്തെ പ്രാവശ്യം പ്രവർത്തിക്കുമ്പോൾ 0 ആണ് വിലാസ തേരാട്ട് കൂടുന്നത്. ഈത് പ്രത്യേകിച്ച് മാറ്റമെന്നും വരുത്തുന്നില്ല. ഇൻപുട്ട് ഡാറ്റ മെമ്മറി സ്ഥാനത്ത് സൂക്ഷിക്കുന്നു. ലൂപ്പ് ബോധിയുടെ അടുത്ത പ്രവർത്തനത്തിൽ വിലാസ തേരാട്ട് 1 കൂടുന്നു. അങ്ങനെ അടുത്ത ഇൻഡിജർ സ്ഥാനത്തിൽ രണ്ടാമത്തെ ഡാറ്റ ശേഖരിക്കപ്പെടുന്നു. ഈ പ്രവർത്തി റ തവണ ആവർത്തിക്കപ്പെടുന്നു. അങ്ങനെ റ തവണ കൂട്ടിക്കളുടെ ഡാറ്റയും തുടർച്ചയായ ഇൻഡിജർ സ്ഥാനങ്ങളിൽ ശേഖരിക്കപ്പെടുന്നു. ഓരോ ഡാറ്റയും നൽകുന്ന സമയത്തു തന്നെ അതിന്റെ തുകയും കണക്കാക്കപ്പെടുന്നു. അവ സാനും ശരാശരി ഉയരം കണക്കാക്കി പ്രവർശിപ്പിക്കുന്നു. കൃത്യമായ ശരാശരി ഉയരം ലഭിക്കുവാൻ തുകരെ ബാഹ്യതരം മാറ്റലിലൂടെ (Explicit type conversion) ഫ്ലോട്ടാക്കി മാറ്റിയിരിക്കുന്നു. പ്രോഗ്രാമിന്റെ ഒരുപുട്ട് ചുവടെ ചേർക്കുന്നു.

```
Enter the number of students: 5
Enter the height of student 1 - 170
Enter the height of student 2 - 169
Enter the height of student 3 - 175
Enter the height of student 4 - 165
Enter the height of student 5 - 177
Average height of students in the class = 171.199997
```

പ്രോഗ്രാം 1.3 മറ്റാരു കാര്യം കൂടി വിവരിക്കുന്നു. ഒരേതരം ഡാറ്റയുടെ കൂടുതൽ അനുയാസമായി പോയിന്ററിലൂടെ കൈകാര്യം ചെയ്യാം. നമ്മക്കറിയാം 11-ാം ക്ലാസിൽ ഇതെ

ആവശ്യത്തിനു വേണ്ടി നമ്മൾ അരെ ഉപയോഗിച്ചു. പക്ഷെ അരെ പ്രവൃംപനത്തോടൊപ്പം തന്നെ അരെയുടെ വലുപ്പവും നൽകിയിരിക്കണം. ഈ മെമ്മറി അനാവശ്യമായി നഷ്ടപ്പെടുത്തുകയും ആവശ്യത്തിന് മെമ്മറി ലഭിക്കാതിരിക്കാൻ കാരണമാവുകയും ചെയ്യുന്നു. എന്നാൽ പോയിന്ത്ര വേരിയബിളുപയോഗിച്ച് ദൈനംദിന അലോക്കേഷൻ നടത്തുമ്പോൾ അരെയുടെ വലുപ്പം കൂടുതു കുട്ടികളുടെ എള്ളൂത്തിനുസരിച്ചായിരിക്കും. ഈ മെമ്മറി പാശാക്കപ്പെടുന്നില്ല.

പക്ഷെ ഈ തരത്തിലുള്ള മെമ്മറി ഉപയോഗത്തിന് ചില പ്രശ്നങ്ങളുണ്ട്. പ്രോഗ്രാം 1.3 എപ്പോഴും റ എന്ന് എല്ലാ വിലകർക്കും പ്രവർത്തിക്കണം എന്നില്ല. GCC, avg\_ht എന്ന വേരിയബിളിൽ ഒരു ഒരുപ്പുട്ടും ചിലപ്പോൾ നൽകില്ല. ചില അപ്രതീക്ഷിത വിലകൾ പ്രദർശിപ്പിക്കാനും സാധ്യതയുണ്ട്. ഈ കാരണം ht\_dtr റ പ്രാരംഭവിലയായി നൽകുന്നത് ഏതെങ്കിലും ഒരു മെമ്മറി സ്ഥാനത്തിൽ വിലാസമാണ്. നമ്മൾ ഉപയോഗിക്കുന്ന പോയിന്ത്രിൽ അരിതമാറ്റിക്ക് ഓപ്പറേഷനും നേരത്തെ പറഞ്ഞപോലെ അനുവാദമില്ലാത്ത സ്ഥാനങ്ങൾ ഉപയോഗിക്കുന്നതിന് കാരണമായേക്കാം. ഈ കാരണം പ്രോഗ്രാമിൽ അപ്രതീക്ഷിതമായ അവസാനിപ്പിക്കലിന് കാരണമാവും. പലപ്പോഴും മെമ്മറിയിൽ സുക്ഷിച്ച ഡാറ്റയും നഷ്ടപ്പെട്ടു പോകും, അങ്ങനെ നമുക്ക് കൂടുതുമായ ഒരുപ്പുട്ടും ലഭിക്കുകയില്ല. ഈ പ്രശ്നങ്ങൾ പരിഹരിക്കാൻ നമുക്ക് ഭാഗം 1.5 ലെ ദൈനംദിന അരെ കൾ ഉപയോഗപ്പെടുത്താം.

#### 1.4.2 പോയിന്ത്രിലെ റിലേഷണൽ ഓപ്പറേഷനുകൾ

നാം പരിച്ച ആർ റിലേഷണൽ ഓപ്പറേറ്ററുകളിൽ == (Equality-തുല്യം), != (Non equality-തുല്യമല്ല) എന്നീ ഓപ്പറേറ്ററുകൾ മാത്രമേ പോയിന്ത്ര വേരിയബിളുകളിൽ ഉപയോഗിക്കാറുള്ളു. മെമ്മറി വിലാസം എന്നാൽ ഒരു മെമ്മറി സ്ഥാനം തിരിച്ചറിയാനുള്ള സവിശേഷ സംവ്യൂദ്ധം. അതിനാൽ ഈ മുകളിൽ മറ്റ് റിലേഷണൽ ഓപ്പറേറ്ററുകൾ ഉപയോഗിക്കുന്നതിൽ അർഹമില്ല. റ, ണ, ണ എന്നിവ രണ്ട് പോയിന്ത്രുകൾ ആണെങ്കിൽ അവയിൽ തുല്യമായതോ/വ്യത്യസ്തമായതോ ആയി മെമ്മറി സ്ഥാനങ്ങളുടെ വിലാസങ്ങളാണ് ഉണ്ടാവുക. ഈ പരിശോധിക്കാൻ റ==ണ അല്ലെങ്കിൽ റ != ണ എന്നീ പദ്ധതിയാണ് ഉപയോഗിക്കാം.

#### 1.5 അരെയും പോയിന്ത്രിനും

##### നിജങ്ങളുടെ പുരോഗതി അറിയാം

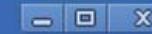


1. C++ ലെ ദൈനംദിന അലോക്കേഷൻ ഓപ്പറേറ്റർ ആണ് \_\_\_\_\_.
2. ചുവവുടെ കൊടുത്ത പ്രസ്താവന പ്രവർത്തിക്കുമ്പോൾ എന്ത് സംഭവിക്കുന്നു?
 

```
int *b = new int(10);
```
3. അർഹന്തസ് മെമ്മറി സ്ഥോക്ക് എന്നാലെന്ത്?
 

```
b = 10;
```
4. റ ഒരു ഇൻഡിക്സ് പോയിന്ത്രോഗണനിതിക്കു. ചുവവുടെ കൊടുത്തവയിൽ എത്രൊക്കെയോണ് അസാധ്യവായത്?
 

a. cout << &b;	b. p=p*5;	c. p>0
d. p++;	e. p=1500;	f. cout << *p * 2;



ഒരു പേരിൽ ഒരേ തരത്തിലുള്ള ഓൺലൈൻ ഡാറ്റകളെ ശേഖരിക്കാൻ അനേക്കൾ കഴിയുമെന്ന് നാം പറിച്ച് കഴിഞ്ഞാലോ. തുടർച്ചയായ മെമ്മറി സ്ഥാനങ്ങളിൽ ഡാറ്റ ശേഖരിക്കുന്നതിനു അനേ ഉപയോഗിക്കാം. ചിത്രം 1.9 കാണിക്കുന്നത് ar [10] എന്ന int ടെപ്പ് അനേയുടെ ചിത്രമാണ്. ഇതിൽ 10 സംഖ്യകളുണ്ട്.

ഈ അനേ 1000 എന്ന മെമ്മറി ലോക്കേഷൻിൽ തുടങ്ങുന്ന താഴീ സ്കാർപ്പിക്കുക. ഇതിനകത്തെ ഓരോ സ്ഥാനവും 4 ബെബ്രൂ വീതം ആണ് (GCC അനുസരിച്ച്). ഈ അനേയിലെ ഏത് അംഗത്തെയും അനേയുടെ പേരും സബ്സ്ക്രൈപ്റ്റും ഉപയോഗിച്ച് സൂചിപ്പിക്കാം. ഉദാഹരണം ar [0] എന്നത് 34 എന്ന സംഖ്യ നൽകുന്നു. ar [1] 12 ഉം ar [9] 19 ഉം തിരിച്ചുതരുന്നു.

ptr എന്നത് ഒരു ഇൻഡിക്കേറ്റർ പോയിന്റർ ആണെങ്കിൽ ar എന്ന

ar	
1000	34
1004	12
1008	8
1012	18
1016	24
1020	38
1024	43
1028	14
1032	7
1036	19

ചിത്രം 1.9: ar എന്ന അനേ

യുടെ മെമ്മറി നീക്കിവര്ക്കൽ



### മുക്ക് ചെയ്യാം

ar ഈ അനേയിലെ 10 സംഖ്യകളും പ്രദർശിപ്പിക്കുന്നതിനുള്ള C++ പ്രോഗ്രാം എഴുതുക.

എങ്ങനെയാണ് ഈ അനേയുടെ ഓൺലൈൻ സ്ഥാനത്തിന്റെ വിലാസം ഒരു പോയിന്ററിൽ ശേഖരിക്കുന്നത്?

അനേയുടെ ആദ്യത്തെ സ്ഥാനത്തിന്റെ വിലാസം താഴെ കൊടുക്കുന്ന രീതിയിൽ പോയിന്റ് റിൽ ശേഖരിക്കാം.

```
ptr = &ar[0];
```

```
ഇനി മുക്ക് ചുവടെ ചേർത്ത വാചകങ്ങളുടെ ഒരു പരിശോധനാം.
cout<<ptr; //1000 പ്രദർശിപ്പിക്കുന്നു, ar[0] യുടെ വിലാസം
cout<<*ptr; //34 പ്രദർശിപ്പിക്കുന്നു, ar[0] യുടെ വില
cout<<(ptr+1); //1004 പ്രദർശിപ്പിക്കുന്നു, ar[1] എം്റെ വിലാസം
cout<<*(ptr+1); //12 പ്രദർശിപ്പിക്കുന്നു, ar[1] എം്റെ വില
cout<<(ptr+9); //1036 പ്രദർശിപ്പിക്കുന്നു, ar[9] എം്റെ വിലാസം
cout<<*(ptr+9); //19 പ്രദർശിപ്പിക്കുന്നു, ar[9] എം്റെ വില

cout<<ar; എന്ന പ്രസ്താവനയുടെ ഒരു പുറ്റ് നീങ്ങൾക്ക് കണ്ണടത്താൻ കഴിയുമോ?
ഒരു പുറ്റ് 1000 ആണ്. ഈ അനേയുടെ ആദ്യത്തെ സ്ഥാനത്തിന്റെ വിലാസമാണ്. ഈ വിലാസത്തെ അനേയുടെ ഭേദം അഡ്യസ്റ്റ് എന്ന് പറയുന്നു. വിലാസം ശേഖരിക്കുന്ന വേദിയബിൾ ഒരു പോയിന്ററാണ് എന്ന് നമ്മകൾക്കിയാം. അതുകൊണ്ട് അനേയുടെ പേരായ ar ഉം ഒരു പോയിന്ററാണ്. ആയതിനാൽ താഴെ കൊടുത്ത വാചകങ്ങൾ ശരിയാണ്.
cout<<ar; //1000 പ്രദർശിപ്പിക്കുന്നു, ar[0] യുടെ വിലാസം
ptr=ar; //ptr=&ar[0] എന്നതിന് തുല്യം;
cout<<*ar; //34 പ്രദർശിപ്പിക്കുന്നു, cout<<ar[0]; എന്നതിന് തുല്യം
```

## 1. സ്റ്റ്രക്കചറു പോയിന്ററു

```
cout<<(ar+1); //1004 പ്രദർശിപ്പിക്കുന്നു, ar[1]; എൻ വിലാസം  
cout<<*(ar+1); //34 പ്രദർശിപ്പിക്കുന്നു, cout<<ar[1]; എന്തിന് തുല്യം  
ചുവടെ കൊടുത്ത C++ വാചകങ്ങൾ ഈ അരയുടെ ഏലും അംഗങ്ങളേയും പ്രദർശി  
പ്പിക്കുന്നു.
```

```
for (int i=0; i<10; i++)  
    cout<<*(ar+i)<<'\t';
```

സാധാരണ പോയിന്ററു അരയുടെ പേരും തമിൽ ചില വ്യത്യാസങ്ങളുണ്ട്. ptr++; എന്ന വാചകം ശരിയാണ് ഈത് ptr=ptr+1; എന്തിന് തുല്യമാണ്. ഈ വാചകം പ്രവർത്തിപ്പിച്ചു കഴിഞ്ഞാൽ ptr, ar[1] എൻ മെമ്മറി സ്ഥാനത്തേക്ക് പോയിന്ത് ചെയ്യും. അതായത് ptr തും ar[1] എൻ വിലാസം ആണ് ശേഖരിക്കപ്പെടുന്നത്. ptr++-ന് പകരമായി ar++ എന്ന് ഉപയോഗിക്കാൻ കഴിയില്ല. കാരണം അരയുടെ പേര് എപ്പോഴും അരയുടെ വേസ് അധ്യയ്യിച്ച് മാത്രം ഉൾക്കൊള്ളുന്നു. ഈത് മാറ്റാൻ സാധ്യമല്ല.

### ബൈനാർഡ് അഭേദം

C++ തും അരും ഒരു തരതിലുള്ള ഡാറ്റ കൈകാര്യം ചെയ്യാൻ ഉപയോഗിക്കുന്നു. പക്ഷേ ഡാറ്റയുടെ എല്ലാം മുൻകൂട്ടി അറിയില്ല എങ്കിൽ അരും പ്രവ്യാപിക്കാൻ പ്രയാസം നേരിടുന്നു. ഒരു ജില്ലയിലെ എല്ലാ ഹയർ സെക്കണ്ടറി സ്കൂളുകളിലെയും വിജയ ശതമാനം ശേഖരിക്കുന്നതിന് ഒരു അരും പ്രവ്യാപിക്കുന്നത് എങ്ങനെയാണ്? float pass[n]; float pass[]; എന്നീ രണ്ട് പ്രവ്യാപനങ്ങളും C++ തും തെറ്റാണ്. അരയുടെ വലുപ്പം ഒരു ഇൻ്റിജർ കോൺസ്ലീറ്റായിരിക്കണം. അങ്ങനെ പ്രവ്യാപിക്കുന്നോൾ മെമ്മറി പാഴാക്കപ്പെടുവാനോ തികയാതെ വരുവാനോ കാരണമാകാം. ഡിസ്ട്രിക്ടിലെ സ്കൂളുകളുടെ എല്ലാം പ്രോഗ്രാം തയാറാക്കുന്നോൾ അറിയില്ല എങ്കിൽ, ഇൻപുട്ടിനു സംബന്ധിച്ച് മെമ്മറി അലോക്കേറ്റ് ചെയ്യുന്ന ഒരു അരും ഉപയോഗിച്ചാൽ മതിയാകും. ഇവിടെയാണ് ബൈനാർഡ് അരയുടെ ആവശ്യം ഉണ്ടാകുന്നത്. ബൈനാർഡ് അരും പ്രോഗ്രാം പ്രവർത്തിക്കുന്നോണ് നിർമ്മിക്കപ്പെടുന്നത്. ഇതിനുവേണ്ടി ബൈനാർഡ് മെമ്മറി അലോക്കേഷൻ ഓഫ്രോറായ new ഉപയോഗിക്കുന്നു. ഈതിന്റെ വാക്കും ചുവടെ ചേർക്കുന്നു.

```
pointer = new data_type[size];
```

ഇവിടെ size ഒരു സ്ഥിരവില (constant), വേരിയബിൾ, ഇൻഡിക്യർ എക്സ്പ്രഷൻ ഇവയിൽ ഏതെങ്കിലും ഒന്നായാൽ മതി.

പ്രോഗ്രാം 1.4, ബൈനാർഡ് അരയുടെ ആശയം വിവരിക്കുന്നു. ഈ പ്രോഗ്രാമിൽ വിജയശതമാനം ശേഖരിക്കുന്നത് ഉപയോകതാവ് രണ്ട് കെമിൽ നൽകുന്ന സ്കൂളിന്റെ എല്ലാംഗുസതിച്ചാണ്.

### Program 1.4: ഏറ്റവും കുറിയ വിജയ ശതമാനം കാണാൻ

```
#include <iostream>  
using namespace std;  
int main()  
{  
    float *pass, max;
```



```
intput n; - 12
cout<<"Enter the number of schools: ";
cin>>n; // സ്കൂളിന്റെ എണ്ണം ഇൻപുട്ട് ചെയ്യാൻ
pass = new float[n]; // n അംഗങ്ങളുള്ള ഡൈറക്ടറി അംഗങ്ങൾ ഫോറ്മാറ്റിൽ പ്രവർത്തിച്ചു
for (i=0; i<n; i++)
{
    cout<<"Percent of pass by school "<<i+1<<": ";
    cin>>pass[i]; // സബ്സക്രിപ്റ്റ് വേരിയബിലിന്റെ ഉപയോഗം
}
max=pass[0];
for (i=1; i<n; i++)
    if (pass[i]>max) max = *(pass+i);
/* അംഗങ്ങളെ സബ്സക്രിപ്റ്റ് ഉപയോഗിച്ചും പോയിന്ത്തിരുന്ന ഓപ്പറേഷൻിലും ഉപയോഗിക്കുന്നു. */
cout<<"Highest percent is "<<max;
return 0;
```

### Output:

```
Enter the number of schools: 5
Percent of pass by school 1: 75.6
Percent of pass by school 2: 66.5
Percent of pass by school 3: 89.3
Percent of pass by school 4: 71
Percent of pass by school 5: 70.6
Highest percent is 89.3
```

പ്രോഗ്രാം 1.4 ത്രി ഡൈറക്ടറി അംഗങ്ങൾ ഉപയോഗിച്ചാണ് ഡാറ്റ ശേഖരിക്കുന്നത്. മെമ്മറി നീകിവിയ്ക്കൽ നടക്കുന്നത് റൺ ടെസ്റ്റിംഗ് മാത്രമാണ്. 4 ബൈറ്റ് വീതമുള്ള 5 സ്ഥാനങ്ങൾ pass എന്ന അംഗത്വത്തിൽ റൺ ടെസ്റ്റിംഗ് തയാറാക്കുന്നു. ഈ അംഗത്വത്തിൽ പോയിന്ത്തിരോ സബ്സക്രിപ്റ്റോ ഉപയോഗിച്ച് എടുക്കാൻ കഴിയും.



നമ്മുക്ക് ചെയ്യാം

പ്രോഗ്രാം 1.4 തയാറാക്കുന്ന സമയത്ത് ജില്ലയിലെ സ്കൂളുകളുടെ എണ്ണം കൂടുതുമായി പ്രോഗ്രാമർക്ക് അറിയില്ല. ഇവിടെ സാധാരണ അംഗ ഉപയോഗിക്കുകയാണെങ്കിൽ ഒരു കുടിയ സംഖ്യ വലുപ്പമായി നൽകി അംഗ പ്രവ്യാഹിക്കേണ്ടതായി വരുകയും ഇത് ശേഷിക്കുന്ന മെമ്മറി പാശാക്കാൻ കാരണമാവുകയും ചെയ്യും. എന്നാൽ ഇവിടെ ഡൈറക്ടറി അംഗ ഉപയോഗിച്ചാൽ ഉപയോക്താവ് നൽകിയ കൂടുതുമായ എണ്ണത്തിൽ അംഗ നിർമ്മിക്കുന്നതിനാൽ മെമ്മറി പാശാക്കപ്പെടുന്നില്ല.

താഴെയുള്ള റൺ പ്രസ്താവനകൾ തമിലുള്ള വ്യത്യാസം എഴുതുക:

```
int *ptr = new int(10);
int *ptr = new int[10];
```

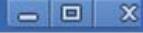
## 1.6 പോയിന്റും സ്റ്റ്രീം

സ്റ്റ്രീം കാരക്ടർ അറെ വഴി ശേവർക്കാമെന്നും അറെയുടെ പേരിനെ സ്റ്റ്രീം വേരിയബിളായി കണക്കാക്കാമെന്നും നമ്മൾ 11-ാം ക്ലാസിൽ പറിച്ചു. അറെയുടെ പേരിൽ അതിന്റെ ബേസ് അധിക്ക് അടങ്ങിയിരിക്കുന്നു എന്നും നമ്മൾ മനസ്സിലാക്കി. അതു കൊണ്ട് അറെയെ ഒരു പോയിന്റും കണക്കാക്കാം. ഈ നമുക്ക് ഈ ആശയങ്ങളെ യോജിപ്പിച്ചുകൊണ്ട് ഒരു സ്റ്റ്രീം കാരക്ടർ പോയിന്റും ഉപയോഗിക്കാമെന്നും അതിന് മറ്റ് പോയിന്റുമായി എൽക്ക് വ്യത്യാസമുണ്ടെന്നും ചർച്ച ചെയ്യാം. താഴെ കൊടുത്ത വാചകങ്ങൾ പരിശോധിക്കുക.

```
char str[20];           // ക്യാരക്ടർ അറെ പ്രവ്യാപനം
char *sp;               // ക്യാരക്ടർ പോയിന്റർ പ്രവ്യാപനം
cin>>str;              // "Program" എന്ന സ്റ്റ്രീം കാരക്ടർ ചെയ്യുന്നു എന്ന്
                        // കരുതുക
cout<<str;              // "Program" എന്ന സ്റ്റ്രീം പ്രദർശിപ്പിക്കാൻ
sp=str;                 // str എന്ന സ്റ്റ്രീം ഉള്ളടക്കം, പോയിന്റർ ട്രാഡിലേഷൻ കോപ്പി
                        // ചെയ്യുന്നു.
cout<<sp;                // "Program" എന്ന സ്റ്റ്രീം പ്രദർശിപ്പിക്കുന്നു
cout<<&str[0];            // "Program" എന്ന സ്റ്റ്രീം പ്രദർശിപ്പിക്കുന്നു
cout<<sp+1;              // "rogram" എന്ന സ്റ്റ്രീം പ്രദർശിപ്പിക്കുന്നു
cout<<&str+1;              // "rogram" എന്ന സ്റ്റ്രീം പ്രദർശിപ്പിക്കുന്നു
cout<<str[0];            // 'P' എന്ന ക്യാരക്ടർ പ്രദർശിപ്പിക്കുന്നു
cout<<*sp;                // 'P' എന്ന ക്യാരക്ടർ പ്രദർശിപ്പിക്കുന്നു
cout<<&str;                // str അറെയുടെ ബേസ് അധിക്ക് പ്രദർശിപ്പിക്കുന്നു
cout<<&sp;                // sp എന്ന പോയിന്റും വിലാസം പ്രദർശിപ്പിക്കുന്നു
```

ഒരു അറെയിൽ ശേവർച്ചിത്തിക്കുന്ന സ്റ്റ്രീം, അഭ്യന്തരിച്ചെന്നും ഓപ്പേറേറ്റ് (=) ഉപയോഗിച്ച് മറ്റാരു കാരക്ടർ അറെയിലേക്ക് കോപ്പി ചെയ്യാൻ സാധിക്കില്ല. (ഈതിനു വേണ്ടി നമ്മൾ strcpy() എന്ന ഫംശൻ ഉപയോഗിക്കുന്നു) പകുശ അഭ്യന്തരിച്ചെന്നും ഓപ്പേറേറ്റർ (=) ഉപയോഗിച്ച് സ്റ്റ്രീം കോപ്പി ചെയ്യാൻ കാരക്ടർ പോയിന്റുകൾക്ക് സാധിക്കും. മുകളിലെത്തെ ഉദാഹരണങ്ങളിൽ sp=str; cout<<sp; എന്നീ വാചകങ്ങൾ ഈ വസ്തുതയുടെ സ്ഥിരീകരിക്കുന്നു. ഈ തിന്റെ കാരക്ടർ പോയിന്റീന് ഒരു സ്റ്റ്രീം കാരക്ടർ പോയിന്റും ഒരു സ്റ്റ്രീം വേരിയബിൾ ആയി കണക്കാക്കാം.

മറ്റാരു രസകരമായ കാര്യം cout<<&str[0]; എന്ന വാചകം മുഴുവൻ സ്റ്റ്രീം നേരും പ്രദർശിപ്പിക്കുന്നു എന്നതാണ്. ഈതിന്റെ അർമ്മാ നമ്മൾ ഒരു സ്റ്റ്രീം ബേസ് അധിക്ക് പ്രദർശിപ്പിക്കാൻ ശ്രമിച്ചാൽ ആ സ്റ്റ്രീം മുഴുവനായും പ്രദർശിപ്പിക്കേണ്ടും. ഈതുപോലെ ഒരു int അറെയുടെയോ float അറെയുടെയോ ഒന്നാമത്തെ സ്ഥാനത്തിന്റെ വിലാസം പ്രദർശിപ്പിക്കാൻ ശ്രമിച്ചാൽ അതിന്റെ ബേസ് അധിക്ക് യോജിപ്പിക്കും



പ്രദർശിപ്പിക്കപ്പെടുന്നത്.

```
ഉദാഹരണത്തിന് int a[3] = {8, 9, 10};  
cout<<&a[0];
```

എന്നീ പ്രസ്താവനകൾ പ്രവർത്തിപ്പിക്കുന്നോൾ ആ എന്ന അനേയുടെ വേസ് അധ്യാള്സ് ആയിരിക്കും പ്രദർശിപ്പിക്കപ്പെടുന്നത്.

### കാരക്ടർ പോയിന്ററിഡിസ്റ്റ് നേട്ടണം

സ്റ്റിങ്ചിനെ ഫ്ലാർ ചെയ്യാൻ കാരക്ടർ പോയിന്ററിൽ ഉപയോഗിക്കുന്നോൾ ചുവടെ കൊടുത്ത നേട്ടങ്ങൾ ഉണ്ടാകുന്നു.

- വലുപ്പം പ്രവ്യാപന പ്രസ്താവനയിൽ രേഖപ്പെടുത്താത്തതുകൊണ്ട്, എത്ര വലുപ്പ് മുള്ള സ്റ്റിങ്ചും ശേഖരിക്കാം. മെമ്മറി പാഴാക്കപ്പെടുകയോ തികയാതിരിക്കുകയോ ചെയ്യുന്നില്ല. എന്നാൽ ഈ പ്രാരംഭ വില നൽകുന്നോൾത്തെന ചെയ്തിരിക്കണം. (e.g., char \*str = "Program";)
- അസൈൻമെന്റ് ഓപ്പറേറ്റ് (=) ഉപയോഗിച്ച് സ്റ്റിങ്ച് കോപ്പി ചെയ്യാം.
- സ്റ്റിങ്ചിലെ എത്ര കാരക്ടറും പോയിന്ററിൽ ഉപയോഗിച്ച് എല്ലാപ്പും പുറത്തുക്കാം.
- കാരക്ടർ അനേകണ്ണെ കുറഞ്ഞ മെമ്മറി സ്ഥലം ഉപയോഗിച്ച് കൈകാര്യം ചെയ്യാം.

### സ്റ്റിങ്ചുകളുടെ അഭി

നമുക്ക് ഒരാഴ്ചയിലെ ദിവസങ്ങളുടെ പേരുകൾ ഒരു അനേയിൽ ശേഖരിക്കണമെന്നു കരുതുക. ഒരു കാരക്ടർ അനേകൾ അല്ലെങ്കിൽ കാരക്ടർ പോയിന്ററിന് ഒരു പേര് മാത്രമേ ശേഖരിക്കാൻ കഴിയു. നമുക്ക് സുചിപ്പിക്കേണ്ടത് സ്റ്റിങ്ചുകളുടെ ഒരു കൂട്ടമാണ്. ("Sunday", "Monday", ..., "Saturday") തീർച്ചയായും ഇതിനു വേണ്ടി നമ്മൾ ഉപയോഗിക്കുന്നത് കാരക്ടർ അനേകളുടെ ഒരു അനേയാണ്; (char തരത്തിലുള്ള ഒരു 2D അറേ) അല്ലെങ്കിൽ കാരക്ടർ പോയിന്ററുകളുടെ ഒരു അനേയാണ്. കാരക്ടർ പോയിന്ററുകളുടെ ഒരു അറേ പ്രവ്യാഹിക്കുന്നതിന്റെ രീതി താഴെ കൊടുത്തിരിക്കുന്നു.

```
char *week[7];
```

ഈ അനേകൾ പരമാവധി 7 സ്റ്റിങ്ചുകളെ ഉൾകൊള്ളുന്നുള്ള കഴിവുണ്ട്. അതിൽ ഓരോ സ്റ്റിങ്ചിലും പരിധിയില്ലാതെ കാരക്ടറുകൾ ഉൾപ്പെടുത്താവുന്നതാണ്. പോയിന്റർ അനേകൾ പ്രാരംഭവിലെ നൽകുന്ന വിധം താഴെ ചേർക്കുന്നു.

```
char *week[7]={ "Sunday", "Monday", "Tuesday", "Wednesday",
    "Thursday", "Friday", "Saturday" };
```

ചിത്രം 1.10 ലെ മെമ്മറി നീക്കിവയ്ക്കലിന്റെ കൃത്യമായ വിവരങ്ങം കൊടുത്തിരിക്കുന്നു. ഷേഡ് ചെയ്തിരിക്കുന്ന ഭാഗമാണ് നീക്കിവയ്ക്കപ്പെട്ടിരിക്കുന്നത്.



### നമുക്ക് ചെയ്യാം

സ്ലാസ് 11 ത്ത് പഠിച്ചിരുന്ന സോൾട്ടീൻഗ് ടെക്നിക്ക് ഉപയോഗിച്ച് പേരുകൾ സോൾട്ട് ചെയ്യാനുള്ള ഒരു C++ പ്രോഗ്രാം എഴുതുക. കാരക്ടർ പോയിസ്റ്റുൾ ഉപയോഗിക്കുന്നതിനാൽ സ്റ്റ്രിങ്ക്കിനെ അബ്സൈൽമെന്റ് ഓപ്പറേറ്റർ (=) ഉപയോഗിച്ച് കോഡി ചെയ്യുക. കമ്പ്യൂട്ടർ ലാബിൽ വച്ച് നിങ്ങൾ എഴുതിയ പ്രോഗ്രാം ശരിയാണോ എന്ന് പരിശോധിക്കുക.

Week

S	u	n	d	a	y	\0			
M	o	n	d	a	y	\0			
T	u	e	s	d	a	y	\0		
W	e	d	n	e	s	d	a	y	\0
T	h	u	r	s	d	a	y	\0	
F	r	i	d	a	y	\0			
S	a	t	u	r	d	a	y	\0	

ചിത്രം 1.10: സ്റ്റ്രിങ്കളുടെ മെമ്മറി നീക്കി വയ്ക്കാൻ



സ്റ്റ്രിങ്കളെ കൈക്കാരും ചെയ്യുന്നതിന് പോയിസ്റ്റുൾ അഭിയ്ക്ക് പകരം സ്റ്റ്രിങ്കളുടെ അറബി ഉപയോഗിക്കാം. സ്റ്റ്രിങ്കളുടെ അറബി തയാറാക്കാൻ 2D കാരക്ടർ അറബി ഉപയോഗിക്കുന്നു. പ്രവ്യാഹരണ പ്രസ്താവന താഴെ കൊടുത്തിരിക്കുന്നു.

```
char name[10][20];
```

ഈ അഭിയ്ക്ക് 10 പേരുകൾ ഉൾക്കൊള്ളാനുള്ള കഴിവുണ്ട്. ഓരോ പേരിലും മാക്സിമം 10 ക്യാരക്ടർ വരെ അനുവദിച്ചിരിക്കുന്നു. ഒരു ബെഡ് നൽ ക്യാരക്ടറിനു (' \'0 ') വേണ്ടി റിസർവ് ചെയ്യപ്പെട്ടിരിക്കുന്നു. ഓരോ സ്റ്റ്രിങ്കും name[i] എന്ന എക്സ്പ്രഷൻലൈറ്റ് റഹർ സൂചിപ്പിക്കുന്നു. ഇതിൽ സബ്സ്ക്രീപ്റ്റ് i യ്ക്ക് 0 മുതൽ 9 വരെയുള്ള വിലകൾ സീക്രിക്കാം. **ഇവിടെ strcpy()** ഫംഗ്ഷൻ ഉപയോഗിച്ച് മാത്രമേ സ്റ്റ്രിങ്കൾ കോഡി ചെയ്യാൻ കഴിയു.



ചിത്രം 1.10 ത്ത് കാണിച്ചിരിക്കുന്ന week എന്ന അഭൈയ ചിത്രം തരത്തിലുള്ള ഒരു 2D അറബി ഉപയോഗിച്ച് പ്രവ്യാഹരിക്കുന്ന രീതി ചുവടെ കൊടുത്തിരിക്കുന്നു.

```
char week[7][11]={ "Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"};
```

എന്താണ് ഈ അഭിയും മുമ്പ് സൂചിപ്പിച്ച ക്യാരക്ടർ പോയിസ്റ്റുൾ അഭിയും ഉപയോഗിക്കുന്നതിലുള്ള വ്യത്യാസം? ക്യാരക്ടർ പോയിസ്റ്റുൾ അറബി ഉപയോഗിക്കുന്നുണ്ട് അഭിയുടെ നീക്കി വയ്ക്കാതെ ഭാഗങ്ങൾ (ചിത്രം 1.10 ത്ത് ഷേഡ് ചെയ്യാതെ ഭാഗങ്ങൾ) മറ്റ് മെമ്മറി നീക്കി വയ്ക്കലിനുവേണ്ടി ഉപയോഗപ്പെടുത്താം. എന്നാൽ 2D ക്യാരക്ടർ അറബി ഉപയോഗിക്കുന്നുണ്ട് ഇന്ന് ഭാഗങ്ങൾ നീക്കിവയ്ക്കാൻ കഴിയാതെ പാശാക്കപ്പെടുന്നു.

താഴെ കൊടുത്ത വാചകം സ്റ്റ്രിങ്കളെ ഉപയോഗിക്കുന്ന രീതി വിവരിക്കുന്നു.

```
for (i=0; i<7; i++)
    cout<<week[i];
```



### നിങ്ങളുടെ പുരോഗതി അറിയാം



1. ദൈനംദിന അരു എന്നാൽ എന്ത്?
2. ഒരു അറൈലെ ഒന്നാമത്തെ ലോക്കേഷൻ വിലാസത്തെ \_\_\_\_\_ എന്ന് പറയുന്നു.
3. arr ഒരു ഇൻഡിക്സ് അറൈയാണെങ്കിൽ താഴെ കൊടുത്തവയിൽ തെറ്റേത്?
  - a. cout<<arr;
  - b. arr++;
  - c. cout<<\* (arr+1);
  - d. cin>>arr;
  - e. arr=1500;
  - f. cout<<\*arr \* 2;
4. 10 പുസ്തകങ്ങളുടെ പേരുകൾ പോയിന്റീർ ഉപയോഗിച്ച് സൂചിപ്പിക്കാം നൂളുള്ള C++ പ്രവ്യാപന പ്രസ്താവന എഴുതുക.
5. ഒരു പോയിന്റീർ പ്രവ്യാഹിച്ച് അതിൽ നിങ്ങളുടെ പേര് പ്രാരംഭ വിലയായി ചേർക്കുക.

### 1.7 പോയിന്റീറും സ്ട്രക്ചറും

ഈ അധ്യായത്തിന്റെ തുടക്കത്തിൽ നമ്മൾ സ്ട്രക്ചർ ഡാറ്റാതരത്തെക്കുറിച്ചും അതിന്റെ ഉപയോഗത്തെക്കുറിച്ചും ചർച്ച ചെയ്തു. ഈ ഭാഗത്ത് നമ്മൾ സ്ട്രക്ചർ വേറിയബിളിനെ പോയിന്റീറെ സഹായത്താൽ എങ്ങനെ ഉപയോഗിക്കാമെന്ന് പറിക്കുന്നു. ഒരു employee സ്ട്രക്ചർ ചുവടെ ചേർക്കുന്നു.

```
struct employee
{
    int ecode;
    char ename[15];
    float salary;
};
```

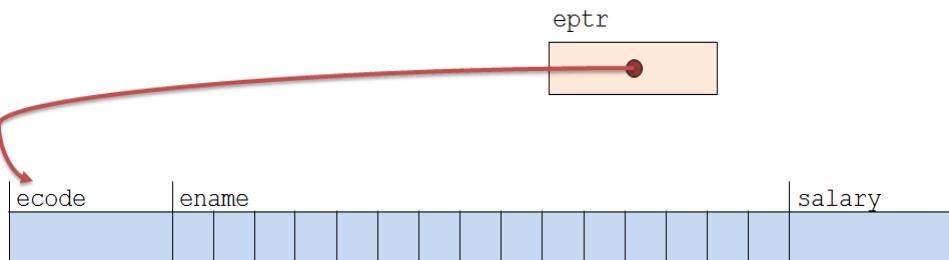
ഈ ചുവടെ കൊടുത്തിരിക്കുന്ന പ്രവ്യാപന പ്രസ്താവന പതിശോധിക്കാം.

```
employee *eptr;
```

ഈവിടെ eptr എന്ന പോയിന്റീറിന് ഒരു employee സ്ട്രക്ചറിന്റെ അധീക്ഷിതക്കൊള്ളാൻ കഴിയും. ഈ താഴെ കൊടുത്ത പ്രസ്താവന ശ്രദ്ധിക്കുക.

```
eptr = new employee;
```

ഈ വാചകം 23 ബൈറ്റ് മെമ്മറി നീക്കിവയ്ക്കുകയും ചെയ്യുകയും അതിന്റെ ബേസ് അധീക്ഷിത eptr എന്ന പോയിന്റീറിൽ ശേഖരിക്കുകയും ചെയ്യും. ചിത്രം 1.11 തോം ഇക്കാര്യം വിവരിക്കുന്നു.



ചിത്രം 1.11: employee ടെപ്പ് ശാറ്റുടെ ഡൈഗ്രാഫിക് അലോക്കേഷൻ

സ്ട്രക്ചർ അംഗങ്ങളെ താഴെ കൊടുത്ത വിധമാണ് ഉപയോഗിക്കുന്നത് എന്ന് നമുക്ക് റിയാം.

structure\_variable.element\_name

ഈ അംഗങ്ങളെ ഉപയോഗിക്കാൻ നമുക്ക് ഇവിടെ സ്ട്രക്ചർ വേരിയബിൾ ഇല്ല. eptr എന്ന പോയിസ്റ്റിലുണ്ടെന്ന് ഇവയെ ഉപയോഗിക്കേണ്ടത്. ഇതിന്റെ വാക്യാലടക്ക ചുവടെ ചേർക്കുന്നു.

structure\_pointer->element\_name

ശ്രദ്ധിക്കുക, സ്ട്രക്ചർ പോയിസ്റ്റും അംഗങ്ങളും ‘->’ ഓപ്പറേറ്റിലുടെ ‘->’ ബന്ധിപ്പിക്കപ്പെട്ടിരിക്കുന്നു. ഈ ഓപ്പറേറ്റർ ‘-’ നു പുറത്തെ ‘>’ ചിഹ്നം കൂടി ഉപയോഗിച്ചാണ് ഇഴുതുന്നത്. ചുവടെ കൊടുത്ത പ്രസ്താവനകൾ ചിത്രം 1.11 ലെ സ്ട്രക്ചർ അംഗങ്ങളെ ആക്സസ് ചെയ്യാൻ ഉപയോഗിക്കാം.

```
eptr->ecode = 657346; //employee code ന് വില നൽകുന്നു
gets(eptr->ename); //employee യുടെ പേര് ഇൻപുട്ട് ചെയ്യുന്നു
cin>> eptr->salary; //employee യും സാലാറി ഇൻപുട്ട് ചെയ്യുന്നു
cout<< eptr->salary * 0.12; //സാലാറിയുടെ 12% പ്രദർശിപ്പിക്കുന്നു
```



**നമുക്ക് ചെയ്യാം**

1.3.1 എന്ന ഭാഗത്തിൽ നമ്മൾ cx\_ptr എന്ന പോയിസ്റ്റിൽ വേരിയബിൾ complex ടെപ്പ് സ്ട്രക്ചർ പോയിസ്റ്റിൽ ഉപയോഗിച്ചു. ഒരു complex നമ്പർ ഇൻപുട്ട് ചെയ്ത് അതിനെ തമാർമ്മ രൂപത്തിൽ പ്രദർശിപ്പിക്കാനുള്ള C++ പ്രസ്താവനകൾ സ്ട്രക്ചർ പോയിസ്റ്റിൽ ഉപയോഗിച്ച് എഴുതുക.

നമുക്ക് ഒരു പുതിയ അംഗത്തെ കൂടിച്ചേർത്തുകൊണ്ട് employee സ്ട്രക്ചർ പരിഷ്കരിക്കാം.

```
struct employee
{
    int ecode;
    char ename[15];
    float salary;
    int *ip;
};
```

തീർച്ചയായും ഇ എന അംഗം ഒരു ഇൻഡിജർ പോയിൻറ്ററാണ്. ഇതിന് ഒരു ഇൻഡിജർ സ്ഥാനത്തിന്റെ വിലാസം സുകഷിക്കാം. ചുവരെ കൊടുത്ത പ്രസ്താവനകൾ പോയിൻറ്റർ റിപ്പോർട്ടുകളും വിവരിക്കുന്നു.

```
eptr->ip = new int(5); /* ഡെക്കാമിക് അലോക്കേഷൻ നടത്തി ഇ യെ
രു ഇൻഡിജർ ലോക്കേഷൻമായി ബന്ധപ്പെട്ടിട്ടും അതിൽ 5 ശേവർക്കുന്നു * .
cout << *(eptr->ip); // 5 എന വില പ്രദർശിപ്പിക്കുന്നു
int n = eptr->*ip+1; // 5 നോട് 1 കൂട്ടി ന തു ശേവർക്കുന്നു
നോക്കുക ip പോയിൻ്റ് ചെയ്യുന്ന വില രണ്ട് രീതിയിൽ പുറത്തെടുക്കാം.
*(eptr->ip) അല്ലെങ്കിൽ eptr->*ip. ഒരു സ്ട്രക്ചറിന് ഏത് തരത്തിലുള്ള പോയിൻ്റ്
റിനേയും അംഗങ്ങളും കാം. വേണമെങ്കിൽ ഒരു സ്ട്രക്ചറിന്റെ തരത്തിൽ തന്നെയുള്ള
പോയിൻ്ററിനെ അതിന് അംഗമാക്കാം.
```

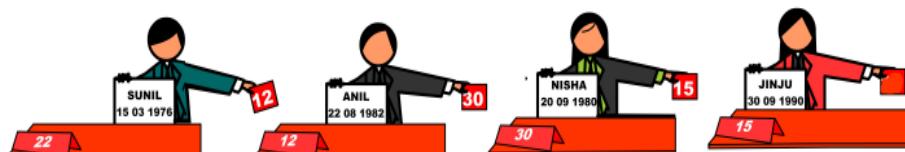
```
struct employee
{
    int ecode;
    char ename[15];
    float salary;
    employee *ep;
};

ഈ എന അംഗം employee
തരത്തിലുള്ള പോയിൻ്റർ ആണ്.
```

ഇപ്പോൾ സ്ട്രക്ചർ employee യെ സയം സുചിത സ്ട്രക്ചർ (Self referential structure) എന് വിളിക്കുന്നു. ഇതിനെക്കുറിച്ച് നമുക്ക് വിശദമായി ചർച്ച ചെയ്യാം.

### സ്വയം സുചിത സ്ട്രക്ചർ (Self referential structure)

സയം സുചിത സ്ട്രക്ചർ ഒരു പ്രത്യേക തരം സ്ട്രക്ചറാണ്. ഇതിൽ ഇതേ സ്ട്രക്ചർ തരത്തിലുള്ള ഒരു പോയിൻ്റർ വേരിയബിൾ ഉണ്ടായിരിക്കും. ഈ പോയിൻ്റിന് ഇതേ പോലുള്ള മറ്റാരു സ്ട്രക്ചറിലേക്ക് പോയിൻ്റ് ചെയ്യാം. ഇങ്ങനെ നിർമ്മിക്കുന്ന കണ്ണി എത്ര വേണമെങ്കിലും ദീർഘിപ്പിക്കാം. ചിത്രം 1.12 ഈ ആശയം വിവരിക്കുന്നു.



ചിത്രം 1.12: ഒരു employee സ്ട്രക്ചർ മറ്റാനെന്ന പോയിൻ്റ് ചെയ്യുന്നു

"Sunil" എന എംപ്ലോയി "Anil" എന എംപ്ലോയിയെ പോയിൻ്റ് ചെയ്യുന്നു. അയാളുടെ ഫേബിൾ നമ്പർ 12 ആണ്. "Anil" പോയിൻ്റ് ചെയ്യുന്നത് "Nisha" എന എംപ്ലോയിയെയൊണ്ടിന് ഇത് തുടരുന്നു.

C, C++ ഭാഷകളിൽ ലിക്ക്യലിന്റ്, ട്രി തുടങ്ങിയ ഡയനാമിക് ഡാറ്റാ സ്ട്രക്ചറുകളുടെ നിർമ്മാണത്തിന് സഹായകമായ ഒരു പ്രധാന ടൂൾ ആണ് സയം സുചിത സ്ട്രക്ചർ. പ്രോഗ്രാമിന്റെ പ്രവർത്തന സമയത്ത് നീക്കി വയ്ക്കുന്ന മെമ്മറി സ്ഥാനങ്ങളിലെ ഡാറ്റയുടെ കൂടുതൽ ഡെക്കാമിക് ഡാറ്റാ സ്ട്രക്ചർ എന്നു വിളിക്കുന്നു. ലിക്ക്യലിന്റ് ഡാറ്റാ സ്ട്രക്ചറിനെക്കുറിച്ച് കൂടുതലായി അധ്യായം 3 തു ചർച്ച ചെയ്യുന്നു.



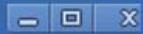
## നമ്മകൾ സംഗ്രഹിക്കാം

C++ ലെ നുതനമായ ധാരാത്തങ്ങളെ കുറിച്ച് നാം ചർച്ച ചെയ്തു. ശ്രദ്ധ ധാരാത്തെ ഒരു പേരിൽ പ്രതിനിധാനം ചെയ്യാൻ സ്റ്റെക്ക്‌ചർ ധാരാത്തം ഉപയോഗിച്ചു. സ്റ്റെക്ക്‌ചർ അംഗങ്ങളെ ഡോട്ട് ഓപ്പറേറ്ററി (.) ലുടെ ഉപയോഗിക്കുന്നത് നാം ചർച്ച ചെയ്തു. പോയിന്റ്‌റൂം പ്രത്യേക ധാരാ തരം ആബന്നന് മനസ്സിലാക്കി. പോയിന്റ്‌റൂംമായി ബന്ധപ്പെട്ട ഓപ്പറേഷനുകൾ പദ്ധതിയിൽ സഹായത്തോടെ വിവരിച്ചു. ദൈനന്ദിനിക് മെമ്മറി അലോക്കേഷൻ എന്ന ആശയവും, ആവശ്യമായ ഓപ്പറേറ്ററുകളും നേട്ടങ്ങളും ചർച്ച ചെയ്തു. അനേയും പോയിന്റ്‌റൂം തമ്മിലുള്ള ബന്ധം വിവരിച്ചു. സ്റ്റെക്ക് ധാര പോയിന്റ്‌റൂം ഉപയോഗിക്കുന്നതും പോയിന്റ്‌റൂം സ്റ്റെക്ക്‌ചറ്റിനേയും ബന്ധപ്പെട്ടിരിക്കുന്ന വിധവും മനസ്സിലാക്കി. ഈ അധ്യാത്മത്തിൽ പതിച്ച പല കാര്യങ്ങളും അധ്യായം - 3 പഠിക്കുന്നതിനുള്ള അടിസ്ഥാനമായി ഉപയോഗിക്കാം.



## നമ്മകൾ പരിശീലിക്കാം

1. ടെലിഫോൺ വർക്കാരെൽ വിവരങ്ങൾ അടങ്കുന്ന ഒരു സ്റ്റെക്ക്‌ചർ നിർവ്വചിക്കുക. അതിൽ പേര്, ടെലിഫോൺ നമ്പർ എന്നീ അംഗങ്ങൾ ഉണ്ടായിരിക്കണം. ഈതുപയോഗിച്ച് പേര് നൽകിയാൽ നമ്പർ തരുന്നതും, അല്ലെങ്കിൽ നമ്പർ നൽകിയാൽ പേര് തരുന്നതുമായ ഒരു മെമ്പു നിയന്ത്രിതമായ C++ പ്രോഗ്രാം തയാറാക്കുക.
2. ഒരു ബാക്കിലെ ഉപഭോക്താവിൻ്റെ വിവരങ്ങൾ അടങ്കുന്ന സ്റ്റെക്ക്‌ചർ നിർവ്വചിക്കുക. അക്കൗണ്ട് നമ്പർ, പേര്, അക്കൗണ്ട് ഓപ്പൺിംഗ് തീയതി, ബാലൻസ് എന്നീ വിവരങ്ങൾ ഉണ്ടായിരിക്കണം. ഈതുപയോഗിച്ച് ഡെപ്പോസിറ്റ്, വിഡ്യോ, വ്യൂ എന്നീ പ്രവൃത്തികൾ നടത്താനാവശ്യമായ ഒരു മെമ്പു നിയന്ത്രിത പ്രോഗ്രാം തയാറാക്കുക. ഡെപ്പോസിറ്റും വിഡ്യോവലും നടക്കുമ്പോൾ ബാലൻസ് സംഖ്യയിൽ മാറ്റം വരണ്ണം. അക്കൗണ്ടിൽ മിനിമം ബാലൻസായി Rs. 1000/- ഉണ്ടായിരിക്കണം (അക്കൗണ്ട് ഓപ്പൺിംഗ് തീയതി നെറ്റുഡ്യൂ സ്റ്റെക്ക്‌ചറ്റായി ഉൾപ്പെടുത്താം).
3. വിദ്യാർത്ഥികൾക്ക് കമ്പ്യൂട്ടർ സയൻസിൽ കിട്ടിയ TE സ്കോർ അവരോധണ ക്രമത്തിൽ ക്രമീകരിക്കുന്നതിനു പോയിന്റർ ഉപയോഗിച്ച് ഒരു C++ പ്രോഗ്രാം എഴുതുക.
4. ക്യാരക്കർ പോയിന്റർ ഉപയോഗിച്ച് ഇൻപുട്ട് ചെയ്ത ഒരു സ്റ്റെക്ക് പാലിൻഡ്രോം ആബന്ന അല്ലെങ്കിലും എന്ന് പരിശോധിക്കാനുള്ള ഒരു C++ പ്രോഗ്രാം തയാറാക്കുക.
5. പോയിന്റർ ഉപയോഗിച്ച്, നിങ്ങളുടെ കൂട്ടാളിലെ എല്ലാ കൂട്ടികളുടെയും പേരുകൾ ഇൻപുട്ട് ചെയ്യുക. ഈതിൽ നിന്നും ഒരു റോൾ ലിസ്റ്റ് നിർമ്മിക്കുക. ഈ ലിസ്റ്റിൽ പേരുകൾ അക്ഷരമാല ക്രമത്തിലും റോൾ നമ്പർ 1, 2, 3 ... എന്ന ക്രമത്തിലും വരണ്ണം.
6. രജിസ്ട്രേഷൻ നമ്പർ, പേര്, ആർ വിഷയത്തിലെ CE മാർക്കുകൾ എന്നീ വിവരങ്ങൾ അടങ്കുന്ന ഒരു student സ്റ്റെക്ക്‌ചർ നിർവ്വചിക്കുക. സ്റ്റെക്ക്‌ചർ പോയിന്റർ ഉപയോഗിച്ച് ഈ വിവരങ്ങൾ ഇൻപുട്ട് ചെയ്യാനും രജിസ്ട്രേഷൻ നമ്പർ, പേര്, സി.എൽ. മാർക്കുകളുടെ തുക എന്നിവ പ്രദർശിപ്പിക്കാനും വേണ്ട പ്രോഗ്രാം എഴുതുക.



## നമ്മക്ക് വിലയിരുത്താം

1. C++ ലെ അനേയും സ്ട്രuk്റ്റും താരതമ്യം ചെയ്യുക.
2. താഴെ കോടുത്തിരിക്കുന്ന സ്ട്രuk്റ്റും നിർവ്വചനത്തിലെ തെറ്റുകൾ കണ്ടെത്തി അതിനുള്ള കാരണമെഴുതുക.

```
struct
{
    int roll, age;
    float fee=1000;
};
```

3. താഴെ കോടുത്തിരിക്കുന്ന സ്ട്രuk്റ്റും നിർവ്വചനം വായിച്ച് ചോദ്യങ്ങൾക്ക് ഉത്തരം എഴുതുക.

```
struct book
{
    int book_no;
    char bk_name[20];
    struct
    {
        short dd;
        short mm;
        short yy;
    } dt_of_purchase;
    float price;
};
```

- a. ഈ സ്ട്രuk്റ്റുഡിലെ വിവരങ്ങൾ സൂചിപ്പിക്കാൻ book എന്നിൽ ഒരു വേതിയ ബിൾ പ്രവൃത്തിക്കാനുള്ള C++ പ്രസ്താവന എഴുതുക. വേതിയബിളിന് എത്ര മെമ്മറി ആവശ്യമുണ്ട്. ന്യായീകരിക്കുക.
  - b. നിങ്ങളുടെ കമ്പ്യൂട്ടർ സയൻസ് ടെക്നോളജി ബുക്കിൽ വിവരങ്ങൾ ഉപയോഗിച്ച് ഈ സ്ട്രuk്റ്റുഡിന് പ്രാരംഭവിലെ നൽകാനുള്ള C++ പ്രസ്താവന എഴുതുക.
  - c. book എൻ വിവരങ്ങൾ പ്രദർശിപ്പിക്കാൻ ആവശ്യമായ C++ പ്രസ്താവനകൾ എഴുതുക.
  - d. “അക്കദാനുള്ള സ്ട്രuk്റ്റുഡി ടാഗിംഗ് അഭാവം ഒരു തെറ്റും വരുത്തുന്നില്ല” - ഈ വാചകം ശരിയോ തെറ്റോ എന്ന് കാരണസഹിതം സമർപ്പിക്കുക.
4. “സ്ട്രുക്റ്റുഡി ഒരു ഉപഭോക്ത്യനിർവ്വചിത ധാരാത്തരമാണ്” - ഉദാഹരണ സഹിതം സമർപ്പിക്കുക.

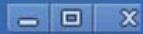
## 1. സ്റ്റ്രക്ക്‌ചറും പോയിസ്റ്റ്രും

5. താഴെ കൊടുത്തിരിക്കുന്ന വാചകങ്ങൾ വായിക്കുക:
  - i. C++ ലെ സ്റ്റ്രക്ക്‌ചർ നിർവ്വചിക്കുമ്പോൾ ടാഗ് ഒഴിവാക്കാം.
  - ii. ഒരു സ്റ്റ്രക്ക്‌ചർ വേരിയബിളിലെ ഡാറ്റ മറ്റാരു സ്റ്റ്രക്ക്‌ചർ വേരിയബിളി ലേക്ക് കോപ്പി ചെയ്യണമെങ്കിൽ രണ്ട് സ്റ്റ്രക്ക്‌ചർ വേരിയബിളുകളും ഒരേ ടാഗിൽ നിർവ്വചിച്ചിരിക്കണം.
  - iii. സ്റ്റ്രക്ക്‌ചർ അംഗങ്ങളെ `structure_name.element` എന്ന രീതിയിലാണ് സൂചിപ്പിക്കുന്നത്.
  - iv. ഒരു സ്റ്റ്രക്ക്‌ചറിന് മറ്റാരു സ്റ്റ്രക്ക്‌ചറിനെ ഉൾക്കൊള്ളാം.
- ഈ താഴെ കൊടുത്തവയിൽ ശരിയായത് തിരഞ്ഞെടുക്കുക.
  - a. (i) ഉം (ii) ഉം വാചകങ്ങൾ ശരിയാണ്
  - b. (ii) ഉം (iv) ഉം വാചകങ്ങൾ ശരിയാണ്
  - c. (i) ഉം, (ii) ഉം, (iv) ഉം വാചകങ്ങൾ ശരിയാണ്
  - d. എല്ലാ വാചകങ്ങളും ശരിയാണ്
6. ചുവടെ ചേർത്തിരിക്കുന്ന C++ പ്രസ്താവനകൾ വായിക്കുക.

```
int * p, a=5;
p=&a;
```

  - a. p എന്ന വേരിയബിളിന്റെ പ്രത്യേകതയെന്ത്?
  - b. രണ്ടാമത്തെ വാചകം പ്രവർത്തിപ്പിച്ചതിനുശേഷം p യുടെ ഉള്ളടക്കം എന്തായിരിക്കും.
  - c. \*p+1, \*(p+1) എന്നീ പദപ്രയോഗങ്ങൾ എങ്ങനെ വ്യത്യാസപ്പെടിരിക്കുന്നു?
7. ചുവടെ കൊടുത്തിരിക്കുന്ന C++ പ്രോഗ്രാം ശകലത്തിലെ തെറ്റ് കണ്ണുപിടിച്ച് കാരണമെന്തുതുക.

```
int *p,*q, a=5;
float b=2;
p=&a;
q=&b;
cout<<p<<*p<<*a;
if (p<q) cout<<p;
cout<<*p * a;
```
8. ഒരു പ്രോഗ്രാം തയാറാക്കുമ്പോൾ ദൈനാധിക് അലോക്കേഷൻ രീതി ഉപയോഗിച്ചു. എന്നാൽ `delete` ഓപ്പറേറ്റർ ഉപയോഗിച്ചുള്ള വാചകം ആ പ്രോഗ്രാമിൽ ഇല്ലായിരുന്നു. ഈത് സൃഷ്ടിക്കുന്ന പ്രശ്നം വിവരിക്കുക.



9. താഴെ കൊടുത്തിരിക്കുന്ന C++ പ്രസ്താവനകൾ വായിച്ച് ചോദ്യങ്ങൾക്ക് ഉത്തരമുള്ളതുക.

```
int mark[] = {34, 12, 25, 56, 38};  
int *p = mark;
```

- a. p യുടെ വിലയെന്ന്
- b. \*p + \*(ar+2) എന്ന പദ്ധത്യാഗത്തിൽ വില എന്നാണ്?
- c. ar++; എന്ന വാചകം തെറ്റാണ് - എന്തുകൊണ്ട്? ഇത് p++; എന്ന വാചകത്തിൽ നിന്നും എങ്ങനെ വ്യത്യാസപ്പെടിരിക്കുന്നു.

10. താഴെ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം ശകലത്തിൽ പ്രവർത്തനം വിവരിച്ച് ഒരുപുട്ട് കണ്ണഭ്രംതുക.

```
char *str = "Tobacco Kills";  
for (int i=0; str[i]!='\0'; i++)  
    if (i>8)  
        *(str+i) = toupper(*(str+i));  
cout<<str;
```

11. താഴെ കൊടുത്ത C++ പ്രസ്താവനകൾ ശ്രദ്ധിക്കുക.

```
int ar[] = {14, 29, 32, 63, 30};
```

ചുവടെ ചേർത്തവയിൽ ഒരു വാചകം 32 എന്ന അംഗത്വത്തെ നൽകുന്നില്ല. എത്ര സ്ഥാനങ്ങൾ ഉണ്ട്?

- a. ar[2]
- b. ar[\*ar%3]
- c. \*ar+2
- d. \* (ar+2)

12. new, delete എന്നീ ഓപ്പറേറ്ററുകളുടെ പ്രവർത്തനങ്ങൾ ഉദാഹരണ സഹിതം വിവരിക്കുക.

13. മെമ്മറി ലൈംഗ് എന്നാലെന്ത്? എന്താണ് ഇതിന് കാരണം? ഈ സാഹചര്യം എങ്ങനെ ഒഴിവാക്കാം?

14. ചുവടെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകൾ താരതമ്യം ചെയ്യുക.

```
int a=5;  
int *a=new int(5);
```

15. ചുവടെ കൊടുത്തിരിക്കുന്ന സ്ട്രuk്ചർ നിർവ്വചനം വായിച്ച് ചോദ്യങ്ങൾക്ക് ഉത്തരമുള്ളതുക.

```
struct sample  
{  
    int num;  
    char *str;  
} *sptr;
```



- a. sample ഡാറ്റാ തരത്തിലെ ഒരു ലോകേഷൻിൽ ദൈനാമിക് അലോക്കേഷൻ നടത്തി അധീസ്റ്റ് Sptr ത്രക്കേയിക്കാൻ വേണ്ട C++ പ്രസ്താവന എഴുതുക.
- b. Sptr പോയിന്റ് ചെയ്യുന്ന മെമ്മറി സ്ഥാനത്തെക്ക് ഡാറ്റ ഇൻപുട്ട് ചെയ്യാനുള്ള C++ പ്രസ്താവനകൾ എഴുതുക.
- c. ഈ സ്റ്റെക്ചർിനെ സ്വയം സൂചിത്ത് സ്റ്റെക്ചരായി നവീകരിക്കുക.





## ഒമ്പജക്ക് ഓറിയൻ്റെ പ്രോഗ്രാമിങ്ങിന്റെ തത്ത്വങ്ങൾ

2

### പ്രധാന പഠനരേഖകൾ



ഈ അധ്യായത്തിൽ പുർത്തീകരണത്തോടെ പരിതാവിന്ന്

- വിവിധ പ്രോഗ്രാമിങ്ങ് മാതൃകകളെ താരതമ്യം ചെയ്യാൻ സാധിക്കുന്നു.
- പ്രോസൈജർ ഓറിയൻ്റെ പ്രോഗ്രാമിങ്ങ് മാതൃകയുടെ വിവിധ സവിശേഷതകൾ സൂചിപ്പിക്കാൻ സാധിക്കുന്നു.
- ഒമ്പജക്ക് ഓറിയൻ്റെ പ്രോഗ്രാമിങ്ങ് മാതൃകയുടെ ഗുണങ്ങൾ സൂചിപ്പിക്കാൻ കഴിയുന്നു.
- ഡാറ്റ അബ്സ്ട്രക്ഷൻ, ഡാറ്റ എൻക്റൂപ്പ് പസ്യലേഷൻ എന്നീ ആശയങ്ങൾ ഉദാഹരണസഹിതം വിശദീകരിക്കാൻ സാധിക്കുന്നു.
- ഇൻഡോറിറ്റൻസ്, പോളിമോർഫിസം എന്നിവ യമാർമ്മ ജീവിത ഉദാഹരണങ്ങൾ ഉപയോഗിച്ച് വിശദീകരിക്കാൻ സാധിക്കുന്നു.

കമ്പ്യൂട്ടറുകളെ കൂടുതൽ ഉപയോഗപ്രദമാക്കുന്നതിനായി നമ്മൾ പ്രോഗ്രാമുകൾ (Software) വികസിപ്പിക്കുന്നു. അതിനായി വിവിധ പ്രോഗ്രാമിങ്ങ് ഭാഷകൾ നമ്മൾ ഉപയോഗിക്കുന്നു. പ്രോഗ്രാമിന്റെ വലിപ്പം കൂടുതോറും അത് നിർമ്മിക്കുവാനുള്ള ബുദ്ധിമുട്ടും കൂടുന്നു. ഈ ബുദ്ധിമുട്ട് ഒഴിവാക്കുന്നതിനായി IDE, ഡൈവെഗ്രേഡ്, കംബേപലർ എന്നിങ്ങനെ വിവിധ ഉപാധികൾ നമുക്ക് ഉപയോഗിക്കാം. കൂടാതെ വിവിധ നിർമ്മാണസമീപന രീതികളായ മോഡ്യുലാർ, സ്റ്റെക്ചർഡ്, പ്രോസൈജറൽ, ഒമ്പജക്ക് ഓറിയൻ്റെ എന്നിവ സോഫ്റ്റ്‌വെയർ നിർമ്മാണത്തിന് നമുക്ക് പിന്തുടരാവുന്നതാണ്. ഈ ഉപാധികളും സമീപനങ്ങളും സോഫ്റ്റ്‌വെയർ നിർമ്മാണ സമയത്തെ സോഫ്റ്റ്‌വെയർ ഉൽപ്പന്നങ്ങളുടെ പരിപാലനം (Maintainability), പുനരുപയോഗം (Reusability), വഹനീയത, (Portability), സുരക്ഷിതത്താം, സമഗ്രത (Integritiy), ഉപയോക്തസ്തുപ്രയോഗം മുതലായ പ്രസ്താവങ്ങളെ അഭിമുഖീകരിക്കാൻ സഹായിക്കുന്നു.

മുൻ അധ്യായത്തിൽ വിവിധ പ്രോഗ്രാമിങ്ങ് ആശയങ്ങളെ പറിയും C++ ഭാഷ ഉപയോഗിച്ച് പ്രശ്ന പരിഹാരത്തിനായി പ്രോഗ്രാമുകൾ നിർമ്മിക്കുന്നതിനെ പറിയും നമ്മൾ പരിച്ചതാണ്. ഇൻപുട്ടുകളെ പ്രോസസ്സ് ചെയ്ത് ഒരുപ്പുകൾ ലഭ്യമാകുക എന്നതായിരുന്നു അവിടെയെല്ലാം നമ്മുടെ ലക്ഷ്യം. തയാറാക്കിയ പ്രോഗ്രാമുകൾ ചെറുതും ലളിതവും ആയിരുന്നതിനാൽ കൈക്കൊണ്ട സമീപനങ്ങളെ പറ്റിയോ ഡാറ്റയുടെ സുരക്ഷിതത്തെത്തെ പറ്റിയോ നമ്മൾ ഒരിക്കലും ചിന്തിച്ചിരുന്നില്ല.

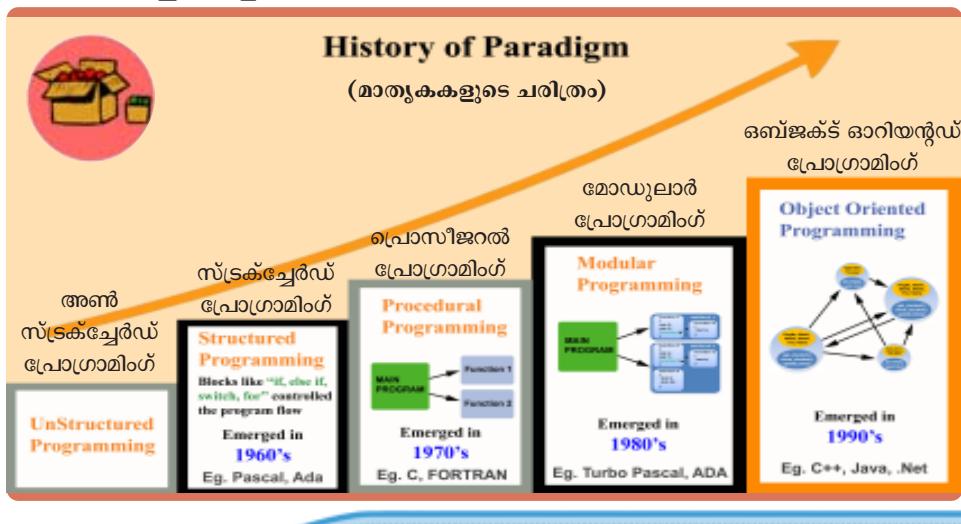
എന്നാലും പ്രശ്ന പരിഹാരത്തിനായി പ്രോഗ്രാമുകൾ തയ്യാറാക്കുന്നവോൾ അറിഞ്ഞെന്നൊരു അറിയാതയോ നാം ഒരു സമീപനം വിനൃതിക്കുന്നു കൊണ്ടിരുന്നു. പ്രോഗ്രാം വികസനത്തിനായി ഏറ്റവും ഉചിതമായ സമീപനം തിരഞ്ഞെടുക്കാൻ നമ്മുൾപ്പെടെ പ്രോഗ്രാമുകളും ഇവ അധ്യായത്തിൽ, നാം ഈ വരെ ഉപയോഗിച്ച് പോന്ന പ്രോഗ്രാമിങ്ങ് സമീപനത്തെപ്പറ്റി ചർച്ച ചെയ്യുകയും പുതിയ സമീപനം അവതരിപ്പിക്കുകയും ചെയ്യുന്നു.

## 2.1 പ്രോഗ്രാമിങ്ങ് മാതൃക (Programming paradigm)

ഒരു പ്രോഗ്രാം എങ്ങനെ ചിട്ടപ്പെടുത്തിയിരിക്കുന്നു എന്നതിനെ സുചിപ്പിക്കുന്നതാണ് പ്രോഗ്രാമിങ്ങമാതൃക. പ്രോഗ്രാം വളരെ ചെറുതാണെങ്കിൽ പ്രോഗ്രാം ചിട്ടപ്പെടുത്താൻ പ്രത്യേക തത്ത്വമൊന്നും പാലിക്കേണ്ട കാര്യമില്ല. എന്നാൽ പ്രോഗ്രാം വലുതാകുംതോറും അതിന്റെ സക്രീംന്ത കുറയ്ക്കുവാനും അതിനെ പരിപാലിക്കുന്നതിനും ചില മുൻകരുതലുകൾ എടുക്കേണ്ടി വരുന്നു.

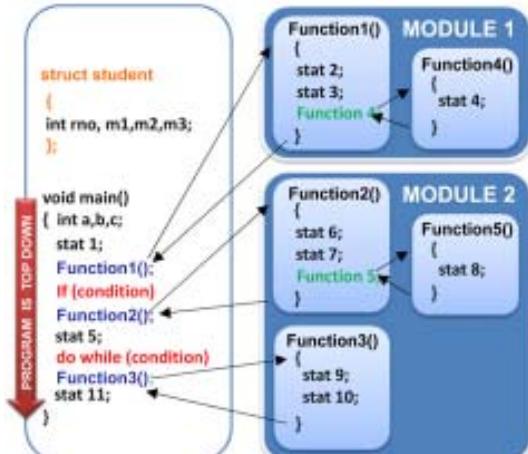
ചില മാതൃകകൾ നടപടിക്രമങ്ങൾക്കു കൂടുതൽ പ്രാധാന്യം കൊടുക്കുന്നവോൾ മറ്റു ചിലത് ഡാറ്റക്സ് കൂടുതൽ പ്രാധാന്യം കൊടുക്കുന്നു. വിവിധ പ്രോഗ്രാമിങ്ങ് ഭാഷകളുടെ കാര്യക്ഷമതയും ശ്രദ്ധിയുമെല്ലാം അവ അവലംബിക്കുന്ന പ്രോഗ്രാം മാതൃകകളാണ് നിർവ്വചിക്കുന്നത്. മൊധുലാർ പ്രോഗ്രാമിങ്ങ്, ഫോപ് ഡാബ് ഡൈസ് പ്രോഗ്രാമിങ്ങ്, ബോട്ടം അപ്പ് പ്രോഗ്രാമിങ്ങ്, സ്ട്രക്ചർഡ് പ്രോഗ്രാമിങ്ങ് മുതലായ രീതികൾ നേരത്തെ പരീക്ഷിക്കപ്പെട്ടവയാണ്. ഈവ ഓരോനും പ്രോഗ്രാമിങ്ങിന്റെ സക്രീംന്ത കുറയ്ക്കുവാനും വിശദ്ദിച്ചവും പരിപാലന യോഗ്യവുമായ പ്രോഗ്രാമുകൾ നിർമ്മിക്കുവാനും വേണ്ടിയാണ് ഉപയോഗിച്ചത്.

ഒരു പ്രോഗ്രാമിങ്ങ് മാതൃക മാത്രം വിനൃതിക്കുന്ന രീതിയിലാണ് ചില പ്രോഗ്രാമിങ്ങ് ഭാഷകൾ രൂപകൽപ്പന ചെയ്തിരിക്കുന്നത്. എന്നാൽ മറ്റു ചിലത് ഒന്നിലധികം മാതൃകകൾ പിനുറുന്നു. C++ ഒന്നിലധികം പ്രോഗ്രാമിങ്ങ് മാതൃകകൾ പ്രയോജനപ്പെടുത്തുന്ന ഭാഷയാണ്. C++ ഉപയോഗിച്ച് ഏറ്റവും പ്രധാനപ്പെട്ട രീതു മാതൃകകളായ പ്രോസിജറൽ മാതൃകയും (Procedural paradigm) ഒബ്ജക്ട് ഓറിയൻ്റെ മാതൃകയും (Object Oriented Paradigm) നമുക്ക് പ്രാവർത്തികമാക്കാവുന്നതാണ്. ഈവ ഓരോനിനെയും പറ്റി നമുക്ക് വിശദമായി ചർച്ച ചെയ്യാം.



## 2.1.1 പ്രോസിജർ ഓറിയൻ്റെ പ്രോഗ്രാമിങ് മാതൃക (Procedure-Oriented Programming paradigm)

പ്രോസിജർ ഓറിയൻ്റെ പ്രോഗ്രാമിങ്, ഒരു പ്രോഗ്രാം നിർമ്മിക്കാൻ ആവശ്യമായ നല്ല രീതിയിൽ ക്രമീകരിച്ച ഘട്ടങ്ങളുടെ ശ്രേണിയെയും പ്രവർത്തനങ്ങളെയും സൂചിപ്പിക്കുന്നു. ഒരു പ്രവ്യതിയെ അല്ലെങ്കിൽ പ്രോഗ്രാമിനെ പുർത്തീകരിക്കുന്നതിനായുള്ള ചിട്ടയായ ക്രമീകരണങ്ങൾ ഉള്ള പ്രസ്താവനകളും നടപടികളും നിർദ്ദേശങ്ങളും ഇവയിൽ അടങ്കിയിരിക്കും. ഈ നിർദ്ദേശങ്ങൾ ഇൻപുട്ട് സ്വീകരിക്കാനുള്ള തോ, ഗണിതപരമോ യുക്തിപരമോ ആയ ക്രീയകൾ നിർവഹിക്കാനുള്ള തോ, അതുമല്ലെങ്കിൽ പരിണിതമഹലം



ചിത്രം 2.1 പ്രോസിജർ മാതൃക

പ്രദർശിപ്പിക്കാനുള്ളതോ ആവാം. ഈ സമീപത്തിൽ കാര്യങ്ങൾ ചെയ്യുന്നതിനാണ് ഉള്ളത് നൽകിയിരിക്കുന്നത്. ഈ മാതൃകയിൽ പ്രോഗ്രാം സകീർണ്ണവും വലുപ്പം കുടിയതുമാകുമ്പോൾ നിർദ്ദേശങ്ങളുടെ പട്ടികയെ വിജിച്ചു നിർദ്ദേശങ്ങളെ ഫംഗ്ഷനുകളായി രൂപപ്പെടുന്നു. പ്രോഗ്രാമിലെ മറ്റ് ഫംഗ്ഷൻ വ്യക്തമായി നിർവചിക്കുന്നു. അതു വഴി പ്രോഗ്രാമിന്റെ സകീർണ്ണത കുറക്കാൻ ഫംഗ്ഷൻ സഹായിക്കുന്നു. സകീർണ്ണത വീണ്ടും കുറയ്ക്കുന്നതിനായി പൊതുവായ ക്രീയകളുമായി ബന്ധപ്പെട്ട ഫംഗ്ഷനുകളെ മോഡ്യൂളുകളായി രൂപപ്പെടുത്തുന്നു. ഒരു വലിയ പ്രോഗ്രാമിനെ അഞ്ചു വ്യത്യസ്ത ഫംഗ്ഷനുകളായി വിഭജിക്കുകയും പൊതുവായ പ്രവർത്തനങ്ങൾ ഉൾപ്പെടുന്ന ഫംഗ്ഷനുകളെ കൂടിച്ചേർത്തു രണ്ട് മോഡ്യൂളുകളുകയും ചെയ്യുന്ന പ്രോസിജർ ഓറിയൻ്റെ മാതൃകയാണ് ചിത്രം 2.1 തോണിച്ചിരിക്കുന്നത്.

C, Pascal, FORTRAN, BASIC മുതലായവ പ്രോസിജർ ഓറിയൻ്റെ പ്രോഗ്രാമിങ് ഭാഷകളാണ്. C++ പ്രോഗ്രാമിങ് പതം പ്രക്രിയയിൽ നമ്മൾ ഇത് വരെ പിന്തുടരുന്നത് പ്രോസിജർ ഓറിയൻ്റെ മാതൃകയാണ്. പ്രോസിജർ ഓറിയൻ്റെ പ്രോഗ്രാമിങ് ഭാഷകളെ ദോപ് ഡൗൺ പ്രോഗ്രാമിങ് ഭാഷകൾ എന്നും വിളിക്കുന്നു.

പ്രക്രിയ പ്രോഗ്രാം വലുപ്പം കുടുകയും സകീർണ്ണത വർധിക്കുകയും ചെയ്യുന്നതോടെ അതിന്റെ പരിമിതികൾ വെളിപ്പെട്ടു തുടങ്ങുന്നു. ഈ സമീപനും എത്ര നല്ല രീതിയിൽ പ്രാവർത്തികമാക്കിയാലും വലിയ പ്രോഗ്രാമുകൾ സകീർണ്ണമായിത്തന്നെ തുടരുന്നു. പ്രോസിജർ ഓറിയൻ്റെ ഭാഷകളുടെ സകീർണ്ണത വർധിക്കുവാനുള്ള പ്രധാന കാരണങ്ങൾ താഴെ പറയുന്നവയാണ്.

- ഡാറ്റയുടെ പ്രാധാന്യം കുറച്ചു കാണുന്നു.
- പുതിയ ഡാറ്റ അംഗത്വത്തെ കൂടിച്ചേർക്കുമ്പോൾ എല്ലാ ഫംഗ്ഷനുകൾക്കോ അല്ലെങ്കിൽ ചില ഫംഗ്ഷനുകൾക്കോ മാറ്റം വേണ്ടി വരുന്നു.

c. പുതിയ ഡാറ്റ തരങ്ങൾ നിർമ്മിക്കുക പ്രയാസകരമാണ്.

d. അമാർമ ജീവിത മാതൃകക്കു യോജിച്ചതല്ല.

നമുക്ക് ഈത് ഓരോന്നിനെക്കുറിച്ചും ഇവിടെ ചർച്ച ചെയ്യാം.

#### a. ഡാറ്റയുടെ പ്രാധാന്യം കുറച്ചു കാണുന്നു

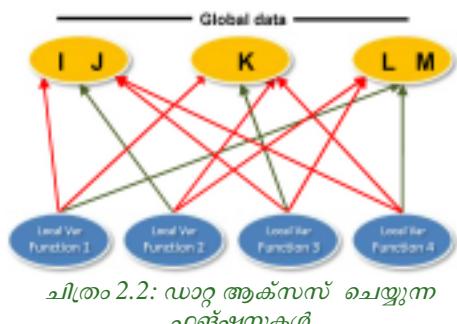
പ്രോസിജർ ഓറിയൻ്റെ ഭാഷകൾ, കാരുജങ്ങൾ ചെയ്യുന്നതിനാണ് പ്രാമുഖ്യം നൽകുന്നത്. ഡാറ്റയ്ക്ക് പ്രാധാന്യം കുറവാണ്. ഒരു ഉദാഹരണത്തിന്റെ സഹായത്തോടെ ഈത് വിശദീകരിക്കാം. സ്ക്രൂളിലെ പ്രവർത്തനങ്ങളെ തന്റെ വർത്തകരിക്കുന്നതിനായി ഒരു സോഫ്റ്റ്‌വെയർ വികസിപ്പിക്കുന്നു എന്ന് കരുതുക. ഒരു പുതിയ വിദ്യാർമ്മിയെ ഉൾപ്പെടുത്തുക, ഒരു വിദ്യാർമ്മിയെ നീക്കു ചെയ്യുക, ഫീ ശേഖരണ വിവരങ്ങൾ രേഖപ്പെടുത്തുക മുതലായവയാകാം ഈതിൽ ഉൾപ്പെടുന്ന പ്രവർത്തനങ്ങൾ. സോഫ്റ്റ്‌വെയറിൽ ഈ പ്രവർത്തനങ്ങൾക്കായി പ്രത്യേക ഫംഗ്ഷനുകൾ ഉപയോഗിച്ചിരിക്കുന്നു എന്ന് അനുമാനിക്കുക. വിദ്യാർമ്മികളുടെ വിവരങ്ങൾ അടങ്കിയ സ്ട്രേക്ഷൻുകളുടെ (സുധാരണ) അരേ ഉപയോഗിച്ച് ഈ ഫംഗ്ഷനുകളുടെ പ്രവർത്തനത്തിനായുള്ള ഡാറ്റ സംഭരിക്കാവുന്നതാണ്. ഈ ഡാറ്റ എല്ലാ ഫംഗ്ഷനുകൾക്കും ലഭ്യമാക്കണമെങ്കിൽ അരേയെ ദ്രോബത്ത് ആയി നിർവ്വചിക്കേണ്ടി വരും (അനാം വർഷ പാഠപുസ്തകത്തിന്റെ ഭാഗം 10.5 നേരക്കുക). ഈപ്പോൾ ഈ ഫംഗ്ഷനുകളെ കൂടാതെ പ്രോഗ്രാമിലുള്ള മറ്റ് എത്രു ഫക്ഷൻുകൾക്കു വേണമെങ്കിലും ഡാറ്റ ഉപയോഗിക്കാവുന്ന അവസ്ഥയായി. തന്മുലം അറിഞ്ഞേണ്ട അറിയാതെയോ ഡാറ്റയ്ക്ക് മാറ്റം സംഭവിക്കാനുള്ള കളമാരുങ്ങുന്നു. ഈത് തടയുവാൻ നമുക്ക് ധാതൊരു തരത്തിലുമുള്ള നിയന്ത്രണവും ഏർപ്പെടുത്താൻ സാധിക്കും.

അടുത്ത ദിവസം സമർപ്പിക്കേണ്ട അസൈൻമെന്റ് തീൻ മേശപ്പുറത്തു വെച്ചിട്ടു പോകുന്നത് പോലെയാണിത്. ഈ അസൈൻമെന്റ് നാശമാകാനുള്ള സാധ്യത ഇവിടെ വളരെ കുടുതലാണ്. ചെറിയ കൂട്ടികളുണ്ടെങ്കിൽ അവർ അത് കീറുകയോ അല്ലെങ്കിൽ അതിൽ ചിത്രങ്ങൾ വരയ്ക്കുകയോ ചെയ്യാം അതുമല്ലെങ്കിൽ മേശപ്പുറത്തു വച്ചിരിക്കുന്ന ചായ അബദ്ധത്തിൽ അതിന്റെ മേൽ മറിഞ്ഞു വീഴാം. ആർക്കു വേണമെങ്കിലും എടുക്കാവുന്ന രീതിയിൽ അസൈൻമെന്റ് വെക്കുന്നത് കൊണ്ടാണ് ഈത് സംഭവിക്കുന്നത്.

ഒരു പ്രോസിജറിൽ ഓറിയൻ്റെ പ്രോഗ്രാമിങ്ങ് മാതൃകയിലെ ലോകത്ത് വേരിയബിളുകൾ, ദ്രോബത്ത് വേരിയബിളുകൾ, ഫംഗ്ഷൻുകൾ മുതലായവയുടെ ക്രമീകരണമാണ് ചിത്രം 2.2 തീ കാണിച്ചിരിക്കുന്നത്. ഡാറ്റ ആവശ്യമായ ഫക്ഷൻുകളെ പച്ച വരകൾ സൂചിപ്പിക്കുന്നു. ഡാറ്റയുടെ അന്യായിക്കുത്തമായ ഉപയോഗത്തെ ചുവപ്പു വരകൾ സൂചിപ്പിക്കുന്നു.

b. പുതിയ ഡാറ്റ അംഗത്വത്തെ കൂട്ടിച്ചേർക്കുവോൾ എല്ലാ ഫംഗ്ഷനുകൾക്കോ അല്ല കിൽ ചില ഫംഗ്ഷൻുകൾക്കോ മാറ്റം വേണ്ടി വരുന്നു

പല ഫംഗ്ഷനുകളും ദ്രോബത്ത് ഡാറ്റ ഉപയോഗിക്കുന്നതിനാൽ ഡാറ്റ എങ്ങനെ സംഭരിച്ചിരിക്കുന്നു എന്നത് ഏറെ പ്രാധാന്യമുള്ളതാണ്. ഡാറ്റ ഉപയോഗിക്കുന്ന



പണ്ഡിതനുകളെ പരിഷ്കരിക്കാതെ ഡാറ്റയുടെ ക്രമീകരണത്തിൽ മാറ്റം വരുത്തുക സാധ്യമല്ല. നമ്മൾ പുതിയ ഡാറ്റ അംഗത്വത്തെ കൂട്ടി ചേർക്കുന്നോൾ ആ ഡാറ്റ ഉപയോഗിക്കുന്ന എല്ലാ ഫക്ഷനുകളെയും പരിഷ്കരിക്കേണ്ടതുണ്ട്. എന്നാൽ മാത്രമേ ഈ ഫാർശനുകൾക്കു പുതിയ ഡാറ്റ അംഗം ലഭ്യമാകു. ഇത്തരത്തിൽ ഒരു ഡാറ്റയുമായി ബന്ധപ്പെട്ട എല്ലാ ഫാർശനുകളെയും കണ്ണംതുക എന്നത് വളരെ ശ്രദ്ധകരമായ കാര്യമാണ് എന്ന് മാത്രമല്ല പുതിയ ഡാറ്റ ലഭ്യമാക്കുന്നതു രീതിയിൽ അവയെ കൃത്യമായി പരിഷ്കരിക്കുക എന്നത് അതിലും ബുദ്ധിമുട്ടേറിയതാണ്.

നമ്മുടെ സ്കൂൾ സോഫ്റ്റ്‌വെയറിലെ സ്ക്രീനിൽ സ്ട്രെക്ചർലോക്കു ‘എം’ എന്നാരു പുതിയ ഇന കൂട്ടിച്ചേരിക്കണം എന്നിരിക്കുന്നു. സോഫ്റ്റ്‌വെയറിൽ സുഗമമായ പ്രവർത്തനത്തിന് സ്ക്രീനിൽ ഡാറ്റയുമായ ഡാറ്റ ഉപയോഗിക്കുന്ന എല്ലാ ഫാർശനുകളെയും കണ്ണംതു അവയിൽ ആവശ്യമായ മാറ്റം വരുത്തൽ അനിവാര്യമാണ്.

#### c. പുതിയ ഡാറ്റ ദൈപ്പുകൾ നിർമ്മിക്കുന്നത് ബുദ്ധിമുട്ടാണ്

ഇൻറിജർ, മ്പ്ലോട്ട്, ക്യാരക്ടർ മുതലായ ഡാറ്റ ദൈപ്പുകൾ ആണ് കമ്പ്യൂട്ടർ ഭാഷകളിൽ സാധാരണയായി ഉപയോഗിക്കാറുള്ളത്. ചില പ്രോഗ്രാമിൽ ഭാഷകൾ അടിസ്ഥാന ഡാറ്റാതരങ്ങളെ കുടാതെ മറ്റു ഡാറ്റാതരങ്ങൾ നിർമ്മിക്കാൻ അനുവാദം നൽകിയിരിക്കുന്നു. അതിനർത്ഥം അവ വിപുലീകരിക്കാവുന്നവയാണ് എന്നാണ്. അടിസ്ഥാന രൂപകൽപ്പനകളിൽ മാറ്റം വരുത്തുകയോ അല്ലെങ്കിൽ പ്രോഗ്രാം കോഡിൽ കാര്യമായ മാറ്റം വരുത്താതെ പ്രോഗ്രാമിൽ കഴിവും കാര്യക്ഷമതയും കാര്യമായ വർധന വരുത്തുകയോ ചെയ്യാനുള്ള കഴിവിനന്നാണ് വിപുലീകരണ സാധ്യത എന്ന് പറയുന്നത്. ഈ സവിശേഷത പ്രോഗ്രാമുകളുടെ സങ്കീർണ്ണത കുറക്കുവാൻ സഹായിക്കുന്നതിനോടൊപ്പം അതിൽ കഴിവുകൾ വർധിപ്പിക്കുന്നതും ചെയ്യുന്നു. പ്രോസീജറൽ പ്രോഗ്രാമുകൾ വിപുലീകരണ സാധ്യതയില്ലാത്തവയാണ്.

#### d. യമാർമ്മ ജീവിത മാതൃകയ്ക്കു ഡോജിപ്പത്വം

പ്രോസീജറൽ പ്രോഗ്രാമിൽ മാതൃകയിൽ ഉപയോഗിക്കുന്ന ഫാർശനുകളെയും പ്രോഗ്രാമുകളെയും ഒറ്റ ഘടകമായി പരിശീലനിക്കുകയില്ല. മറിച്ചു അവ ഓരോനും സത്രന്മായി നിലകൊള്ളുന്നു. അതിനാൽ ഈ മാതൃകയിൽ ഉപയോഗിക്കുന്ന ഡാറ്റയും ഫാർശനുകളും ഒരു യമാർമ്മ മാതൃകയായി പറയാനാവില്ല.

ഉദാഹരണത്തിന് നമ്മുടെ സ്കൂൾ സോഫ്റ്റ്‌വെയറിൽ നമ്മൾ വിദ്യാർഥികളുടെ ഡാറ്റയും അവയെ കൈകാര്യം ചെയ്യുന്ന ഫാർശനുകളും ഉപയോഗിക്കുന്നു. അതുപോലെ അധ്യാപകരുടെ ഡാറ്റയും അവ കൈകാര്യം ചെയ്യുന്ന ഫാർശനുകളും ഉണ്ടായിരിക്കും. പ്രോസീജറൽ മാതൃകയാണെങ്കിൽ വിദ്യാർഥികളുടെ ഡാറ്റയും അവയുടെ ഫാർശനുകളും ഒറ്റ ഘടകമാക്കാനും അതുപോലെ അധ്യാപകരുടെ ഡാറ്റയും ഫാർശനുകളും മറ്റാരു ഘടകമാക്കി മാറ്റാനും സാധിക്കില്ല.

നിത്യജീവിതത്തിലെ ഓരോ വസ്തുവും പരിശോധിച്ചാൽ, അവയെല്ലാം അവയുടെ സവിശേഷതകളും സഭാവ ഗുണങ്ങളും ഒരു ചേർന്ന ഒറ്റ ഘടകമാണെന്ന് നമുക്ക് കാണാം. ഉദാഹരണത്തിന് മനുഷ്യനെ പരിശീലിച്ചാൽ, പേര്, പഞ്ചത്വം, ലിംഗം മുതലായവ സവിശേഷതകളും സംസാരം, ചിരി മുതലായവ സഭാവ ഗുണങ്ങളും ആവാം. പ്രോസീജറൽ മാതൃകയിൽ, സഭാവ ഗുണങ്ങളെ ഫാർശനുകളായും സവിശേഷതകളെ ഡാറ്റയായും സുചിപ്പിക്കാമെങ്കിലും അവയെ ഒറ്റ ഘടകമാക്കി മാറ്റാൻ സാധിക്കാത്തതിനാൽ, ഈ മാതൃക യമാർമ്മ ജീവിത മാതൃകയുമായി സാമ്യമില്ലാത്തതാണ്.

അടുത്തതായി ബെംജക്ക് ഓറിയൻ്റെ മാതൃകയെപ്പറ്റി ചർച്ച ചെയ്യുകയും അത് പ്രോസിജറൽ മാതൃകയുടെ പരിമിതികൾ എന്നേനെ മറികടക്കുന്നു എന്ന് കാണുകയും ചെയ്യാം.

### നിങ്ങളുടെ പുരോഗതി അറിയുക



1. താഴെ പറയുന്ന മൂന്നു പ്രസ്താവനകൾ ശരിയോ തെറ്റോ എന്ന് പറയുക:
  - a. ഫ്രോബെൽ വേരിയബിളുകൾ നന്ദിയിക്കുന്ന ഫലംഷനുകളിൽ ഉപയോഗിക്കാൻ സാധിക്കില്ല.
  - b. ധമാർമ്മ ലോകവുമായി വളരെ സാമ്യമുള്ളതാണ് പ്രോസിജറൽ മാതൃക.
  - c. പ്രോസിജറൽ മാതൃകയിൽ ഡാറ്റയും ഫംഗ്ഷനുകളും ഒറ്റ ഘടകമായി കൂട്ടിച്ചേര്ക്കാൻ സാധിക്കില്ല.
2. താഴെ പറയുന്നവയിൽ നിന്ന് പ്രോസിജറൽ ഭാഷകൾ തിരഞ്ഞെടുക്കുക.
   
C, C++, ഫോർട്ടാൻ, ജാവ, പാസ്കൽ

## 2.1.2 ബെംജക്ക് ഓറിയൻ്റെ പ്രോഗ്രാമിങ്ങ് മാതൃക (Object-Oriented Programming (OOP) paradigm)

ബെംജക്ക് ഓറിയൻ്റെ പ്രോഗ്രാമിങ്ങ് മാതൃകയിൽ പ്രോസിജറൽ മാതൃകയിലെ പരിമിതികൾ പരിഹരിച്ചു കൊണ്ട് ഡാറ്റയെയും അവയുടെ ഫലംഷനുകളും ഫലംഷനുകളും കൂട്ടിച്ചേര്ത്ത് ഒറ്റ ഘടകമാക്കി മാറ്റുന്നു. ഈ ഘടകത്തെ ബെംജക്ക് (object) എന്ന് വിളിക്കുന്നു.

ഉദാഹരണത്തിന് സ്കൂൾ സോഫ്റ്റ്‌വെയർ ഒപ്പുവെച്ചുള്ള OOP നടപ്പിലാക്കിയാൽ വിദ്യാർത്ഥികളുടെ ഡാറ്റയും അവ ഉപയോഗപ്പെടുത്തുന്ന ഫലംഷനുകളും കൂട്ടിയോജിപ്പിച്ചു നൂറ്റാണ്ട് (Student) എന്ന ബെംജക്ക് നമുക്ക് നിർമ്മിക്കാം. അത് പോലെ അധ്യാപകരുടെ ഡീച്ചർ (Teacher) എന്നൊരു ബെംജക്ക് നിർമ്മിക്കാം. ഒരു ബെംജക്ക് ടിലുള്ള ഫലംഷനുകൾക്ക് അനുവാദം കൂടാതെ മറ്റൊരു ബെംജക്കും ഡാറ്റയെ ഉപയോഗിക്കുവാൻ സാധിക്കുകയില്ല.

Student Object



Teacher Object



ചിത്രം 2.3 ഡാറ്റകളും ഫലംഷനുകളും അടങ്കുന്ന ബെംജക്ക് കൾ

### OOP ഉപയോഗിക്കുന്നത് കൊണ്ടുള്ള മേരകൾ :

- a. സ്പഷ്ടമായ ഘടകങ്ങൾ അടങ്കിയ ഒരു പ്രോഗ്രാമിങ്ങ് ഘടന OOP പ്രദാനം ചെയ്യുന്നു.
- b. അബ്സ്റ്റ്രക്ട് ഡാറ്റക്കുകൾ നിർവ്വചിക്കാൻ തുടർച്ചയാണ്.

- c. എങ്ങനെയാണ് പ്രയോഗത്തിൽ വരുത്തിയിരിക്കുന്നത് എന്നുള്ള വിവരങ്ങൾ മറ്റുള്ള അടക്കങ്ങളിൽ നിന്നും മറച്ചു വയ്ക്കാനും വ്യക്തമായ ഒരു സമർക്കമുഖം പ്രദാനം ചെയ്യാനും ഇത് സഹായിക്കുന്നു.
- d. നിലവിലുള്ളവയെ ബാധിക്കാത്ത വിധത്തിൽ പുതിയ ഒബ്ജക്റ്റുകൾ നിർമ്മിക്കാവുന്നതിനാൽ നിലവിലുള്ള കോഡിനെ മാറ്റാനും പരിപാലിക്കാനും മറ്റും എളുപ്പമാണ്.
- e. തമാർമ്മ ജീവിത സാഹചര്യങ്ങളെ പ്രയോഗത്തിൽ വരുത്താൻ സഹായിക്കുന്നു.
- f. ഓപ്പറേറ്ററുകൾക്കു പുതിയ ധാരാബൈപ്പുകളും പുതിയ പ്രവർത്തനങ്ങളും നിർവ്വചിക്കാൻ സഹായിക്കുന്നു.

## 2.2 OOP സ്റ്റ്രേ അടിസ്ഥാന തത്വങ്ങൾ (Basic concepts of OOP)

ഒബ്ജക്റ്റുകൾ, ക്ലാസ്സുകൾ, ഡാറ്റ അബ്സ്ട്രക്ഷൻ (Data Abstraction), ഡാറ്റ എൻക്യൂപ്പാസ്യുലേഷൻ (Data Encapsulation), മോഡിളാറ്റിംഗ് (Modularitiy), ഇൻഹെരിറ്റൻസ് (Inheritance), പോളിഫോർമിസം (Polymorphism) മുതലായ ആശയങ്ങൾ നൽകുന്നതിലൂടെ ഒബ്ജക്റ്റുകൾ ഓഡിയൻസ് പ്രോഗ്രാമിങ് സോഫ്റ്റ്‌വെയർ വികസനവും പരിപാലനവും ലളിതമാക്കുന്നു. ഈ ആശയങ്ങൾ നമുക്ക് വിശദമായി ചർച്ച ചെയ്യാം.

### 2.2.1 ഒബ്ജക്റ്റുകൾ (Objects)

നമുക്ക് ചുറ്റുമുള്ള ഏതൊരു വസ്തുവിനെയും ഒബ്ജക്റ്റൊയി പരിഗണിക്കാം. എല്ലാ ഒബ്ജക്റ്റുകൾക്കും ഗുണവിശേഷങ്ങളും (ഡാറ്റ/അംഗാ/അവസ്ഥ) പ്രവർത്തന രീതികളും (ഫംഗ്ഷൻകൾ/മെത്രേറ്റീസ്) ഉണ്ടായിരിക്കും. ചിത്രം 2.4 ലെ ഒബ്ജക്റ്റുകളും അവയുടെ ഗുണവിശേഷങ്ങളും മെത്രേറ്റീകളും ഉദാഹരണസഹിതം പട്ടികപ്പെടുത്തിരിക്കുന്നു.

<b>Student</b>	<b>State</b> RegNo, Name, Age, Weight, Height, Mark  <b>Behavior</b> Register, Change mark, Change Height Weight	<b>Radio</b> 	<b>State</b> On_Off, Current Volume, Current Station  <b>Behavior</b> Turn on_off, Increase volume, Decrease volume, seek, scan, and tune	<b>Dog</b> 	<b>State</b> Name, Color, Breed, Hungry  <b>Behavior</b> Barking, Fetching, Wagging tail
<b>Clock</b>	<b>State</b> Dial Color, Hour, Minute  <b>Behavior</b> Set Time, Show time	<b>Car</b> 	<b>State</b> Name, Current Gear, Current Speed, Headlight Status  <b>Behavior</b> Push accelerator, Change gear, light on off	<b>Bike</b> 	<b>State</b> Speed, Acceleration, Current Gear  <b>Behavior</b> Turn the accelerator, Push the brake, Change gear
<b>Stack</b> 	<b>State</b> Top, Length, Full, Empty  <b>Behavior</b> Push, Pop	<b>Array</b> 	<b>State</b> Length, Full, Empty, Current Index  <b>Behavior</b> Insert, Delete, Sort, Traverse, Merge, Print	<b>Window</b> 	<b>State</b> Top, Left, Name, Current State  <b>Behavior</b> Minimise, Maximise, Move, Close

ചിത്രം 2.4: നിരുത്തിയിരിക്കുന്ന ഒബ്ജക്റ്റുകൾ അവയുടെ സവിശേഷതകളും (അവസ്ഥ) മെത്രേറ്റീസുകളും (സഭാവ ഗുണങ്ങൾ) കൂടി



നിങ്ങൾക്ക് ചുറുമുള്ള വസ്തുകൾ നിരീക്ഷിക്കുകയും അവ ഓരോന്നിന്റെയും സവിശേഷതകളും സഭാവസ്ഥാനങ്ങളും തിരിച്ചിറയുക. നിങ്ങളുടെ കണ്ണഭ്രതലുകൾ താഴെ പറയുന്ന പട്ടികയിൽ എഴുതുക:

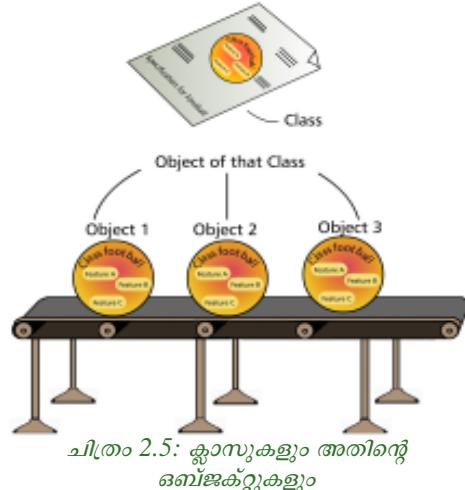
നമ്മൾ ചെയ്യാം	വസ്തുവിന്റെ പേര്	സവിശേഷതകൾ	സഭാവ സ്ഥാനങ്ങൾ

പ്രോഗ്രാമിം ഉപയോഗിച്ച് പ്രശ്ന പരിഹാരത്തിനായി OOP പ്രയോജനപ്പെടുത്തുകയാണെങ്കിൽ, പ്രശ്നത്തിനെ ഫംഗഷനുകളായി വിഭജിക്കുന്നതിനു പകരം ഒബ്ജക്ടുകളായി വിഭജിക്കുന്നതായി പരിഗണിക്കുക. ഫംഗഷനുകളായി പരിഗണിക്കാതെ ഒബ്ജക്ടുകളായി പരിഗണിക്കുന്നോൾ പ്രോഗ്രാമിന്റെ രൂപകൽപ്പന ലളിതമായി മാറുന്നു. എന്ത് കൊണ്ടെന്നാൽ പ്രോഗ്രാമുകളിലെ ഒബ്ജക്ടുകളും ധമാർമ്മ ജീവിതത്തിലെ ഒബ്ജക്ടുകളും തമ്മിൽ വളരെ സാമ്യമുണ്ട്.

ധാരായും അവയുടെ മേൽ പ്രവർത്തിക്കുന്ന ഫക്ഷനുകളും സംയോജിപ്പിച്ചു എറ്റവും അടക്കമാക്കിയാണ് OOP തെളിഞ്ഞിട്ടുള്ളത്. സംയോജനത്തിനു ശേഷം ഒബ്ജക്ടീലൂള്ള ഫക്ഷനുകളെ മെമ്പർ ഫക്ഷൻ എന്നും ധാരായെ മെമ്പർ എന്നും വിളിക്കുന്നു.

## 2.2.2 ക്ലാസുകൾ (Classes)

ഒരു ഒബ്ജക്ടീനെപ്പറ്റിയുള്ള എല്ലാം നിർണ്ണയിക്കുന്ന ക്ലാസ് ഉപയോഗിച്ചാണ് ഒരു ഒബ്ജക്ടുനെ നിർവചിക്കുന്നത്. ഒരു പ്രത്യേക തരത്തിൽപ്പെട്ട എല്ലാ വസ്തുക്കൾക്കും പൊതുവായ സ്ഥാനങ്ങളെ നിർവചിക്കുന്ന മാതൃകയാണ് ക്ലാസ്. ഈ നിർദ്ദേശങ്ങളിൽ ധാരായെ കുറിച്ചും അവയുടെ മേൽ പ്രവർത്തിക്കുന്ന ഫക്ഷനുകളെ കുറിച്ചും വിവരങ്ങൾ അടങ്കിയിരിക്കും. ഒരു ക്ലാസ്സിന്റെ ഒബ്ജക്ടീനെ ആ ക്ലാസ്സിന്റെ ഇൻസ്റ്റിൻസ് മാത്രമല്ല ഒരു ക്ലാസ്സിൽ നിന്നും എത്ര ഒബ്ജക്ട് വേണമെങ്കിലും നിർമ്മിക്കാം. (ചിത്രം 2.5 കാണുക).



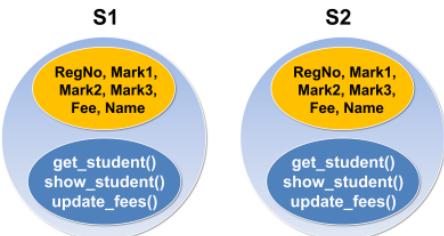


ചിത്രം 2.6: ഒരു കോഡിനു ഡിഫൈൻ ചെയ്യുവാനുള്ള പൊതുവായ രീതി ഉദാഹരണസഹിതം

ഒരു കോഡിന്റെ നിർവ്വചനവും ഉപയോഗവും സ്ട്രക്ചൂറിന് സമാനമാണ്. ഒരു സ്ട്രക്ചൂറിൽ ധാരായെ കുറിച്ചുള്ള വിവരങ്ങളാണുള്ളതെങ്കിൽ കോഡിൽ ധാരായെയും അവയുടെ മേൽ പ്രവർത്തിക്കുന്ന ഫലങ്ങളുകളെയും പറിയുള്ള വിവരങ്ങളാണ് അടങ്ങിയിരിക്കുന്നത് (ചിത്രം 2.6 കാണുക). 'struct' എന്ന കീവേർഡ് ഉപയോഗിച്ചാണ് സ്ട്രക്ചൂർ നിർമ്മിക്കുന്നതെങ്കിൽ 'class' എന്ന കീവേർഡ് ഉപയോഗിച്ചാണ് കോഡ് നിർമ്മിക്കുന്നത്.

Student എന്ന കോഡ് ഉപയോഗിച്ച് 'S1', 'S2' എന്ന് പേരുള്ള രണ്ടു ഓബ്ജക്റ്റുകൾ നിർമ്മിക്കാനുള്ള പ്രസ്താവന താഴെ കൊടുത്തിരിക്കുന്നു. (ചിത്രം 2.7 തിലുള്ളത് പോലെ).

Student S1, S2;



ചിത്രം 2.7: ദുഃഖാന്തരം കോഡിന്റെ ഓബ്ജക്റ്റുകൾ



താഴെ പറയുന്ന പ്രസ്താവന ഉപയോഗിച്ച് ഒരു കോഡിൽ നിന്നും ഓബ്ജക്റ്റുകൾ നിർമ്മിക്കാം.

**Cube S;** അല്ലെങ്കിൽ **Cube \*C;** **C = new Cube;**

ഇവിടെ Cube എന്നത് കോഡിന്റെ പേരും C എന്നത് ഓബ്ജക്റ്റിന്റെ പേരും ആണ്.

രണ്ടു ഓബ്ജക്റ്റുകൾ തമ്മിൽ ആശയവിനിമയം നടത്തുന്നത് സന്ദേശം (മെസേജ്) കൈമാറിക്കൊണ്ടാണ്. ഈ നിത്യ ജീവിതത്തിൽ ആളുകൾ പരസ് പരം സന്ദേശങ്ങൾ കൈമാറുന്നതിന് സമാനമാണ്. ധമാർത്ഥ ജീവിതത്തെ അനുകരിക്കുന്ന തരത്തിലുള്ള സംവിധാനം നിർമ്മിക്കുവാൻ ഈ സഹായിക്കുന്നു. ഒരു ഓബ്ജക്റ്റിന്റെ മെമ്പർ ഫലങ്ങൾ മറ്റാരു ഓബ്ജക്റ്റ് വിളിക്കുന്നതിനെ മെസേജ് പാസിംഗ് എന്ന് പറയുന്നു. മെസേജ് പാസിംഗ് പ്രക്രിയയിൽ ഓബ്ജക്റ്റിന്റെ പേര്, മെമ്പർ ഫലങ്ങൾ പേര്, അയയ്ക്കുവാനുള്ള വിവരം എന്നിവ സൂചിപ്പിക്കുന്നു.

ചീച്ചർ ഓബ്ജക്റ്റ്



ദുഃഖാന്തരം ഓബ്ജക്റ്റ്



ചിത്രം 2.8: ഓബ്ജക്റ്റുകൾ സന്ദേശം കൈമാറുന്നു

ഉദാഹരണത്തിന് നമ്മുടെ സ്കൂൾ സോഫ്റ്റ്‌വെയറിൽ Teacher ഒബ്ജക്ടിനു ഒരു വിദ്യാർമ്മിയുടെ ഫൈല് പരിഷ്കരിക്കണമെങ്കിൽ Student ഒബ്ജക്ടിലേക്കു ഡാറ്റ ആവശ്യപ്പെട്ടു കൊണ്ട് ഒരു സന്ദേശം അയക്കുവാൻ update\_fee() എന്ന മെമ്പർ ഫല്ലിംഗ് വിളിക്കുന്നു (ചിത്രം 2.8 കാണുക).

```
S1.update_fees("Rahul", 1000);
```

ഈവിടെ S1 എന്നത് ട്രാധിനിയിൽ ഒബ്ജക്ട് ആണ്. സന്ദേശം ലഭിക്കുന്ന മുറയ്ക്ക് Student ഒബ്ജക്ടിനു Teacher ഒബ്ജക്ട് നൽകുന്ന ഡാറ്റ ഉപയോഗിച്ച് Student റിംഗ് ഫൈല് പരിഷ്കരിക്കുന്നു.

### നിഞ്ഞുടെ പുരോഗതി അറിയുക



- OOP എന്നാൽ \_\_\_\_\_.
- OOP തെ ഒരു ഒബ്ജക്ടിനുള്ള ബൂ പ്രിൻ്റിനെ \_\_\_\_\_.
- കൗൺസിലി ബന്ധപ്പെട്ട മംഗൾഷനുകളെ \_\_\_\_\_ എന്ന് വിളിക്കുന്നു.
- കൗൺസിലി അക്കദാക്കു നിർവ്വചിക്കുന്ന വേദിയബിള്ളുകളെ \_\_\_\_\_ എന്ന് പറയുന്നു.
- കൗൺസിലം സ്ടേറ്റും തമിലുള്ള വ്യത്യാസം എന്ത്?

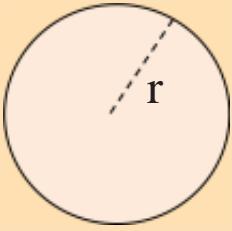


താഴെ പറയുന്ന ഫ്രോഗ്രാം വ്യത്തെത്ത ഒബ്ജക്റ്റീവി ഉപയോഗിക്കുന്നു 'Circle' എന്ന ക്ലാസ്സ് ആവശ്യമായ ഏക ഡാറ്റയായ **radius** പ്രസ്താവിക്കുകയും **radius** നെ സൈക്രിക്കാനും വിസ്തീർണ്ണം പ്രദർശിപ്പിക്കാനുമുള്ള മെമ്പർ ഫല്ലിംഗ്പൗകളെ പ്രസ്താവിക്കുകയും ചെയ്യുന്നു.

```
using namespace std;
#include<iostream>
class Circle
{
private:
    float r; } Member (മെമ്പർ)
public:
void get_radius()
{
    cout << "Enter Radius : ";
    cin >> r;
}
void display_area()
{
    cout<< "Area:"<< 3.14*r*r;
}
};

int main()
{
    Circle C1; } Class (ക്ലാസ്)
    C1.get_radius(); //സേരും അയയ്ക്കുന്നു
    C1.display_area(); //സേരും അയയ്ക്കുന്നു
}

Output :
Enter Radius : 2.0
Area: 12.56
```



Member  
functions  
(മെമ്പർ  
ഫല്ലിംഗ്പൗ)

Class Declaration  
(ക്ലാസ് നിർവ്വചനം)

മെയിൻ ഫ്രോഗ്രാം  
(Main program)

### 2.2.3 ഡാറ്റ അബസ്ട്രക്ഷൻ (Data Abstraction)

ആവശ്യമായ വിവരങ്ങൾ മാത്രം പുറം ലോകത്തിനു വെളിപ്പെടുത്തിക്കൊണ്ട്, ബാക്കി വിവരങ്ങൾ പിന്നണിയിൽ മറച്ചു വക്കുകയും ചെയ്യുന്നതിനെയാണ് ഡാറ്റ അബസ്ട്രക്ഷൻ എന്നത് കൊണ്ട് ഉദ്ദേശിക്കുന്നത്.

നിത്യ ജീവിതത്തിലെ ഒരു ഉദാഹരണം ടിവിയുടെ പ്രവർത്തന രീതി പരിഗണിക്കുക. സിച്ച് ഓൺ ചെയ്യുക പവർ ഓഫ് ചെയ്യുക, ചാനൽ മാറ്റുക, ശബ്ദം നിയന്ത്രിക്കുക, VCR സ്റ്റേറ്റ്, DVD മുതലായ ഉപകരണങ്ങൾ കൂടിച്ചേരുകുക തുടങ്ങിയ കാര്യങ്ങൾ നമുക്ക് ചെയ്യാവുന്നതാണ്. എന്നിരുന്നാലും നമുക്കെതിരെ ആന്തരിക പ്രവർത്തനങ്ങളെ കുറിച്ച് അറിവില്ല. ടിവിയിലേക്ക് എങ്ങനെ സിഗ്നലുകൾ ലഭിക്കുന്നു, അതെങ്ങനെ പരിവർത്തനം ചെയ്യപ്പെടുന്നു, എങ്ങനെയത് സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നു എന്നിവയെക്കുറിച്ചു നമുക്കു അറിയില്ല.

അപോൾ ടെലിവിഷൻ അതിരെ ആന്തരികമായ പ്രവർത്തനതലത്തെയും പാഹ്യമായ സമർക്കമുഖ്യത്തെയും വേർത്തിരിക്കുന്നതായി ഇതിൽ നിന്നും നമുക്ക് മനസ്സിലാക്കാം. ആന്തരിക പ്രവർത്തനത്തെക്കുറിച്ചുള്ള പരിജ്ഞാനം കൂടാതെ തന്നെ പവർ ബട്ടൺ, ചാനൽ മാറ്റുവാനുള്ള ബട്ടൺ, ശബ്ദം നിയന്ത്രിക്കാനുള്ള ബട്ടൺ മുതലായ നമുക്ക് ഉപയോഗിക്കാവുന്നതാണ്.

ഈ പോലെ C++ ക്ലാസുകളും ഡാറ്റ അബസ്ട്രക്ഷൻ പ്രദാനം ചെയ്യുന്നു. ഒരു ഒബ്ജക്ട് റിഫ്രെൻസ് പ്രവർത്തനത്തെ ഉപയോഗപ്പെടുത്തുവാനും അതിലെ ഡാറ്റ ലഭ്യമാക്കാനും പുറം ലോകത്തിനു ഉപയോഗിക്കാവുന്ന തരത്തിൽ പൊതുവായ മെത്തെയുകൾ അവ പ്രദാനം ചെയ്യുന്നു. ക്ലാസിലെ ആന്തരിക ഘടനയെ കുറിച്ച് അറിയാതെ തന്നെ ഈ മെത്തെയുകൾ ഉപയോഗപ്പെടുത്തി ഒബ്ജക്ടുകളെ കൈകാര്യം ചെയ്യാവുന്നതാണ്.

ഉദാഹരണമായി, വിദ്യാർമ്മികളുടെ വിശദാംശങ്ങൾ പ്രദർശിപ്പിക്കുന്നതിനായി Student ഒബ്ജക്ട് റീഞ്ചറുകൂടി show\_student() എന്ന മെമ്പർ ഫല്ലിംഗ് വിളിക്കാവുന്നതാണ്. എത്ര സമയത്തു വേണമെങ്കിലും നമുക്ക് show\_student() റെഞ്ച് പ്രവർത്തന രീതി മാറ്റാവുന്നതാണ്. എന്നിരുന്നാലും ഫല്ലിംഗ് ഫല്ലിംഗ് പേരിന് മാറ്റം വരാത്തിട്ടേണ്ടും കാലം ഫല്ലിംഗ് വിളിക്കുന്നതിന് ഒരു മാറ്റവും വരുന്നില്ല.

ഡാറ്റ അബസ്ട്രക്ഷൻ രണ്ടു പ്രധാനപ്പെട്ട പ്രയോജനങ്ങൾ:

- ഉപയോകത തലത്തിൽ അപ്രതീക്ഷിതമായി വരാവുന്ന തെറ്റുകളിൽ നിന്നും ക്ലാസ്സിലെ ആന്തരിക ഘടനയെ സംരക്ഷിക്കുന്നു. ഈ തെറ്റുകൾ ഒബ്ജക്ട്സിലെ അവസ്ഥയെ പ്രതികുലമായി ബാധിക്കുന്നത് ഒഴിവാക്കുവാൻ സാധിക്കുന്നു.
- ആവശ്യകതയുണ്ടാക്കുന്ന ഉപയോകത തലത്തിലുള്ള കോഡിന് മാറ്റം വരുത്താതെ തന്നെ ലാറ്റിന്റെ പ്രവർത്തനത്തിനു മാറ്റം വരുത്താവുന്നതാണ്. കാലാന്തര ത്തിലെ ആവശ്യകത അനുസരിച്ച് ക്ലാസിലെ നടപ്പിലാക്കലിൽ മാറ്റം ഉണ്ടാക്കുവാൻ, ക്ലാസിനുള്ളിലെ പ്രവർത്തന നിർദ്ദേശങ്ങൾ മാറ്റം വരുത്തേണ്ടതായി വരുന്നീല്ല.

## 2.2.4 ഡാറ്റ എൻക്യൂപ്പസുലേഷൻ (Data Encapsulation)

ഹംഗ്രജനുകൾ, ഡാറ്റ എന്നീ രണ്ടു അടിസ്ഥാന ഘടകങ്ങൾ അടങ്കിയതാണ് C++ ലെ എല്ലാ പ്രോഗ്രാമുകളും. ഡാറ്റയെയും അവയുടെ ഫേഴ്ബൽ പ്രവർത്തിക്കുന്ന ഹംഗ്രജനുകളെയും പരസ്പരം ബന്ധിപ്പിക്കുകയും അവയെ ബാധ്യമായ ഇടപെടലുകളിൽ നിന്നും ഭൗതികമായി നിന്നും സംരക്ഷിക്കുകയും ചെയ്യുന്ന OOP തത്വമാണ് ഡാറ്റ എൻക്യൂപ്പസുലേഷൻ.

കൂൺ എന്ന് പറയുന്ന ഉപയോകത നിർവ്വചിത്മായ ഡാറ്റ ടെസ്റ്റ് ഉപയോഗിച്ചാണ് C++ ഡാറ്റ എൻക്യൂപ്പസുലേഷൻ എന്നതു പ്രാവർത്തികമാക്കുന്നത്. `private`, `protected`, `public` എന്നീ മുന്നുതരം അംഗങ്ങൾ ഒരു ക്ലാസ്സിൽ ഉണ്ടാകാം(ചിത്രം 2.6 കാണുക). ക്ലാസ്സിൽ നിർവ്വചിച്ചിരിക്കുന്ന എല്ലാ അംഗങ്ങളും സ്വാഭാവികമായി `private` ആയിരിക്കും. `private` ആയി നിർവ്വചിക്കപ്പെട്ട അംഗങ്ങൾ ക്ലാസ്സിനു പുറത്തു ദൃശ്യമാകുകയില്ല. `protected` ആയി നിർവ്വചിക്കപ്പെട്ട അംഗങ്ങൾ വേം ക്ലാസ്സിന് പുറമെ ഡിറൈവ്‌സ് ക്ലാസ്സിലും ഉപയോഗിക്കാം (ഭാഗം 2.2.6 ലെ വിശദീകരിച്ചിരിക്കുന്നു). പക്ഷേ അവ ക്ലാസിനു പുറത്തു ദൃശ്യമാകുകയില്ല.

`Student` ക്ലാസിലെ അംഗങ്ങളായ `Regno`, `Name`, `Mark1`, `Mark2`, `Mark3`, `Fee` എന്നിവ `private` ആയിട്ടാണ് നിർവ്വചിച്ചിരിക്കുന്നത്. അതായത് `Student` ക്ലാസ്സിലെ മെമ്പർ ഹംഗ്രജനുകൾക്കു മാത്രമേ അവയെ ഉപയോഗിക്കാൻ സാധിക്കു, പ്രോഗ്രാമിലെ മറ്റു ഭാഗങ്ങളിൽ ഈവ ലഭ്യമാകുകയില്ല. ഇവിടെ ഡാറ്റ മറയ്ക്കുക വഴി എൻക്യൂപ്പസുലേഷൻ സാധ്യമാക്കുകയും ചെയ്യുന്നു.

ക്ലാസിലെ അംഗങ്ങൾ പ്രോഗ്രാമിലെ മറ്റു ഭാഗങ്ങളിലേക്ക് ലഭ്യമാക്കണമെങ്കിൽ അവയെ `public` എന്ന കീവേർഡ് ഉപയോഗിച്ച് നിർവ്വചിക്കണം. `public` എന്ന വാക്കിനു ശേഷം നിർവ്വചിക്കപ്പെട്ടുന്ന എല്ലാ ഡാറ്റയും ഹംഗ്രജനുകളും പ്രോഗ്രാമിൽ മറ്റു ഭാഗങ്ങൾക്കു ലഭ്യമാക്കുന്നതാണ്. ഉദാഹരണത്തിന് `Student` എന്ന ക്ലാസിലെ അംഗങ്ങൾ പ്രോഗ്രാമിൽ മറ്റു ഭാഗങ്ങൾക്കു ലഭ്യമാക്കണമെങ്കിൽ `public` ആയിട്ട് അവയെ പ്രസ്താവിക്കണം.

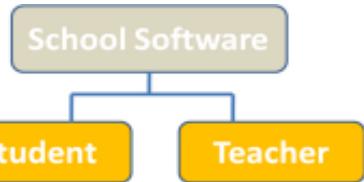
## 2.2.5 മോഡ്യൂലാരിറ്റി (Modularity)

ഒരു പ്രശ്നപരിഹാരണത്തിനായി കമ്പ്യൂട്ടർ പ്രോഗ്രാം എഴുതുമ്പോൾ അതിനെ ഉപ പ്രോഗ്രാമുകളായി വിഭജിച്ച് അവ ഓരോന്നിനും പ്രതിവിധി (പ്രലാം) കണ്ണെത്തുന്നു. സമ്പർക്കമുഖ്യമായി (interface), പ്രവർത്തനതലവും (implementation), വിവരങ്ങളും (specification) അടങ്കിയ വ്യത്യസ്ത ഘടകങ്ങളായിരിക്കും ഓരോ ഉപപ്രോഗ്രാമുകളുടെയും പരിഹാരമായി ലഭിക്കുക.

മോഡ്യൂലാരിറ്റി എന്ന ആശയം ഉപയോഗിച്ചാണ് തന്നിരിക്കുന്ന പ്രശ്നത്തെത്ത സത്രന്തമായ ഘടകങ്ങളായി വിഭജിക്കുന്നത്. ഇത്തരത്തിലെത്തു ഓരോ മൊഡ്യൂലും മറ്റു മൊഡ്യൂലുകളിൽ നിന്ന് സത്രന്തമായി എഴുതപ്പെട്ടുന്നവയായിരിക്കും. പിന്നീട് ഈ മൊഡ്യൂലുകൾ കോർത്തിണക്കി പൂർണ്ണമായ ഒരു സോഫ്റ്റ്‌വെയർ നിർമ്മിക്കുന്നു. ഇത്തരം മൊഡ്യൂലുകൾ ആശയവിനിമയം ചെയ്യുന്നത് പരസ്പരം സന്ദേശങ്ങൾ അയയ്ക്കുന്നതിലൂടെ (മെസൈജ് പാസിംഗ്) ആണ്.

ഹംഗ്രജനുകൾ ഉപയോഗിച്ച് മോഡ്യൂലാരിറ്റി പ്രാവർത്തികമാക്കുന്നത് എങ്ങനെന്നുണ്ട് എന്ന് നമ്മൾ മുന്ന് പറിച്ചിട്ടുണ്ട് (പതിനൊന്നാം തരത്തിലെ പാഠപുസ്തകത്തിൽ അധ്യായം).

10 ഫോറോന്റ് കാസ്റ്റ). മോഡ്യുലാരിറ്റി പ്രവർത്തികമാക്കിയിരിക്കുന്നത് ഒബ്ജക്ട് ഓറിയന്റൽ പ്രോഗ്രാമിങ്ങിൽ കൂടാൻ സഹായത്തോടുകൂടിയാണ്. ഉദാഹരണത്തിന് നമ്മുടെ സ്കൂൾ സോഫ്റ്റ്‌വെയർിൽ കൂടാൻ എന്ന ആശയത്തിന്റെ സഹായത്താടു വിദ്യാർത്ഥികളുമായി ബന്ധപ്പെട്ടതുമായ കാര്യങ്ങൾ പ്രത്യേകം മോഡ്യൂളുകളായി തരം തിരിച്ചിരിക്കുന്നു (ചിത്രം 2.9 കാണുക).

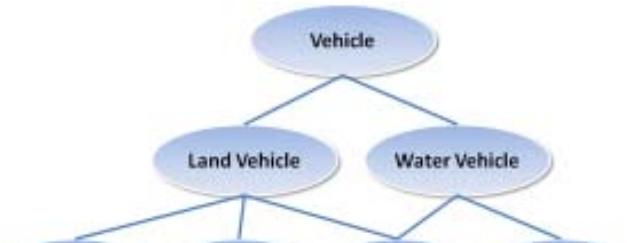


ചിത്രം 2.9: മോഡ്യുലാരിറ്റി

## 2.2.6 ഇൻഹെരിറ്റൻസ് (Inheritance)

ഒരു കൂളിലെ ഒബ്ജക്ടുകൾ മറ്റാരു കൂളിലെ സവിശേഷതകളും പ്രവർത്തനങ്ങളും ആർജിക്കുന്ന പ്രക്രിയയാണ് ഇൻഹെരിറ്റൻസ് എന്ന് പറയുന്നത്. ക്രമപ്രകാരമുള്ള തരം തിരിക്കൽ (Hierarchical Classification), പുനരുപയോഗം (Reusability) എന്നീ തത്ത്വങ്ങളെ ഇൻഹെരിറ്റൻസ് പിന്തുടരുന്നു.

നിത്യ ജീവിതത്തിലെ ഒരു ഉദാഹരണത്തിലൂടെ നമുക്കൊ സാഹചര്യം വിവരിക്കാം. ചിത്രം 2.10 ലെ ഭൂതല വാഹനത്തിനും (land vehicle) ജല വാഹനത്തിനും (water vehicle) വാഹനത്തിന്റെ (Vehicle) സവിശേഷതകൾ (അതായത് ഡാറ്റ അംഗങ്ങളും അംഗങ്ങളായ ഫോറോന്റ് കാസ്റ്റ) കൈവരുന്നു. കാറിനും ട്രക്കിനും

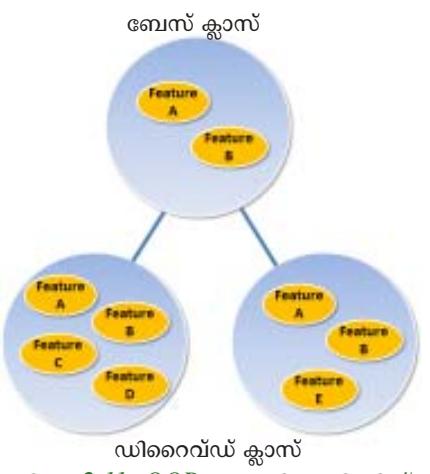


ചിത്രം 2.10: യമാർത്ഥ ജീവിതത്തിലെ ഇൻഹെരിറ്റൻസ്

ഭൂതല വാഹനത്തിന്റെ സവിശേഷതകൾ ലഭ്യമാകുന്നു (അതായത് കാർ/ട്രക്ക് = വാഹനം + ഭൂതല വാഹനം). അത് പോലെ ബോട്ടിന് ജല വാഹനത്തിന്റെ സവിശേഷതകൾ (അതായത് ബോട്ട്=വാഹനം+ജല വാഹനം) ലഭിക്കുന്നു. ജലത്തിലും ഭൂതലത്തിലും സഖ്യവിക്കുന്ന ഒരു ഹോവർ ക്രാഫ്റ്റിനു (hovercraft) ഭൂതല വാഹനത്തിന്റെയും ജല വാഹനത്തിന്റെയും സവിശേഷതകൾ (അതായത് ഹോവർ ക്രാഫ്റ്റ്=വാഹനം+ഭൂതല വാഹനം+ജല വാഹനം) ലഭിക്കുന്നു. അതുപോലെ കാറിനെ വീണ്ടും ഹാച്ച്‌ബാക്ക് (Hatchback), സെഡാൻ (Sedan), എസ് ടൂ വി (SUV) മുതലായവയായി വീണ്ടും തരം തിരിക്കാവുന്നതാണ്. അവ ഓരോനും വാഹനം, ഭൂതല വാഹനം, കാർ എന്നിവയുടെ സവിശേഷതകൾ ആർജിക്കുന്നു, അതേ സമയം തന്നെ അവയുടെതാഴീടുള്ള സവിശേഷതകൾ നിലനിർത്തപ്പെടുകയും ചെയ്യുന്നു. ഇത്തരത്തിലുള്ള ക്രമീകരണം ഏതു തലത്തിലേക്കും വ്യാപിപ്പിക്കാവുന്നതാണ്.

ഈ ഉദാഹരണത്തിൽ അമ്പവാ ഭൂതല വാഹനത്തിന്റെയും ജല വാഹനത്തിന്റെയും പൊതുവായ സവിശേഷതകളും പ്രവർത്തനങ്ങളും വേർപിരിച്ച് വാഹനം എന്ന കൂളിൽ തന്നെ ചെറിയുനില്ലെങ്കിൽ ആ രണ്ടു കൂളികളിലും അവ ആവർത്തിക്കേണ്ടി വന്നേനെ. ഇത് പ്രോഗ്രാമിങ്ങിന്റെ വലിപ്പവും കോഡ് ചെയ്യാനും തെറ്റ് കൈഞ്ഞത്തി തിരുത്താനുമുള്ള സമയവും വർധിപ്പിക്കുന്നു. തന്നീരിക്കുന്ന ചാർട്ടിന്റെ തുടക്കം മുതൽ അവസാനം വരെ നോക്കിയാൽ പ്രോഗ്രാമിന്റെ സക്ഷിഖ്യ എത്രതെന്തോളം കുറയ്ക്കുവാൻ ഇൻഹെരിറ്റൻസ് കൊണ്ട് സാധിക്കുന്നു എന്ന് നമുക്ക് കാണാം.

ക്ലാസ് നിർമ്മിക്കേണ്ടുകയും തെറ്റ് തിരുത്തപ്പെടുകയും ചെയ്തു കഴിഞ്ഞാൽ ആവശ്യമെങ്കിൽ മറ്റു പ്രോഗ്രാമുകൾക്ക് അതിനെ വിതരണം ചെയ്യാവുന്നതാണ്. ഇതിനെ പുനരുപയോഗം (റൈസൈബിലിറ്റി) എന്ന വിളിക്കുന്നു. OOP ത ഇൻഹെരിറ്റൻസ് തത്വം പുനരുപയോഗം എന്ന ആശയത്തെ ഒന്നുകൂടി വിപുലീകരിക്കുന്നു. പുനരുപയോഗത്തിലൂടെ നിലവിലുള്ള ക്ലാസിൽ മാറ്റം വരുത്താതെ തന്നെ കൂടുതൽ വിശേഷഗുണങ്ങൾ നമുക്ക് കുടി ചേർക്കാവുന്നതാണ്. നിലവിലുള്ള ക്ലാസ്സിൽ നിന്നും പുതിയ ഒരു ക്ലാസ്സിനെ ഉത്പാദിപ്പിച്ചിട്ടാണ് ഈ സാധ്യമാകുന്നത്. പുതിയ ക്ലാസിനു അതിന്റെതായ സവിശേഷതകൾ കൈവരുന്നതിനോടൊപ്പം നിലവിലുള്ള ക്ലാസ്സിന്റെ സവിശേഷതകൾ പെട്ടുകൊണ്ടിരിക്കുന്നതാണ്.



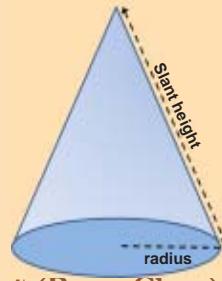
ചിത്രം 2.11: OOP ലെ ഇൻഹെരിറ്റൻസ്



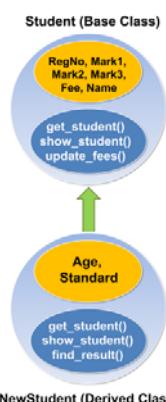
താഴെ പറയുന്ന പ്രോഗ്രാം ഇൻഹെരിറ്റൻസ് വഴി 'Cone' കൂളിലെ ഒബ്ജക്ടും നിന്നും 'Circle' കൂളിലെ ഒരു ഒബ്ജക്ടിനെ ഉൽപ്പാദിപ്പിക്കുന്നു. ആരം (r) നേരത്തെ തന്നെ 'Circle', കൂളിൽ ഡിഫൈൻ ചെയ്തിരിക്കുന്നതിനാൽ ഡിഗ്രീവിൽ കൂളിയ 'Cone' ത്രസ്സാൻ ഹെറ്റ് (s). മാത്രം ഡിഫൈൻ ചെയ്തതാൽ മതിയാകും. ഡാറ്റ സൈക്രിക്കാനും വിസ്തീർണ്ണം പ്രാശ്നിക്കാനും രണ്ടു മെമ്പർ ഫെംഷനുകൾ ഉപയോഗിച്ചിരിക്കുന്നു. 'Circle' ലെ ഡാറ്റ അംഗങ്ങൾ 'protected' ആയി ഡിഫൈൻ ചെയ്തിരിക്കുന്നതിനാൽ, 'Cone' കൂളിനു അവ ലഭ്യമാകുന്നു.

```
#include <iostream>
using namespace std;
class Circle
{ protected:
    float r;
public:
    void get_radius(){
        cout << "Enter Radius : ";
        cin >> r;
    }
    void display_area(){
        cout << "Area:" << 3.14*r*r;
    }
};
class Cone : public Circle
{ private:
    float s; } പുതിയ മെമ്പർ (New member)
public:
    void get_cone_data() { പുതിയ മെമ്പർ
        get_radius(); ഫെംഷൻ
        cout << "Enter slant height:"; ഫെംഷൻ
        cin >> s; കൾ (New
    } മെമ്പർ
    void display_cone_area(){
        cout << "Area :" << 3.14*r*(s+r);
    }
};
int main() { മെയിൻ
    Cone C1; ഫെംഷൻ
    C1.get_cone_data(); // Sending message
    C1.display_cone_area(); // Sending message (Main
} function)
```

**Output:** Enter Radius : 2.0  
Enter slant height: 5.0  
Area :43.96



സിംഗിൾ ഇൻഹെരിറ്റൻസ്, മൾട്ടിപ്പൽ ഇൻഹെരിറ്റൻസ്, ഹൈയറാർക്കീയൽ ഇൻഹെരിറ്റൻസ്, മൾട്ടി ലെവൽ ഇൻഹെരിറ്റൻസ്, ഹൈബ്രിഡ് ഇൻഹെരിറ്റൻസ് എന്നിവയാണ് വിവിധ തരത്തിലുള്ള ഇൻഹെരിറ്റൻസുകൾ (ചിത്രം 2.12 കാണുക).



ചിത്രം 2.13:  
ഇൻഹെരിറ്റൻസിൽ  
ഉപാവശ്യം

സ്കൂൾ സോഫ്റ്റ്‌വെയറിൽ ഇൻഹെരിറ്റൻസ് നടപ്പിലാക്കുന്നത് എങ്ങനെന്നാണ് നോക്കാം. നിലവിലുള്ള ധാരായും ഫ്രെംബോം കൃതാതെ Student ക്ലാസിലേക്ക് വയസ്സ്, പരിക്കുന്ന ക്ലാസ് എന്നീ ധാരായും പരീക്ഷാഫലം കണ്ണു പിടിക്കാനുള്ള ഫ്രെംബോം കൂട്ടിച്ചേര്ക്കുന്നും എന്ന് കരുതുക. Student ക്ലാസിനു മാറ്റം വരുത്താതെ NewStudent എന്ന ഒരു പുതിയ ക്ലാസിനെ നമ്മൾ Student ക്ലാസിൽ നിന്നും ഉൽപ്പാദിപ്പിക്കുന്നു. Student ക്ലാസ് അതേപടി തുടരുന്നു. ഇവിടെ Student ക്ലാസ് ബേസ് ക്ലാസും, NewStudent ക്ലാസ് ഡിരേക്ട് ക്ലാസും ആണ് (ചിത്രം 2.13 കാണുക).

ഡിരേക്ട് ക്ലാസിനെ പ്രവൃംപിക്കാനുള്ള ഘടന താഴെ പറയുന്നത് പോലെയാണ്.

```
class derived_class: AccessSpecifier
    base_class
    {
        // മെമ്പറുകളുടെയും മെമ്പർ ഫ്രെംബോകളുടെയും പ്രവൃംപനം
        (declaration of members and memberfunctions)
    };
```

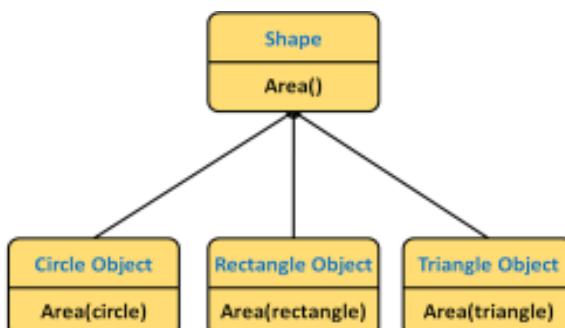
ഇവിടെ derived\_class എന്നത് ഡിരേക്ട് ക്ലാസ്സിൽ നിന്നുണ്ടായ ഡിരേക്ട് ക്ലാസ്സിൽ നിന്നുണ്ടായ ഫ്രെംബോക്കുന്നത് ആ ക്ലാസ്സിനെയുമാണ് സൂചിപ്പിക്കുന്നത്. AccessSpecifier എന്നത് public, protected, private ഇവയിൽ ഏതെങ്കിലും ആവാം. ഈത് ബേസ് ക്ലാസ്സിൽ നിന്നും പെട്ടുകൊണ്ടിരിക്കുന്നതും ലഭിച്ച അംഗങ്ങളുടെ ലഭ്യതയെ സൂചിപ്പിക്കുന്നു.

## 2.2.7 പോളിമോർഫിസം (Polymorphism)

‘പോളി’ എന്നാൽ ഒന്നിലധികം എന്നും ‘മോർഫ്’ എന്നാൽ ആകൃതികൾ എന്നുമാണ് അർഥം. വിവിധ രൂപത്തിൽ പ്രകടിപ്പിക്കാനുള്ള കഴിവിനെ പോളിമോർഫിസം എന്ന് നിർവ്വചിക്കാം. ചിത്രം 2.14 ലെ ഇത് വിശദിക്കിച്ചിരിക്കുന്നു. ഇവിടെ ‘Now Speak’ എന്ന ഒരേ നിർദ്ദേശമാണ് വിവിധ ജീവികൾക്ക് നൽകുന്നത്, പക്ഷെ അവ ഓരോനും വ്യത്യസ്ത രീതികളിലാണ് ഈ നിർദ്ദേശത്തോട് പ്രതികരിക്കുന്നത്.

ഒമ്പജക്ക് കളുടെ ഡാറ്റ ടൈപ്പ് അല്ലെങ്കിൽ ക്ലാസ് അനുസരിച്ച് അവയെ വ്യത്യസ്തമായി പ്രോസസ്സ് ചെയ്യാനുള്ള പ്രോഗ്രാമിങ്ങ് ഭാഷയുടെ കഴിവിനെയാണ് ഒമ്പജക്ക് ഓറിയൻ്റെ പ്രോഗ്രാമിങ്ങിൽ പോളിമോർഫിസം എന്ന് പറയുന്നത്. ഒന്നുകൂടി വ്യക്തമായി പറഞ്ഞാൽ ഡിരേവ്യെൻ ക്ലാസിൽ രീതിക്കെല്ല പുനർ നിർവ്വചിക്കാനുള്ള കഴിവാണ് പോളിമോർഫിസം.

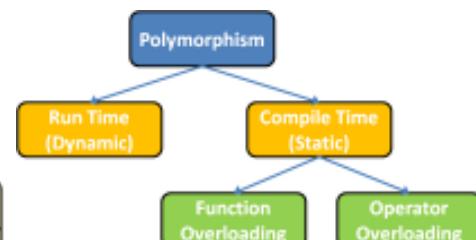
ഉദാഹരണത്തിന് Shape എന്ന ബേസ് ക്ലാസിൽ നിന്നും ഉൽപാദിപ്പിച്ചിരിക്കുന്ന circle, rectangle, triangle മുതലായ ഡിരേവ്യെൻ ക്ലാസ്സുകൾക്ക് area എന്ന പേരിൽ വിവിധ മെത്രേഡ്യുകൾ നിർവ്വചിക്കുവാനുള്ള സൗകര്യം പോളിമോർഫിസം പ്രോഗ്രാമർക്കു നൽകുന്നു (ചിത്രം 2.15 കാണുക). ഒമ്പജക്ക് ഏതായാലും area എന്ന മെത്രേഡ്യു പ്രയോഗിക്കുമ്പോൾ കൃത്യമായ ഫലം ലഭിക്കുന്നു.



ചിത്രം 2.15: പോളിമോർഫിസത്തിൽ ഉദാഹരണം



ചിത്രം 2.14: പോളിമോർഫിസത്തിൽ വർഗ്ഗീകരണം



ചിത്രം 2.16: പോളിമോർഫിസത്തിൽ വർഗ്ഗീകരണം

പോളിമോർഫിസം രണ്ടു തരത്തിലുണ്ട് (ചിത്രം 2.16 കാണുക)

- കംപയിൽ ദെം പോളിമോർഫിസം (എർലി ബൈൻഡിംഗ്/സ്റ്റാറ്റിക്) (early binding/static)
- റൺ ദെം പോളിമോർഫിസം (ലേറ്റ് ബൈൻഡിംഗ്/ഡൈനാമിക്) (late binding/dynamic)
- കംപയിൽ ദെം പോളിമോർഫിസം/എർലി ബൈൻഡിംഗ് (Compile time polymorphism)**

കംപൈൽ സമയത്തു തന്നെ ഫാർഷൻ വിളിക്കുന്നതിനെ ഫാർഷൻ നിർവ്വചനവുമായി ബന്ധിപ്പിക്കാനുള്ള കമ്പലെറിന്റെ കഴിവിനെയാണ് കംപയിൽ ദെം പോളിമോർഫിസം സൂചിപ്പിക്കുന്നത്. ഇതിന്റെ കീഴിൽ വരുന്നതാണ് ഫാർഷൻ ഓവർലോഡിംഗ്, ഓപ്രോറ്റ് ഓവർലോഡിംഗ് എന്നിവ.

**ഹണ്ഡർ ഓവർലോഡിംഗ് (Function Overloading):** ഒരേ പേരും വ്യത്യസ്തങ്ങളായ സിഗ്നേച്ചറുകളും (ഹംഗർഷൻ ആർഗ്യൂമെന്റുകളുടെ എണ്ണവും തരവും) ഉള്ള ഹണ്ഡർനുകൾ വ്യത്യസ്തങ്ങളായി പ്രവർത്തിക്കുന്നു. ഉദാഹരണത്തിൽ `area(int)` എന്നത് ഒരു സമചതുരത്തിന്റെ വിസ്തീർണ്ണ കണ്ണു പിടിക്കാൻ ഉപയോഗിക്കുന്നു, അതെ സമയം `area(int, int)` എന്നത് ഒരു ചതുരത്തിന്റെ വിസ്തീർണ്ണ കണ്ണുപിടിക്കാൻ ഉപയോഗിക്കുന്നു. ഇപ്പോൾ `area()` എന്ന ഒരേ ഹണ്ഡർ വ്യത്യസ്ത സിഗ്നേച്ചറുകൾക്കുസിച്ച് രണ്ടു രീതിയിൽ പ്രവർത്തിക്കുന്നു. ഒന്നിൽ കൂടുതൽ ഹണ്ഡർനുകൾ ഒരേ പേരും വ്യത്യസ്ത സിഗ്നേച്ചറുകളോടും കൂടി നിർവ്വചിക്കുന്നതിനെ ഹണ്ഡർ ഓവർലോഡിംഗ് എന്ന് പറയുന്നു.

**ഓപ്പറേറ്റർ ഓവർലോഡിംഗ് (Operator overloading):** നിലവിലുള്ള C++ ഓപ്പറേറ്റുകൾക്ക് (+, -, =, \* മുതലായവ) പുതിയ നിർവ്വചന നൽകുന്ന ആശയമാണ് ഓപ്പറേറ്റർ ഓവർലോഡിംഗ്. ലഭ്യമാകുന്ന ഓപറേറ്റുകൾക്ക് അനുസരിച്ച് ഒരു ക്ലാസ്സിലെ വ്യത്യസ്ത ഒബ്ജക്റ്റുകളുടെ മേൽ വ്യത്യസ്തമായി പ്രവർത്തിക്കാൻ സാധാരണ ഓപ്പറേറ്റുകളെ പ്രാപ്തരാക്കുന്ന പ്രക്രിയയാണിത്. ഒരു ഓപ്പറേറ്ററിനെ ഓവർലോഡ് ചെയ്യണമെങ്കിൽ ഓവർലോഡ് ചെയ്യുന്ന ഓപ്പറേറ്ററിനായി ഒരു മെസ്റ്റർ ഹണ്ഡർ നിർവ്വചിക്കേണ്ടതാണ്.

ഉദാഹരണത്തിന് C++ ലെ + (പ്ലസ്) എന്ന ഓപ്പറേറ്റർ നിലവിൽ തന്നെ ഓവർലോഡ് ചെയ്യപ്പെട്ടതാണ്. ഇതിന് പുർണ്ണ സംവ്യൂക്തിയും (4 + 5) അസ്ഥിര ദശാംശ സംവ്യൂക്തിയും കൂടുവാൻ (3.14 + 2.6) സാധിക്കുന്നു. ആവശ്യമെങ്കിൽ രണ്ടു ഒബ്ജക്റ്റുകൾ തമ്മിലുള്ള സങ്കലനത്തിനും ഇതേ ഓപ്പറേറ്റർ ഉപയോഗിക്കാം (അതിനുള്ള കോഡ് കൂട്ടിൽ ചേർക്കണം). ഉദാഹരണത്തിന്, `T1 = T2 + T3`. ഇവിടെ `T1`, `T2`, `T3` എന്നത് 'time' എന്ന ക്ലാസ്സിന്റെ ഒബ്ജക്റ്റുകളാണ്. '+' എന്ന ഓപ്പറേറ്റർ ഉപയോഗിച്ച് HH:MM:SS മാതൃകയിലുള്ള രണ്ടു സമയങ്ങളുടെ തുക കണ്ണുപിടിക്കാവുന്നതാണ്.

### b. റൺ ടൈം പോളിഫോർമിസം/ലേറ്റ് ബെവർജിംഗ് (Run time polymorphism)

റൺ ടൈം ഹംഗർഷൻ ഉപയോഗിക്കുന്നു സേവാൾ ഹങ്കർഷൻ നിർവ്വചനവുമായി ബന്ധപ്പെട്ടിരിക്കുന്നതിനെ റൺ ടൈം പോളിഫോർമിസം എന്ന് പറയുന്നു. ഇൻഹെരിറ്റൻസ്, പോയിന്ററുകൾ എന്നീ ആശയങ്ങളാണ് ഇതിനായി ഉപയോഗിക്കുന്നത്.

താഴെ പറയുന്ന പ്രോഗ്രാം ഫംഗ്ഷൻ ഓവർലോഡിംഗ് ഉപയോഗിച്ച് സമചതുരത്തിന്റെയും ദീർഘചതുരത്തിന്റെയും വിസ്തീരണം കണക്കുപിടിക്കുന്നു. സമചതുരത്തിന്റെ വിസ്തീരണം കണക്കുപിടിക്കാൻ വേണ്ടിയും ദീർഘചതുരത്തിന്റെ വിസ്തീരണം കണക്കെടുത്താൻ വേണ്ടിയും രണ്ടു ഫംഗ്ഷനുകൾ ഇത് നിർവ്വചിക്കുന്നു.അവ രണ്ടിനും 'area' എന്ന പിതാവായ പേര് നൽകിയിട്ടുണ്ടെങ്കിലും അവ രണ്ടിലേക്കും നൽകുന്ന വിലകൾ വ്യത്യസ്തമാണ്.

**ഫംഗ്ഷൻ പേരുകൾ  
ഒരുപോലെയാണ് (Function names  
are same)**



```
#include<iostream>
using namespace std;
int area(int s) { //സമചതുരത്തിന്റെ വിസ്തീരണം കണക്കുപിടിക്കാൻ
    return s * s; //സിസ്റ്റേമുകൾ വ്യത്യസ്തമാണ് (Signatures
} //സിസ്റ്റേമുകൾ വ്യത്യസ്തമാണ് (Signatures
int area(int s1, int s2) { //ചതുരത്തിന്റെ വിസ്തീരണം
    return (s1 * s2); //കണക്കുപിടിക്കാൻ
}
int main()
{
    cout << "Area of Square: " << area(5); << endl;
    cout << "Area of Rectangle:" << area(7, 2);
}
```

**Output:**

```
Area of Square: 25
Area of Rectangle: 14
```

### നിണ്ണളുടെ പുരോഗതി അറിയുക



1. ഡാറ്റയെയും ഫക്ഷനുകളെയും ഒരു ഫ്ലാറക്മാക്സി മാറ്റുന്നതിനെ \_\_\_\_\_ എന്ന് വിളിക്കുന്നു.
2. ഡാറ്റയെലോക്കുള്ള അനുമതി നിയന്ത്രിക്കുന്ന സവിശേഷത \_\_\_\_\_ എന്നിയപ്പെടുന്നു.
3. ഓബ്ജക്ടുകൾ സാധാരണയായി \_\_\_\_\_ ഫേബൻഡിംഗ് ഉപയോഗിക്കുന്നു.
4. C++ \_\_\_\_\_, \_\_\_\_\_ ഫേബൻഡിംഗ് ഉപയോഗിക്കുന്നു.
5. ഏർലി ഫേബൻഡിംഗിനെ \_\_\_\_\_ വിളിക്കുന്നു.
6. ലോറ്റ് ഫേബൻഡിംഗിനെ \_\_\_\_\_ വിളിക്കുന്നു.
7. വിവിധ തരത്തിലുള്ള ഇൻഹെരിറ്റൻസ് ഏതൊക്കെയാണ്?



## നമ്മക്ക് സംഗ്രഹിക്കാം

സോഫ്റ്റ്‌വെയർ കമ്പ്യൂട്ടറിനെ ഉപയോഗപ്രദമായ ഒരു ഉപകരണമാക്കി മാറ്റുന്നതിനാൽ, അതിന്റെ നിർമ്മാണത്തിനും പരിപാലനത്തിലും പ്രത്യേക പരിഗണന ആവശ്യമാണ്. മറ്റാരു തരത്തിൽ പറഞ്ഞാൽ സോഫ്റ്റ്‌വെയർ വികസനം ഫലപ്രദമാക്കാനും പരിപാലനത്തിനുള്ള ചെലവ് കുറക്കാനും ഫല തരത്തിലുള്ള മാതൃകകൾ പരീക്ഷിക്കപ്പെട്ടിട്ടുണ്ട്. സ്ടെക്ചർഡ് മാതൃക, ഫ്രോസിജറൽ ഓറിയൻ്റൽ മാതൃക, ഓബജക്റ്റ് ഓറിയൻ്റൽ മാതൃക (OOP) മുതലായവ അവയ്ക്കുഭാഗരണമാണ്. നൂതനവും പ്രചാരത്തിലുള്ളതുമായ ഫ്രോഗ്രാഫിംഗ് ഭാഷകൾ OOP മാതൃക പിന്തുടരുന്നു. പരസ്പരം ആശയവിനിമയം നടത്തുന്ന വസ്തുകളുടെ (ഓബജക്റ്റുകളുടെ) സഹായത്തോടെയാണ് OOP പ്രാവർത്തികമാക്കിയിരിക്കുന്നത്. ഇവിടെ ഡാറ്റക്സ് കൂടുതൽ പ്രാധാന്യം നൽകിയിരിക്കുന്നു. ആധികാരികമല്ലാത്ത ഡാറ്റയുടെ ഉപയോഗത്തിൽ നിന്നും സംരക്ഷണം ലഭിക്കാൻ വിവിധ ആക്സസ് സ്വീപസിഫയറുകൾ ഫംഗ്ഷനുകളുടെ കുടെ ഉൾപ്പെടുത്തുകയും ചെയ്യുന്നു. പുനരുപയോഗം മെച്ചപ്പെടുത്തുന്നതിനോടൊപ്പം കോഡ് വിപുലീകരണത്തിനും ഫലപ്രദമായ ഘടന ക്രമീകരണത്തിനും സൗംഖ്യം ദിവസിക്ക് ദിവസിക്ക് പോളിമോർഫിസം പ്രാവർത്തികമാക്കാനും OOP സഹായിക്കുന്നു.

### നമ്മക്ക് വിലയിരുത്താം

1. അധികാരിക്കത ഇല്ലാത്ത ഫക്ഷനുകളിൽ നിന്നും ഡാറ്റയെ സംരക്ഷിക്കുന്നതാണ് \_\_\_\_\_.
  - a. പോളിമോർഫിസം
  - b. എൻക്യാപ്സ്‌ലേഷൻ
  - c. ഡാറ്റ അബ്സ്ട്രക്ഷൻ
  - d. ഇൻഹെറിറ്റൻസ്
2. ബേസ് ക്ലാസിനെ \_\_\_\_\_ എന്നും വിളിക്കുന്നു.
  - a. ചെത്തേഡ് ക്ലാസ്സ്
  - b. സബ് ക്ലാസ്സ്
  - c. ഡിസ്രീബ് ക്ലാസ്
  - d. പേരൻ്റ് ക്ലാസ്
3. താഴെ പറയുന്നവയിൽ ഇൻഹെറിറ്റൻസ് അല്ലാത്തത് ഏത്?
  - a. ഫൈബ്രിയ്
  - b. മൾട്ടിപ്ലിഷ്
  - c. മൾട്ടിലൈവൽ
  - d. മൾട്ടിക്ലാസ്സ്
4. സബ് ക്ലാസിനു തുല്യമാണ്
  - a. ഡിസ്രീബ് ക്ലാസ്
  - b. സൂപ്ര ക്ലാസ്
  - c. ബേസ് ക്ലാസ്
  - d. ഇതൊന്നും അല്ല
5. സംഭാവികമായിട്ടുള്ള ആക്സസ് സ്വീപസിഫിയർ ആണ്
  - a. പബ്ലിക്
  - b. ഫൈവർ
  - c. പ്രോട്ടക്റ്ററ്റ്
  - d. ഇവ ഒന്നുമല്ല
6. താഴെ പറയുന്നവയിൽ OOP ആശയം അല്ലാത്തതെന്ത്
  - a. ഓവർലോഡിംഗ്
  - b. ഫ്രോസിജറൽ ഓറിയൻ്റൽ ഫ്രോഗ്രാഫിംഗ്
  - c. ഡാറ്റ അബ്സ്ട്രക്ഷൻ
  - d. ഇൻഹെറിറ്റൻസ്

7. ഒരു ഡാറ്റക്സ് അല്ലെങ്കിൽ മെസൈജിന് ഒന്നിലധികം രൂപത്തിൽ പ്രോസസ്സ് ചെയ്യപ്പെടാനുള്ള കഴിവാണ്.
  - a. പോളിമോർഫിസം
  - b. എൻക്യാപ്സ്‌ലേഷൻ
  - c. ഡാറ്റ ഫോറിംസ്
  - d. ഇൻഫോറിറ്റിംസ്
8. C++ ഒരു \_\_\_\_\_ ഭാഷയാണ്.
  - a. ഒബ്ജക്ട്
  - b. നോൺ പ്രോസിജറൽ
  - c. ഒബ്ജക്റ്റ് ഓറിയൻ്റെയ്
  - d. പ്രോസിജറൽ
9. താഴെ പറയുന്നവയിൽ OOP ഏഴ് പ്രത്യേകത അല്ലാത്തതെത്?
  - a. ഡാറ്റയെക്കാൾ നടപടികൾക്ക് പ്രാധാന്യം നൽകുന്നു
  - b. യഥാർത്ഥ ലോകത്തിന്റെ മാതൃകയിലുള്ളത്
  - c. ഡാറ്റയെയും ബന്ധപ്പെട്ട ഫക്ഷനുകളെയും ഒറ്റ ഘടകമായി കൂട്ടിയിണക്കുന്നു.
  - d. ഇവ ഒന്നുമല്ല
10. താഴെ പറയുന്നവയിൽ OOP നെ കുറിച്ച് ശരിയായതെത്?
  - a. ഡാറ്റ അബസ്ട്രാക്ഷൻ പിന്തുണക്കുന്നു
  - b. പോളിമോർഫിസത്തെ പിന്തുണക്കുന്നു
  - c. ഘടനാപരമായ പ്രോഗ്രാമിങ്ങിനെ പിന്തുണക്കുന്നു
  - d. ഇവയെയെല്ലാം പിന്തുണക്കുന്നു
11. പ്രോഗ്രാമിങ്ങ് മാതൃക എന്നാലെന്ത്? പ്രോഗ്രാമിങ്ങ് മാതൃകകളുടെ പേരെഴുതുക.
12. പ്രോസിജറൽ പ്രോഗ്രാമിങ്ങ് സമീപനത്തിന്റെ പരിമിതികൾ എന്തെല്ലാം?
13. ഒബ്ജക്റ്റ് ഓറിയൻ്റെയ് പ്രോഗ്രാമിങ്ങ് മാതൃക എന്നാലെന്ത്? OOP ഏഴ് അടിസ്ഥാന തത്ത്വങ്ങൾ പട്ടികപ്പെടുത്തുക.
14. C++ ത്ര എങ്ങനെയാണു OOP പ്രയോഗിച്ചിരിക്കുന്നത്?
15. എൻക്യാപ്സ്‌ലേഷൻ എന്നാലെന്ത്?
16. കൂസ്സും ഒബ്ജക്റ്റും തമിലുള്ള വ്യത്യാസം കണ്ണെത്തുക.
17. ബേസ് കൂസ്, സബ് കൂസ് എന്നാലെന്ത്? ബേസ് കൂസ്സും സബ് കൂസ്സും തമിലുള്ള ബന്ധം എന്താണ്?
18. ഡാറ്റ അബസ്ട്രാക്ഷൻ എന്ന ആശയം വിവരിക്കുക. ഒരു ഉദാഹരണം നൽകുക.
19. ഇൻഫോറിറ്റിംസിനെ പറ്റി ഒരു ലാല്പു വിവരണം എഴുതുക.
20. ഒരു കാർ പ്രവർത്തിപ്പിക്കുന്നതിനായി ലൂഡിനിംഗ്, ഡ്രേം, ആക്സിലറേറ്റർ മുതലായവ നമ്മൾ ഉപയോഗിക്കുന്നു. ഇവ ഉപയോഗിക്കുന്നേണ്ടാൽ ആന്റികമായി എന്താണ് സംഭവിക്കുന്നത് എന്നു നമുക്കറിയേണ്ടതില്ല. എത്രക്കിലും OOP ആശയവുമായി നിങ്ങൾക്കിതിനെ ബന്ധപ്പെടുത്താമോ? വിശദീകരിക്കുക.
21. ഇൻഫോറിറ്റിംസ് എന്നാലെന്ത്? പുനരുപയോഗത്തിന് ഇത് എങ്ങനെ സഹായകമാകുന്നു?

22. പോളിമോർഫിസം എന്നാലെന്ത്? ഈത് വിവരിക്കുവാൻ ഒരു ഉദാഹരണം നൽകുക.
23. OOP എന്ന ആശയം ഉദാഹരണ സഹിതം വിവരിക്കുക.
24. സ്വിച്ചേറ്റ് കൂടിയ ഒരു പ്ലാറ്റ്‌ഫോർമ് പരിഗണിക്കുക. സ്വിച്ചീറ്റ് പ്രവർത്തനം എങ്ങനെയായിരിക്കും എന്നത്, ഏതു സാഹചര്യത്തിലാണെന്ത് ഉപയോഗിക്കുന്നത് എന്നതിന്റെയും പ്ലാറ്റ്‌ഫോർമ്മിൽ എന്നാണ് അടിസ്ഥിതിക്കുന്നത് എന്നതിന്റെയും അടിസ്ഥാനത്തിയായിരിക്കും. ഏതെങ്കിലും OOP ആശയവുമായി നിങ്ങൾക്കിടിനെ ബന്ധപ്പെടുത്താമോ? വിശദീകരിക്കുക.
25. 'Horse', 'Fish', 'Bird' എന്നീ ഉപക്ഷാസുകളെ ഉൽപ്പാദിപ്പിച്ചിരിക്കുന്ന 'LivingBeings' എന്നൊരു ബേസ് ക്ലാസ് പരിഗണിക്കുക. പരാമർശിച്ചിരിക്കുന്ന എല്ലാ ക്ലാസുകളും ഇൻഹെരിറ്റ് ചെയ്യുന്ന 'Move' എന്നൊരു ഫംഗ്‌ഷൻ LivingBeings ക്ലാസിലുണ്ടെന്ന് അനുമാനിക്കുക. Horse ക്ലാസിന്റെ ഒബ്ജക്ട് 'Move' ഫംഗ്‌ഷൻ വിളിക്കുമ്പോൾ, കുതിക്കുന്നു എന്നായിരിക്കാം സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നത്. അതേസമയം Fish ക്ലാസിന്റെ ഒബ്ജക്ട് അതേ ഫംഗ്‌ഷൻ വിളിക്കുമ്പോൾ നീന്തുന്നു എന്നായിരിക്കാം സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുക. Bird എൻ്റെ ഒബ്ജക്ടിന്റെ കാര്യത്തിൽ പറക്കുന്നു എന്നായിരിക്കാം. ഏതെങ്കിലും OOP ആശയവുമായി നിങ്ങൾക്കിടിനെ ബന്ധപ്പെടുത്താമോ? വിശദീകരിക്കുക.



## പഠന നേട്വേർ

ഈ അധ്യായത്തിൽ പുർത്തീകരണത്തിന് ശേഷം പഠിച്ചാലും

- ധാര സ്റ്റെക്ക്‌ചറുകളുടെ ആശയം ഉദാഹരണ സഹിതം വിവരിക്കുവാൻ സാധിക്കുന്നു.
- ധാര സ്റ്റെക്ക്‌ചറുകളെ വിവിധ മാനദണ്ഡങ്ങൾക്കുസമീച്ചി തരംതിരിക്കാൻ സാധിക്കുന്നു.
- ധാര സ്റ്റെക്ക്‌ചറുകളിലുള്ള വിവിധ പ്രവർത്തനങ്ങൾ ഏതൊക്കെ എന്നും മനസ്സിലാക്കാനും അവയെക്കുറിച്ച് വിശദിക്കിക്കുവാനും സാധിക്കുന്നു.
- ട്രാക്ട് ധാര സ്റ്റെക്ക്‌ചറിന്റെ ഘടന ഉദാഹരണസഹിതം വിശദിക്കിക്കാൻ സാധിക്കുന്നു.
- ട്രാക്ട് പൂർണ്ണ, പോസ്റ്റ് പ്രവർത്തനങ്ങൾക്കുള്ള അൽഗോരിതം വികസിപ്പിക്കാൻ സാധിക്കുന്നു.
- കൂടും സ്റ്റെക്ക്‌ചറിന്റെ ഘടന ഉദാഹരണ സഹിതം വിശദിക്കിക്കാൻ സാധിക്കുന്നു.
- രേഖാചിത്രങ്ങൾക്കുള്ള പ്രവർത്തനങ്ങൾ അൽഗോരിതം വികസിപ്പിക്കാനും സാധിക്കുന്നു.
- വൃത്താകൃതിയിലുള്ള കൂടുവിനു രേഖാചിത്രങ്ങൾക്കുള്ള അൽഗോരിതം വികസിപ്പിക്കാൻ സാധിക്കുന്നു.
- ലിങ്ക്‌ബിൽ ലിംഗ്ഗ് ധാര സ്റ്റെക്ക്‌ചറുകളുടെ ആശയം വിശദിക്കിക്കാനും അഭേദകളും ഒരു മറ്റൊരു ധാര സ്റ്റെക്ക്‌ചറുകളും മേൽ അവക്കുള്ള നേട്വേർ വിവരിക്കാനും സാധിക്കുന്നു.

കമ്പ്യൂട്ടറുകൾ ഉപയോഗിച്ച് പ്രശ്നം പരിഹരിക്കുന്നോൾ മിക്ക സന്ദർഭങ്ങളിലും ധാര പ്രോസസ് ചെയ്യുണ്ടി വരാം. ഈ ധാരയുടെ തരം അടിസ്ഥാനപരമായതേയും സംഗ്രഹമായതേയും (ഒരു കുട്ടം) ആകാം. ഇത്തരം ധാരയെ പരാമർശിക്കുന്നതിനു വേണ്ടി വേരിയവിളുകൾ ആവശ്യമാണെന്ന് നമ്മുക്കേറിയാശുന്നതാണ്. C, C++, ജാവ മുതലായ ഭാഷകളിലെ പ്രോഗ്രാമിൽ വേരിയവിളുകൾ ഉപയോഗിക്കുന്നതിനു മുമ്പ് അവയുടെ പ്രവൃത്താവനം വേണമെന്നു ഉള്ളിപ്പിരുന്നു. അടിസ്ഥാനപരമായ ധാരകൾക്ക് വേണ്ടിയുള്ള വേരിയവിളുകൾ int, char, float, double മുതലായ അടിസ്ഥാന ധാരതരങ്ങളോ അവയുടെ ഒപ്പ് മോഡിഫേറുകളോ ഉപയോഗിച്ചാണ് പ്രവൃത്തിക്കുന്നത് എന്നു നമ്മൾ പറിച്ചതാണ്. കൂടുമായിട്ടുള്ള ധാരയെ സുചിപ്പിക്കാൻ അഭേദകളും സ്റ്റെക്ക്‌ചറുകളും ഉപയോഗിക്കുന്നു എന്നും നാം കണ്ടതാണ്. ഒരേ തരത്തിലുള്ള ധാരയുടെ ശേഖരത്തെ അഭേദ എന്നും വ്യത്യസ്ത തരത്തിലുള്ള ധാരയുടെ ശേഖരത്തെ സ്റ്റെക്ക്‌ചർ എന്നും പറയുന്നു.

വിവിധ തത്ത്വങ്ങളുടെയും മാനദണ്ഡങ്ങളുടെയും അടിസ്ഥാനത്തിൽ പ്രോഗ്രാമിങ് ഭാഷകളിൽ ധാരയെ വ്യത്യസ്ത തരങ്ങളായി ചിട്ടപ്പെടുത്താൻ ഉപയോഗിക്കുന്ന സംവിധാനങ്ങളെയാണ് ഈ അധ്യായത്തിൽ അവതരിപ്പിച്ചിരിക്കുന്നത്. ഒരു തരത്തിൽ ഉൾക്കൊള്ളാവുന്ന ധാരായുടെ അളവും അതിന്റെ മുകളിൽ പ്രാവർത്തികമാക്കാവുന്ന പ്രവർത്തനങ്ങളും, ധാരാ ഗണങ്ങൾ ചിട്ടപ്പെടുത്തുവാൻ ഉപയോഗിക്കുന്ന തത്ത്വങ്ങളുടെ അടിസ്ഥാനത്തിൽ വ്യത്യസ്തമായ തായിരിക്കും.

### 3.1 ഡാറ്റ സ്ടെക്ചർ

ചിത്രം 3.1 ലെ ചില ശുപ്പുകൾ കാണിച്ചിരിക്കുന്നു. ഓരോ ശുപ്പിലും ക്രമീകരണത്തിനായി പലതരത്തിലുള്ള തത്ത്വങ്ങൾ ഉപയോഗിച്ചിരിക്കുന്നു. ഇതിലെ ഓരോ ചിത്രവും ഓരോ വസ്തുകളുടെ ശേഖരമാണ്. ഓരോ ശുപ്പിലും വസ്തുകളെ ക്രമീകരിക്കാൻ ഉപയോഗിച്ചിരിക്കുന്ന തത്ത്വം അല്ലെങ്കിൽ മാത്യുക ഏതാണെന്നു പറയാമോ?



Fig. 3.1: വിവിധ തരത്തിലുള്ള ശേഖരങ്ങൾ

കളിപ്പാടങ്ങളുടെ ഒരു ശേഖരമാണ് ചിത്രം 3.1(a) ലെ കാണുന്നത്. പ്രത്യേകിച്ച് ഒരു ക്രമമോ ക്രമീകരണമോ ഇല്ലാതെ കൂട്ടിയിട്ടിരിക്കുകയാണ് ഈ കളിപ്പാടങ്ങൾ. ചിത്രം 3.1(b) ലെ കാണിച്ചിരിക്കുന്നത് ഒരു തട്ടിലെ ഒരു കൂട്ടം പാതങ്ങളാണ്. ഒന്നിന് പൂരകിൽ മറ്റാനൊധിക്കാണ് പാതങ്ങൾ വച്ചിരിക്കുന്നത്. തട്ടിൽ സുകഷിക്കാവുന്ന പാതങ്ങളുടെ എല്ലാത്തിന് ഒരു പരിമിതിയുണ്ട്. സ്ഥലമുണ്ടെങ്കിൽ തട്ടിൽ എവിടെ വേണമെങ്കിലും പുതിയ പാതം വയ്ക്കാവുന്നതും ഏതു പാതം വേണമെങ്കിലും തിരികെ എടുക്കാവുന്നതുമാണ്. സി.ഡി. പായ്ക്കറ്റിലെ ഒരു കൂട്ടം സി.ഡി.കൾ ആണ് ചിത്രം 3.1(c) ലെ കാണിച്ചിരിക്കുന്നത്. ഈ ശേഖരത്തിലും ഡിസ്കുകളുടെ എല്ലാത്തിന് ഒരു പരിധിയുണ്ട്. ഒരു പുതിയ സി.ഡി. ശേഖരത്തിന്റെ മുകളിൽ മാത്രമേ ചേർക്കാൻ സാധ്യമാകുകയുള്ളൂ. അതുപോലെ മുകളിലുള്ള സി.ഡി. യെ മാത്രമേ ശേഖരത്തിൽ നിന്നും ഒഴിവാക്കുവാനും സാധിക്കുകയുള്ളൂ. പുറകുവശത്തു കൂടി മാത്രം അടുത്ത വ്യക്തിക്ക് (അല്ലെങ്കിൽ അടുത്ത ഓട്ടോറിക്ഷക്ക്) വരിയിൽ ചേരാൻ കഴിയുന്ന തരത്തിലുള്ള ഒരു വരി ആണ് ചിത്രം 3.1(d) ലെ സുചിപ്പിക്കുന്നത്. വരിയുടെ മുൻവശത്തു കൂടി മാത്രമേ ഒരു വ്യക്തിക്ക് (അല്ലെങ്കിൽ ഓട്ടോറിക്ഷയ്ക്ക്) വരിയിൽ നിന്ന് പുറത്തേക്കു പോകുവാൻ സാധിക്കുകയുള്ളൂ. ഈ വരിയിൽ വ്യക്തികളുടെ എല്ലാത്തിന് പരിധി ഉണ്ടാക്കണമെന്നില്ല.

ചിത്രം 3.1 ലെ (b), (c), (d) എന്നീ ചിത്രങ്ങളിലെ ശേഖരങ്ങളുമായി സാമ്യമുള്ളതാണ് ഡാറ്റാസ്ട്രക്ചർകളുടെ തത്ത്വം. പതിനൊന്നാം തരത്തിൽ നമ്മൾ പരിച്ച് അരെയുമായി സാമ്യമുള്ളതാണ് ചിത്രം 3.1(b) ലെ കാണിച്ചിരിക്കുന്ന ശേഖരം. അതുകൊണ്ട് അരെ ഒരു ഡാറ്റാസ്ട്രക്ചർ ആണെന്ന് നമുക്ക് പറയാവുന്നതാണ്. ഒറ്റ ഘടകമായി പ്രോസസ് ചെയ്യാവുന്ന യുക്തിപരമായി ഒരേപോലെ ഉള്ളതും അല്ലെങ്കിൽ വ്യത്യസ്തമായതും ആയ ഡാറ്റയെ പ്രത്യേക റീതിയിൽ ക്രമീകരിക്കാനുള്ള മാർഗത്തെയാണ് കൃത്യമായി ശാസ്ത്രത്തിൽ ഡാറ്റാസ്ട്രക്ചർ എന്ന് പറയുന്നത്. ഉപയോകതാവിന് വിവിധ തരത്തിലുള്ള ഡാറ്റയെ കൂട്ടിയോജിപ്പിക്കാൻ മാത്രമല്ല അവയെ ഒറ്റ ഘടകമായി കണക്കാക്കി പ്രവർത്തിക്കാനും ഡാറ്റാസ്ട്രക്ചർ സഹായിക്കുന്നു.

### 3.1.1 ഡാറ്റാസ്ട്രക്ചർകളുടെ തരം തിരിക്കൽ (Classification of data structures)

പ്രാഥമികമായ ഡാറ്റാസ്ട്രക്ചർകളുായ അരെ, സ്ട്രക്ചർ മുതലായവ നമുക്ക് പരിചിതരാണ്.

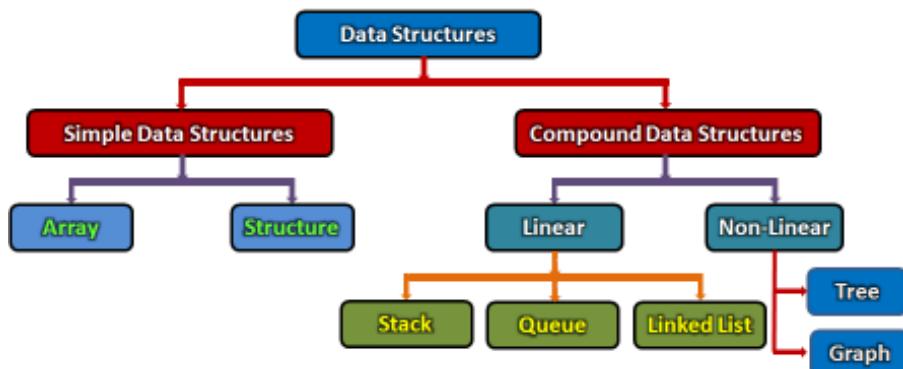


Fig. 3.2: Classification of data structures

ഡാറ്റയുടെ ശേഖരത്തെ സൂചിപ്പിക്കുന്നതിനായി C++ പ്രോഗ്രാമുകളിൽ നമ്മൾ അവ ഉപയോഗിച്ചതുമാണ്. ഇത്തരം പ്രാഥമിക ഡാറ്റാസ്ട്രക്ചർകളെ വിവിധ തരത്തിൽ കൂട്ടിയോജിപ്പിച്ച് സക്രീണമായ ഡാറ്റ സ്ട്രക്ചർകൾ രൂപപ്പെടുത്തുന്നു. ചിത്രം 3.2 ലെ കാണിച്ചിരിക്കുന്നത് പോലെ സക്രീണമായ ഡാറ്റ സ്ട്രക്ചർകളെ രേഖാചിത്രത്തും (Linear) രേഖാചിത്രമല്ലാത്തതും (Non-Linear) എന്ന് വീണ്ടും തരംതിരിച്ചിരിക്കുന്നു. ഒരു ഡാറ്റാസ്ട്രക്ചർ ചരിത്ര ഡാറ്റാ അംഗങ്ങൾ അനുവർത്തനമായി ക്രമീകരിക്കപ്പെടുകയാണെങ്കിൽ ആ ഡാറ്റാ സ്ട്രക്ചർ ചരിത്ര അംഗങ്ങളെ പ്രതിനിധിക്കാൻ തുടർച്ചയായ മെമ്മറി ലോക്കേഷൻുകൾ ഉപയോഗിക്കുന്നു. ഒരു ഡാറ്റാസ്ട്രക്ചർ ചരിത്ര അംഗങ്ങൾ മെമ്മറിയിൽ അനുവർത്തന ക്രമത്തിലെല്ലാക്കിൽ ആ ഡാറ്റാസ്ട്രക്ചർ ചരിത്ര ഒരു രേഖാചിത്രത്തെ ഡാറ്റാസ്ട്രക്ചർ എന്ന് പറയുന്നു. ഇത്തരം ഡാറ്റാസ്ട്രക്ചർകളിൽ ക്രമത്തിലെല്ലാതെയുള്ള മെമ്മറി ലോക്കേഷൻുകളിലാണ് ഡാറ്റാ അംഗങ്ങൾ സംഭരിക്കപ്പെടുക. മാത്രമല്ല അവ ഉപയോഗിക്കപ്പെടുന്നത് ക്രമത്തിലായിരിക്കണമെന്നില്ല. രേഖാചിത്രമല്ലാത്ത ഡാറ്റാ സ്ട്രക്ചർകൾ വളരെ സക്രീണമായതിനാൽ ഉപരിപരം സമയത്ത് അവയെപ്പറ്റി വിശദമായി പരിക്കുന്നതാണ്. രേഖാചിത്ര ഡാറ്റാസ്ട്രക്ചർകളായ സ്റ്റാക്ക് (Stack), ക്യൂ (Queue), ലിങ്ക്ഡ് ലിസ്റ്റ് (Linked List) എന്നിവയെപ്പറ്റി വിശദമായി ഈ പാഠാഗത്തിൽ അവതരിപ്പിച്ചിരിക്കുന്നു.

ഡാറ്റാസ്ട്രക്ചറുകൾ ഡാറ്റായുടെ ശേഖരത്തെ സുചിപ്പിക്കുന്നതിനാൽ അവയ്ക്ക് കമ്പ്യൂട്ടറിലെ മെമ്മറിയുമായി വളരെ അടുത്ത ബന്ധമാണുള്ളത്. എന്തെന്നാൽ ഡാറ്റയെ സംഭരിക്കാനുള്ള സഹായാനിം മെമ്മറി. മെമ്മറി പ്രാഥമികമോ ദിനീയമോ ആകാം. മെമ്മറി നീകിവയ്പിനുസരിച്ച് ഡാറ്റാസ്ട്രക്ചറുകളെ സ്റ്റിറ്റ് (static) ഡാറ്റാസ്ട്രക്ചറർ എന്നും അസ്റ്റിറ്റ് (dynamic) ഡാറ്റാസ്ട്രക്ചർ എന്നും രണ്ടായി തരം തിരിക്കാം. സ്റ്റിറ്റ് ഡാറ്റാ സ്ട്രക്ചറുകൾ പ്രാഥമിക മെമ്മറിയുമായി മാത്രം ബന്ധപ്പെട്ടിരിക്കുന്നു. പ്രോഗ്രാമിന്റെ പ്രവർത്തനത്തിനു മുമ്പായി ആവശ്യമായ മെമ്മറി നീകിവയ്പുകുകയും പ്രവർത്തന ഘട്ടത്തിലുടനീളം അതിനു മാറ്റംവരാതെ നിലകൊള്ളുകയും ചെയ്യുന്നു. അതായത് ഡാറ്റാസ്ട്രക്ചർ രൂപകൽപ്പന ചെയ്യുന്നോർത്തെന്ന അതിന്റെ വലുപ്പം നിർണ്ണയിക്കുന്നതുകൊണ്ട് പിന്നീട് അത് മാറ്റാൻ സാധിക്കുന്നതല്ല. അരൈകൾ ഉപയോഗിച്ച് രൂപകൽപ്പന ചെയ്തിരിക്കുന്നതും പ്രാവർത്തികമാക്കിയിരിക്കുന്നതും ആയ ഡാറ്റാ സ്ട്രക്ചറുകൾ സ്റ്റിറ്റ് സഖാവമുള്ളവയായിരിക്കും. പക്ഷേ അസ്റ്റിറ്റ് ഡാറ്റാ സ്ട്രക്ചറുകൾക്ക് പ്രവർത്തന ഘട്ടത്തിലാണ് മെമ്മറി നിർണ്ണയിക്കപ്പെടുക. ലിങ്ക് ലിസ്റ്റുകൾ ഉപയോഗിച്ച് പ്രാവർത്തികമാക്കിയിരിക്കുന്ന ഡാറ്റാസ്ട്രക്ചറുകൾ അസ്റ്റിറ്റേഷ്യായിരിക്കും. അസ്റ്റിറ്റ് ഡാറ്റാസ്ട്രക്ചറുകൾക്കിൽ ശേഖരത്തിന്റെ വലുപ്പം മുൻകൂട്ടി പ്രസ്താവിച്ചിരിക്കുകയില്ല, പകരം ഉപയോകതാവിന്റെ ഇഷ്ടാനുസരണം പ്രവർത്തന വേളയിൽ വളരുകയും ചൂരുങ്ങുകയും ചെയ്തു കൊണ്ടിരിക്കും. ഡാറ്റാ സംഭരണത്തിനായി ദിനീയ മെമ്മറി നമ്മൾ ഉപയോഗിക്കുന്നോൾ അത് ഫയലുകളുടെ രൂപത്തിലായിരിക്കും. ഡാറ്റ കുട്ടിച്ചേരകുന്നതിനുസരിച്ചു ഫയലിന്റെ വലുപ്പം കുടുകയും ഡാറ്റ നീക്കം ചെയ്യുന്നതിനുസരിച്ച് വലുപ്പം കുറയുകയും ചെയ്യുന്നു. അതിനാൽ ഫയലുകളെയും അസ്റ്റിറ്റ് ഡാറ്റാസ്ട്രക്ചറുകൾ എന്ന് പറയുന്നു.

### 3.1.2 ഡാറ്റാസ്ട്രക്ചറുകളുടെ മേഖലുള്ള പ്രവർത്തനങ്ങൾ (Operations on data structures)

ഡാറ്റാസ്ട്രക്ചറുകൾ ഉപയോഗിച്ച് പ്രതിനിധികരിച്ചിരിക്കുന്ന ഡാറ്റ ചില പ്രവർത്തന ഓഫീസ് സഹായത്തോടെയാണ് പ്രോസസ് ചെയ്യപ്പെടുന്നത്. വാസ്തവത്തിൽ ചില പ്രത്യേക പ്രവർത്തനങ്ങൾ എത്ര തവണ പ്രവർത്തിക്കണം എന്നതിനുസരിച്ചാണ് ഡാറ്റാസ്ട്രക്ചർ തിരഞ്ഞെടുക്കുന്നത്. കടന്നു പോകുക, തിരയുക, ഉൾപ്പെടുത്തുക, നീക്കം ചെയ്യുക, ക്രമപ്പെടുത്തുക, ലാഡ്പിക്കുക മുതലായവയാണ് ഡാറ്റാസ്ട്രക്ചറുകളുടെ മേൽ ചെയ്യാവുന്ന പ്രവർത്തനങ്ങൾ. ഇവയെ നമ്മൾ പരിചയപ്പെടാം.

#### a. കടന്നുപോകുക (Traversing)

ഡാറ്റാ സ്ട്രക്ചറുകൾ ഓരോ അംഗത്തെയും സന്ദർശിക്കുന്ന പ്രവർത്തനമാണ് കടന്നു പോകൽ. ആദ്യത്തെ അംഗത്തിൽ തുടങ്ങി അവസാനത്തെ അംഗം വരെ സഞ്ചാരം തുടരുന്നു. സന്ദർശിച്ച അംഗത്തെ എങ്ങനെ പ്രോസസ് ചെയ്യണമെന്നുള്ളത് പ്രശ്നത്തിന്റെ ആവശ്യകതയ്ക്കുന്നതിലും ഒരു അറേയിലുള്ള എല്ലാ അംഗങ്ങളും തിരിച്ചടക്കുന്നത് കടന്നു പോകുന്ന പ്രവർത്തനത്തിന് ഒരു ഉദാഹരണമാണ് (പതിനൊന്നാം തരത്തിലെ കമ്പ്യൂട്ടർ സയൻസ് പുസ്തകത്തിലെ അധ്യായം 8 നോക്കുക).

#### b. തിരയൽ (Searching)

ഒരു ഡാറ്റാസ്ട്രക്ചറിലെ ഒരു പ്രത്യേകഅംഗത്തിന്റെ സ്ഥാനം കണ്ടുപിടിക്കാനുള്ള പ്രക്രിയയെയാണ് തിരയൽ എന്നത് കൊണ്ടുദ്ദേശിക്കുന്നത്. ഒന്നോ അതിലധികമോ വ്യവസ്ഥകൾ നിറവേദ്യുന്ന എല്ലാ അംഗങ്ങളുടെയും സ്ഥാനം കണ്ടുപിടിക്കാനുള്ള

പ്രകിയയെയും തിരയൽ എന്നുപറയാം. മറ്റാരല്പത്തിൽ തിരയലെന്നാൽ ഡാറ്റാ സ്ട്രക്ചറിൽ സംഭരിച്ച വച്ചിരിക്കുന്ന വിലകൾ കണ്ടെത്തി ഉപയോഗിക്കുക എന്നാണ്. പതിനൊന്നാം തരത്തിൽ തിരയലിനുള്ള രണ്ടുമാർഗങ്ങൾ നമ്മൾ പരിച്ചതാണ്.

### c. ഉൾപ്പെടുത്തൽ (Inserting)

ഡാറ്റാസ്ട്രക്ചറിൽ ഒരു പ്രത്യേക സ്ഥാനത്തെക്ക് പുതിയ അംഗത്തെ ചേർക്കുന്ന പ്രകിയയാണ് ഉൾപ്പെടുത്തൽ എന്ന് പറയുന്നത്. ചില സാഹചര്യങ്ങളിൽ, പ്രത്യേകിച്ച് ഡാറ്റാസ്ട്രക്ചറിലെ ഡാറ്റാഅംഗങ്ങൾ ക്രമത്തിലാണെങ്കിൽ, പുതിയ ഡാറ്റാ ഉൾപ്പെടുത്തേണ്ട സ്ഥാനം ആദ്യം കണ്ടെത്തുകയും അതിനു ശേഷം ഡാറ്റാ ഉൾപ്പെടുത്തുകയും ചെയ്യുന്നു.

### d. നീകം ചെയ്തൽ (Deleting)

ഡാറ്റാസ്ട്രക്ചറിൽ നിന്നും ഒരു അംഗത്തെ ഒഴിവാക്കുന്ന പ്രകിയയെയാണ് നീകം ചെയ്തൽ എന്നു പറയുന്നത്. നീകം ചെയ്യുന്നതിനായി ഡാറ്റാ അംഗത്തിൻ്റെ സ്ഥാനമോ നീകം ചെയ്യേണ്ട അംഗത്തെ തന്നെയോ പരാമർശിക്കണം.

### e. ക്രമപ്പെടുത്തൽ (Sorting)

ബബിൾ സോർട്ട്, സൈലക്ഷൺ സോർട്ട് എന്നീ രണ്ടു റീതികൾ ഉപയോഗിച്ച് ഒരു അറൈയ ക്രമപ്പെടുത്തുന്നത് നമ്മൾ നേരത്തെ പതിചയപ്പെട്ടതാണ്. അംഗങ്ങളെ ഒരു പ്രത്യേക ക്രമത്തിൽ അതായത് ആരോഹണ ക്രമത്തിലോ അവരോഹണ ക്രമത്തിലോ അടുക്കി വയ്ക്കുന്ന റീതിയെയാണ് ക്രമപ്പെടുത്തൽ എന്ന് പറയുന്നത്. ഡാറ്റാസ്ട്രക്ചറിലെ ഡാറ്റാ അംഗങ്ങളെ ക്രമപ്പെടുത്തുന്നത് തിരയൽ വേഗത്തിലാക്കാൻ സഹായിക്കുന്നു.

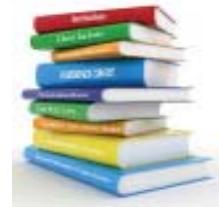
### f. ലയിപ്പിക്കൽ (Merging)

ക്രമപ്പെടുത്തിയിരിക്കുന്ന രണ്ടു ഡാറ്റാസ്ട്രക്ചററുകളുടെ അംഗങ്ങളെ കൂട്ടിയോജിപ്പിച്ച് പുതിയ ഒരു ഡാറ്റാസ്ട്രക്ചർ രൂപീകരിക്കുന്ന പ്രകിയയെ ലയിപ്പിക്കൽ എന്ന് പറയുന്നു. ലയിപ്പിക്കലിൻ്റെ ഏറ്റവും ലളിതമായ രൂപം, അംഗങ്ങളില്ലാത്ത ഒരു ഡാറ്റാസ്ട്രക്ചറിലേക്കു മറ്റു രണ്ടു ഡാറ്റാസ്ട്രക്ചററുകളെ കൂട്ടിച്ചേര്ക്കുന്നതാണ്. അരെ ഉപയോഗിച്ചു ചെയ്യുകയാണെങ്കിൽ ആദ്യം ഒരു അറൈയിലെ അംഗങ്ങൾ, അംഗങ്ങളില്ലാത്ത മുന്നാമത്തെ അരൈയിലേക്ക് പകർത്തുകയും, ശേഷം രണ്ടാമത്തെ അരൈയിലെ അംഗങ്ങൾ മുന്നാമത്തെ അരൈയിലേക്ക് കൂട്ടിച്ചേര്ക്കുകയും ചെയ്യുന്നു.

തിരയൽ, ക്രമപ്പെടുത്തൽ, ലയിപ്പിക്കൽ എന്നീ മൂന്ന് പ്രവർത്തനങ്ങളാണ് സംഭരണ ഉപകരണങ്ങളിൽ നിന്നും ഡാറ്റാ വീബ്സൈറ്റുകളും പ്രകിയ എല്ലപ്പും വേഗത്തിലും കാര്യക്ഷമവുമാക്കുന്നത്. അരെ എന്ന ഡാറ്റാസ്ട്രക്ചറിനുകൂടിച്ചും അതിനേക്ക് പ്രാവർത്തികമാക്കാവുന്ന മേല്പരിശീല പ്രവർത്തനങ്ങളെ പറിയും നമ്മൾ പതിനൊന്നാം തരത്തിൽ പറിച്ചതാണ്. മാത്രമല്ല സ്ട്രക്ചററുകളെയും അവയുടെ അംഗങ്ങളുടെ മേലുള്ള പ്രവർത്തനങ്ങളെയും കൂടിച്ചു ഇല്ല പുന്നതക്കതിന്റെ ഒന്നാം അധ്യായത്തിൽ നാം ചർച്ച ചെയ്തിരുന്നു. ഈ നമ്മൾ സക്കീറണമായ രേഖാചിത്ര ഡാറ്റാസ്ട്രക്ചററുകളായ സ്റ്റാക്സ്, ക്ലൗഡ്, ലിങ്ക് ലിസ്റ്റ് മുതലായവയെപ്പറ്റി ചർച്ച ചെയ്യാം.

### 3.2 സ്റ്റാക്ക് (Stack)

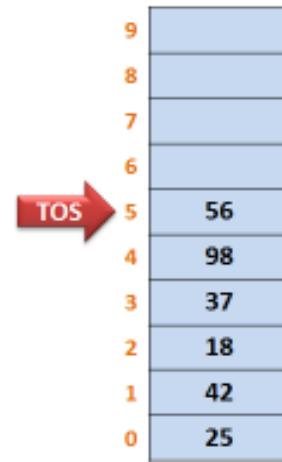
ചിത്രം 3.1(c) ഒരിക്കൽക്കുടി ശ്രദ്ധിക്കുക. താഴെ കൊടുത്തിരിക്കുന്ന ചിത്രം 3.3 ലേ കാണിച്ചിരിക്കുന്ന ശൈവരവും നോക്കുക. ഈ രണ്ടു വിഭാഗത്തിലും കൊടുത്തിരിക്കുന്ന ഇനങ്ങളുടെ ഘടന എന്നുതന്നെന്നയാണ്.



ചിത്രം 3.3: നിയുജിവിതത്തിലെ സ്റ്റാക്കിന് ഉദാഹരണങ്ങൾ

ഒരു ഇനത്തിന്റെ മുകളിൽ മറ്റാന് ചേർത്തിട്ടാണ് ഇവിടെ ശൈവരം രൂപീകരിച്ചിരിക്കുന്നത്. മറ്റാരു തരത്തിൽ പറഞ്ഞാൽ ഇനങ്ങൾ ഏറ്റവും മുകളിലാണ് ചേർക്കുന്നത്. അതുപോലെ ഏറ്റവും അവസാനം ചേർത്ത ഇനം മാത്രമേ നമ്മുക്ക് നീക്കം ചെയ്യാൻ സാധിക്കു. ഈ ക്രമീകരണ തത്ത്വത്തെ ലാറ്റ് ഇൻ എറ്റ് (LIFO) എന്ന് പറയുന്നു. LIFO തത്ത്വം പിന്തുടരുന്ന ഡാറ്റ സ്ട്രക്ചറിനെ സ്റ്റാക്ക് എന്ന് പറയുന്നു. LIFO എന്ന് പറയുന്ന അട്ടത്തു നിന്ന് മാത്രം ഇനങ്ങൾ കൂടിച്ചേർക്കുകയും നീക്കംചെയ്യുകയും ചെയ്യുന്ന ക്രമപ്പെടുത്തിയ പട്ടികയാണ് ഈത്.

സ്റ്റാക്ക് എന്നത് യുക്തിപരമായ ഒരു ആശയമാണ്. സ്റ്റാക്ക് നിർമ്മിക്കാൻ പ്രോഗ്രാമിൽ ഭാഷകളിൽ ഒരു പ്രത്യേക സംവിധാനം ഇല്ല. അരെ ഉപയോഗിച്ച് നമ്മുക്ക് സ്റ്റാക്ക് പ്രാവർത്തിക മാക്കാവുന്നതാണ്. അതരത്തിലുള്ള ഒരു സ്റ്റാക്കിന് അരെയുടെ എല്ലാ ഗുണങ്ങളും ഉണ്ടായിരിക്കുന്നതാണ്. ഇതിന്റെ വലുപ്പം മുമ്പേ നിർണ്ണയിക്കപ്പെട്ടിരിക്കുന്നതിനാൽ ഇത് സ്ഥിരമായിരിക്കുന്നതാണ്. അരെ ഉപയോഗിച്ച് പ്രാവർത്തികമാക്കിയിരിക്കുന്ന ഒരു സ്റ്റാക്ക് ആണ് ചിത്രം 3.4 ലേ കാണിച്ചിരിക്കുന്നത്. പരമാവധി 10 പുർണ്ണസംഖ്യകളെ ഇതിൽ ഉൾപ്പെടുത്താനാകും. ചിത്രം 3.4 പ്രകാരം നിലവിൽ സ്റ്റാക്കിൽ ആറു അംഗങ്ങൾ ആണുള്ളത്. അതുപോലെ അഞ്ചാം സ്ഥാനത്തുള്ള 56 എന്നതാണ് ഈ സ്റ്റാക്കിലെ അവസാനത്തെ അംഗം. അങ്ങനെ ദേഹിക്കി തുടർന്ന് TOS എന്ന മൂല്യം 5 ആയിരിക്കും. സ്റ്റാക്കിന്റെ അവസാനത്തെ അംഗത്തെ സൂചിപ്പിക്കുന്നതിനു വേണ്ടി സ്റ്റാക്കിന്റെ പേരിന് StackName[TOS] എന്ന പദപ്രയോഗം ഉപയോഗിക്കുന്നു. ഇവിടെ STACK[TOS] എന്നത് 56 ആയിരിക്കും. എന്തെന്നാൽ സ്റ്റാക്ക് നിർമ്മിക്കാൻ നമ്മൾ അരെ ആണ് ഉപയോഗിച്ചിരിക്കുന്നത്. അരെയിൽ ആദ്യത്തെ അംഗത്തെ സൂചിപ്പിക്കാൻ 0 എന്ന സൂചികയാണ് ഉപയോഗിക്കുന്നത്. ഇവിടെ STACK[0] എന്നത് 25 ആകുന്നു.



ചിത്രം 3.4:  
പുർണ്ണസംഖ്യകളുടെ സ്റ്റാക്ക്

### 3.2.1 സ്റ്റാക് പ്രാവർത്തികമാക്കൽ (Implementation of stack)

അരെ ഉപയോഗിച്ച് സ്റ്റാക് പ്രാവർത്തികമാക്കാം എന്ന് നേരത്തെ പരാമർശിച്ചിരുന്നു. അങ്ങനെന്നെയകിൽ ഒരു സ്റ്റാക് ഉപയോഗിച്ച് പ്രതിനിധികരിക്കാവുന്ന അംഗങ്ങളുടെ എല്ലംത്തിന് പരിധിയുണ്ട്. മാത്രമല്ല അത് അരെയുടെ വലുപ്പത്തിനുസരിച്ചായിരിക്കും. തുടക്കത്തിൽ സ്റ്റാക് ശുന്നമാണ് എന്നതു സുചിപ്പിക്കുന്നതിനായി, TOS എഴു വില 1 ആയി നൽകുന്നു. അരെയുടെ സുചികയുടെ ഏറ്റവും ഉയർന്ന വില എത്തുന്നത് വരെ, സ്റ്റാക്കിലേക്ക് ഓരോ അംഗവും കുടിച്ചേരക്കപ്പെടുവോൾ TOS എഴു വില ഒന്നു വച്ച് കൂടുന്നു. N അംഗങ്ങളുള്ള Top എന്ന അരെ ഉപയോഗിച്ച് സ്റ്റാക് പ്രവർത്തികമാക്കുകയാണെങ്കിൽ, TOS എഴു വില 0 മുതൽ (N-1) വരെ മാറി വരാം. അതുപോലെ സ്റ്റാക്കിലെ അംഗങ്ങളെ സുചിപ്പിക്കാൻ STACK[0], STACK[1], STACK[2], ..., STACK[N-1] മുതലായ പദ്ധതി ഉപയോഗിക്കുന്നു.

### 3.2.2 സ്റ്റാക്കിലെ പ്രവർത്തനങ്ങൾ (Operations on stack)

സ്റ്റാക് ഡാറ്റാസ്ട്രക്ചറുകൾ അരെ ഉപയോഗിച്ചാണ് പ്രാവർത്തികമാക്കിയിരിക്കുന്നതെങ്കിലും അരെക്കും ബാധകമായിട്ടുള്ള എല്ലാ പ്രവർത്തനങ്ങളും സ്റ്റാക്കിൽ അതെ രൂപത്തിൽ ഉപയോഗിക്കുന്നില്ല. ഉദാഹരണത്തിന് അരെയിൽ ഏതു സ്ഥാനത്തു വേണമെങ്കിലും ഉൾപ്പെടുത്തൽ, നീക്കം ചെയ്യൽ എന്നീ പ്രവർത്തനങ്ങൾ ചെയ്യാവുന്നതാണ്. എന്നാൽ സ്റ്റാക്കിൽ മുകളിൽ നിന്ന് മാത്രമേ ഈ പ്രവർത്തനങ്ങൾ ചെയ്യാൻ സാധിക്കും. സ്റ്റാക്കിൽ നിർവ്വഹിക്കുന്ന ഉൾപ്പെടുത്തൽ, നീക്കം ചെയ്യൽ പ്രവർത്തനങ്ങളെ യാക്കം പുഷ്ട് (push) എന്നും പോപ്പ് (pop) എന്നും പറയുന്നു. ഈ പ്രവർത്തനങ്ങൾ എങ്ങനെന്നൊരു ചെയ്യുന്നത് എന്ന് നമുക്ക് നോക്കാം.

#### a. പുഷ് പ്രവർത്തനം (Push operation)

സ്റ്റാക്കിലേക്കു ഒരു പുതിയ അംഗത്തെ ഉൾപ്പെടുത്തുന്ന പ്രക്രിയയെയാണ് പുഷ് പ്രവർത്തനം എന്ന് പറയുന്നത്. സ്റ്റാക്കിൽ നിർമ്മാണം എന്ന് പറയുന്നത് പുഷ് പ്രവർത്തനത്തിലെ ആവർത്തിച്ചുള്ള പ്രയോഗം കൊണ്ടാണ് സാധ്യമാകുന്നത്.



നമുക്ക് ചെയ്യാം

പുഷ് പ്രവർത്തനം നടപ്പിലാക്കുവോളുള്ള സ്റ്റാക്കിൽ സ്ഥിതിയാണ് ചിത്രം 3.5 ത്ത് കാണിച്ചിരിക്കുന്നത്. സ്റ്റാക് പ്രാവർത്തികമാക്കുന്ന തിനായി ഒരു അരെ നിർമ്മിച്ചതായും TOS-1 ആയും നമുക്ക് അനുമാനിക്കാം. ചിത്രം നിരീക്ഷിച്ചു പ്രവർത്തനം നിർവ്വഹിക്കാനുള്ള നടപടിക്രമങ്ങൾ എഴുതുക.



ചിത്രം 3.5: തുടർച്ചയായ പൂഷ് പ്രവർത്തനത്തിന് ശേഷമുള്ള സ്റ്റാക്കിൾറ്റ് അവസ്ഥ

ചിത്രം 3.5 തുടർച്ചയായ പൂഷ് പ്രവർത്തനത്തിന് ശേഷമുള്ള സ്റ്റാക്കിൾറ്റ് അവസ്ഥ സ്റ്റാക്കിൾറ്റ് മേലുള്ള പൂഷ് പ്രവർത്തനത്തിനു വേണ്ടി താഴെ പറയുന്ന ഘട്ടങ്ങൾ നിർവ്വചിക്കാൻ കഴിയുന്നുണ്ടോ എന്ന് പരിശോധിക്കുക.

- എട്ടം 1: സ്റ്റാക്കിലേക്കു ഉൾപ്പെടുത്തുവാനുള്ള വില ഒരു വേരിയബിളിലേക്കു സ്വീകരിക്കുക.
- എട്ടം 2: Top എ വില ഓൺ വച്ച് കൂടുക.
- എട്ടം 3: TOS എ സ്ഥാനത്തെ വില സംഭരിക്കുക.

സ്റ്റാക്കിൾറ്റ് ശുന്നമായ സമലം ഉണ്ടക്കിൽ മാത്രമേ പുതിയ അംഗങ്ങളെ ഉൾപ്പെടുത്താനുള്ള മുകളിൽ പറഞ്ഞിരിക്കുന്ന ഘട്ടങ്ങളുടെ പ്രവർത്തനങ്ങൾ സാധ്യമാക്കുകയുള്ളൂ. ചിത്രം 3.5 തൊന്തരിക്കുന്ന സ്റ്റാക്കിൽ TOS എ വില 9 ആയിരിക്കണമെന്നുള്ള പുതിയ അംഗത്തെ ഉൾപ്പെടുത്താൻ കഴിയില്ല. സ്റ്റാക്ക് നിരയുകയും പുതിയ ഒരു അംഗത്തെ ഉൾപ്പെടുത്താൻ നാം ശ്രമിക്കുകയും ചെയ്താൽ ഉടലെടുക്കുന്ന സാഹചര്യത്തെ (stack overflow) എന്ന് പറയുന്നു. സ്റ്റാക്കിലെ പൂഷ് പ്രവർത്തനത്തിനുള്ള അൽഗോറിതം നമുക്ക് നോക്കാം.

### സ്റ്റാക്കിന് മേൽ പൂഷ് പ്രവർത്തനം ചെയ്യാനുള്ള അൽഗോറിതം

N പരമാവധി വലുപ്പമുള്ള STACK[N] എന്ന അരു പരിശോധിക്കുക. സ്റ്റാക്കിൾറ്റ് മുകൾഭാഗത്തിന്റെ സ്ഥാനം സൂചിപ്പിക്കുന്നതിനായി TOS എന്ന വേരിയബിളിയും ഉണ്ട്. VAL എന്ന വേരിയബിളിൽ ധാരം സൂക്ഷിച്ചിരിക്കുന്നു. VAL-നെ സ്റ്റാക്കിലേക്ക് ചേർക്കണം. എന്ന് നിർദ്ദേശത്തിനും അവസാനിപ്പിക്കുന്ന നിർദ്ദേശത്തിനും ഇടയിലായി പൂഷ് പ്രവർത്തനത്തിനാവശ്യമായ ഘട്ടങ്ങൾ നൽകിയിരിക്കുന്നു.

തുടങ്ങുക

- 1: അമുഖം (TOS < N-1) ആണെങ്കിൽ //സ്റ്റാക്ക് ശുന്നമാണോ എന്ന് പരിശോധിക്കുന്നു
- 2: TOS = TOS + 1
- 3: STACK[TOS] = VAL
- 4: അല്ലെങ്കിൽ

5: ‘സ്ലാക്സ് ഓവർ ഹ്യോ’ എന്ന് പ്രിൻ്റ് ചെയ്യുക

6: പുറത്തേക്കു പോകുക

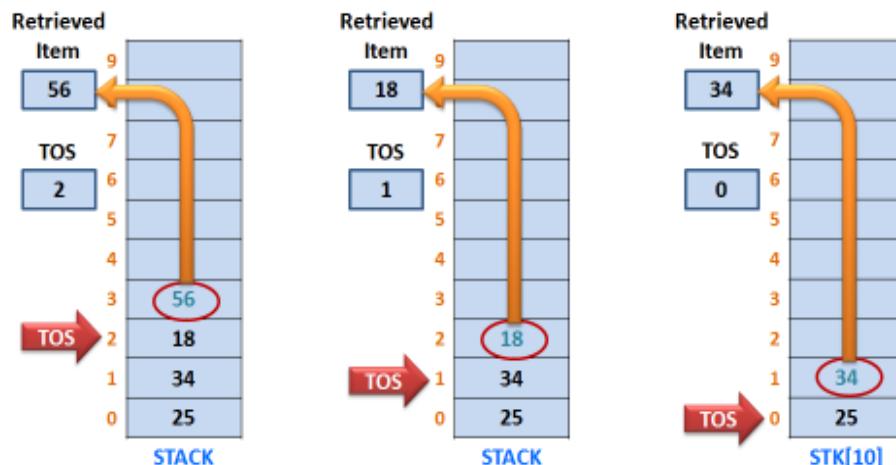
അവസാനിപ്പിക്കുക

### a. പോപ്പ് പ്രവർത്തനം (Pop operation)

ഒരു സ്ലാക്സിൽ ഏറ്റവും മുകളിലുള്ള അംഗത്തെ നീകം ചെയ്യുന്ന പ്രക്രിയയെയാണ് പോപ്പ് പ്രവർത്തനം എന്ന് പറയുന്നത്. ഓരോ പോപ്പ് പ്രവർത്തനത്തിന് ശേഷവും TOS എണ്ണ വില ഒന്ന് വച്ച് കുറയുന്നു.



ഒരു സ്ലാക്സിൽ ഏറ്റവും മുകളിലുള്ള അംഗത്തെ നീകം ചെയ്യാനുള്ള ഘട്ടങ്ങൾ നിർവ്വചിക്കാൻ നമുക്ക് ശ്രമിക്കാം. ഇതിനായി ചിത്രം 3.6 തന്നിരിക്കുന്നു. ചിത്രം നിരീക്ഷിച്ചു പോപ്പ് ഓപ്പറേഷൻ വേണ്ടിയുള്ള നമുക്ക് ചെയ്യാം ഘട്ടങ്ങൾ കണ്ടുപിടിച്ചു എഴുതുക.



അംഗത്തെ തിരിച്ചെടുത്തശേഷം

TOS 2 ആയി മാറുന്നു

അംഗത്തെ തിരിച്ചെടുത്തശേഷം

TOS 1 ആയി മാറുന്നു

അംഗത്തെ തിരിച്ചെടുത്തശേഷം

TOS 0 ആയി മാറുന്നു

ചിത്രം 3.6: തൃക്കരിച്ചയായ പോപ്പ് പ്രവർത്തനത്തിന് ശേഷമുള്ള സ്ലാക്സിൽ അവസ്ഥ

സ്ലാക്സിനു മേലുള്ള പോപ്പ് പ്രവർത്തനത്തിനുവേണ്ടി താഴെ പറയുന്ന ഘട്ടങ്ങൾ നിർമ്മിക്കാൻ കഴിയുന്നുണ്ടോ എന്ന് പരിശോധിക്കുക

ഘട്ടം 1: TOS എണ്ണ സ്ഥാനത്തുള്ള അംഗത്തിൽ വില ഒരു വേറിയവില്ലിൽ സംഭരിക്കുക.

ഘട്ടം 2: TOS എണ്ണ വില ഒന്ന് കുറയ്ക്കുക.

സ്ലാക്സിൽ അംഗങ്ങളുണ്ടെങ്കിൽ മുകളിൽ പറഞ്ഞ രണ്ടു ഘട്ടങ്ങൾ പ്രവർത്തിക്കുന്നു. അരെ ഉപയോഗിച്ചുള്ള സ്ലാക്സിൽ, പോപ്പ് പ്രവർത്തനം നടത്തുമ്പോൾ യഥാർദ്ദീതിയിൽ അംഗങ്ങളെ നീകം ചെയ്യുന്നില്ല. മറിച്ച് TOS എണ്ണ വില കുറച്ചുകൊണ്ട് അത്തരം അംഗങ്ങളെ ഉപയോഗിക്കുന്നതിൽ നിന്നും തടസ്സപ്പെടുത്തുന്നു. ഏറ്റവും അവസാനത്തെ അംഗം നീകം ചെയ്യപ്പെടുന്നതുവരെ ചിത്രം 3.6 തീ കാട്ടുത്തിരിക്കുന്ന സ്ലാക്സിൽ പോപ്പ് പ്രവർത്തനം ചെയ്യാവുന്നതാണ്. ഏറ്റവും അവസാനത്തെ അംഗം നീകം ചെയ്യപ്പെട്ടു കഴിഞ്ഞാൽ TOS

ഈ വില -1 ആയി മാറുന്നു. ഈപ്പോൾ സ്റ്റാക്ക് ശുന്നുമാണ്. ഈഅനേ ശുന്നുമായ സ്റ്റാക്കിൽ നിന്നും ഒരു അംഗത്തെ നീക്കം ചെയ്യാൻ ശ്രമിക്കുകയാണെങ്കിൽ ഉടലെടുക്കുന്ന പ്രതിസന്ധിയെ സ്റ്റാക്ക് അംഗൾ ഫ്ലോ (stack underflow) എന്ന് പറയുന്നു. ഈ നമുക്ക് സ്റ്റാക്കിന് മെല്ലുള്ള പോപ്പ് പ്രവർത്തനത്തിൽ അൽഗോറിതം എഴുതി നോക്കാം.

### സ്റ്റാക്കിന് മെല്ലുള്ള പോപ്പ് പ്രവർത്തനത്തിനുള്ള അൽഗോറിതം

പരമാവധി N അംഗങ്ങളെ സംഭരിക്കാവുന്ന ഒരു സ്റ്റാക്കിനെ പ്രാവർത്തികമാക്കാൻ വേണ്ടി STACK[N] എന്ന ഒരു അടി പരിഗണിക്കുക. സ്റ്റാക്കിലെ ഏറ്റവും മുകളിലെ സ്ഥാനത്തിൽ വിവരം സൂക്ഷിക്കുന്നതിനായി TOS എന്നൊരു വേരിയബിൾ ഉപയോഗിക്കുന്നു. സ്റ്റാക്കിൽ നിന്നും നീക്കം ചെയ്യുന്ന അംഗങ്ങളെ സൂക്ഷിക്കുന്നതിനായി VAL എന്ന വേരിയബിൾ ഉപയോഗിക്കുന്നു. പോപ്പ് പ്രവർത്തനത്തിനുള്ള നിർദ്ദേശങ്ങൾ തുടങ്ങുക, അവസാനിക്കുക എന്നിവയ്ക്കിടയിൽ കൊടുത്തിരിക്കുന്നു.

തുടങ്ങുക

- 1: അമ്മവാ (TOS > -1) ആണെങ്കിൽ //സ്റ്റാക്ക് ശുന്നുമാണോ എന്ന് പരിഗണിക്കുന്നു  
(അംഗൾ ഫ്ലോ)
- 2:       VAL = STACK[TOS]
- 3:       TOS = TOS - 1
- 4:       അല്ല എങ്കിൽ
- 5:       ‘സ്റ്റാക്ക് അംഗൾ ഫ്ലോ’ എന്ന് പ്രിൻ്റ് ചെയ്യുക
- 6:       പുറത്തേക്കു പോകുക

അവസാനിപ്പിക്കുക

 <b>സ്റ്റാക്ക് പ്രവർത്തനങ്ങൾക്ക് വേണ്ടിയുള്ള C++ ഫലങ്ങളുകൾ</b> <b>tos, n എന്നിവ ഫ്രോബെൽ വേരിയബിൾ ആണെന്ന് കരുതുക</b>	
പുഷ് പ്രവർത്തനം	പോപ്പ് പ്രവർത്തനം
<pre>void push(int stack[],int val) {     if (tos &lt; n-1)     {         tos++;         stack[tos]=val;     }     else         cout&lt;&lt;"Overflow"; }</pre>	<pre>int pop(int stack[]) {     int val;     if (tos &gt; -1)     {         val=stack[tos];         tos--;     }     else         cout&lt;&lt;"Underflow";     return val; }</pre>

TOS, N എന്നീ വേരിയബിളുകൾ ഫ്രോബെൽ വേരിയബിളുകളായി പരിഗണിച്ചിരിക്കുന്നു.



## സാക്കിന്റെ ഉപയോഗം

### 3.3 ക്യൂ (Queue)

வரிகல் நமுக்கு ஸுபரிசிதண்ணலான். பல ஸாஹசரணமிலும் நம்முள் வரிகலை எசுமாகாருமென்க. சிறும் 3.7 க்கு போஜின் ஸ்டேஷனிலே வரியான் ஸுபிபிசிரிக்கூடும். ஹவிரெட் வரியுடை முனிலுத்தே அடுபுதை வோட்டு வோட்டு செய்யும். புதிய வழக்கு வரியில் சேருவேஶ ஏற்றுவீர் புரிகிலாயிடுமான் வாயு நித்தகூக். ஹூ உபாஹரளத்தில் நினைவு வழக்கமாக்கூடும், அடுபு வரியில் நித்தகூடு வழக்கியான் வரியில் நினைவு அடுபு வருத்து போகுக். ஹூ தரத்தில் ஒரு கூடும் ஹன்னைலை க்ரமிகிரிக்கூடுமதிரை மற்று ஹன்று ஹக்கு (FIFO) தற்பொழுது என் பியத்து. அதிகாலை FIFO தற்பொழுதிலே அடிஸமாக்கத்தில் பிரவர்த்திக்கூடு யார் ஸ்டக்கினை கூடு (Queue) என் விளிக்கூடு. சிறுத்தில் கங்கது போலை ஒரு கூடுவிடு ரஸ்க் அடுபுக்கு உள்க: முன்னாலவும் (Front) பின்னாலவும் (Rear). கூடுவில் புதிய யார் அங்கைலை உச்சப்படுத்தும் பின்னாலத்தும் நிகை செய்யும் முன்னாலத்தும் அடுபிரிக்கூடு. ஸுக்கில் பிரித்து போலை கூடுவும்



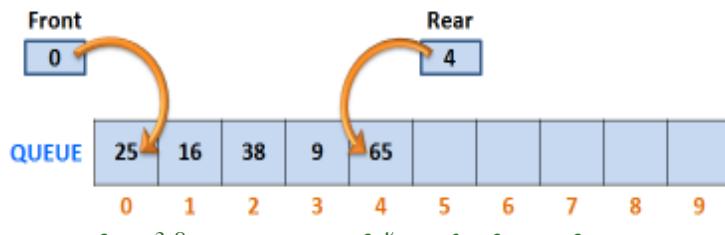
3.7: കേരളിൽ നടപ്പാക്കിയ വരൾ

### 3.3.1 ക്യൂവിനെ പ്രയോഗത്തിക്കൊള്ളൽ

## (Implementation of Queue)

எரு அரை உபயோகிப்பு கூடு பொயோகிகமாகவூக்கத்தாளைகிடல், அதில் உச்ச கொண்டிக்கொவுடன் அங்கேயெல்க்க ஏறு பறியிடுவது. Queue[10] என அரை உபயோகிப்பு

നടപ്പിലാക്കിയിരിക്കുന്ന ഒരു ക്യൂ ആണ് ചിത്രം 3.8 ത്ത് കാണുന്നത്. ഈ പരമാവധി 10 അംഗങ്ങളെ മാത്രമേ ഉൾക്കൊള്ളാനാകു. ഈ ചിത്രത്തിൽ പ്രകാരം 0 മുതൽ 4 വരെ സ്ഥാനങ്ങളിലായി അണ്വ് അംഗങ്ങളാണ് ക്യൂവിലുള്ളത്. അതായത് QUEUE 0 വും Rear 4 ലും ആയിരിക്കും. ക്യൂവിലെ ആദ്യത്തെ അംഗത്തെ QUEUE[Front] എന്നും അവസാനത്തെ അംഗത്തെ QUEUE[Rear] എന്നും സൂചിപ്പിക്കാവുന്നതാണ്.



ചിത്രം 3.8: അരെ ഉപയോഗിച്ച് പ്രവർത്തിക്കാക്കിയ ക്യൂ

അണ്ണാൻ അവസാന സൂചികയായ 9 ആണ് ക്യൂവിന് അനുവദനീയമായിട്ടുള്ള ഏറ്റവും ഉയർന്ന വില. തുടക്കത്തിൽ മുൻഭാഗത്തിന്റെയും (Rear) പിൻഭാഗത്തിന്റെയും (Front) വില -1 ആയിരിക്കും. ഈ ക്യൂ ശൂന്യമാണെന്നതിനെ സൂചിപ്പിക്കുന്നു. ക്യൂ വിലേക്കു ആദ്യത്തെ അംഗത്തെ ഉൾപ്പെടുത്തുമ്പോൾ ഈ രണ്ടു വിലകളും 0 ആയി മാറുന്നു. തുടർന്ന് ഓരോ പുതിയ അംഗത്തെ ഉൾപ്പെടുത്തുമ്പോഴും Rear റെഞ്ച് വില ഒന്ന് വച്ചു കൂടുന്നു. ഏറ്റവും ഉയർന്ന സൂചിക (ഇവിടെ 9) എത്തുന്നത് വരെ ഈത് തുടർന്ന് പോകാവുന്നതാണ്. അത് പോലെ ഓരോ അംഗത്തെ നീക്കം ചെയ്യുമ്പോഴും Front റെഞ്ച് വില ഒന്നുവച്ചു കൂടുന്നതാണ്. Front റെഞ്ച് വില Rear നെക്കാൾ കൂടുതലാക്കുന്നതു വരെ ഈത് തുടരാവുന്നതാണ്.

### 3.3.2 ക്യൂവിലെ പ്രവർത്തനങ്ങൾ (Operations on queue)

സ്കാക്കിലേതു പോലെ ക്യൂവിലും ഉൾപ്പെടുത്തൽ, നീക്കം ചെയ്തൽ പ്രവർത്തനങ്ങൾക്കു ചീല നിയന്ത്രണങ്ങളുണ്ട്. ഒരു സാധാരണ അണ്ണാൻ ഉൾപ്പെടുത്തൽ, നീക്കം ചെയ്തൽ എന്നീ പ്രവർത്തനങ്ങൾ ഏതു സ്ഥാനത്തു വേണമെങ്കിലും ചെയ്യാവുന്നതാണ്. എന്നാൽ ഒരു സ്കാക്കിൽ ഇത് ഏറ്റവും മുകളിൽ നിന്ന് മാത്രമായിരിക്കും. അതുപോലെ ക്യൂവിൽ ഉൾപ്പെടുത്തൽ, നീക്കം ചെയ്തൽ പ്രവർത്തനങ്ങൾ രണ്ട് അന്തരു മാത്രമായിരിക്കും നടക്കുക.

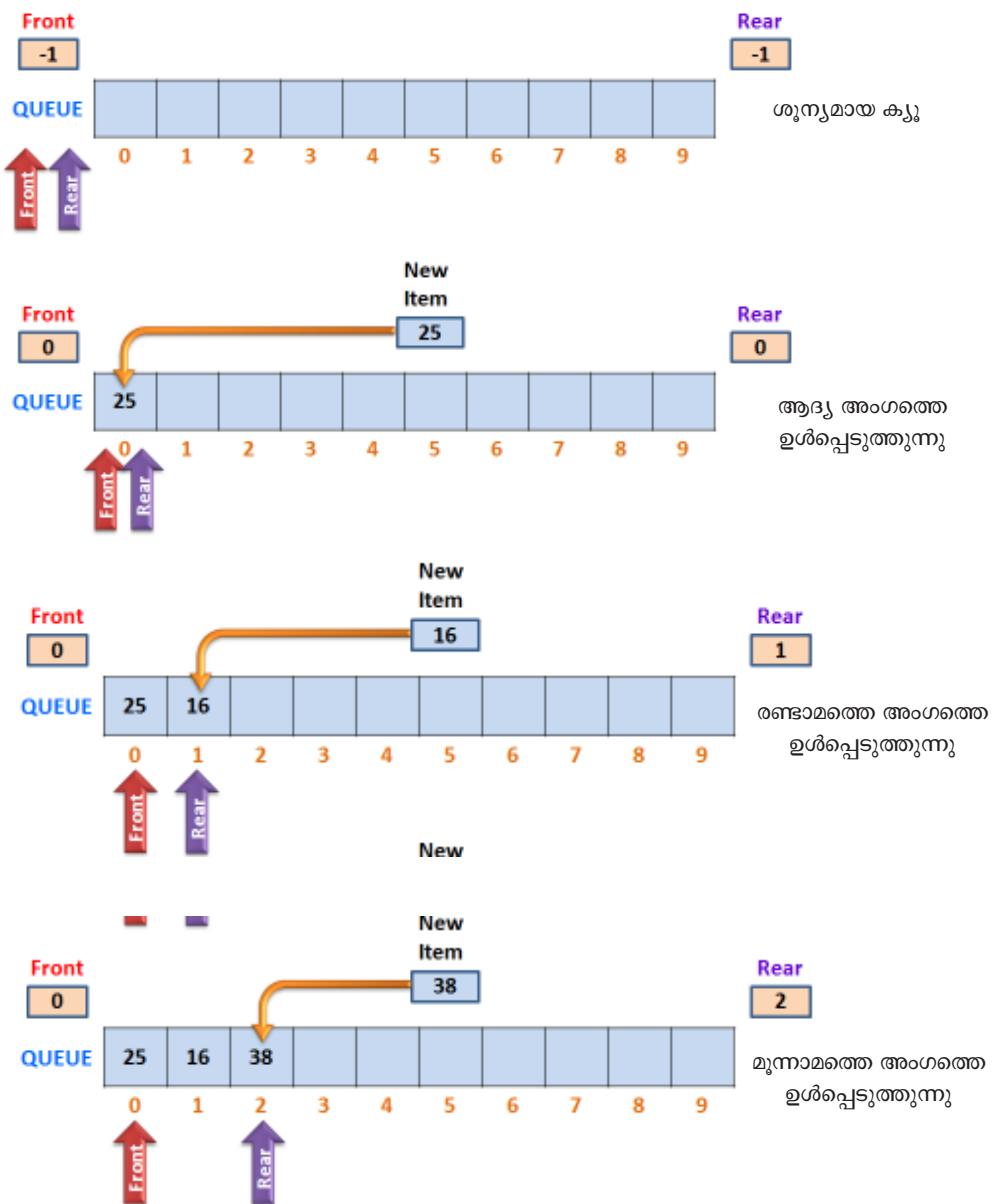
#### a. ഉൾപ്പെടുത്തൽ പ്രവർത്തനം (Insertion operation)

ക്യൂവിന്റെ പിൻഭാഗത്തു പുതിയ അംഗത്തെ കൂട്ടിച്ചേര്ക്കുന്ന പ്രക്രിയയെയാണ് ഉൾപ്പെടുത്തൽ എന്ന് പറയുന്നത്. അടുത്ത മെമ്മറി സ്ഥാനത്തെ സൂചിപ്പിക്കാൻ Rear റെഞ്ച് വില ആദ്യം കൂടുകയും പിന്നീട് ആ സ്ഥാനത്തു അംഗത്തെ ഉൾപ്പെടുത്തുകയും ചെയ്യുന്നു. ഉൾപ്പെടുത്തൽ പ്രവർത്തനം തുടർച്ചയായി ചെയ്യുമ്പോഴാണ് ഒരു ക്യൂ രൂപപ്പെടുന്നത്.



ഉൾപ്പെടുത്തൽ പ്രവർത്തനത്തിന് വിധേയമാകുന്ന ഒരു ക്യൂവിന്റെ അവസ്ഥ ചിത്രം 3.9 ത്ത് കാണിച്ചിരിക്കുന്നു.

**ഈതിനായി** ഒരു അരെ നിർമ്മിക്കുകയും, Front, Rear എന്നിവയ്ക്ക് -1 നമ്മക്ക് ചെയ്യാം എന്ന വില നൽകുകയും ചെയ്യുന്നു ചിത്രം നിരീക്ഷിച്ചു പ്രവർത്തനം ചെയ്യുവാനുള്ള നടപടിക്രമം എഴുതുക.



ചിത്രം 3.9: തുടർച്ചയായ ഉൾപ്പെടുത്തൽ പ്രവർത്തനം നടത്തിയതിന് ശേഷമുള്ള ക്യാവിലേ അവസ്ഥ ഉൾപ്പെടുത്തൽ പ്രവർത്തനത്തിന് വേണ്ടി താഴെ പറയുന്ന ഘട്ടങ്ങൾ നിർവ്വചിക്കുവാൻ കഴിയുമോ എന്ന് പരിശോധിക്കുക.

- ഘട്ടം 1: ക്യാവിലേക്കു ഉൾപ്പെടുത്തുവാനുള്ള വില ഒരു വേരിയബിളിലേക്കു സ്ഥികരിക്കുക
- ഘട്ടം 2: Rear എം്പി വില ഒന്ന് കുടുക്ക.
- ഘട്ടം 3: Rear എം്പി സ്ഥാനത്ത് വില സംഭരിക്കുക.

ശുന്നമായ ഒരു ക്യൂവിലേക്കു ആദ്യമായി ഉൾപ്പെടുത്തൽ പ്രവർത്തനം ചെയ്യുന്നോൾ Front, Rear എന്നിവയുടെ വില ഒന്നുവച്ചു കൂടുന്നു. അതായൽ Rear ലും Front ലും 0 ആയി മാറുന്നു. അതിനു ശേഷം വരുന്ന ഉൾപ്പെടുത്തലുകൾക്കുണ്ടാണ് Rear മാത്രം കൂടുന്നു. Rear 9 ആകുന്നതു വരെ (അരു യുടെ അവസാനത്തെ സൂചിക) ഇത് തുടരുന്നതാണ്. അതിനു ശേഷം വീണ്ടും ഉൾപ്പെടുത്താൻ പ്രവർത്തനം ചെയ്യുവാൻ ശ്രമിക്കുകയാണെങ്കിൽ സ്ലാക്കിലേതു പോലെ ‘ക്യൂ ഓവർ ഫ്ലോ’ (queue overflow) സംഭവിക്കുന്നു. ക്യൂവിൽ ഉൾപ്പെടുത്തൽ പ്രവർത്തനം നടത്തുവാനുള്ള അൽഗോറിതം നമുക്ക് എഴുതി നോക്കാം.

### ക്യൂവിൽ ഉൾപ്പെടുത്തൽ പ്രവർത്തനം ചെയ്യുവാനുള്ള അൽഗോറിതം

ക്യൂ നടപ്പിലാക്കുന്നതിന് വേണ്ടി, N വലുപ്പമുള്ള QUEUE[N] എന്ന ഒരു അരേ പരിഗണിക്കുക. ക്യൂവിൽ മുൻഭാഗത്തെയും പിൻഭാഗത്തെയും സൂചിപ്പിക്കുന്നതിനായി FRONT, REAR എന്നീ രേഖ വേരിയബില്ലുകൾ ഉപയോഗിക്കുന്നു. ക്യൂവിലേക്കു ഉൾപ്പെടുത്താനുള്ള ഡാറ്റ VAL എന്ന വേരിയബില്ലിൽ സംഭരിച്ചു വയ്ക്കുന്നു. തുടങ്ങുക, അവസാനിപ്പിക്കുക എന്നീ നിർദ്ദേശങ്ങൾക്കിടയിൽ കൊടുത്തിരിക്കുന്ന ഘട്ടങ്ങൾ ഉൾപ്പെടുത്തൽ പ്രവർത്തനത്തെ വിശദീകരിക്കുന്നു.

തുടങ്ങുക

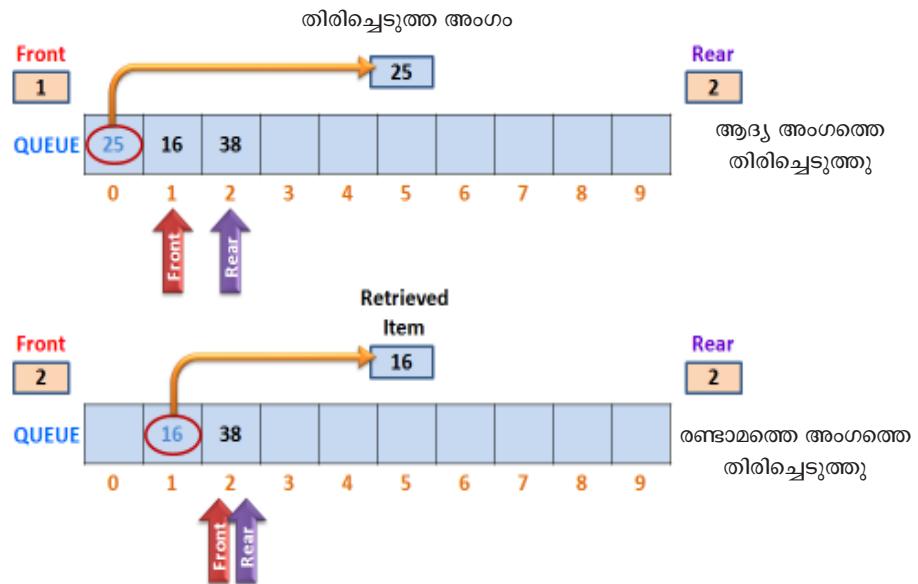
- 1: അമവാ (REAR == -1) ആണെങ്കിൽ //സഹായിക്കുന്ന ലഭ്യത പരിശോധിക്കുന്നു
- 2: FRONT = REAR = 0 ‘ക്യൂ ഓവർ ഫ്ലോ’ പ്രിൻ്റ് ചെയ്യുക
- 3: Q[REAR] = VAL പൂരിതേക്കു പോകുക
- 4: അല്ലെങ്കിൽ അമവാ (REAR < N-1) ആണെങ്കിൽ // സഹായിക്കുന്ന ലഭ്യത പരിശോധിക്കുന്നു.
- 5: REAR = REAR + 1
- 6: Q[REAR] = VAL
- 7: അല്ലെങ്കിൽ
- 8: Print "Queue Overflow "
- 9: പൂരിതേക്കു പോകുക  
അവസാനിപ്പിക്കുക

### b. നീക്കം ചെയ്ത പ്രവർത്തനം (Deletion operation)

ക്യൂ വിശദീകരിക്കുന്ന മുൻഭാഗത്തു നിന്നും അംഗത്വത്തെ ഒഴിവാക്കുന്ന പ്രക്രിയയെയാണ് നീക്കം ചെയ്ത എന്ന് പറയുന്നത്. നീക്കം ചെയ്തിന് ശേഷം Front എഴു വില ഒന്നുവച്ചു കൂടുന്നു. അരു ഉപയോഗിച്ചുള്ള ക്യൂവിൽ നീക്കം ചെയ്ത പ്രവർത്തനം നടത്തുന്നോൾ തമാർപ്പത്തിൽ അംഗങ്ങളെ നീക്കം ചെയ്യുന്നില്ല. മരിച്ച് Front എഴു വില കൂട്ടിക്കൊണ്ട് അതരം അംഗങ്ങളെ ഉപയോഗിക്കുന്നതിൽ നിന്നും തടസ്സപ്പെടുത്തുന്നു.



അരു ഉപയോഗിച്ച് നടപ്പിലാക്കിയിരിക്കുന്ന ഒരു ക്യൂ പരിഗണിക്കുക. പിത്രം 3.10 തുണിക്കും ചെയ്ത പ്രവർത്തനത്തിന് വിധേയമാകുന്ന ക്യൂ വിശദീകരിക്കുന്നു. ഈ പ്രവർത്തനത്തിനുള്ള വിവിധ ഘട്ടങ്ങൾ നിർവ്വചിക്കാൻ നമുക്ക് ശ്രമിക്കാം.



ചിത്രം 3.10: തുടർച്ചയായ നീക്കം ചെയ്ത് പ്രവർത്തനം നടത്തിയതിന് ശേഷമുള്ള ക്യൂവിൻ്റെ അവസ്ഥ നീക്കം ചെയ്ത് പ്രവർത്തനത്തിന് താഴെ പറയുന്ന ഘട്ടങ്ങൾ നമുക്ക് നിർവചിക്കാം.

- Step 1: Front ന്റെ സ്ഥാനത്തുള്ള അംഗത്തെ ഒരു വേരിയബിളിലേക്കു സ്വീകരിക്കുക.
- Step 2: Front ന്റെ വില ഒന്ന് കുടുക.

ചിത്രം 3.10 പ്രകാരം ക്യൂവിൻ്റെ മുൻഭാഗത്തെക്ക് അംഗങ്ങളുടെ സ്ഥാനമാറ്റം നടക്കുന്നില്ല. അതായത് നിരുജീവിതത്തിലെപ്പോലെയല്ല ഇവിടത്തെ ക്യൂവിൻ്റെ സകൽപ്പം. ക്യൂ ഡാറ്റസ്ട്രക്ചറിൽ നീക്കം ചെയ്ത് പ്രവർത്തനത്തിൽ അംഗങ്ങളുടെ സ്ഥാനമാറ്റം നടത്തുന്നതിന് പകരം Front ന്റെ വില കുടുകയാണ് ചെയ്യുന്നത്. ക്യൂവിൽ അംഗങ്ങൾ ഇനിതേതാളം കാലം നീക്കം ചെയ്ത് പ്രവർത്തനത്തിന് മേൽനിരവചിച്ച രണ്ടു ഘട്ടങ്ങൾ മതിരാകുന്നതാണ്. രണ്ടാമത്തെ അംഗത്തിന്റെ നീക്കം ചെയ്തുകിട്ടിയാൽ ശേഷമുള്ള ക്യൂവിൻ്റെ അവസ്ഥ നോക്കുക. Front ന്റെയും Rear ന്റെയും വില ഒരേ അംഗത്തെ സൂചിപ്പിക്കുന്നു. അതായത് രണ്ട് എന്ന അരയുടെ സൂചിക. ക്യൂവിൽ ഒരു നീക്കം ചെയ്ത് പ്രവർത്തനം കൂടി നടത്തുന്നതായി അനുമാനിക്കുക. നടപടി ക്രമം പ്രകാരം Front ന്റെ വില 3 ആയി മാറുന്നു. ഈ റൈറ്റ് Rear ന്റെ വിലയേക്കാൾ കുടുതലാണ്. നമുക്കിനിയാം ഒരു ക്യൂവിൽ അത് ഉചിതമല്ല. മാത്രമല്ല ക്യൂ ഇപ്പോൾ ശുന്നമാണെന്നു നമുക്ക് കാണാവുന്നതാണ്. ക്യൂ ശുന്നമായാൽ Front ന്റെയും Rear ന്റെയും വില -1 ആക്കണമെന്ന് നേരത്തെ പരാമർശിച്ചതാണ്. അപ്പോൾ അവസ്ഥാനുഠനം അംഗത്തെ നീക്കം ചെയ്താൽ Front ന്റെയും Rear ന്റെയും വില -1 ആക്കാനുള്ള ഘട്ടം കൂടി അൽഗോറിത്മത്തിൽ ഉൾപ്പെടുത്തേണ്ടതാണ്. ഈ അവസ്ഥയിൽ വീണ്ടും നീക്കം ചെയ്ത് അനുവദനിയമല്ല. ശുന്നമായ ക്യൂ വിൽ നിന്നും നീക്കം ചെയ്യാൻ ശ്രമിക്കുമ്പോളുണ്ടാകുന്ന അവസ്ഥയെ ‘ക്യൂ അണ്ഡർഫ്ലോ’ (queue underflow) എന്ന് പറയുന്നു. ഈ നമുക്ക് ക്യൂവിൽ നിന്നും നീക്കം ചെയ്ത് പ്രവർത്തനം നടത്താനുള്ള അൽഗോറിതം തയാറാക്കാം.

### രു കൃവിൽ നിന്ന് നീകം ചെയ്ത പ്രവർത്തനം നടത്താനുള്ള അവലോകനം

കൃ നടപ്പിലാക്കാൻ വേണ്ടി പരമാവധി  $N$  അംഗങ്ങളുള്ള  $QUEUE[N]$  എന്ന രു അറ പരിഗണിക്കുക. FRONT, REAR എന്നീ വേറയിബിള്ളുകൾ കൃവിന്റെ മുൻഭാഗത്തയും പിൻഭാഗത്തയും സൂചിപ്പിക്കാൻ ഉപയോഗിക്കുന്നു. കൃവിൽ നിന്നും നീകം ചെയ്യുന്ന അംഗത്തെ സംഭരിക്കുന്നതിനായി VAL എന്ന വേറയിബിൾ ഉപയോഗിക്കുന്നു. നീകം ചെയ്ത പ്രവർത്തനത്തിനുള്ള റല്ലിങ്കുക, അവസാനിപ്പിക്കുക എന്നീ നിർദ്ദേശങ്ങൾക്കിടയിൽ നൽകിയിരിക്കുന്നു.

തുടങ്ങുക

- 1: അമവാ ( $FRONT > -1$ ) ആണെങ്കിൽ // ശുന്ധമായ അവസ്ഥ പതിശോധിക്കുന്നു
- 2:  $VAL = Q[FRONT]$
- 3:  $FRONT = FRONT + 1$
- 4: അല്ലെങ്കിൽ
- 5: ‘കൃ ഓവർ ഫ്ലോ’ എന്നു പ്രിൻ്റ് ചെയ്യുക
- 6: പുറത്തേക്കു പോകുക
- 7: അമവാ ( $FRONT > REAR$ ) ആണെങ്കിൽ // അവസാനത്തെ അംഗത്തിന്റെ നീകം ചെയ്ത പതിശോധിക്കുന്നു
- 8:  $FRONT = REAR = -1$
- 9: പുറത്തേക്കു പോകുക

അവസാനിപ്പിക്കുക



#### കൃ പ്രവർത്തങ്ങൾക്ക് വേണ്ടിയുള്ള C++ ഹാൻഡ്കൗകൾ

$n$ , front, rear എന്നിവ ഫ്രോബെൽ വേറയിബിൾ ആണെന്ന് കരുതുക

കൃ വിൽ ഉൾപ്പെടുത്തൽ പ്രവർത്തനം	കൃവിൽ നീകം ചെയ്ത പ്രവർത്തനം
<pre>void ins_q(int queue[], int val) {     if (rear == -1)     {         front=0;         rear=0;         q[rear]=val;     }     else (if rear &lt; n-1)     {         rear++;         q[rear]=val;     }     else         cout&lt;&lt;"Overflow"; }</pre>	<pre>int del_q(int queue[]) {     int val;     if (front &gt; -1)     {         val=q[front];         front++;     }     else         cout&lt;&lt;"Underflow";     if (front &gt; rear)     {         front= -1;         rear= -1     }     return val; }</pre>



### ക്യൂവിൻ്റെ ഉപയോഗം

ക്യൂ അധികവും കമ്പ്യൂട്ടർ സയൻസിൽ ജോബ് ആസൃതണം ചെയ്യുവാൻ വേണ്ടിയാണ് ഉപയോഗിക്കുന്നത്. മഹാരി, പ്രോസസ്സർ, എയലുകൾ മുതലായവ ആസൃതണം ചെയ്യുന്നതിനാണ് ഓപ്പറേറ്റീംഗ് സിസ്റ്റം ക്യൂ ഉപയോഗിക്കുന്നത്. ഇതിന് ഒരു ഉദാഹരണമാണ് പ്രിൻ്റർ ക്യൂ. പ്രോസസ്സറിനെ അപേക്ഷിച്ചു പ്രിൻ്ററിന് വേഗത കുറവായതിനാൽ പ്രിൻ്റർ ചെയ്യുവാനുള്ള ജോലികളെല്ലാം പ്രിൻ്റർ ബഹരിൽ നിന്നേക്കും പ്രിൻ്റ് ബഹർ FIFO തത്ത്വം പിന്തുടരുന്നതിനാൽ അതിനെ ഒരു ക്യൂ ആയി കണക്കാക്കാം.

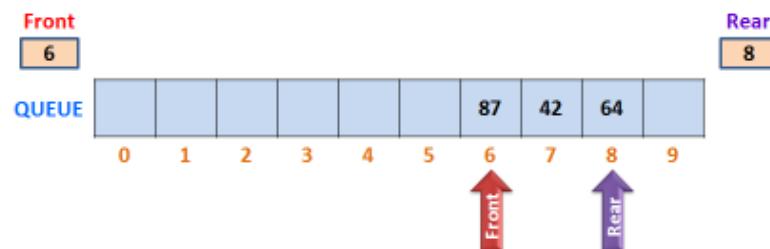
### 3.3.3 വൃത്താകൃതി ക്യൂ (Circular queue)

നമ്മൾ ഇതുവരെ ചർച്ച ചെയ്ത ക്യൂവിനെ രേഖാചിത്രം എന്നാണ് പറയുക. ഇതിലെ അംഗങ്ങൾ ഒരു നേർരേഖയായിട്ട് അല്ലെങ്കിൽ ഒരു നേർവാലിയായിട്ട് അണംബേണ്ട ദുകൾ. ഇത്തരം ക്യൂകളുടെ രണ്ടുംശൾ തമ്മിൽ ഒരിക്കലും കൂട്ടിമുട്ടുകയില്ല.

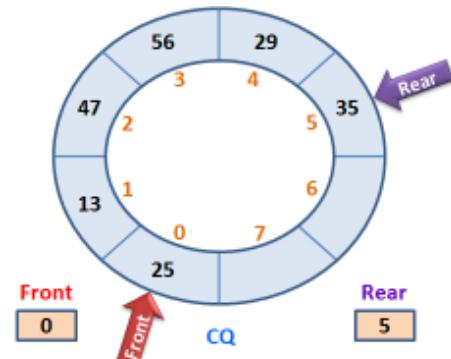
രേഖാചിത്രം ക്യൂവിൽ ഒരു നൃത്യതയ്യുണ്ട്. ചിത്രം 3.11 ത്ത് കൊടുത്തതിനിക്കുന്ന, 6 അംഗങ്ങളെ നീക്കം ചെയ്തു പ്രവർത്തനങ്ങൾക്കു വിധേയമായ ക്യൂ പരിഗണിക്കുക. നിലവിൽ അതിൽ മൂന്ന് അംഗങ്ങൾ മാത്രം മേഖലയിലുണ്ട്. സ്വാഭാവികമായിട്ടും Front റെഞ്ച് വില 6 ഉം Rear റെഞ്ച് വില 8 ഉം ആയിരിക്കും.

ആദ്യത്തെ 6 സ്ഥാനങ്ങൾ ശുന്നമാണെങ്കിലും ഉൾപ്പെടുത്തൽ അൽഗോറിതം പ്രകാരം നമുക്ക് ഒരു അംഗത്തെ മാത്രമേ കൂട്ടിച്ചേര്ക്കാൻ കഴിയു. അവസാനത്തെ സ്ഥാനത്ത് ഒരു അംഗം മാത്രമേ ഉള്ള എന്ന് കരുതുക. അങ്ങനെയെങ്കിൽ പുതിയ തായി ഒരു അംഗത്തെ കൂട്ടിച്ചേര്ക്കുവാൻ ശ്രമിക്കുകയാണെങ്കിൽ ‘ക്യൂ ഓവർ ഫ്ലോ’ (Queue Overflow) സംഭവിക്കുന്നു. രേഖാചിത്രം ക്യൂവിൻ്റെ ഇള പരിമിതി വ്യത്താക്യൂതി ക്യൂ ഉപയോഗിച്ച് പരിഹരിക്കാവുന്നതാണ്. ചിത്രം 3.12 ത്ത് കാണിച്ചിരിക്കുന്നത് പോലെ രണ്ടുംശളും കൂട്ടി മുട്ടുന്ന ഒരു ക്യൂ ആണിത്.

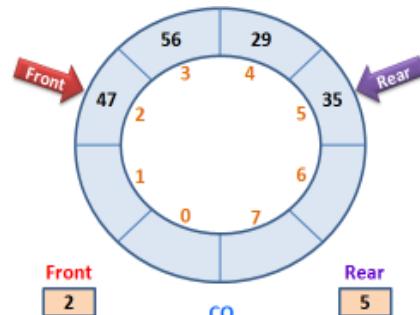
ചിത്രം 3.13 ത്ത് രണ്ട് അംഗങ്ങളെ നീക്കംചെയ്തു പ്രവർത്തനങ്ങൾ നടത്തി എന്ന് അനുമാനിക്കുക. അപ്പോൾ Front റെഞ്ച് വില 2 ഉം ചിത്രം 3.13(a) ത്ത് കാണിച്ചിരിക്കുന്നത് പോലെ 4 ശുന്നമായ സ്ഥാനങ്ങളും സ്ഥലങ്ങളും



ചിത്രം 3.11: തുടർച്ചയായ 6 അംഗങ്ങളെ നീക്കം ചെയ്തതിനു ശേഷം 3 അംഗങ്ങളുള്ള ക്യൂ

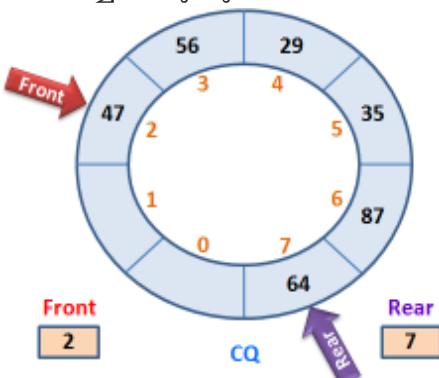


ചിത്രം 3.12: വൃത്താകൃതി ക്യൂ

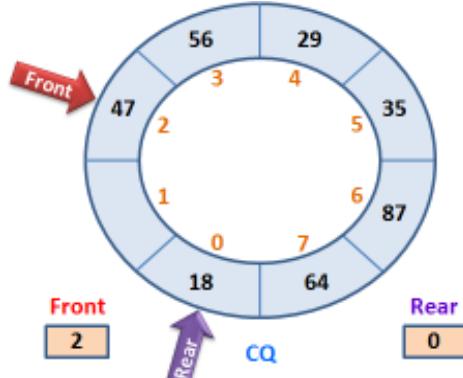


ചിത്രം 3.13: നീക്കം ചെയ്തു പ്രവർത്തനത്തിന് ശേഷമുള്ള വൃത്താകൃതി ക്യൂ

നമുക്ക് ലഭിക്കുന്നു. ഈ സ്ഥാനങ്ങളുടെ സുചികകളുടെ വില തമാക്രമം 6, 7, 0, 1 എന്നിങ്ങനെയായിരിക്കും. നമ്മൾ വീണ്ടും രണ്ട് അംഗങ്ങളെ കൂട്ടിച്ചേർക്കുകയാണെങ്കിൽ Rear റെ വില 7 ആയി മാറുന്നു. ചിത്രം 3.13(b) തുടർന്നു കാണിച്ചിരിക്കുന്നത് പോലെ 0,1 എന്നീ സുചികകളുടെ സ്ഥാനം ഇപ്പോഴും ശുന്നമായിട്ടാണുള്ളത്. അതിനാൽ വീണ്ടും ഉൾപ്പെടുത്തൽ പ്രവർത്തനം ഇവിടെ നിർവ്വഹിക്കാവുന്നതാണ്. ഇത്തവണ Rear റെ വില 0 ആയി നിശ്ചയിക്കുകയും അതിനു ശേഷം ഉൾപ്പെടുത്താൻ പ്രവർത്തനം നിർവ്വഹിക്കുകയും ചെയ്യാവുന്നതാണ്. കൂടും കൂടും ഈ അവസ്ഥയെ ചിത്രം 3.13(c) തുടർന്നു കാണിച്ചിരിക്കുന്നു.



ചിത്രം 3.13 (b): Rear റെ അതിന്റെ ഉയർന്ന വില



ചിത്രം 3.13(c): Rear റെ അതിന്റെ താഴ്ന്ന വില

### നമുക്ക് വിലയിരുത്താം



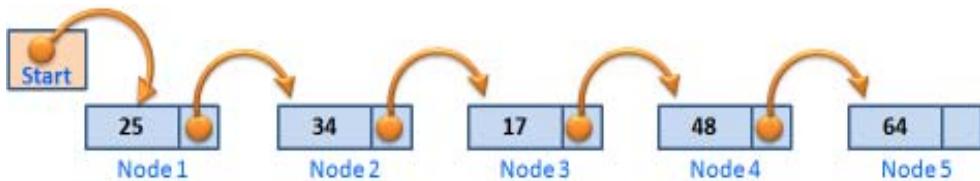
1. ഡാറ്റാ സ്റ്റെക്ക്‌ചർ എന്നാലെന്ത്?
2. ഡാറ്റ ക്രമീകരണത്തിനായി സ്ലാക്കിൽ ഉപയോഗിക്കുന്ന തത്ത്വം \_\_\_\_\_ ആകുന്നു.
3. FIFO എന്ന തത്ത്വം ഉപയോഗപ്പെടുത്തുന്ന ഡാറ്റാസ്റ്റെക്ക്‌ചർിന്റെ പേരേഴുതുക.
4. അരെയിലെ അണ്ഡർഫ്ലോ (underflow) എന്നാലെന്ത്?
5. സ്ലാക്കിന്റെ ഏത് അംഗമാണ് നീക്കം ചെയ്യാൻ സാധിക്കുക (ആദ്യത്തെത്ത് അല്ലെങ്കിൽ അവസാനത്തെത്ത്)?

### 3.4 ലിക്ക്‌ഡ് ലിസ്റ്റ് (Linked list)

അംഗങ്ങളുടെ എല്ലാത്തിന് പരിമിതിയില്ലാത്ത ഡാറ്റ അംഗങ്ങളുടെ ശേഖരത്തെയാണ് ലിക്ക്‌ഡ് ലിസ്റ്റ് എന്ന് പറയുന്നത്. അരെകൾ ഉപയോഗിച്ച് പ്രാവർത്തികമാക്കുന്ന യുക്തിപരമായ ആശയങ്ങൾ മാത്രമാണ് കഴിഞ്ഞ ഭാഗങ്ങളിൽ വിശദീകരിച്ച സ്ലാക്, കൂടും മുതലായ ഡാറ്റാസ്റ്റെക്ക്‌ചറുകൾ. അതിനാൽ ഇവ (സ്ലാറിക്) ഡാറ്റാസ്റ്റെക്ക്‌ചറുകളാണ്. എന്നാൽ ലിക്ക്‌ഡ് ലിസ്റ്റ് ഒരു (ബൈനറിക്) ഡാറ്റാസ്റ്റെക്ക്‌ചർ ആണ്. പുതിയ ഡാറ്റ ഇനങ്ങൾ കൂട്ടിച്ചേർക്കുമ്പോൾ ഇത് വളരുകയും നീക്കം ചെയ്യുമ്പോൾ ചുരുങ്ങുകയും ചെയ്യുന്നു. ലിക്ക്‌ഡ് ലിസ്റ്റിന് മുഴുവൻ അംഗങ്ങൾക്കും ആവശ്യമായ മെമ്മറി ആദ്യമേ

അനുവദിക്കുകയില്ല. ലിസ്റ്റിലേക്ക് ഒരു പുതിയ അംഗത്വത്തുനൽകി തൊട്ടു മുമ്പാണ് ആ അംഗത്വത്തിന് ആവശ്യമായ മെമ്മറി അനുവദിക്കുന്നത്. അതുകൊണ്ടാണ് ഇതിനെ (ഡൈഗ്രാഫിക്) ഡാറ്റാ സ്റ്റ്രക്ചറുകളായി കരുതുന്നത്. അരെ അധിഷ്ഠിത ഡാറ്റാ സ്റ്റ്രക്ചറുകളുമായി ലിങ്ക് ലിസ്റ്റിനുള്ള മഡ്രാരു വ്യത്യാസം ലിങ്ക് ലിസ്റ്റിലെ അംഗങ്ങൾ മെമ്മറിയിൽ പല ഭാഗത്തായിട്ടാണ് സംഭരിക്കപ്പെടുന്നത്. എന്നാൽ അവ പോയിന്ത്രു കളുടെ സഹായത്തോടെ പരസ്പരം ബന്ധപ്പിച്ചിരിക്കുന്നു. നമ്മൾ അധ്യായം ഒന്നിൽ പഠിച്ചത് പ്രകാരം പോയിന്ത്രു എന്നാൽ മെമ്മറി സ്ഥാനങ്ങളുടെ അടയാളം സംഭരിച്ചു വച്ചിരിക്കുന്ന വേരിയബിള്ളാണ്. അപ്പോൾ ഒരുക്കാരും വ്യക്തമാണ് ലിങ്ക് ലിസ്റ്റിലെ ഒരു അംഗത്വത്തിന് ഡാറ്റയും അടയാളം ഉണ്ടായിരിക്കും. ലിങ്ക് ലിസ്റ്റിലെ ഒരു അംഗത്വത്തെ നോഡ് (Node) എന്ന് വിളിക്കുന്നു. നോഡിൽ അടങ്കിയിരിക്കുന്ന അടയാളിനെ ലിങ്ക് (Link) എന്ന് പറയുന്നു.

ഒരു ഡാറ്റയും ലിക്കും (ലിസ്റ്റിലെ അടുത്ത നോഡിലേക്കുള്ള പോയിന്ത്രു) അടങ്കിയ നോഡുകളുടെ ശേഖരമാണ് ലിങ്ക് ലിസ്റ്റ്. അതായത് ലിസ്റ്റിലെ ആദ്യത്തെ നോഡിൽ ആദ്യത്തെ ഡാറ്റ അംഗവും രണ്ടാമത്തെ നോഡിന്റെ അടയാളും അടങ്കിയിരിക്കുന്നു. രണ്ടാമത്തെ നോഡിൽ രണ്ടാമത്തെ ഡാറ്റ അംഗവും മൂന്നാമത്തെ നോഡിന്റെ അടയാളും അടങ്കിയിരിക്കുന്നു. ഇപ്രകാരം ലിസ്റ്റ് തുടരുന്നു. അവസാനത്തെ നോഡിൽ അവസാനത്തെ ഡാറ്റയും ഒരു നശി പോയിന്ത്രും (Null Pointer) അടങ്കിയിരിക്കുന്നു. എവിടെയും പോയിന്ത്രു ചെയ്യാതെ പോയിന്ത്രീഡ് Null Pointer. അങ്ങനെയെങ്കിൽ ആദ്യത്തെ നോഡിന്റെ അടയാളം എവിടെയാണുണ്ടാകുക? ആദ്യത്തെ നോഡിന്റെ അടയാളം അടങ്കിയ ഒരു പ്രത്യേക പോയിന്ത്രു എല്ലാ ലിങ്ക് ലിസ്റ്റിലും ഉണ്ടായിരിക്കും. ഇതിനെ സ്റ്റാർട്ട് (Start) അല്ലെങ്കിൽ ഹെഡർ (Header) എന്നു പറയുന്നു. അഞ്ച് അംഗങ്ങൾ അടങ്കിയ ഒരു ലിങ്ക് ലിസ്റ്റിനെ ചിത്രം 3.14 ത്ത് പ്രദർശിപ്പിച്ചിരിക്കുന്നു.



ചിത്രം 3.14: നോഡുകളുള്ള ലിങ്ക് ലിസ്റ്റ്

ഡാറ്റയായി ഒരു സംവയയും പോയിന്ത്രു ആയി അടുത്ത നോഡിലേക്കു ചുണ്ടിക്കാണിക്കുന്ന ലിക്കും അടങ്കിയ നോഡുകൾ ആണ് ചിത്രത്തിൽ കാണിച്ചിരിക്കുന്നത്. എല്ലാ നോഡുകളുടെയും വലുപ്പം സമമായിരിക്കും. അതായത് ഓരോ നോഡിനു വേണ്ടിയും മാറ്റി വയ്ക്കുന്ന മെമ്മറി സമമായിരിക്കും.

### 3.4.1 ലിങ്ക് ലിസ്റ്റ് പ്രാവർത്തനക്രമാക്കൽ (Implementation of linked list)

ഓരോ നോഡിനും മെമ്മറി അനുവദിക്കുന്ന നോഡുകളുടെ ശേഖരമാണ് ലിങ്ക് ലിസ്റ്റ് എന്ന് നമ്മൾ നേരത്തെ കണ്ണു കഴിഞ്ഞു. നോഡിനു വേണ്ടിയുള്ള മെമ്മറിയിൽ ഏറ്റവും കുറവുത്തത് രണ്ടു തരത്തിൽപ്പെട്ട ഡാറ്റയാണ് അടങ്കിയിരിക്കുന്നത്, അതിൽ ഒന്ന് അംഗത്വത്തിന്റെ തമാർമ്മ ഡാറ്റയും രണ്ടാമത്തെത്തെ അടുത്ത നോഡിലേക്കുള്ള പോയിന്ത്രും

ആയിരിക്കും. വിവിധ തരത്തിൽപ്പെട്ട ഡാറ്റകൾ അടങ്ങിയ ഉപയോകത്യു നിർവ്വചിതമായ ഡാറ്റ എന്ന് ആണ് സ്റ്റ്രക്ചർ എന്നത് നാം അഡ്യായം ഒന്നിൽ കണ്ടൊണ്ട്. സ്റ്റ്രക്ചർിലെ അംഗം ഒരു പോയിന്റർ ആകാം എന്ന് മാത്രമല്ല അതേ സ്റ്റ്രക്ചർിലേക്കു സൂചിപ്പിക്കുന്ന പോയിന്ററുമാകാം. അത്തരം സ്റ്റ്രക്ചറുകളെ സ്വയം സൂചിതസ്റ്റ്രക്ചറുകൾ (Self Referential Structure) എന്ന് പറയുന്നു. അതായത് സ്വയം സൂചിത സ്റ്റ്രക്ചറുകളുടെ സഹായത്തോടെയാണ് ലിക്ക് ലിസ്റ്റുകൾ നിർമ്മിക്കുന്നത്. താഴെ പറയുന്ന സ്വയം സൂചിതസ്റ്റ്രക്ചർ ഉപയോഗിച്ച് ചിത്രം 3.14 ത്തെ കാണിച്ചിരിക്കുന്ന ലിക്ക് ലിസ്റ്റിന്റെ നോഡുകൾ രൂപകൽപ്പന ചെയ്തിരിക്കുന്നു.

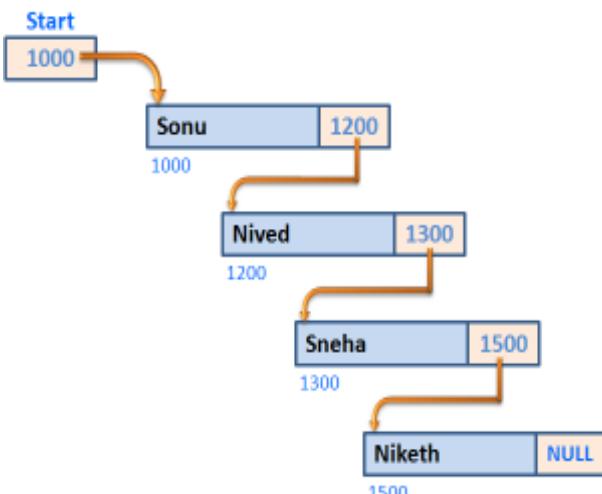
```
struct Node
{
    int data;
    Node *link;
};
```

സ്റ്റ്രക്ചർിൽ പോൾ Node ആണെന്നും സ്റ്റ്രക്ചർിലെ രണ്ടാമത്തെ അംഗം അതേ സ്റ്റ്രക്ചർിൽ തരത്തിലുള്ള ഒരു പോയിന്റർ ആണെന്നും നമുക്ക് കാണാം. താഴെ പറയുന്ന പ്രസ്താവന ഉപയോഗിച്ച് ആദ്യത്തെ നോഡിനെ സൂചിപ്പിക്കുന്ന Start എന്ന പ്രത്യേക പോയിന്റർ നിർമ്മിക്കാവുന്നതാണ്.

```
struct Node * Start;           or           Node *Start;
```

ചിത്രം 3.15 ത്തെ സ്റ്റ്രീഞ്ചുകളുടെ ഒരു ലിക്ക് ലിസ്റ്റ് ആണ് കാണിച്ചിരിക്കുന്നത്. ഇതിലെ നോഡുകളുടെ ഡാറ്റാസ്റ്റ്രീഞ്ചുകൾ ഉപയോഗിച്ചും ലിക്കുകൾ മറ്റു നോഡുകളുടെ അഡ്യൻ ഉപയോഗിച്ചും നിറച്ചിരിക്കും. ശ്രദ്ധിക്കേണ്ണ കാര്യം ഇവിടെ അഡ്യസ്റ്റുകൾ എന്നത് അനുമാനിച്ചിരിക്കുന്ന ചില സംവ്യക്ഷൾ ആണ്. ഈ നോഡുകളെ താഴെ പറയുന്ന സ്റ്റ്രക്ചർ ഉപയോഗിച്ച് പ്രതിനിധീകരിക്കാവുന്നതാണ്.

```
struct Node
{
    char data[10];
    Node *link;
};
```



ചിത്രം 3.15: ലിക്ക് ലിസ്റ്റ് പ്രവർത്തനക്കമാക്കൽ



പ്രവർത്തന സമയത്ത് ആദ്യത്തെ നോഡിന് 1000 എന്ന അധിക്കൂഷി മെമ്മറി സ്ഥാനം നീക്കിവയ്ക്കുന്നതായി അനുമാനിക്കുക. അതിനാൽ Start പോയിൻറിന്റെ വില 1000 ആയിരിക്കും. ഒന്നാമത്തെ നോഡിന്റെ ധാര ഭാഗത്തു "Sonu" എന്ന സ്ട്രൈം സംഭരിച്ചിരിക്കുന്നു. രണ്ടാമത്തെ നോഡിന് 1200 എന്ന മെമ്മറി അധിക ആണ് നൽകിയിരിക്കുന്നത്. അതിന്റെ ധാര യാറ്റാഭാഗത്തു "Nived" എന്ന വില സംഭരിച്ചിരിക്കുന്നു. രണ്ടാമത്തെ നോഡ് ആയതിനാൽ ആദ്യത്തെ നോഡിന്റെ ലിങ്ക് ഭാഗത്ത് ഈ നോഡിന്റെ അധിക സംഭരിച്ചിരിക്കുന്നു. ചിത്രത്തിൽ കാണിച്ചിരിക്കുന്നത് പോലെ അവസാനത്തെ നോഡിൽ ലിങ്കിന്റെ സ്ഥാനത്ത് NULL പോയിൻറും ആയിരിക്കും ഉണ്ടായിരിക്കുക.

### 3.4.2 ലിങ്ക് ലിസ്റ്റിലെ പ്രവർത്തനങ്ങൾ (Operations on linked list)

ഭാഗം 3.1.2 ലെ പ്രതിപാദിച്ചിരിക്കുന്ന എല്ലാ പ്രവർത്തനങ്ങളും ഉപാധികളില്ലാതെ ലിങ്ക് ലിസ്റ്റിൽ ചെയ്യുവാൻ സാധിക്കുന്നതാണ്. പക്ഷേ നിർമ്മാണം, കടന്നുപോകൽ, ഉൾപ്പെടുത്തൽ, നീക്കം ചെയ്യൽ എന്നീ പ്രവർത്തനങ്ങളെപ്പറ്റി മാത്രമേ നമ്മൾ ഇവിടെ ചർച്ച ചെയ്യുന്നുള്ളൂ. ഉപരിപറമ സമയത്തു ബാക്കി പ്രവർത്തനങ്ങളെപ്പറ്റി നിങ്ങൾ പഠിക്കുന്നതാണ്.

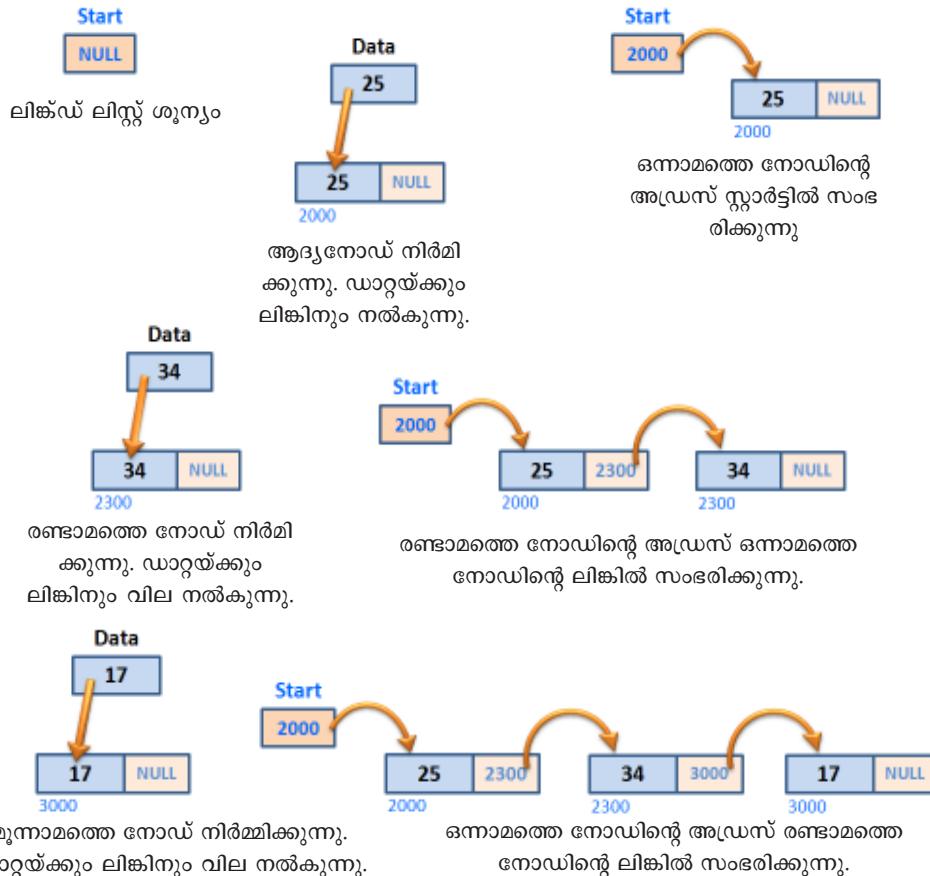
#### a. ലിങ്ക് ലിസ്റ്റിന്റെ നിർമ്മാണം (Creation of linked list)

തുടക്കത്തിൽ നമുക്ക് സയം സൂചകമായി ഒരു സ്ട്രക്ചർ നിർവ്വചിക്കേണ്ടതുണ്ട്. Start അല്ലെങ്കിൽ Header എന്ന ഒരു പോയിൻറർ വേരിയബിളിനെ നിർവ്വചിച്ച് നൾ (NULL) എന്ന വില നൽകുന്നു. ഈ ആവശ്യകതയുസരിച്ച് നോഡുകൾക്കുള്ള മെമ്മറി അസ്ഥിരമായി നീക്കിവച്ചുകൊണ്ട് ലിങ്ക് ലിസ്റ്റ് നിർമ്മിക്കാം.



നമുക്ക് ചെയ്യാം

ലിങ്ക് ലിസ്റ്റ് നിർമ്മാണ സമയത്തെ അവസ്ഥയാണ് ചിത്രം 3.16 ലെ കാണിച്ചിരിക്കുന്നത്. Node എന്ന പേരുള്ള ഒരു സയം സൂചിത സ്ട്രക്ചർ നിർമ്മിച്ചതായും Start എന്ന NULL തരത്തിൽപ്പെട്ട പോയിൻറിനു Node എന്ന വില നൽകിയതായും നമുക്ക് അനുമാനിക്കാം. ഈ താഴെ പറയുന്ന ചിത്രങ്ങൾ നിരീക്ഷിച്ച് ലിങ്ക് ലിസ്റ്റ് നിർമ്മാണത്തിനാവശ്യമായ നടപടിക്രമം തയാറാക്കുക.



ചിത്രം 3.16: ലിക്ക്‌ൾ ലിറ്റ് നിർമ്മാണത്തിൽ തുടർച്ചയായ പ്രവർത്തനങ്ങൾ

താഴെ പറയുന്ന ഘട്ടങ്ങൾ വികസിപ്പിക്കുവാൻ സാധിക്കുമോ എന്ന് പരിശോധിക്കുക.

- എടു 1: ഒരു നോധ് നിർമ്മിച്ച് അതിന്റെ മെമ്മറി അദ്യനോധ ലഭ്യമാക്കുക.
- എടു 2: ഡാറ്റയ്ക്കും NULL എന്ന വിലയും നോധിൽ സംഭരിക്കുക.
- എടു 3: അത് അദ്യനോധ നോധ് ആബ്ലേഷിൽ അതിന്റെ അദ്യനോധ node തും സംഭരിക്കുക.
- എടു 4: അത് അദ്യനോധ നോധ് അല്ലെങ്കിൽ അതിന്റെ അദ്യനോധ തൊട്ടു മുമ്പായാൽ നോധിന്റെ ലിങ്കിൽ സംഭരിക്കുക.
- എടു 5: ഉപയോകതാവിൽ ആവശ്യാനുസരണം എടു 1 മുതൽ 4 വരെ ആവർത്തിക്കുക.

യമാർമ്മത്തിൽ ലിക്ക്‌ൾ ലിറ്റിന്റെ നിർമ്മാണം എന്നത് ലിക്ക്‌ൾ ലിറ്റിന്റെ അവസാന ഭാഗത്ത് ആവർത്തിച്ചു നടത്തുന്ന ഉൾപ്പെടുത്തൽ പ്രവർത്തനം ആണ്. രണ്ടാമത്തെ നോധ് മുതൽ ഉൾപ്പെടുത്തൽ പ്രവർത്തനം ചെയ്യണമെങ്കിൽ നിലവിലുള്ള ലിസ്റ്റിലെ അവസാനത്തെ അംഗത്തിന്റെ അദ്യനോധ ലഭിക്കുന്നതിനാവശ്യമായ കടനുപോകൽ പ്രവർത്തനം നടത്തണം. അതിനാൽ കടനുപോകൽ പ്രവർത്തനത്തെപ്പറ്റി നമുക്ക് വിശദമായി ചർച്ച ചെയ്യാം.

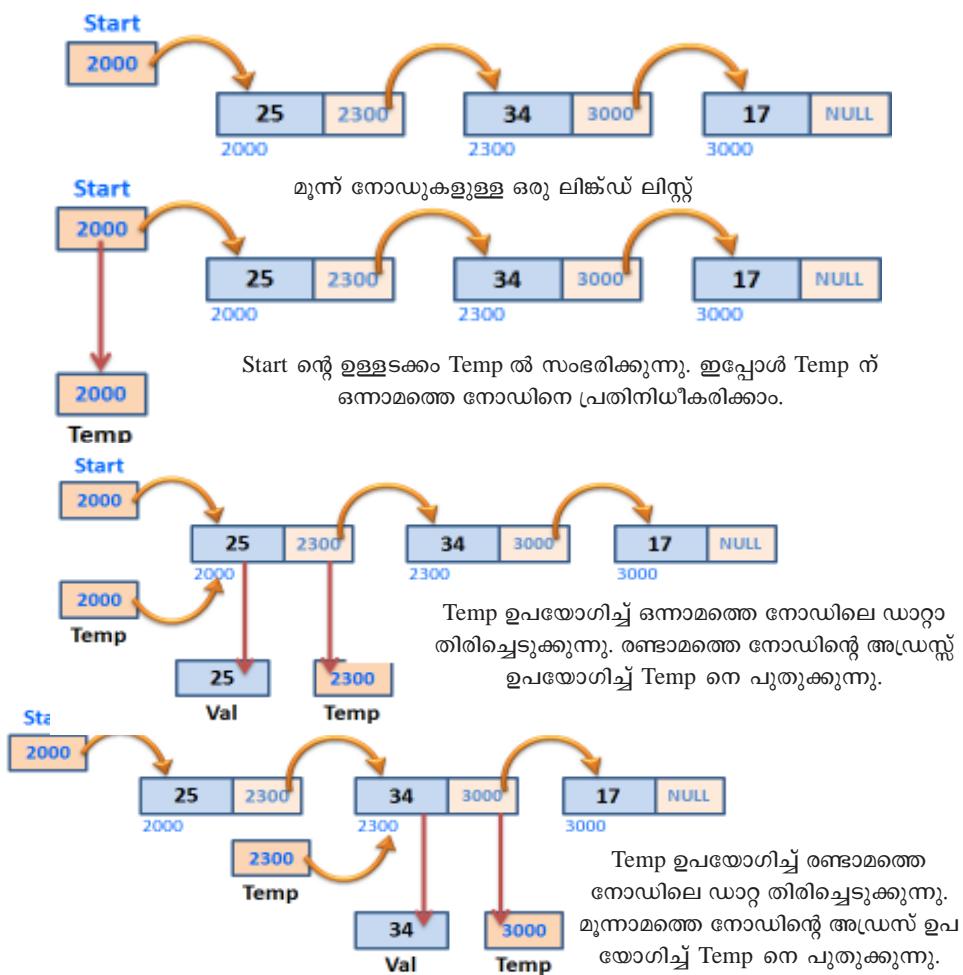
### b. ലിക്സ് ലിസ്റ്റിൽ കടന്നുപോകൽ (Traversing a linked list)

നമുക്കെന്നാം ഒരു ഡാറ്റ സ്റ്റ്രക്ചറിൽ കടന്നുപോകൽ എന്നാൽ അതിലുള്ള ഏല്ലാ അംഗങ്ങളെയും സംശയിക്കലാണ്. ലിക്സ് ലിസ്റ്റുകളുടെ കാര്യത്തിൽ യാത്ര ചെയ്യൽ ആരംഭിക്കുന്നത് ആദ്യത്തെ നോഡ് മുതലാണ്. ആദ്യത്തെ നോഡിന്റെ അധിസ് arrow എന്ന പോയിന്ററിൽ നിന്നും ലഭിക്കുന്നു. ഇതുപയോഗിച്ച് ആരോ (->) ഓപ്പറേറ്ററിൻ്റെ സഹായത്തോടെ നമുക്ക് ഡാറ്റ ഭാഗത്തെ ലഭ്യമാക്കാം. ശേഷം രണ്ടാമത്തെ നോഡിന്റെ അധിസ് അടങ്ങിയ ആദ്യത്തെ നോഡിന്റെ ലിക്സ് ഭാഗത്തെ ഉപയോഗപ്പെടുത്തുന്നു. ഈ അധിസ് ഉപയോഗിച്ച് രണ്ടാമത്തെ നോഡിലെ ഡാറ്റയും ലഭ്യമാക്കുന്നു. നോഡിന്റെ ലിക്സിൽ NULL ലഭിക്കുന്നത് വരെ ഈ പ്രക്രിയ തുടരുന്നു.



നമുക്ക് ചെയ്യാം

ചിത്രം 3.17 തെ നൽകിയിരിക്കുന്ന ലിക്സ് നിരീക്ഷിച്ച് യാത്ര ചെയ്യൽ പ്രവർത്തനത്തിന് വേണ്ട ഘട്ടങ്ങൾ നമുക്ക് നിർവ്വചിക്കാം. Node തരത്തിൽപ്പെട്ട ഒരു പോയിന്റർ ആൺ TEMP എന്നും VAL എന്നത് നോഡിൽ നിന്നുമുള്ള ഡാറ്റ സംഭരിക്കാനാവശ്യമായ ഒരു വേർത്തിയിൽ ആണെന്നും കരുതുക.





ചിത്രം 3.17: ലിങ്ക്ഡ് ലിസ്റ്റിലെ കടന്നുപോകൽ പ്രവർത്തനം

താഴെ പറയുന്ന ഘട്ടങ്ങൾ നിങ്ങൾക്ക് നിർവചിക്കാൻ കഴിയുന്നുണ്ടോ എന്ന് പരിശോധിക്കുക.

- ഘട്ടം 1: ആദ്യത്തെ നോഡിന്റെ അടയാളം Start തോന്നിനും Temp ലേക്ക് സംഭരിക്കുക.
- ഘട്ടം 2: Temp ലെ അടയാളം ഉപയോഗിച്ച് ആദ്യത്തെ നോഡിന്റെ ധാരാ വലിൽ സംഭരിക്കുക.
- ഘട്ടം 3: ഈ നോഡിന്റെ ലിങ്ക് ഭാഗത്തെ അടയാളം (അതായത് അടുത്ത നോഡിന്റെ അടയാളം) Temp ലേക്ക് സ്വീകരിക്കുക.
- ഘട്ടം 4: Temp ലെ അടയാളം NULL അല്ലെങ്കിൽ ഘട്ടം 2 ലേക്ക് പോകുക അല്ലെങ്കിൽ അവസാനിപ്പിക്കുക.

Start ന്റെ വില NULL അല്ലാത്ത സാഹചര്യത്തിൽ ഒരു ലിങ്ക്ഡ് ലിസ്റ്റിനെ നിർമ്മിക്കാൻ മേൽപ്പറഞ്ഞ ഘട്ടങ്ങൾ ഉപയോഗിക്കുന്നു. ഈ തരം സാഹചര്യത്തിൽ അവസാനത്തെ നോഡിന്റെ അടയാളം കണ്ടുപിടിക്കേണ്ടതുണ്ട്. എന്തിനെന്നാൽ പുതിയ നോഡിന്റെ അടയാളം അതിന്റെ ലിങ്ക് ഭാഗത്താണ് സംഭരിക്കേണ്ടത്. Start തോന്നിനും ആദ്യത്തെ നോഡിന്റെ അടയാളം സ്വീകരിച്ചുകൊണ്ടാണ് യാത്രചെയ്യൽ പ്രവർത്തനം ആരംഭിക്കുന്നത്. ഈ അടയാളം താൽക്കാലിക പോയിന്റർ വേരിയബിലിലേക്ക് പകർത്തുകയും (ചിത്രത്തിൽ Temp) പിന്നീട് Temp ചുണ്ടുന്ന നോഡിലെ ലിങ്ക് ഭാഗത്തുള്ള അടയാളം പകർത്തിയെടുത്തു. താൽക്കാലിക പോയിന്റർ വേരിയബിലി പുതുക്കപ്പെടുകയും ചെയ്യുന്നു. Temp ചുണ്ടുന്ന നോഡിന്റെ ലിങ്ക് ഭാഗത്തു നശി വില ലഭിക്കുന്നത് വരെ ഈ സംർഖം തുടർന്ന് കൊണ്ടെതിരിക്കുന്നു.

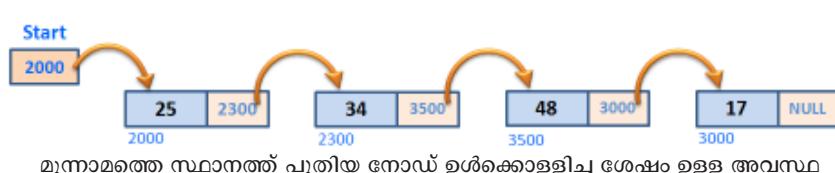
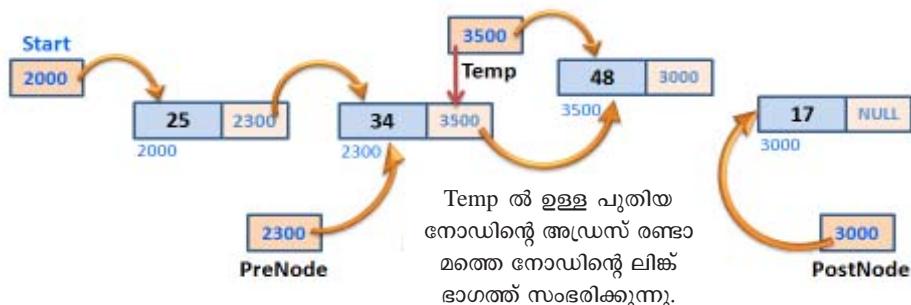
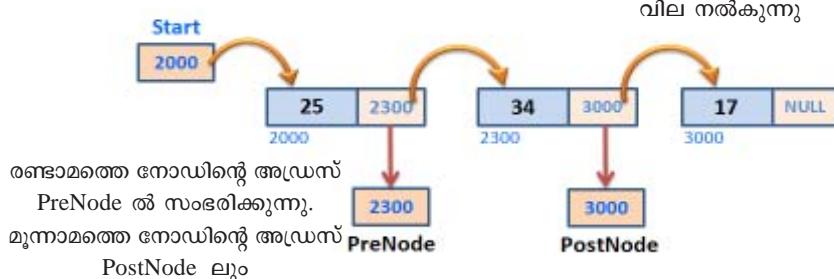
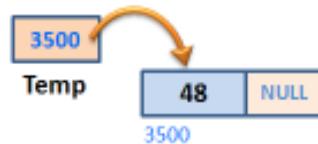
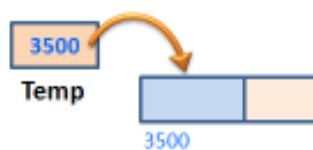
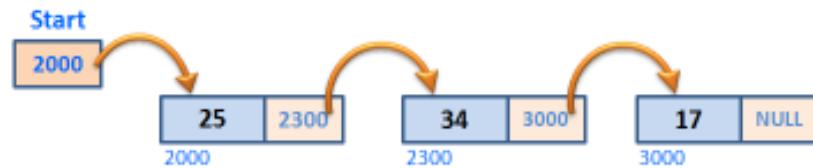
### c. ലിങ്ക്ഡ് ലിസ്റ്റിലെ ഉൾപ്പെടുത്തൽ പ്രവർത്തനം (Insertion in a linked list)

ലിങ്ക്ഡ് ലിസ്റ്റിൽ ഒരു അംഗത്തെ ഉൾപ്പെടുത്തൽ എന്നാൽ ആ അംഗം ഉൾപ്പെടുന്ന നോഡിനെ നിശ്ചിതസ്ഥാനത്തു സ്ഥാപിക്കുന്ന പ്രക്രിയയാണ്. അരെയിലേതു പോലെ



അമുക്ക് ചെയ്യാം

ആരംഭത്തിൽ മുന്ന് നോഡുകളുള്ള ഒരു ലിങ്ക്ഡ് ലിസ്റ്റിന്റെ മുന്നാം സ്ഥാനത്തെക്കു നോഡിനെ ഉൾപ്പെടുത്താനുള്ള പ്രവർത്തനങ്ങളാണ് ചിത്രം 3.18 തോന്നിട്ടിരിക്കുന്നത്. ചിത്രം നിരീക്ഷിച്ചു ലിങ്ക്ഡ് ലിസ്റ്റിലേക്ക് ഒരു നോഡിനെ ഉൾപ്പെടുത്താനുള്ള പ്രവർത്തനത്തിന്റെ ഘട്ടങ്ങൾ നമുക്ക് നിർവചിക്കാം. Node തരത്തിൽപ്പെട്ട പോയിന്ററുകളാണ് Temp, PreNode, PostNode എന്നിവ എന്നിനും POS നോഡിനെ ഉൾപ്പെടുത്തേണ്ട സ്ഥാനം സംഭരിച്ചിരിക്കുന്ന വേരിയബിലി ആണെന്നും നമുക്ക് അനുമാനിക്കാം.



ചിത്രം 3.18: ലിക്കിൾ ലിസ്റ്റിലെ ഉൾപ്പെടുത്തൽ പ്രവർത്തനം

ലിക്ക്‌ഡ ലിസ്റ്റിൽ നോഡിനെ എവിടെ വേണമെങ്കിലും ഉൾപ്പെടുത്താവുന്നതാണ്. തുടക്കത്തിലോ, അവസാനത്തിലോ അല്ലെങ്കിൽ രണ്ടു നോഡുകൾക്കിടയിലോ ആകാം.

ഉൾപ്പെടുത്തൽ പ്രവർത്തനത്തിന് താഴെ പറയുന്ന ഘട്ടങ്ങൾ നിർവ്വചിക്കാം:

- എട്ട് 1: നോഡ് നിർമ്മിച്ച് അതിന്റെ അധ്യസ്ത Temp ത്ത് സംഭരിക്കുന്നു.
- എട്ട് 2: Temp ഉപയോഗിച്ച് ഈ നോഡിന്റെ ഡാറ്റാഭാഗവും ലിക്ക് ഭാഗവും സംഭരിക്കുക.
- എട്ട് 3: യാത്ര ചെയ്ത പ്രവർത്തനത്തിന്റെ സഹായത്തോടെ (POS-1), (POS+1) എന്നീ സ്ഥാനങ്ങളിലുള്ള നോഡുകളുടെ അധ്യസ്ത തമാക്രമം PreNode, PostNode എന്നീ പോയിന്ററുകളിലേക്കു സ്വീകരിക്കുക.
- എട്ട് 4: Temp (POS-1) സ്ഥാനത്തുള്ള (നോഡ്) എൻഡ് link ഭാഗത്തിലേക്കു PreNode (പുതിയ നോഡ്) എൻഡ് അധ്യസ്ത സംഭരിക്കുക.
- എട്ട് 5: Temp (പുതിയ നോഡ്) എൻഡ് link ഭാഗത്തിലേക്കു PostNode (POS+1) സ്ഥാനത്തുള്ള (നോഡ്) എൻഡ് അധ്യസ്ത സംഭരിക്കുക.

#### d. ലിക്ക്‌ഡ ലിസ്റ്റിൽ നിന്നും നീക്കം ചെയ്ത് (Deletion from a linked list)

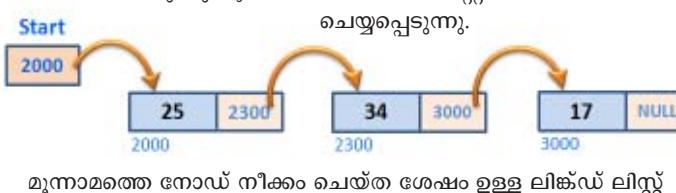
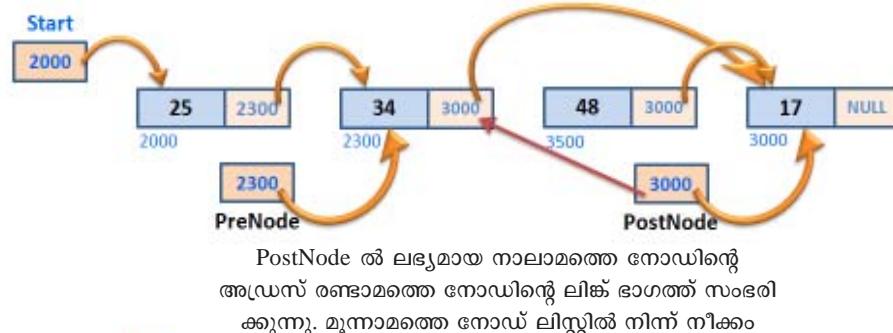
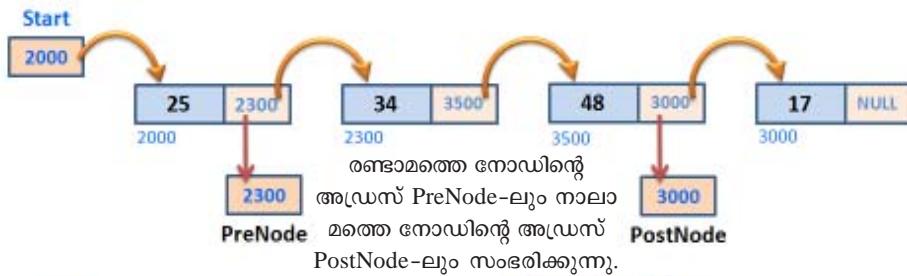
ലിക്ക്‌ഡ ലിസ്റ്റിലുള്ള ഒരു അംഗത്തെ നീക്കം ചെയ്ത് എന്നാൽ ലിസ്റ്റിൽ നിന്നും ഒരു നോഡിനെ ഒഴിവാക്കുന്ന പ്രക്രിയയാണ്. നീക്കം ചെയ്യാനുള്ള നോഡിന്റെ സ്ഥാനം നൽകിയിട്ടുണ്ടായിരിക്കും. അതിനു പകരം ഡാറ്റയാണ് നൽകിയിരിക്കുന്നതെങ്കിൽ, ആ ഡാറ്റ അടങ്കിയ നോഡ് തിരഞ്ഞെടുപാടിക്കുകയും അതിന്റെ സ്ഥാനം കണ്ണഡിത്തം നോഡും അടങ്കിയാണ്. അതിനു ശേഷം നീക്കം ചെയ്ത് പ്രവർത്തനത്തിനുള്ള നടപടിക്രമങ്ങൾ നടപ്പിലാക്കേണ്ടതാണ്. ലിസ്റ്റിലെ ഏതു നോഡിനെയും നീക്കം ചെയ്യുന്നതിന് സാധ്യമാണ്. ഒന്നാമത്തെ നോഡാണ് നീക്കം ചെയ്യേണ്ടതെങ്കിൽ ഒന്നാമത്തെ നോഡിലെ link ലെ അധ്യസ്ത START ലേക്ക് സംഭരിക്കണം. അതുപോലെ അവസാനത്തെ നോഡാണ് നീക്കം ചെയ്യേണ്ടതെങ്കിൽ അവസാനത്തെ നോഡിന്റെ തൊട്ടു മുമ്പുള്ള നോഡിലെ link എൻഡ് വില NULL ആയി സംഭരിക്കണം. ഒരു പ്രത്യേക സ്ഥാനത്തുള്ള നോഡിനെ നീക്കം ചെയ്യുന്നതിന് ആവശ്യമായ നടപടിക്രമങ്ങൾ നമുക്ക് ചർച്ച ചെയ്യാം.



നമുക്ക് ചെയ്യാം

തുടക്കത്തിൽ നാല് നോഡുകളുള്ള ലിക്ക്‌ഡ ലിസ്റ്റിൽ നിന്നും മൂന്നാമത്തെ നോഡിനെ നീക്കം ചെയ്യാനുള്ള നടപടികളാണ് ചിത്രം 3.19 ത്ത് പ്രതിപാദിച്ചിരിക്കുന്നത്. ചിത്രം നിരീക്ഷിച്ച് ലിക്ക്‌ഡ ലിസ്റ്റിൽ നിന്നും ഒരു നോഡിനെ നീക്കം ചെയ്യാനുള്ള പ്രവർത്തനത്തിന്റെ ഘട്ടങ്ങൾ നമുക്ക് നിർവ്വചിക്കാം. Node തരത്തിൽപ്പെട്ട പോയിന്ററുകളാണ് PreNode, PostNode എന്നിവ എന്നും POS നോഡിനെ നീക്കം ചെയ്യേണ്ട സ്ഥാനം സംഭരിച്ചിരിക്കുന്ന വേരിയബിൾ ആണെന്നും നമുക്ക് അനുമാനിക്കാം.

### 3. ഡാറ്റാസ്ട്രക്ചർകളും (പ്രവർത്തനങ്ങളും)



**ചിത്രം 3.19: ലിക്ക് ലിസ്റ്റിലെ നീക്കം ചെയ്ത പ്രവർത്തനം**

നീക്കം ചെയ്ത പ്രവർത്തനത്തെ നിർവ്വചിക്കാൻ താഴെ പറയുന്ന നടപടികൾ നിങ്ങളെ സഹായിക്കുന്നു.

- എടു 1: കടന്നുപോകൽ പ്രവർത്തനത്തിൽ സഹായത്തോടെ (POS-1), (POS+1) എന്നീ സ്ഥാനങ്ങളിലുള്ള നോഡുകളുടെ അധിസ് യഥാക്രമം PreNode, PostNode എന്നീ പോയിറ്റുകളിലേക്കു സ്വീകരിക്കുക.
- എടു 2: PostNode ((POS+1) സ്ഥാനത്തുള്ള നോഡ്) ന്റെ link ഭാഗത്തിലേക്കു PreNode ((POS-1) സ്ഥാനത്തുള്ള നോഡിൽ അധിസ് സംബന്ധിക്കുക.
- എടു 3: POS സ്ഥാനത്തുള്ള നോഡിനെ മെമ്മറിയിൽ നിന്നും ഒഴിവാക്കുക.

ചിത്രം 3.19 തെ സൂചിപ്പിച്ചിരിക്കുന്ന ആദ്യത്തെ റണ്ടു എടുങ്ങൾ പാലിച്ച് മൂന്നാമത്തെ നോഡിനെ രണ്ടാമത്തെത്തിൽ നിന്നും വേർപ്പെടുത്തിയാലും, മൂന്നാമത്തെ നോഡ്

നാലാമത്തെത്തിലേക്കു ചുണ്ടിക്കൊണ്ട് മെമ്മറിയിൽത്തന്നെ അവഗ്രഹിക്കുന്നു. അതിനാൽ പ്രോഗ്രാമിൽ ഭാഷകളിലെ മെമ്മറി സ്വാതന്ത്ര്യമാക്കൽ സംവിധാനങ്ങൾ ഉപയോഗപ്പെടുത്തി ആ നോഡിനെ നീകിനു ചെയ്യണമെന്നതാണ്. അതുപോലെ ലിക്ക് ഡി ലിസ്റ്റുകളിൽ പ്രവർത്തനങ്ങൾ നടത്തുവോൾ പ്രവർത്തനത്തിന് ശേഷം TEMP, PreNode, PostNode മുതലായ താൽക്കാലിക പോയിന്റുകളും സ്വതന്ത്രമാക്കപ്പെടേണ്ടവയാണ്.

### പുരോഗതി ഇതുവരെ



1. ദൈനാനിക് ഡാറ്റാസ്ട്രക്ചർ ഓഫീസിൽ ഒരു ഉദാഹരണം പറയുക.
2. ലിക്ക് ഡി ലിസ്റ്റ് എന്നാലെന്ത്?
3. ഒരു ലിക്ക് ഡി ലിസ്റ്റിന്റെ നോഡിൽ \_\_\_\_\_ ഉം \_\_\_\_\_ ഉം അടങ്ങിയിരിക്കും.
4. ഒരു ലിക്ക് ഡി ലിസ്റ്റിലെ നോഡിനെ നിർവ്വചിക്കാൻ ഉപയോഗിക്കുന്ന പ്രോഗ്രാമിൽ ഭാഷയിലെ സംവിധാനം ഏതാണ്?
5. ഒരു ലിക്ക് ഡി ലിസ്റ്റിൽ Start അല്ലെങ്കിൽ Header എന്ന് ഉള്ളടക്കം ഏതായിരിക്കും?



സ്ഥാക്കും കൂടും ലിക്ക് ഡി ലിസ്റ്റുകൾ ഉപയോഗിച്ചും നിർമ്മിക്കാവുന്നതാണ്. അവയെ ദൈനാനിക് സ്ഥാക്കുകയും കൂടും ലിക്ക് ഡി ലിസ്റ്റുകുന്നു. അടുത്ത നോഡിലേക്കു മാത്രം ചുണ്ടുന്നതാകയാൽ നമ്മൾ ചർച്ച ചെയ്ത ലിക്ക് ഡി ലിസ്റ്റുകളെ സിസ്റ്റെലിക്ക് ലിസ്റ്റ് എന്ന് വിളിക്കുന്നു. എന്നാൽ അടുത്ത നോഡിലേക്കും തൊട്ടു മുമ്പുള്ള നോഡിലേക്കും ചുണ്ടുന്ന ധാരിക്കി ലിക്ക് ഡി ലിസ്റ്റും ഉണ്ട്. സ്വയം സൂചിത്തസ്ട്രക്ചർ റണ്ട് പോയിന്റുകൾ ഉപയോഗിച്ചാണ് ഈ സാധ്യമാക്കുന്നത്. ട്രീ (TREE) പോലുള്ള സ്കൈറ്റ്‌ലൈഡ് ഡാറ്റാസ്ട്രക്ചർകൾ ധാരിക്കി ലിക്ക് ഡി ലിസ്റ്റുകൾ ഉപയോഗിച്ചാണ് നിർമ്മിക്കുന്നത്.



### നമുക്ക് സംഗ്രഹിക്കാം

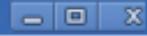
ഡാറ്റാസ്ട്രക്ചറുകൾ എന്ന ആശയത്തെയും അവയുടെ മേൽ നടപ്പിലാക്കാവുന്ന പ്രവർത്തനങ്ങളെയും നമ്മൾ പരിചയപ്പെട്ടി. ഏതു തരത്തിലുള്ള ഡാറ്റയെയും പ്രതിനിധികരിക്കാൻ അനുയോജ്യമായ വ്യത്യസ്തങ്ങളായ ഡാറ്റാസ്ട്രക്ചറുകളുണ്ട്. ഇവയിൽ ഉപയോഗിക്കുന്ന പ്രവർത്തനങ്ങളും അംഗങ്ങളെ ഗണങ്ങളാക്കാൻ ഉപയോഗിക്കുന്ന തത്ത്വത്തിനുസരിച്ചു വ്യത്യാസപ്പെട്ടിരിക്കും. ചില ഡാറ്റാസ്ട്രക്ചറുകൾ യുക്തിപരമായ ആശയങ്ങൾ മാത്രമാണെങ്കിലും അവയുടെ നടപ്പിലാക്കലിനുവിച്ചും നമ്മൾ ചർച്ച ചെയ്തു. അനൈയും ലിക്ക് ഡി ലിസ്റ്റും തമ്മിലുള്ള വ്യത്യാസങ്ങളുള്ളപ്പറ്റിയും ചർച്ചയിൽ പരാമർശിച്ചിരുന്നു. കമ്പ്യൂട്ടർ സയൻസിന്റെ ഉപരിപഠനത്തിനു ഇതു അധ്യായത്തിൽ പറഞ്ഞ ആശയങ്ങൾ വളരെ വേണ്ടപ്പെട്ടവയാണ്.

## നമുക്ക് പരിശോധിക്കാം

1. താഴെ പറയുന്ന പ്രസ്താവനകൾ വായിക്കുക:
  - ( i ) ഒറ്റ ഐടകമായി ഫ്രോസൻ ചെയ്യപ്പെടുന്ന ഡാറ്റയുടെ ശേഖരം.
  - ( ii ) അരെ ഉപയോഗിച്ചാണ് എല്ലാ ഡാറ്റാസ്ട്രക്ചറുകളും പ്രാവർത്തികമാക്കിയിരിക്കുന്നത്.
  - ( iii ) സ്റ്റാക്ക്, ക്യൂ എന്നിവ യുക്തിസഹമായ ആശയങ്ങളും അരെ, ലിക്ക് ലിസ്റ്റ് എന്നിവ ഉപയോഗിച്ച് പ്രാവർത്തികമാക്കിയവയുമാണ്.
  - ( iv ) സ്റ്റാറ്റിക് ഡാറ്റാസ്ട്രക്ചറുകളുടെ കാര്യത്തിൽ ഓവർഹെഡോ സംഭവിക്കുന്നു.

മുകളിൽ കൊടുത്തിരിക്കുന്ന പ്രസ്താവനകളിൽ എതാൻ ശരിയായിട്ടുള്ളത്? എറ്റവും ഉചിതമായത് തിരഞ്ഞെടുക്കുക.

  - a. പ്രസ്താവനകൾ (i), (iii) മാത്രം
  - b. പ്രസ്താവനകൾ (i), (ii), (iii) മാത്രം
  - c. പ്രസ്താവനകൾ (i), (iii), (iv) മാത്രം
  - d. പ്രസ്താവനകൾ (i), (ii), (iv) മാത്രം
2. ഡാറ്റാസ്ട്രക്ചറുകൾ സ്റ്റാറ്റിക്കോ ഡൈനാമിക്കോ ആവാം.
  - a. സ്റ്റാറ്റിക് ഡാറ്റാസ്ട്രക്ചറുകൾക്കു രണ്ടു ഉദാഹരണം നൽകുക.
  - b. സ്റ്റാറ്റിക് ഡാറ്റാസ്ട്രക്ചറുകൾ ഓവർ ഹെഡോ സാഹചര്യം നേരിട്ടേണ്ടി വരാം. എന്തുകൊണ്ട്?
  - c. ലിക്ക് ലിസ്റ്റ് ഒരു ഡൈനാമിക് ഡാറ്റാസ്ട്രക്ചർ ആണ്. ഈ പ്രസ്താവനയെ ന്യായീകരിക്കുക.
3. സ്റ്റാക്കിലേക്കു ഒരു അംഗത്വത്തെ ഉൾപ്പെടുത്താനുള്ള അൽഗോറിതം എഴുതുക.
4. പുഷ്ട്, പോസ്റ്റ് പ്രവർത്തനങ്ങൾ എന്നാലെന്ത്?
5. സ്റ്റാക്കിൽ നിന്ന് ഒരു അംഗത്വത്തെ നീക്കം ചെയ്യാനുള്ള അൽഗോറിതം എഴുതുക.
6. രേഖീയക്കൃവിൽ ഉൾപ്പെടുത്തൽ, നീക്കംചെയ്തൽ പ്രവർത്തനങ്ങൾ പ്രാവർത്തികമാക്കാനുള്ള അൽഗോറിതം എഴുതുക.
7. രേഖീയക്കൃവിശ്രേഷ്ഠ പരിമിതികൾ വൃത്താകൃതി ക്കു എങ്ങനെ മറികടക്കുന്നു.
8. ഡാറ്റാസ്ട്രക്ചറുകളിൽ നടപ്പാക്കാവുന്ന ചില പ്രവർത്തനങ്ങൾ താഴെ പറയുന്നു
  - i. ഉൾപ്പെടുത്തൽ ii. നീക്കംചെയ്തൽ iii. തിരയൽ iv. കേമപ്പെടുത്തൽ
  - a. ഇവയിൽ എത്ര പ്രവർത്തനത്തിനാണ് അംഗൾ ഹെഡോ സാഹചര്യം നേരിട്ടേണ്ടി വരിക.
  - b. ഈ സാഹചര്യം ഉചിതമായ ഡാറ്റാസ്ട്രക്ചറിശ്രേഷ്ഠ പശ്ചാത്തലവത്തിൽ വിവരിക്കുക.



## 9 . ചേരുംപടി ചേർക്കുക

- | <b>A</b>             | <b>B</b>     | <b>C</b>   |
|----------------------|--------------|--|
| a. അരെ               | i. റ്ലാർട്ട് | 1 . വ്യത്യസ്ത ഭാഗങ്ങളിൽ ഉൾപ്പെടുത്തലും<br>നികം ചെയ്യലും സംഭവിക്കുന്നു. |
| b. റ്ലാസ്സ്          | ii. സുചിക    | 2 . ഒരേ ഭാഗത്തു തന്ന ഉൾപ്പെടുത്തലും<br>നികം ചെയ്യലും സംഭവിക്കുന്നു     |
| c. ക്യാ?             | iii. റിയർ    | 3 . സ്വയം സുചിത സ്ട്രക്ടറുകൾ ഉപയോ<br>ഗപ്പെടുത്തുന്നു                   |
| d. ലിക്ഷ്യ് ലിസ്റ്റ് | iv. ടോപ്പ്   | 4 . അംഗത്വിന്റെ സ്ഥാനം ഉപയോഗി<br>ച്ചിടാണ് അതിനെ ആക്സസ് ചെയ്യുന്നത്     |

10 . അരെ കേന്ദ്രീകൃത ഡാറ്റാസ്ട്രക്ചറുകളുടെ പോലെ ലിക്ഷ്യ് ലിസ്റ്റുകൾക്ക് ഓവർ പ്രൈം സാഹചര്യം നേരിട്ടേണ്ടി വരാത്തത് എന്തുകൊണ്ടുണ്ട് വിശദീകരിക്കുക.



## പ്രധാന പഠനനേട്ടങ്ങൾ

പരിതാവ് ആർജിക്കേണ്ട ശേഷികൾ

- സുരക്ഷിത ആശയവിനിമയം വൈബ്‌പോജുകളിലൂടെ സാധ്യമാകുന്നതെങ്ങനെ എന്ന് വിശദീകരിക്കുന്നു.
- വൈബ് സെർവീസിൽനിന്നും ഉപയോഗവും വൈബ്‌ഹോസ്റ്റിംഗിനെക്കുറിച്ചുള്ള അറിവും വിവരിക്കുന്നു.
- സൗഖ്യിക, ദൈനനാമിക് എന്നീ വൈബ്‌പോജുകൾ താരതമ്യം ചെയ്യുന്നു.
- ഫ്രോഗ്രാമിംഗ് ഭാഷകളും സ്ക്രിപ്റ്റുകളും തമിലുള്ള വ്യത്യാസം തിരിച്ചറിയുന്നു.
- വിവിധ സ്കെഡ്യൂലിംഗ് ഭാഷകൾ താരതമ്യം ചെയ്യുന്നു.
- കാസ്കേഡിംഗ് റെസ്റ്റ് ഷൈറ്റിംഗ് ആവശ്യകത വിശദീകരിക്കുന്നു.
- ലഭിതമായ HTML എലമെൻ്റുകൾ ഉപയോഗിച്ചുള്ള വൈബ്‌പോജ് നിർമ്മാണം മനസ്സിലാക്കുന്നു.
- അടിസ്ഥാന HTML ടാഗുകളും അവയുടെ പ്രധാന ആട്ടിബ്യൂട്ടുകളും പട്ടികപ്പെടുത്തുന്നു.
- HTML ടാഗുകളെ തരം തിരിക്കുന്നു.
- HTML പേജുകൾ ആകർഷകമാക്കുന്നതിന് ഫോർമാറ്റിംഗ് ടാഗുകൾ സാമർലോചിതമായി HTML ഡോക്യുമെന്റുകളിൽ ഉപയോഗിക്കുന്നത് എങ്ങനെ എന്ന് മനസ്സിലാക്കുന്നു.
- ഫോർമാറ്റിംഗ് ടാഗുകൾ തമിലുള്ള സാമ്പ്രദായിക വ്യത്യാസങ്ങളും തിരിച്ചറിയുന്നു.
- <PRE>, <DIV> എന്നീ ടാഗുകളുടെ ഉപയോഗം നിരീക്ഷിക്കുന്നു.
- ഒരു ഡോക്യുമെന്റിലെ ഉള്ളടക്കമേം മറ്റ് ഓൺലൈൻ കൂക്കളോ (വസ്തുക്കളോ) ചലിപ്പിക്കുന്നതിനുള്ള ടാഗ് തിരിച്ചറിയുന്നു.
- <FONT> ടാഗ് ഉപയോഗിച്ച് ടെക്സ്റ്റ് ഉള്ളടക്കം ഫലപ്രദമായി രൂപപ്പെടുത്തുന്നു.
- HTML ഡോക്യുമെന്റുകളിൽ കമ്പ്യൂട്ടുകൾ ഉപയോഗിക്കുന്നു.
- <IMG> ടാഗ് ഉപയോഗിച്ച് HTML ഡോക്യുമെന്റുകളിൽ ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുന്നു.

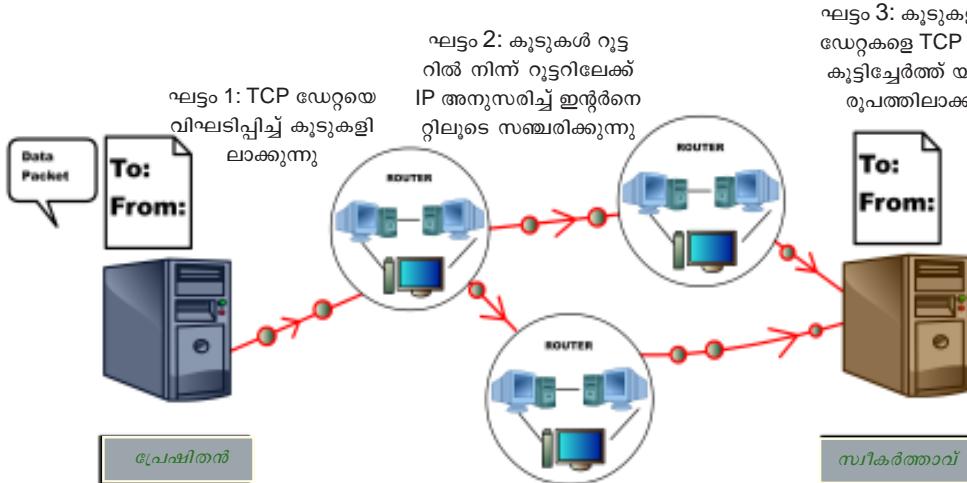
ഇള്ളർന്നെന്ന് യുഗത്തിലാണ് നാമെല്ലാം ജീവിക്കുന്നത്. വിവരങ്ങൾ ആവശ്യമായി വരുന്നോൾ നാം ആദ്യം തിരയുന്നത് ഇള്ളർന്നെറ്റിലായിരിക്കും. 11-ാം ക്ലാസിലെ പരീക്ഷാഫലം വൈബ് സെറ്റിൽ തിരഞ്ഞെടുത്ത നിങ്ങൾ കണ്ടിട്ടുണ്ട്. ഓൺലൈൻ അപേക്ഷ സമർപ്പണം, വൈബ് സെറ്റിലുള്ള കാര്യങ്ങൾ പരിശോധിക്കൽ, സിനിമ കാണൽ, ധനവ്യാപാരസ്ഥാപനങ്ങളിലെ ക്രയവിക്രയം, ഓൺലൈൻ ക്രയവിക്രയത്തിലുടെ സാധനങ്ങൾ വാങ്ങൽ തുടങ്ങി വിവിധ ആവശ്യങ്ങൾക്കായി ധാരാളം വൈബ് സെറ്റുകളെ നാം ആശ്രയിക്കാറുണ്ട്. വൈബ് പേജുകളുടെ ഒരു കൂട്ടമാണ് വൈബ് സെറ്റ് എന്ന് നമുക്കറിയാം. ചിത്രങ്ങൾ (അക്ഷരങ്ങളോ അല്ലാതെ വരവയോ ആയത്), ചിത്രങ്ങൾ, ശാപ്പുകൾ, ചലനാമകൾച്ചിത്രങ്ങൾ, ശബ്ദവ വീഡിയോകൾ തുടങ്ങിയവ വൈബ് പേജുകളിലുണ്ടാകാം. സ്ഥാപനങ്ങൾ, ഉൽപ്പന്നങ്ങൾ, സേവനങ്ങൾ തുടങ്ങിയ വരയക്കുറിച്ചുള്ള വിവരം പ്രദാനം ചെയ്യാനാണ് ധാരാളം പേജുകളുള്ള ഒരു വൈബ് സെറ്റ് രൂപകൾ തിലുന്ന ചെയ്യുന്നത്. ഈ വൈബ് സെറ്റ് ഇള്ളർന്നെന്നെന്ന് നിന്നും നമുക്ക് എങ്ങനെ ലഭിക്കുന്നു? ഇള്ളർന്നെറ്റുമായി ബന്ധിപ്പിച്ചിട്ടുള്ള വൈബ് സെറ്റവറുകളിൽ ഈ പേജുകൾ മറ്റൊള്ളവർക്ക് ലഭിക്കത്തെത്തവിധി സംഭവിച്ച് വച്ചിരിക്കുന്നു. ഈ അധ്യായത്തിൽ ഇള്ളർന്നെന്ന് വഴിയുള്ള ആശയ വിനിമയവും വൈബ് സെറ്റവറുകൾക്ക് അതിലുള്ള പ്രാധാന്യവും അവലോകനം ചെയ്യുന്നു. വൈബ് സെറ്റുകൾ വികസിപ്പിക്കുന്നതിനാവശ്യമായ

വിവിധ ടുളുകളെക്കുറിച്ചും, സാങ്കേതികതകളെക്കുറിച്ചും ഇവിടെ പ്രതിപാദിക്കുന്നുണ്ട്. പരിവർത്തനാത്മക (dynamic) വെബ് പേജുകളുടെ ആശയവും അവ പരിവർത്തനാ തമകം ആക്കുന്നതിന് ഉപയോഗിക്കുന്ന സ്ക്രീനിൽ ഭാഷകളുടെ അവലോകനവും ഇവിടെ ചർച്ചചെയ്യുന്നു. ഹൈപ്പർ ടെക്നോളജീസ് മാർക്കറ്റ് ഭാഷ ഉപയോഗിച്ചാണ് വെബ് പേജുകൾ നിർമ്മിക്കുന്നത്. ഇൽസ്റ്ററേറ്റീരേഖ ഭാഷയായി HTML അറിയപ്പെടുന്നു. ഉള്ളടക്കം ബേഹസർ വിന്റോടിൽ എങ്ങനെ പ്രദർശിപ്പിക്കണമെന്നാണ് HTML ബേഹസറിനോട് പറയുന്നത്. HTML ഉപയോഗിച്ചുള്ള വെബ് പേജ് നിർമ്മിതിയുടെ അടിസ്ഥാന തത്വങ്ങൾ ഈ അധ്യായത്തിൽ നമുക്ക് പരിചയപ്പെട്ടാം.

#### **4.1 വെബ്മീലുടെയുള്ള ആശയ വിനിമയം (Communication on the web)**

ഇൽസ്റ്ററേറ്റിലുള്ള ഒരു വെബ് സൈറ്റ് വെബ് ബേഹസർ (ക്ലയർ) പരിശോധിക്കുന്നേം എന്നതാക്കെ കാര്യങ്ങൾ നടക്കുന്നുവെന്ന് 11-ാം ക്ലാസ്സിൽ നാം പറിച്ചതാണ്. അനുബന്ധ ഇൽസ്റ്ററേറ്റ് പ്രോട്ടോക്കോൾ (IP) മേൽവിലാസത്തിനായി URL (യൂണിഫോം റിസോഴ്സ് ലോക്കേറ്റർ)-നെ DNS (ഡോമേണിനും സൈറ്റിനും സിസ്റ്റം) സൈർവർിലേയ്ക്ക് അയയ്ക്കുകയും തുടർന്ന് ലഭിക്കുന്ന IP മേൽവിലാസം ഉപയോഗിച്ച് ബേഹസർ സൈർവറുമായി ബന്ധം സ്ഥാപിക്കുകയും ചെയ്യുന്നു. സൈർവർ-കമ്പ്യൂട്ടറിലുള്ള സൈർവർ-സോഫ്റ്റ്‌വെയർ ഈ ആവശ്യം പ്രോസസ് ചെയ്ത് കാണിക്കേണ്ട വെബ് പേജിനെ ക്ലയർിന് അയച്ച് കൊടുക്കുന്നു. ക്ലയർ ബേഹസർ വെബ് പേജ് ക്രമമായി കാണിക്കുന്നു.

വെബ് ഉപയോഗിച്ചുള്ള ആശയവിനിമയത്തിന്, കമ്പ്യൂട്ടറുകൾ അല്ലക്കിൽ ഉപകരണ അഞ്ച് പരസ്പരം തിരിച്ചറിയേണ്ടതുണ്ട്. ട്രാൻസ്മിഷൻ കൺട്രോൾ പ്രോട്ടോക്കോൾ/ ഇൽസ്റ്ററേറ്റ് പ്രോട്ടോക്കോൾ (TCP/IP) എന്ന ഒരേ പ്രോട്ടോക്കോൾ തന്നെ എല്ലാ ഉപകരണങ്ങളും അനുസരിക്കുന്നത് കൊണ്ടാണ് ഈത് സാധ്യമാകുന്നത്. TCP/IP പ്രോട്ടോക്കോ ഇളക്കളെക്കുറിച്ചും അവയുടെ പ്രവർത്തനത്തെക്കുറിച്ചും 11-ാം ക്ലാസ്സിലെ കമ്പ്യൂട്ടർ ശൃംഖല (കമ്പ്യൂട്ടർ നേര്ദ്ദവർക്ക്) എന്ന അധ്യായത്തിൽ നാം വിശദമായി ചർച്ച ചെയ്തിട്ടുണ്ട്. അയയ്ക്കേണ്ട ഡാറ്റയെ, TCP/IP പ്രോട്ടോക്കോൾ വിൾഡപ്പിച്ച് ചെറിയ കുടുകളിലാക്കി സീകർത്താവിരെ കമ്പ്യൂട്ടർ വിലാസത്തോടൊപ്പം ചേർക്കുന്നു. ഇൽസ്റ്ററേറ്റ് പ്രോട്ടോക്കോളിന്റെ സഹായത്താൽ, റൂട്ടറുകൾ (വഴികാട്ടുന്ന ഉപകരണങ്ങൾ) ഈ ഡാറ്റ ഉൾക്കൊള്ളുന്ന കുടുകളെ അവ എത്തേണ്ട കമ്പ്യൂട്ടറിലെത്തിക്കുന്നു. അയക്കുന്നയാൾ (പ്രോഫിൽ) ഡോറ്റായുടെ ഒരു കുടിനെ സീകർത്താവിലേക്ക് വഴികാട്ടി എത്തിക്കുന്നത് എങ്ങനെ എന്ന് എന്ന് ചിത്രം 4.1 തുടർന്നിരിക്കുന്നു.

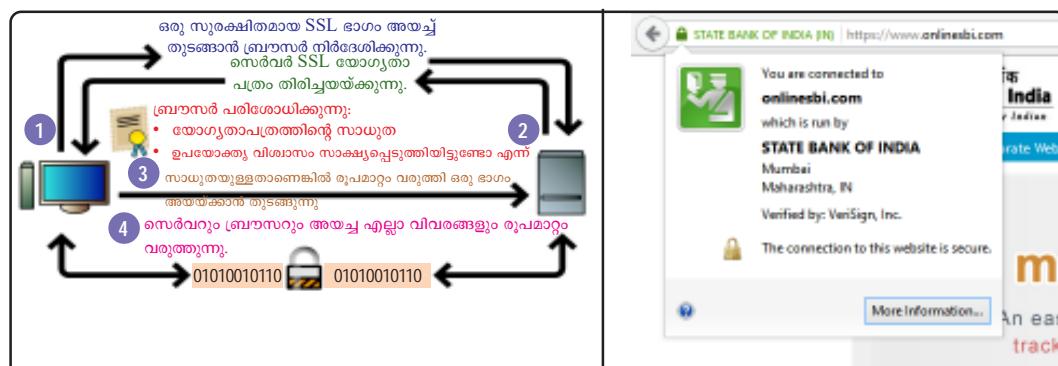


**ചിത്രം 4.1:** സാറ്റ് ഉൾക്കൊള്ളുന്ന കൃടുകൾ അയക്കുന്നയാളിൽ നിന്ന് സ്വീകർത്താവിലേക്ക് വഴികടക്കാൻ

வெவ்வே சௌர் பறிஶோயிக்கல், இ-மெதித் அயற்கலை துடങ்கிய பலதரத்திலும் ஆஶயவினிமயண்டு ஹஸ்ரெந்றில் நடக்கும். வெவ்வே சௌரூப்யக்கல் வேள்கிழுத்து திரய்த் HTTP (ரைப்பு எக்ஸ்பிளிக்ஷன் ப்ரோடோகோஸ்) உபயோகிப்பு இ-மெதித் வசியுத்து ஆஶயவினிமயம் SMTP (ஸினிஸ் மெதித் டொஸ்ஸ்பார் ப்ரோடோகோஸ்) உபயோகிப்புமான் நடக்கும்தென் நாம் பரிசீலிக்கும். தாഴெதலத்திலும் ஹஸ்ரெந்ற் ப்ரோடோகோலிலே ஆர்வும் முக்கியமான ஒரு ரெட் ப்ரோடோகோலுக்கும் பிரதிக்குமானத். ஹஸ்ரெந்ற் வசியுத்து ஆஶயவினிமயம் லஜிதமாக்குமாத் ஹஸ்ரெந்ற் ப்ரோடோகோலான். இந் ப்ரோடோகோஸ் தென் உபயோகிக்கும்துகொள்க் யாராஜம் நேட்டங்களும் என்க. விவியதரம் யேற் கைகாரும் செறுப்புகளினால் ரூப்ரூப்கள் பிரதேகம் போராம் செய்யுள்ளிவருளில். கொள்கூபோகேள் யார்யெக்குவிப்பு ஶஹிகாரத் அவு உர்க்கொல்லும் கூடுக்கல் ஏதுமிகேளை மேற்விலாஸ்தெக்குவிப்பு மாதும் ஶஹிப்பால் மதியாகும். கூடுக்கலிலை யார் உர்க்கொல்லும் காரம் ஸுதாருமாயதினால், புதிய போடோகோலுக்கல் ரூப்பெட்டுத்தி ஹஸ்ரெந்றில் உபயோகிக்குவாருத்து ஸுதாருத்து நமுக்கள் லலி கூனு. TCP/IP யூட் இந் ஸுதாருத்தையும் வாச்சவுமான், ஸங்காரங்கள் கைகாரும் செறுப்புகளினுத்து ஸமுதாயமாயும் சௌரூப்கள், ஶவ்வால், வீயிலேய துடங்கியவு உத்தக்க ணத்தாயி வரும் சௌரூப்கள், ஸுரக்ஷிதப்பனமிடபாடுக்கள்க்காலி யநகாருஸ்மாபன ணால் உபயோகிக்கும் சௌரூப்கள் துடங்கியவத்தையில் போடோகோலுக்கல் நிர்மிக்கும்தெனிக் ஸஹாயிப்புத். ஹஸ்ரெந்றிலே ஸாப்பத்திக் விஜயத்தெனிக் காரளம் இதாயிரும். வெவ்வே வசியுத்து ஆஶயவினிமயம் கூட்டுப்பு பொருள்களில் நின்க் வெவ்வே சௌரவிலே கைநூல், வெவ்வே சௌரவிலே நின்க் வெவ்வே சௌரவிலேலைக்கைநூல், ரெட்டாயி தரங்கிரி சீரிக்கைநூல். ஹஸ்ரெந்ற் வசியுத்து ஆஶய வினிமயத்தெனிக் அதிகாரிக்கத்தையும் ஸுரக்ஷி தத்தையும் அதூரவஸ்மான். வெவ்விலுத்து ஏரு கப்பூட்டு/சௌரவர் யமாற்ம அவகாசி தாய் கப்பூட்டுவன்க் கூட்டுப்பு வருத்தும் பிரகியத்தையில் அதிகாரிக்கத் தூப்பிக்கல் (authentication). ஹஸ்ரெந்ற் வசியுத்து ஆஶயவினிமயத்தெனிக் ஸுரக்ஷிதத்தை லலிக்கும் தெனால், கப்பூட்டுவிலை விவரங்கள் சோத்துமானவர்க்க (ஹாகேஷ்ஸிக்) ஸங்காரங்கள் பிகிசெட்டுக்கூவானோ, ரூப்மார்ட் வருத்துவானோ ஸாயிக்கூக்குமிடி.

#### 4.1.1 കൂട്ടിൽ നിന്നും വെബ് സെർവിലേക്കുള്ള ആശയവിനിമയം (Client to web server communication)

കൂട്ടിൽ നിന്ന് വെബ് സെർവിലേക്കുള്ള ആശയവിനിമയത്തിന് സാധാരണഗതിയിൽ ആധികാരികത ഉറപ്പിക്കേണ്ടതില്ല. എന്നാൽ, ധനകാര്യസ്ഥാപനങ്ങളുടെ ഉപയോഗത്തിനോ, ഇ-മെയിൽ സേവനത്തിനോ ആണെങ്കിൽ യുണിറ്റ് നാമവും (യുണിറ്റ് നെയിം) പാസ് വേഡ്യും സെർവിലേക്ക് അയക്കേണ്ടതുണ്ട്. സുരക്ഷിതത്തകാരണങ്ങളാൽ, ഈ വിവരങ്ങൾ സാധാരണ ചിഹ്നങ്ങളായി/അക്ഷരങ്ങളായി സെർവിലേക്ക് അയയ്ക്കാറില്ല. അമീവാ അങ്ങനെ ചെയ്താൽ നൃഥത്ത് കയറ്റക്കാർക്ക് (ഹാക്കേഴ്സിന്) യുണിറ്റ് നാമവും പാസ്‌വേഡ്യും മോഷ്ടിക്കാനാകും. അതുകൊം സാഹചര്യങ്ങളിൽ യുണിറ്റ് നാമവും പാസ്‌വേഡ്യും രൂപമാറ്റം വരുത്തി (എൻക്രിപ്റ്റ് ചെയ്ത്) സെർവിലേക്ക് അയക്കാനായി നാം HTTPS (ഹൈപ്പർ ടെക്നോളജിക്സ് ടെക്നോളജി സെക്യൂറിറ്റി) എന്ന സാങ്കേതിക വിദ്യ ഉപയോഗിക്കുന്നു. ഈർക്കന്നറിലൂള്ള കമ്പ്യൂട്ടറുകൾ തമ്മിൽ ഈത്തരത്തിൽ ബന്ധം സ്ഥാപിക്കാൻ ഉതകുന്ന സുരക്ഷിത സാങ്കേതിക വിദ്യയായ SSL (സെക്യൂറിറ്റി സേവകൾ ലൈറ്റ്) ഉപയോഗിച്ച് ആൺ HTTPS പ്രവർത്തിക്കുന്നത്. HTTP യ്ക്ക് സുരക്ഷിതത്താം പ്രദാനം ചെയ്യുന്നത് HTTPS ആണ്. SSL പ്രോട്ടോക്കോൾ സംകാരത്തു ഉറപ്പ് വരുത്തുന്നതോടൊപ്പം മറ്റാരു വെബ് സെറ്റിനും ഉപയോകതാവിരുന്ന് ലോഗിൻ അക്കൗണ്ടിൽ ആർമ്മാറാട്ടം നടത്താനോ അയച്ച വിവരം മാറ്റം വരുത്താനോ സാധിക്കാതെ തന്നെ സുരക്ഷ ഉറപ്പ് വരുത്തുകയും ചെയ്യുന്നു.



ചിത്രം 4.2(a) : കൂട്ടിൽ-സെർവർ ആധികാരികത ഉറപ്പിക്കൽ പ്രവർത്തനം

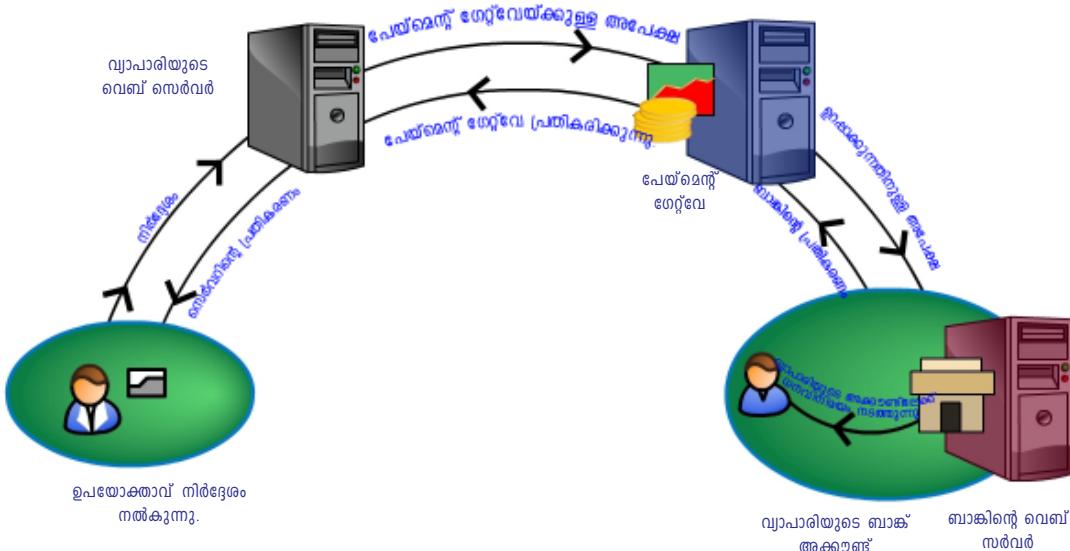
ചിത്രം 4.2(b): SBI വെബ് സെറ്റിനും ആധികാരികത ഉറപ്പിക്കൽ

ബൗണർ ഒരു സുരക്ഷിത പേജ് ആവശ്യപ്പെടുന്നോൾ, സെർവർ ആദ്യം അതിരുന്ന് SSL യോഗ്യതാപ്രതം തിരിച്ച് നൽകുന്നു. തുടർന്ന് ഈ സാക്ഷ്യപ്രതം ബന്ധപ്പെട്ട സാക്ഷ്യ പ്രത അധികാരികളുമായി ഒത്തുനോക്കി സാധുത ഉറപ്പ് വരുത്തുന്നു. സാക്ഷ്യപ്രത അധികാരിക്കു നൽകിയ യോഗ്യതാപ്രതം സാധുത ഉള്ളതാണെങ്കിൽ, അധികാരി സാക്ഷ്യ പ്പെടുത്തുന്നു. ഒരു പ്രത്യേക വെബ് സെറ്റ് അത് അവകാശപ്പെടുന്ന സ്ഥാപനത്തിന്റെ തന്നെ ആണെന്ന് ഉറപ്പിക്കലാണ് ഇത്. വെൽസെസൽ ഇൻകോർപ്പറേഷൻ ഒരു സാക്ഷ്യ പ്രത അധികാരിയാണ്. ഒരു SSL സാക്ഷ്യപ്രതം സാധുത ഉള്ളതായാൽ, ബൗണർ രൂപമാറ്റം വരുത്തിയുള്ള ആശയ വിനിമയത്തിന് വേണ്ടി ഒരു നിശ്ചിത സമയം തുടങ്ങിവ

യക്കുന്നു. ഈ പ്രവർത്തനം ചിത്രം 4.2 (a)-യിൽ ചിത്രീകരിച്ചിരിക്കുന്നു. ഇന്ത്യയിൽ ധനകാര്യസ്ഥാപനങ്ങളുമായുള്ള ഇടപാടുകൾക്ക് ഉപയോഗിക്കുന്ന വെബ് സൈറ്റുകൾ, കൂട്ടയള്ളുകളിൽ നിന്നും രഹസ്യവിവരങ്ങൾ സ്വീകരിക്കുവാൻ HTTPS ഉപയോഗിക്കണമെന്ന് വിവര സാങ്കേതികവിദ്യനിയമം നിഷ്കർശിച്ചിട്ടുണ്ട്. നിങ്ങൾ അധികം ബാറിലുള്ള ലോകിൽ കൂടിക്ക് ചെയ്താൽ ചിത്രം 4.2 (b)-യിൽ തന്നെ സാക്ഷ്യപത്രം കാണാവുന്നതാണ്.

#### **4.1.2 വെബ് സെർവിസ് നിന്നും വെബ് സെർവിസിലേക്കുള്ള ആദ്യ തവിനിമയം (Web server to web server communication)**

ചില വെബ്സ് ആപ്ലിക്കേഷൻകളുടെ പ്രവർത്തനത്തിന് വെബ്സ് സെർവ്വറിൽ നിന്നും വെബ്സ് സെർവ്വറിലേക്കുള്ള ആഗ്രഹവിനിമയം ആവശ്യമാണ്. ഉദാഹരണമായി, ഇൻറർനെറ്റ് വഴി സാധനങ്ങൾ വാങ്ങാൻ ഉപയോഗിക്കുന്ന ഒരു വെബ്സ് സെസ്റ്റ് സൂക്ഷിച്ചിരിക്കുന്ന വെബ്സ് സെർവ്വറിന് (ഇൻറർനെറ്റിലെ വ്യാപാരികൾ/വിൽപ്പനക്കാരൻ), ധനകാര്യസ്ഥാപന തത്ത്വജ്ഞൻ വെബ്സ് സെർവ്വറിലേക്കും അതുപോലെ തിരിച്ചും രഹസ്യ വിവരങ്ങൾ അയയ്ക്കേണ്ടി വരുന്നു. അതുകൊം സന്ദർഭങ്ങളിൽ വ്യാപാരിയുടെയും ധനകാര്യ സ്ഥാപനത്തിന്റെയും വെബ്സ് സെർവ്വറുകളുടെ ആധികാരിക്കതു ഉറപ്പിക്കേണ്ടതുണ്ട്. സ്പീകർക്കുപെട്ട ഡാറ്റ യഥാർത്ഥ സെർവ്വറുകളിൽ നിന്നുള്ളവയാണോ എന്നുറപ്പിക്കാൻ ഡിജിറ്റൽ സാക്ഷ്യം



**ചിത്രം 4.3 :** ഒരു മുതൽ മറ്റ് അറ്റം വരെയുള്ള ഓൺലൈൻ പ്രവർത്തനം

പത്രങ്ങൾ സഹായിക്കുന്നു. ഒരിക്കൽ സെർവീസുകളുടെ ആധികാരിക്കത് ഉറപ്പിച്ചു കഴി നേതാൽ, അവ രൂപമാറ്റം വരുത്തിയ (എൻകീപ്പർ) ഡാറ്റ ഉപയോഗിച്ച് ആശയവിനിമയം നടത്തുന്നു. ഇല്ലാതന്നു വഴി പണം അടയ്ക്കുകയോ/കൈമാറുകയോ ചെയ്യുന്നോൾ, വ്യാപാരിയുടെ സെർവീസും ധനകാര്യസ്ഥാപനത്തിൽ സെർവീസും ഇടയ്ക്ക് ഒരു പേരുമെല്ലാം ഗേറ്റ്‌വേ സെർവർ പ്രവർത്തിച്ച് രൂപമാറ്റം വരുത്തി ധനവിനിമയം നടത്തുന്നു. ഈ പ്രവർത്തനം ചിത്രം 4.3-ൽ കാണിച്ചിരിക്കുന്നു.

## 4.2 വെബ് സെർവർ സാങ്കേതിക വിദ്യ (Web server technologies)

ഹയർസൈക്കൾസ്റ്റി ഡയറക്ടറോൾസ്റ്റി (DHSE-ഡയറക്ടറോൾ ഓഫ് ഹയർ സൈക്കൾസ്റ്റി എയ്യു കേഷർ) ഔദ്യോഗിക വെബ് സൈറ്റായ [www.dhsekerala.gov.in](http://www.dhsekerala.gov.in) സബർശിക്കുന്നേം സംഭവിക്കുന്ത് എന്തൊക്കെയെന്ന് നമുക്ക് നോക്കാം. ആദ്യമായി വെബ് സൈറ്റിലേ പ്രോം പേജ് (ഒരു വെബ് സൈറ്റ് പരിശോധിക്കുന്നേം കാണിക്കുന്ന ആദ്യത്തെ പേജ്), DHSE യുടെ സെർവർിൽ നിന്നും നമ്മുടെ (കൂട്ടാളിലേ) കമ്പ്യൂട്ടറിലേക്ക് കൈമാറ്റം ചെയ്യുന്നു. സാധാരണയായി, വെബ് പേജിന് അഭ്യർത്ഥന നടത്തുന്ന ഒരു കമ്പ്യൂട്ടറോ മൊബൈൽ ഉപകരണമോ ആയിരിക്കും കൂട്ടാളി. വെബ് സെർവർിൽ ശേഖരിച്ചു വച്ചിരിക്കുന്ന വെബ് സൈറ്റിലെ പേജുകൾ, കൂട്ടാളിലേ അപേക്ഷയ്ക്കനുസരിച്ച്, സെർവർ സോഫ്റ്റ്‌വെയർ വിട്ടു നൽകുന്നു. വെബ് സെർവർിലേ സവിശേഷതകളുണ്ടായാണ് അവ ക്രമീകരിക്കുവാൻ ഉപയോഗിക്കുന്ന വസ്തുതകളുണ്ടായാണ് നമുക്ക് ചർച്ച ചെയ്യാം.

### 4.2.1 വെബ് സെർവർ (Web server)

വെബ് സൈറ്റുകളുള്ള സെർവർ കമ്പ്യൂട്ടർ റൂകളെ സുചിപ്പിക്കാനാണ് പലപ്പോഴും വെബ് സെർവർ എന്ന പദം ഉപയോഗിക്കുന്നത്. ഒരു സെർവർ കമ്പ്യൂട്ടറിനെ വെബ് സെർവറാക്കാനായി അതിൽ സ്ഥാപിച്ചിട്ടുള്ള വെബ് സെർവർ സോഫ്റ്റ്‌വെയറിനെന്നും ഈ പദം ഉപയോഗിച്ച് സുചിപ്പിക്കാറുണ്ട്. വെബ് സെർവർ കമ്പ്യൂട്ടറിൽനിന്നും വെബ് സെർവർ സോഫ്റ്റ്‌വെയറിലേന്നും സവിശേഷതകൾ എന്നൊക്കെയാണെന്ന് നമുക്ക് നോക്കാം.



ചിത്രം 4.4 : ഒരു ഡാറ്റാ സൈറ്റ്

ഇൻ്റർനെറ്റ് ഉപയോക്താക്കൾക്ക് ഇ-മെയിൽ, ബ്ലോഗ് തുടങ്ങിയ സേവനങ്ങളോ വെബ് പേജുകളോ വിട്ട് നൽകാൻ വെബ് സെർവർ സഹായിക്കുന്നു. www, ഇ-മെയിൽ തുടങ്ങിയ സേവനങ്ങൾ ഇൻ്റർനെറ്റിലും നൽകുന്നതിനു വേണ്ടി ഒരു സെർവർ ഓപ്പറേറ്റിംഗ് സിസ്റ്റമും സെർവർ സോഫ്റ്റ്‌വെയറും പ്രവർത്തനക്ഷമമാക്കിയിട്ടുള്ള ഒരു സെർവർ കമ്പ്യൂട്ടറാണ് വെബ് സെർവർ. ഇത് വളരെ കാര്യക്ഷമതയുള്ളതും എപ്പോഴും പ്രവർത്തിക്കുന്നതും ഉയർന്ന ബാൻഡ്‌വിഡ്യോം ഇൻ്റർനെറ്റ് ബന്ധമുള്ളതും ആയിരിക്കും. ലോകമെമ്പാടുമുള്ള ഇൻ്റർനെറ്റ് ഉപയോക്താക്കൾക്ക് ഇതിലുള്ള വെബ് സൈറ്റുകളും സേവനങ്ങളും ഉപയോഗിക്കാനാകും. ഒരു വെബ് സെർവറിൽ ഒന്നോ അതിലധികമോ പ്രോസസ്സുകളും, ഉയർന്ന വേഗതയുള്ള റാമും, വർഡിച്ച കാര്യക്ഷമതയുള്ള ഹാർഡ് ഡിസ്കുകളും, ദ്രുതഗതിയിലുള്ള ആശയവിനിമയത്തിന് ഉപകരിക്കുന്ന ഇന്റർനെറ്റ് കാർഡും മൊക്കെയുണ്ടാകും. വേഗതയുള്ള ഇൻ്റർനെറ്റ് ബന്ധം ഉറപ്പാക്കാനുസരണം വെദ്യുതി എത്തിക്കാനും മറ്റൊരു മായി ഒരു വെബ് സെർവർ സാധാരണ ഗതിയിൽ ഡാറ്റാ സൈറ്റുകളിലാണ് സ്ഥാപനങ്ങളുടെ സെർവറുകളും നന്ദിവർക്ക് ഉപ

കരണങ്ങളും സ്ഥാപിക്കുന്നതിന് മാത്രമായി വിനിയോഗിച്ചിട്ടുള്ള ഭേദികസഹഘടനകളാണ് ഡാറ്റാസെൻ്റർകൾ. കൂടുതലിന് വേണ്ടി, നിയുക്ത പ്രവൃത്തിയുമായി ബന്ധപ്പെട്ട നിർണ്ണായക ഡാറ്റാസെൻ്റർകൾ പ്രോസസ് ചെയ്യുന്നതിനും, പ്രയോജനപ്പെടുത്തുന്നതിനും വേണ്ടിയാണ് ഡാറ്റാസെൻ്റർകൾ ഉപയോഗിക്കുന്നത്. വൈദ്യുതവിതരണത്തിന് തടസം വരാതിരിക്കുവാനുള്ള സംവിധാനം, ശൈത്യീകരണ സംവിധാനം, വേഗമേറിയ ഇൻഡ്രോഗ്രാഫ് ബന്ധം, സുരക്ഷിതത്തുകൂട്ടമീകരണങ്ങൾ തുടങ്ങിയ സൗകര്യങ്ങൾ ഡാറ്റാസെൻ്ററിൽ ആവശ്യമാണ്. ഇത്തരത്തിലുള്ള ഒരു ഡാറ്റാസെൻ്റർ ചിത്രം 4.4-ൽ കാണിച്ചിരിക്കുന്നു. ചിത്രം 4.5-ൽ കാണുന്ന തരത്തിലുള്ള എളുപ്പത്തിൽ പരിപാലിക്കാവുന്നതും വാതായനങ്ങളോട് കൂടിയതുമായ പ്രത്യേകതരം ചടക്കുടുകളിൽ ഉറപ്പിച്ച ധാരാളം സെർവറുകൾ ഡാറ്റാസെൻ്റർകളിലുണ്ടാകും.

വിവിധ ലിനക്സ് പത്രിപ്പുകൾ (റെഡ്‌ഹാർട്ട്, ഓപ്പൺസുസ്, ഡിബിയൻ, ഉബുണ്ടു തുടങ്ങിയവ), മെമ്പ്രോകാസോഫ്റ്റ് വിന്റോസ് സെർവർ, ഫ്രീ ബി.എസ്.ഡി, ഓക്സിൻ സൊള്വേരൈസ് എന്നിവ പ്രചാരത്തിലുള്ള സെർവർ ഓപ്പറേറ്റീംഗ് സിസ്റ്റങ്ങളാണ്. ഒരു സെർവർ സ്ഥാപിച്ച് അതിൽ ഒരു വെബ് സെർവർ സോഫ്റ്റ്‌വെയർ ഉൾക്കൊള്ളിച്ച് ഓപ്പറേറ്റീംഗ് സിസ്റ്റത്തിന് അനുയോജ്യമായ രീതിയിൽ അതിനെ രൂപപ്പെടുത്തിയെടുക്കേണ്ടതുണ്ട്. സംഭവിച്ച് വച്ചിട്ടുള്ള ഫയലുകൾ ഉപയോകതാക്കാക്ഷർക്ക് സമയബന്ധിതമായി നൽകുന്നതിനു വേണ്ടി, ഫൈല്പുർട്ടേക്റ്റോകോളിജീ സഹായത്താൽ കൂടുതലിന് സെർവർ മോഡലിൽ പ്രവൃത്തിക്കുന്ന ഒരു പ്രോഗ്രാമാണ് വെബ് സെർവർ സോഫ്റ്റ്‌വെയർ. ഉപയോകതാക്ഷർ നൽകുന്ന അപേക്ഷയ്ക്കനുസരിച്ച്, വെബ് സെർവർ സോഫ്റ്റ്‌വെയറിന്റെ സഹായത്തോടെ ഈ ഫയലുകൾ വെബ് പ്രോഗ്രാമ്മാക്കി കൈകൊള്ളിക്കുന്നു. തുടർന്ന് കൂടുതലിന് പാക്കേജുകളായ FTP (ഫയൽ ട്രാൻസ്�ഫർ പ്രോട്ടോക്കോൾ), ഇ-മെയിൽ, DNS, ഡാറ്റാബേസ് തുടങ്ങിയവ കൂടി ഉൾപ്പെടുത്താവുന്നതാണ്. വെബ് സെറ്റിംഗ് സുരക്ഷിതത്തിന് രൂപംകൊടുക്കുക, ഓലിലാറിക്കം വെബ് സൈറ്റുകളുടെ നിർമ്മാണം സാധ്യമാക്കുക എന്നിങ്ങനെയുള്ള വ്യവസ്ഥകൾ വെബ് സെർവർ സോഫ്റ്റ്‌വെയർ തന്നെ കൈകാര്യം ചെയ്യും.



ചിത്രം 4.5 : പ്രത്യേകതരം ചടക്കുള്ളിലുറപ്പിച്ചിട്ടുള്ള സെർവറുകൾ

വെബ് സെർവർ സോഫ്റ്റ്‌വെയർ സ്ഥാപിച്ച് പ്രവർത്തന യോഗ്യമാക്കിയശേഷം അതോടൊപ്പം സോഫ്റ്റ്‌വെയർ പാക്കേജുകളായ FTP (ഫയൽ ട്രാൻസ്�ഫർ പ്രോട്ടോക്കോൾ), ഇ-മെയിൽ, DNS, ഡാറ്റാബേസ് തുടങ്ങിയവ കൂടി ഉൾപ്പെടുത്താവുന്നതാണ്. വെബ് സെറ്റിംഗ് സുരക്ഷിതത്തിന് രൂപംകൊടുക്കുക, ഓലിലാറിക്കം വെബ് സൈറ്റുകളുടെ നിർമ്മാണം സാധ്യമാക്കുക എന്നിങ്ങനെയുള്ള വ്യവസ്ഥകൾ വെബ് സെർവർ സോഫ്റ്റ്‌വെയർ തന്നെ കൈകാര്യം ചെയ്യും.

#### 4.2.2 സോഫ്റ്റ്‌വെയർ പോർട്ടുകൾ (Software ports)

കമ്പ്യൂട്ടർ സിസ്റ്റത്തിലെ ഹാർഡ്‌വെയർ പോർട്ടുകളെക്കുറിച്ച് 11-ാം ക്ലാസിൽ നാം പറിച്ചിട്ടുണ്ട്. പുറിമേ നിന്നുള്ള ഉപകരണങ്ങൾ കമ്പ്യൂട്ടറുമായി ബന്ധപ്പെട്ടിരിക്കുവാൻ ഹാർഡ്‌വെയർ പോർട്ടുകൾ ഉപയോഗിക്കുന്നു. ഈ പോർട്ടുകൾ ഉപയോഗിച്ചാണ് ഇത്തരം ഉപകരണങ്ങൾ പ്രവർത്തിക്കുന്നത്.

ഉപയോഗിക്കുന്നതിനും, PS/2 പോർട്ട് കീബോർഡ്/മാസ് എൻവിവയുമായി ബന്ധിപ്പിക്കുന്നതിനും ഉപയോഗിക്കുന്നു. അതുപോലെ, ഒരു സോഫ്റ്റ്‌വെയർ പോർട്ട് ഉപയോഗിച്ചുകൊണ്ട് കൂടുതൽ കമ്പ്യൂട്ടറിനെ സെർവ്വറുമായി ബന്ധിപ്പിച്ച്, അതിലെ സേവനങ്ങളായ HTTP, FTP, SMTP തുടങ്ങിയവ ഉപയോഗിക്കാനാകും. പോർട്ടുകളെ തമിൽ തിരിച്ചറിയുന്നതിനായി അവയ്ക്ക് അതിവിഹിഷ്ടമായ സംവ്യാനം നൽകുന്നു. ഈ-മെയിൽ, മയൽ കൈമാറ്റം തുടങ്ങിയ ഒരേ സെർവ്വർ കമ്പ്യൂട്ടറിൽ തന്നെ നടക്കുന്ന വ്യത്യസ്ത സേവനങ്ങളെ തിരിച്ചറിയുക എന്നതാണ് സോഫ്റ്റ്‌വെയർ പോർട്ടിന്റെ ഉദ്ദേശം. വ്യത്യസ്ത പോർട്ട് സംവ്യാനം മാറ്റുപയോഗിച്ച്, സെർവ്വറിൽ ലഭ്യമായ ഓരോ സേവനവും തുടങ്ങാനും, ഉപയോഗിക്കാനും/എടുക്കാനും സാധിക്കുന്നു. പോർട്ട് സംവ്യാനം എന്നത് ഒരു 16-ബിറ്റ് സംഖ്യയാണ്. അത് കമ്പ്യൂട്ടറിന്റെ IP മേൽവിലാസത്തോടൊപ്പം (URL-ന് ഒപ്പ്) ചേർത്തുപയോഗിച്ച് സെർവ്വറിൽ ലഭ്യമായ ഒരു പ്രത്യേക സേവനവുമായി ആശയവിനിമയം സാധ്യമാക്കുന്നു. ഒരു വെബ് സൈറ്റിലുള്ള സേവനങ്ങൾ താഴെ തന്നിരിക്കുന്ന വിധം ഉപയോഗിക്കാൻ സാധിക്കും.

<http://google.co.in:80>

ഇവിടെ http എന്നത് പ്രോട്ടോക്കോളും, google.co.in എന്നത് ഡോമേനിൻ നാമവും 80 എന്നത് പോർട്ടിന്റെ സംവ്യാനം മാറ്റുമാണ്. പട്ടിക 4.1-ൽ അറിയപ്പെടുന്ന ചില പോർട്ട് സംവ്യാനം മാറ്റും അനുബന്ധ സേവനങ്ങളും കൊടുത്തിരിക്കുന്നു.

തന്ത്ര പോർട്ട് സംഖ്യ	സേവനം
20 & 21	മയൽ കൈമാറ്റുന്നതിനുള്ള മയൽ ട്രാൻസ്ഫർ പ്രോട്ടോക്കോൾ (FTP)
22	സുരക്ഷിത ഷേർ (സൈക്കുൾ ഷേർ) (SSH)
25	ലഭ്യ സന്ദേശങ്ങൾ കൈമാറ്റുന്നതിനുള്ള സിസിൾ മെയിൽ ട്രാൻസ്ഫർ പ്രോട്ടോക്കോൾ (SMTP)
53	ഡോമേനിൻ നെയിം സിസ്റ്റം (DNS)
80	ഹൈപ്പർ ടെക്നോളജി ട്രാൻസ്ഫർ പ്രോട്ടോക്കോൾ (HTTP)
110	പോസ്റ്റ് ഓഫീസ് പ്രോട്ടോക്കോൾ (POP3)
443	HTTP സൈക്കുൾ (HTTPS)

പട്ടിക 4.1 : പോർട്ടുകളും അവയുടെ സേവനങ്ങളും

### 4.2.3 DNS സെർവീസ് (DNS servers)

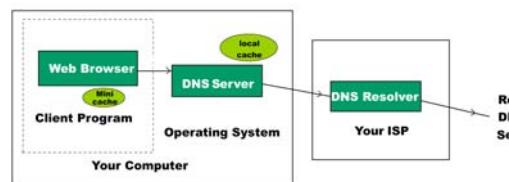
11-ാം ക്ലാസിൽ കമ്പ്യൂട്ടർ ശൈലികൾ എന്ന അധ്യായത്തിൽ ഡോമേനേം നാമകരണ രീതിയെക്കുറിച്ച് നാം പറിച്ചിട്ടുണ്ട്. ഡോമേനേം നാമങ്ങളുടെ ധാരാഖേസും അനുബന്ധ IP മേൽ വിലാസങ്ങളും ഉള്ള ഒരു പ്രത്യേക നേര് വർക്കിംഗ് സോഫ്റ്റ്‌വെയർ ഓരോ DNS സെർവീസും പ്രവൃത്തിക്കുന്നുണ്ടാകും. ഡോമേനേം നാമകരണ സ്വന്തായം ഒരു DNS സെർവീസ് റിൽ പ്രവർത്തിക്കുകയും കൂടുതലും കമ്പ്യൂട്ടർ ഡോമേനേം നാമത്തിന്റെ IP മേൽവിലാസം തിരിച്ച് നൽകുകയും ചെയ്യുന്നു.



ചിത്രം 4.6 : ലോകമെമ്പാടുമുള്ള റൂട്ട് സെർവീസ്

ഒരു ഡോമേനേം നാമകരണ സ്വന്തായായത്തിൽ ധാരാളം DNS സെർവീസുകളുണ്ടായിരിക്കും. ഈ സ്ഥാനക്രമമനുസരിച്ച് കുറീകരിച്ചിരിക്കും .com, .org തുടങ്ങിയ ഉയർന്നതും മുകൾ തടിലുള്ളതുമായ ഡോമേനുകളോ, രാജ്യങ്ങളേയും സുചിപ്രിക്കുന്ന ഡോമേനേം

നാമങ്ങളായ .in അല്ലെങ്കിൽ .uk തുടങ്ങിയവയോ ഉൾപ്പെടുന്ന 13 റൂട്ട് സെർവീസുകളായിരിക്കും ഏറ്റവും മുകളിലുള്ളത്. ചിത്രം 4.6-ൽ കാണിച്ചിരിക്കുന്നതു പോലെ ലോകമെമ്പാടുമായി പലസ്ഥലങ്ങളിൽ റൂട്ട് സെർവീസുകളുടെ ധാരാളം പകർപ്പുകൾ സ്ഥാപിച്ചിട്ടുണ്ടാകും. മറ്റൊരു DNS സെർവീസുകളും, സ്ഥാനക്രമത്തിന്റെ താഴെത്തലങ്ങളിൽ പ്രതിഷ്ഠിച്ചിരിക്കും. DNS എങ്ങനെയാണ് ഡോമേനേം നാമത്തിന്റെ IP മേൽവിലാസം തിരഞ്ഞെടുക്കുന്നതെന്ന് നമുക്ക് നോക്കാം.



ചിത്രം 4.7 : DNS തീരുമാനം

കേരളത്തിലെ പോലീസ് വകുപ്പിന്റെ വെബ് സേസ്റ്റ് നാം സന്ദർശിക്കുന്നു എന്ന് കരുതുക. www.keralapolice.org എന്ന പേരിലുള്ള പോലീസ് വകുപ്പിന്റെ ഡോമേനേം നമുക്ക് ബേഹസിൽ കേൾപ്പ് ചെയ്യാം. DNS എങ്ങനെനുസരിച്ച് സ്ഥാപിച്ചമാക്കുന്നു എന്ന് താഴെ തന്നീടുള്ള ഘട്ടങ്ങൾ വിശദീകരിക്കുന്നു. ചിത്രം 4.7 കാണുക.

1. എല്ലാ ബേഹസരുകളും അതിന്റെ കൂണാഷ് മെമ്മറിയൽ സമീപകാലത്ത് സന്ദർശിച്ച് IP മേൽവിലാസങ്ങൾ സുക്ഷിക്കുന്നുണ്ടാകും. അതിനാൽ, ബേഹസർ ആദ്യം പ്രാദേശിക മെമ്മറി (മിനി കൂണാഷ്) പരിശോധിച്ച് IP മേൽവിലാസമുണ്ടോ എന്ന് ഉറപ്പ് വരുത്തുന്നു. ഉണ്ടെങ്കിൽ അതുപയോഗിക്കുന്നു.
2. ബേഹസരിന്റെ കൂണാഷ് മെമ്മറിയൽ IP മേൽവിലാസം ഇല്ലെങ്കിൽ അത് ഓപ്പറേറ്റിംഗ് സിസ്റ്റമ്മിന്റെ പ്രാദേശിക കൂണാഷ് മെമ്മറി പരിശോധിക്കുന്നു.
3. അവിടെയുമില്ലെങ്കിൽ ISP-യുടെ പ്രാദേശിക DNS സെർവീസിൽ പരിശോധിക്കുന്നു.

4. ISP യുടെ DNS സെർവർിലും ഡോമേനിൽ നാമമില്ല എങ്കിൽ ISP യുടെ DNS സെർവർ, റൂട്ട് സെർവറിൽ തുടങ്ങി ആവർത്തിച്ചുള്ള തിരച്ചിലിലുടെ അത് കണ്ണഭത്തുന്നു.
5. ISP-യുടെ DNS സെർവർ, IP മേൽവിലാസം ബേഹസറിന് തിരിച്ച് നൽകുന്നു.
6. www.keralapolice.gov യുടെ IP മേൽവിലാസം ഉപയോഗിച്ചുകൊണ്ട് ബേഹസർ, വൈബ് സെർവറുമായി ബന്ധപ്പെടുകയും, ബേഹസർ വിൻഡോയിൽ വൈബ് പേജ് കാണിക്കുകയും ചെയ്യുന്നു. IP മേൽവിലാസം കിട്ടിയില്ലായെങ്കിൽ ‘സെർവർ കണ്ണ തിരിയില്ല’ എന്ന സന്ദേശം ബേഹസർ വിൻഡോയിലെത്തിക്കുന്നു.



ഡോമേനിൽ നാമങ്ങളുടെയും അനുബന്ധ മേൽവിലാസങ്ങളുടെയും മൊത്തമായുള്ള ഡാറ്റ ബേന് ലോകമെമ്പാടുമുള്ള 13 റൂട്ട് സെർവറുകളാണ് കൈകാര്യം ചെയ്യുന്നത്. ഓരോ റൂട്ട് സെർവറും പലരാജ്യങ്ങളിൽ സ്ഥാപിച്ചിട്ടുള്ള സെർവറുകളുടെ ഒരു ശൃംഖലയിലാണുള്ളത്. അവയ്ക്ക് A, B, C, D, E, F, G, H, I, J, K, L, M എന്നിങ്ങനെ നാമകരണം നൽകിയിരിക്കുന്നു. ICANN, NASA (നാഷണൽ ഏരോസൈറ്റിക്സ് ആൻഡ് സ്പേസ് അഫ്മിനിസ്ട്രേഷൻ), മെരി ലാൻ്റ് സർവകലാശാല, വൈൽഡേസൻ ഇൻകോർപ്പറേഷൻ തുടങ്ങിയ സ്ഥാപനങ്ങളാണ് ഈ സെർവറുകളെ പതിപാലിക്കുന്നത്. റൂട്ട് സെർവറുകളെ പതിപാലിക്കുന്ന സ്ഥാപനങ്ങൾ ചേർന്നാണ് ICANN (ഇൻഡൈന്റ് കോർപ്പറേഷൻ ഫോർ അരെസ്റ്റർവ് ടെക്നിക്സ് ആൻഡ് നോവ്ഹെൽസ്) റൂട്ട് സെർവർ സ്ഥാപായക ഉപദേശക സമിതി ഉണ്ടായിരിക്കുന്നത്. ഇവരാണ് ഇൻഡൈന്റ് ലൈബ്രറി റൂട്ട് സെർവർ സ്ഥാപായത്തിന്റെ പ്രവർത്തനം, കാര്യനിർവ്വഹണം, സുരക്ഷിതത്വം, സന്ധുരണ്ട ഉറപ്പുകൾ തുടങ്ങിയവയുമായി ബന്ധപ്പെട്ട ഉപദേശ അഥവാ ICANN റൂട്ട് സെർവറുകൾ. ഇന്ത്യയിൽ NIXI (നാഷണൽ ഇൻഡൈന്റ് എക്സ്ചേഞ്ച് ഓഫ് ഇന്ത്യ) യാണ് മുംബൈ (I റൂട്ട്), ഡൽഹി (K റൂട്ട്), ചെറേന (F റൂട്ട്) എന്നിവിടങ്ങളിൽ റൂട്ട് സെർവറുകൾ നിർമ്മിച്ച് നൽകിയിട്ടുള്ളത്.

നൂറ് കണക്കിന് കമ്പ്യൂട്ടറുകളും ഉപകരണങ്ങളും ഇൻഡൈന്റന്റുമായി ബന്ധപ്പെട്ടിട്ടുള്ള വിദ്യാ ഭ്യാസസ്ഥാപനങ്ങൾ, സർക്കാർ വകുപ്പുകൾ, സോഫ്റ്റ്‌വെയർ നിർമ്മാണശാലകൾ തുടങ്ങിയവർ അവരുടെ ഇൻഡൈന്റന്റിൽ സ്വന്തം DNS സെർവർ സ്ഥാപിച്ചിട്ടുണ്ടാകും. ഉപയോഗത്താക്കൾ സ്ഥിരമായി ഉപയോഗിക്കുന്ന ഡോമേനിന്റെ അവയുടെ IP മേൽവിലാസങ്ങളും ഉൾപ്പെടുത്തിക്കൊണ്ട് ഇൻഡൈന്റന്റുമുണ്ട്. ഇൻഡൈന്റന്റിലെ കമ്പ്യൂട്ടർ ഒരു വൈബ് സെസറ്റ് ഉപയോഗിക്കാൻ ശ്രമിക്കുന്നേം പ്രസ്തുത വൈബ് സെസറ്റിന്റെ ഡോമേനിലും കണ്ണഭത്തുന്നു. ഈ രീതിയിൽ IP മേൽവിലാസം കണ്ണ താംഗായില്ലായെങ്കിൽ ISP-യുടെ DNS സെർവറിൽ അനേകിക്കുന്നു.



നിലവിലുള്ള നമ്മുടെ DNS ഭാതാവിന് പകരം ഉപയോഗിക്കാവുന്ന സത്യത സെവ നമാണ് ഗുഡിലൂഡ് പൊതു DNS. ഇതിന്റെ IP മേൽവിലാസങ്ങളാണ് 8.8.8.8, 8.8.4.4 എന്നിവ. ഈ പൊതു DNS റൂട്ട് IP മേൽവിലാസങ്ങളിൽ എതിലേക്കുന്നതും നമ്മുടെ കമ്പ്യൂട്ടറിനെ നയിക്കുന്നതിയിൽ നമ്മുടെ നേറ്റ്വർക്കിനെ ക്രമീകരിക്കാവുന്നതാണ്.

### നിങ്ങളുടെ പുരോഗതിയിൽക്കൂടുക



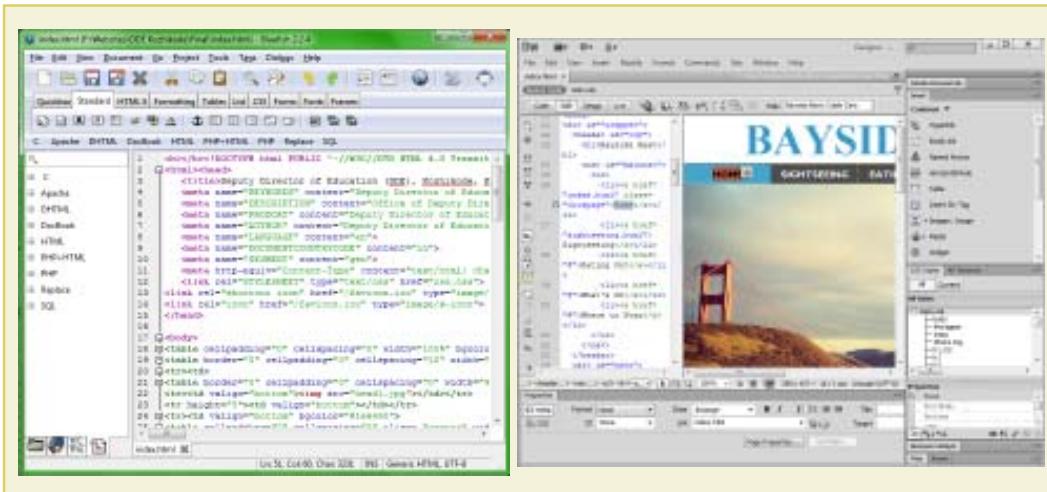
1. ഇൻഫർമേഷൻ പ്രോട്ടോക്കോളിജീസ് (IP) ഉയർന്ന തലത്തിൽ പ്രവർത്തി ക്കുന്ന ഒരു പ്രോട്ടോക്കോളിജീസ് പേരേഴുതുക.
2. HTTPS എന്നതിന്റെ പുർണ്ണ രൂപം എഴുതുക.
3. ഡാറ്റാ സെസ്റ്റുകളിൽ വെബ് സെർവ്വറുകൾ സ്ഥാപിക്കുന്നതിന്റെ നേട്ട് അംഗൾ എന്തെല്ലാം?
4. ശത്രൂവായ തെറ്റോ എന്ന് പ്രസ്താവിക്കുക.
  - a. കൂയൽ-സെർവ്വർ മാതൃക അനുസരിച്ചാണ് വെബ് സെർവ്വർ സോഫ്റ്റ്‌വെയർ പ്രവർത്തിക്കുന്നത്.
  - b. സെർവ്വർ ഓപ്പറേറ്റിംഗ് സിസ്റ്റം വെബ് സെർവ്വർ സോഫ്റ്റ്‌വെയറും ചേർന്നതാണ് വെബ് സെർവ്വർ.
5. ഒരു സോഫ്റ്റ്‌വെയർ പോർട്ട് സംഖ്യാനാമസ്ക്രിപ്റ്റായതിലെ ബിറ്റുകളുടെ എണ്ണം \_\_\_\_\_ ആണ്.
  - a. 8
  - b. 16
  - c. 32
  - d. 64
6. ഡൈാമെനിം നാമകരണസ്വഭാവം ഡൈാമെനിം നാമത്തിന്റെ \_\_\_\_\_ തിരികെ നൽകുന്നു.

### 4.3 വെബ് രൂപകൽപ്പന ചെയ്യാം (Web designing)

വെബ് സെസ്റ്റ് നിർമ്മാണത്തിന്റെ ആദ്യഘട്ടം വെബ് പേജുകളുടെ ആസുത്രണമാണ്. നമ്മുടെ വിദ്യാലയത്തിനായി ഒരു വെബ് സെസ്റ്റ് വികസിപ്പിക്കുന്നു എന്ന് കരുതുക. വെബ് പേജുകളും അവ തമിലുള്ള ബന്ധങ്ങളും തീരുമാനിച്ച് കഴിഞ്ഞാൽ അവയുടെ രൂപകൽപ്പനയിലേക്ക് കടക്കാം. ഇതിനായി ഏതെങ്കിലും ഒരു ടെക്നോളജി എഡിറ്റർ ഉപയോഗിക്കാം. ഒരു ഹോം പേജും, സ്കൂളിൽ ലഭ്യമാകുന്ന കോഴ്സുകൾ, സഹകരിക്കാൻ, ബന്ധപ്പെടേണ്ട മേൽവിലാസം തുടങ്ങിയവ ദൃശ്യമാക്കുന്ന മറ്റൊരു പേജും നിർമ്മിച്ച് അവയെ തമിൽ മെനു ഉപയോഗിച്ച് ബന്ധപ്പെടുത്തുന്ന രീതിയിൽ മനോഹരമായ ഒരു വെബ് സെസ്റ്റ് നമുക്ക് നിർമ്മിക്കാം.

വെബ് രൂപകൽപ്പന ചെയ്യുന്നതിന് നിരവധി സോഫ്റ്റ്‌വെയറുകൾ ലഭ്യമാണ്. വേഡ് പ്രോസസ് സൗകര്യ ഉപയോഗിച്ച് അനാധാരമായി ഒരു ഡോക്യുമെന്റ് തയ്യാറാക്കുന്നത് പോലെ, വെബ് നിർമ്മാണ ടൂളുകളിൽ ലഭ്യമായിട്ടുള്ള സംഗ്രഹിതങ്ങൾ സഹായത്തോടെ വെബ് പേജുകൾ നിർമ്മിക്കാവുന്നതാണ്. വെബ് പേജുകൾ രൂപകൽപ്പന ചെയ്യുന്നതിനും അവയെ പരസ്പരം ബന്ധപ്പെടിച്ച് കൊണ്ട് വെബ് സെസ്റ്റ് രൂപീകരിക്കുന്നതിനുമുള്ള സംവിധാനങ്ങൾ ഈ സോഫ്റ്റ്‌വെയറുകൾ നൽകുന്നു. FTP പ്രോട്ടോക്കോളുകൾ ഉപയോഗിച്ച് ഫയലുകൾ സെർവ്വറുകളിലേക്ക് കൈമാറ്റം ചെയ്യുന്നതിനുള്ള സ്ഥാലാവിക സൗകര്യം ഇത്തരം സോഫ്റ്റ്‌വെയറുകളിൽ ഉണ്ടാകും. ബൈപിഷ്ട്, ബൈട്ട് സ്ട്രോപ്പ്, അഡ്വോസ് ശ്രീംവീവർ, മെമ്പ്രോക്സോഫ്റ്റ് എക്സ്പ്രസ് വെബ് തുടങ്ങിയവ വെബ് നിർമ്മാണത്തിന്

ഉപയോഗിക്കുന്ന പ്രമുഖ സോഫ്റ്റ്‌വെയറുകളാണ് ചിത്രം 4.8-ൽ വെബ് നിർമ്മാണത്തിനു പയയോഗിക്കുന്ന പ്രശസ്തമായ ചില സോഫ്റ്റ്‌വെയറുകളുടെ IDE (ഇൻഗേറ്റീവ് ഡാബ്ല്യൂഇഡി) ലാബ്മെന്റ് എൻവയോൺമെന്റ്) കാണിച്ചിരിക്കുന്നു.

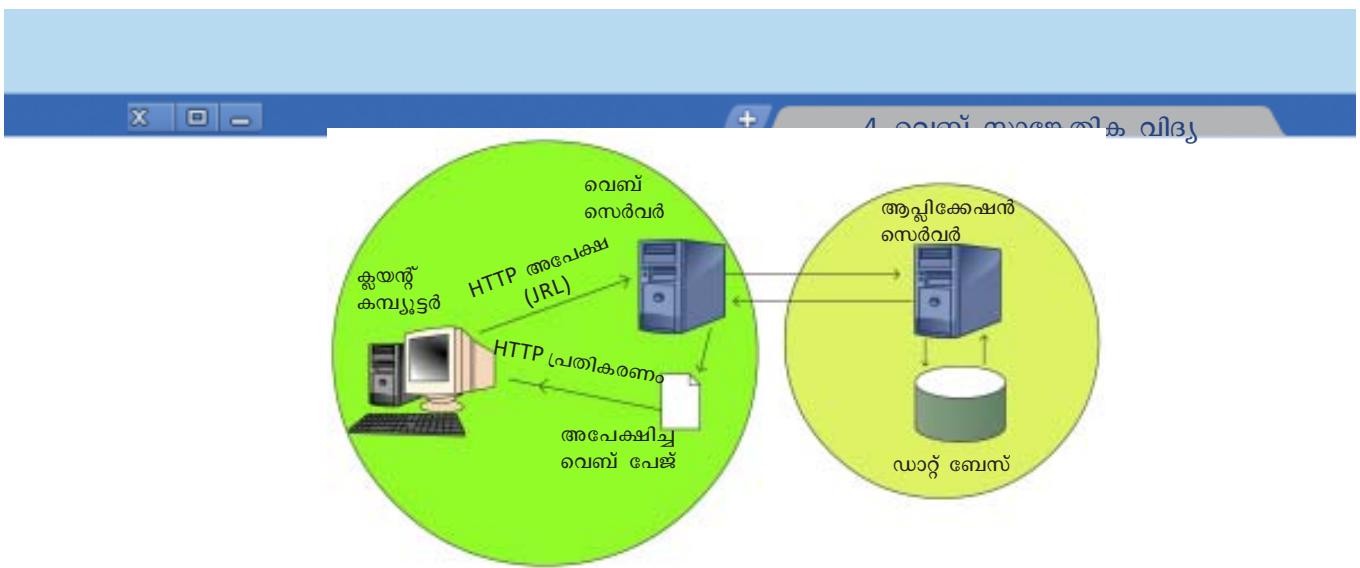


**ചിത്രം 4.8 :** വെബ് നിർമ്മാണത്തിനുപയോഗിക്കുന്ന സോഫ്റ്റ്‌വെയറുകളുടെ IDE

വെബ് സൈറ്റുകളുടെ നിർമ്മാണത്തെക്കുറിച്ച് ഹൈസ്കൂൾ ക്ലാസ്സുകളിൽ നിങ്ങൾ പഠിച്ചിട്ടുണ്ട്. HTML-ൽ ഉള്ള ടാഗുകളും അവയുടെ ആട്ടിബ്യൂട്ടുകളും ഉപയോഗിച്ച് വെബ് പേജുകൾ രൂപകൽപ്പന ചെയ്യാം.

#### 4.4 സ്റ്റാറ്റിക്, ഡയനാമിക് വെബ് പേജുകൾ (Static and dynamic web pages)

ചെറുകിട വ്യവസായങ്ങൾ, വിദ്യാലയങ്ങൾ തുടങ്ങിയവയുടെ വെബ് സൈറ്റുകൾ എപ്പോൾ സംഭവിച്ചാലും, പേജുകളിലെ ഉള്ളടക്കം സ്ഥിരമായി (സ്റ്റാറ്റിക് ആയി) നിൽക്കുന്നത് ശ്രദ്ധിച്ചു കാണുമ്പോൾ? ഇത്തരം വെബ് സൈറ്റുകളെ സ്റ്റാറ്റിക് വെബ് സൈറ്റുകൾ എന്ന് വിളിക്കുന്നു. SSLC/ഹയർസൈക്കൽസ് പരീക്ഷ എഴുതിയ വിദ്യാർഥികൾ തങ്ങളുടെ ഒരു റജിസ്ട്രേഷൻ നമ്പർ നൽകുന്നേയാൾ, അവരവരുടെ പരീക്ഷാഫലം ദൃശ്യമാക്കുന്നവിധം ഉള്ളടക്കം മാറിവരുന്ന വെബ് സൈറ്റുകളും നിങ്ങൾ കണ്ടിട്ടുണ്ടാകും. ഈ ഡയനാമിക് പേജുകൾ എന്നറിയപ്പെടുന്നു. പ്രോഗ്രാം കോഡിൽ മാറ്റം വരുത്താതിട്ടേണ്ടാൽ ഉള്ള ടക്കം സ്ഥിരമായിത്തന്നെന്ന നിലനിൽക്കുന്ന പേജുകളാണ് സ്റ്റാറ്റിക് വെബ് പേജുകൾ. ആദ്യ കാലത്ത് HTML മാത്രം ഉപയോഗിച്ചാണ് ഈ സ്റ്റാറ്റിക് വെബ് പേജുകൾ നിർമ്മിച്ചിരുന്നത്. പിന്നീട് ജാവ സ്ക്രിപ്റ്റ്, വിബി സ്ക്രിപ്റ്റ് തുടങ്ങിയ സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ വരവോടെ വെബ് പേജുകൾക്ക് ചലനാത്മകത കൈവന്നു. വെബ് പേജിന്റെ ചില ഭാഗങ്ങളിൽ മറ്റ് പോയിന്റ് വയ്ക്കുന്നേയാൾ ആ ഭാഗത്ത് നിന്നവും രൂപമാറ്റവും സംഭവിക്കുന്നതും, ചിത്രങ്ങൾ ഓൺ പിരകേ ഓനായി ക്രമീകരിക്കപ്പെടുന്നതുമെല്ലാം സ്ക്രിപ്റ്റിംഗ് ഭാഷ ഉപയോഗിക്കുന്നതു കൊണ്ടാണ്. ഇത്തരം പ്രത്യേകതകളുള്ള വെബ് പേജുകളും സ്റ്റാറ്റിക് വെബ് പേജുകളായിട്ടാണ് കണക്കാക്കിയിട്ടുള്ളത്.



സ്റ്റാറ്റിക് വെബ് പേജിനായുള്ള നിർദ്ദേശം

ഡയനിമിക് വെബ് പേജിനായുള്ള നിർദ്ദേശം

ചിത്രം 4.9 : സ്റ്റാറ്റിക്, ഡയനിമിക് എന്നീ വെബ് പേജുകൾക്കായുള്ള നിർദ്ദേശങ്ങൾ

അരോ തവണ വെബ് പേജ് എടുക്കുന്നോ സെർവർ ഭാഗത്ത് പേജിലുള്ള കോഡ് അത് നിർമ്മിക്കുന്നു എങ്കിൽ അത്തരം പേജിനെ ഡയനിമിക് വെബ് പേജ് എന്ന് വിളിക്കുന്നു. വെബ് സെർവിസ് സ്ഥാപിച്ചിട്ടുള്ള അപ്ലിക്കേഷൻ പ്രോഗ്രാം ഉപയോഗിച്ച് ഡയനിമിക് പേജുകൾ പ്രവർത്തിപ്പിക്കുന്നു. വെബ് പേജിലെ സ്ക്രിപ്റ്റ്, വെബ് സെർവിസ് പ്രവർത്തിക്കുന്നോൾ ഉണ്ടാകുന്ന HTML പേജുകൾ സ്റ്റാറ്റിക് പ്രൈസറിലേക്കയെപ്പെട്ടു നു. മികവാറും സന്ദർഭങ്ങളിൽ ഇത്തരം വെബ് പേജുകൾ നിർമ്മിക്കാൻ ഡാറ്റാബേസ് ലുള്ള ഡാറ്റയാണ് ഉപയോഗിക്കുന്നത്. SSLC/HSE പരീക്ഷാഫലങ്ങൾ കാണിക്കുന്ന വെബ് സെറ്റുകൾ, റോഡ്, റെയിൽ, വേദാമ യാത്രകൾക്കുള്ള ടിക്കറ്റ് മുൻകൂട്ടി ഉറപ്പിക്കാനോകുന്ന സെറ്റുകൾ, പണ്മിച്ചപാടുകൾക്കുള്ള വെബ് സെറ്റുകൾ തുടങ്ങിയവയുടെ പേജുകളിൽ പരിവർത്തനാത്മക ഉള്ളടക്കമുണ്ടാവുകയും അവ ഡയനിമിക് പേജുകളായി കണക്കാക്കുകയും ചെയ്യുന്നു. PHP, ASP, JSP തുടങ്ങിയവ ഉപയോഗിച്ച് ഡയനിമിക് വെബ് പേജുകൾ നിർമ്മിക്കുന്നു. സ്റ്റാറ്റിക്, ഡയനിമിക് എന്നീ വെബ് പേജുകൾ തമ്മിലുള്ള ഒരു താരതമ്യം പട്ടിക 4.2-ൽ തന്നീടുണ്ട്.

സ്റ്റാറ്റിക് വെബ് പേജുകൾ	ഡയനിമിക് വെബ് പേജുകൾ
വെബ് പേജിൽ ഉള്ളടക്കവും രൂപരേഖയും സ്ഥിരമായിരിക്കും.	ഉള്ളടക്കവും രൂപരേഖയും പ്രവർത്തിക്കുന്നോൾ വ്യത്യാസപ്പെടുന്നു.
സ്റ്റാറ്റിക് വെബ് പേജുകൾ ഒരിക്കലും ഡാറ്റാബേസ് ഉപയോഗിക്കുന്നില്ല.	ഡാറ്റാബേസ് ഉപയോഗിച്ച് കുറികളുടെ സഹായത്താൽ പരിവർത്തനാത്മക ഉള്ളടക്കം സൃഷ്ടിക്കുന്നു.
സ്റ്റാറ്റിക് വെബ് പേജുകൾ ബൈൽസിൽ നേരിട്ട് പ്രവർത്തിക്കുന്നതിനാൽ സെർവർ ഭാഗത്തുള്ള അപ്ലിക്കേഷൻ പ്രോഗ്രാമുകളുടെ ആവശ്യമില്ല.	ഡയനിമിക് വെബ് പേജുകൾ സെർവർ ഭാഗത്തുള്ള അപ്ലിക്കേഷൻ പ്രോഗ്രാമുകളുടെ സഹായത്താൽ പ്രവർത്തിച്ച് ഫലങ്ങൾ നൽകുന്നു.
സ്റ്റാറ്റിക് വെബ് പേജുകൾ എഴുപ്പുത്തിൽ നിർമ്മിക്കാനോകും	ഡയനിമിക് വെബ് പേജുകളുടെ നിർമ്മാണത്തിന് പ്രോഗ്രാമിംഗ് വൈദഗ്ധ്യം ആവശ്യമാണ്.

പട്ടിക 4.2: സ്റ്റാറ്റിക്, ഡയനിമിക് എന്നീ വെബ് പേജുകളുടെ താരതമ്യം

## 4.5 സ്ക്രിപ്റ്റുകൾ (Scripts)

HTML പേജുകളിൽ ഉൾക്കൊള്ളിക്കുന്ന പ്രോഗ്രാം കോഡുകളാണ് സ്ക്രിപ്റ്റുകൾ. നോട്ടപാഡ് പോലുള്ള ഒരു ടെക്നോളജി എഴിവും ഉപയോഗിച്ചാണ് അവ തയാറാക്കുന്നത്. ദൈനംദിന വെബ് പേജുകൾ നിർമ്മിക്കുന്നതിനായി ജാവ സ്ക്രിപ്റ്റ്, വി.ബി. സ്ക്രിപ്റ്റ്, പി.എച്ച്.പി, പേൾ തുടങ്ങിയ സ്ക്രിപ്റ്റീംസ് ഭാഷകൾ ഉപയോഗിക്കുന്നു. ഓപ്പറേറ്റിംഗ് സിസ്റ്റം തിരിക്കേണ്ട സഹായത്തോടെ കമ്പ്യൂട്ടർ ഹാർഡ്‌വെയറിനെക്കാണ്ട് നിർദ്ദേശങ്ങൾ നടപ്പിലാക്കിക്കുന്നവയാണ് സാമ്പത്തിക പ്രോഗ്രാമിംസ് ഭാഷകൾ. എന്നാൽ വെബ് സെർവ്വറുകളാൽ വ്യാവ്യാനിക്കപ്പെടുന്നവയാണ് സ്ക്രിപ്റ്റീംഗ് ഭാഷകൾ. ഈന്, ഭൂരിഭാഗം ഏക കമ്പ്യൂട്ടർ അധിഷ്ഠിത പ്രോഗ്രാമുകളും, വെബ് അധിഷ്ഠിത പ്രോഗ്രാമുകളാൽ പുനസ്ഥാപിക്കപ്പെട്ടിട്ടുണ്ട്. ധനകാര്യസ്ഥാപനങ്ങളിലും മറ്റും ഉപയോഗിച്ചിരുന്ന സോഫ്റ്റ്‌വെയറുകൾ, അതാൽ ശാഖകളിൽത്തന്നെ സ്ഥാപിച്ചിരുന്നവയാണ്. എന്നാൽ ഈന് ഈ സ്ഥാപനങ്ങൾ തങ്ങളുടെ സോഫ്റ്റ്‌വെയർ വെബ് സെർവ്വറിൽ ലഭ്യമാക്കുകയും അത് ഇൻറെന്റിന്റെ സഹായത്താൽ ജീവനകാർ ഉപയോഗിക്കുകയും ചെയ്യുന്നു. ധന വിനിമയ സ്ഥാപനങ്ങളിൽ ആവശ്യമായ സോഫ്റ്റ്‌വെയറുകൾ, ഹയർസെക്ക്യൂറി പ്രവേശനത്തിനുള്ള സോഫ്റ്റ്‌വെയർ തുടങ്ങിയ വെബ് അധിഷ്ഠിത സോഫ്റ്റ്‌വെയറുകളുടെ നിർമ്മാണത്തിന് സ്ക്രിപ്റ്റീംസ് ഭാഷകൾ ഉപയോഗിക്കുന്നു. ഒരു HTML പേജിൽ <SCRIPT>, </SCRIPT> എന്നീ ടാഗുകൾക്ക് ഉള്ളിലായാണ് സ്ക്രിപ്റ്റ് എഴുതുന്നത്. ഒരു HTML ഡോക്യുമെന്റിൽ സ്ക്രിപ്റ്റ് ഉൾപ്പെടുത്തുന്നതിനോ, സൂചിപ്പിക്കുന്നതിനോ ആണ് <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നത്. ഇതിന് **Type**, **Src** എന്നീ ആട്ടിബ്യൂട്ടുകളുണ്ട്. സ്ക്രിപ്റ്റ് ടാഗിന് ഉള്ളിലുള്ള സ്ക്രിപ്റ്റീംബാഷ തിരിച്ചറിയാനായാണ് Type ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. സ്ക്രിപ്റ്റ് കോഡ് ഉൾക്കൊള്ളുന്ന ഫയലിന്റെ പേര് നൽകാനാണ് SRC ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. HTML ടാഗ്‌കളെയും ആട്ടിബ്യൂട്ടുകളെയും കൂറിച്ച് നമുക്ക് തുടർന്നുള്ള ഭാഗങ്ങളിൽ മനസിലാക്കാം.

ഒരു HTML കോഡിനുള്ളിൽ ജാവ സ്ക്രിപ്റ്റ് ഉൾപ്പെടുത്താൻ <SCRIPT Type="text/javascript"> ഉപയോഗിക്കുന്നു.

### 4.5.1 വിവിധതരം സ്ക്രിപ്റ്റീംസ് ഭാഷകൾ (Types of scripting languages)

ഇൻറെന്റിലെ ഒരു വെബ് സെറ്റിന്റെ ലോഗിൻ പേജ് പരിഗണിക്കുക (സാധാരണ യാതി ഉപയോക്താവ് യുസർ നാമവും, പാസ്വോഡും നൽകാനുപയോഗിക്കുന്ന പേജാണിത്). ഉപയോക്താവ് യുസർ നാമവും പാസ്വോഡും നൽകിയശേഷം ലോഗിൻ ബട്ടണിൽ ക്ലിക്ക് ചെയ്യും. എന്നാൽ ഈ നൽകാതെ ലോഗിൻ ബട്ടണിൽ ക്ലിക്ക് ചെയ്താൽ എന്ത് സാഭവിക്കും? സാഭാവികമായും ഈ നൽകിയിട്ടില്ലെന്ന് കമ്പ്യൂട്ടർ ഉപയോക്താവി നോക് പറയും. രണ്ട് കമ്പ്യൂട്ടറുകളാണ് ഈ വെബ് പേജ് നിയന്ത്രിക്കുന്നത്-വെബ് പേജ് പ്രദർശിപ്പിക്കുന്ന ക്ലിക്ക് കമ്പ്യൂട്ടറും വെബ് പേജ് നൽകുന്ന സെർവ്വർ കമ്പ്യൂട്ടറും. ഉപയോക്താവ് യുസർ നാമം നൽകിയിട്ടുണ്ടോ എന്ന് എവിടെയാണ് പരിശോധിക്കുന്നത്. ക്ലിക്ക് കമ്പ്യൂട്ടറിലോ അതോ സെർവ്വർ കമ്പ്യൂട്ടറിലോ? ധാരാളം ആളുകൾ ഒരേ വെബ് സെറ്റ് സന്ദർശിക്കുന്നതിനാലും സെർവ്വർ തന്നെ ഇത്തരം അപേക്ഷകളെല്ലാം കൈകാര്യം

ചെയ്യേണ്ടതിനാലും കൂട്ടയർ കമ്പ്യൂട്ട് ടീംനേക്കാൾ വളരെയധികം തിരഞ്ഞെടുപ്പുണ്ടോളെ അഭ്യന്തരിച്ചു നിന്നും ഒരു മാറ്റം ആയതിനാൽ കൂട്ടയർ ഭാഗത്ത് ചെയ്യാവുന്ന പ്രവൃത്തികളെല്ലാം അവിടെത്തന്നെ ചെയ്യുന്നതാണ് ഉള്ളിടത്തിനു പരിശോധന സെർവീസിലാണ് നടക്കുന്നതെങ്കിൽ, ഉപയോകതാവും നൽകിയ അപേക്ഷ കൂട്ടയർ നിന്നും ഇൻറർനെറ്റിലൂടെ സെർവീസിൽ അയയ്ക്കേണ്ടിവരും. ചിലപ്പോഴെങ്കിലും സെർവീസിലെത്താൻ ഇതിന് ഇൻറർനെറ്റിലൂടെ വളരെ ദുരം സമുത്തിക്കേണ്ടിയും വരും. വളരെ അധികം കൂട്ടയർ കളിൽ നിന്ന് ലഭിക്കുന്ന ഇത്തരം അപേക്ഷകൾ വന്ന ക്രമത്തിൽ സെർവീസിലെത്താൻ അയയ്ക്കേണ്ടിവരും. സെർവീസിലെപ്പൊന്നും ചെയ്യുന്നത് വരെ അയച്ച അപേക്ഷ കാത്ത് നിരയിൽ നിൽക്കുന്നു. കൂട്ടയർ ഏതെങ്കിലും യുസർ നാമമോ പാസ്വോഡോ നൽകിയിട്ടുണ്ടോ എന്ന് ക്രമം അനുസരിച്ച് സെർവീസിലെത്താൻ അയയ്ക്കേണ്ടിവരും. നൽകിയിട്ടില്ലെങ്കിൽ, ചിത്രം 4.10-ൽ കാണി

### Login Failed. Please Check User Name.

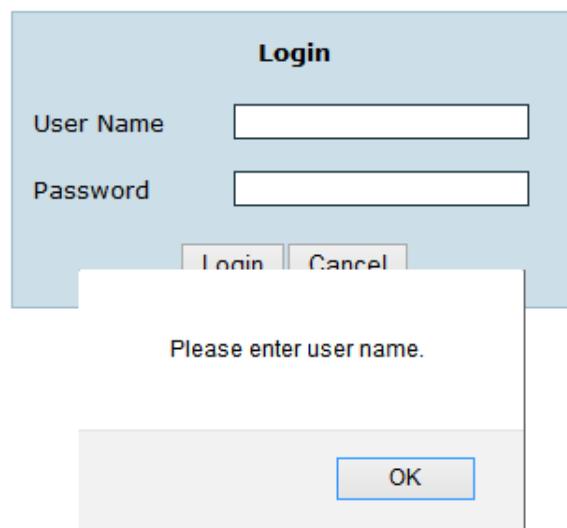
**Login**

User Name

Password

**Login** **Cancel**

ചിത്രം 4.10 : സെർവീസിൽ പ്രതികരണം



ചിത്രം 4.11 : കൂട്ടയർ ബൈസാർഡ് പ്രതികരണം

ചീട്ടുള്ളതുപോലെ ഉചിതമായ ഒരു സന്ദേശം സെർവീസിലേക്ക് അയയ്ക്കുന്നു. ഈ സന്ദേശവും ഇൻറർനെറ്റിലൂടെ വളരെ ദുരം സമുത്തിച്ച് കൂട്ടയർിലെത്തുന്നു. ചുരുക്കെത്തിൽ യുസർ നാമം നൽകാതെ ഉപയോകതാവ് സബ്മിറ്റ് ബട്ടണിൽ കൂടിക്കുള്ള ചെയ്താൽ യുസർ നാമം നൽകിയിട്ടില്ലെന്ന എന്ന സന്ദേശം കൂട്ടയർ കമ്പ്യൂട്ടിൽ ലഭിക്കുന്നത് വരെ അയാൾ കാത്തിരിക്കേണ്ടിവരുന്നു. കൂടാതെ കൂട്ടയർ നിന്ന് സെർവീസിലേക്കും തിരിച്ചും ധാരം സമുത്തിക്കുന്നതിനാൽ ശുംഖലയുടെ ഗതാഗതത്തിനു വർധിക്കുകയും ചെയ്യുന്നു. ഈ പരിശോധന കൂട്ടയർ ഭാഗത്തു തന്നെ നടക്കുകയാണെങ്കിൽ, ഉപയോകതാവ് സബ്മിറ്റ് ബട്ടണം അമർത്ഥത്തുനോൾ, യുസർ നാമമോ, പാസ്വോഡോ നൽകിയിട്ടുണ്ടോ എന്ന് കൂട്ടയർ ഭാഗത്തെ സ്ക്രീപ്പറ്റ് കോഡിന് തന്നെ പരിശോധിക്കാവുന്നതാണ്. അല്ലാത്തപക്ഷം ഉചിതമായ ഒരു സന്ദേശം ദൃശ്യമാക്കാവുന്നതുമാണ്. ഈ പ്രക്രിയയിൽ, ധാരം ഇൻറർനെറ്റിലൂടെ സെർവീസിലേക്ക് സമുത്തിക്കുകയോ, ലഭിതമായ ഈ ദൗത്യത്തിന് സെർവീസിനെ ശല്യപ്പെടുത്തുകയോ ചെയ്യുന്നില്ല. ഉപയോകതാവ് സബ്മിറ്റ് ബട്ടണം അമർത്ഥത്തുനോൾ

യുസർനാമവും/പാസ്വോദ്ദേശവും നൽകിയിട്ടില്ല എന്ന സന്ദേശം നിമിഷനേരം കൊണ്ട് ചിത്രം 4.11-ൽ കാണുന്നത് പോലെ ഉപയോകതാവിനു ലഭിക്കും. ഇതിനായി ശുംഖലയിലെ വിഭവങ്ങൾ ഉപയോഗിക്കേണ്ടിവരുന്നുമില്ല.

ഈ നമ്മക്ക് മറ്റാരു സാഹചര്യം പരിചയപ്പെട്ടാം. ഒരു ഉപയോകതാവ് തെറ്റായ യുസർ നാമവും പാസ്വോദ്ദേശവും നൽകിയെന്ന് കരുതുക. യുസർ നാമവും പാസ്വോദ്ദേശവും നൽകിയിട്ടുണ്ടോ എന്ന് കൂട്ടൽ കമ്പ്യൂട്ടറിന് പരിശോധിക്കുവാൻ കഴിയേ്തെങ്കാം. എന്നാൽ ഈ ശരിയാണോ എന്ന് പരിശോധിക്കുവാൻ കഴിയില്ല. യുസർ നാമങ്ങളുടെയും അനുബന്ധ പാസ്വോദ്ദേശവുകളുടെയും വിശദാംശങ്ങൾ സെർവർ കമ്പ്യൂട്ടറിലാണ് സുക്ഷിക്കുന്നത് എന്ന താണ് ഇതിന് കാരണം. ഇവയുടെ പരിശോധന സെർവർ ഭാഗത്ത് മാത്രമേ ചെയ്യാൻ സാധിക്കുകയുള്ളതും ഡാറ്റയുടെ സാധൃത ഉറപ്പ് വരുത്തുന്നതിന് സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്ന സന്ദർഭമാണിൽ. സെർവർിലുള്ള വിവരങ്ങൾ ഉപയോഗപ്പെടുത്തിക്കൊണ്ട് ഡാറ്റയുടെ സാധൃത ഉറപ്പ് വരുത്തേണ്ടിവന്നാൽ അത് സെർവർ ഭാഗത്ത് തന്നെ ചെയ്യേണ്ടതുണ്ട്.

സ്ക്രിപ്റ്റിൽ ഭാഷകളെ കൂട്ടൽ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ എന്നും സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ എന്നും തരംതിരിച്ചിരിക്കുന്നു എന്ന് ഇതിൽ നിന്നും വ്യക്തമാണമ്പോ? കൂട്ടൽ ഭാഗത്തെ സ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്നത് കൂട്ടൽ ഭാഗത്ത് ചെയ്യേണ്ട ജോലി ബൈസറിൽ പ്രവർത്തിപ്പിച്ച് ചെയ്യിക്കുന്നതിനാണ്. സെർവറിൽ പ്രവർത്തിക്കുന്ന സ്ക്രിപ്റ്റുകളെ സെർവർ ഭാഗത്തുള്ള സ്ക്രിപ്റ്റുകൾ എന്നു പറയുന്നു. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ പ്രവർത്തിക്കുന്നോൾ ലഭിക്കുന്ന ഒരുപ്പുട്ട് HTML പേജുകളുടെ രൂപത്തിൽ ഉള്ളതും കൂട്ടലിലേക്ക് അയയ്ക്കാവുന്നതും ആയിരിക്കും.

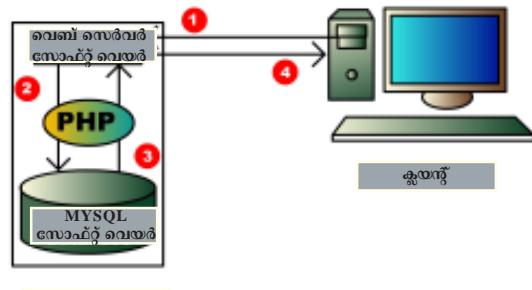
### A. കൂട്ടൽഭാഗത്തെ സ്ക്രിപ്റ്റിൽ (Client side scripting)

കൂട്ടൽ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്ങിൽ ഡാറ്റയുടെ സാധൃത ഉറപ്പാക്കുന്ന സ്ക്രിപ്റ്റ് കോഡ് HTML കോഡിനൊപ്പം ബൈസറിലേക്ക് ഡാറ്റാണിലോധി ചെയ്യപ്പെടുന്നു. ഡാറ്റ അസാധു ആണെങ്കിൽ ഉചിതമായ സന്ദേശം ദൃശ്യമാക്കുന്നു. ശരിയായ സാധൃത ഉറപ്പാക്കിയശേഷം മാത്രമേ ഡാറ്റയെ വെബ് സെർവറിലേക്ക് അയയ്ക്കുകയുള്ളതും.

സ്ക്രിപ്റ്റ്, കൂട്ടൽ ബൈസറിൽ പ്രവർത്തിക്കുന്നതിനാൽ അത് ഉപയോകതാക്കശകൾ ഉടൻ തന്നെ പ്രതികരണം നൽകുന്നു. വെബ് ആളൂട്ടിക്കേഷൻ പ്രവർത്തിക്കുന്നോൾ, ചില ചുമർ തലകൾ കൂട്ടൽ ബൈസറുമായി പങ്കുവയ്ക്കുവാൻ ഇത്തരം സ്ക്രിപ്റ്റിങ്ങിലൂടെ സാധിക്കുന്നു. സ്ക്രിപ്റ്റുകളെ പിന്തുണയ്ക്കാതെ ബൈസറുകൾ ഉള്ളതാണ് കൂട്ടൽ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്ങിൽ കൂട്ടൽ പരിമിതി. ചില അവസരങ്ങളിൽ ബൈസറിൽ സ്ക്രിപ്റ്റുകൾ പ്രവർത്തിക്കുന്നതിനുള്ള അനുമതി ഉപഭോക്താക്കൾ നിശ്ചയിച്ചിട്ടുമണ്ണാകും. അത്തരം സാഹചര്യങ്ങളിൽ കൂട്ടൽ ഭാഗത്തെ സ്ക്രിപ്റ്റ് പ്രവർത്തിക്കുകയില്ല. ജാവാസ്ക്രിപ്റ്റും വി.സി.സ്ക്രിപ്റ്റും പ്രമുഖ സ്ക്രിപ്റ്റിങ്ങ് സാങ്കേതിക വിദ്യകളാണ്. ഡാറ്റ സെർവറിലേക്ക് അയയ്ക്കുന്നതിനു മുൻപ്, സാധുകരണങ്ങളും ലളിതമായ കണക്കുകളുടെയും ചെയ്യേണ്ടിനാണ് കൂട്ടൽ ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്ങ് സാധാരണയായി ഉപയോഗിക്കുന്നത്.

## B. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗ് (Server side scripting)

ബഹുമാനിക്കുന്ന വെബ് പ്രോഗ്രാമുകളും കൗൺസിൽ ഭാഗത്തെ നാം ചർച്ച ചെയ്തിട്ടുള്ളതാണ്. ഈതരം പ്രോഗ്രാമുകൾ ഉണ്ടാക്കാൻ സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗ് ഉപയോഗിക്കുന്നു. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗിനുള്ള മറ്റാരു ഉദാഹരണം നോക്കാം. SSLC പരീക്ഷാ ഫലം പ്രസിദ്ധീകരിക്കുന്ന വെബ് സെർവ്വർ പരിഗണിക്കുക. നാം ഒരു കുട്ടിയുടെ രജിസ്ട്രേഷൻ നമ്പർ നൽകുമ്പോൾ, വെബ് സെർവ്വറിൽ ആ കുട്ടിയുടെ മാർക്ക് ലിസ്റ്റ് കാണപ്പെടും. ഓരോ കുട്ടിയുടെ കാര്യവും ഇതുപോലെ ആയിരിക്കും. SSLC പരീക്ഷയെഴുതുതിൽ ലക്ഷ്യ കമ്മാക്കിന് കുട്ടികൾ ഓരോരുത്തർക്കും വേണ്ടി ഓരോ വെബ് പ്രോഗ്രാമുകളെ നന്ദി പ്രായോഗികമണ്ണം. അങ്ങനെന്നെയകിൽ ഇത് സാധിക്കുന്നതെങ്ങനെ? ലക്ഷ്യകമ്മാക്കിന് കുട്ടികളുടെ പരീക്ഷാഫലം വെബ് സെർവ്വറിലുള്ള ഒരു ഡാറ്റാബേസിൽ സംഭരിക്കുന്നു. ഉപയോഗത്താവും രജിസ്ട്രേഷൻ നമ്പർ നൽകുമ്പോൾ ആ വിദ്യാർത്ഥിയുടെ പരീക്ഷാഫലം ഡാറ്റാബേസിൽ നിന്നും എടുക്കുന്നു. പിന്നീട് ഈ പരീക്ഷാഫലം ഉപയോഗിച്ച് സെർവ്വർ ഭാഗത്തെ സ്ക്രിപ്റ്ററുകൾ ഒരു വെബ് പ്രോഗ്രാമുകളും തുടർന്ന് ഇതു വെബ് പ്രോഗ്രാമുകളും ബൈബാൾലേഡ് അയയ്ക്കുകയും ബൈബാൾ അത് പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. ഈതരത്തിൽ സെർവ്വർ ഭാഗത്തെ സ്ക്രിപ്റ്റ്, SSLC പരീക്ഷയെഴുതുതിയ ഓരോ കുട്ടിക്കുമായി ഓരോ വെബ് പ്രോഗ്രാമുകളും അടിസ്ഥാനമാക്കിയുള്ള അപ്പിക്കേഷനുകളുടെ തരിത ഗതിയിലുള്ള വളർച്ച സെർവ്വർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകളുടെ ഉപയോഗം കുട്ടിയിട്ടുണ്ട്.



ചിത്രം 4.12 : സെർവ്വർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകളുടെ പ്രവർത്തനം

ഉപയോഗത്താവും നിർദ്ദേശിക്കുന്നതും, സെർവ്വർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഉൾപ്പെട്ടിട്ടുള്ള വെബ് പ്രോഗ്രാമുകൾ പ്രവർത്തിക്കുകയും, തൽപ്പലമായി ലഭിക്കുന്ന HTML കോഡ് കൂടുതലും ബൈബാൾലേഡ് അയയ്ക്കുകയും ചെയ്യുന്ന സാങ്കേതിക വിദ്യയാണ് സെർവ്വർ ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗ്. സെർവ്വർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഉള്ളടക്കം മാറിക്കൊണ്ടിരിക്കുന്ന വെബ് പ്രോഗ്രാമുകൾ നിർമ്മിക്കുന്നു. സ്ക്രിപ്റ്റുകൾ സെർവ്വറിൽ പ്രവർത്തിക്കുന്നതിനാൽ കുട്ടിക്കുമ്പും കുട്ടിക്കുമ്പും ബൈബാൾലേഡ് അപ്പരോടിംഗ് സിസ്റ്റത്തിന്റെയോ പതിപ്പും തരവും ഇതു പ്രവർത്തനത്തെ ബാധിക്കുന്നില്ല. സെർവ്വറിലുള്ള വിഭവങ്ങൾ ഉപയോഗിച്ച് കൊണ്ടാണ് ഇത് പ്രവർത്തിക്കുന്നത്. സെർവ്വർ ഭാഗത്തെ സ്ക്രിപ്റ്റ് എഴുതുന്നതിന് ഉപയോഗിക്കുന്ന പ്രമുഖമായ സ്ക്രിപ്റ്റിംഗ് ഭാഷകളാണ് PERL, PHP, ASP, JSP തുടങ്ങിയവ.

സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ തരംതിരിക്കലിനെക്കുറിച്ചുള്ള ഒരു താരതമ്യപരമാം പട്ടിക 4.3-ൽ

കൂയൻ്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗ്	സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗ്
സ്ക്രിപ്റ്റ് കൂയൻ്റ് ബേഹസിലേയ്ക്ക് പകർത്തുന്നു.	സ്ക്രിപ്റ്റ് വെബ് സെർവറിൽ നിലനിർത്തുന്നു.
സ്ക്രിപ്റ്റ് കൂയൻ്റ് ബേഹസിൽ പ്രവർത്തിക്കുന്നു.	സ്ക്രിപ്റ്റ് വെബ് സെർവറിൽ പ്രവർത്തിക്കുന്നു. സ്ക്രിപ്റ്റ് വെബ് പേജ് കൂയൻ്റ് ബേഹസിലേക്കയെക്കുന്നു.
കൂയൻ്റ് ഭാഗത്തെ ധാരായുടെ സാധുകരണത്തിനായാണ് പ്രധാനമായും സ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്നത്.	ധാരാബേസുമായി ബന്ധപ്പെട്ടുന്നതിനും വെബ് സെർവറിൽ നിന്നും ധാരാ ബേഹസിലേയ്ക്ക് അയക്കുന്നതിനുമാണ് സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഉപയോഗിക്കുന്നത്.
ഉപയോകതാവിന് കൂയൻ്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗ് പ്രവർത്തനം തടയാൻ സാധിക്കും.	സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകളുടെ പ്രവർത്തനം ഉപയോകതാവിന് തടയുവാൻ സാധിക്കില്ല.
ബേഹസിന്റെ തരവും പതിപ്പും കൂയൻ്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗ് പ്രവർത്തനത്തെ മാറ്റിക്കുന്നു.	വെബ് ബേഹസിന്റെ സവിശേഷതകൾ സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിംഗ് പ്രവർത്തനത്തെ മാറ്റിക്കുന്നില്ല.

പ്രതി 4.3: കൂയൻ്റ് ഭാഗത്തെയും സെർവർ ഭാഗത്തെയും സ്ക്രിപ്റ്റുകൾ തമിലുള്ള താരതമ്യം



കൂയൻ്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റുകൾ ഉപയോകതാവിംഗ് ബേഹസിൽ വച്ചുള്ള സാധുകരണ അശ്വകാണ് ഉപയോഗിക്കുന്നത്. മാത്രമല്ല അത് സെർവറിംഗ് ജോലി ഭാരവും ശുംഖവും ഇലെ ഗതാഗതക്കുരുക്കും കുറയ്ക്കുകയും ചെയ്യുന്നു. ആയതിനാൽ, സാധുത പരിശോധിക്കുന്നതിനായുള്ള സ്ക്രിപ്റ്റുകൾ പൂർണ്ണമായും കൂയൻ്റ് ഭാഗത്തെക്ക് മാറ്റിയിടുണ്ട്. അതുകൊണ്ട്, സെർവറിലേക്ക് അയയ്ക്കുന്ന ധാരാ പൂർണ്ണമായും തെറ്റുകളില്ലാത്തതും നേരിട്ട് ധാരാബേസിൽ സംഭരിക്കാവുന്നതുമാണ്. എന്നാൽ കൂയൻ്റ് ബേഹസൾ, സ്ക്രിപ്റ്റുകൾ മനസ്സിലാക്കാതിരിക്കുകയോ, ഉപയോകതാവും സുരക്ഷിത കാരണങ്ങളാൽ സ്ക്രിപ്റ്റിംഗ് ബേഹസിലുള്ള പ്രവർത്തനം തടയുകയോ ചെയ്താൽ, സാധുകരണം നടത്താതെ ധാരാ സെർവറിലേക്ക് അയയ്ക്കപ്പെടും. ഈകാരണം താഴെ അസാധ്യ ആയ ധാരാ ധാരാബേസിൽ സംഭരിക്കപ്പെടും. ധാരായുടെ സാധുത സെർവർ ഭാഗത്ത് കൂടി ഉറപ്പുകരാനായാൽ ധാരാബേസിൽ സംഭരിക്കുന്ന ധാരായുടെ സാധുത സംരക്ഷിക്കപ്പെടും.

തന്നിട്ടുണ്ട്.

#### 4.5.2 സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ (Scripting languages)

രണ്ടു തരം സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ നാം പരിചയപ്പെട്ടു കഴിഞ്ഞു. പ്രചാരത്തിലുള്ള ചില സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ സവിശേഷതകൾ നമുക്ക് ചർച്ച ചെയ്യാം.

##### A. ജാവ സ്ക്രിപ്റ്റ് (Java Script)

ഉപയോകതാവുമായി സംവദിക്കുന്ന വെബ് പേജുകളുടെ നിർമ്മാണത്തിന് ഉപയോഗിക്കുന്ന സ്ക്രിപ്റ്റിംഗ് ഭാഷയാണ് ജാവാസ്ക്രിപ്റ്റ്. നേര്റ്റെക്കേപ്പ് കമ്പ്യൂണിക്കേഷൻ കോർപ്പറേഷനിൽ പ്രവർത്തിച്ചിരുന്ന കാലത്ത് ബ്രെഡൻ ഇച്ച് (Brenden Eich ചിത്രം 4.13) നിർമ്മിച്ചു.

ചുതാൻ ജാവ സ്കീപ്പറ്റ് ഇൻഡപ്രെക്ട് ചെയ്യാവുന്ന ഭാഷ ആയിട്ടാണ് കൂട്ടയൻ്റെ ഭാഗത്ത് ജാവ സ്കീപ്പറ്റ് പ്രയോഗത്തിൽ വരുത്തിയിരിക്കുന്നത്. ജാവ സ്കീപ്പറ്റ് കോഡ് എഴുതുവാൻ ജിനി IDE യോഗോട്ടൊപ്പാഡ്യാ പോലുള്ള ഒക്കന്റെ എസിറ്റിറ്റുകൾ ഉപയോഗിക്കാം വുന്നതാണ്. കൂട്ടയൻ്റെ ഭാഗത്തെ ഒരു പ്രമുഖ സ്കീപ്പറ്റിംഗ് ടൂൾ ആയ ഇത് മിക്കവാറും ബ്രൗസറുകളിൽ പ്രവർത്തിക്കുന്നു. ജാവാസ്ക്രീപ്പറ്റ് HTML കോഡിൽ ഉൾപ്പെടുത്തിയോ പ്രത്യേക ഫയലിലെഴുതി HTML ഫയലിനോട് ലിങ്ക് ചെയ്തോ ഉപയോഗിക്കാം. പ്രത്യേക ഫയലിൽ എഴുതിയ ജാവ സ്കീപ്പറ്റ് .js എക്സ്ക്ലൂസിഫോടുകൂടി സൂക്ഷിക്കണം. ജാവാസ്ക്രീപ്പറ്റ് ഫയലുകളെ തിരിച്ചറിയാൻ ഇത് സഹായിക്കുന്നു. കൂട്ടയൻ്റെ ഭാഗത്തെ ഫോമുകളിലെ ധാരായുടെ സാധൂത ഉറപ്പിക്കാൻ ഉപയോഗിക്കുന്ന ടൂൾ എന്ന നിലയിൽ ജാവാ സ്കീപ്പറ്റ് പ്രഗസ്റ്റമാണ്. ലളിതമായ കമ്മക്കുടലുകളും ചലിക്കുന്ന കാർട്ടുൺ ചിത്രങ്ങളും വെബ് പേജുകളിൽ ഉൾപ്പെടുത്തുന്ന തിന്ന് ഇത് ഉപയോഗിക്കുന്നു. അധ്യായം 6-ൽ ജാവ സ്കീപ്പറ്റിനേക്കുറിച്ച് വിശദമായി ചർച്ച ചെയ്യാം.



ചീതു 4.13 ബേബ്ലോ

㉙ (1961 - )

## V.B. සැකිප්දු (VB Script)

விஷயத் வேளிக் என போகாமிட்டு காலை அடிசாலமாகி மெல்கொண்டாம்ப்ரி கோர்ப்புரேஷன் விகஸிப்பிசு ஸ்கிப்பிட்டிட்டு காலையான் VB ஸ்கிப்பிட்டு. மெல்கொண்டாம்ப்ரி ஹஸ்ற்கெந்தி ஏக்ஸ்பிளைனிகாயி கூடியது காலைத்த ஸ்கிப்பிட்டிட்டு காலையாயும் மெல்கொண்டாம்ப்ரி ஹஸ்ற்கெந்தி ஹஸ்ற்மேஷன் ஸ்கிப்பிட்டிட்டு காலையாயும் VB ஸ்கிப்பிட்டு உபயோகிக்கூடியு. நிற்கொருவர்கள், மற்ற பொருள்கள்கள்

VB സ്ക്രിപ്റ്റ് കോഡ് കൃത്യമായി മനസിലാക്കാനോ പ്രദർശിപ്പിക്കാനോ സാധിക്കില്ല. അതുകൊണ്ടു തന്നെ, കൂയൻ്റെ ഭാഗത്തെ പ്രചാരം കുറഞ്ഞ ഒരു സ്ക്രിപ്റ്റിൽ ടുൾ ആണിത്. സെർവർ ഭാഗത്ത് വിൻഡോസ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റത്തിന് പ്രചാരമേരെയുള്ളതിനാൽ സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റേഴുതുന്നതിന് VB സ്ക്രിപ്റ്റ് കൂടുതലായി ഉപയോഗിക്കുന്നു.

.NET ഫ്രെയിംവർക്കിന്റെ - ഉപയോഗയോഗ്യമായ ഫ്രോഗ്രാം കോഡിന്റെ ഒരു ശേഖരം, ആവിർഭാവത്തോടെ VB സ്ക്രിപ്റ്റ് .NET ഫ്രെയിംവർക്കിലെ ASP.NET എം്റെ ഭാഗമായി മാറി.

## C. PHP

PHP എന്നത് ഫൈലുകൾ ടെക്സ്റ്റ് പ്രീഡ്രോസസറിനെ സൂചിപ്പിക്കുന്നു. വെബ് വികസനത്തിന് അനുയോജ്യമായതും, HTML- ലെ ഉൾക്കൊള്ളൽക്കാവുന്നതുമായ ഈ പൊതു ആവശ്യങ്ങൾക്കായുള്ള ഒരു ഓപ്പൺ സോഴ്സ് സ്ക്രിപ്റ്റിൽ ഭാഷയാണ്. സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റ് എഴുതുന്നതിനുള്ള ടുൾ ആയ ഇതിന്റെ കോഡ് ജാവ, C, PERL എന്നി വയ്ക്ക് സമാനമാണ്. വളരെ എളുപ്പത്തിൽ പരിവർത്തനാത്മക വെബ് പേജുകൾ നിർമ്മിക്കാൻ ഇത് ഉപയോഗിക്കാം. 1994-ൽ റാസ്മസ് ലെർഡോർഫ് (Rasmus Lerdorf-ചിത്രം 4.14) ആണ് PHP വികസിപ്പിച്ചത്, എക്കിലും PHP സമൂഹമാണ് തുടർന്ന് അത് പരിപോഷിപ്പിച്ച് കൊണ്ടിരിക്കുന്നത്. PHP കോഡുള്ള വെബ് പേജുകളുടെ ഫയൽനാമത്തിന്റെ എക്സ്റ്റിംഷൻ .php എന്നായിരിക്കും. ഉപയോക്താവ് PHP വെബ് പേജിനായി അപേക്ഷിക്കുന്നേം, വെബ് സെർവർ HTML കോഡിൽ ഉൾപ്പെടുത്തിയിട്ടുള്ള PHP കോഡ് വ്യാവ്യാം നികുകയും പ്രവർത്തിപ്പിക്കുകയും ചെയ്യുന്നു. PHP കോഡ് വെബ് സെർവർിൽ പ്രവർത്തിക്കണമെങ്കിൽ ഒരു PHP ഇൻ്റർപ്പട്ടർ വെബ് സെർവർിൽ ഉണ്ടായിരിക്കണം. വെബ് സെർവർ റിൽ PHP കോഡ് പ്രവർത്തിക്കുന്നേം നിർമ്മിക്കപ്പെടുന്ന HTML പേജ് കൂയൻ്റെ ബേഖാസരി ലേക്ക് അയയ്ക്കുന്നു. ഡാറ്റാബേസ് ഫ്രോഗ്രാഫിന് സഹായിക്കുന്നു എന്നതാണ് PHP യുടെ പ്രധാനപ്പെട്ട സവിശേഷത. PHP യോടൊപ്പം സാധാരണ ഉപയോഗിക്കുന്ന ഡാറ്റാബേസാണ് MySQL. എല്ലാ ഓപ്പറേറ്റിംഗ് സിസ്റ്റങ്ങൾക്കും PHP ഇൻ്റർപ്പട്ടർ ലഭ്യമാണ്. സൗജന്യമായി ഇൻ്റർനെറ്റിൽ നിന്നും പകർത്താവുന്ന LAMP (Linux, Apache, MySQL, PHP) സെർവർ സോഫ്റ്റ്‌വെയറാണ് ലിനക്സ് പ്ലാറ്റ്‌ഫോമാമുകളിൽ ഉപയോഗിക്കുന്നത്. ഇവിടെ, ലിനക്സ് എന്നത് സെർവർ ഓപ്പറേറ്റിംഗ് സിസ്റ്റവും, അപ്പാശേഷ എന്നത് വെബ് സെർവറും, MySQL എന്നത് ഡാറ്റാബേസും, PHP എന്നത് സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റീംഗ് ഭാഷയും ആയി പ്രവർത്തിക്കുന്നു. ഇൻ്റർനെറ്റിൽ നിന്നും സൗജന്യമായി പകർത്താവുന്ന WAMP (WINDOWS, Apache, MySQL, PHP) സെർവർ സോഫ്റ്റ്‌വെയറിൽ വിൻഡോസ് ഓപ്പറേറ്റിംഗ് സിസ്റ്റം ഉപയോഗിക്കുന്നത്. 10-ാമതെത അധ്യായത്തിൽ PHP-യെക്കുറിച്ച് കൂടുതലായി ചർച്ച ചെയ്യാം.



ചിത്രം 4.14  
റാസ്മസ്  
ലെർഡോർഫ്  
(1968 - )

## D. ആക്റ്റീവ് സെർവർ പേജുകൾ (Active Server Pages)

പരസ്പരവിനിമയം നടത്തുന്ന വെബ് ആപ്ലിക്കേഷൻകൾ നിർമ്മിക്കുന്നതിനും പ്രവൃത്തി

പ്രീക്കുന്നതിനും വേണ്ടി സെർവർ ഭാഗത്ത് ഉപയോഗിക്കുന്ന സ്ക്രിപ്റ്റിൽ സംവിധാനമാണ് മെമ്പ്രോക്സേഡ്‌റീസ്റ്റ് ആക്റ്റീവ് സെർവർ പ്രോഗ്രാഫ് (ASP). ASP -യിൽ HTML-ഉം സ്ക്രിപ്റ്റിൽ ഭാഷയിലെ കോഡുമുണ്ടായിരിക്കും. സ്ക്രിപ്റ്റിൽ ഭാഷ VB സ്ക്രിപ്റ്റോ ജാവാ സ്ക്രിപ്റ്റോ ആയിരിക്കും. ASP ഫയലുകളുടെ എക്സിസ്റ്റിംഗ് .asp എന്നായിരിക്കും മെമ്പ്രോക്സേഡ്‌റീസ്റ്റ് സെർവർ സോഫ്റ്റ്‌വെയറായ ഇൻറൈറ്റേറ്റ് ഇൻഫർമേഷൻ സർവീസസിൽ (IIS-ൽ) തന്ത്രാധികാരി നിർമ്മിക്കപ്പെട്ടിട്ടുള്ള സവിശേഷത ഉപയോഗിച്ച് ഈ ഫയലുകൾ കുറയിൽ ചെയ്യുന്നു. ASP ഫയലുകൾ വിസ്തേയാസ് ഓപ്പറേറ്റീസ് സിസ്റ്റം തിൽ മാത്രം പ്രവർത്തിക്കുന്നവയാണ്. സെർവറിൽ എക്സിക്യൂട്ട് ചെയ്തതശേഷം ലഭിക്കുന്ന HTML വെബ് പേജ് കൂടുതലും ഭാഗത്തെ ബേഖാറിലേക്കയക്കുന്നു. വിവിധ തരത്തിലുള്ള ധാരാബേസുകളെ പിന്തുണയ്ക്കുന്ന, വളരെ പ്രമുഖമായ ഒരു സ്ക്രിപ്റ്റിൽ ഭാഷയാണിത്. ASP.NET എം്പത്രും വരവോടെ, ASP യുടെ തുടർപ്പതിപ്പുകൾ പുറത്തിരക്കുന്നത് മെമ്പ്രോക്സേഡ്‌റീസ്റ്റ് നിർത്തലാക്കി. കോഡുകളുടെ ചുരുക്കെഴുത്തും, വ്യത്യസ്ത അളവായ ബട്ടൺകൾ, ടെക്സ്റ്റ് ബോക്സും തുടങ്ങിയവയുടെ ലഭ്യതയും വെബ് ആളും ക്ഷേമനുകൾ നിർമ്മിക്കാനായി ASP.NET-ലുള്ള സവിശേഷതകളാണ്.

### E. ജാവ സെർവർ പ്രോഗ്രാഫ് (Java Server Pages)

പരിവർത്തനാത്മക വെബ് പേജ് നിർമ്മിക്കുന്നതിനുള്ള എളുപ്പവും വേഗമേറിയതുമായ സാങ്കേതിക വിദ്യ JSP പ്രദാനം ചെയ്യുന്നു. 1999-ൽ സണ്സ് മെമ്പ്രോക്സേഡ് നിർമ്മിച്ച സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റിൽ ഭാഷയാണിത്. JSP, PHP-കൾ സമാനമാണെങ്കിലും പ്രോഗ്രാമിൽ ഭാഷയായി ജാവയാണ് ഉപയോഗിക്കുന്നത്. JSP ഫയലുകളുടെ ഫയൽ എക്സിസ്റ്റിംഗ് .jsp എന്നാണ്. JSP പ്രവർത്തിപ്പിക്കുന്നതിന് അപ്പാശേഷ ടോക്യൂറ്റ് വെബ് സെർവർ ആവശ്യമാണ്. JSP കോഡിലുള്ള HTML കോഡും ജാവകോഡും വെബ് സെർവർ നിൽ പ്രവർത്തിച്ച് (എക്സിക്യൂട്ട് ചെയ്ത്) ഫലമായി ലഭിക്കുന്ന HTML കോഡിനെ ബേഖാറിലേയ്ക്ക് അയക്കുന്നു. വൻതോതിലുള്ള വെബ് അധിഷ്ഠിത സോഫ്റ്റ്‌വെയറുകൾ നിർമ്മിക്കുന്നതിനും പ്രവർത്തിപ്പിക്കുന്നതിനും ഉപയോഗിക്കുന്ന ജാവ 2 പ്ലാറ്റ്‌ഫോം എൻ്റർപ്പ്രൈസ് എഡിഷൻ അവിഭാജ്യ ഘടകമാണ് JSP.

### 4.6 കാസ്കേഡിൽ സ്റ്റൈൽ ഷീറ്റ് (Cascading Style Sheet)

HTML ഭാഷയിൽ എഴുതിയിട്ടുള്ള ഒരു ലിഖിതത്തിന്റെ ഘടന വിവരിക്കാൻ ഉപയോഗിക്കുന്ന സ്റ്റൈൽ ഷീറ്റ് ഹിറ്റ് (CSS). ഒരു വെബ് പേജിലുള്ള ചിഹ്നങ്ങളുടെ നിറം, അക്ഷരങ്ങളുടെ രീതി, വസ്തികകൾക്ക് ഇടയിലുള്ള സ്ഥലം, നിരകളുടെ വലുപ്പം പ്രദർശിപ്പിക്കുന്ന രീതി, അരകുകളും അവയുടെ നിറങ്ങളും, പശ്ചാത്യല ചിത്രങ്ങൾ/നിറങ്ങൾ എങ്ങനെയാണ് ഉപയോഗിക്കുന്നത് തുടങ്ങിയ വിവിധതരത്തിലുള്ള പ്രതീതികൾ നിയന്ത്രിക്കുവാൻ CSS ഉപയോഗിക്കുന്നു. HTML ഉള്ളടക്കത്തെ അതിന്റെ രൂപഭാഗങ്ങളിൽ നിന്നും വേർത്തിരിച്ച് നിർത്താൻ CSS നമ്മുടെ അനുവദിക്കുന്നു. മുന്തെ വ്യത്യസ്ഥ രീതികളിൽ CSS നമുക്ക് നടപ്പിലാക്കാം - ഇൻലൈൻ (അപ്പംതനം), എംബെഡ്ഡ് (ഉൾക്കൊള്ളിച്ച്), ലിങ്ക് (ബന്ധിപ്പിച്ച്) എന്നിവയാണവ.

- ഒപ്പമെചുത്തുന്ന രീതിയിൽ, വെബ് പേജിന്റെ ബോധി ഭാഗത്ത്, സ്റ്റൈൽ ആട്ടിബുക്ക് ഉപയോഗിച്ച് കൊണ്ട്, CSS ശൈലി ഓരോ ടാഗിനും പ്രത്യേകമായി പ്രയോഗിക്കാവുന്നതാണ്.

- ഉൾക്കൊള്ളിക്കുന്ന CSS കോഡുകൾ, വെബ്പേജിൽ ഫോറ്മേറ്റിംഗ് ഭാഗത്ത് സഹാപിക്കുന്നു.
- CSS കോഡ് മാത്രമുണ്ടാക്കാത്തുന്ന ഒരു ബാഹ്യ ഫയലിനെ വെബ്പേജുമായി ബന്ധിപ്പിച്ച് കൊണ്ടാണ് CSS നിർവ്വഹണം ലിങ്ക് റീതിയിൽ നടപ്പിലാക്കുന്നത്.

എല്ലാ പേജുകൾക്കും ഒരേ കോഡ് പുനരുപയോഗിക്കാമെന്നതാണ് CSS രീതു. ഒരു വെബ് സൈറ്റിനുവേണ്ടിയുള്ള CSS രൈറ്റ്റൽ ബാഹ്യഫയലായി ലിങ്ക് ചെയ്താണ് നടപ്പിലാക്കുന്നതെങ്കിൽ, ഒരു രൈറ്റ്റൽ വ്യത്യാസപ്പെടുത്തുന്നതിലൂടെ, വെബ് സൈറ്റിലെ എല്ലാ പേജുകളിലെയും ടാഗുകളുടെ അവതരണ രീതി വ്യത്യാസപ്പെടുത്താനാകും. CSS രൈറ്റ്റൽ പൊതുവായി ഒരു സ്ഥലത്ത് എഴുതുന്നതിനാൽ HTML-ലും CSS-ലും വേർത്തിരിച്ച് നിർത്താൻ സാധിക്കുകയും അങ്ങനെ പരിപാലനം എളുപ്പമാവുകയും ചെയ്യുന്നു. കൂടാതെ വെബ് പേജുകളിലുള്ള ടാഗുകളുടെ രൈറ്റ്റൽ വിവരങ്ങൾ മാറ്റുമ്പോൾ ഒരു റീതിയിൽ ബന്ധപ്പെടുത്തി ക്രമീകരിച്ചിരിക്കുന്നതുകൊണ്ട് എളുപ്പത്തിൽ അത് മനസ്സിലാക്കാനാവുന്നു. ഈത് വെബ് പേജുകളുടെ ബൈറ്റുകളിലുള്ള വലുപ്പം കുറയ്ക്കുകയും അങ്ങനെ വേഗത്തിൽ വെബ് പേജുകൾ ബൈറ്റുകളിൽ ദൃശ്യമാക്കുകയും ചെയ്യുന്നു.

വ്യത്യസ്ത സ്ക്രീൻ വലുപ്പമുള്ള ഉപകരണങ്ങളായ ഡെസ്ക്ടോപ്പ് മോണിട്ടറുകൾ, ടാബ് കമ്പ്യൂട്ടറുകൾ, മൊബൈൽ ഫോൺകൾ തുടങ്ങിയവയിൽ വെബ് പേജുകൾ പ്രദർശിപ്പിക്കാൻ അനുയോജ്യമാക്കുന്നത് CSS ആണ്. ആനിമേഷനുകളും, പരസ്പരസംവേദനവും കൊണ്ടുവരാനാകുന്ന HTML-ൽ അടുത്ത പതിപ്പായ HTML5-ൽ ഉപയോഗിച്ചിരിക്കുന്നത് ജാവ സ്ക്രിപ്റ്റിനോടൊപ്പമുള്ള CSS ആണ്.

## നിങ്ങളുടെ പുരോഗതിയിൽക്കൂടി



- കോഡ് വ്യത്യാസപ്പെടുത്തുന്നത് വരെ ഉള്ളടക്കം മാറാതെ നിൽക്കുന്ന വെബ് പേജുകളെ \_\_\_\_\_ എന്ന് വിളിക്കുന്നു.
- ഡെയനാമിക് വെബ് പേജുകൾ നിർമ്മിക്കുന്നതിന് ഉപയോഗിക്കുന്ന രണ്ട് സാങ്കേതിക വിദ്യകളുടെ പേരേഴുതുക.
- സ്ക്രിപ്റ്റ് ഉൾക്കൊള്ളിക്കാൻ ഉപയോഗിക്കുന്ന ടാഗാണ് \_\_\_\_\_.
- ക്ലിക്ക് ഭാഗത്ത് സ്ക്രിപ്റ്റ് എഴുതുന്നതിൽ എത്രക്കിലും ഒരു ഉപയോഗമെഴുതുക.
- ഒരു ജാവ സ്ക്രിപ്റ്റ് ഫയലിൽ എക്സിക്യൂഷൻ \_\_\_\_\_ ആണ്.
- അജാക്സ് ഉപയോഗിക്കുന്നത് കൊണ്ടുള്ള നേടങ്ങൾ എന്തൊക്കെയാണ്?
- താഴെന്നിട്ടുള്ള സ്ക്രിപ്റ്റിൽ ഓഷകളെ ക്ലിക്ക് ഭാഗത്തെത്തന്നും സെർവർ ഭാഗത്തെത്തന്നുമായി തരംതിരിക്കുക.
- ജാവ സ്ക്രിപ്റ്റ്, PHP, ASP, VB സ്ക്രിപ്റ്റ് .asp ഫയലുകൾ \_\_\_\_\_ എന്ന വെബ് സെർവർ സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ചാണ് കമ്പയിൽ ചെയ്യുന്നത്.
- CSS പ്രാവർത്തികമാക്കുന്നതിനുള്ള വ്യത്യസ്തമാർഗ്ഗങ്ങൾ എഴുതുക.

## 4.7 HTML ലിവിൽ അടിസ്ഥാന ആഗ്രഹം (Basic concepts of HTML documents)

വെബ് പ്രോഗ്രാമുടെ നിർമ്മാണത്തിന് വ്യാപകമായി ഉപയോഗിക്കുന്ന ഭാഷയാണ് HTML. ഓരോ വെബ് പ്രോഗ്രാം ധമാർമ്മത്തിൽ ഒരു HTML ഫയലാണ്. ഹൈപ്പർ ടെക്നോളജി ലിവിൽ അടിസ്ഥാന ആഗ്രഹം (ഡോക്യുമെന്റുകൾ) നിർമ്മിക്കുന്നതിനുള്ള ഒരു കുട്ടം നിർദ്ദേശങ്ങൾ അടങ്കിയ ലളിതമായ വാചകങ്ങളാണ് (ടെക്നോളജികള്). ഓരോ HTML ഫയലും, ഈ നിർദ്ദേശങ്ങൾ HTML ടാഗുകൾ എന്ന് അറിയപ്പെടുന്നു. ഈ ടാഗുകൾ ഉപയോഗിക്കുമ്പോൾ നിർദ്ദേശങ്ങൾക്ക് കൂടുതൽ വ്യക്തതയുള്ളതാക്കാനായി ചില കീ-വേഡ്യുകൾ അവയോടൊപ്പം ചേർക്കുന്നു. ഈ വാക്കുകൾ ആട്ടിബ്യൂട്ടുകൾ എന്നറിയപ്പെടുന്നു. ഒരു വെബ് പ്രോഗ്രാം ഉള്ളടക്കം ബോസറിൽ എങ്ങനെ പ്രവർശിപ്പിക്കണമെന്ന് തീരുമാനിക്കുന്നത് HTML ലിവിൽ തത്തിലുള്ള ടാഗുകളും ആട്ടിബ്യൂട്ടുകളും ഒരുമിച്ച് പ്രവർത്തിച്ചുകൊണ്ടാണ്. HTML ടാഗുകളെയും അവയുടെ ആട്ടിബ്യൂട്ടുകളെയും കൂറിച്ചുള്ള പഠനമാണ് ധമാർമ്മത്തിൽ HTML പഠനം. ടാഗുകളുടെയും ആട്ടിബ്യൂട്ടുകളുടെയും വിശദാംശങ്ങളിലേക്ക് കടക്കുന്നതിന് മുമ്പ് ഒരു HTML ലിവിൽ അടിസ്ഥാന ഘടന നമുക്ക് പരിശോധിക്കാം.

### 4.7.1 ഒരു HTML ലിവിൽ അടിസ്ഥാനഘടന (Basic structure of an HTML document)

ഉദാഹരണം 4.1-ൽ ഒരു HTML ലിവിൽ അടിസ്ഥാനഘടന കാണിച്ചിരിക്കുന്നു.

#### ഉദാഹരണം 4.1: HTML ലിവിൽ മാതൃക

```
<HTML>
  <HEAD>
    <TITLE> This is the title of web page </TITLE>
  </HEAD>
  <BODY>
    Hello, Welcome to the world of web pages!
  </BODY>
</HTML>
```

ഒരു ജോഡി ആംഗിൾ ബോക്കറൂക്കൾക്കുള്ളിലായി (<, >) വലിയ അക്ഷരങ്ങളിലുള്ള ചില വാക്കുകൾ നിങ്ങൾക്ക് കാണാം. ഈ HTML ടാഗുകളാണ്. HTML ടാഗുകൾ വലിയ അക്ഷരങ്ങളിലേബാത്തണമെന്ന് നിർബന്ധമില്ല. HTML-ൽ വലിയ അക്ഷരവും ചെറിയ അക്ഷരവും തമ്മിൽ വ്യത്യാസമില്ല (കേൽസ് സെൻസിറ്റീവ് അല്ല). നമുക്ക് വലിയ അക്ഷരത്തിലോ, ചെറിയ അക്ഷരത്തിലോ അവ കൂട്ടികലർത്തിയോ ഉപയോഗിക്കാവുന്നതാണ്. മറ്റ് വാക്കുകളിൽ നിന്നും അല്ലെങ്കിൽ ചിഹ്നങ്ങളിൽ നിന്നും വേർത്തിരിച്ചറിയാനായി, HTML ടാഗുകളും അവയുടെ ആട്ടിബ്യൂട്ടുകളുടെ ആദ്യക്ഷരവും വലിയ അക്ഷരങ്ങളായും മറ്റൊള്ളവ ചെറിയ അക്ഷരങ്ങളായുമാണ് ഈ പുസ്തകത്തിൽ ഉപയോഗിച്ചിട്ടുള്ളത്.

ഉദാഹരണം 4.1-ൽ കാണിച്ചിരിക്കുന്നത് പോലെ, എല്ലാ HTML പേജുകളും തുടങ്ങുന്നത് <HTML> ടാഗിലും അവസാനിക്കുന്നത് </HTML> ടാഗിലുമായിരിക്കും. ഒരു HTML ലിഖിതത്തെ ഹൈ ഭാഗമെന്നും ബോധി ഭാഗമെന്നും റണ്ടായി തിരിച്ചിരിക്കുന്നു. ഹൈ വിശദീകരിക്കുന്നതിന് <HEAD> ടാഗുപയോഗിക്കുന്നു. വെബ്പ് പേജിന്റെ ശൈർഷകം ഉൾപ്പെടെ ലിഖിതത്തെക്കുറിച്ചുള്ള വിവരങ്ങൾ ഹൈ ഭാഗത്തുണ്ടാകും. ബോധാർ വിൻഡോയിലെ ടെറ്റിൽ ബാറിൽ കാണിക്കുന്ന വെബ്പ് പേജിന്റെ ശൈർഷകം നിർണ്ണയിക്കുന്നത് <TITLE> ടാഗ് ഉപയോഗിച്ചുകൊണ്ടാണ്. ബോധി ഭാഗം നിർവ്വചിക്കുന്നതിന് <BODY> ടാഗ് ഉപയോഗിക്കുന്നു. ബോധി ഭാഗത്തുള്ളത് വെബ്പ് പേജിന്റെ ഉള്ളടക്കമായിരിക്കും. മുകളിൽ തന്നിട്ടുള്ള HTML ലിഖിതം ബോധാർ വിൻഡോയിൽ കണ്ടാൽ ചീതം 4.15-ൽ കാണുന്നത് പോലുണ്ടാകും.



**ചിത്രം 4.15 :** വെബ് ബോസർ ഉപയോഗിച്ച് തുറന്ന ഒരു വെബ് പേജ്



**வினியில் HTML படிப்புக்கால் :** 1991 ஏழ் அவ்வாணம் டி வெர்னன்சன் மீ (Tim Berners Lee) அன்று HTML நிர்மித்து. என்னால், நிலவாரமுத்து அடுத்ததை HTML படிப்பாய் HTML 2.0 பிரஸிலுக்கிட்டு 1995-லான். பழுவ பிரஸிலுக்கிட்டு HTML 4.01 பிரஸிலுக்கிட்டு 1999-க்குத் தொகை அவ்வாணமான். 2012-லான் ஏற்றுவும் புதிய படிப்பாய் HTML 5 புரித்திருக்கிறது. இது கூடுதல் கூடுதல் தூசுகளையும் உலகப்படித்திருக்கிற ஒரேஒரு பரிசுக்கிட்டுக்கொண்டிருக்கிறது.

#### 4.7.2 HTML ലിഖിത്തൈലെ ടാഗുകൾ (Tags in HTML document)

മുൻപ് സൂചിപ്പിച്ചത് പോലെ, വൈബ് പേജുകളുടെ ഉള്ളടക്കം ഏത് രീതിയിൽ ക്രമീകരിക്കണമെന്നും പ്രദർശിപ്പിക്കണമെന്നും ബേഖസിനോക് പഠയുടെ നിർവ്വേശങ്ങളാണ് HTML ടാഗുകൾ. എല്ലാ ടാഗുകൾക്കും <, > എന്നീ ആക്ഷിൾ ബൊധ്യക്കരുകളിലടക്കം ചെയ്ത ടാഗ് നാമമുണ്ടായിരിക്കും.

#### **4.7.3 കണ്ടയൻ ടാഗുകളും എംറി ടാഗുകളും (Container tags and empty tags)**

അയികമായുണ്ടായിരിക്കും. തുടങ്ങുന്ന കാഗ്യം അവസാനിപ്പിക്കുന്ന കാഗ്യം ആവശ്യമുള്ള കാഗ്യകളാണ് കണ്ടെത്തൻ കാഗ്യകൾ. ഒരു സൈക്കണ്ടിലേക്ക് ഉപയോഗിക്കാവുന്നവയാണ് കണ്ടെത്തൻ കാഗ്യകൾ. സൈക്കണ്ട് തുടക്കത്തിൽ ഓപ്പൺിംഗ് കാഗ്യം അവസാനം ഫ്രോസിംഗ് കാഗ്യം നൽകുന്നു. ഉദാഹരണമായി, ഒരു HTML ലിഖിതത്തിൽ തുടക്കവും ഒന്തുകവും ക്രമപൂട്ടുത്തുന്ന കാർഡ് ജോഡികളാണ് <HTML>, </HTML> എന്നിവ.

ചില ടാഗുകൾ ഈ രിതിയിലല്ല ഉപയോഗിക്കുന്നത്. അതരം ടാഗുകൾക്ക് ക്ഷോസിൽ ടാഗ് കാണുകയില്ല. അവ എറ്ററി ടാഗുകൾ എന്നറിയപ്പെടുന്നു. ഈ ടാഗുകൾ ഒരു സൈക്സ് നിലേക്കുള്ളവയല്ല. <BR>, <HR>, <IMG> തുടങ്ങിയവ എറ്ററി ടാഗുകൾക്ക് ഉദാഹരണം ആജ്ഞാബന്ധം. ഈ അധ്യായത്തിലെ തുടർന്നുള്ള ഭാഗങ്ങളിൽ ഇതരം ടാഗുകൾ നമുക്ക് കാണാനാകും.

#### 4.7.4 ടാഗുകളുടെ അടിബന്ധങ്ങൾ (Attributes of tags)

നിരു, അളവ്, സ്ഥാനം, പൊരുത്തപ്പടൽ, അല്ലകിൽ മറ്റ് ആകാരങ്ങൾ തുടങ്ങിയ അധികവിവരങ്ങൾ വെബ് ബേഹസിന് നൽകാനായി ചില ഘടകങ്ങൾ ഓപ്പണിങ് ടാഗിനുള്ളിൽ നൽകുന്നത് പതിവാണ്. ഈ ഘടകങ്ങളെ ആട്ടിബൃട്ടുകൾ എന്ന് വിളിക്കുന്നു. ഭൂതികാശം ആട്ടിബൃട്ടുകൾക്കും ഒരു വില നൽകേണ്ടത് ആവശ്യമാണ്. HTML-ൽ ഏക ഉദാഹരണികൾക്കുള്ളിലോ ജോഡികളായ ഉദാഹരണികൾക്കുള്ളിലോ ആണ് വില നൽകുന്നത് (അതായത്, ആട്ടിബൃട്ട് = ‘വില’ അല്ലകിൽ ആട്ടിബൃട്ട് = “വില”). ഉപയോഗിക്കാവുന്ന ഒരു കുടം ആട്ടിബൃട്ടുകൾ ഓരോ ടാഗിനോടൊപ്പവും ഉണ്ടായിരിക്കും. ഒരു ആട്ടിബൃട്ട് ഉപയോഗിക്കുകയാണെങ്കിൽ, സാധാരണയായി ടാഗ് നാമത്തിനുശേഷം ഒരു സ്വർപ്പം കഴിഞ്ഞാകും അത് നൽകുന്നത്. ഓലിവയികം ആട്ടിബൃട്ടുകൾ ഉപയോഗിക്കുമോൾ, അവയുടെ ക്രമത്തിന് പ്രാധാന്യം നൽകേണ്ടതില്ല.

ଉଦ୍‌ବାହରଣମାତ୍ରୀ, ଏବୁ ବେଳୀ ପେଜିଲେ ପଶୁଆତଲାଗିଙ୍କ ମନ୍ତର୍ୟାକ୍ଷର ମାଧ୍ୟାନ୍ତିକ <BODY Bgcolor = "Yellow">. ହୁଵିଦ �Bgcolor ଏବନ୍ତ କାଗିରେ ଅନ୍ତିବ୍ୟୁକ୍ତିକୁ ବ୍ୟାପାରିବାକୁ ଏବନ୍ତ ଅତିରେ ବିଲାଯୁମାନ. କାଗୁକଳ୍ପିତ ଅବସ୍ଥାରେ ଅନ୍ତିବ୍ୟୁକ୍ତିକଳ୍ପିତ ତୁଳିନ ବରୁଣ ଭାଗଙ୍କାନ୍ତିରେ ନାହିଁ ଚର୍ଚ ଚେଯୁଗନତାରେ.

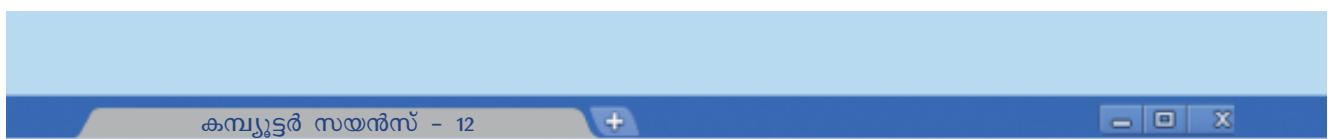
#### 4.7.5 HTML എലമെന്റുകൾ (HTML Elements)

ଓৱ জেৱাৰি টাৰুকভূম অৱত্যকিতায়ি উৰিকোষ্ঠিচীড়ুজ্জ উজ্জিঞ্চকবৰ্বুং প্ৰেৰণ  
তাৰে এৱলমৰ্দি. চিৰে 4.16-ত কাৰণিচীৰিককুনত পোল, বোয়া এৱলমৰ্দিৰ  
ওপুণিৰ টাৰায <BODY>  
যুং কেুৰাসীৰ টাৰায </  
BODY> যুং অৱত্যকিতায  
লাৰ্জ ইজিকবৰ্বুং ওলোকো.

Element
<BODY> Welcome to HTML </BODY>

The diagram illustrates the structure of an HTML document. At the top, the word "Element" is centered above a horizontal line. Below this line, the text "<BODY> Welcome to HTML </BODY>" is written in a monospaced font. The word "Welcome" is positioned between the opening tag "<BODY>" and the closing tag "</BODY>". To the left of "Welcome", the label "Opening tag" is placed below the line. To the right of "Welcome", the label "Closing tag" is placed below the line.

ചിത്രം 4.16 : ഒരു HTML ഘടകം

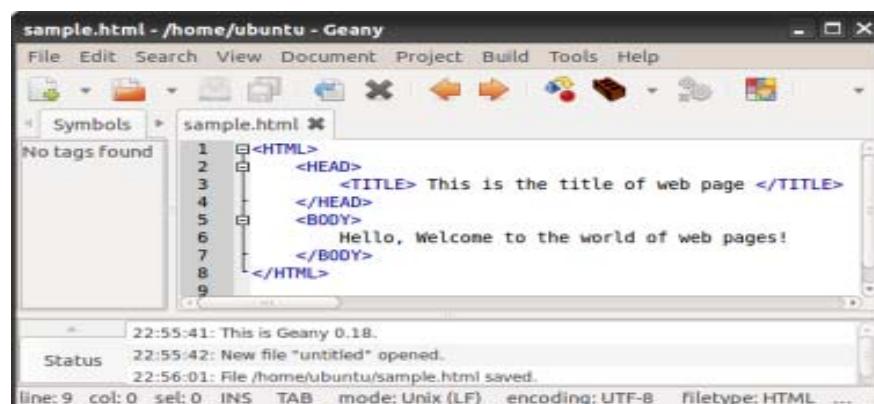


ഒരു HTML ലിവിത്തിന്റെ അടിസ്ഥാനപദ്ധതിൽ നാല് ജോഡി ടാഗുകളുണ്ടാകും. അവയാണ് :

```
<HTML>      </HTML>
<HEAD>      </HEAD>
<TITLE>      </TITLE>
<BODY>      </BODY>
```

## 4.8 ഒരു HTML ലിവിതം നിർമ്മിക്കുന്നു (Creating an HTML document)

ഉദാഹരണം 4.17-ൽ തന്നിട്ടുള്ള HTML കോഡ് ഉപയോഗിച്ച് ലഭിതമായ ഒരു വെബ് പേജ് നമുക്ക് നിർമ്മിക്കാം. ജിനി (Geany), ജിഫൈറ്റ് (Gedit), ടെക്സ്റ്റ്‌പാഡ് (TextPad), നോട്ട് പാഡ് (NotePad), നോട്ട്‌പാഡ്++ (NotePad++) തുടങ്ങിയ ടെക്സ്റ്റ് എഡിറ്ററുകളിൽ ഏതെങ്കിലും ഉപയോഗിച്ച് HTML ലിവിതം നിർമ്മിക്കാം. ഒരു ഫയൽ നാമവും എക്സെഞ്ചർ ആയി .htm അല്ലെങ്കിൽ .html എന്നും നൽകി ഫയൽ സേവ് ചെയ്യണം (ഉദാഹരണമായി, Sample.html). ജിനി എഡിറ്ററിൽ Sample.html എന്ന പേരിൽ സേവ് ചെയ്തിട്ടുള്ള HTML കോഡ് നമുക്ക് ചിത്രം 4.17-ൽ കാണാം.



ചിത്രം 4.17 : ജിനി എഡിറ്ററിലുള്ള HTML കോഡ്

## ഒരു HTML ലിവിതം ബ്രൗസറിലുണ്ട് കാണുന്നു (Viewing an HTML document in a Browser)

HTML ലിവിതം തയാറാക്കിക്കഴിത്താൽ, അത് ബ്രൗസർ ഉപയോഗിച്ച് കാണാവുന്നതാണ്. മോസില്പ് ഫയർഫോക്സ് (Mozilla Firefox), ഗൂഗിൾ ക്രോം (Google Chrome), ഇൻറെന്റ് എക്സ്‌പ്ലോറർ (Internet Explorer), നെറ്റ്സൈപ്പ് നാവിഗേറ്റർ (Netscape Navigator) തുടങ്ങിയ ധാരാളം ബ്രൗസറുകളുണ്ട്. വെബ് പേജ് മോസില്പ് ഫയർഫോക്സിൽ തുറന്നപ്പോൾ കിട്ടിയ സ്ക്രീൻ ചിത്രങ്ങളാണ് ഈ പുസ്തകത്തിൽ തന്നിട്ടുള്ളത്. ജിനി യുടെ ടുശ്ബാവിലുള്ള Execute (എക്സിക്യൂട്ട്) ബട്ടൺ ഫീക്സ് ചെയ്യുന്നോ, HTML കോഡിന്റെ ഒരു പുറ്റ് ലഭിക്കും.

## 4.9 പ്രധാനപ്പെട്ട HTML ടാഗുകൾ (Essential HTML tags)

எனவே வெப்ப பேஜ் நிர்மிக்கும்பொது அவசியம் வேண்ட சாலைகளைக்கூரிச் சுமூகம் மறையும் லாக்கோ. சாலைகள், அவசியம் உபயோகம், பயிர்ப்புகிடக்கூடும் ஆட்டிப்பூடுகளும் அவசியமாக விலக்கலும், பூங்கள் விஸ்தேயில் அவசியமாக ரூபா தூண்ணியவ ஹவிடெ விழடுகளிக்கூடும்.

#### 4.9.1 <HTML>-െല്ലാ HTML പേജ് തുടങ്ങുന്നത് (<HTML> - Starting an HTML page)

HTML ലിഖിതം പുർണ്ണമായി ഒരു ജോഡി <HTML>, </HTML> എന്നീ ടാഗ് ജോഡി കൾക്കുള്ളിൽ വലയം ചെയ്തിരിക്കുന്നു. ഒരു HTML ലിഖിതം തിരിച്ചറയാനാണ് <HTML> ടാഗ് ഉപയോഗിക്കുന്നത്. പൊതുവായിപ്പറഞ്ഞാൽ ഒരു HTML പേജിൽ <HTML> ആദ്യ ടാഗും </HTML> അവസാന ടാഗുമായിരിക്കും. വെബ്സ് പേജിലെ മറ്റൊന്നും ഈ രണ്ട് ടാഗുകൾക്ക് ഇടയിലായിരിക്കും. അതായത് ഹെഡ് ഭാഗവും ബോധി ഭാഗവും <HTML>, </HTML> എന്നീ ടാഗുകൾക്കിടയിലായിരിക്കും ഉൾക്കൊള്ളിച്ചിരിക്കുന്നത്. ഇത് ഒരു ജോഡി ക്ലോസ്റ്റർ ടാഗുകളാണ്. **Dir**, **Lang** എന്നിവയാണ് <HTML> ടാഗിൽ പ്രധാന ആട്ടിവ്യൂട്ടുകൾ.

Dir

വൈഡ് പ്രോജക്റ്റ് പ്രദർശിപ്പിക്കുന്ന (കാണിക്കുന്ന)

Sl. No.	Code	Language	സൂചിപ്പിക്കുന്നതിനാണ് r ആട്ടിബൃംഗ് ഉപയോഗിക്കു രേറ്റ് - ഇടത് നിന്ന് ദു ലൈഫ്റ്റ് - വലത് നിന്ന് ലകളാണ് ഈ ആട്ടിബൃംഗിടിരേ തന്ത്രവില വില 1tr എം അവതരിപ്പിക്കുന്നതിന് പനിംസ് തുടങ്ങിയ ഭാഷകൾ എൽ എന വില നൽകണം. L Dir = "rtl" > എന്ന വശത്ത് നിന്നും ഇടത് വശന് സൂചിപ്പിക്കുന്നു.
1	En	English	
2	F r	French	
3	D e	German	
4	I t	Italian	
5	E l	Greak	
6	E s	Spanish	
7	A r	Arabic	
8	J a	Japanese	
9	H i	Hindi	
10	R u	Russian	

#### പട്ടിക 4.3: പ്രചാരത്തിലുള്ള ചില ഭാഷകളും അവയുടെ കോഡുകളും

ലിഖിതത്തിനുള്ളിൽ പൊതുവായി ഉപയോഗിക്കുന്ന ഭാഷയെ സുചിപ്പിക്കുന്നതിന് <HTML> ടാഗിന്റെ **Lang** അട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. “en” എന്നത് ഇംഗ്ലീഷ് ഭാഷയെയും “it” എന്നത് ഇറ്റാലിയൻ ഭാഷയെയും സുചിപ്പിക്കുന്നു.

ଉଡାହରଣମାତ୍ର ଏକ `<HTML Lang = "ar">` ଏଣ କୋଣ ଅବଲିକ ଭାଷାଯିଲାଙ୍କ HTML ଲିଖିତମାତ୍ର ତିଥିକୁ ଉପରେ ଲାଗୁ କରିବାକୁ ପାଇଁ ପରିଚୟ ଦିଆଯାଇଛି।



#### 4.9.2 <HEAD> - ഹെഡ് ഭാഗം നിർമ്മിക്കുന്നു (<HEAD> - Creating head)

ശീർഷകം, ഉപയോഗിച്ചിട്ടുള്ള സ്ക്രിപ്റ്റുകൾ, ശൈലി നിർവ്വചനങ്ങൾ തുടങ്ങിയ ലിഖിതത്തെ കുറിച്ചുള്ള വിവരങ്ങൾ HTML ലിഖിതത്തിൽ ഹൈഡ്രോഗത്ത് ഉൾപ്പെടുത്തുന്നു. ഹൈഡ്രോഗം പ്രസ്താവിക്കുന്ന ടാഗ് ജോഡികളാണ് <HEAD>, </HEAD> എന്നിവ. ഇതും ഒരു ക്ലെറ്റയ്ക്കൻ ടാഗ് ജോഡിയാണ്.

#### **4.9.3 <TITLE> - ശൈറ്റുകം നിർമ്മിക്കുന്നു (<TITLE> - Creating a title)**

#### 4.9.4 <BODY>-വെബ് പ്രോഗ്രാമിംഗ് ഒരു സോഫ്റ്റ്‌വെയർ നിർമ്മിക്കുന്നത് (<BODY>- Creating a body)

ലിവിത്തിന്റെ ബോധി ഭാഗം വിവർക്കുന്നതിന് ഉപയോഗിക്കുന്ന ബോധി ടാഗ് ജോധി കളാണ് <BODY>, </BODY> എന്നിവ. പ്രൗഢാർ വിസ്തേയായിൽ പ്രദർശിപ്പിക്കേണ്ട ഉള്ളടക്കം ഈ ഭാഗത്ത് ഉൾക്കൊള്ളിക്കുന്നു. ലിവിത്തിന്റെ ഉള്ളടക്കം നിർവ്വചിക്കുന്ന മറ്റൊരു ടാഗുകളും ബോധി ഭാഗത്താണ് ഉൾപ്പെടുത്തുന്നത്. ഈ ടാഗുകളെക്കുറിച്ച് വിശദിക്കാനുള്ള ഒരു ടാഗിന്റെ അടിബന്ധനകൾ ആവശ്യമാണ്.

## Background

வெவ்வேறு பேஜிக் டிராஸ்யூட்டில் சிறுதான் நிலைகளைப் போன்று வெளியிடப்பட்டிருக்கின்றன. அதிலே மிகவும் பொருளான தீவிரமான பேஜிகள் எடுத்து விடப்பட்டிருக்கின்றன.

```
<BODY Background = "URL of the picture">
```

ആകാശത്തിന്റെ ചിത്രം ഒരു വെബ് പേജിന് പദ്ധതിലെമായി നൽകുന്നതിനുള്ള HTML കോഡ് ഉഭാവരണം 4.2-ൽ കാണിച്ചിരിക്കുന്നു.

**ഉദാഹരണം 4.2:** ഒരു വെബ് പ്രേജിന് പത്രാത്തലമായി ചിത്രം നൽകുന്നു.

```
<HTML>
<HEAD>
    <TITLE> Background Image </TITLE>
</HEAD >
<BODY Background = "Sky.jpg">
    Hello, Welcome to the world of Web Pages!.....
</BODY>
```

#### 4. വെബ് സാങ്കേതിക വിദ്യ

</HTML>

ഉദാഹരണം 4.1-ൽ ഉള്ള HTML കോഡിലെ <BODY> സാഗിന് ഒരു **Background** ആട്ടിബ്യൂട്ടും അതിന് വിലയായി "Sky.jpg" എന്നും നൽകിക്കൊണ്ട് <BODY Background="Sky.jpg"> എന്ന വ്യത്യാസപ്പെടുത്തിയിരിക്കുന്നു. വെബ് പേജ് തുറക്കുന്നതിന് മുമ്പ് ചിത്രമയൽ ഇപ്പോൾ ഉപയോഗിച്ചുകൊണ്ടിരിക്കുന്ന ഫോർഡിൽ/ഡയറക്ടറിൽ ഉൾപ്പെടുത്തേണ്ടതാണ്. ചിത്രം 4.18-ൽ കാണിച്ചിരിക്കുന്നത് പോലെ വെബ് പേജ് പ്രദർശിപ്പിക്കാവുന്നതാണ്.

### Bgcolor

ലിവിത്തതിന്റെ ബോധി ഭാഗത്തെ പശ്വാതലവത്തിന് ഒരു നിറം നിർദ്ദേശിക്കുന്ന ആട്ടിബ്യൂട്ടാണിത്. ഉദാഹരണമായി, <BODY Bgcolor = "grey"> എന്ന കോഡ് പശ്വാതലവത്തിന് ചാരനിറം നൽകുന്നു.

Bgcolor ആട്ടിബ്യൂട്ടിന് രണ്ട് രീതിയിൽ വില നൽകാനാകും.

- Color\_name** - പശ്വാതലവത്തിന് നിറ നാമം നിർദ്ദേശിക്കുന്നു ("red", "grey" തുടങ്ങിയവ പോലെ).
- Hex\_number** - പശ്വാതലവത്തിന് പതിനാറ് അക്ക കോഡിലുള്ള നിറം നിർദ്ദേശിക്കുന്നു. ("#ff6080", "#303030" തുടങ്ങിയവ പോലെ). ഓരോ പതിനാറക്ക കോധും ഹാഷ് ചിഹ്നത്തിലാണ് ആരംഭിക്കുന്നത്. ആറുക്ക സംഖ്യയും അക്ഷരങ്ങളും ചേർന്ന നിറങ്ങൾക്ക് RGB (ചുവപ്പ്, പച്ച, നീല) വിലകൾ നൽകി പ്രദർശിപ്പിക്കുന്നു. ആർ അക്കങ്ങളിൽ ആദ്യ രണ്ടുണ്ണം ചുവപ്പിൽ അളവിനെയും, അടുത്ത രണ്ട് അക്കങ്ങൾ പച്ചയും, അവസാനത്തെ രണ്ടുക്കങ്ങൾ നീലയും അളവിനെയും 00 മുതൽ FF വരെയുള്ള വിലകൾ കൊണ്ട് സൂചിപ്പിക്കുന്നു. ഉദാഹരണമായി, #000000 എന്നത് കറുപ്പും, #FF0000 എന്നത് തിളക്കമുള്ള ചുവപ്പും, #00FF00 എന്നത് തിളങ്ങുന്ന പച്ചയും, #FFFFFF എന്നത് വെള്ളയുമായിരിക്കും (മുന്ന് നിറങ്ങളും പുർണ്ണമായും ഉൾക്കൊള്ളിച്ചിരിക്കുന്നു) പതിനാറക്ക കോഡിലെ വിവിധ നിറക്കുടുകൾ



ചിത്രം 4.18 : ഒരു പശ്വാതല ചിത്രം

Colour	Colour Name	Colour HEX
Black	Black	#000000
Red	Red	#FF0000
Green	Green	#00FF00
Blue	Blue	#0000FF
Yellow	Yellow	#FFFF00
Aqua	Aqua	#00FFFF
Grey	Grey	#C0C0C0
White	White	#FFFFFF

ചെബിൾ 4.4 : നിറങ്ങളുടെയും, അവയുടെ പേരിന്റെയും, 16 അക്ക കോഡ് വിലയുടെയും പട്ടിക

നമുക്ക് പരീക്ഷിക്കാവുന്നതാണ്. ചില നിറങ്ങളും, അവയുടെ പേരുകളും, പതിനൊരു കോഡ് വിലകളും പട്ടിക 4.4-ൽ കാണിച്ചിരിക്കുന്നു.

### Text

പേജിന്റെ ഉള്ളടക്കത്തിലുള്ള വാക്കുങ്ങളുടെ നിറം സൂചിപ്പിക്കുന്ന ആട്ടിബ്യൂട്ടാണ് ഈ. തന്ത്ര രൂപത്തിൽ വെള്ള/ചാര നിറമുള്ള പശ്ചാത്തലവും വാക്കുങ്ങൾക്ക് കറുപ്പ് നിറ വുമായി ഭേദസർ കാണിക്കുന്നു. Bgcolor ആട്ടിബ്യൂട്ട് ഉപയോഗിച്ച് പശ്ചാത്തല ത്തിന്റെ നിറം എങ്ങനെ വ്യത്യാസപ്പെടുത്താമെന്ന് നാം ചർച്ച ചെയ്ത് കഴിഞ്ഞു. അതു പോലെ, Text ആട്ടിബ്യൂട്ടുപയോഗിച്ചുകൊണ്ട് വാക്കുങ്ങളുടെ നിറം വ്യത്യാസപ്പെടുത്താനാകും. ഉദാഹരണമായി, <BODY Text = "yellow"> എന്നത് മണ്ണ നിറത്തിലുള്ള വാക്കുത്തെ പ്രദർശിപ്പിക്കുന്നു. Bgcolor-ന് വില നൽകിയത് പോലെ നിറത്തിന്റെ പേരോ പതിനാറു കോഡോ ഉപയോഗിച്ച് Text ആട്ടിബ്യൂട്ടിന് വില നൽകാവുന്നതാണ്. ഉദാഹരണമായി Text = "Blue" അല്ലെങ്കിൽ Text = "#00FFDD" തുടങ്ങിയവ.

### Link, Alink and Vlink

ക്ലിക്ക് ചെയ്യുന്നോൾ, ഒരു ലിഖിതത്തിൽ നിന്ന് മറ്റാന്നിലേക്കോ, അല്ലെങ്കിൽ അതേ ലിഖിതത്തിന്റെ വോറാരു ഭാഗത്തെക്കോ നീങ്ങാൻ സഹായിക്കുന്ന ഒരു ഘടകം/ഒരു വാക്യം/ഒരു ചിത്രമാണ് ഹൈപ്പർലിങ്ക്. ഒരു ലിഖിതത്തെ മൊത്തമായോ അല്ലെങ്കിൽ ലിഖിതത്തിലുള്ള പ്രത്യേക ഭാഗത്തെയോ ആണ് ഹൈപ്പർ ലിങ്ക് സൂചിപ്പിക്കുന്നത്. ഹൈപ്പർ ലിങ്കിനെക്കുറിച്ച് വിശദമായി പിന്നീട് നമുക്ക് ഈ അധ്യായത്തിൽ ചർച്ച ചെയ്യാം.

**Link:** കാണുന്നയാൾ സന്ദർശിക്കാത്ത ഹൈപ്പർ ലിങ്കിന്റെ നിറം സൂചിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്ന ആട്ടിബ്യൂട്ടാണ് ഈ. Link ആട്ടിബ്യൂട്ടിന്റെ തന്ത്ര നിറം നീലയായിരിക്കും.

**Alink:** സജീവമായ ഹൈപ്പർ ലിങ്കിന്റെ നിറമാണ് ഈ സൂചിപ്പിക്കുന്നത്. മൗസ് അതി നുമേൽ ക്ലിക്ക് ചെയ്യുന്ന നിമിഷം മാത്രമാണ് ആ ലിങ്ക് സജീവമാകുന്നത്. ആയ തിനാൽ, തിരഞ്ഞെടുക്കുന്നോൾ ഹൈപ്പർ ലിങ്കിന്റെ നിറം Alink എന്ന് വില സൂചിപ്പിക്കും. ഇവിടെ ഹൈപ്പർ ലിങ്കിന്റെ തന്ത്ര നിറം പച്ചയായിരിക്കും.

**Vlink:** കാണുന്നയാൾ സന്ദർശിച്ച ഹൈപ്പർ ലിങ്കിന്റെ നിറം സൂചിപ്പിക്കാനാണ് ഈ നൽകുന്നത്. Vlink എന്ന് തന്ത്ര നിറം ധൂമനിറം (പർപ്പിൾ) ആയിരിക്കും.

### Leftmargin, Topmargin

പേജിന്റെ അരികിൽ നിന്നുള്ള ശുന്നസ്ഥലത്തെയാണ് മാർജിന് സൂചിപ്പിക്കുന്നത്. വിന്റോഡൈൽ ലിഖിതത്തിന്റെ ഇടത് ഭാഗത്ത് കുറച്ച് ശുന്നപ്രദേശം ഒഴിച്ചിടാനായി Leftmargin ആട്ടിബ്യൂട്ടം മുകളറുത്ത് കുറച്ച് ശുന്ന സ്ഥലം നൽകാനായി Topmargin ആട്ടിബ്യൂട്ടം ഉപയോഗിക്കുന്നു. വിലനൽകുന്നത് പിക്സൽ വോയിൽുകളുടെ എണ്ണമായാണ്.

ഉദാഹരണമായി, <BODY Leftmargin = "60" Topmargin = "70"> എന്നത്,

#### 4. വെബ് സാങ്കേതിക വിദ്യ

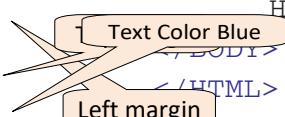
പ്രദർശിപ്പിക്കേണ്ട വാചകം താഴീരെ ഇടത്തെ അറുത്ത് നിന്ന് 60 പിക്സൽ പോയിരും അകലെയും, മുകളിൽത്ത് നിന്ന് 70 പിക്സൽ പോയിരും അകലെയും ആരണ്ണന് കാണിക്കുന്നു. **<BODY>** ടാഗിൽ Bgcolor, Text, Topmargin, Leftmargin എന്നി ആടിബ്യൂട്ടുകൾ ഉദാഹരണം 4.3-ലെ കോഡിലൂടെ വിശദിക്കിയിരിക്കുന്നു. ചിത്രം 4.19-ൽ അനുബന്ധ വെബ് പേജ് കാണാവുന്നതാണ്.



ചിത്രം 4.19 : BODY ടാഗിനോടൊപ്പം ആടിബ്യൂട്ടുകളുടെ ഉപയോഗം

#### ഉദാഹരണം 4.3: ഒരു വെബ് പേജിൽ പദ്ധതിയെത്തിന് നിറം കൊടുക്കുന്നു

```
<HTML>
<HEAD>
    <TITLE> This is the title of web page </TITLE>
</HEAD>
<BODY Bgcolor= "cyan" Text= " Blue"
      Topmargin= "70" Leftmargin= "60">
    Hello, Welcome to the world of Web Pages!.......
```



#### 4.10 പൊതുവായ ചില ടാഗുകൾ (Some common tags)

##### നിങ്ങളുടെ പുരോഗതി അറിയുക



1. HTML പ്രതിനിധികരിക്കുന്നത് \_\_\_\_\_ നേരാണ്.
2. എന്താണ് കണ്ണഡയ്ക്കൻ ടാഗ്?
3. അവസാനിപ്പിക്കുന്ന ടാഗ് ഇല്ലാതെ തുടക്ക ടാഗ് മാത്രം വേണ്ട ഇനം ടാഗാണ് \_\_\_\_\_.
4. ശരിയോ തെറ്റോ എന്ന് പ്രസ്താവിക്കുക.
  - a. ടാഗുകൾ കേയ്ക്ക് സെൻസിറ്റീവ് ആണ്.
  - b. <BODY> ടാഗിൽ ഒരു ആടിബ്യൂട്ടാണ് Bgcolor.
  - c. ഒരു എംറ്റി ടാഗാണ് <TITLE>.
  - d. <HEAD> ടാഗിൽ ഒരു ആടിബ്യൂട്ടാണ് Dir.
5. <HTML> ടാഗിൽ ആടിബ്യൂട്ടുകളുടെ പേരെഴുതുക.
6. ഒരു ടാഗിൽ ആടിബ്യൂട്ടുകളുടെ ഉപയോഗമെന്താണ്?
7. <BODY> ടാഗിൽ വിവിധ ആടിബ്യൂട്ടുകൾ എഴുതുക.



നിങ്ങളുടെ വിദ്യാലയത്തിൻ്റെ പേരും അതോടൊപ്പം പശ്ചാത്തലത്തിൽ വിദ്യാലയത്തിൻ്റെ ചിത്രവും പ്രദർശിപ്പിക്കുന്ന ഒരു HTML ലിബിറ്റം നിർമ്മിക്കുക. തുടർന്ന് വാക്കുത്തിൻ്റെ നിറവും പശ്ചാത്തലവും വ്യത്യാം നമുക്ക് ചെയ്യാം സാമ്പത്തി പേജ് പരിഷ്കരിക്കുക.

HTML ലിബിറ്റത്തിന് വേണ്ട അടിസ്ഥാന ടാഗുകളെയും അവയുടെ ആട്ടീസ്യൂട്ടുകളെയും കുറിച്ച് നാം ചർച്ച ചെയ്തു കഴിഞ്ഞു. വെബ് പേജിലെ ഉള്ളടക്കം കൂടുതൽ ആകർഷക മാക്കാൻ ഉപയോഗിക്കുന്ന മറ്റു ടാഗുകളുമുണ്ട്. ഇക്കുട്ടത്തിൽ, HTML ലിബിറ്റത്തിൻ്റെ ബോധി ഭാഗത്തെ ഉള്ളടക്കത്തിലൂള്ള വാക്കുങ്ങളുടെ ഘടന രൂപസ്ഫുട്ടത്താൻ ഉപയോഗിക്കുന്ന ടാഗുകളെ മോർമാറ്റിംഗ് ടാഗുകൾ എന്ന് വിളിക്കുന്നു. ബോധി ഭാഗത്തെ ഉള്ളടക്കത്തിൻ്റെ രൂപാലടന്ത്രങ്ങൾ ആവശ്യമായ ടാഗുകളിൽ ചിലത് നമുക്ക് പരിചയപ്പെടാം.

#### 4.10.1 <H1>, <H2>, <H3>, <H4>, <H5>, <H6> - ശൈർഷക ടാഗുകൾ

ഒരു വാണികയുടെ തുടക്കത്തിൽ അത് എന്നാണെന്ന് വിശദീകരിക്കുന്നതിനായി ഏകദൃശ്യമുണ്ട്. ഒരു വാക്കോ, വാക്യമോ, വാചകമോ ആണ് ശൈർഷകം. ബോധി ഭാഗത്തെ സാധാരണ വാക്കുങ്ങളെക്കാൾ വലുതോ കടുപ്പിച്ചതോ ആയ സവിശേഷതയുള്ള അക്ഷരങ്ങൾ കൊണ്ടാണ് ശൈർഷകം പ്രദർശിപ്പിക്കുന്നത്. <H1> മുതൽ <H6> വരെ ആർത്തലാം ലൂള്ള ശൈർഷകങ്ങൾ HTML-ൽ ഉണ്ട്. ഇവിടെ <H1> ഏറ്റവും വലുതും <H6> ഏറ്റവും ചെറുമായ ശൈർഷകങ്ങളാണ് നിർമ്മിക്കുന്നത്. ശൈർഷകം പ്രദർശിപ്പിക്കുന്നതിന് മുന്നിലും പിന്നിലുമായി ബേഹസർ ഓരോവരി ചേർക്കുന്നു. ഈ ടാഗുകൾ പ്രധാനപ്പെട്ട ആട്ടീസ്യൂട്ട് Align ആണ്. അതിന് നൽകാവുന്ന വിലകൾ താഴെ തന്മൂലമായി പറയുന്നതാണ്.

Left : വാക്യം ഇടത് അരികുമായി (മാർജിനുമായി) പൊരുത്തപ്പെട്ട് നിൽക്കും.

Right : വാക്യം വലത് അരികുമായി (മാർജിനുമായി) പൊരുത്തപ്പെട്ട് നിൽക്കും.

Center : വാക്യം പേജിൻ്റെ മാധ്യമാഗത്ത് പ്രത്യക്ഷപ്പെടും.

വിവിധങ്ങളായ ശൈർഷകങ്ങളും, പൊരുത്തപ്പെടലുകളും ഉദാഹരണം 4.4 - റൈറ്റിംഗ് റിഫറൻസിലോടൊപ്പം അനുബന്ധ വെബ് പേജ് ചിത്രം 4.20-ൽ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.

**ഉദാഹരണം 4.4:** വിവിധ ശൈർഷക ശൈലികൾ വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

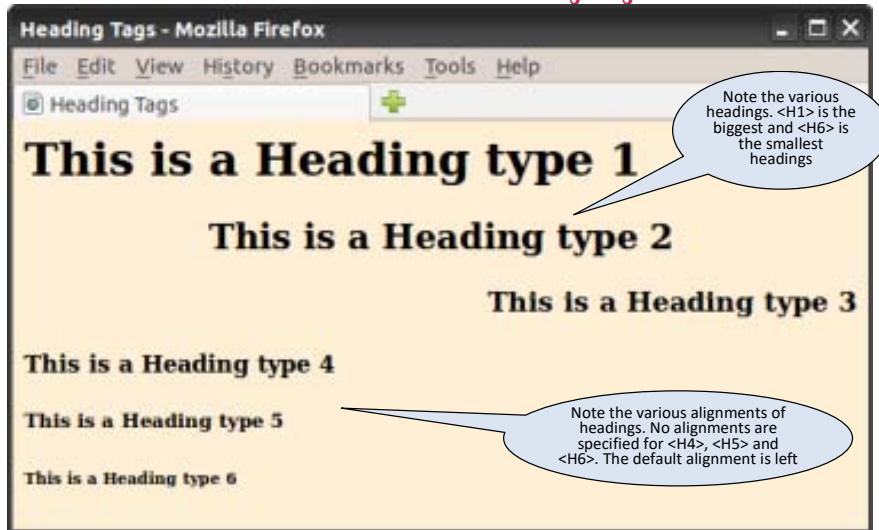
```
<HTML>
<HEAD>
    <TITLE> Heading Tags </TITLE>
</HEAD >
<BODY Bgcolor= "#FFEFDD">
    <H1 Align= "left"> This is a Heading type 1 </H1>
    <H2 Align= "center"> This is a Heading type 2 </H2>
    <H3 Align= "right"> This is a Heading type 3 </H3>
```

```

<H4> This is a Heading type 4 </H4>
<H5> This is a Heading type 5 </H5>
<H6> This is a Heading type 6 </H6>
</BODY>
</HTML>

```

#### 4.10.2 <P> ടാഗ് - വസ്തീകരണ നിർമ്മിക്കുന്നു



ചിത്രം 4.20 : വ്യത്യസ്ത റൈറ്റിംഗ് ടാഗുകളോടൊപ്പുള്ള വിവിധ പൊരുത്തപ്പെടലുകൾ

<BODY> ടാഗിലെ വാക്കുങ്ങളെ വിവിധ വസ്തീകരണിൽ ആക്കുവാൻ നമ്മുടെ സഹായിക്കുന്നത് <P> ടാഗാണ്. ഈ ഒരു പുതിയ വസ്തീകരണ സൂചിപ്പിക്കുകയും ഒരു ശൃംഖല വസ്തീകരിക്കാൻ മുന്നിൽ ചേർക്കാൻ ഭൗതികനാട്ട് നിർദ്ദേശിക്കുകയും ചെയ്യുന്നു. മറ്റ് ഏതൊരു വേർവ്വ് പ്രോസസറിലെ വസ്തീകരണ പോലെ തന്നെയാണ് HTML ലെ വസ്തീകരണം. ഒരു വസ്തീകരണ <P> ടാഗിൽ തുടങ്ങി </P> ടാഗിൽ അവസാനിക്കുന്നു. Left, right, center, justify എന്നീ വിലകൾ നൽകി വസ്തീകരിക്കിയാൽ വാക്കുങ്ങളുടെ പൊരുത്തപ്പെടൽ ഉറപ്പിക്കുന്നത് Align ആട്ടിബ്യൂട്ട് ഉപയോഗിച്ച് കൊണ്ടാണ് വ്യത്യസ്ത പൊരുത്തപ്പെടൽ കുടിയിൽ നൽകി വസ്തീകരണ എങ്ങനെ രൂപപ്പെടുത്തുന്നുവെന്ന് ഉദാഹരണം 4.5-ലുള്ള കോഡിൽ തന്നിരിക്കുന്നു. അങ്ങനെ ലഭിക്കുന്ന വെബ് പേജ് ചിത്രം 4.21-ൽ കാണിച്ചിരിക്കുന്നു.

**ഉദാഹരണം 4.5:** വിവിധ പൊരുത്തപ്പെടലുകളോട് കൂടിയ വസ്തീകരണ രൂപപ്പെട്ട തുട്ടിനിന്ന് വേണ്ടിയുള്ളത്

```

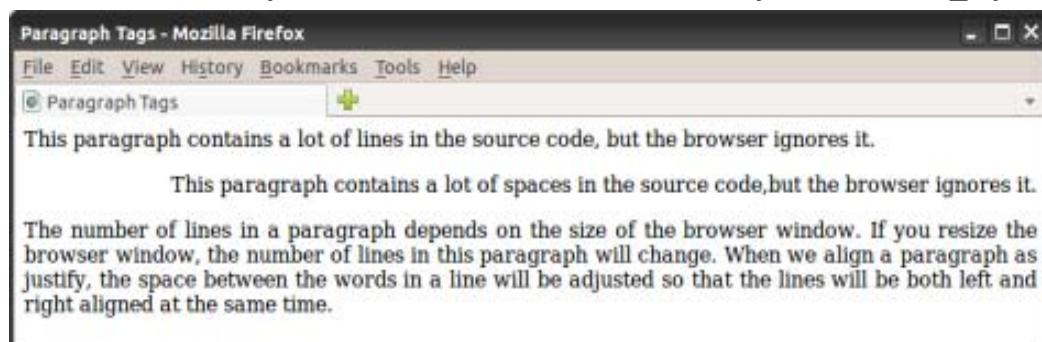
<HTML>
<HEAD>
    <TITLE> Paragraph Tags </TITLE>
</HEAD>
<BODY>

```



```
<P>
This paragraph contains
a lot of lines in the source code,
but the browser ignores it.
</P>
<P Align= "right">
This paragraph contains a lot of spaces
in the source code, but the browser ignores it.
</P>
<P Align= "justify">
The number of lines in a paragraph depends on
the size of the browser window.
If you resize the browser window, the number of lines
in this paragraph will change.
When we align a paragraph as justify, the space between
the words in a line will be adjusted
so that the lines will be both left and right aligned
at the same time.
</P>
</BODY>
</HTML>
```

ചിത്രം 4.5-ലെ സോഴ്സ് കോഡിൽ അധിക ശുന്ധസ്ഥലങ്ങളോട് കൂടിയ മൂന്ന് വരികൾ ഒന്നാം വണ്ണിക്കയില്ലോ, രണ്ട് വരികൾ രണ്ടാം വണ്ണിക്കയില്ലോ ഉൾക്കൊള്ളിച്ചിട്ടുള്ളത്



ചിത്രം 4.21 : Align ആട്ടിവ്യൂട്ട് ഉൾക്കൊള്ളിച്ചിട്ടുള്ള <P> ടാഗിന്റെ ഉപയോഗം

നമുക്ക് കാണാനാകും. എന്നാൽ ചിത്രം 4.21-ൽ കാണിച്ചിട്ടുള്ള ഈ കോഡിന്റെ വെബ് പേജിൽ ആദ്യത്തെ രണ്ട് വണ്ണികകൾക്ക് ഓരോ വരിയാണുള്ളത്. അതായത്, പേജ് പ്രദർശിപ്പിക്കുമ്പോൾ, അധികമുള്ള വരികളും ശുന്ധസ്ഥലങ്ങളും ഭേദഗതി നീക്കം ചെയ്യുന്നു. രണ്ടാം വണ്ണിക വലതുവശത്ത് പൊരുത്തപ്പെട്ട് നിൽക്കുന്നതും മുന്നാം വണ്ണിക ഇരുവശങ്ങളിലും പൊരുത്തപ്പെട്ട് നിൽക്കുന്നതും ശമ്പിക്കുക. ഒന്നിലധികം ശുന്ധസ്ഥലങ്ങളും വരികളും ഒരോറും ശുന്ധസ്ഥലമായാണ് കരുതുന്നത്. അതുകൊണ്ട്,

HTML കോഡിൽ അധികമായി ശുന്ത്യസഹായമായി ഉപയോഗിച്ചുള്ള ഒരു കോഡിന്റെ പേരാണ് <BR>. ടാഗ് ഉപയോഗിച്ച് സാധ്യമാകും. ഉള്ളടക്ക വാക്കുണ്ടായിൽ <BR> ടാഗ് ഉപയോഗിച്ചുകൊണ്ട് അധികവരി ചേർക്കാം. വലുതോ ചെറുതോ ആയ സ്ക്രീനുകളിലും, വലുപ്പം വ്യത്യാസപ്പെടുത്തിയ വിൻഡോകളിലും ലഭിക്കുന്ന വെബ് പേജുകൾ, ചിത്രം 4.21-ൽ തന്നിട്ടുള്ളതിന് സമാനമായിരിക്കുകയില്ല എന്നത് ശ്രദ്ധിക്കുക.

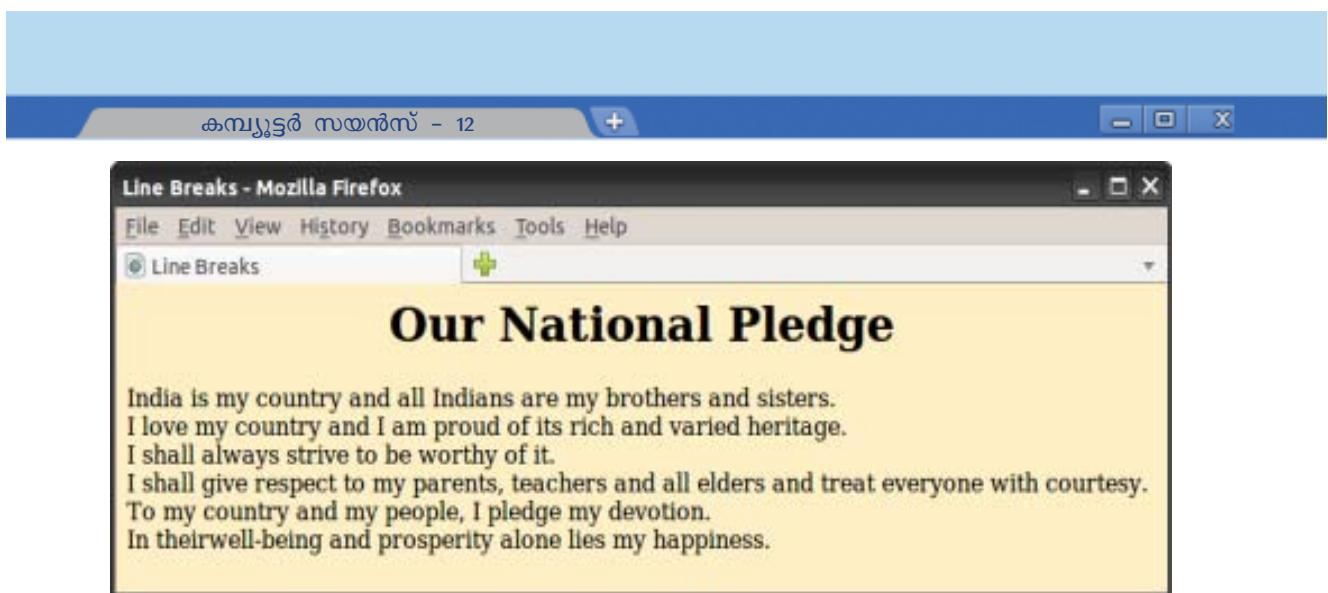
### 4.10.3 <BR> ടാഗ് - വരി വിജയിപ്പിക്കുന്നു

ഒരു വെബ് പേജിലെ വരി മാറ്റിയെഴുതുക എന്നതാണ് <BR> ടാഗിൽ ഉദ്ദേശം. നിലവിലുള്ള വരിയിലെ വാക്യം മുറിച്ച് അടുത്ത വരിയുടെ തുടക്കത്തിൽ നിന്നാരംഭിക്കാനാണ് <BR> ടാഗുപയോഗിക്കുന്നത്. ഈതൊരു എംറി ടാഗാണ്.

ഉദാഹരണം 4.6-ൽ തന്നിട്ടുള്ള HTML കോഡ് നമ്മുടെ രാജ്യത്തിന്റെ പ്രതിജ്ഞ പ്രദർശിപ്പിക്കുന്നു. ചിത്രം 4.22-ൽ കാണിച്ചിട്ടുള്ള പരിശീലന ഫലമായി ലഭിക്കുന്ന പേജിൽ <BR> ടാഗിൽ ഉപയോഗവും <BR>, <P> എന്നീ ടാഗുകൾ തമ്മിലുള്ള വ്യത്യാസവും നമുക്ക് കാണാം.

**ഉദാഹരണം 4.6:** ദേശീയ പ്രതിജ്ഞയിൽ വരികൾ മുറിക്കുന്നത് (Line break) കാണിക്കുവാൻ വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
    <TITLE> Line Breaks </TITLE>
</HEAD >
<BODY Bgcolor = "#FFEFCC">
    <H1 Align = "center"> Our National Pledge </H1>
    <P>India is my country and all Indians
        are my brothers and sisters.<BR>
        I love my country and I am proud of its
        rich and varied heritage.<BR>
        I shall always strive to be worthy of it.<BR>
        I shall give respect to my parents, teachers and all
        elders and treat everyone with courtesy.<BR>
        To my country and my people, I pledge my
        devotion. <BR>In their well-being and prosperity alone
        lies my happiness.
    </P>
</BODY>
</HTML>
```



ചിത്രം 4.22 : <BR> ടാഗിന്റെ ഉപയോഗം കാണിക്കുന്ന ഒരു പേര്



ഉദാഹരണം 4.6, ചിത്രം 4.24 എന്നിവ നൽകുന്ന സൂചനയ്ക്ക് അനുസ്യത മായി, <P> ടാഗും <BR> ടാഗും തമിൽ വേർത്തിരിക്കുന്ന ആശയ ഓർക്കേജാംഗ് പട്ടിക 4.5 പുറിപ്പിക്കുക.

നമ്മൾക്ക് ചെയ്യാം

<P> tag	  tag
	Breaks the current line and continues to the next line.
Container tag	

ചിത്രം 4.3 : <P> ടാഗും <BR> ടാഗും

#### 4.10.4 <HR> ടാഗ് - തിരഞ്ഞീനമായ വരി സൂച്ചിക്കുന്നു

ബൈൽസർ വിൻധോയുടെ വീതിക്ക് കുറുകേ തിരഞ്ഞീനമായ രൂവര നിർമ്മിക്കുന്നത് <HR> ടാഗാണ്. അതിന്റെ ആട്ടിബ്യൂട്ടുകളായ **Size**, **Width** എന്നിവ ഉപയോഗിച്ച് നമുക്ക് വരയുടെ വീതിയും കമ്പാറ്റം വ്യത്യാസപ്പെടുത്താം. കനത്തിന് പിക്സലുകളായും വീതിക്ക് പിക്സലുകളോ മൊത്തം വീതിയുടെ ശതമാനമായോ ആയും വില നൽകുന്നു. <HR> ടാഗിന്റെ മറ്റ് രണ്ട് ആട്ടിബ്യൂട്ടുകളാണ് Noshade, Color എന്നിവ. Noshade ആട്ടിബ്യൂട്ടിന് വിലകാണുകയാണ്. വരയ്ക്ക് നിറം കൊടുക്കുന്നത് Color ആട്ടിബ്യൂട്ടാണ്. left, center, right എന്നിങ്ങനെ വരയുടെ പൊരുത്തപ്പെടലിന് ഉപയോഗിക്കുന്ന Align ആട്ടിബ്യൂട്ടാണ് മറ്റൊന്ന്. <HR> രൂ എന്റെ ടാഗാണ് എന്ന് ഓർക്കുക.

<HR> ടാഗിന്റെയും അതിന്റെ ആട്ടിബ്യൂട്ടുകളുടെയും പ്രവർത്തനം വ്യക്തമാക്കുന്ന കോഡ് ഉദാഹരണം 4.7-ൽ തന്നിട്ടുണ്ട്. അതിന്റെ ഒരു പൂട്ട് ചിത്രം 4.23-ൽ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.

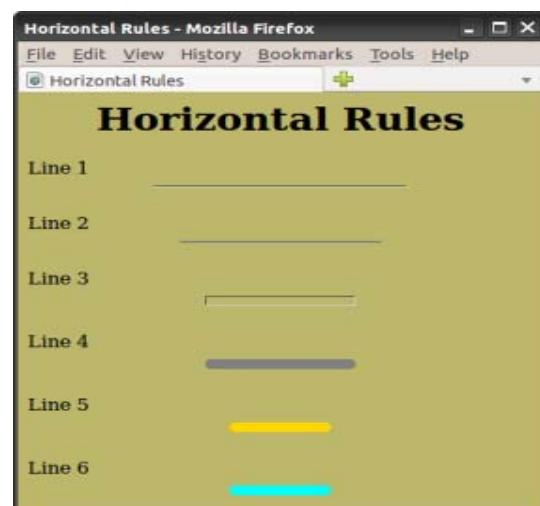
**ഉദാഹരണം 4.7:** വിവിധ തരത്തിലുള്ള രേഖകൾ വരയ്ക്കുന്നതിന്

```

<HTML>
<HEAD>
    <TITLE> Horizontal Rules </TITLE>
</HEAD >
<BODY Bgcolor= "#BDB76B">
<H1 Align= "center"> Horizontal Rules </H1>
Line 1<HR Width= "50%" Align= "center"> <BR>
Line 2<HR Width= "40%" Align= "center" Noshade> <BR>
Line 3<HR Size= "10" Width= "30%" Align= "center"> <BR>
Line 4<HR Size= "10" Width= "30%" Align= "center" Noshade><BR>
Line 5<HR Size= "10" Width= "20%" Align= "center" Noshade
        Color= "gold"> <BR>
Line 6<HR Size="10" Width="20%" Align="center" Color="Aqua">
</BODY>
</HTML>

```

ഈ HTML ലിവിൽവും അതിന്റെ ഒരു പുട്ടും പരിശോധിച്ചശേഷം, വരകളുടെ ആകൃതികളിലൂളിൽ സമാനതകളും വ്യത്യാസങ്ങളും നിങ്ങൾക്ക് തിരിച്ചറിയാനാകുമോ? ഈ ടെ, ആദ്യത്തെ രേഖ സൈസ്, Noshade എന്നീ ആട്ടിബ്യൂട്ടുകളില്ലാതെയും, രണ്ടു മത്തെ രേഖ സൈസ് ആട്ടിബ്യൂട്ടി ല്ലാതെയുമാണ് വരച്ചിരിക്കുന്നത്. രേഖ മുന്നിനും, നാലിനും വലുപ്പം 10 ആണ്. എന്നാൽ മുന്നാമത്തെത്തിന് Noshade ആട്ടിബ്യൂട്ട് ഇല്ല. അവസാനത്തെ ഒരു രേഖകൾക്ക് Color ആട്ടിബ്യൂട്ട് അധിക മായുണ്ട്. ഓരോ അവസ്ഥയിലുമുള്ള ഈ രേഖകളുടെ ആകൃതികൾ ശ്രദ്ധിക്കുക. നിങ്ങളുടെ പരീക്ഷണ ശാലയിലെ പ്രവർത്തനത്തിൽ സാധ്യമായ എല്ലാ വിലകളും വ്യത്യസ്തങ്ങളായ ആട്ടിബ്യൂട്ടുകൾക്ക് നൽകിക്കൊണ്ട് നിങ്ങൾക്ക് കോഡ് മാറ്റാവുന്നതും വെബ് പേജിൽ അപ്പോഴുണ്ടാകുന്ന മാറ്റം ദർശിക്കാവുന്നതുമാണ്.



ചിത്രം 4.23 : പലതരത്തിലുള്ള തിരഞ്ഞീറമായ രേഖകൾ

#### 4.10.5 <CENTER> ടാഗ് - ഉള്ളടക്കം മധ്യഭാഗത്താക്കുന്നു

വെബ് പേജിലെ ഉള്ളടക്കം തിരഞ്ഞീറമായി മധ്യഭാഗത്താക്ക് കൊണ്ട് വരുന്നത് <CENTER> ടാഗാണ്. ഉള്ളടക്കം സാധാരണയായി വാക്കുങ്ങൾ, ചിത്രങ്ങൾ, പട്ടികകൾ തുടങ്ങിയവയായിരിക്കും. ഈതോറു കണ്ണെയ്ക്കൻ ടാഗ് ആയതുകൊണ്ട് <CENTER>, </CENTER> എന്നി ടാഗ് ജോധികൾക്ക് ഇടയ്ക്ക് നൽകുന്ന ഉള്ളടക്കം ബൗസർ വിൻഡോയുടെ മധ്യഭാഗത്ത് നിർത്തുന്നു. ഈ ടാഗിന് ആട്ടിബ്യൂട്ടുകൾ ഇല്ല.

#### 4.10.6 വാക്യങ്ങൾക്ക് രൂപരൂപന നൽകുന്ന ടാഗ്കൾ

വേഡ് പ്രോസസിംഗ് പ്രോഗ്രാമുകൾ ചെയ്യുന്നത് പോലെ, നമുക്ക് വെബ് പ്രോജെക്റ്റിലെ വാക്യങ്ങൾക്ക് രൂപരൂപന നൽകാനാകും. വാക്യങ്ങളുടെ പ്രാധാന്യത്തെ സൂചിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്ന സവിശേഷതകളാണ് ബോൾഡ്, ഇറ്റാലിക്, അംഗൾ ലൈൻ തുടങ്ങിയ വ. ഇതിനായുള്ള HTML ടാഗ്കൾക്കുണ്ട് നമുക്ക് പരിച്ച ചെയ്യാം.

##### **<B>** - ടെക്സ്റ്റ് ബോൾഡ് ആക്കുന്നു

ഈ ടാഗ് ടെക്സ്റ്റുണ്ടിൽ ശ്രദ്ധി ബോൾഡ് ആക്കുന്നു. ഉള്ളടക്കത്തിലെ വാക്യങ്ങളുടെ ബാഹ്യരൂപം കൂടുതൽ ബോൾഡ് ആയി പ്രദർശിപ്പിക്കുന്നത് **<B>** ടാഗാണ്.

##### **<I>** - ടെക്സ്റ്റ് ഇറ്റാലിക്‌സിലാക്കുന്നു

ഈ വാക്യങ്ങളുടെ ശ്രദ്ധി ഇറ്റാലിക്‌സിലാക്കി രൂപപ്പെടുത്തുന്നു. **<I>** ടാഗിനും **</I>** ടാഗിനുമിടയ്ക്കുള്ള ഉള്ളടക്കം ഇറ്റാലിക്‌സിലാക്കുന്നു.

##### **<U>** - ടെക്സ്റ്റുണ്ടിന് അടിവര ഇടുന്നു

വെബ് പ്രോജെക്റ്റിലെ വാക്യത്തിന് അടിവര നൽകാനാണ് **<U>** ടാഗുപയോഗിക്കുന്നത്. **<U>** ടാഗിനും **</U>** ടാഗിനുമിടയ്ക്ക് ഉൾക്കൊള്ളിച്ചിട്ടുള്ള ഉള്ളടക്കം അടിവര ഉള്ളതായി മാറുന്നു. രൂപരൂപന നൽകുന്ന ടാഗുകളായ **<U>**, **<B>**, **<I>** എന്നിവ ഒന്നിച്ച് ഉപയോഗിക്കാനാകും എന്നതിനാൽ, ഉള്ളടക്കം ബോൾഡും, ചരിഞ്ഞതും, അടിവരയുള്ളതുമാകും.

##### **<S>, <STRIKE>** - ടെക്സ്റ്റുണ്ടിന് കുറുക്കുന്നു

ങ്ങെ പ്രക്രിയയ്ക്കാണ് **<S>**, **<STRIKE>** എന്നീ ടാഗുകൾ ഉപയോഗിക്കുന്നത്. വാക്യങ്ങൾ കുറുക്കുക വെച്ചിയ രൂപത്തിൽ പ്രദർശിപ്പിക്കുന്നു. ഉദാഹരണമായി, കുറുക്കുക വെച്ചിയ വാക്യമാണ് *Thank you all*.

##### **<BIG>** - ടെക്സ്റ്റ് വലുപ്പം ഉള്ളതാക്കുന്നു

സാധാരണ വലുപ്പത്തിലുള്ള ടെക്സ്റ്റുനേക്കാൾ വലുപ്പം ഉള്ളടക്കത്തിന് നൽകാനാണ് **<BIG>** ടാഗ് ഉപയോഗിക്കുന്നത്. ദിവിത്തതിലെ വാക്കുകൾക്ക് അല്ലെങ്കിൽ വർക്കൾക്ക് പ്രത്യേക പ്രാധാന്യം നൽകാനാണ് ഈ ഉപയോഗിക്കുന്നത്. സാധാരണ ഗതിയിൽ **<BIG>** ടാഗിലെ അക്ഷരങ്ങളുടെ വലുപ്പം നിലവിലുള്ള വലുപ്പത്തെക്കാൾ ഒരുപടി കൂടുതലായിരിക്കും.

##### **<SMALL>** - ടെക്സ്റ്റ് ചെറുതാക്കുന്നു

ടെക്സ്റ്റുകളുടെ നിലവിലുള്ള വലുപ്പത്തെക്കാൾ ചെറുതാക്കാനാണ് **<SMALL>** ടാഗ് ഉപയോഗിക്കുന്നത്. സാധാരണയായി, **<SMALL>** ടാഗിലെ അക്ഷരങ്ങളുടെ വലുപ്പം, നിലവിലുള്ള അക്ഷരങ്ങളുടെ വലുപ്പത്തെക്കാൾ ഒന്ന് കുറവായിരിക്കും.

##### **<STRONG>** - ടെക്സ്റ്റ് കൂടുതൽ ബോൾഡുമാക്കുന്നു

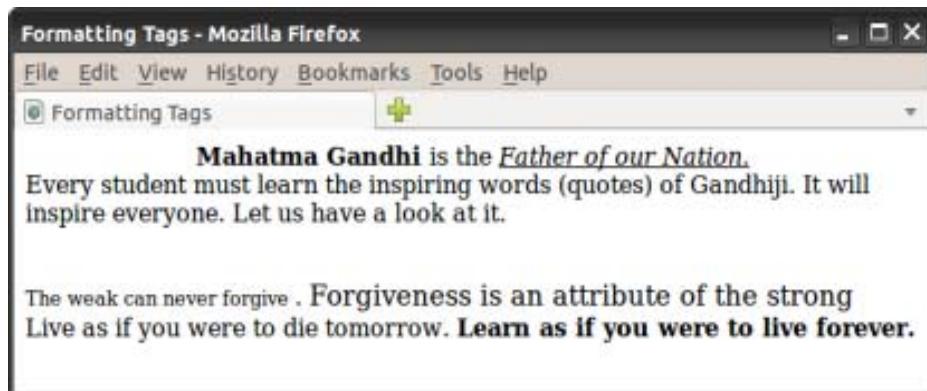
ഒരു ഉപവാക്യ ടാഗാണ് **<STRONG>** ഈ പ്രാധാന്യമുള്ള വാക്യത്തെ നിർവ്വചിക്കുന്നു. **<STRONG>** വാക്യം സാധാരണയായി കൂടുതൽ ബോൾഡുമാക്കിയ ബാഹ്യാകൃതിയോട് കൂടിയാണ് ആവിഷ്കരിക്കുന്നത്. **<B>** ടാഗിനെപ്പോലെയാണിത്. വാക്യരൂപത്തിലുള്ള ഉള്ളടക്കത്തിലെ ഒരു ഉപവാക്യത്തിന് കൂടുതൽ പ്രാധാന്യം നൽകാനാണ് **<STRONG>** ടാഗ് ഉപയോഗിക്കുന്നത്.

#### 4. വെബ് സാങ്കേതിക വിദ്യ

ഉദാഹരണം 4.8-ൽ തന്നിട്ടുള്ള കോഡ് ഈ ടാഗുകളുടെ പ്രായോഗികത വിവരിക്കുന്നു. സന്ദർഭോച്ചിതമായി, നമുക്ക് മഹാത്മാഗാന്ധിയുടെ വാക്കുകൾ ഉല്ലരിക്കാം. ഈ കോഡ് ആവിഷ്കരിക്കുന്നതിന്റെ ഫലമായുണ്ടാകുന്ന വെബ് പേജ് ചിത്രം 4.24-ൽ കാണിച്ചിരിക്കുന്നു.

**ഉദാഹരണം 4.8:** വാക്യങ്ങൾ രൂപപ്പെടുത്തുന്ന ടാഗുകളെക്കുറിച്ച് വിവരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
    <TITLE> Formatting Tags </TITLE>
</HEAD>
<BODY>
    <P>
        <CENTER><B>Mahatma Gandhi </B>is the <I> <U> Father of
        our Nation.</U> </I> </CENTER>
    Every student must learn the inspiring words (quotes)
    of Gandhiji. It will inspire everyone. Let us have a
    look at it.
    </P><BR>
    <SMALL> The weak can never forgive </SMALL>.
    <BIG> Forgiveness is an attribute of the strong</BIG>
    <BR> Live as if you were to die tomorrow.
    <STRONG> Learn as if you were to live forever.</STRONG>
</BODY>
</HTML>
```



ചിത്രം 4.24 : വാക്യാഖരന രൂപപ്പെടുത്തുന്ന ടാഗുകളെക്കുറിച്ചുള്ള വിവരണം

വാക്യങ്ങൾക്ക് ചില പ്രത്യേക രൂപം നൽകുന്ന ടാഗുകൾ നമുക്ക് പരിചയപ്പെടാം.

**<EM> - വാക്യങ്ങൾക്ക് പ്രത്യേക പ്രാധാന്യം നൽകുന്നു**

വാക്യങ്ങൾക്ക് പ്രത്യേക പ്രാധാന്യം നൽകാൻ ഉപയോഗിക്കുന്ന ടാഗാണ് <EM>. പ്രയോഗത്തിൽവരുന്നോൾ, ആ ഭാഗം സാധാരണ മറ്റാലിക്സിലൊയിരിക്കും ആവിഷ്കരിക്കുന്നത്. <I> ടാഗുപയോഗിക്കുന്നോഴ്രുള്ള ഗുണം തന്നെയാകും <EM> ടാഗും തരുന്നത്.

## <SUB>, <SUP> എന്നീ ടാഗുകൾ - സബ് സ്കീപ്പറ്റും സൂപ്പർ സ്കീപ്പറ്റും നിർക്കി കുന്നു

ജലം, സൽഫൈറിക് അല്ലെങ്കിൽ തുടങ്ങിയവയുടെ തമാത്രാ വാക്യം ഫോൺകുർ കൂല്ലുകളിൽ നാം പറിച്ചിട്ടുള്ളതാണ്. അവ ധമാക്രമം  $H_2O$ ,  $H_2SO_4$  എന്നിങ്ങനെ ആയിരിക്കും. HTML-ൽ അത്തരം പ്രതീകങ്ങൾ നമുക്ക് എങ്ങനെ ചിത്രീകരിക്കാനാകും? അകങ്ങലശ്രൂതി സബ്-സ്കീപ്പറ്റ് രൂപത്തിലാണ് എഴുതിയിട്ടുള്ളതെന്ന് നമുക്ക് കാണാം. ഒരു വെബ് പേജിൽ സബ്-സ്കീപ്പറ്റ് നിർമ്മിക്കാനാണ് <SUB> ടാഗുപയോഗിക്കുന്നത്.  $H<SUB>2</SUB>$ O എന്ന കോഡുപയോഗിച്ച് കൊണ്ട്  $H_2O$  എന്ന വാക്യം നമുക്ക് പ്രദർശിപ്പിക്കാനാകും.

അതുപോലെ,  $(a+b)^2 = a^2 + 2ab + b^2$  പോലുള്ള ബീജഗണിത ആവിഷ്കാര രീതിയിലെ സൂപ്പർ സ്കീപ്പറ്റുകൾ <SUP> ടാഗ് ഉപയോഗിച്ച് നമുക്ക് സൂചിപ്പിക്കാനാകും. ഈ സൂത്രവാക്യം നമുക്ക് ഇങ്ങനെ എഴുതാം.

$$(a+b)^2</SUP> = a<SUP>2</SUP> + 2ab + b<SUP>2</SUP>$$

## <BLOCKQUOTE>, <Q> എന്നീ ടാഗുകൾ - ഒരു ഉദ്ദരണി പ്രദർശിപ്പിക്കുന്നു

<BLOCKQUOTE> ടാഗ് അവയ്ക്കിടയിൽ ഉൾക്കൊള്ളിച്ചിട്ടുള്ള ഉള്ളടക്കം മാറ്റി പ്രദർശിപ്പിക്കാൻ ഉപയോഗിക്കുന്നു. വാക്യങ്ങളെ ഇരട്ട ഉദ്ദരണികൾക്കുള്ളിൽ മാറ്റി പ്രദർശിപ്പിക്കുന്നത് സൂചിപ്പിക്കാനാണ് <Q> ടാഗ് ഉപയോഗിക്കുന്നത്. വണ്ണിക മുറിക്കാതെയുള്ള ചുരുങ്ഗിയ ഉദ്ദരണികൾക്ക് വേണ്ടിയുള്ളതാണ് ഈ ടാഗ്. എന്നാൽ <BLOCKQUOTE> വലിയ ഉദ്ദരണികൾക്ക് ഉപയോഗിക്കാൻ വേണ്ടിയുള്ളതാണ്.

ഉദാഹരണം 4.9-ൽ മുകളിൽ സൂചിപ്പിച്ചിട്ടുള്ള ടാഗുകളുടെ വിശദീകരണമുണ്ട്. ലോക പരിസ്ഥിതി ദിനം ആദ്ദേഹിക്കുന്നത് ജൂൺ 5-ന് ആണ് എന്നത് നമുക്ക് അനുസ്മരിക്കാം. ഈ ദിവസതേക്കുള്ള ചിത്രകൾ നാം സൂഷ്ടിക്കുകയും ഇത്തരം ടാഗുകളുപയോഗിച്ച് ഒരു വെബ് പേജ് നിർമ്മിക്കുകയും ചെയ്യാം. ചിത്രം 4.25-ൽ അനുബന്ധമായ വെബ് പേജ് കാണിച്ചിട്ടുണ്ട്.

**ഉദാഹരണം 4.9: <SUP>, <BLOCKQUOTE>, <Q> എന്നീ ടാഗുകൾ വിശദീകരിക്കുന്നതിന്**

```

<HTML>
<HEAD>
    <TITLE> BlockQuote and Q tags </TITLE>
</HEAD>
<BODY Bgcolor= "#98FB98" Text= "#008000">
    Every year we celebrate World Environment Day on 5<SUP>
    th</SUP> June. Let us have a message to all on this
    occassion.
    <BLOCKQUOTE> <B>June 5<SUP>th</SUP> is World Environment
    Day. </B>

```

#### 4. വെബ് സാങ്കേതിക വിദ്യ

```
Mother nature too needs care and protection. Show her  
your care by caring for her trees. Love trees and love  
nature. And work for a greener environment because  
generations have to come... The future depends on us...  
</BLOCKQUOTE>  
<Q>Keep your world clean and green. Save trees, Save the  
environment!!  
</Q>  
</BODY>  
</HTML>
```

ആദ്യ വണ്ണിക ആരംഭിക്കുന്നത് ബോഗസർ വിസ്വേഡായുടെ ഇടത് മുലയിൽ നിന്നാണെന്ന് ചിത്രം 4.25-ൽ നിന്നും നമുക്ക് മനസിലാക്കാം. അതിൽ സുപ്പർ സ്കീപ്പറ്റ് പ്രയോഗിച്ചിട്ടുണ്ട്. <BLOCKQUOTE> ടാഗ് ഉപയോഗിച്ചതിനാൽ രണ്ടാമതെത്ത് വണ്ണിക മാറ്റി പ്രദർശിപ്പിച്ച് കൊണ്ടാണ് തുടങ്ങിയിരിക്കുന്നത്. അവസാനമായി, മുന്നാമതെത്ത് വണ്ണിക ഇടു ഉദ്ദരണികൾക്ക് ഉള്ളിലാണ്. അതിന്റെ ഉള്ളടക്കം <Q> ടാഗ് ജോധികൾക്ക് ഉള്ളിൽ ഉൾപ്പെടുത്തിയിരിക്കുന്നു. ഈ വണ്ണികകളുടെ പൊരുത്ത പ്ലേറ്റ് കൂടി നിരീക്ഷിക്കുക. <BLOCKQUOTE> ടാഗിൽ ഒന്നിലധികം വണ്ണികകൾ ഉൾപ്പെടുത്തിയിരിക്കുന്നത് ശ്രദ്ധിക്കുക.



ചിത്രം 4.25 : ഉദ്ഘാടനം 4.9 ലെ  
കോധിന് അനുബന്ധമായ വെബ് പേജ്



## നിങ്ങളുടെ പുരോഗതി അറിയുക

1. വാക്യങ്ങൾക്ക് ഘടന നൽകുന്ന ചില ടാഗുകളുടെ പേരെഴുതുക.
2. <HR> ടാഗിൽ വ്യത്യസ്ത ആട്ടിബ്യൂട്ടുകളുടെ നാമാവലിയുണ്ടാകുക.
3. എത്ര തലങ്ങളിലുള്ള ശീർഷക ടാഗുകൾ HTML ലെ ലഭ്യമാണ്?
4.  $x^3 + y^3$  എന്ന് പ്രദർശിപ്പിക്കുന്നതിനുള്ള HTML കോഡ് ശകലം എഴുതുക.
5. ശരിയോ തെറ്റോ എന്ന് പ്രസ്താവിക്കുക.
  - a. <BR> ടാഗ് ഒരു എംബി ടാഗാണ്.
  - b. HTML ലിഖിതത്തിൽ <EM> ടാഗിനും <I> ടാഗിനും ഒരേ ഉപയോഗമാണുള്ളത്.
  - c. <U> ടാഗിൽയും <I> ടാഗിൽയും ഒരുമിച്ചുള്ള ഉപയോഗം അനുവദനീയമല്ല.
6. <STRONG> ടാഗിൽ ഉപയോഗമെന്താണ്?
7. <STRONG> ടാഗില്ലേതിന് സമാനമായ കൃത്യം നിർവ്വഹിക്കുന്ന ടാഗ് എത്രാണ്?
8. താഴെന്നിട്ടുള്ളവയിൽ നിന്നും ഒറ്റപ്പെട്ടത് പുറത്തെടുക്കുക.
  - a. HTML
  - b. ALIGN
  - c. HEAD
  - d. CENTER

### 4.10.7 <PRE> - ഘടന രൂപപ്പെടുത്തിയ ടെക്നിക്കുൾ പ്രദർശിപ്പിക്കുന്നത്

ടെക്നിക്ക് എധിറ്ററിൽ നൽകിയ ഉള്ളടക്കം നമുക്ക് അതുപോലെ പ്രദർശിപ്പിക്കണം എന്നിരിക്കുന്നത്. ഈ ആവശ്യം <PRE> ടാഗിന് നിർവ്വഹിക്കാനാകും. സാധാരണ ഗതിയിൽ വെറ്റ് സ്പേസുകൾ, ന്യൂ ലൈൻ ക്യാർക്കറ്റുകൾ, ടാബ് സ്പേച്ചസുകൾ തുടങ്ങിയ വയ്ക്ക് ബേഖസർ പരിധി നിശ്ചയിച്ചിട്ടുണ്ട്. അതുകൊണ്ട്, ബേഖസറിൽ സ്വയം ഘടന നിർവ്വചിക്കുന്ന സ്വഭാവം <PRE> ടാഗുപയോഗിച്ച് നമുക്ക് പ്രവർത്തന രഹിതമാക്കാനാകും. ഈ ടാഗ്, അതിലുശ്രദ്ധപ്പെടുത്തിയിട്ടുള്ള വാക്യത്തിൽ ഘടന രൂപപ്പെടുത്തിക്കഴി ഞ്ഞതാണ് എന്ന് ബേഖസറിനോട് പറയുകയും, ആയതിനാൽ വീണ്ടും ഘടന പുനർന്നിർണ്ണയിക്കേണ്ടതില്ല എന്ന് ഓർമ്മിപ്പിക്കുകയും ചെയ്യുന്നു. അതായത്, വാക്യം അതേപടി തന്നെ പ്രദർശിപ്പിക്കുവാൻ ബേഖസറിനോട് പറയുന്നു.

ഉദാഹരണം 4.10, ചിത്രം 4.26 എന്നിവ ഈ ടാഗിനെക്കുറിച്ച് ഒരു ധാരണ നമുക്ക് നൽകുന്നു. ലോക പരിസ്ഥിതിഭിന്നത്തോടനുബന്ധിച്ച് കുറച്ച് വാക്യങ്ങളുള്ള ഒരു വെബ് പേജ് നമുക്ക് നിർമ്മിക്കാം.

#### ഉദാഹരണം 4.10: <PRE> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```

<HTML>
<HEAD>
    <TITLE> Pre Formatting tags </TITLE>
</HEAD>
<BODY Bgcolor = "#eee8aa" Text = "#b22222">
    <PRE>
        Don't Pollute Water,
        Don't Pollute Air,
        Don't Pollute Environment,
        And Don't Pollute Yourself,
        Celebrate World Environment Day ...
    </PRE>
</BODY>
</HTML>

```

<PRE>, </PRE> എന്നീ ടാഗുകൾക്ക് ഇടയിലുള്ള തെരുവും HTML ലിഖിത ത്തിലുള്ളത് പോലെ തന്നെ പ്രദർശിപ്പിക്കുന്ന എന്ന് ചിത്രം 4.26-ൽ കാണിച്ചിട്ടുള്ള വെബ് പേജിൽ നിന്നും മനസിലാക്കാം.



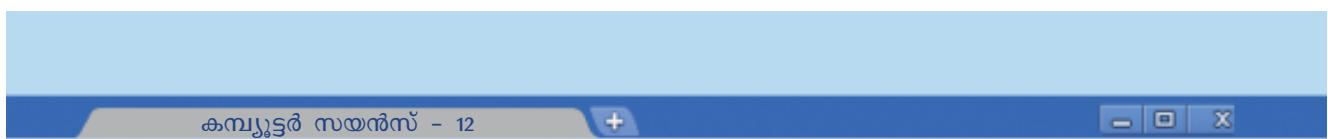
ചിത്രം 4.26 : <PRE> ടാഗിനെക്കുറിച്ചുള്ള വിശദീകരണം

#### 4.10.8 <ADDRESS>

##### - മേൽവിലാസം പ്രദർശിപ്പിക്കുന്നു

ഒരു ലിഖിതത്തിന്റെ അല്ലെങ്കിൽ പ്രബന്ധത്തിന്റെ രചയിതാവിനെ/ഉടമസ്ഥനെ ബന്ധപ്പെടാനുള്ള വിവരം പോലുള്ളവ നിർവ്വചിക്കുവാനുള്ള ടാഗാണ് <ADDRESS> ടാഗ്. പേര്, ദേശപോണ്ട് നമ്പർ, പിൻ നമ്പർ, e-മെയിൽ മേൽവിലാസങ്ങൾ തുടങ്ങിയവയാണ് ഈ ടാഗിന്റെ ഉള്ളടക്കമായി ഉൾക്കൊള്ളിക്കുന്നത്. വാക്കുങ്ങളെ ഇറ്റാലിക്സ് രീതിയിലാണ് മിക്കവാറും ഭേദസറൂകൾ പ്രദർശിപ്പിക്കുന്നത്.

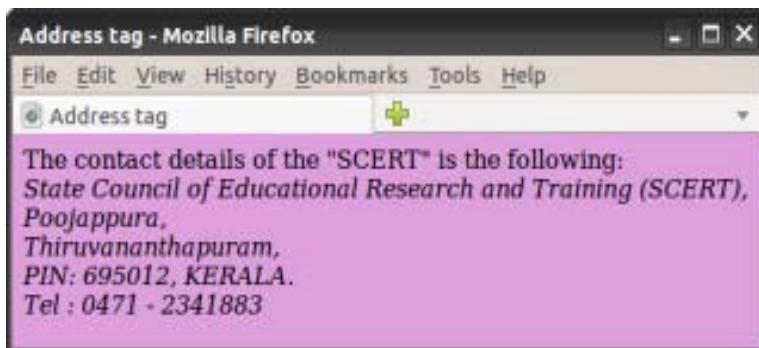
ഉദാഹരണം 4.11-ൽ തന്നിട്ടുള്ള കോഡ് <ADDRESS> ടാഗിനെക്കുറിച്ച് വിശദീകരിക്കുന്നു. പേജിന്റെ രൂപം ചിത്രം 4.27-ൽ കാണിച്ചിരിക്കുന്നു.



#### ഉദാഹരണം 4.11: <ADDRESS> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
    <TITLE> Address tag </TITLE>
</HEAD>
<BODY Bgcolor= "#DDA0DD">
    The contact details of the "SCERT" is the following:
<ADDRESS>
    State Council of Educational Research and Training
    (SCERT) ,<BR>
    Poojappura,<BR>
    Thiruvananthapuram,<BR>
    PIN: 695012, KERALA.<BR>
    Tel : 0471 - 2341883
</ADDRESS>
</BODY>
</HTML>
```

ബന്ധപ്പെടാനുള്ള വിവരങ്ങൾ ഒരു ഭാഗമായി വരുമ്പോൾ, <ADDRESS> ടാഗ് സാധാരണ ഗതിയിൽ തപാൽ മേൽവിലാസം വിവരിക്കാനാണ്.



ഈ ഉപയോഗിക്കുന്നത്. ചിത്രം 4.27 : അധിസ് ടാഗ് ഉൾപ്പെടുത്തിയിട്ടുള്ള രീതി വെം്പ് പേജ് എന്നിരുന്നാലും, തന്ത്രം ശൈലിയിൽ <EM>, <I> എന്നിവ പ്രദർശിപ്പിക്കുന്ന ശൈലിക്ക് സമാനമായാണ് <ADDRESS> ടാഗ്യും വാക്യം പ്രദർശിപ്പിക്കുന്നത്. ആയതിനാൽ, ബന്ധപ്പെടാനുള്ള വിവരം കൈകാര്യം ചെയ്യാൻ അനുയോജ്യമായ രീതിയിൽ അത് ഉപയോഗിക്കാവുന്നതാണ്. എഴുത്തുകാരൻ്റെ പേര്, പകർപ്പുവകാശം തുടങ്ങിയ വിവരങ്ങൾ ഉൾക്കൊള്ളുന്ന അടിക്കുറിപ്പിനുള്ളിലാണ് <ADDRESS> ടാഗ് എലുമെന്ത് മാതൃകാപരമായി നൽകാൻ മുൻ്നെടുത്തത്.

#### 4.10.9 <MARQUEE> - മാർക്കു വാക്യം പ്രദർശിപ്പിക്കുന്നു

നാം ഇതുവരെ ചർച്ച ചെയ്ത പ്രവർത്തനം ടാഗ്യും ഉള്ളടക്കങ്ങൾ ബേഖാസിരിൽ പ്രദർശിപ്പിക്കുക മാത്രമാണ് ചെയ്യുന്നത്. എന്നാൽ രീതി വാക്യശകലം അല്ലെങ്കിൽ ചിത്രം വെം്പ് പേജിൽ തിരഞ്ഞെടുത്താണ് ലാംബമായോ മാർക്കു ആയി കാണിക്കുന്നതിന്, <MARQUEE> ടാഗ് ഉപയോഗിക്കുന്നു.

<MARQUEE> ടാഗിനൊപ്പം ഉപയോഗിക്കുന്ന പ്രധാനപ്പെട്ട ആട്ടിബ്യൂട്ടുകളുടെ നാമാവലി താഴെ തന്മുഖം കാണാം.

- **Height:** പിക്സലായോ ബ്രൗസർ വിൻഡോയുടെ ഉയരത്തിന്റെ ശതമാനമായോ മാർക്കുവിന്റെ ഉയരം നിശ്ചയിക്കുന്നു.
  - **Width:** പിക്സലായോ, ബ്രൗസർ വിൻഡോയുടെ വീതിയുടെ ശതമാനമായോ വീതി സൂചിപ്പിക്കുന്നു.
  - **Direction:** ഏത് ദിശയിലാണ് ടെക്സ്റ്റ് ചലിക്കേണ്ടതെന്ന് സൂചിപ്പിക്കുന്നു. up, down, left, right എന്നി വിലകളാണ് ഇതിനുള്ളത്.
  - **Behaviour:** മാർക്കുവിന്റെ സ്ക്രോളിംഗ് ഏത് വിധത്തിലാകണമെന്ന് സൂചിപ്പിക്കുന്നു. scroll, slide, alternate എന്നി വിലകളാണ് ഇതിനുള്ളത്.
  - **Scrolldelay:** ഓരോ സ്ക്രോളിനും ഇടയ്ക്കുള്ള സമയത്തെറിയും സൂചിപ്പിക്കുന്നു. 10, 15 തുടങ്ങിയവ പോലെ സെക്കന്റിലുള്ള വിലകളാണ് ഇതിനുള്ളത്.
  - **Scrollamount:** മാർക്കു ടെക്സ്റ്റിന്റെ വേഗത സൂചിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്നു.
  - **Loop:** മാർക്കു ടെക്സ്റ്റ് എത്രപ്രാവശ്യം സ്ക്രീനിൽ സ്ക്രോൾ ചെയ്യണമെന്ന് സൂചിപ്പിക്കുന്നു. ഇതിന്റെ തന്ത്ര വില Infinite ആയിരിക്കും, അങ്ങനെ ആയാൽ മാർക്കു അവസാനമില്ലാതെ സ്ക്രോൾ ആകും.
  - **Bgcolor:** നിറമോ, നിറത്തെ സൂചിപ്പിക്കുന്ന ഫോറോന്റുമൊരുള്ള സംവ്യൂഹം ഉപയോഗിച്ച് പശ്വാത്തലവനിനും സൂചിപ്പിക്കുന്നു.
  - **Hspace:** ഇത് മാർക്കുവിന് ചുറ്റുമുള്ള തിരഞ്ഞീനമായ സ്ഥലം സൂചിപ്പിക്കുന്നു. ഇതിന്റെ വില പിക്സൽ ആയോ ശതമാനവിലെ ആയോ നൽകുന്നു.
  - **Vspace:** മാർക്കുവിനു ചുറ്റുമുള്ള ലംബമായ ശുന്തസ്ഥലത്തെ സൂചിപ്പിക്കുന്നു. ഇതിന്റെ വിലയും പിക്സൽ ആയോ ശതമാനവിലെ ആയോ ആണ് നൽകുന്നത്.
- ഉദാഹരണം 4.12: <MARQUEE> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

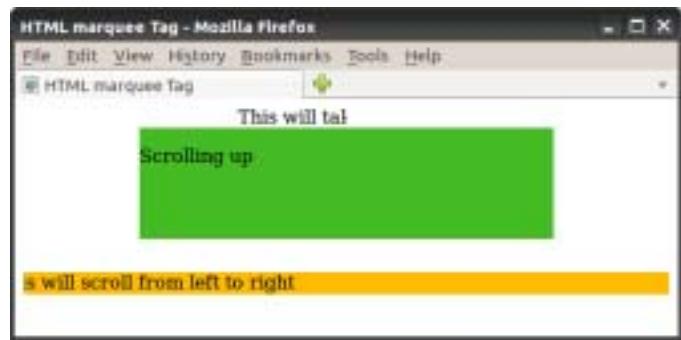
```

<HTML>
<HEAD>
    <TITLE> HTML marquee Tag </TITLE>
</HEAD>
<BODY>
    <MARQUEE Width= "50%"> This will take only 50% width of
        Browser Window</MARQUEE>
    <MARQUEE Height= "100" Hspace= "100" Bgcolor= "#44BB22"
        Direction= "up"> Scrolling up </MARQUEE>
    <MARQUEE Height= "20" Vspace= "30" Bgcolor= "#FFBB00"
        Direction= "right"> This will scroll from left to right
    </MARQUEE>
</BODY>
</HTML>

```

വീതി 50% ആയതിനാൽ ആദ്യത്തെ മാർക്കു മധ്യഭാഗത്ത് നിന്നും തുടങ്ങി വിൻഡോ യുടെ ഇടത് വശത്തെക്ക് സ്ക്രോൾ ചെയ്യുന്നു. രണ്ടാമത്തെ മാർക്കു പച്ചനിറത്തോടുകൂടിയതും 100 പിക്സൽ ഉയരം സ്ക്രോൾ ചെയ്യാൻ ഉപയോഗിച്ചിട്ടുള്ളതുമാണ്.

ഇടത് സീമയിൽ നിന്നും തിരശ്വീനമായി 100 പിക്സൽ അകലെ ആയാണ് സ്ക്രോൾ ഏറിയ സജീകരിച്ചിട്ടുള്ളത്. മുന്നാമത്തെ മാർക്കു “#FFBB00” നിന്നും കൊണ്ട് പശ്ചാത്തലം ഒരു കി യിട്ടുള്ളതും മുന്നിലുള്ള മാർക്കുവിൽ നിന്നും ലംബമായി 30 പിക്സൽ താഴേക്ക് സ്ഥാപിച്ചിട്ടുള്ളതുമാണ്. സ്ക്രോൾ ചെയ്യുന്ന വിൻഡോ 40 പിക്സൽ ഉയരത്തിൽ ലുള്ളതും ഇടത്തെ നിന്ന് വലത്തെക്ക് നീണ്ടുനിന്നും തുമാണ്. വാക്കുകളേം, വാക്കുങ്ങളേം മാത്രമല്ല ചിത്രങ്ങളും മാർക്കു വിന്റെ ഉള്ളടക്കമായി നമുക്ക് ഉപയോഗിക്കാം.



ചിത്രം 4.28 : Marquee കാര്യം അതിന്റെ ആട്ടിബ്യൂട്ടുകളും

#### 4.10.10 <DIV> - വാക്യങ്ങളുടെ ഒരു ഭാഗം ചിട്ടപ്പെടുത്തുന്നു

ലിഖിതത്തിന്റെ ഒരു ഭാഗം അല്ലെങ്കിൽ ഒരുന്നിര ചിട്ടപ്പെടുത്താനാണ് <DIV> ടാഗ് ഉപയോഗിക്കുന്നത്. HTML ലിഖിതത്തിലുള്ള വലിയ ഭാഗങ്ങൾ ഒരുമിച്ച് ഒരു കുടമാക്കി ചിട്ടപ്പെടുത്താനും <DIV> ടാഗ് കൊണ്ട് സാധിക്കുന്നു. വണികകൾ, പട്ടികകൾ തുടങ്ങിയവ ഈ ഭാഗത്തുണ്ടാകും. <DIV> ടാഗിന് മുൻപും പിൻപും മിക്കവാറും ബ്രൗസറുകൾ ഒരു വരി (ശൃംഖല) ചേർത്തിട്ടുണ്ടാകും. <DIV> ടാഗിന്റെ ആട്ടിബ്യൂട്ടുകൾ താഴെ തന്നിട്ടുള്ളവയാണ്.

**Align :** left, right, center, justify എന്നീ വിലകൾ കൊണ്ട് തിരശ്വീനമായ പൊരുത്തപ്പെടൽ നിജപ്പെടുത്തുന്നു.

**Id :** ടാഗിനെ തിരിച്ചറിയാനായി സവിശേഷമായ ഒരു സൂചകം നൽകുന്നു.

**Style :** ഉള്ളടക്കത്തിലെ അക്ഷരം, നിന്നും തുടങ്ങിയവ എങ്ങനെ ആവിഷ്കരിക്കണമെന്ന് നിർദ്ദേശിക്കുന്നു.

<DIV> ടാഗിന്റെ Align, Style എന്നീ ആട്ടിബ്യൂട്ടുകളുടെ ഉപയോഗം ഉദാഹരണം 4.13-ലും, തൽപ്പലമായി ലഭിക്കുന്ന വെബ്പോജ് ചിത്രം 4.29-ലും കാണാവുന്നതാണ്.

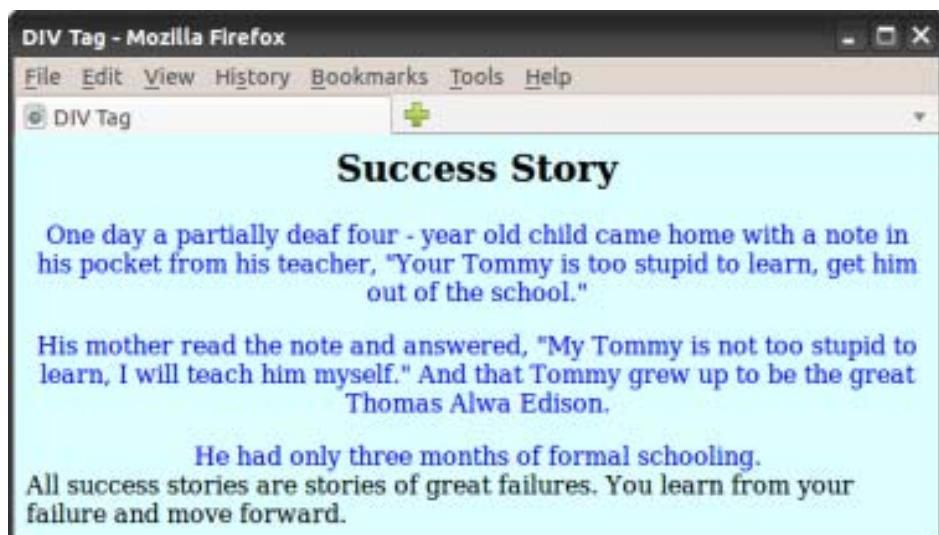
#### ഉദാഹരണം 4.13: <DIV> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
    <TITLE> DIV Tag </TITLE>
</HEAD >
```

```

<BODY Bgcolor= "#ddffff">
    <H2 Align= "center"> Success Story </H2>
    <DIV Align= "Center" Style= "Color:#0000FF"> One day a
    partially deaf four - year old child came home with a
    note in his pocket from his teacher, "Your Tommy is too
    stupid to learn, get him out of the school."
    <P>His mother read the note and answered, "My Tommy is
        not too stupid to learn, I will teach him myself." And
        that Tommy grew up to be the great Thomas Alwa Edison.
    </P> He had only three months of formal schooling.
    </DIV>
    All success stories are stories of great failures.
    You learn from your failure and move forward.
</BODY>
</HTML>

```

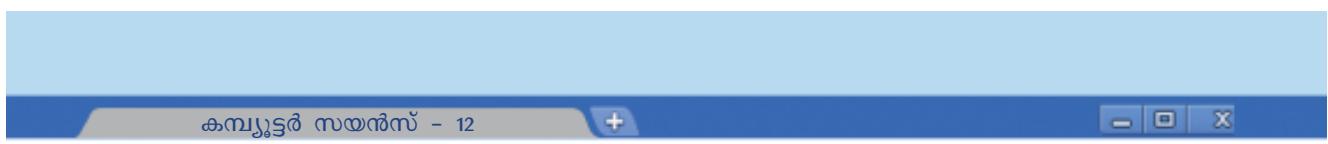


ചിത്രം 4.29 : &lt;DIV&gt; ടാഗിന്റെ ഉപയോഗം

#### 4.10.11 <FONT> - ലിപികളുടെ സാമ്പത്തികവിശേഷതകൾ നിർദ്ദേശിക്കുന്നത്

<FONT>, </FONT> എന്നീ ടാഗുകൾക്ക് ഇടയിൽ ഉൾക്കൊള്ളിച്ചിട്ടുള്ള വാക്കുങ്ങളുടെ ശൈലി, വലുപ്പം, നിരു എന്നിവ വ്യത്യാസപ്പെടുത്തുവാൻ <FONT> ടാഗ് നമ്മു സഹായിക്കുന്നു. ലിപിത്തതിലെ ഒരു ചെറിയ ഭാഗത്തിന്റെ രൂപരൂപത മാറ്റുന്നതിനാണ് ഈ സാധാരണയായി ഉപയോഗിക്കുന്നത്. <FONT> ടാഗിന്റെ ആട്ടിവ്യൂട്ടുകൾ താഴെ നന്നിട്ടുള്ളവയാണ്.

**Color :** നിരു നാമം ആയോ, ഹൈക്സാഡെസിമൽ വില ആയോ ടെക്സ്റ്റിന് നിരു നൽകുന്നു.



**Face** : അക്ഷരങ്ങളുടെ ശൈലി സൂചിപ്പിക്കുന്ന ആട്ടിബ്യൂട്ടാണ് ഈ. വില സൂചിപ്പിച്ചിട്ടില്ല എങ്കിൽ ബേഹസർ അനുകൂലിക്കുന്ന ശൈലികളിൽ ആദ്യത്തെ തന്ത്ര (Default) ശൈലിയായി ലിഖിതത്തിലെ വാക്കുങ്ങൾക്ക് നൽകുന്നു.

**Size** : അക്ഷരങ്ങളുടെ വലുപ്പം 1 മുതൽ 7 വരെയുള്ള വിലകൾ കൊണ്ട് സൂചിപ്പിക്കാൻ ഉപയോഗിക്കുന്ന ആട്ടിബ്യൂട്ടാണ് ഈ. തന്ത്ര വില 3 ആയിരിക്കും.

<FONT> ടാഗിന്റെ ഉപയോഗം ഉദാഹരണം 4.14-ൽ തനിട്ടുള്ള കോഡ് വിശദീകരിക്കുകയും അനുബന്ധ വെബ് പേജ് ചിത്രം 4.30 കാണിക്കുകയും ചെയ്യുന്നു.

#### ഉദാഹരണം 4.14: <FONT> ടാഗ് വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD> <TITLE> Font tags </TITLE> </HEAD>
<BODY Bgcolor= "#eeee8aa">
    Every success story is also a story of great failure.
    The only difference is that every time they failed,
    they bounced back. <BR><BR>
    <FONT Size="6" Face="Courier New" Color="#B22222">
        Successful people don't do great things,
        they only do small things in a great way.
    </FONT>
</BODY>
</HTML>
```

മുകളിൽ തനിട്ടുള്ള കോഡിൽ ഒരു ഭാഗം &nbsp; നാം പച്ചനിറത്തിൽ രേഖപ്പെടുത്തിയിട്ടുണ്ട്. ഈ അർമ്മമാക്കുന്നത് എന്താണ്? don't എന്ന വാക്കിലെ ഏക ഉല്ലംശി ചിഹ്നം കൊണ്ട് വരുന്നതിന് വേണ്ടിയാണ് ഈ എന്ന ലഭിക്കുന്ന വെബ് പേജിൽ നിന്നും നമുക്ക് മനസിലാക്കാനാകും. ഈ പോലെ മറ്റ് പല ചിഹ്നങ്ങളുമുണ്ട്. തുടർന്ന് വരുന്ന ഭാഗങ്ങളിൽ അവ ചർച്ച ചെയ്യാം.



ചിത്രം 4.30: <FONT> ടാഗിന്റെ ഉപയോഗം

### 4.11 സാർമ്മക ചിഹ്നങ്ങൾക്ക് വേണ്ടി HTML-ൽ ഉപയോഗിക്കുന്ന സവിശേഷ വസ്തുക്കൾ (HTML entities for reserved characters)

<, >, & തുടങ്ങിയ ചിഹ്നങ്ങൾക്ക് പ്രത്യേകമായ അർമ്മമാണ് HTML-ൽ ഉള്ളത്. മാത്രമല്ല, HTML ലിഖിതത്തിന്റെ ഉള്ളടക്കത്തിലെ വാക്കുങ്ങളിൽ ഈവ ഉൾപ്പെടുത്താനും സാധിക്കില്ല. HTML ടാഗുകളോടും ആട്ടിബ്യൂട്ടുകളോടും ഒപ്പം കാണപ്പെടുന്ന വിരാമ ചിഹ്നങ്ങൾക്ക് വേണ്ടിയും ഇവ ഉപയോഗിക്കുന്നതാണ്.

ഉള്ളപ്പോലെ ഈ ചിഹ്നങ്ങൾ ബേഹസർ കണക്കാക്കും.

ഉദാഹരണമായി, ആക്ഷിൾ ബോക്കറ്റുകളായ <, > എന്നിവ ഉപയോഗിക്കുന്നത് ടാഗുകൾ സുചിപ്പിക്കാനാണ്. വെബ് പേജിലെ വാക്യങ്ങളിൽ ഈ അടയാളങ്ങൾ നമുക്ക് എപ്പോഴുക്കിലും പ്രദർശിപ്പിക്കേണ്ടി വന്നാൽ, HTML-ലെ സവിശേഷ വസ്തുകൾ നമുക്ക് ഉപയോഗിക്കേണ്ടി വരും. പട്ടിക 4.6-ൽ സാർമ്മക ചിഹ്നങ്ങളുടെയും അനുബന്ധ ഹ്യാംഗിൾ നമ്മുടെ നിബന്ധനയിൽ ഉൾപ്പെടെയും നാമാവലി നൽകിയിരിക്കുന്നു.

ഒരു വെബ് പേജിൽ A < B & A > C എന്ന വാക്യം പ്രദർശിപ്പിക്കുന്നതിന്, **A &lt; B &amp; A &gt; C** എന്ന് HTML കോഡിൽ നൽകാം.

**പട്ടിക 4.6 : എൻറിക്കളുടെ പട്ടികയും അവയുടെ വിവരങ്ങളും**

## 4.12 HTML ലിവിത്തിൽ കമൺ്റുകൾ ഉൾപ്പെടുത്തുന്നു (Adding comments in HTML document)

Character	Entity	Description	ചുവർന്നു കാണുന്നതു മനസ്സിലാക്കാൻ കമൺ്റുകൾ നമ്മുടെ സഹായിക്കുന്നു. തയ്യാറാക്കുന്ന സോഴ്സ് കോഡിൽ കമൺ്റുകൾ ടാഗുകളുണ്ട്. കമൺ്റുകൾ ബേഹസർ വിന്ദോയിൽ കൂന എത്ത് ഉള്ളടക്കവും കമൺ്റായി കണക്കാക്കുകയും വഗണിക്കുകയും ചെയ്യുന്നു. ജിനി എഡിറ്ററിൽ കമൺ്റു ശിപ്പിക്കാറുള്ളത്. ഉദാഹരണം 4.15-ലെ കോഡ് കമൺ്റു സൗകര്യക്കയും ചിത്രം 4.31-ലെ വെബ് പേജ് അങ്ങനെയും ചെയ്യുന്നു.
&n b s p ;	Non Breaking Space		
"	&q u o t ;	Double quotation mark	
'	&a pos ;	Single quotation mark	
&	&a m p ;	Ampersand	
<	&l t ;	Less than	
>	&g t ;	Greater than	
©	&c o p y ;	Copyright Symbol	നിംഫുളം വിവരിക്കുന്നതിന് വേണ്ടിയുള്ളത്
™	&t r a d e ;	Trademark Symbol	header Starts -->
®	&r e g ;	Registered Symbol	s </TITLE>

</HEAD>

കമ്പ്യൂട്ടർ സയൻസ് - 12

```

<BODY Bgcolor= "#D8D8D8">
    <!-- This is a comment -->
    <p>The paragraph starts here. Comment statements are
    not displayed in the browser window</p>
    <!-- Comments are not displayed in the browser -->
</BODY>
</HTML>
```

ചിത്രം 4.31 : HTML കമ്മറ്റുകൾ

### നിങ്ങളുടെ പുരോഗതി അറിയുക



1. സാർമക ചിഹ്നങ്ങൾ എങ്ങനെന്താണ് HTML ലെ സൂചിപ്പിക്കുന്നത്?
2. \_\_\_\_\_ ടാഗിനോട് ഒപ്പുമാണ് Face ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്.
3. <FONT> ടാഗിന്റെ ആട്ടിബ്യൂട്ടുകളുടെ നാമാവലി തയാറാക്കുക.
4. <PRE> ടാഗിന്റെ ഉപയോഗമെന്താണ്?
5. ഒരു ടെക്നോളജി സ്ക്രോളിംഗ് ടൈപ്പ് ടാഗ് ഉപയോഗിക്കുന്നു.
6. <MARQUEE> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്ടിബ്യൂട്ടുകൾ ഏതെല്ലാമാണ്?
7. <ADDRESS> ടാഗിന്റെ ഉപയോഗമെന്താണ്?
8. <FONT> ടാഗിലെ അക്ഷരങ്ങളുടെ തനത് വലുപ്പം എത്രയായിരിക്കും?
9. <DIV> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്ടിബ്യൂട്ടുകളുടെ പേരെഴുതുക.

### 4.13 ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുന്നു (Inserting images)

ഉള്ളടക്ക വിവരണം കുടുതൽ ആകർഷകവും വിനിമയാത്മകവും ആകുന്നത് ചിത്രങ്ങളാണ്. ഇക്കാലത്ത് വെബ് സൈറ്റുകൾ അധികവും ചിത്രങ്ങളാൽ സമ്പന്നമാണ്. വെബ് വികസിപ്പിക്കുന്നതിന് വേണ്ട ധാരാളം സവിശേഷതകളുള്ള പുതിയ HTML പതിപ്പുകളിൽ പോലും ചിത്രങ്ങൾ ഉൾപ്പെടുത്തുന്നതിനുള്ള കോഡ് വളരെ ലളിതമാണ്. HTML പേജുകളിൽ ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുന്നതിന് <IMG> ടാഗ് ഉപയോഗിക്കുന്നു. വളരെ ലളിതമായി ഈ ടാഗ് ഉപയോഗിക്കുന്ന വിധം താഴെ കാടുത്തിരിക്കുന്നു.

<IMG Src = "picture1.jpg">

<IMG> ടാഗ് ഒരു എംറ്റി ആയ ടാഗാണ്. മാത്രമല്ല, അതിന് ധാരാളം ആട്ടിബൃട്ടുകളും ഉണ്ട്. പേജിൽ ഉൾക്കൊള്ളിക്കേണ്ട ചിത്രത്തിന്റെ ഫയൽ നാമം സുചിപ്പിക്കുന്ന Src ആട്ടിബൃട്ടുകാണ് എറ്റവും പ്രധാനപ്പെട്ടത്. നമ്മുടെ ആവശ്യത്തിന് അനുസരിച്ച് JPEG, PNG, GIF എന്നീ ചിത്രഫലങ്ങൾ ഉപയോഗിക്കാനുമെങ്കിലും, ശരിയായ ഫയൽ നാമവും അതോടൊപ്പുള്ള എക്സ്റ്റൻഷൻും Src ആട്ടിബൃട്ട് ഉപയോഗിച്ച് സുചിപ്പിച്ചിട്ടുള്ള കണക്ക് ഉറപ്പ് വരുത്തണം. ചിത്രഫല ഇപ്പോൾ ഉപയോഗത്തിലുള്ള ഫോർമാറ്റിൽ ഇല്ലെങ്കിൽ, അത് എവിടെയാണോ ലഭ്യമാകുന്നത് അവിടേക്കുള്ള വഴി (പാത) / URL നാം സുചിപ്പിക്കണം.

### ചിത്രത്തിന് വേണ്ടി സ്ഥലം ഒരുക്കുന്നു (Setting space for the image)

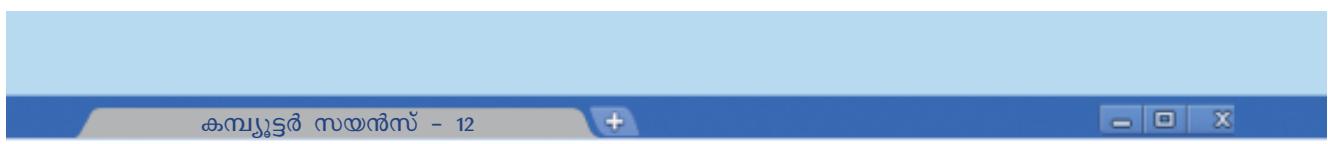
**Width, Height** എന്നീ ആട്ടിബൃട്ടുകൾക്ക് വിലകൾ നൽകിക്കൊണ്ട് വെബ് പേജിൽ ചിത്രത്തിനുവേണ്ടിയുള്ള ഇടം നമുക്ക് ഒരുക്കാം. പിക്സലിന്റെ എല്ലാമായോ അല്ലെങ്കിൽ യഥാർമ്മ വലുപ്പത്തിന്റെ ശതമാനമായോ വിലകൾ നൽകാവുന്നതാണ്. ഈ ആട്ടിബൃട്ടുകൾ ഉപയോഗിച്ചിട്ടില്ലെങ്കിൽ ചിത്രത്തിന്റെ യഥാർമ്മ വലുപ്പത്തിൽ ബോസർ അത് പ്രദർശിപ്പിക്കും.

ചിത്രങ്ങൾക്ക് ഇടയ്ക്കുള്ള ശുന്നസ്ഥലങ്ങൾ എങ്ങനെ ക്രമപ്പെടുത്താമെന്ന് നമുക്ക് ചർച്ച ചെയ്യാം. ഒരു ജാലകത്തിൽ (വിൻഡോയിൽ) ചിത്രങ്ങൾ സ്ഥാപിക്കുന്നോൾ രണ്ട് രീതി തിലാണ് ശുന്ന സ്ഥലങ്ങൾ അവയ്ക്കിടയിൽ വരാറുള്ളത്-തിരഞ്ഞീകരണയും, ലംബമായും. വെബ് പേജിൽ തിരഞ്ഞീകരണമായും ലംബമായും ചിത്രങ്ങൾക്കിടയ്ക്ക് സ്ഥലം നൽകുന്ന തിന് HTML-ൽ ലഭ്യമാകുന്ന രണ്ട് ആട്ടിബൃട്ടുകളാണ് **Hspace, Vspace** എന്നിവ.

രണ്ട് രീതിയിൽ ചിത്രങ്ങൾക്ക് ഇടയ്ക്ക് ശുന്നസ്ഥലം നൽകുന്നത് ഉദാഹരണം 4.16-ൽ തന്നെ ഇടയ്ക്കുള്ള HTML കോഡ് വിശദീകരിക്കുന്നു. അനുബന്ധ വെബ് പേജ് ചിത്രം 4.31-ൽ കാണിച്ചിരിക്കുന്നു.

**ഉദാഹരണം 4.16:** ചിത്രങ്ങൾക്കിടയിൽ പലതരത്തിലുള്ള ശുന്നസ്ഥലം നൽകുന്ന തിന് വേണ്ടിയുള്ളത്

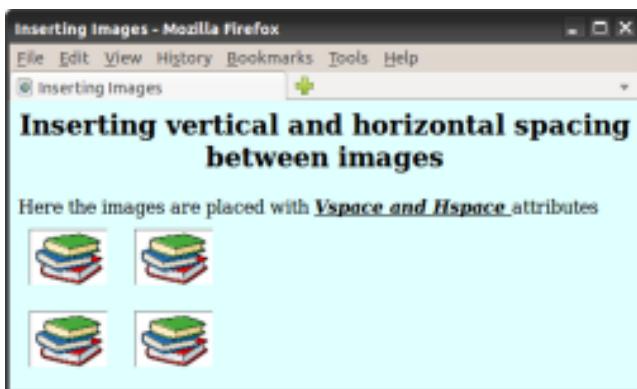
```
<HTML>
<HEAD>
    <TITLE> Inserting Images </TITLE>
</HEAD>
<BODY Bgcolor="#E0FFFF">
    <H2 Align="center">Inserting vertical and horizontal
                           spacing between images</H2>
    Here the images are placed with <B><I><U> Vspace and
    Hspace </U></I></B> attributes <BR>
    <IMG Src= "book3.jpg" Height= "50" Width= "70"
```



```
Vspace= "10" Hspace= "10">
<IMG Src= "book3.jpg" Height= "50" Width= "70"
Vspace= "10" Hspace= "10"> <BR>
<IMG Src= "book3.jpg" Height= "50" Width= "70"
Vspace= "10" Hspace= "10">
<IMG Src= "book3.jpg" Height= "50" Width= "70"
Vspace= "10" Hspace= "10"> <BR>
</BODY>
</HTML>
```

നിർദ്ദിഷ്ട വീതിയിലും ഉയരത്തിൽ  
ലുമാൺ ചിത്രങ്ങൾ വിന്റുസിച്ചിട്ടു  
ഇള്ള തെന്നും തിരശ്വീനമായും  
ലാംബമായും നൽകിയ ശുന്ന  
സ്ഥലം സുചിപ്പിച്ചിട്ടുണ്ട് പോലെയാണ് ചിത്രങ്ങൾക്കിടയിൽ  
ഒരു അകലമെന്നും ചിത്രം 4.32-ൽ  
കാണാൻ സാധിക്കും.

വാക്യങ്ങൾ ആരംഭിക്കുന്ന വരീ  
അടിസ്ഥാനമാക്കി ചിത്രങ്ങൾ  
വിന്റുസിക്കുന്നതിന് ഉപയോഗിക്കുന്ന <IMG> ടാഗിന്റെ Align  
ആട്ടിബ്യൂട്ടിനേക്കുറിച്ച് നമുക്ക് ചർച്ച ചെയ്താം. ഈ ആട്ടിബ്യൂട്ടിന് നൽകാനാകുന്ന വിലകൾ താഴെക്കാടുത്തിരിക്കുന്നു.



ചിത്രം 4.32 : നിർദ്ദേശിച്ച വലുപ്പങ്ങളിലുള്ളതും, ഇടയ്ക്കൽ തിരികെടുത്തും ലാംബവുമായി ശുന്നസ്ഥലങ്ങൾ ഉൾക്കൊള്ളിച്ചിട്ടുണ്ട്

**Bottom** : ചിത്രത്തിന്റെ കീഴ്ഭാഗം വാക്യങ്ങൾ ആരംഭിക്കുന്ന വരിയുമായി പൊരുത്ത പെട്ടിക്കുന്നു.

**Middle** : ചിത്രത്തിന്റെ മധ്യഭാഗം (ലാംബമായി) വാക്യങ്ങൾ ആരംഭിക്കുന്ന വരിയുമായി പൊരുത്തപെടുന്നു.

**Top** : വാക്യങ്ങളുടെ മുകൾഭാഗവുമായി ചിത്രം പൊരുത്തപെടുന്നു.

Align ആട്ടിബ്യൂട്ടിന് വിലകൾ നൽകുന്നേം ഉണ്ടാകുന്ന പ്രതീതി നമുക്ക് കാണാം. ഉദാഹരണം 4.17, ചിത്രം 4.33 എന്നിവ ഇത് വിശദീകരിക്കുന്നു.

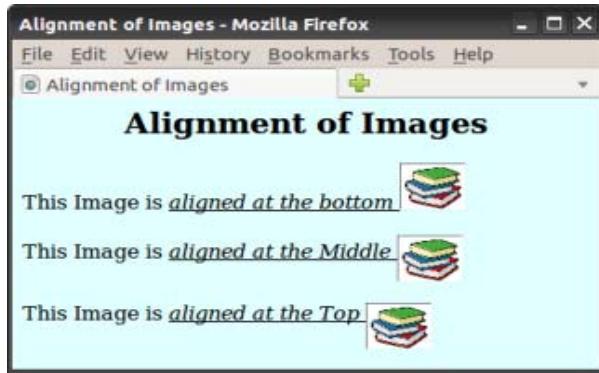
**ഉദാഹരണം 4.17:** ഒരു ചിത്രത്തിന് വ്യത്യസ്ത പൊരുത്തപെട്ടലുകൾ നൽകുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
    <TITLE> Alignment of Images </TITLE>
</HEAD>
```

#### 4. വെബ് സാങ്കേതിക വിദ്യ

```
<BODY Bgcolor= "#E0FFFF">
    <H2 Align= "center">Alignment of Images</H2>
    This Image is <I><U>aligned at the bottom </U></I>
    <IMG Src= "book3.jpg" Height= "40" Width= "50"
        Align= "Bottom"> <BR> <BR>
        This Image is <I><U>aligned at the Middle </U></I>
    <IMG Src= "book3.jpg" Height= "40" Width= "50"
        Align= "Middle"> <BR> <BR>
        This Image is <I><U>aligned at the Top </U></I>
    <IMG Src= "book3.jpg" Height= "40" Width= "50" Align="Top">
</BODY>
</HTML>
```

<IMG> ടാഗിൽ ഏറ്റവും മുൻ ചില വിലകൾ കുറിയുണ്ട്. അവ പിരുത്തെത്ത ബോസർ ജാലകത്തിൽ ഇടത് ഭാഗത്തെക്കും വലതുഭാഗത്തെക്കും യമാടകമാണ് വിന്യസിക്കുന്ന left - ഉം right - ഉം ആണ്.



ചിത്രം 4.33 : വാക്യവുമായുള്ള പിരുത്തെത്തിൽ വിവിധ പൊതുത്തെപ്പട്ടംകൾ



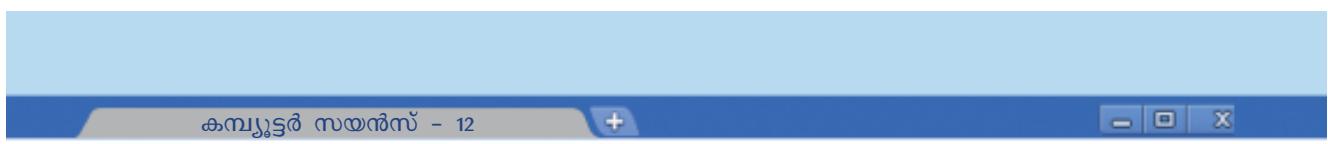
ചിത്രം 4.34 : ബോസർ വിൻഡോയ്ക്ക് ഇടത്തോട് വലതുമായി വിന്യസിക്കുന്നു ചിത്രങ്ങൾ

ഈ തെരോട്ടും വലതോട്ടുമുള്ള ബോസർ ജാലകത്തിലെ പിരുത്തെത്തിൽ വിന്യാസം ചിത്രം 4.34 തുടർന്നെല്ലാം കാണിച്ചിട്ടുള്ളത് നിരീക്ഷിക്കുക. ഈ പേജിന് വേണ്ട HTML ലിഖി നമുക്ക് ചെയ്യാം തമെഴുതുക (തയാറാക്കുക).



#### ഒരു പിരുത്തിന് ബോധർ നൽകുന്നു (Setting border around an image)

ഒരു വെബ് പേജിൽ ഉൾപ്പെടുത്തിയിട്ടുള്ള പിരുത്തിന് നമുക്ക് ബോധർ നൽകണമെന്ന് വിചാരിക്കുക. <IMG> ടാഗിൽ Border എടുവുക ഉപയോഗിച്ച് ഈ സാധ്യമാക്കാം. ഏറ്റവും അനുയോജ്യമായ വില നൽകിക്കൊണ്ട് ബോധറിന് കനം നൽകാം. ഉദാഹരണം 4.18-ലെ HTML കോഡും ചിത്രം 4.35-ലെ അനുബന്ധ വെബ് പേജും Border എടുവുകിൽ ഉപയോഗം നമുക്ക് കാണിച്ചു തരുന്നു.



#### ഉദാഹരണം 4.18: ചിത്രത്തിന് സോൾ നൽകുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD>
    <TITLE> Inserting Images </TITLE>
</HEAD>
<BODY Bgcolor= "#E0FFFF">
    <H2 Align= "center">Inserting Border to Images</H2>
    Here is an image <B><I><U> with Border </U></I></B>
    attribute
    <IMG Src=
        "book3.jpg"
        Height= "50"
        Width= "70"
        Border= "5">
    <BR>Here is
    an image<B><I><U>
    without Border</
    U></I></B>
    attribute
    <IMG Src= "book3.jpg" Height= "50" Width= "70">
</BODY>
</HTML>
```



ചിത്രം 4.35 : സോൾ സൗകര്യം ഉള്ളതും മല്ലാത്തതുമായ ചിത്രങ്ങൾ

<IMG> ടാഗിന്റെ വിവിധ ആട്ടിബ്യൂട്ടുകളെക്കുറിച്ചും അവ ചിത്രത്തിൽ അവയുണ്ടാകുന്ന വ്യതിയാനങ്ങളെക്കുറിച്ചും നാം പറിച്ച് കഴിഞ്ഞു. Src ആട്ടിബ്യൂട്ടിൽ നിർദ്ദേശിച്ച ചിത്രമായിൽ കണ്ണെത്താനായില്ല എങ്കിൽ വെബ് പേജ് കാഴ്ചയിൽ എങ്ങനെന്നുണ്ടാകും? വെബ് പേജിൽ ചിത്രത്തിനായി നിർദ്ദേശിച്ച സ്ഥലം ശുന്നുമായിരിക്കും. മറ്റ് ചില കാരണങ്ങളാലും ബൈറ്റസിന് ചിത്രം പ്രദർശിപ്പിക്കാനാകാതെ വരാറുണ്ട്. അത്തരം സന്ദർഭങ്ങളിൽ, ചിത്രത്തിനു പകരം ഒരു വാക്കും ബൈറ്റസിൽ നമുക്ക് ദൃശ്യമാക്കാനാകും. ബൈറ്റസിൽ ചിത്രം പ്രദർശിപ്പിക്കാനാകാതെ വന്നാൽ പകരം ഒരു വാക്കും സൂചിപ്പിക്കാൻ HTML നൽകുന്ന ആട്ടിബ്യൂട്ടാണ് **Alt**. ഉദാഹരണം 4.19-ലെ കോഡ് Alt ആട്ടിബ്യൂട്ടിന്റെ ഉപയോഗം വിശദീകരിക്കുന്നു. അങ്ങനെ ലഭിക്കുന്ന വെബ് പേജ് ചിത്രം 4.36-ൽ കാണിച്ചിരിക്കുന്നു.

#### ഉദാഹരണം 4.19: <IMG> ടാഗിലെ Alt ആട്ടിബ്യൂട്ടിന്റെ ഉപയോഗം വിശദീകരിക്കുന്നതിന് വേണ്ടിയുള്ളത്

```
<HTML>
<HEAD> <TITLE> Inserting Images </TITLE> </HEAD >
<BODY Bgcolor= "#E0FFFF">
    <H2 Align= "center">Inserting Images</H2>
```

#### 4. വെബ് സാങ്കേതിക വിദ്യ

If the browser cannot display the image, then the text entered in the `<B>Alt</B>` attribute will be displayed.  
`<IMG Src= "book5.jpg" Height= "20%" Width= "20%"`  
`Alt= "Image of an opened book"> <BR>`  
`</BODY>`  
`</HTML>`



ചിത്രം 4.36 : `<IMG>` ടാഗിലെ Alt ആട്ടിവ്യൂട്ടിന്റെ ഉപയോഗം

### നിങ്ങളുടെ പുരോഗതി അറിയുക



- ഒരു HTML ലിഖിതത്തിൽ ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുന്നതിനായി \_\_\_\_\_ ടാഗ് ഉപയോഗിക്കുന്നു.
- `<IMG>` ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്ടിവ്യൂട്ടാണ് \_\_\_\_\_.
- `<IMG>` ടാഗിൽ Alt ആട്ടിവ്യൂട്ടിന്റെ ഉപയോഗമെന്ത്?
- ഒരു ചിത്രം ഒരു പ്രത്യേക വലുപ്പത്തിൽ പ്രദർശിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്ന ആട്ടിവ്യൂട്ടുകളുടെ പേരെഴുതുക.
- രണ്ട് ചിത്രങ്ങൾക്കിടയിൽ തിരഞ്ഞീനവും ലംബവുമായ സൂലങ്ങൾ ലഭ്യമാക്കുന്ന ആട്ടിവ്യൂട്ടുകൾ ഏതൊക്കെയാണ്?



### നമുക്ക് സംഗ്രഹിക്കാം

ഇള്ളർന്നെറ്റിന്റെ സഹായ നിർണ്ണയിക്കുന്ന ഒരു ഉടക്കമാണ് സുരക്ഷിതമായ ആശയ വിനിമയം. ഇള്ളർന്നെറ്റിലുണ്ടുള്ള വ്യവഹാരങ്ങളുടെ സുരക്ഷിതത്വം നിർവ്വഹിക്കുന്നത് HTTPS, ഡിജിറ്റൽ സാക്ഷ്യപത്രങ്ങൾ എന്നിവ ഉപയോഗിച്ചാണ്. ഡാറ്റ സംഭരിക്കുന്ന തിനും കൈമാറ്റം (വിനിമയം) നടത്തുന്നതിനുമായി ഉപയോഗിക്കുന്ന വെബ് സെർവ്വർ, സോഫ്റ്റ്‌വെയർ പ്രോഡക്ടുകൾ, DNS സെർവ്വറുകൾ തുടങ്ങിയ അടിസ്ഥാന സൗര്യങ്ങൾ ഇള്ളർന്നെറ്റിന് ആവശ്യമാണ്. ഒന്നോ അതിലധികോ വെബ് പേജുകൾ ചേർന്നാണ് വെബ് സെറ്റ് ഉണ്ടായിരിക്കുന്നത്. HTML കോഡ് എഴുതിയോ അല്ലെങ്കിൽ വെബ് രൂപകൽപ്പന ചെയ്യാനുപയോഗിക്കുന്ന സോഫ്റ്റ്‌വെയറുകളുടെ സഹായത്താലോ നമുക്ക് വെബ് പേജുകൾ നിർമ്മിക്കാം. വെബ് പേജുകൾ പരിവർത്തനാത്മകമെന്നും സ്ഥിരമായതെന്നും തരം തിരിച്ചിട്ടുണ്ട്. സ്ക്രിപ്റ്റുകൾ ഉപയോഗിച്ചുകൊണ്ട് പരിവർത്തനാത്മക പേജുകൾ വികസിപ്പിക്കാം. കൂയൻ്റ് ഭാഗത്തെ സ്ക്രിപ്റ്റിങ്ങ് ഭാഷകളായ ജാവ സ്ക്രിപ്റ്റ്, VB സ്ക്രിപ്റ്റ് തുടങ്ങിയവ ഉപയോഗിച്ച് കൂയൻ്റ് ഭാഗത്ത് ഡാറ്റയുടെ സാധൃത ഉറപ്പിക്കാനും. പരി

വർത്തനാത്മക വെബ് പേജുകൾ സെർവിൽ നിർമ്മിക്കുന്നതിനായി സെർവർ ഭാഗത്തെ സ്ക്രിപ്റ്റുകളായ PHP, ASP തുടങ്ങിയവ ഉപയോഗിക്കുന്നു. വെബ് സെറ്റിന് ആകമാനം ഒരു ഏകീകൃത ശൈലി ലഭ്യമാക്കാൻ കാസ്കേഡിംഗ് റെസൽ ഷൈറ്റ് (CSS) ഉപയോഗിക്കുന്നു. HTML ലോജിക്കൽ വെബ് പേജ് രൂപപ്പെടുത്തലിനേയും കുറിച്ചുള്ള അടിസ്ഥാന ആശയങ്ങൾ നാം ചർച്ച ചെയ്തു. വ്യത്യസ്തങ്ങളായ ടാഗുകളേയും അവയുടെ പ്രധാനപ്പെട്ട ആട്ടിബ്യൂട്ടുകളേയും നാം പരിചയപ്പെട്ടു. ചില ടാഗുകൾ ജോഡിക്കളായതും മറ്റ് ചിലത് ജോഡിക്കളാൽ പരിപൂരിപ്പിച്ചു അണ്ട്. വിവിധ ടാഗുകൾ കൊണ്ട് വാക്കുങ്ങൾ ഭാഗിയായി വിനൃസിച്ച് നമുക്കൊരു വെബ് പേജ് നിർമ്മിക്കാം. മാർക്കൂകൾ, ചിത്രങ്ങൾ, ശബ്ദങ്ങൾ, ചലിക്കുന്ന ചിത്രങ്ങൾ എന്നിവ ഉൾപ്പെടുത്തി വെബ് പേജുകളുടെ ഭാഗി വർധിപ്പിക്കാം. ഈ അധ്യായത്തിൽ ചർച്ച ചെയ്ത ആശയങ്ങളെക്കുറിച്ച് വ്യക്തമായ ധാരണയും HTML ലിഖിതങ്ങൾ നിർമ്മിക്കുന്നതിന് വേണ്ട പ്രായോഗിക പരിചയവും നമുക്ക് ഉള്ളതു കൊണ്ട് തുടർന്ന് വരുന്ന അധ്യായങ്ങളിലെ ആശയങ്ങൾ എളുപ്പത്തിൽ ഗ്രഹിക്കാവുന്ന താണ്. കൂടാതെ, മനോഹരമായ വെബ് സെറ്റുകളുടെ മാതൃകയുണ്ടാക്കുന്നതിനും, വെബ് ആപ്ലിക്കേഷൻകൾ സ്വയം നിർമ്മിക്കുന്നതിനും നാം പ്രാപ്തരാകുകയും ചെയ്യും.



## നമുക്ക് പരിശീലനക്കാം

1. താഴെ കോടുത്തിരിക്കുന്ന സവിശേഷതകളോടുകൂടിയ കേരളത്തിന്റെ ഒരു വെബ് പേജ് നിർമ്മിക്കുന്നതിന് വേണ്ട HTML കോഡ് എഴുതുക.
  - വാക്കുങ്ങൾക്ക് രൂപപ്പെടുത്തുന്ന ടാഗുകളും ആട്ടിബ്യൂട്ടുകളും ഉപയോഗിച്ച് ഒരു തലക്കെട്ടും അണ്വെ വാചകങ്ങളുമുള്ള ഒരു വണ്ണിക തയാറാക്കുക.
  - വെബ് പേജിന്റെ പശ്ചാത്തല ചിത്രമായി ഒരു പ്രകട്ടി ദൃശ്യം വേണം.
2. ചുവവുടെ കോടുത്തിരിക്കുന്ന വിശദാംശങ്ങളും സവിശേഷതകളുമുള്ള നിങ്ങളുടെ സ്കൂൾ വെബ് പേജിന് ആവശ്യമായ HTML കോഡെഴുതുക.
  - വാക്കുങ്ങൾ രൂപപ്പെടുത്തുന്ന ടാഗുകളും ആട്ടിബ്യൂട്ടുകളും ഉപയോഗിച്ച് കൊണ്ട് ഓരോ ജില്ലയ്ക്കും ഒരു തലക്കെട്ടും തുടർന്ന് മുന്ന് വാചകങ്ങളുള്ള ഒരു വണ്ണികയും തയാറാക്കുക.
  - പേജിന്റെ പശ്ചാത്തലത്തിന് ഒരു നിറം നൽകുക.
  - സ്കൂളിന്റെ ഒരു ചിത്രം നൽകുക.
3. ചുവവുടെ കോടുത്തിരിക്കുന്ന വിശദാംശങ്ങളും സവിശേഷതകളുമുള്ള നിങ്ങളുടെ സ്കൂൾ വെബ് പേജിനാവശ്യമായ HTML കോഡെഴുതുക.
  - വാക്കുങ്ങൾ രൂപപ്പെടുത്തുന്ന ടാഗുകളും ആട്ടിബ്യൂട്ടുകളും ഉപയോഗിച്ച് കൊണ്ട് ഓരോ ജില്ലയ്ക്കും ഒരു തലക്കെട്ടും തുടർന്ന് മുന്ന് വാചകങ്ങളുള്ള ഒരു വണ്ണികയും തയാറാക്കുക.
  - സ്കൂളിന്റെ തപാൽ മേൽവിലാസം ഉൾപ്പെടുത്തുക.
  - “പുതിയ അധ്യായ വർഷത്തെക്കുള്ള പ്രവേശനം മെയ് മാസം 10-ാം തീയതി ആരംഭിക്കുന്നു” എന്ന മാർക്കൂ ഉൾപ്പെടുത്തുക.

4. ചുവടെ കൊടുത്തിരിക്കുന്ന സവിശേഷതകൾ ഉൾപ്പെടുത്തി നമ്മുടെ ദേശീയ ഗാന ത്തിന്റെ വരികൾ പ്രദർശിപ്പിക്കുന്ന ഒരു വെബ് പേജിന് ആവശ്യമായ HTML കോഡ് എഴുതുക.
- വ്യത്യസ്ത സവിശേഷതകൾ ഉള്ള ഫോൺ ഉപയോഗിച്ച് ഒരു തലക്കെട്ട് ഉണ്ടാക്കണം.
  - നമ്മുടെ ദേശീയ പതാകയുടെ ഒരു ചിത്രം.

### നമുക്ക് വിലയിരുത്താം |||

1. ഇൻറർനെറ്റിലുണ്ടായുള്ള ധാരാധൈദ സഖാരത്തിൽ റൂട്ടറുകളുടെ പങ്ക് എന്ത്?
2. സമൂഹ മാധ്യമ വെബ് സൈറ്റുകൾ ആശയവിനിമയത്തിന് അവരുടെ സ്വന്തം പ്രോട്ടോക്കോളി ഉപയോഗിക്കുന്നത്. ഇൻറർനെറ്റിൽ TCP/IP പ്രോട്ടോക്കോൾ ഉപയോഗിക്കുന്നത് കൊണ്ട് ഇതെങ്ങനെ സാധ്യമാക്കും?
3. ഇ-മെയിൽ അക്കൗണ്ടിലേക്ക് യുസർനാമവും പാസ്വോഡ്യും ഇൻറർനെറ്റിലുണ്ട് സുരക്ഷിത ക്ഷിതമായാണ് അയയ്ക്കേണ്ടത്.
  - a. സൈറ്റിലേക്ക് ധാരാ അയക്കാൻ ഉപയോഗിക്കുന്ന സുരക്ഷിത സാങ്കേതിക വിദ്യ യുടെ പേരെഴുതുക.
  - b. സുരക്ഷിത ധാരാവിനിമയത്തിന് ഇ സാങ്കേതിക വിദ്യ എങ്ങനെ സഹായിക്കുന്നു?
4. ഓൺ-ലൈനിലുണ്ടായുള്ള വ്യവഹാരത്തിൽ പേരെന്തു് ശേറ്റ് വേയുടെ ധർമമെന്താണ്?
5. ABC എഞ്ചിനിയറിങ് കോളേജിന്റെ കാമ്പസിൽ ഏകദേശം 1000 കമ്പ്യൂട്ടറുകൾ ഇൻറർനെറ്റുമായി ബന്ധിപ്പിച്ചിട്ടുണ്ട്. ഒരു പ്രാദേശിക സൈറ്റിൽ ഇതു സാങ്കേതികവിദ്യാഭ്യാസ സ്ഥാപനത്തിൽ ഉണ്ടാകുന്നതിന്റെ നേട്ടമെന്താണ്?
6. ഒരു വെബ് സൈറ്റിൽ ഓഫോറ്റോം സിസ്റ്റത്തിനും വെബ് സൈറ്റിൽ പാക്കേജിനും ഉദാഹരണമെഴുതുക.
7. വെബ് സൈറ്റിൽ സോഫ്റ്റ്‌വെയർ പോർട്ടീറ്റു് ഉപയോഗമെന്ത്?
8. HTTP ക്ക് ഉപയോഗിച്ചിട്ടുള്ള പോർട്ടാണ് \_\_\_\_\_.
9. നിങ്ങൾ www.prdkerala.org എന്ന വെബ്സൈറ്റ് പതിശോധിക്കുകയാണെന്ന് കരുതുക. DNS എങ്ങനെയാണ് IP മേൽവിലാസം വേർത്തിരിച്ചെടുക്കുന്നതെന്ന് വിശദിക്കുക.
10. സ്ക്രിപ്റ്റുകൾ എന്നാൽ എന്ത്? സ്ക്രിപ്റ്റ് എഴുതാൻ ഉപയോഗിക്കുന്ന വിവിധ ഭാഷകൾ വിശദിക്കുക.
11. നിങ്ങളുടെ സ്കൂൾ വെബ് സൈറ്റിലെ ഹോം പേജും പതിനേന്നാം കൂസിലെ പരീക്ഷാഫലങ്ങൾ കാണിക്കുന്ന വെബ് പേജും പതിനേന്നിക്കുക.
  - a. ഇ രണ്ട് പേജുകളും തമ്മിൽ നിർമ്മാണത്തിലുള്ള വ്യത്യാസങ്ങൾ താരതമ്യം ചെയ്യുക.
  - b. ഇ വെബ് പേജുകൾ വികസിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്ന സാങ്കേതിക വിദ്യകളെക്കുറിച്ച് എഴുതുക.

12. a. ജാവ സ്ക്രിപ്റ്റ് മാലിന്റെ ഫയൽ എക്സിഞ്ചറിൽ \_\_\_\_\_ ആണ്.
- b. ഒരു വൈബ് പേജിൽ ജാവ സ്ക്രിപ്റ്റിന്റെ പ്രധാനപ്പെട്ട രണ്ട് ഉപയോഗങ്ങളും തുക.
13. എന്താണ് അജാക്സ്? അതിന്റെ ഉപയോഗമെന്ത്?
14. നിങ്ങളുടെ കൂസിലെ കുട്ടികൾക്ക് ഓരോ പരീക്ഷയ്ക്കും ലഭിച്ച മാർക്ക് പ്രദർശിപ്പിക്കുന്നതിനായി രവി ഒരു വൈബ് പേജ് നിർമ്മിക്കാൻ ആഗ്രഹിക്കുന്നു.
  - a. അത് നിർവഹിക്കുന്നതിന് ഒരു സാങ്കേതിക വിദ്യ നിർദ്ദേശിക്കുക.
  - b. നിങ്ങളുടെ നിർദ്ദേശം ന്യായീകരിക്കുക.
15. MySQL-ലെ ഡാറ്റാബേസ് ഉപയോഗിച്ചുകൊണ്ട് PHP -യിൽ മനോജ് ഒരു വൈബ് സെറ്റ് വികസിപ്പിക്കുവാൻ ആലോചിക്കുന്നു. ലിനക്സ് വൈബ് ഹോസ്റ്റിംഗ് സംവിധാനമാണ് അയാൾ ഉപയോഗിക്കുന്നത് എങ്കിൽ, അതിന്റെ നിർവഹണത്തിന് ആവശ്യമായവ എന്തെല്ലാമാണ്?
16. “മികവൊറും എല്ലാ വൈബ് സെറ്റുകളും വികസിപ്പിക്കുവോൾ CSS ഉപയോഗിക്കാറുണ്ട്. വൈബ് സെറ്റുകളിൽ CSS ഉപയോഗിക്കുന്നത് കൊണ്ടുള്ള നേടങ്ങൾ എന്തോക്കെയാണ്?”
17. HTML വികസിപ്പിച്ചത് ആരാണ്?
18. HTML-ൽ പ്രധാനമായും രണ്ട് ഭാഗങ്ങളാണുള്ളത്. അവയുടെ പേരുകൾ എഴുതുക.
19. നിങ്ങൾ വൈബ് സെറ്റുകൾ വിശകലനം ചെയ്യുകയാണെങ്കിൽ ലിങ്കുകൾ, സന്ദർഭിച്ച ലിങ്കുകൾ, പശ്വാത്തലം തുടങ്ങിയവയ്ക്ക് വിവിധ നിങ്ങൾ കാണാനാകും. HTML-ൽ ഈ എങ്ങനെ ചെയ്യുന്നുവെന്ന് ഉദാഹരണസഹിതം വിശദമാക്കുക.
20. HTML-ലെ ജോഡിയായ ടാഗുകളും ജോഡിയല്ലാത്ത ടാഗുകളും ഉദാഹരണസഹിതം താരതമ്യം ചെയ്യുക.
21. Alink ആട്ടിബ്യൂട്ടിന്റെ തന്ത്രാധികാരി നിരം ഏതാണ്?
22. Valink ആട്ടിബ്യൂട്ടിന്റെ തന്ത്രാധികാരി നിരം ഏതാണ്?
23. താഴെ തന്നിട്ടുള്ള HTML വാക്കുകളെ തരംതിരിക്കുക.  
BR, IMG, ALIGN, FONT, FACE
24. Noshade ആട്ടിബ്യൂട്ട് ഉള്ള ടാഗിന്റെ പേരെഴുതുക.
25. ഒരു വൈബ് പേജിൽ ചിത്രഫലവുകൾ ഉൾക്കൊള്ളിക്കുന്നതിന് ഉപയോഗിക്കുന്ന <IMG> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്ടിബ്യൂട്ട് ഏത്?
26. <IMG> ടാഗിൽ Alt ആട്ടിബ്യൂട്ടിന്റെ ആവശ്യകത പ്രസ്താവിക്കുക.
27. <IMG> ടാഗ് ഉപയോഗിക്കുവോൾ ലഭിക്കുന്ന ഒരു ചിത്രത്തിന്റെ തന്ത്രാധികാരി നിരം \_\_\_\_\_ ആകുന്നു.
28. <FONT> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്ടിബ്യൂട്ടുകളുടെ നാമാവലി തയാറാക്കുക.



## പാഠനേട്ടങ്ങൾ

- ഈ അധ്യായത്തിന്റെ പാഠനലിംഗം പരിതാവ്
- വെബ് പേജുകളിലെ വിവരങ്ങൾ ഫലപ്രദമായി അവതരിപ്പിക്കുവാൻ വിവിധ ഇന്തലിന്റെ ഉപയോഗിക്കുന്നു.
  - ഫോപ്പർ ലിങ്കിൾ പ്രസക്തി തിരിച്ചറിഞ്ഞ വിവിധ തരം ലിങ്കിങ്ങിനായി <A> ടാഗ് ഉപയോഗിക്കുന്നു.
  - <EMBED> ടാഗ് ഉപയോഗിച്ച് ഓഡിയോ വീഡിയോ എന്നിവ വെബ് പേജിൽ ഉൾപ്പെടുത്തുന്നു.
  - വെബ് പേജിനുള്ളിൽ പശ്വാത്തലസംഗ്രഹവും വീഡിയോയും ഉൾപ്പെടുത്തുന്നു.
  - ടേബിൾ നിർമ്മിക്കുവാൻ അവധ്യമായ ടാഗുകളുടെ അവയുടെ ആട്ടിബ്യൂട്ടുകളുടെയും പട്ടിക തയാറാക്കി വിശദീകരിക്കുന്നു.
  - പലവിധ സവിശേഷതകൾ ഉള്ള ടേബിളിന്റെ രൂപകൽപ്പനയ്ക്കായി <TABLE> ടാഗുമായി ബന്ധപ്പെട്ട ടാഗുകൾ ഉപയോഗിക്കുന്നു.
  - വ്യത്യസ്തങ്ങളായ ടേബിൾ രൂപകൽപ്പനക്കായി വേണ്ട ടാഗുകളും അവയുടെ ആട്ടിബ്യൂട്ടുകളും മനസ്സിലാക്കുന്നു.
  - വെബ് പേജിനുള്ളിൽ ഫോറ്മേറ്റിനുള്ളിടെ അവയുടെ ശൃംഖല തിരിച്ചറിയുന്നു.
  - ഫോറ്മേറ്റിനുള്ളിൽ ഒരു ബോളഡ് വിൻഡോ തിൽ തന്നെ ഓൺലൈൻ വെബ് പേജുകൾ പ്രദർശിപ്പിക്കുന്നതിന് വേണ്ട ടാഗുകൾ ഉപയോഗിക്കുന്നു.
  - വെബ് പേജിനുള്ളിൽ ഫോം എന്ന ആശയം തിരിച്ചറിയുന്നു.
  - ഒരു ഫോം വിവിധ ഭാഗങ്ങളും അവ നിർമ്മിക്കുന്നതിനാവശ്യമായ ടാഗുകളും അവയുടെ ആട്ടിബ്യൂട്ടുകളും പരിചയപ്പെടുന്നു.
  - ടേബിളുകൾ, ഫോറ്മേറ്റിനുള്ളിടെ ഫോം എന്നിവ ഉപയോഗിച്ച് വെബ് പേജുകൾ രൂപകൽപ്പന ചെയ്യുന്നു.

HTML -ലെ അടിസ്ഥാന ടാഗുകൾ കഴിഞ്ഞ അധ്യായത്തിലുടെ നാം പറിച്ചു. അത്തരം ടാഗുകളും അവയുടെ ആട്ടിബ്യൂട്ടുകളും ഉപയോഗിച്ച് ലളിതമായ വെബ്പോജുകൾ രൂപകൽപ്പന ചെയ്യുന്നതും നാം പറിച്ചു. എന്നാൽ കൂടുതൽ സൗകര്യങ്ങളും അവശ്യ സേവനങ്ങളും നൽകുന്ന മറ്റ് ചില വെബ്സൈറ്റുകളും നമുക്ക് പരിചിതമാണോ? വിവിധതരം ലിന്റുകൾ അടങ്കുന്ന വെബ്സൈറ്റുകൾ ഉണ്ട്. എങ്ങനെയാണ് വിവരങ്ങൾ പട്ടികാരുപത്തിൽ രേഖപ്പെടുത്തുന്നത്? വെബ്പോജുകളുടെ പരസ്പര ബന്ധനമാണ് വേർഡ് വേർഡ് വെബ്സൈറ്റുകളും. വിവിധതരം ലിങ്കിങ്ങിനെ (Linking) കുറിച്ച് ഈ അധ്യായത്തിൽ നമ്മൾ ചർച്ച ചെയ്യുന്നുണ്ട്. ചില വിവരങ്ങൾ പട്ടികാരുപത്തിൽ കാണപ്പെടാറുണ്ട്. ചില സമയത്ത് ഒരേ ബേഖസർ വിൻഡോയിൽ ഓൺലൈൻ വെബ്പോജുകൾ നമ്മൾ കാണാറുണ്ട്. അതുപോലെതന്നെ പരീക്ഷയുടെ മാർക്ക് ലിന്റു ലഭിക്കുവാനായി രജിസ്ട്രർ നമ്മൾ നൽകുന്നതിനും സ്‌കൂൾ, കോളേജ് പ്രവേശനം, സ്‌കോളർഷിപ്പ് എന്നിവയ്ക്കുള്ള അപേക്ഷകൾ സമർപ്പിക്കുന്നതിനും, വൈദ്യുതി, വെള്ളം എന്നിവയ്ക്കുള്ള ബിൽ അത്യക്കുന്നതിനും, വെബ്സൈറ്റുകൾ നമുക്ക് പരിചിതങ്ങളാണ്. ഒരേ ബേഖസർ വിൻഡോയിൽ ഓൺലൈൻ വെബ്പോജുകൾ വെബ്സൈറ്റുകൾ നമുക്ക് പരിചിതങ്ങളാണ്. ഉൾക്കൊള്ളിക്കാൻ ആകുമെങ്കിൽ അത് എങ്ങനെയോ? ഉപയോക്താവിൽ നിന്നും യേറ്റ സീക്രിച്ച് കൂടുതൽ വിവരങ്ങൾ നൽകുന്ന വെബ് പേജുകൾ ഉണ്ടാക്കുന്നതെങ്ങനെ? വെബ്പോജുകൾ നിർമ്മാതാക്കൾക്ക് ഈവിധ എല്ലാ സൗകര്യങ്ങളും HTML-ൽ ലഭ്യമാണ്. ഈ ചോദ്യങ്ങൾക്കുള്ള ഉത്തരങ്ങൾ ഈ അധ്യായത്തിൽ

## 5.1 HTML-ലെ ലിസ്റ്റുകൾ (Lists in HTML)

വിവരങ്ങൾ അവതരിപ്പിക്കുന്നേം ലിസ്റ്റുകൾ ആശയവിനിമയ സാധ്യത വർദ്ധിപ്പിക്കുന്നു. വിവിധതരം ലിസ്റ്റുകൾ ഉണ്ട്. അക്കമീട് ലിസ്റ്റുകളും ബുള്ളേറ്റ് ലിസ്റ്റുകളും നമുക്ക് പരിചിതമാണെല്ലാം. ഇതരം ലിസ്റ്റുകൾ വെബ്പോജിൽ ഉൾപ്പെടുത്താൻ HTML സാകര്യം ചെയ്യുന്നു. എല്ലാ ലിസ്റ്റുകളിലും ഒന്നൊ അതിലധികമോ ഘടകങ്ങൾ ഉണ്ടായിരിക്കും. HTML-ലെ ലിസ്റ്റുകളെ ക്രമരഹിത ലിസ്റ്റ്, ക്രമലിസ്റ്റ്, വൈഫനിഷൻ ലിസ്റ്റ് എന്നിങ്ങനെ മുന്നായി തരംതിരിച്ചിരിക്കുന്നു.

### 5.1.1 ക്രമരഹിത ലിസ്റ്റുകൾ (Unordered lists)

ക്രമരഹിത ലിസ്റ്റ് അമൈവാ ബുള്ളേറ്റ് ലിസ്റ്റുകളിൽ ഓരോ ഇനത്തിന്റെയും മുന്നിൽ ഒരു ബുള്ളേറ്റോ അല്ലെങ്കിൽ മറ്റൊരുക്കിലും ശ്രാവിക ചിഹ്നങ്ങളോ പ്രദർശിപ്പിക്കുന്നു. **<UL>**, **</UL>** എന്നീ ടാഗ് ജോധികൾ ഉപയോഗിച്ച് ക്രമരഹിത ലിസ്റ്റുകൾ നിർമ്മിക്കുവാൻ നമുക്ക് സാധിക്കും. ലിസ്റ്റിലെ ഒരോ ഇനവും **<LI>**, **</LI>** ടാഗ് ജോധികൾ ഉപയോഗിച്ച് ചേർക്കപ്പെടുന്നു. ലിസ്റ്റിലെ ഇനങ്ങൾ യാതൊരു ക്രമവും ഇല്ലാതെ പ്രദർശിപ്പിക്കുന്നതിനാണ് ക്രമരഹിത (Unordered) ലിസ്റ്റുകൾ ഉപയോഗിക്കുന്നത്.

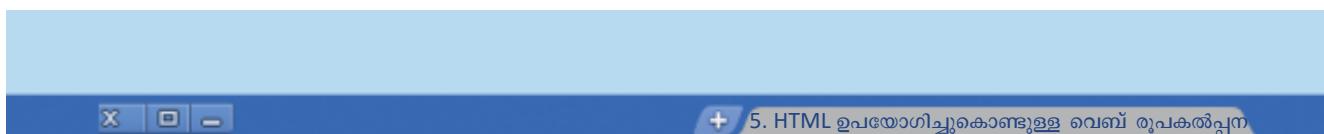
ഉദാഹരണം : 5.1-ൽ ഒരു കമ്പ്യൂട്ടർ ഏതാനും ഹാർഡ്‌വെയർ ഭാഗങ്ങൾ ബുള്ളേറ്റ് ലിസ്റ്റായി പ്രദർശിപ്പിക്കുന്നതിനുള്ള HTML കോഡ് നൽകുന്നു. ഇതിനുസൃതമായ വൈബ്പോജ് ചിത്രം 5.1 തോന്തരം കാണിച്ചിരിക്കുന്നു.

#### ഉദാഹരണം 5.1: ക്രമരഹിത ലിസ്റ്റ് നിർമ്മിക്കാനാവശ്യമായ HTML കോഡ്

```
<HTML>
<HEAD>
    <TITLE> Unordered Lists </TITLE>
</HEAD>
<BODY Bgcolor= "#DEB887">
    <CENTER> <H2> Unordered List </H2> </CENTER>
    While buying a computer, we have to consider many items.
    Here are some important items
    to consider.
    <UL>
        <LI> RAM </LI>
        <LI> Hard Disk </LI>
        <LI> Mother Board </LI>
        <LI> Processor </LI>
    </UL>
</BODY>
</HTML>
```



ചിത്രം 5.1: ക്രമരഹിത ലിസ്റ്റ് ഉപയോഗിച്ച് വൈബ്പ പേജ്



<UL> ടാഗിലെ Type അട്ടിവ്യൂട്ടിന് Disc (തന തുവില), Square, Circle എന്നീ മുന്ന് വിലകൾ നൽകാവുന്നതാണ്. ഇങ്ങനെ ചെയ്യുമ്പോൾ ലിസ്റ്റിലെ ഓരോ ഇനത്തിന്റെയും മുന്നിൽ അതി നന്ദസ്യത്തായ ബുള്ളത്ര ചിത്രകരിക്കുന്നു. താഴെ നൽകിയിട്ടുള്ള HTML കോഡ് ചിത്രം 5.2 തോന്തരം കാണിച്ചിരിക്കുന്ന ലിസ്റ്റ് അടങ്കിയ വെബ്പോജ്ഞ പ്രദർശിപ്പിക്കുന്നു.

```
<UL Type= "Square">
    <LI> RAM </LI>
    <LI> Hard Disk </LI>
    <LI> Mother Board </LI>
    <LI> Processor </LI>
</UL>
```

ചിത്രം 5.2: ബുള്ളത്ര ക്രമരഹിത ലിസ്റ്റ്



<UL> ടാഗിന്റെ Type അട്ടിവ്യൂട്ടിന് None എന്ന വില നൽകുകയാണെങ്കിൽ ക്രമരഹിത ലിസ്റ്റിലെ ഓരോ ഇനത്തിന്റെയും മുന്നിൽ ബുള്ള രൂക്കളൊന്നും പ്രദർശിപ്പിക്കുകയില്ല. മുകളിൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിൽ <UL Type = "None"> എന്ന മാറ്റം വരുത്തി പറിശോധിക്കാം.

## 5.1.2 ക്രമലിസ്റ്റ് (Ordered lists)

ലിസ്റ്റിലെ ഇനങ്ങളുടെ മുന്നിൽ സംഖ്യകളോ അക്ഷരങ്ങളോ ചേർത്ത് ക്രമമായി പ്രദർശിപ്പിക്കുന്നതിന് ക്രമലിസ്റ്റ് ഉപയോഗിക്കുന്നു. <OL>, </OL> എന്നീ ടാഗ്ജോഡികൾ ഉപയോഗിച്ചാണ് ഇത്തരം ലിസ്റ്റ് തയാറാക്കുന്നത്. ക്രമലിസ്റ്റിലെ ഓരോ ഇനവും പ്രദർശിപ്പിക്കുന്നതിന് <LI>, <ILI> എന്നീ ടാഗ് ജോഡികൾ ഉപയോഗിക്കുന്നു. ഇതിനെ സംഖ്യാനുസൃത ലിസ്റ്റ് എന്നും പറയാറുണ്ട്.

ഉദാഹരണം 5.2 തോന്തരം ക്രമലിസ്റ്റ് നിർമ്മിക്കാനാവശ്യമായ കോഡ് കൊടുത്തിരിക്കുന്നു. ഇതിനുസൃതമായ വെബ്പോജ്ഞ ചിത്രം 5.3 തോന്തരം പ്രദർശിപ്പിക്കുന്നു.

### ഉദാഹരണം 5.2: ക്രമലിസ്റ്റ് നിർമ്മിക്കാനാവശ്യമായ HTML കോഡ്

```
<HTML>
<HEAD>
    <TITLE> Ordered Lists </TITLE>
</HEAD>
<BODY Bgcolor= "#DDA0DD">
    <H2 Align= "center"> Ordered List </H2>
    Consider the memory devices of a computer.
    Then according to the speed of data processing,
```

we can arrange the memory devices as follows.

```

<OL>
    <LI> Registers </LI>
    <LI> Cache </LI>
    <LI> RAM </LI>
    <LI> Hard Disk </LI>
</OL>
</BODY>
</HTML>

```

ചിത്രം 5.3 ത്ത് 1 മുതൽ 4 വരെയുള്ള ക്രമനമ്പരി ഉൾക്കൊള്ളുന്ന ലിസ്റ്റ് പ്രദർശിപ്പിക്കുന്നു. നമുക്ക് മറ്റ് പല സംവ്യാരിതികളും ഉപയോഗിച്ച് ലിസ്റ്റിലെ ഇനങ്ങൾ പ്രദർശിപ്പിക്കാം. നമുക്ക് അനുയോജ്യമായ താഴെപ്പറയ്ക്കേണ്ട വിലകൾ **Type** ആട്ടിവ്യൂട്ടിന് നൽകി ലിസ്റ്റിലെ ഇനങ്ങളുടെ സംവ്യാരിതിയിൽ മാറ്റം വരുത്താം.



ചിത്രം 5.3: ക്രമലിസ്റ്റ് അടങ്കിയ വൈവരം

1 തന്ത്രാധികാരിയിൽ (1, 2, 3, ...)

A ആംഗലേയ ഭാഷയിലെ വലിയ അക്ഷരങ്ങൾ (A, B, C, ...)

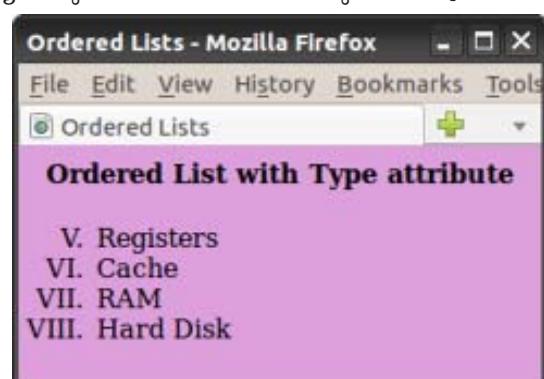
a ആംഗലേയ ഭാഷയിലെ ചെറിയ അക്ഷരങ്ങൾ (a, b, c, ...)

I റോമൻ ലിപിയിലെ വലിയ സംവ്യക്തി (I, II, III, ...)

i റോമൻ ലിപിയിലെ ചെറിയ സംവ്യക്തി (i, ii, iii, ...)

ഒരു ക്രമലിസ്റ്റിലെ ഇനങ്ങളുടെ ക്രമനമ്പരിഞ്ഞ് തുടക്കം സ്ഥാഭാവികമായും ആ സംവ്യാരിതിയിലെ ആദ്യ നമ്പറിലായിരിക്കും. അതായത് ക്രമനമ്പരി തുടങ്ങുന്നത് 1, A, a, I, i എന്നിവയിൽ ഏതെങ്കിലും ഒന്നിലായിരിക്കും. ഇത് നമ്പുതൃക്കുന്ന ഒരു സംവ്യാരിലോ അക്ഷരത്തിലോ ആരംഭിക്കാം. ഇതിനായി സ്റ്റാർട്ട് (Start) ആട്ടിവ്യൂട്ട് ഉപയോഗിക്കാം. ഒരു ക്രമ ലിസ്റ്റിഞ്ഞ് ഇനങ്ങളുടെ ക്രമനമ്പരി 5 ത്ത് തുടങ്ങണമെങ്കിൽ <OL Start= "5"> എന്നതിലൂടെ സാധ്യമാകും. അതിനാൽ 5 ത്ത് തുടങ്ങിയ ക്രമനമ്പരി (6, 7, 8, ...) എന്നിങ്ങനെ തുടരുന്നതാണ്.

എത്ര സംവ്യാരിതികൾ ഉപയോഗിക്കും നോക്കുന്നതു പോലെ സ്റ്റാർട്ട് ആട്ടിവ്യൂട്ടിന് നൽകുന്ന വില പുർണ്ണസംവ്യാരിയിൽ ആയിരിക്കും. ഉദാഹരണത്തിന് താഴെ കൊടുത്തിരിക്കുന്ന ക്രമലിസ്റ്റിഞ്ഞ് തുടക്കനമ്പരി V ആണ് പിന്നീട് VI, VII, VIII, ... എന്നിങ്ങനെ തുടരുന്നു. ഇതിഞ്ഞ് ഒരുപുത്രം ചിത്രം 5.4-ൽ കാണിച്ചിരിക്കുന്നു.



ചിത്രം 5.4: സംവ്യാരിതിയും തുടക്കവിലയും വ്യത്യസ്ഥമായ ക്രമലിസ്റ്റ്

```

<BODY Bgcolor = "#DDA0DD">
<H4 Align="center">Ordered List with Type attribute</H4>
<OL Type= "I" Start= "5">
    <LI> Registers </LI>
    <LI> Cache </LI>
    <LI> RAM </LI>
    <LI> Hard Disk </LI>
</OL>
</BODY>

```



രു ക്രമലിന്നിലെ ഇനങ്ങളുടെ ക്രമനമ്പർ അവരോഹണ ക്രമത്തിൽ പ്രദർശിപ്പിക്കാം. ഇതിനായി <OL> ടാഗിൽ ആട്ടിബ്യൂട്ടായ Reversed ഉപയോഗിക്കാം. ഉദാഹരണം 5.2 ത്ത് നൽകിയിരിക്കുന്ന HTML കോഡിൽ <OL Reversed> എന്ന മാറ്റം വരുത്തി നമുക്ക് പരിശോധിക്കാം. Reversed ഒരു ബുളിയൻ ആട്ടിബ്യൂട്ട് ആയതിനാൽ ഈതിന് വില നൽകേണ്ടതില്ല.

രു ക്രമലിന്നിലെ ഇനങ്ങളുടെ ക്രമനമ്പർ നമുക്ക് അനുയോജ്യമായ വില നൽകി പ്രദർശിപ്പിക്കുവാൻ <CI> ടാഗിൽ ആട്ടിബ്യൂട്ടായ Value ഉപയോഗിക്കാം. മുകളിൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിൽ <LI Value = "10"> RAM </LI> എന്നാണ് നൽകുന്നതെങ്കിൽ, 10 RAM എന്ന തിരിക്കും വെബ്പേജിൽ പ്രദർശിപ്പിക്കുക. ഇതേ രീതിയിൽ <OL> ടാഗിനുള്ളിലെ ഓരോ <LI> ഇനങ്ങൾക്കും Value ആട്ടിബ്യൂട്ട് ഉപയോഗിച്ച് ഓരോരോ വിലകൾ നൽകാം.

### 5.1.3 ഡെഫനിഷൻ ലിസ്റ്റ് (Definition lists)

കുറച്ച് പദങ്ങളും അവയുടെ നിർവ്വചനങ്ങളും നൽകുന്നതിന് ഡെഫനിഷൻ ലിസ്റ്റ് ഉപയോഗിക്കുന്നു. പദങ്ങളിൽ നിന്നും തിരശ്വീനമായി നിശ്ചിത അകലം കഴിഞ്ഞതിന് ശേഷമാണ് നിർവ്വചനങ്ങൾ പ്രദർശിപ്പിക്കുന്നത്. <DL>, </DL> എന്നീ ടാഗ് ജോഡികൾ ഉപയോഗിച്ച് ഡെഫനിഷൻ ലിസ്റ്റ് നിർമ്മിക്കുവാൻ നമുക്ക് സാധ്യമുണ്ട്. ഇവിടെ ബുള്ളട്ടുകളോ സംവ്യക്കളോ ഇനങ്ങൾക്ക് മുൻപിൽ പ്രദർശിപ്പിക്കാറില്ല. <DT> ടാഗ് പദങ്ങൾ നിർമ്മിക്കുന്നതിനും. <DD> ടാഗ് അതിൻ്റെ നിർവ്വചനങ്ങൾ ചേർക്കുന്നതിനും ഉപയോഗിക്കുന്നു.

ഇൻ്റർനെറ്റ് സുരക്ഷയുമായി ബന്ധപ്പെട്ട പദങ്ങളും അവയുടെ നിർവ്വചനങ്ങളും പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ് പേജിൽ HTML കോഡ് ഉദാഹരണം 5.3-ൽ നൽകിയിരിക്കുന്നു. ഇതിൻ്റെ ഒരു പൃഷ്ഠ ചിത്രം 5.5-ൽ കാണാം.

#### ഉദാഹരണം 5.3: ഡെഫനിഷൻ ലിസ്റ്റ് നിർമ്മാണം

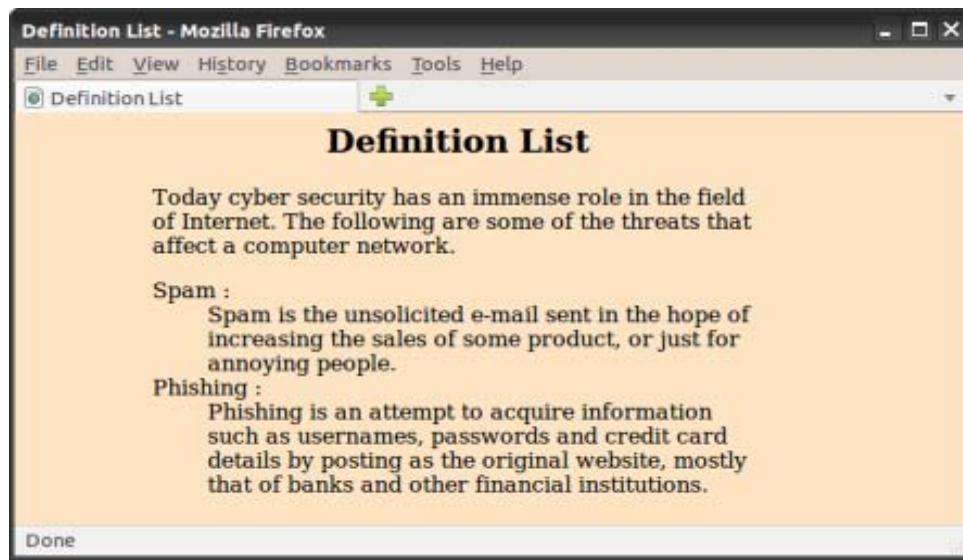
```

<HTML>
<HEAD> <TITLE> Definition List </TITLE> </HEAD>
<BODY Bgcolor= "#FFE4C4" Leftmargin= "100" Rightmargin= "150">

```



```
<H2 Align= "center"> Definition List </H2>
Today cyber security has an immense role in the
field of Internet. The following are some of
the threats that affect a computer network.
<DL>
<DT>Spam :</DT>
<DD> Spam is the unsolicited e-mail sent in the
hope of increasing the sales of some product, or
just for annoying people.</DD>
<DT>Phishing :</DT>
<DD> Phishing is an attempt to acquire information
such as usernames, passwords and credit card details
by posting as the original website, mostly that
of banks and other financial institutions. </DD>
</DL>
</BODY>
</HTML>
```

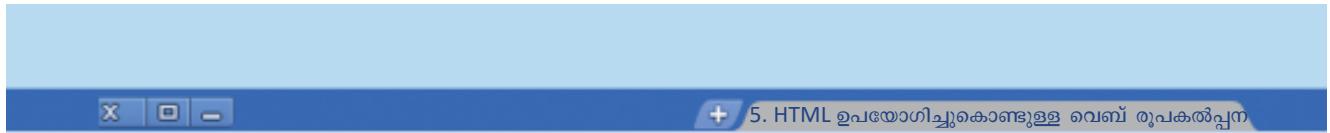


ചിത്രം 5.5: ശ്യാമലിന്റെ ലിറ്റ് ഉൾക്കൊള്ളുന്ന വെബ്പേജ്

ചിത്രം 5.5-ലെ നിർവ്വചനങ്ങളുടെ ഇൻഡ്രോഷൻ, ഇടതും വലതും ഭാഗങ്ങളിലുള്ള മാർജ്ജി നുകളും ശ്രദ്ധിക്കുക.

#### 5.1.4 നെസ്റ്റഡ് ലിസ്റ്റ് (Nested lists)

ങ്ങു ലിസ്റ്റിലെ ഏതെങ്കിലും ഇനത്തിന്റെ താഴെ മറ്റാരു ലിസ്റ്റ് നൽകുവാനാകും. ഇത്തരം ലിസ്റ്റുകളെ നെസ്റ്റഡ് ലിസ്റ്റ് എന്നിയപ്പെടുന്നു. ഈ വിവിധ തരത്തിൽ തയാറാക്കാം.



ഉദാഹരണമായി ഒരു ക്രമലിസ്റ്റിനുള്ളിൽ മറ്റാരു ക്രമലിസ്റ്റോ, ക്രമരഹിതലിസ്റ്റോ നൽകുവാനാകും. അതുപോലെ ഒരു ക്രമരഹിത ലിസ്റ്റിനുള്ളിൽ മറ്റാരു ക്രമലിസ്റ്റോ ക്രമരഹിതലിസ്റ്റോ നൽകാനുമാകും. ഉദാഹരണം 5.4 തുടർന്ന് നൽകുന്ന HTML കോഡ് നേരും ലിസ്റ്റിന്റെ ആഴയം വിശദമാക്കുന്നു.

#### ഉദാഹരണം 5.4: നേരും ലിസ്റ്റ് നിർമ്മാണം

```
<HTML>
<HEAD>
    <TITLE> Nested Lists </TITLE>
</HEAD>
<BODY Bgcolor= "#E0FFFF">
    <H2 Align= "center"> Nested List </H2>
    Consider the devices of a computer.
    We can list some of them as follows.
    <OL>
        <LI> Input Devices </LI>
        <UL>
            <LI>Keyboard</LI>
            <LI>Mouse</LI>
            <LI>Scanner</LI>
            <LI>MICR</LI>
        </UL>
        <LI> Output Devices </LI>
        <UL Type= "Square">
            <LI>Printers</LI>
            <LI>Monitors</LI>
            <LI>Speakers</LI>
        </UL>
        <LI> Memory Devices </LI>
        <UL Type= "Circle">
            <LI>Hard Disc</LI>
            <LI>CD Rom</LI>
            <LI>Flash Drive</LI>
        </UL>
    </OL>
</BODY>
</HTML>
```

ഉദാഹരണം 5.4 തുടർന്ന് മുന്ത് ക്രമരഹിത ലിസ്റ്റുകൾ ഒരു ക്രമലിസ്റ്റിലേക്ക് നേരും ചെയ്തിരിക്കുന്നു.



ചിത്രം 5.6: നേരും ലിസ്റ്റ് ഉപയോഗിച്ച് ഉപകരണ അളുവാൻ ത്രാംസിലീവ് പ്രോഫീലിക്കുന്നു

### നിങ്ങളുടെ പുരോഗതി അറിയിക്കുക



1. HTML ലെ വിവിധതരം ലിസ്റ്റുകൾ എത്രതാക്കേയാണ്?
2. നിങ്ങളുടെ ക്ലാസിലെ കൂട്ടികളുടെ ലിസ്റ്റ് തയാറാക്കുവാൻ അധ്യാപകൻ ആവശ്യപ്പെട്ടു എന്നു കരുതുക. ഏതുതരം ലിസ്റ്റായിരിക്കും നിങ്ങൾ തിരഞ്ഞെടുക്കുന്നത്? എന്തുകൊണ്ട്?
3. <UL>, <OL> എന്നീ ടാഗുകളുടെ പൊതുവായ ആട്ടിബൃട്ടുകൾ എത്രതാക്കേയാണ്?
4. <UL>, <OL> എന്നീ ടാഗുകൾ തമിലുള്ള വ്യത്യാസങ്ങൾ എഴുതുക.
5. ഡൈഫോണിഷൻ ലിസ്റ്റിൽ ഉപയോഗിക്കുന്ന ടാഗുകളുടെ പേര് പറയുക.

## 5.2 ലിങ്കുകളുടെ നിർമ്മാണം (Creating links)

ഒരു വെബ്ബ് പേജിൽ നിന്ന് മറ്റാരു വെബ്ബ് പേജിലേക്കോ അതേ വെബ്ബ്‌പേജിലെ മറ്റാരു ഭാഗത്തേക്കോ ബന്ധിപ്പിക്കുന്നതിന് സഹായിക്കുന്ന ഘടകമാണ് ഹൈപ്പർലിങ്ക്. ഹൈപ്പർലിങ്കിനെ സാധാരണയായി ലിങ്കുകൾ എന്നാണ് വിളിക്കുന്നത്. HTML-ൽ <A> ടാഗിനും </A> ടാഗിനും ഇടയിൽ എത്ത് നൽകിയാലും അത് ലിങ്കിന്റെ ഭാഗമാകും. ഉപയോക്താവിന് അവിടെ ക്ലിക്ക് ചെയ്തുകൊണ്ട് ലിങ്ക് ചെയ്തിരിക്കുന്ന ഡോക്യുമെന്റിലേക്ക് എത്താനാകും. <A> ടാഗിന്റെ മുഖ്യ ആട്ടിബൃട്ടാണ് ഹൈപ്പർ റഫറൻസ് എന്ന് അറിയപ്പെടുന്ന **Href**. ഇതിന്റെ വിലയായി സാധാരണ URL ആണ് നൽകുന്നത് (ഹൈപ്പർലിങ്ക് ബന്ധിപ്പിക്കേണ്ട വെബ്ബ് പേജിന്റെ വിലാസം).

ഉദാഹരണത്തിന് താഴെകാടുത്തിരിക്കുന്ന കോഡ് ഭാഗം ശ്രദ്ധിക്കുക.

```
<A Href= "http://www.dhsekerala.gov.in">Higher Secondary</A>
ഈത് “http:// www.dhsekerala.gov.in.” എന്ന വെബ്ബ്‌സൈറ്റിനെ ലക്ഷ്യം വയ്ക്കുന്ന ഹൈപ്പർലിങ്ക് സൃഷ്ടിക്കുന്നു. ഉപയോക്താവ് ഈ ലിങ്കിൽ ക്ലിക്ക് ചെയ്യുന്നോൾ ഈ URL-റ്റെ ഹോംപേജ് ബോർഡിൽ ലഭ്യമാകും. <A>, </A> എന്നിവയ്ക്കിടയിൽ നൽകുന്ന വാക്കുങ്ങൾ വ്യത്യസ്ത നിരത്തിലും അടിവരയോടുകൂടിയും കാണപ്പെടും.
```

ഹൈപ്പർലിങ്ക് സൃഷ്ടിക്കുന്ന HTML കോഡ് ഉദാഹരണം 5.5ലും അതിന്റെ ചിത്രം 5.7ലും കാണിച്ചിരിക്കുന്നു.

### ഉദാഹരണം 5.5: ഒരു വെബ്ബ്‌പേജിൽ ഹൈപ്പർലിങ്കിന്റെ നിർമ്മാണം

```
<HTML>
<HEAD>
    <TITLE> Anchor Tag </TITLE>
</HEAD>
<BODY Bgcolor= "#FFFFFF">
    <H2 Align= "center"> Hyperlinks </H2>
    <P>Now this will create a hyperlink to the website of
        Higher Secondary Department.<BR>
```



Kindly click on the words  
<A Href= "http://www.dhsekerala.gov.in">Higher Secondary Education</A>.  
</BODY>  
</HTML>

ചിത്രം 5.7 തേ കാണിച്ചിരിക്കുന്നതു പോലെ ഫൈലിൽ ലിങ്ക് ചെയ്ത വാക്കും (Higher Secondary Education) അടിയിൽ വരയിട്ട് രീതിയിലും വ്യത്യസ്തമായ നിരത്തിലും ആണ്.

ഫൈലിൽ ലിങ്കുകളുടെ ലക്ഷ്യസ്ഥാനത്തിനുസരിച്ച് അവയെ ഇൻഡ്രോണൽ അല്ലെങ്കിൽ എക്സ്റ്റ്രോണൽ എന്ന് തരംതിരിച്ചിരിക്കുന്നു.



ചിത്രം 5.7: ഫൈലിൽ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വൈബ് പേജ്

### 5.2.1 ഇൻഡ്രോണൽ ലിങ്കിംഗ് (Internal linking)

ഒരു വൈബ് പേജിനെത്തന്നെ മറ്റാരു ഭാഗത്തെക്ക് ബന്ധിപ്പിക്കുന്നത് ഇൻഡ്രോണൽ ലിങ്കിംഗ് എന്നറിയപ്പെടുന്നു. ഇതിനായി Name ആട്ടിബ്യൂട്ട് ഉപയോഗിച്ച് ലിങ്ക് ചെയ്യേണ്ട ഭാഗം തിരിച്ചറിയുവാൻ പ്രത്യേകം നാമകരണം ചെയ്യേണ്ടതുണ്ട്. ഈ പേര് <A> ടാഗിൽ ആട്ടിബ്യൂട്ടായ Href-ആണ് വിലയായി നൽകി ലിങ്കിംഗ് സാധ്യമാകും.

ഉദാഹരണത്തിന് “Environment Pollution” എന്ന വിഷയത്തെ സംബന്ധിക്കുന്ന ഒരു വൈബ്‌പേജ് നിർമ്മിക്കണം എന്ന് കരുതുക. ഇതിൽ മുന്ന് വാന്നികകളിലായി “Indroduction”, “Air Pollution”, “Water Pollution” എന്നിവ നൽകിയിട്ടുണ്ട്. ഈ വാന്നികകളെ സൂചിപ്പിക്കുന്നതിനായി <A> ടാഗിൽ നേരിട്ട് ആട്ടിബ്യൂട്ടായി Indroduction, Air, Water എന്നീ പേരുകൾ നൽകിയിരിക്കുന്നു.

```
<A Name= "Introduction"> INTRODUCTION </A>
<A Name= "Air"> Air Pollution </A>
<A Name= "Water"> Water Pollution </A>
```

വൈബ് പേജിൽ ഈ ഭാഗങ്ങളിലേക്ക് ബന്ധിപ്പിക്കുന്നതിനായി <A> ടാഗിൽ Href ആട്ടിബ്യൂട്ടിന് #Introduction, #Air, #Water എന്നീ വിലകൾ നൽകണം. ഇതിനുള്ള HTML കോഡ് ഭാഗം താഴെക്കൊടുക്കുന്നു.

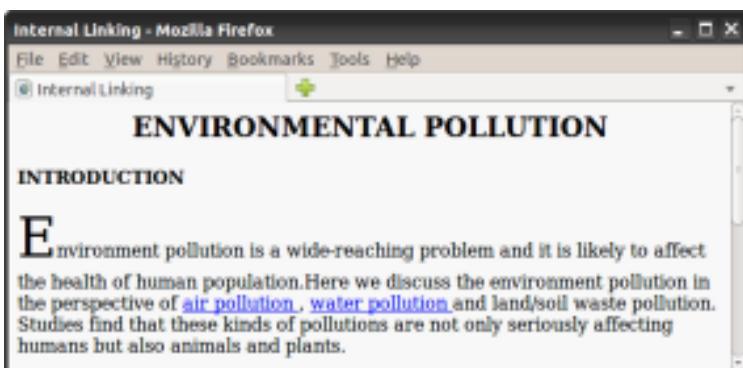
```
<A Href = "#Introduction"> Go to Introduction </A>
<A Href = "#Air"> Air pollution </A>
```

ഇൻഡ്രോണൽ ലിങ്കിംഗ് ആശയം ഉൾപ്പെടുത്തുന്ന വൈബ് പേജ് നമുക്ക് നിർമ്മിക്കാം. ഇതിനുള്ള HTML കോഡ് ഉദാഹരണം 5.6 ലും അതിനുസ്യതമായ വൈബ് പേജ് ചിത്രം 5.8 ലും നൽകിയിരിക്കുന്നു.

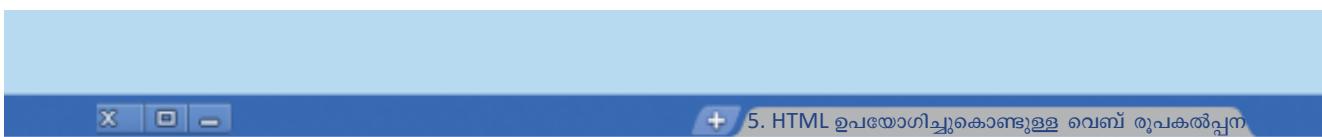
### ഉദാഹരണം 5.6: ഇൻഡിന്റൽ ലിങ്ക് ഉൾപ്പെടുത്തുന്ന വൈബ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE> Internal Linking </TITLE> </HEAD>
<BODY Bgcolor= "f8f8f8">
    <H2 Align= "center">ENVIRONMENTAL POLLUTION</H2>
    <A Name= "Introduction"><B>INTRODUCTION</B></A>
    <P><FONT Size= "15">E</FONT>nvironment pollution is a wide-reaching problem and it is likely to affect the health of human population. Here we discuss the environment pollution in the perspective of <A HREF= "#Air">air pollution </A>,
    <A Href= "#Water"> water pollution </A>and land/soil waste pollution. Studies find that these kinds of pollutions are not only seriously affecting humans but also animals and plants.
    </P>
    <A Name= "Air"><B> Air Pollution</B></A>
    <P>The air we breathe is an essential ingredient for our health and wellbeing. Unfortunately polluted air is common throughout the world, especially in developed countries.
    </P>
    <A Name= "Water"><B> Water pollution</B></A>
    <P>The water we drink is an essential ingredient for our health and wellbeing. Unfortunately polluted water and air are common throughout the world. Water pollution is caused by the discharge of industrial effluents, sewage water and agricultural or household waste.
    </P>
    <A Href= "#Introduction">Go to Introduction </A>
</BODY>
</HTML>
```

ചിത്രം 5.8 ത്ത് ഈ വൈബ് പേജിന്റെ ഒരു ഭാഗം കാണിച്ചിരിക്കുന്നു.  
ഉദാഹരണം 5.8 ത്ത് കാണിച്ചിരിക്കുന്ന air pollution, water pollution എന്നീ ലിങ്കുകളിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ



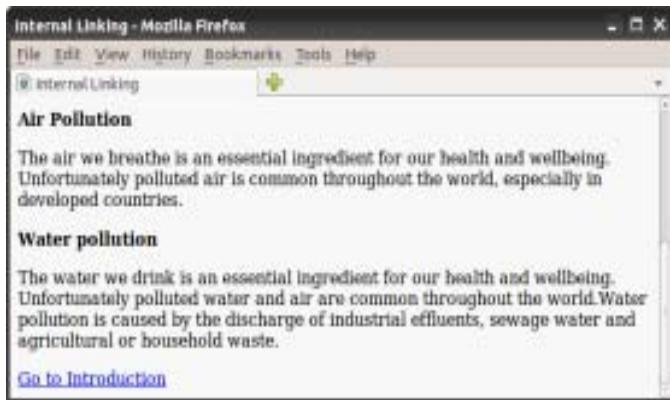
ചിത്രം 5.8: ഇൻഡിന്റൽ ലിങ്കുകൾ ഉൾപ്പെടുത്തുന്ന വൈബ് പേജ്



ചിത്രം 5.9 ത് കാണിച്ചിരിക്കുന്ന വൈബ്‌പേജിന്റെ അനുയോജ്യമായ ഭാഗം ദ്വാരാക്കും. ഇതുപോലെ ചിത്രം 5.9 ലെ വൈബ്‌പേജിന്റെ താഴെ ഭാഗത്തുള്ള [Go to Introduction](#) എന്ന ലിങ്കിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ Introduction എന്ന ഭാഗം ദ്വാരാക്കും.

### 5.2.2 ഏകസ്റ്ററേണ്ട ലിങ്കിംഗ് (External linking)

ഒരു വൈബ് പേജിൽ നിന്ന് മറ്റൊരു വൈബ് പേജിലേക്ക് ലിങ്ക് ചെയ്യുന്ന പ്രവർത്തനത്തെ എക്സ്റ്ററേണ്ട ലിങ്കിംഗ് എന്ന് പറയുന്നു. നിലവിൽ ലഭ്യമായ വൈബ്‌പേജിന്റെ <A> ടാഗിന്റെ Href അട്ടിബ്യൂട്ടിന്റെ വിലയായി എക്സ്റ്ററേണ്ട ഫയലിന്റെ URL നൽകി ഇതുസാധ്യമാക്കാം. ഇതിന്റെ പ്രവർത്തനങ്ങൾ നാം 5.2-ൽ പരിച്ചെച്ചത്താണ്.



ചിത്രം 5.9: വൈബ്‌പേജിന്റെ ശൈപ്പർലിങ്ക് ചെയ്തപ്പെട്ട ഭാഗങ്ങൾ

### 5.2.3 URL എന്ന ആശയം (Concept of URL)

URL-ന്റെ പൂർണ്ണരൂപം യൂണിഫോം റിസോഴ്സ് ലോക്കേറ്റർ എന്നാണ്. വൈബ് അധികം എന്നാണിതിന്റെ അർമ്മം. റിലേറ്റീവ്, അബ്സല്യൂട്ട് എന്നീ രണ്ടു തരം URL കൾ നിലവിലുണ്ട്.

<A Href= "http://www.scertkerala.gov.in"> എന്നത് ഒരു അബ്സല്യൂട്ട് URL ആണ്. കാരണം ഈവിടെ വൈബ് അധികം പൂർണ്ണമായും നൽകിയിരിക്കുന്നു. എന്നാൽ ലിങ്ക് <A Href = "image.html"> എന്നാണ് എഴുതിയതെങ്കിൽ അത് റിലേറ്റീവ് ലിങ്ക് ആയി പരിഗണിക്കും. ഈവിടെ നമ്മൾ നൽകിയ “image.html” ഒരു റിലേറ്റീവ് URL ആണ്. എന്തുകൊണ്ടും image.html ഉം വൈബ്‌പേജും ഒരേ ഡയറക്ടറിൽ ശേഖരിച്ചിരിക്കുന്നു. നിലവിൽ ലഭ്യമായ വൈബ്‌പേജ് /home/kite/HTML/hyperlink.html എന്നാണ് ശേഖരിച്ചിരിക്കുന്നതെങ്കിൽ <A Href = "image.html"> എന്ന ശൈപ്പർ ലിങ്ക് സൂചിപ്പിക്കുന്നത് /home/kite/HTML എന്ന ഫോർഡിലെ image.html എന്ന ഫയലിനെ ആയിരിക്കും.

### 5.2.4 ഗ്രാഫിക്കൽ ശൈപ്പർ ലിങ്ക് നിർമ്മാണം (Creating graphical hyperlinks)

ഒക്ടറ്റുകളെ ശൈപ്പർ ലിങ്കായി ഉപയോഗിക്കുന്ന രീതി നാം മനസിലാക്കിക്കഴിഞ്ഞു. ഇതുപോലെ ഇമേജുകളും ശൈപ്പർ ലിങ്കാക്കി ഉപയോഗിക്കാം. ഇതിനായി <A>, </A> എന്നീ ടാഗുകളുടെ ഇടയിൽ <IMG> ടാഗ് നൽകിയാൽ മതിയാകും. ഇതിന്റെ HTML കോഡ് ഉദാഹരണം 5.7-ൽ കാണിച്ചിരിക്കുന്നു. ഇതു സംബന്ധിച്ച വിവരങ്ങൾ ചിത്രം 5.10 ലും നൽകിയിരിക്കുന്നു.



#### ഉദാഹരണം 5.7: ശ്രാഫിക്കൽ ഫോപ്പർ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ്സൈറ്റ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE> Graphical Hyperlink </TITLE> </HEAD>
<BODY Bgcolor = "#E0FFFF">
    <H2 Align= "center">Graphical Hyperlink</H2>
    Here is the image with <I>Graphical hyperlink </I>
    <A Href= "https://www.wikipedia.org">
        <IMG Src= "wiki.jpg" Alt= "Image of Wiki"
            Height= "30" Width= "40" Border= "1"> </A>.
        We can click over this image and the home page of
        linked site, wikipedia.org will open.
    </BODY>
</HTML>
```

നിങ്ങൾ മൗസ് പോയിസ്തേർ വിക്കൈപീഡിയിൽ ലോഗോയുടെ മുകളിൽ കൊണ്ടുവരുന്നോൾ അൽ ഹാൻ്റ് സിനിലായി മാറുന്നു. ഈ സൂചിപ്പിക്കുന്നത് ഈ ലോഗോ ഒരു ശ്രാഫിക്കൽ ഫോപ്പർ ലിങ്ക് ആണെന്നാണ്. അവിടെ ഓംക്രീക്ക് ചെയ്യുന്നോൾ പ്രേരണ റിൽ [www.wikipedia.org](https://www.wikipedia.org) എന്ന വെബ്സൈറ്റ് ലഭ്യമാകും.

#### 5.2.5 ഇ-മെയിൽ ലിങ്ക് നിർമ്മിക്കുന്ന വിധം (Creating e-mail linking)

ഫോപ്പർലിങ്ക് പ്രോട്ടോക്കോളായ [mailto](mailto:) ഉപയോഗിച്ച് നമുക്ക് ഇ-മെയിൽ ഫോപ്പർലിങ്ക് നിർമ്മിക്കുവാനാകും. ഈ വിശദമാക്കുന്ന HTML കോഡ് ഉദാഹരണം 5.8 ലും വെബ്സൈറ്റിൽ ചിത്രം 5.11 ലും നൽകിയിട്ടുണ്ട്.

#### ഉദാഹരണം 5.8: ഇ-മെയിൽ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ്സൈറ്റ് നിർമ്മാണം



ചിത്രം 5.10: ശ്രാഫിക്കൽ ഫോപ്പർ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ്സൈറ്റ്

```
<HTML>
<HEAD> <TITLE> e-mail Linking </TITLE> </HEAD >
<BODY Bgcolor= "#E0FFFF">
    <H2 Align= "center">e-mail linking</H2>
    Now we can create an <B><I>e-mail hyperlink </I></B> to
    SCERT in the following way. Kindly click on the word
    <A Href= mailto: "scertkerala@gmail.com"> SCERT</A> Kerala.
</BODY>
</HTML>
```

ചിത്രം 5.11 കാണിച്ചിരിക്കുന്ന വെബ്പോജിൽ [SCERT](mailto:scertkerala@gmail.com) എന്ന ലിങ്ക് നൽകിയിട്ടുണ്ട്. ഈതിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ scertkerala@gmail.com എന്ന് അധിസന്ദർഭം ചെയ്യപ്പെട്ട ഒഴിവു മെന്റേ ജ്ഞാനോക്സോടുകൂടിയ ഇ-മെയിൽ പ്രോഗ്രാം തുറക്കുന്നു.



ചിത്രം 5.11: ഇ-മെയിൽ ഫോസ്റ്റർ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ്പോജ്



ഒരു വെബ്പോജിൽ നിന്ന് ഇ-മെയിൽ ലിങ്ക് നൽകി ഇ-മെയിൽ പ്രോഗ്രാം തുറക്കുന്നുമ്പോൾ, ഇ-മെയിൽ പ്രോഗ്രാമിലെ എല്ലാ ഭാഗങ്ങൾക്കും തന്ത്രം വിലകൾ നൽകാവുന്നതാണ്. ഉദാഹരണമായി "Thank you sir" എന്ന സന്ദേശം "Director SCERT" എന്ന സബ്ജക്ടോടുകൂടി scertkerala@gmail.com എന്ന വിലാസത്തിലേക്കും കാർബൺ കോപ്പി യായി (CC) scertkerala1@gmail.com എന്ന വിലാസത്തിലേക്കും bcc ആയി scertkerala2@gmail.com എന്ന വിലാസത്തിലേക്കും ഒരു ഇ-മെയിൽ ലിങ്ക് വഴി അയക്കണമെങ്കിൽ മുകളിൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിലെ <A> ടാഗ് താഴെ പറയുന്ന വിധം മാറ്റി എഴുതിയാൽ മതിയാകും.

```

<A Href="mailto:scertkerala@gmail.com?
cc=scertkerala1@gmail.com&
bcc=scertkerala2@gmail.com
&subject=Director%20SCERT
&body=Thank%20You%20Sir">
send E-mail with cc bcc subject and body </A>
<A> ടാഗിലെ ? ചിലനം ആദ്യത്തെ വിഭാഗത്തൈയും & മറ്റ് വിഭാഗങ്ങളൈയും
വേർത്തിക്കാൻ ഉപയോഗിക്കുന്നു.

```

### 5.3 മ്യൂസിക്, വീഡിയോ ഉൾപ്പെടുത്തൽ (Inserting music and video)

ഇപ്പോൾ ലഭ്യമാകുന്ന മിക്കവാറും വെബ് പേജുകളിലും ടെക്നോളജികൾ, ചിത്രങ്ങൾ എന്നിവ കൂടാതെ മ്യൂസിക്, വീഡിയോ എന്നിവ ഉൾപ്പെടുത്തി സംസ്ക്രാഫ്റ്റോക്സിയിരിക്കുന്നു. എങ്കിൽ നന്ദിയാണ് ഒരു വെബ് പേജിൽ മ്യൂസിക്, വീഡിയോ തുടങ്ങിയവ ഉൾപ്പെടുത്തുന്നത്? മർട്ടിമീഡിയിൽ വിഭവങ്ങൾ ഉൾപ്പെടുത്താൻ ഇൻലൈൻ, എക്സ്റ്റോണൽ എന്നീ രണ്ടുതരം റീതികൾ ഉപയോഗിക്കുന്നു. ഇൻലൈൻ റീതിയിൽ ഫയലുകളും അതിനാവശ്യമായ വിവരങ്ങളും വെബ്പോജിൽ ഭാഗമായാണ് കൈകാര്യം ചെയ്യുന്നത്. ബഹുസർവ്വ വിസ്തേജി തുറക്കുന്നുമ്പോൾ മ്യൂസിക്കുകളും വീഡിയോകളും ഉപയോക്താവിന് കേൾക്കുവാനും കാണുവാനും സാധിക്കും. എന്നാൽ വെബ് പേജിൽ ലഭ്യമാകുന്ന മർട്ടിമീഡിയിൽ വിഭവങ്ങൾ ആ കമ്പ്യൂട്ടറിൽ ലഭ്യമായ സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് തുറക്കുന്ന

രീതിയാണ് എക്സ്പ്രസ്സ് നാല്. നമുക്ക് .jpg, .gif, .avi, .png, .tiff, .mp3, .mp4, തുടങ്ങിയ എക്സ്പ്രസ്സ് നാല് കാണപ്പെടുന്ന മൾട്ടിമീഡിയിയ ഫയലുകൾ ഒരു വെബ്പോജിലേക്ക് ലിങ്ക് ചെയ്യുവാനാകും.

മൾട്ടിമീഡിയിയ വിഭവങ്ങളായ മ്യൂസിക്കുകളും വീഡിയോകളും ഉൾപ്പെടുത്താൻ HTML ടാഗായ <EMBED> എന്ന ടാഗാണ് ഉപയോഗിക്കുന്നത്. വെബ്പോജിൽ ഈ ടാഗ് ഉൾപ്പെടുത്തുന്നേം മൾട്ടിമീഡിയ മാർഗ്ഗങ്ങളുടെ നിയന്ത്രണത്തിനുവേണ്ട ഉപയോകളും ലഭ്യമാകുന്നു. എന്നാൽ ഈ ടാഗിന്റെ പ്രവർത്തനം പിന്തുണയ്ക്കാത്ത ഫൈസറുകളിൽ <NOEMBED> എന്ന ടാഗ് ഉപയോഗിക്കാം.

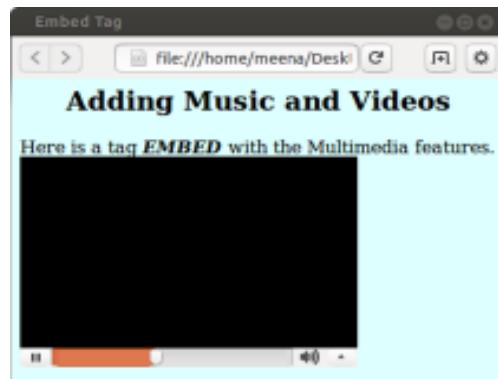
<EMBED> ടാഗിന്റെ പ്രധാന ആട്ടിബ്യൂട്ട് **Src** ആണ്. ഈ ടാഗിൽ വിലയായി വെബ്പോജിൽ ഉൾപ്പെടുത്തേണ്ട മ്യൂസിക് വീഡിയോ എന്നിവയുടെ URL നൽകണം. മറ്റ് ആട്ടിബ്യൂട്ടുകൾ **Height**, **Width**, **Align**, **Alt**, തുടങ്ങിയവ ആണ്. ഈ ടാഗിൽ വിലകൾ നമുക്ക് പരിചിതമാണോ? ഈ ടാഗിൽ ആട്ടിബ്യൂട്ടും മ്യൂസിക്കുകളോ, വീഡിയോകളോ ഉപയോകതാവിന് കാണണമെങ്കിൽ **Hidden** ആട്ടിബ്യൂട്ടിന് തന്ത വിലയായി False എന്നും കാണണം എങ്കിൽ True എന്ന വിലയും നൽകണം.

രു വെബ്പോജിൽ രു ഓഡിയോ ഉൾപ്പെടുത്തിയതെന്നെന്നെന്നും 5.9 ലും വെബ്പോജിൽ ചിത്രം 5.12 ലും നൽകുന്നു.

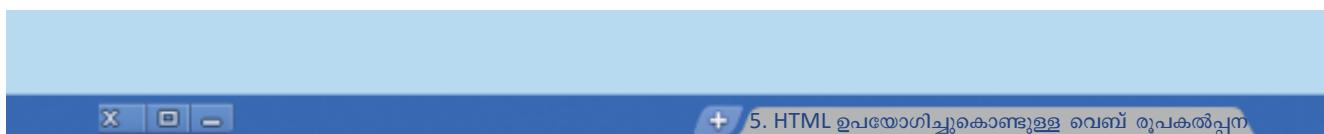
#### ഉദാഹരണം 5.9: ഓഡിയോ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ്പോജ് നിർമ്മാണം

```
<HTML>
<HEAD>
    <TITLE> Embed Tag </TITLE>
</HEAD>
<BODY Bgcolor = "#DDFFFF">
    <H2 Align= "center"> Adding Music and Videos </H2>
    Here is a tag <B><I>EMBED </I></B> with the Multimedia
    features.<BR>
    <EMBED Src= "song1.mp3" Width= "300" Height= "60">
    </EMBED>
</BODY>
</HTML>
```

ഉദാഹരണം 5.9 തേ കൊടുത്തിരിക്കുന്ന HTML പേജ് ഫൈലിൽ വഴി തുറക്കുന്നേം അതിൽ ഉള്ളടക്കം ചെയ്യുട്ട് മ്യൂസിക്കുകൾ കേൾക്കാൻ സാധിക്കുന്നു. ഈ ടാഗിൽ നിയന്ത്രണങ്ങളായ ഷൈറ്റ്, പോസ്റ്റ്, ശബ്ദം എന്നിവയും വെബ്പോജിൽ ലഭ്യമാകും. ഈ ടാഗിൽ വീഡിയോ വെബ്പോജിൽ ലഭ്യമാക്കുന്നതിന് <EMBED> ടാഗും അതിന്റെ ആട്ടിബ്യൂട്ടായി **Src** യും ഉപയോഗിക്കാം.



ചിത്രം 5.12: ഓഡിയോ ലിങ്ക് ഉൾക്കൊള്ളുന്ന വെബ്പോജ്



ഉദാഹരണം 5.10 ത്ത് കൊടുത്തിരിക്കുന്ന HTML കോഡിന് അനുസൃതമായ വെബ്പോജ്ഞ് (ചിത്രം 5.13) ഈ ലിങ്കിൽ ചിത്രീകരിക്കുന്നു.

**ഉദാഹരണം 5.10:** ഒരു വെബ് പേജിൽ വീഡിയോ ലഭ്യമാക്കുന്ന HTML കോഡ്

```
<HTML>
<HEAD>
    <TITLE> Embed Tag </TITLE>
</HEAD >
<BODY Bgcolor = "#DDFFFF">
    <H2 Align="center">Adding Music and Videos</H2>
    Here is a tag <B><I>EMBED </I></B>with the Multimedia
    features.
    <EMBED Src= "alan.mp4" Width= "300" Height= "150">
    </EMBED>
    <NOEMBED><IMG Src= "book2.jpg"
        Alt= "Alternative Media">
    </NOEMBED>
</BODY>
</HTML>
```

നമ്മൾ ഉദാഹരണം 5.9 ത്ത് നൽകിയിരിക്കുന്ന HTML പേജ് തുറക്കുമ്പോൾ ഉള്ളടക്കം ചെയ്ത വീഡിയോ ഫേൾ ചെയ്ത് തുടങ്ങും. ഇവിടെയും ശബ്ദം, വീഡിയോ എന്നിവ നിയന്ത്രിക്കാനാവശ്യമായ വോളിയം, പോസ്, ഫൂൾ സ്ക്രൈൻ മോഡ് എന്നീ കൺട്രോളുകൾ ലഭ്യമാകും. മറ്റ് വെബ് പേജിലുള്ള ഓഡിയോയും വീഡിയോയും നമ്മുടെ വെബ് പേജിലേക്ക് ലിങ്ക് ചെയ്ത് ഉൾപ്പെടുത്താം.

വെബ് പേജിന്റെ പശ്ചാത്തലത്തിൽ ഓഡിയോ ഇൻലെവൻ ആയി മുൻസിക് ഫേൾ ചെയ്യുന്നതിന് വേണ്ടി <BGSOUND> എന്ന ടാഗ് ഉപയോഗിക്കുന്നു. ഇതിന്റെ ഉദാഹരണം 5.11 ത്ത് കൊടുത്തിരിക്കുന്നു.

**ഉദാഹരണം 5.11: <BGSOUND> ടാഗിന്റെ ഉപയോഗം വ്യക്തമാക്കുന്ന HTML കോഡ്**

```
<HTML>
<HEAD> <TITLE> Background Music </TITLE> </HEAD >
<BODY Bgcolor = "#DDFFFF">
    <H2 Align= "center">Adding Background Music </H2>
    Here is a tag <B><I> BGSOUND </I></B> which helps
    us to play background music in our web page.
    <BGSOUND Src= "Song2.mp3" Loop= "Infinite">
</BODY>
</HTML>
```



ചിത്രം 5.13: വീഡിയോ ഉൾക്കൊള്ളുന്ന വെബ്പോജ്ഞ്

മുൻസിപിലെ ശബ്ദത്തിനെതിരെ നമുക്ക് <BG SOUND> ടാഗിൽ Volume ആട്ടി ബ്യൂട്ട് ഉപയോഗിക്കാം. HTML പേജിൽ എക്സൈസിൽ ഫയൽ ഉൾപ്പെടുത്തുന്നതിനായി Src, Href ആട്ടിബ്യൂട്ടുകളിൽ അവയുടെ കൃത്യമായ വഴി നൽകിയിരിക്കും. ഉദാഹരണം 5.10-ൽ ഈ ആട്ടിബ്യൂട്ടുകളുടെ വിലകളിൽ വരുത്തി പരീക്ഷിച്ചു നോക്കാവുന്നതാണ്. ലുപ് ആട്ടിബ്യൂട്ട് ഉപയോഗിച്ച് മുൻസിപ് എത്ര തവണ അവർത്തിക്കുന്നു എന്നു ഒള്ളത് തീരുമാനിക്കാം. ഇതിന്റെ വില ആയി Infinite നൽകിയാൽ വെബ്പോജ്ഞി ബൈറ്റസിൽ ലഭ്യമാകുന്നിടത്തോളം തുടർച്ചയായി മുൻസിപ് പ്ലേ ചെയ്യപ്പെടുന്നതാണ്.

## 5.4 വെബ് പേജിൽ ടേബിൾ ഉൾപ്പെടുത്തുന്ന വിധം (Creating tables in a web page)

ചിലസമയങ്ങളിൽ ധാരാളം വിവരങ്ങൾ നമുക്ക് ചിട്ടയായരീതിയിൽ പ്രദർശിപ്പിക്കേണ്ട തായിവരും. ഉദാഹരണത്തിന് ഒരു ആശുപത്രിയിൽ 2012-2014 കാലയളവിൽ പുതിയ തായി ക്യാൻസർ കണ്ടെത്തിയ രോഗികളിൽ പുകവലി, പാൻ, മദ്യം, ഇതല്ലാതെ മറ്റൊരു രോഗങ്ങളാൽ ക്യാൻസർ ബാധിച്ചവരുടെ എണ്ണം എന്നിവ പട്ടിക രൂപത്തിൽ ടേബിൾ 5.1 തോന്തരിക്കുന്നു.

**ഒരു ആശുപത്രിയിൽ 2012 - 2014 കാലയളവിൽ പുതിയതായി ക്യാൻസർ കണ്ടെത്തിയ രോഗികൾ**

Year	2012	2013	2014
<b>Smokers</b>	129	140	143
<b>Pan users</b>	54	56	49
<b>Alcohol users</b>	74	68	77
<b>Other cases</b>	95	93	92

ടേബിൾ 5.1: ഒരു ആശുപത്രിയിൽ ക്യാൻസർ കണ്ടെത്തിയ രോഗികൾ

വരിയും നിരയുമായി സെല്ലുകൾ ചിട്ടയായി അടുക്കിവച്ചിരിക്കുന്നതാണ് ടേബിൾ എന്നത് നമുക്കേല്ലാം അറിയാം ഈ അറകളിൽ (cells) ചിത്രങ്ങളോ വാക്കുങ്ങളോ മറ്റാരു ടേബിളോ ഉൾക്കൊള്ളിക്കാം. HTML-ൽ <TABLE> എന്ന ടാഗ് ഉപയോഗിച്ചാണ് ടേബിൾ നിർമ്മിക്കുന്നത്. ഈ ടാഗിനോടൊപ്പം <TH>, <TR>, <TD> എന്നീ ടാഗുകളും ടേബിൾ നിർമ്മിക്കാൻ ഉപയോഗിക്കുന്നു. ഈ ടാഗുകളെക്കുറിച്ച് തുടർന്നുള്ള ഭാഗങ്ങളിൽ ചർച്ച ചെയ്യാം.

### 5.4.1 <TABLE> ടാഗ്

ടേബിൾ ടാഗ് ഒരു കണ്ടയിനർ ടാഗാണ്. ടേബിൾ നിർമ്മാണത്തിനാവശ്യമായ എല്ലാ ഉള്ള ടക്കങ്ങളും ടാഗ് ജോഡികളായ <TABLE>, </TABLE> എന്നിവയുടെ ഇടയിൽ നൽകും. ടേബിളിന്റെ കരടുരുപരേഖ തയാറാക്കുന്നതിനായി ആട്ടിബ്യൂട്ടുകൾ ഉപയോഗിക്കുന്നു. ഇതിൽ മുഖ്യമായവ താഴെ കൊടുത്തിരിക്കുന്നു.

1. **Border:** ടേബിളിന് ചുറുമുള്ള വരകളുടെ കൗമാന് ഈ ആട്ടിബ്യൂട്ടിലുടെ വ്യക്ത മാക്കുന്നത്. ഈ ആട്ടിബ്യൂട്ടിന് പുജ്യമല്ലാത്ത വില (പിക്സലായാണ് നൽകേണ്ടത്) നൽകിയാൽ ടേബിളിന്റെ ഭോർഡ് നിർമ്മിക്കാം. ഇതിന് പുജ്യം എന്ന വില നൽകുമ്പോൾ ടേബിളിന് ചുറുമുള്ള വരകൾ അപേത്യക്ഷമാകും.
2. **Bordercolor:** ടേബിളിന്റെ അതിരുകൾക്ക് നിറം നൽകുന്നു.
3. **Align:** ഭേദസർ വിസ്തേയായിലെ ടേബിളിന്റെ സ്ഥാനം നിർണ്ണയിക്കുന്നു. ഇതിന്റെ സാധ്യമായ വിലകൾ left (തനതുവില) right, center എന്നിവയാകുന്നു.
4. **Bgcolor:** ടേബിളിന് പശ്ചാത്തലവനിനു നൽകുന്നതിന് ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു.
5. **Background:** ടേബിളിന്റെ പശ്ചാത്തലത്തിൽ ഒരു ചിത്രം പ്രദർശിപ്പിക്കുന്നതിന് ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഇതിനായി ചിത്രം സുക്ഷിച്ചിത്തിക്കുന്ന സ്ഥലം (path) Background ആട്ടിബ്യൂട്ടിന്റെ വിലയായി നൽകുണ്ട്. ഉദാഹരണത്തിന് <TABLE Background = "images/flower.gif"> image എന്ന ഫോർമാർഡ് റിൽ സുക്ഷിച്ചിരിക്കുന്ന flower.jpg എന്ന ചിത്രം ടേബിളിന്റെ ഉള്ളിൽ പ്രദർശിപ്പിക്കുന്നു. <TABLE> ടാഗിന്റെ ആട്ടിബ്യൂട്ടുകളായി Bgcolor, Background എന്നിവ രണ്ടും നൽകിയാൽ Background ആട്ടിബ്യൂട്ടിന്റെ വിലയായ ചിത്രമായിരിക്കും പ്രദർശിപ്പിക്കുക.
6. **Cellspacing:** ടേബിൾ അറകൾക്ക് (Cell) ഇടയിൽ സ്ഥലം ഉണ്ട്. ഈ സ്ഥലം കൂടുവാനോ കുറയ്ക്കുവാനോ കഴിയും. Cellspacing എന്ന ആട്ടിബ്യൂട്ട് അറകൾക്ക് ഇടയിലുള്ള സ്ഥലം നിർണ്ണയിക്കുന്നു. ഇതിന്റെ വിലയായി പിക്സലുകളുടെ എള്ളൂമാൻ നൽകുന്നത്.
7. **Cellpadding:** ടേബിളിനുള്ളിലെ അറകളിൽ അതിൽ അടങ്ങിയിരിക്കുന്ന വസ്തുവും അറകളുടെ അതിർത്തിയും തമിലുള്ള സ്ഥാനവ്യത്യാസം നിർണ്ണയിക്കുന്നതിനായി ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. പിക്സലിലാണ് ഈ ഇതിന്റെ വിലയും നൽകേണ്ടത്.
8. **Width and Height:** ടേബിളിന്റെ വീതിയും ഉയരവും ഈ ആട്ടിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് നൽകാം. ഇതിന്റെ വിലകൾ പിക്സലിലോ ഭേദസർ വിസ്തേയായുടെ വീതിയുടെയും ഉയരത്തിന്റെയും ശതമാനമായോ നൽകാം.
9. **Frame:** ടേബിളിന്റെ അതിരുകൾ (Border) ഏത് രീതിയിൽ പ്രദർശിപ്പിക്കണം എന്ന തിനായി ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കാം. ഈ ആട്ടിബ്യൂട്ടിന്റെ വിലകൾ പട്ടികാരം രൂപത്തിൽ ടേബിൾ 5.2 റെ നൽകുന്നു.

വില	വിവരണം
Void	ടേബിളിന്റെ അതിരുകൾ പ്രദർശിപ്പിക്കാതിരിക്കാൻ.
Above	മുകൾഭാഗത്തെ മാത്രം അതിർ പ്രദർശിപ്പിക്കുന്നതിന്.
Below	ടേബിളിന്റെ താഴ്ഭാഗത്തെ മാത്രം അതിർ പ്രദർശിപ്പിക്കുന്നതിന്.



Value	Description
Hsides	ഡേവിലിന് തിരശ്വീനമായി (മുകളിലും താഴയും) മാത്രം ബോർഡ് പ്രദർശിപ്പിക്കുന്നതിന്.
lhs or rhs	ഡേവിലിന്റെ ഇടത് അല്ലെങ്കിൽ വലത് ഭാഗത്ത് മാത്രം ബോർഡ് പ്രദർശിപ്പിക്കുന്നതിന്.
Vsides	ഡേവിലിന്റെ ലംബമായി ഇടത്തും വലത്തും ഭാഗത്ത് മാത്രം ബോർഡ് പ്രദർശിപ്പിക്കുന്നതിന്.
box or border	ഡേവിലിന്റെ എല്ലാ ഭാഗത്തും ബോർഡ് പ്രദർശിപ്പിക്കുന്നതിന് (തന്തു വില)

ചെണിശ് 5.2: Frame അടിബ്യൂട്ടിന്റെ വിലകൾ

10. **Rules:** അറകളുടെ ചുറ്റുമുള്ള വരകൾ വരയ്ക്കുന്നതിനായി ഈ അടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഈ തിരിക്കേം വിലകൾ ഡേണിശ് 5.3-ൽ കാണിച്ചിരിക്കുന്നു.

വില	വിവരണം
none	അറകൾക്ക് (Cell) ചുറ്റും വരകൾ പ്രദർശിപ്പിക്കുന്നില്ല. (border)
cols	നിരകൾക്ക് (Columns) മാത്രം വരകൾ പ്രദർശിപ്പിക്കുന്നതിന്
rows	വരികൾക്ക് (rows) മാത്രം വരകൾ പ്രദർശിപ്പിക്കുന്നതിന്
groups	വരികളുടെ കൂട്ടങ്ങൾക്കും നിരകളുടെ കൂട്ടങ്ങൾക്കും വരകൾ പ്രദർശിപ്പിക്കുന്നതിന്
all	എല്ലാ വരികൾക്കും നിരകൾക്കും ഇടയിൽ വരകൾ പ്രദർശിപ്പിക്കുന്നതിന്

ചെണിശ് 5.3: Rules അടിബ്യൂട്ടിന്റെ വിലകൾ

ഈ `<TABLE>` ടാഗ്ഗമായി ബന്ധപ്പെട്ട മറ്റു ടാഗ്ഗകളെക്കുറിച്ച് ചർച്ച ചെയ്യാം.

#### 5.4.2 <TR> ടാഗ്

ഒരു ഡേണിശ് ടാഗിനുള്ളിൽ വരികൾ (rows) നിർമ്മിക്കുന്നതിനായി `<TR>` ടാഗ് ഉപയോഗിക്കുന്നു. ഈ ഒരു കണ്ണയ്ക്കൻ ടാഗ് ആണ്. ഓരോ വരിയും `<TR> </TR>` ടാഗ് ജോധികൾക്കിടയിൽ ഉൾപ്പെടുത്തുന്നു. `<TR>` ടാഗ് എല്ലായ്ക്കൊഴിഞ്ഞും `<TABLE>` ടാഗിനുള്ളിൽ മാത്രമേ ഉപയോഗിക്കാൻ കഴിയും. ഓരോ വരിയിലും ഒരു കൂട്ടം അറകൾ (Cells) അടങ്ങിയിരിക്കുന്നു. ഒരു ഡേവിലിലെ എറ്റവും ചെറിയയല്ദകമാണ് ഒരു അറ (Cell). ഡേവിലിനുള്ളിലെ അറകളെ (Cells) ഹൈഡ്രേറ്റ്, ഡാറ്റ എന്നിങ്ങനെ രണ്ടായി തരം തിരിക്കാം. ഡേണിശ് 5.1 ലെ കാണിച്ചിരിക്കുന്നതുപോലെ ചുവന്ന നിറത്തിൽ നൽകിയിരിക്കുന്നത് ഹൈഡ്രേറ്റ് ഡാറ്റ എന്നിൽ നിന്നും, നീല നിറത്തിൽ കാണുന്നത് ഡാറ്റ എന്നിലുകളും ആണ്.

### 5.4.3 <TH> ടാഗ്

ഹൈഡ്രിക്സ് സെല്ലുകൾ (Heading cells) നിർവ്വചിക്കുന്നതിനാണ് <TH> ടാഗ് ഉപയോഗിക്കുന്നത്. ഈതും ഒരു കണ്ടയ്ക്കൽ ടാഗ് ആണ്. ഹൈഡ്രിക്സ് <TR> </TR> എന്നീ ടാഗുകൾക്കിടയിൽ നൽകുന്നു. ഈതിൽ കൊടുത്തിരിക്കുന്ന ഹൈഡ്രിക്സ് കുടിയുള്ളതും (Bold face) സെല്ലിൽ മധ്യത്തിലും (Center) പ്രദർശിപ്പിക്കുന്നു. <TR> ടാഗിനുള്ളിലാണ് <TH> ടാഗ് എല്ലായ്പ്പോഴും നൽകുന്നത്.

### 5.4.4 <TD> ടാഗ്

<TH> ടാഗിന് ഏറ്റവും സമാനമായ <TD> ടാഗ് യേറ്റാസെല്ലുകൾ പ്രദർശിപ്പിക്കുന്നതിന് ഉപയോഗിക്കുന്നു. ഈത് ഒരു കണ്ടയ്ക്കൽ ടാഗ് ആണ്. <TD> </TD> ടാഗ്ജോഡികൾക്ക് ഇടയിൽ പ്രദർശിപ്പിക്കേണ്ട വിവരങ്ങൾ നൽകുന്നു. ഈ ടാഗ് <TH> ടാഗിനെ പോലെ <TR> ടാഗിന് ഉള്ളിലായാണ് നൽകുന്നത്.

ഉദാഹരണം 5.12 നൽകിയിരിക്കുന്ന HTML Code ഒരു ഫേബിൾ നിർമ്മിക്കാൻ ഉപയോഗിക്കുന്നു. ഈതിൽ വെബ്‌പോജ്ഞ് ചിത്രം 5.14-ൽ കാണിച്ചിരിക്കുന്നു.

**ഉദാഹരണം 5.12:** ഒരു സാധാരണ ഫേബിൾ ഉൾക്കൊള്ളുന്ന വെബ് പോജ്ഞ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE> Html Tables </TITLE>
</HEAD>
<BODY>
<TABLE Border="1">
<TR>
<TH>Roll No</TH>
<TH>Name</TH>
</TR>
<TR>
<TD>1</TD>
<TD>Aliya</TD>
</TR>
<TR>
<TD>2</TD>
<TD>Arun</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Html Tables - Mozilla Fi -	
File Edit View History Bookmarks	
Html Tables	
Roll No	Name
1	Aliya
2	Arun

ചിത്രം 5.14: ഒരു നിരകൾ ഉള്ള ഒരു സാധാരണ ഫേബിൾ ഉൾക്കൊള്ളുന്ന വെബ് പോജ്ഞ്

Year	2012 - 14
Smokers	412
Pan users	159
Alcohol users	219
Other cases	280

മുകളിൽ പ്രതിപാദിച്ച ഏതാനും ചില ആട്ടിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് നമുക്ക് താഴെ പറയുന്ന ഫേബിൾ നിർമ്മിക്കാം.

ഉദാഹരണം 5.13 - നൽകിയിരിക്കുന്ന HTML Code ഉപയോഗിച്ച് ചിത്രം 5.15 കാണു ന്നതുപോലുള്ള വെബ്പേജ് നിർമ്മിക്കാം.

**ഉദാഹരണം 5.13:** ടേബിളിന് ചുറ്റും ബോർഡിനും നിറങ്ങൾ നൽകിയതുമായ ടേബിൾ ഉൾക്കൊള്ളുന്ന വെബ്പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE> Hospital Table </TITLE> </HEAD>
<BODY>
    <TABLE Border= "3" Bordercolor= "RED" Bgcolor= "#4EB0AF"
           Align= "left" Cellspacing= "16" Cellpadding= "5"
           Width= "50%">
        <TR>
            <TH> Year </TH>
            <TH> 2012-14 </TH>
        </TR>
        <TR>
            <TH> Smokers </TH>
            <TD> 412 </TD>
        </TR>
        <TR>
            <TH> Pan users </TH>
            <TD> 159 </TD>
        </TR>
        <TR>
            <TH> Alcohol users </TH>
            <TD> 219 </TD>
        </TR>
        <TR>
            <TH> Other cases </TH>
            <TD> 280 </TD>
        </TR>
    <TABLE>
<BODY>
<HTML>
```

ഉദാഹരണം : 5.13-ൽ <TR>, </TR> സാർ ജോഡി കൾക്കിടയിൽ <TH> സാർ ഉപയോഗിക്കുന്നോൾ ചിത്രം 5.15 ലെ കാണിച്ചിരിക്കുന്നതുപോലെ ടേബിൾ ഭിന്നൾ ആദ്യത്തെ നിര ഫോറ്മേറ്റിനും തുല്യ മായിരിക്കും.

Year	2012-14
Smokers	412
Pan users	159
Alcohol users	219
Other cases	280

ചിത്രം 5.15: Cellspacing മുൻപുണ്ടാക്കിയ ടേബിൾ

### <TR> ടാഗിന്റെ ആട്ടിബ്യൂട്ടുകൾ

ഒരു വർത്തനയുടെ സഭാവസ്ഥിശേഷതകൾ മാറ്റുന്നതിനായി <TR> ടാഗിന്റെ ആട്ടിബ്യൂട്ടുകൾ ഉപയോഗിക്കുന്നു.

- Align:** അതത് വർത്തനിലെ അക്കളിലെ വാക്കുങ്ങൾ തിരശ്വീനമായി സ്ഥാപിക്കുന്ന തിന്ന് ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. Left, right or center എന്നീ വിലകൾ ഇതിന് നൽകാം. ഈ ആട്ടിബ്യൂട്ടിന്റെ തന്ത്ര വില ഡാറ്റായെ സംബന്ധിച്ചിട്ടെന്നാൽ left മുണ്ടായിരിക്കും. അതിനും left മുണ്ടായിരിക്കും.
- Valign:** ഏതെങ്കിലും ഒരു വർത്തനിലെ (row) അറയിലെ ഉള്ളടക്കം ലാംബമായി സ്ഥാപിക്കുന്നതിന് Valign ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കാം. ഇതിന്റെ വിലയായി top, middle, bottom or baseline എന്നീവ നൽകാം. Baseline ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുമ്പോൾ അക്കളിലെ വാക്കുങ്ങളെല്ലാം ഉള്ളടക്കാത്തയും അതിന്റെ വേസ് ലൈനോടുകൂടി സ്ഥാപിക്കുന്നു (ചിത്രം 5.15 കാണുക).
- Bgcolor:** ഒരു പ്രത്യേക വർത്തന പശ്വാത്തലനിറം നൽകുന്നതിനായി bgcolor ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഒരു വർത്തന പ്രത്യേകമായി എടുത്ത് കാണിക്കുന്ന തിന്ന് ഈത് സഹായിക്കുന്നു.

ഉദാഹരണം 5.13-ൽ നൽകിയ HTML കോഡിന്റെ പരിഷ്കരിച്ച കോഡ് ഭാഗം ചൂഡുന്നതും ഇതിൽ മുന്നാമത്തെ വർത്തനും പശ്വാത്തലം തിരശ്വീനവും ലാംബവുമായ അലെൻമെൻ്റും പരിഷ്കരിച്ച് ചിത്രം 5.15 ലെ പോലെ കാണിച്ചിരിക്കുന്നു.

```
<TR Bgcolor= "yellow" Align= "right"
Valign= "middle">
<TH> Pan users </TH>
<TD> 159 </TD>
</TR>
```

പരിഷ്കരിച്ച വെബ്പോജ് ചിത്രം 5.16 ത്ത് കാണിച്ചിരിക്കുന്നു.

### <TH> <TD> ടാഗുകളുടെ ആട്ടിബ്യൂട്ടുകൾ

<TH> <TD> ടാഗുകൾ ടേബിൾ അക്കൾ cells നിർമ്മിക്കുവാൻ ഉപയോഗിക്കുന്നതിനാൽ ഇതിന്റെ മികവൊറും ആട്ടിബ്യൂട്ടുകൾ ഒരു പോലെയാണ്. ഇതിൽ പ്രധാന പ്ലേറ്റ് ചില ആട്ടിബ്യൂട്ടുകൾ നമുക്ക് ചർച്ച ചെയ്യാം.

- Align:** മികവൊറും അറയിലെ വാക്കുങ്ങൾ തിരശ്വീനമായി സ്ഥാപിക്കുന്നതിന് വേണ്ടിയാണ് ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. ഈ ആട്ടിബ്യൂട്ടിന്റെ വിലയായി left, right, center ഇവയിൽ ഏതെങ്കിലും നൽകാം.
- Valign:** അക്കളിലെ ഉള്ളടക്കം ലാംബമായി സ്ഥാപിക്കുന്നതിന് ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഇതിന്റെ വിലയായി top, bottom, middle, baseline എന്നീവ നൽകാം.

ചിത്രം 5.16: Bgcolor, Align, Valign എന്നീവ ഉപയോഗിച്ച് പരിഷ്കരിച്ച വരി ഉൾപ്പെടുത്തുന്ന ടേബിൾ

- 3 . **Bgcolor:** എത്ത് അറയ്ക്കും (cell) ഒരു പ്രത്യേക പദ്ധതിലെന്നിം നൽകുന്ന തിനാണ് ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. <TABLE>, <TR>, <TD>/<TH> എന്നിവയ്ക്കുള്ളാം Bgcolor ആട്ടിബ്യൂട്ടുകൾ നൽകിയാലും <TH>/<TD> ടാഗിൽ Bgcolor ആട്ടിബ്യൂട്ടിന് നൽകുന്ന വിലയായിരിക്കും അറയുടെ പദ്ധതിലെന്നിര മായി പ്രദർശിപ്പിക്കുക.
- 4 . **Colspan:** സാധാരണയായി ഒരു അറ ഒരു നിരയിൽ മാത്രം ഒരുംബുന്നതായി നിക്കും. എന്നാൽ ചില നിരകളെ നിന്നും കുടുതൽ അറകളിലേക്ക് ചിലപ്പോൾ വ്യാപിച്ചിക്കേണ്ടതായി വന്നേക്കാം. ഒരു നിര എത്ര അറകളിലായി വ്യാപിച്ചിരി ക്കണം എന്നതെന്നുസിച്ച് Colspan ആട്ടിബ്യൂട്ടിൽ വിലനൽകാം. ഉദാഹരണം <TH Colspan="3"> എന്നത് ഒരു അറ (Cell) മുന്ത് നിരകളിലായി വ്യാപിച്ചിരി ക്കുന്നു.
- 5 . **Rowspan:** Colspan-നിന്ന പോലെ തന്നെ എത്ര വരികളിലായി (row) ഒരു അറ (cell) വ്യാപിച്ചിരിക്കുന്നു എന്നതിനാണ് ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നത്. ഉദാഹരണത്തിന് <TD Rowspan="4"> എന്ന് നൽകുന്നോൾ 4 വരികളിലായി ഒരു അറ വ്യാപിച്ചിരിക്കുന്നു.

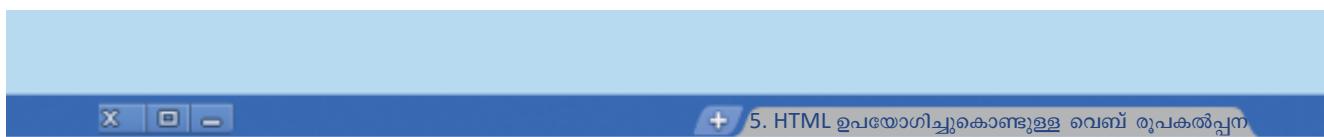
മുകളിൽ പറഞ്ഞ ആട്ടിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് ഒരു ഫേബിൾ നിർമ്മിക്കുന്നതിനുള്ള HTML കോഡ് ഉദാഹരണം 5.14 ത്ത് നൽകിയിരിക്കുന്നു. വെബ് പേജിൽ ചിത്രം 5.17 ത്ത് നൽകുന്നു.

#### **ഉദാഹരണം 5.14: Colspan, Rowspan എന്നീ ആട്ടിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് ഫേബിൾ നിർമ്മിക്കുന്നു**

```
<HTML>
<HEAD> <TITLE> Students Registration </TITLE>
</HEAD>
<BODY>
  <TABLE Border= "1" Cellspacing= "3" Cellpadding= "5">
    <TR>
      <TH Colspan= "3"> No. of Registered Students </TH>
    </TR>
    <TR>
      <TH Rowspan= "2"> Year </TH>
      <TD> 2014 </TD> <TD> 75 </TD>
    </TR>
    <TR>
      <TD> 2015 </TD> <TD> 88 </TD>
    </TR>
  </TABLE>
</BODY>
<HTML>
```

No. of Registered Students		
Year	2014	75
2015	88	

**ചിത്രം 5.17: Row span, column span എന്നീ ആട്ടിബ്യൂട്ടുകൾ ഉപയോഗിച്ച് ഫേബിൾ നിർമ്മിക്കുന്നു**



ഡേബിൾ 5.1-ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ക്യാൻസർ റോഗികളുടെ വിവരങ്ങൾ ഉൾക്കൊള്ളുന്ന വെബ്‌പേജ് നിർമ്മിക്കാം. ഈതിന്റെ HTML കോഡ്, ഉദാഹരണം 5.15 ലും ഫേബ്രുവരി 5.18 ലും നൽകിയിരിക്കുന്നു.

#### ഉദാഹരണം 5.15: ക്യാൻസർ റോഗികളുടെ വിവരങ്ങൾ അടങ്കുന്ന വെബ്‌പേജ്

```
<HTML>
<HEAD> <TITLE> CompleteTable </TITLE> </HEAD>
<BODY Bgcolor= "silver">
    <TABLE Border= "3" Bordercolor= "red" Bgcolor= "#4EB0AF"
           Align= "left" Cellspacing= "2" Cellpadding= "2"
           Width= "50%">
        <TR>
            <TH Colspan= "5"> Number of cancer patients reported
                               at the hospital </TH>
        </TR>
        <TR Align= "center">
            <TH Colspan= "2"> Year </TH>
            <TH> 2012 </TH>
            <TH> 2013 </TH>
            <TH> 2014 </TH>
        </TR>
        <TR Align= "center">
            <TH Rowspan= "4"> Cancer Origin </TH>
            <TH> Smokers </TH>
            <TD> 129 </TD>
            <TD> 140 </TD>
            <TD> 143 </TD>
        </TR>
        <TR Align= "center">
            <TH> Pan users </TH>
            <TD> 54 </TD>
            <TD> 56 </TD>
            <TD> 59 </TD>
        </TR>
        <TR Align= "center">
            <TH> Alcohol users </TH>
            <TD> 74 </TD>
            <TD> 68 </TD>
            <TD> 77 </TD>
        </TR>
        <TR Align= "center">
```



```

<TH> Other cases </TH>
<TD> 95 </TD>
<TD> 93 </TD>
<TD> 92 </TD>
</TR>
<TR Align= "center">
<TH Colspan= "2"> TOTAL Patients </TH>
<TD> 352 </TD>
<TD> 357 </TD>
<TD> 371 </TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Fig. 5.18: Web page containing a table of cancer patients Bgcolor, Rowspan, Colspan, Align and Valign



നിങ്ങളുടെ സ്കൂളിലെ വിവിധ രണ്ടാം ഭാഷകൾ പരിക്കുന്ന കുട്ടികളുടെ എല്ലാം പ്രദർശിപ്പിക്കുന്ന ദേഖിൽ നിർമ്മിക്കുക. ഈ ദേഖിലിൽ ഹയർ സെക്കൻഡറി വിഭാഗത്തിലെ ഓരോ കൂസിലെയും വിവരങ്ങൾ നമുക്ക് ചെയ്യാം ഉണ്ടായിരിക്കണം.

#### 5.4.5 <CAPTION> ടാഗ് ഉപയോഗിച്ച് ദേഖിളിരേണ്ട് ശീർഷകം

<CAPTION> ടാഗ് ഉപയോഗിച്ച് ഒരു ദേഖിളിരേണ്ട് ശീർഷകം നൽകാം. ഈ ടാഗ് ഉപയോഗിച്ച് ഒരു ദേഖിളിനെക്കുറിച്ചുള്ള വിവരങ്ങം അതിരേണ്ട് ശീർഷകമായി എല്ലാപ്പുത്തിൽ നൽകാം. നമുക്ക് ചിത്രം 5.19 ത്ത് നൽകിയിരിക്കുന്ന ദേഖിളിന് ശീർഷകം നൽകണം എന്ന് കരുതുക. ഉദാഹരണം 5.15 ലെ HTML കോഡിലെ <TR> ടാഗിന് മുൻപുള്ള ഭാഗത്തിന് താഴെ കൊടുക്കുന്നതുപോലെ മാറ്റം വരുത്താം.

5. HTML ഉപയോഗിച്ചുകൊണ്ടുള്ള വെബ് രൂപകൽപ്പന

```

<TABLE Border= "3" Bordercolor= "red" Bgcolor= "skyblue"
       Align= "left" Cellspacing= "2" Cellpadding= "2"
       Width= "50%">
<CAPTION> Number of new cancer patients reported at the
           hospital during 2012-14
</CAPTION>

```

ഇങ്ങനെ ഉണ്ടാക്കിയ ടൈപ്പ് ഫീൽഡിൽ 5.19 തോന്തരത്തിൽ കാണിച്ചിരിക്കുന്നു.

<b>Number of cancer patients reported at the hospital</b>				
<b>Year</b>		<b>2012</b>	<b>2013</b>	<b>2014</b>
<b>Cancer Origin</b>	<b>Smokers</b>	129	140	143
	<b>Pan users</b>	54	56	59
	<b>Alcohol users</b>	74	68	77
	<b>Other cases</b>	95	93	92
<b>TOTAL Patients</b>	352	357	371	

ചിത്രം 5.19: റൈറ്റ്സ്കോ ചേർത്തുകൊണ്ട് പുതുക്കിയ ടൈപ്പ്

### നിങ്ങളുടെ പുരോഗതി അറിയാം



1. <TABLE> ടാഗുമായി ബന്ധപ്പെട്ട അനുബന്ധമായ റെക്സ് ടാഗുകളുടെ പേരെഴുതുക.
2. താഴെക്കാടുത്തിരിക്കുന്നവയിൽ നിന്നും വ്യത്യാസമുള്ളത് തിരഞ്ഞെടുത്തുക.
  - a. TABLE
  - b. TR
  - c. TH
  - d. COLSPAN
3. <TD>, <TH> എന്നീ ടാഗുകളുടെ വ്യത്യാസങ്ങൾ എഴുതുക.
4. ഒരു empty ടാഗാണ് <TABLE> ഈ പ്രസ്താവന ശരിയോ തെറ്റോ എന്ന് എഴുതുക.
5. <TR> ടാഗിന്റെ ഏതെങ്കിലും റെക്സ് ആട്ടിബ്യൂട്ടുകൾ എഴുതുക.

## 5.5 ബ്രൗസർ വിൻഡോയുടെ വിഭജനം (Dividing the browser window)

നമുക്ക് ചിലപ്പോൾ ഒരു വെബ് ബ്രൗസർ വിൻഡോയിൽ ഓൺലൈനുടെ വെബ് പേജുകൾ ഉൾപ്പെടുത്തേണ്ടിവരും. ബ്രൗസർ വിൻഡോയെ രണ്ടോ അതിലധികമോ ഭാഗങ്ങളാക്കി തിരിച്ച് ഒരേ സമയം ഓരോ ഭാഗത്തും വ്യത്യസ്തങ്ങളായ വെബ് പേജുകൾ ഉൾക്കൊള്ളിക്കാം. HTML ലെ ബ്രൗസർ വിൻഡോയെ വിവിധ ഭാഗങ്ങളായി വിഭജിക്കുവാൻ ഫ്രെയിംസെറ്റ് എന്ന സൗകര്യം നൽകുന്നു. ഇങ്ങനെയുള്ള ഓരോ ഭാഗത്തിലും ഓരോ വെബ് പേജുകൾ ഉൾക്കൊള്ളിക്കാനാവും. ഒരു ഫ്രെയിംസെറ്റ് ഉപയോഗിച്ച് നിർമ്മി



കുന്ന ഓരോ വിഭാഗത്തിനേയും ഫ്രെയിംഗ് എന്നു വിളിക്കുന്നു. മുൻ ഫ്രെയിംഗുകൾ ഉൾക്കൊള്ളുന്ന ലളിതമായ ഫ്രെയിംഗ് സെറ്റിംഗ് ചിത്രം 5.20-ൽ ചിത്രീകരിച്ചിരിക്കുന്നു. നമുക്ക് <FRAMESET>, <FRAME> എന്നീ ടാഗുകൾ ഉപയോഗിച്ച് ഒരു ഫ്രെയിംഗ് സെറ്റ് നിർമ്മിക്കാം. ചിത്രം 5.20 മുൻ ഫ്രെയിംഗുകൾ അടങ്ങിയ ഒരു ഫ്രെയിംഗ് സെറ്റ്.



ചിത്രം 5.20: മുൻ ഫ്രെയിംഗുകൾ അടങ്ങിയ ഒരു സാധാരണ ഫ്രെയിംഗ്

### 5.5.1 <FRAMESET> ടാഗ്

ഒരു കണ്ടയ്ക്കൻ ടാഗായ <FRAMESET> ഒരു ബൈറ്റസർ വിൻഡോയെ വിവിധ ഫ്രെയിംഗ് ഭാഗങ്ങളായി വിജ്ഞിക്കുന്നു. <FRAMESET> </FRAMESET> എന്നീ ടാഗുകളുടെ ഇടയിലാണ് ഫ്രെയിംഗുകൾ നിർവ്വചിക്കുന്നത്. ഈ ടാഗിൽ പ്രധാനപ്പെട്ട ആട്ടിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.

- Cols:** ഈ ആട്ടിബ്യൂട്ട് ഫ്രെയിംഗ് പേജിലെ ലംബമായ ഫ്രെയിംഗുകളുടെ എള്ളൂദും അവയുടെ അളവുകളും നിർണ്ണയിക്കുന്നു. ഫ്രെയിംഗിൽ വീതി മൊത്തം വീതിയുടെ ശതമാനത്തിലോ പിക്സലിൽ എള്ളൂദും നൽകാം. ഉദാഹരണ മായി <FRAMESET Cols = "30%, 500, \*"> എന്നത് മുൻ ലംബമായ ഫ്രെയിംഗുകൾ സൃഷ്ടിക്കുന്നു. ഇതിൽ ആദ്യത്തെ ഫ്രെയിംഗ് വിൻഡോയുടെ മൊത്തം വീതി യുടെ 30 ശതമാനവും അടുത്തതിന് 500 പിക്സൽ സ്ഥലവും മുന്നാമത്തെ ഫ്രെയിംഗിൽ അളവ് (\*) ചിഹ്നത്തിനാൽ രേഖപ്പെടുത്തിയിരിക്കുന്നതിനാൽ ആകെ വീതി യുടെ ബാക്കിവരുന്ന സ്ഥലവും ആയിരിക്കും.
- Rows:** ഈ ആട്ടിബ്യൂട്ട് Cols, Rows നെ പോലെ തിരഞ്ഞീനമായ ഫ്രെയിംഗുകളുടെ എള്ളൂദും അളവും നിർവ്വചിക്കുന്നു.
- Border:** ഈ ആട്ടിബ്യൂട്ട് ഫ്രെയിംഗുകളുടെ അതിരിൽ (Border) കനം പിക്സലിൽ വ്യക്തമാക്കുന്നു. ഇതിൽ വില പിക്സലിൽ എള്ളൂദും നൽകുന്നത്.
- Bordercolor:** ഈ ആട്ടിബ്യൂട്ട് ഫ്രെയിംഗുകളുടെ അതിരിൽ (border) നിറം നൽകുവാൻ ഉപയോഗിക്കുന്നു.

### 5.5.2 <FRAME> ടാഗ്

<FRAMESET> ടാഗിനുള്ളിൽ ഫ്രെയിംഗുകൾ നിർവ്വചിക്കുന്ന എംറി ടാഗാണിത്. <FRAMESET> ടാഗിന് ഉള്ളിലെ ഓരോ ഭാഗത്തിനും തുല്യമായി <FRAME> ടാഗ് നൽകണം. ഫ്രെയിംഗുകളുടെ HTML പേജ് പ്രദർശിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്ന Src ആട്ടിബ്യൂട്ട് എപ്പോഴും <FRAME> ടാഗിനൊപ്പം ഉപയോഗിച്ചിരിക്കണം. <FRAME> ടാഗുമായി ബന്ധപ്പെട്ട പ്രധാനപ്പെട്ട ആട്ടിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.



1. **Src:** നമ്മൾ മുൻപ് ചർച്ചചെയ്തതുപോലെ, ഫ്രേമിലിൽ പ്രദർശിപ്പിക്കേണ്ട വെബ് പേജിന്റെ URL Src ആട്ടിബ്യൂട്ടിന്റെ വിലയായി സൂചിപ്പിക്കുന്നു. ഉദാഹരണമായി <FRAME Src = "school.html"> എന്നത് school.html എന്ന പേജ് ഫ്രേമിലിൽ പ്രദർശിപ്പിക്കുന്നു.
2. **Scrolling:** ആട്ടിബ്യൂട്ട് ഫ്രേമിലിലെ HTML പേജിൽ ലംബവും തിരഞ്ഞീസ്വാം മായ സ്ക്രോൾബാറുകൾ പ്രദർശിപ്പിക്കേണ്ടോ വേണ്ടയോ എന്ന് തീരുമാനിക്കുന്നു. Scrolling ആട്ടിബ്യൂട്ടിന് Yes, No, Auto എന്നീ വിലകൾ നൽകാം. Yes എന്ന വില നൽകുമ്പോൾ സ്ക്രോൾ ബാറുകൾ സ്ഥിരമായി പ്രത്യുക്ഷപ്പെടും. No എന്ന വില നൽകുമ്പോൾ സ്ക്രോൾ ബാറ് ഒന്നും പ്രത്യുക്ഷപ്പെടുന്നില്ല. ഇതിന്റെ തനതുവിലയായ Auto നൽകിയാൽ ഫ്രേമിലിന്റെ ഉള്ളടക്കം അതിന്റെ വലുപ്പത്തെ കാശ് അധികമാക്കുമ്പോൾ സ്ക്രോൾബാറുകൾ പ്രത്യുക്ഷപ്പെടും.
3. **Noresize:** ഉപയോക്താവ് ഒരു പ്രത്യേക ഫ്രേമിലിന്റെ ബോർഡറുകൾ വലിച്ച് നീട്ടാതിരിക്കാൻ ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഈ ആട്ടിബ്യൂട്ടിന് വിലകൾ അവസ്ഥയില്ല. ഉദാഹരണമായി <FRAME Src= "school.html" Noresize>.
4. **Marginwidth and Marginheight:** ഒരു ഫ്രേമിലിലെ ലംബവും തിരഞ്ഞീസ്വാം മായ മാർജിനുകൾ (margins) പിക്സലിന്റെ (pixels) എണ്ണത്തിൽ നൽകുന്നതിന് യഥാക്രമം Marginwidth, Marginheight എന്നീ ആട്ടിബ്യൂട്ടുകൾ ഉപയോഗിക്കുന്നു.
5. **Name:** ഫ്രേമിലിന് പേര് നൽകുന്നതിന് name ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. തുടർന്ന് വരുന്ന HTML കോഡിൽ ഫ്രേമിലെ സൂചിപ്പിക്കുന്നതിനായി ഈ പേര് ഉപയോഗിക്കാം.

ചുവവുടെ ചേർക്കുന്ന HTML കോഡ് ഉപയോഗിച്ച് നിർമ്മിച്ച ഫ്രേമിം സെറ്റ് ചിത്രം 5.20 ത്തെ കാണിച്ചിരിക്കുന്നു. മുൻ HTML പേജുകളായ sampleframe1.html, sampleframe2.html, sampleframe3.html എന്നിവ ഇതിന് മുമ്പേ നിർമ്മിച്ചിട്ടുണ്ട് എന്ന് കരുതുക.

```
<HTML>
<HEAD> <TITLE> A simple Frameset </TITLE> </HEAD>
<FRAMESET Rows= "20%, 30%, 50%">
    <FRAME Src= "sampleframe1.html">
    <FRAME Src= "sampleframe2.html">
    <FRAME Src= "sampleframe3.html">
</FRAMESET>
</HTML>
```

### 5.5.3 ഫ്രേമിലെ ടാർഗറ്റ് ചെയ്യുന്നു (Targeting frames)

ഒരു ഫ്രേമിംസെറ്റ് പേജിൽ നമ്മൾ നൽകുന്ന ഹൈപ്പർലിങ്കിലൂടെ ലിങ്ക് ചെയ്യുന്ന പേജിനെ മറ്റാരു ഫ്രേമിലേക്ക് ലക്ഷ്യം വയ്ക്കാം. നമ്മൾ ഫ്രേമിലിലെ ലിങ്കിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ സ്വന്ധപ്പെട്ട പേജ് മറ്റാരു ഫ്രേമിലിൽ തുറക്കും. ഇതിനുവേണ്ടി ആദ്യം നമ്മൾ ലക്ഷ്യം ഫ്രേമിംമിന് Name ആട്ടിബ്യൂട്ട് ഉപയോഗിച്ച് പേര് നൽകണം.



അതിനുശേഷം ലിങ്കിന്റെ <A> ടാഗിന്റെ Target ആട്ടിബ്യൂട്ട് ഉപയോഗിച്ച് പേര് നൽകിയ ഫ്രേമായിമിനെ സൂചിപ്പിക്കാം.



മുന്ന് HTML ഫയലുകളായ *bio.html*, *poem.html*, *fiction.html* എന്നിവ നിർമ്മിച്ച് അതിൽ ആത്മകമാ, കവിത, സോവൽ എന്നീ ഗണത്തി നമുക്ക് ചെയ്യാം ലുപ്പേ പുസ്തകങ്ങളുടെ ലിസ്റ്റ് തയാറാക്കുക.

ഇപ്പോൾ രണ്ട് ഫ്രേമുകൾ ഉൾപ്പെടുന്ന ഒരു വെബ് പേജ് നിർമ്മിക്കാം. ഈ ഫ്രേമുകളിലേക്കുള്ള ലിങ്കും രണ്ടാമത്തെത്തിൽ ലിങ്കുമായി ബന്ധപ്പെട്ട വെബ് പേജ് തുറക്കുന്നതിനും ഉപയോഗിക്കുന്നു. ഒരു ഉപയോക്താവ് ഒന്നാമത്തെ ഫ്രേമിലിൽ സൂചിപ്പിച്ച ഏതെങ്കിലും ഒരു ലിങ്കിൽ ക്ലിക്കു ചെയ്യുന്നേം അതിന് അനുബന്ധമായ ഫയൽ രണ്ടാമത്തെ ഫ്രേമിൽ തുറക്കുന്നു.

നിങ്ങൾ മുകളിൽ പ്രതിപാദിച്ചിരിക്കുന്ന മുന്ന് ഫയലുകൾ നിർമ്മിച്ചതിനുശേഷം ഉദാഹരണം 5.16 ലേ കൊടുത്തിരിക്കുന്നു. HTML കോഡ് ഉപയോഗിച്ച് ടാർജ്ജ് ആട്ടിബ്യൂട്ടിനെ കൂറിച്ച് മനസ്സിലാക്കാം.

#### ഉദാഹരണം 5.16: ഫ്രേമായുള്ള ലക്ഷ്യം വയ്ക്കുന്നതിന്റെ ഉദാഹരണം

താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് *main.html* എന്ന ഫയലിൽ സേവ് ചെയ്യുക.

```
<HTML>
<HEAD> <TITLE> Left Frame </TITLE> </HEAD>
<BODY Bgcolor= "#00AFFF" Text= "#282D2F">
    <H2> Select the option </H2>
    <FONT Size= "5">
        <A Href= "bio.html" Target= "right_frame">Biography</A><BR>
        <A Href= "poem.html" Target= "right_frame">Poetry</A><BR>
        <A Href= "fiction.html" Target= "right_frame">Fiction</A>
    </FONT>
</BODY>
</HTML>
```

ഇപ്പോൾ താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് ശേഖരിക്കുവാനുള്ള ഒരു ഫയൽ നിർമ്മിക്കാം.

```
<HTML>
<HEAD> <TITLE> Targeting
Frames </TITLE> </HEAD>
<FRAMESET Cols= "200, *">
    <FRAME Src= "main.html"
Name= "left_frame">
    <FRAME Name= "right_frame">
</FRAMESET>
</HTML>
```



Fig. 5.21 : Web page with target frame

മുകളിലെ യോക്കുമെന്ത് എക്സിക്യൂട്ട് ചെയ്യുമ്പോൾ രണ്ട് ഫ്രേയിമുകൾ ഉൾക്കൊള്ളുന്ന വെബ് പേജ് തുറക്കും. ബോസർ വിൻധോയിലെ ആദ്യത്തെ നിരയിൽ *main.html* കാണിക്കുന്നതും റണ്ടാമത്തെ നിര ഓഫൈസിക്കുകയും ചെയ്യും. ആ ഫ്രേയിമിന് വേണ്ടി **Src** ആട്ടിവും സൂചിപ്പിച്ചിട്ടില്ല. എന്നാൽ *right\_frame* എന്ന പേര് നൽകിയിട്ടുണ്ട് എന്ന കാര്യം ശ്രദ്ധിക്കുക. ഇതിൽ ആദ്യത്തെ ലിങ്കായ (*biography*) തിരഞ്ഞെടുത്താൽ *bio.html* എന്ന പയലിന്റെ ഉള്ളടക്കം HTML പയലുകളിലേക്ക് ലിങ്ക് ചെയ്യുവാൻ <A> ടാഗുകൾ ഉൾപ്പെടുത്തിയിട്ടുണ്ട്. വെബ്‌പേജിലെ റണ്ടാമത്തെ ഫ്രേയിമിൽ *bio.html* എന്ന പയലിന്റെ ഉള്ളടക്കം ചിത്രത്തിൽ കാണിച്ചിരിക്കുന്നതുപോലെ ആണെന്ന് കരുതുക. ചിത്രത്തിലെ ലിങ്കുകളുടെ നിരത്തിലെ വ്യത്യാസങ്ങളും ശ്രദ്ധിക്കുക.

*main.html* എന്ന HTML പേജിൽ മുന്ന് ജോഡി <A> ടാഗുകൾ മുന്ന് HTML ഫയലുകളിലേക്ക് ലിങ്ക് ചെയ്യുവാൻ ഉൾപ്പെടുത്തുന്നു. ടാഗിന്റെ ആട്ടിബ്യൂട്ടായ **Target** ഉപയോഗിച്ച് ലിങ്ക് ചെയ്ത ഫയൽ തുറക്കേണ്ട ഫ്രെയിമിനെ സൂചിപ്പിക്കുന്നു. ഫ്രെയിമിന് പേര് കൊടുക്കേണ്ടതിന്റെ ഒച്ചിത്യും ഇവിടെ സ്പഷ്ടമാകുന്നു.



HTML റേഖ 5-00 പതിപ്പ് <FRAMESET> നെ പിന്തുണയ്ക്കുന്നില്ല.

ഏമ്പയിംഗ്സെറ്റിന്റെ ഉപയോഗത്തിന് എതിരായി ധാരാളം വാദങ്ങൾ ഉയർന്നു വരുന്നു. ഇതിൽ ചിലത് ചുവപ്പേ ചേർക്കുന്നു.

- ഫ്രെയിംസെറ്റിനുള്ളിൽ സാധാരണയായി ബ്രഹ്മസിരേണ്ട് ബാക്കബട്ടൺ പ്രവർത്തനക്കു മമ്പി.
  - ഒരു ഫ്രെയിം സെറ്റിനുള്ളിൽ ഒരു പ്രത്യേക ഡ്യോക്കുമെന്റ് തുറക്കുവാൻ ബുദ്ധിമുട്ടാണ്.
  - ഒരു പ്രത്യേകമായ ഫ്രെയിം റീലോഡ് ചെയ്യാൻ ശ്രമിക്കുന്നോൾ മൊത്തത്തിലുള്ള ഫ്രെയിംസെറ്റ് റീലോഡ് ചെയ്തേക്കാം. ഈതിന്റെ ഫലമായി ഫ്രെയിംിന്റെ ഉള്ളടക്ക ഔദിയോഗിക്കുന്നതിലേക്ക് റീലോഡ് ചെയ്യും.
  - ഫ്രെയിംിലെ ഡ്യോക്കുമെന്റുകളിലുടെ നാവിഗേറ്റ് ചെയ്യാൻ സർച്ച് എൻജിനുകൾക്ക് ബുദ്ധിമുട്ടാണ്.

#### 5.5.4 ഫ്രേമൈംഗ് കളുടെ നെസ്റ്റിംഗ് (Nesting of framesets)

<p>ചിത്രം 5.20, 5.21 ലും ബേഹസർ വിൻഡോയിലെ ഫ്രെയിമുകൾ നമുക്ക് കാണാൻ സാധിക്കും. ആദ്യത്തെ ചിത്രത്തിൽ ബേഹസർ വിൻഡോ തിരഞ്ഞീനമായ മൂന്ന് ഫ്രെയിമുകളായി തിരിക്കുന്നു. നമുക്ക് ബേഹസർവിൻഡോ ചിത്രം 5.22-ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ലംബമായ രെങ്ക് ഫ്രെയിമുകളായി ഭാഗിക്കണമെന്ന് കരുതുക. ഈത് കൂട്ടിയിണക്കിയ ഫ്രെയിമുകൾ ഉപയോഗിച്ച് സാധ്യമാകും. ഒരു ഫ്രെയി</p>		
<b>Frame 1</b>	<b>Frame 2</b>	<b>Frame 3</b>

Frame 1	
Frame 2	Frame 3

### ചിത്രം 5.22: ഐഡിയോസെറ്റിന്റെ നേരുക്കിടങ്ങ്

സെറ്റിനുള്ളിൽ മറ്റാരു ഫേയിം സെറ്റ് കൂടിച്ചേർക്കുന്ന പ്രവർത്തനത്തെ നേറ്റിംഗ് ഓഫ് ഫേയിംസെറ്റ് എന്നു വിളിക്കുന്നു.

ചിത്രം 5.22-ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഫേയിംസെറ്റ് നിർമ്മിക്കുന്നതിനുള്ള വിവിധ ഘട്ടങ്ങൾ ചുവടെ ചേർക്കുന്നു.

1 . രണ്ട് വരികളായി വിഭജിക്കുന്ന ആദ്യത്തെ ഫേയിം നിർമ്മിക്കുക.

```
<FRAMESET Rows= "85, *">
    <FRAME Src= "sampleframe1.html">
</FRAMESET>
```

ബ്രൗസർ വിൻഡോയെ തിരശ്വീനമായ രണ്ട് ഫേയിമുകളാക്കി വിഭജിക്കുകയും ആദ്യത്തെ വരി *sampleframe1.html* എന്നതിന് മാറ്റി വയ്ക്കുകയും ചെയ്യുക.

2. രണ്ടാമത്തെ വരി സ്വതന്ത്രമായി ഇരിക്കുകയാണല്ലോ? ഇപ്പോൾ നമ്മൾ ഇതിനെ ലംബമായ രണ്ട് ഫേയിമുകളാക്കി വിഭജിക്കുന്നു. അതിന്റെ രണ്ടാമത്തെ <FRAME> ടാഗിന് പകരം പുതിയൊരു <FRAMESET> ടാഗ് തുറക്കുന്നു.

```
<FRAMESET Rows= "85, *">
    <FRAME Src= "sampleframe1.html">
    <FRAMESET Cols= "220, *">
        </FRAMESET>
    </FRAMESET>
```

ഈ കോഡ് രണ്ടാമത്തെ വരിയെ രണ്ട് നിരയായി ഭാഗിക്കുന്നു. ഇപ്പോൾ അക്കത്തെ ഫേയിം സെറ്റിനുള്ളിൽ നമുക്ക് രണ്ട് <FRAME> ടാഗുകൾ കൂടിച്ചേർത്ത് HTML കോഡ് പൂർത്തിയാക്കാം. ഇതിന്റെ ഉദാഹരണം 5.17 ത്ത് കാണിച്ചിരിക്കുന്നു. ഇതിന്റെ വെബ്പേജിന്റെ ചിത്രം 5.23 കാണിക്കുന്നു. ഇതിൽ ഉപയോഗിച്ചിരിക്കുന്ന മുൻ HTML പേജുകൾ ചിത്രം 5.17 കാണിച്ചിരിക്കുന്നതുപോലെ നേരത്തെ തന്നെ നിർമ്മിച്ചിട്ടുണ്ട് എന്ന് കരുതുക.

#### ഉദാഹരണം 5.17: നേറ്റിംഗ് ഫേയിം സെറ്റിന്റെ ആഴയം വിശദമാക്കുന്നു

```
<HTML>
    <HEAD>
        <TITLE> nesting frames </TITLE>
    </HEAD>
    <FRAMESET Rows= "85, *">
        <FRAME Src= "sampleframe1.html">
        <FRAMESET Cols= "200, *">
            <FRAME Src= "sampleframe2.html">
            <FRAME Src= "sampleframe3.html">
        </FRAMESET>
    </FRAMESET>
</HTML>
```

### 5.5.5 <NOFRAMES> ടാഗ്

പഴയ ചില ബേഖസറുകൾ ഫ്രേമീമുകൾ പിന്തു നിയ്ക്കുന്നില്ല. ഈ സാഹചര്യം ബേഖസറിന് ഉപയോകതാവിനെ അറിയിക്കേണ്ടതുണ്ട്. ഈതിനായി <NOFRAMES> </NOFRAMES> ടാഗ് ജോധികൾ ഉപയോഗിച്ച് ഫ്രേമീംസ് പിന്തുണ യ്ക്കാത്ത ബേഖസർ വിൻഡോയിൽ ചില വിവരങ്ങൾ പ്രദർശിപ്പിക്കാനാകും. സന്ദേശങ്ങൾ നൽകാനാവും. താഴെക്കാടുത്തിരിക്കുന്ന കോഡ് <NOFRAMES> ടാഗിന്റെ ഉപയോഗം വിവരിക്കുന്നു.



ചിത്രം 5.23: നേഡ്യൂൾ ഫ്രേമീംസ് സൈറ്റ്

```
<HTML>
  <HEAD> <TITLE> A simple Frameset </TITLE>
  </HEAD>
  <FRAMESET Rows= "20%, 30%, 50%">
    <FRAME Src= "sampleframe1.html">
    <FRAME Src= "sampleframe2.html">
    <FRAME Src= "sampleframe3.html">
  </FRAMESET>
  <NOFRAMES>
    <P> Your browser does not support frames.<BR>
      Click <A Href="index.htm">here... </A></P>
  </NOFRAMES>
</HTML>
```

ബേഖസർ <FRAMESET> ടാഗ് പിന്തുണയ്ക്കുന്നില്ലെങ്കിൽ "Click here..." എന്ന സന്ദേശത്തിനോട് കൂടി *index.html* ലേക്കുള്ള ലിങ്ക് പകരം ലഭ്യമാകും.

### നിങ്ങളുടെ പുരോഗതി അറിയാം

1. <FRAMESET Rows="100, \* "> എന്നത് ബേഖസർ വിൻഡോയെ ഭാഗങ്ങളാക്കി വിഭജിക്കുന്നു.
2. <FRAME> ടാഗിന്റെ ഏതെങ്കിലും മുന്ന് ആട്ടിബ്യൂട്ടുകൾ എഴുതുക.
3. നേഡ്യൂൾ ഫ്രേമീംസൈറ്റ് എന്നാൽ എന്ത്?
4. <NOFRAME> ടാഗിന്റെ ഉപയോഗം എന്താണ്?
5. ഫ്രേമീംസൈറ്റ് പേജിന് <BODY> ഭാഗത്തിന്റെ ആവശ്യകത ഇല്ല. ശരിയോ തെറ്റോ?



## 5.6 വെബ് പേജുകളിലെ ഫോമുകൾ (Forms in web pages)

ഉപയോകതാവിൽ നിന്ന് ധാരാ ശേഖരിച്ച് പ്രവർത്തനം നടത്തുവാൻ HTML ഫോമുകൾ ആവശ്യമാണ്. ഉദാഹരണമായി കേരളത്തിലെ XI-ാം ക്ലാസിലെ പ്രവേശനത്തിന് www.hscap.kerala.gov.in എന്ന വെബ്സൈറ്റിൽ നിങ്ങളുടെ പേര് SSLC റജിസ്ട്രേഷൻ നമ്പർ, ശ്രേണികൾ, ആവശ്യമുള്ള വിഷയങ്ങൾ, നേക്കുൾ തുടങ്ങിയ വിവരങ്ങൾ നൽകേണ്ടതുണ്ട്. ഇതിനായി നമ്മൾ HTML ഫോം ഉപയോഗിക്കാം.

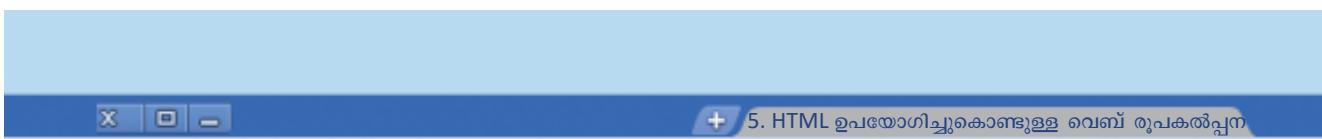
ഉപയോകതാവിൽ നിന്ന് വിവരങ്ങൾ സീക്രിച്ച് ബാക്സ് -എർഡ് ആബ്ലിക്കേഷൻകളായ കോമൺഗ്രേറ്റ്‌വേ ഇൻർഫേസ് (CGI) ആക്ടിവ് സെർവർ പേജുകൾ (ASP), PHP ഒരു HTML ഫോം വെബ്പേജ് തുടങ്ങിയവയിലേക്ക് ധാരാപ്രോസസിൽ നടത്തുവാനായി അയയ്ക്കുന്നു. അടുത്ത അധ്യായത്തിൽ ഈ സാങ്കേതിക വിദ്യയെക്കുറിച്ച് കൂടുതലായി പഠിക്കാം. ഒരു ഫോമിന് <FORM> രണ്ട് ഇനങ്ങൾ ഉണ്ട്. ഒരു ഫോമിന് ഫോം കണികയ് നടപ്പിലുള്ള വിവിധ കണ്ട്രോളുകളുമാണ്. വിവിധ തരത്തിലുള്ള ഫോം കണ്ട്രോളുകൾ ലിന്റ്, ടെക്സ്റ്റ് ഏരിയ, ഫ്രോപ്-ഡാൽ മെനുകൾ, റേഡിയോ ബട്ടണുകൾ, ചെക്ക് ബോക്സ് തുടങ്ങിയവയാണ്.

### 5.6.1 <FORM> ടാഗ്

<FORM> ടാഗ് ഫോം നിർമ്മിക്കുന്നതിനുള്ള കണക്കുന്നർ ആയി പ്രവർത്തിക്കുന്നു. <FORM> തും ആരംഭിക്കുന്നതും </FORM> തും അവസാനിക്കുന്നതുമായ ടാഗ് ജോഡി ഉപയോഗിച്ചാണ് ഫോം നിർമ്മിക്കുന്നത്. ഒരു വെബ് ബ്രൗസറിന് ഫോമിൽ കൂടെ മാത്രമേ വിവരങ്ങൾ ശേഖരിക്കുവാൻ കഴിയു. ശേഖരിച്ച ഡോറ്റ് കൈകാര്യം ചെയ്യാൻ ചില ബാക്സ്-എർഡ് ആബ്ലിക്കേഷൻകളായ CGI, JavaScript, PHP എന്നിവ ഉപയോഗിക്കാം. ഒരു ഫോമിൽ നിന്ന് ശേഖരിച്ച എല്ലാ ഇൻപുട്ടുകളും ഒരേ ഫോം ഹാൻഡലറിൽ കൂടി (Form handler) മാത്രമേ പ്രോസസ് ചെയ്യുവാൻ കഴിയു. ഒരു ഫോമിൽക്കൂടി അയച്ച ധാരാ നിയന്ത്രിക്കുന്ന വെബ് സർവ്വറിലെ ഒരു പ്രോഗ്രാമാണ് ഫോം ഹാൻഡലർ (Form handler) <FORM> ടാഗിലെ Action ആട്ടിബ്യൂട്ടിന്റെ വിലയായി ഫോം ഹാൻഡലറിന്റെ പേര് നൽകുന്നു. സർവ്വർ അനുബന്ധസാങ്കേതിക വിദ്യകൾ എന്നിവയുടെ ആശയം അടുത്ത അധ്യായത്തിൽ ചർച്ചചെയ്യാം.

<FORM> ടാഗിൽ പതിവായി ഉപയോഗിക്കുന്ന ആട്ടിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.

- Action:** ഈ ആട്ടിബ്യൂട്ടിന്റെ വിലയായി ധാരാ സീക്രിക്കുവാനും പ്രോസസ് ചെയ്യുവാനും കഴിയുന്ന ഫോം ഹാൻഡലറിന്റെ URL നൽകുന്നു.
- Method:** ഈ ആട്ടിബ്യൂട്ടിന്റെ വിലയായി ധാരാ അപ്ലോഡ് (upload) ചെയ്യുന്ന രീതി സൂചിപ്പിക്കുന്നു. സാധാരണയായി Get ഉം Post ഉം രീതികളാണ് അവലംബിക്കുന്നത്.
- Target:** ഈ ആട്ടിബ്യൂട്ട് സ്ക്രിപ്റ്റിന്റെ ഫലം പ്രദർശിപ്പിക്കേണ്ട വിന്റോദ്ദേശ ഫോറമോ സൂചിപ്പിക്കുന്നു. ഇതിന്റെ വിലകൾ പട്ടിക 5.4-ൽ നൽകുന്നു \_blank, \_self, \_parent.



വില	വിവരണം
_blank	പുതിയ വിൻധോയായിൽ ലിങ്ക് ചെയ്ത ഡോക്യുമെന്റ് തുറക്കുന്നു.
_self	ലിങ്ക് ചെയ്ത ഡോക്യുമെന്റ് അതേ ഫ്രെയിമിൽ തുറക്കുന്നു.
_parent	പേരുൾ ഫ്രെയിംസൈറ്റിൽ ലിങ്ക് ചെയ്യേണ്ട ഡോക്യുമെന്റ്.
_top	ഇപ്പോഴുള്ള ഫ്രെയിമിന് പകരം മുഖ്യഭേദസർ വിൻധോയായിൽ തുറക്കുന്ന നിലവിലുള്ള ഫ്രെയിമിനെ മാറ്റിക്കൊണ്ട് ലിങ്ക് ചെയ്യേണ്ട ഡോക്യുമെന്റ് തുറക്കുന്നു.
name	സൂചിപ്പിച്ചിരിക്കുന്ന പേര് ഉപയോഗിച്ച് വിൻധോയായിൽ ലിങ്ക് ചെയ്യേണ്ട ഡോക്യുമെന്റ് തുറക്കുന്നു.

ഡേബിൾ 5.4: Target ആട്ടിബ്യൂട്ടിന്റെ വിലകൾ



ബേദസറിൽ നിന്ന് വെബ് സെസ്റ്റർവിലെ ഫോം ഹാൻഡ്ലറിലേക്ക് ഫോം സമർപ്പിക്കുന്നതിന് മുമ്പ് ഉപയോകതാവ് ഫോമിൽ എല്ലാ അവസ്യവിവരങ്ങളും നൽകിയിട്ടുണ്ടോ എന്നും അല്ലെങ്കിൽ സീക്രിച്ച് വിവരങ്ങൾ നിലനിൽക്കുന്ന താണ്ടാ അല്ലെങ്കിലും എന്നും പരിശോധിക്കുന്നതിനാണ് ജാവാ സ്ക്രിപ്റ്റ് അല്ലെങ്കിൽ മറ്റൊരു ഫോം ഫോം സെസ്റ്റർ സ്ക്രിപ്റ്റിൽ ഭാഷ സാധാരണയായി ഉപയോഗിക്കുന്നത്.

## ഫോം കൺട്രോളുകൾ (Form controls)

HTML ഫോമിൽ നിന്ന് യേറ്റ് ശേഖരിക്കാൻ വിവിധതരം ഫോം കൺട്രോളുകൾ ഉപയോഗിക്കാം. ഇവയിൽ ഒക്ലൂൾ ബോക്സ്, പാസ്വോർഡ്, ചെക്ക് ബോക്സ്, റേഡിയോ ബട്ടൺ, ഒക്ലൂൾ ഏരിയ, സെലക്ക് ബോക്സ്, സബ്മിറ്റ്, റീസെറ്റ് ബട്ടൺ തുടങ്ങിയവ ഉൾപ്പെടുന്നു. ഇതിലെ മിക്ക കൺട്രോളുകളും <INPUT> ടാഗ് വഴി ഫോമിൽ ഉൾപ്പെടുത്താം.

### 5.6.2 <INPUT> ടാഗ്

ഉപയോകതാക്കളിൽ നിന്ന് ധാരാ സീക്രിക്കാൻ ഫോമിൽ വിവിധതരം കൺട്രോളുകൾ പ്രദർശിപ്പിക്കും. ഇൻപുട്ടിന്റെ തരം അനുസരിച്ച് വിവിധതരത്തിലുള്ള കൺട്രോളുകളായ ഒക്ലൂൾ ബോക്സ്, റേഡിയോ ബട്ടൺ, സബ്മിറ്റ് ബട്ടൺ തുടങ്ങിയവ <INPUT> ടാഗ് ഉപയോഗിച്ച് നിർമ്മിക്കാം ഇതിന്റെ Type ആട്ടിബ്യൂട്ട് കൺട്രോളിന്റെ തരം തീരുമാനിക്കുന്നു.

### <INPUT> ടാഗിന്റെ ആട്ടിബ്യൂട്ടുകൾ

1. **Type:** <INPUT> ടാഗിന്റെ ആട്ടിബ്യൂട്ടുകൾ <INPUT> ടാഗ് നിർമ്മിക്കുന്ന കൺട്രോളിന്റെ തരം നിർണ്ണയിക്കുന്നത് ഇതു ആട്ടിബ്യൂട്ടാണ്. ഇതിന്റെ പ്രധാനപ്പെട്ട വിലകൾ ഡേബിൾ 5.5ൽ കൊടുത്തിരിക്കുന്നു.

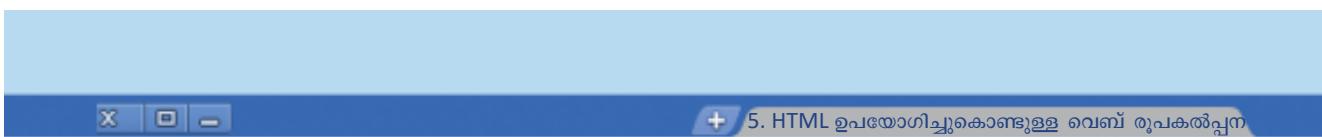
വില	വിവരണം
ടെക്സ്റ്റ്	ടെക്സ്റ്റ് ബോക്സ് നിർമ്മിക്കുന്നു.
പാസ്വേർഡ്	ടെക്സ്റ്റ് ബോക്സ് തന്നെയാണ് ഈതും നിർമ്മിക്കുന്നത് എന്നാൽ ഇതിൽ നൽകുന്ന അക്ഷരങ്ങൾ എല്ലാം അസ്സിന്റെ (* ) പോലുള്ള ചിഹ്നങ്ങളാണ് പ്രദർശിപ്പിക്കുക.
ചെക്ക് ബോക്സ്	അതെ, അല്ല എന്നീ വിലകൾ നൽകുന്നതിന് നിർമ്മിച്ചിരിക്കുന്നു. ഒരു കൂടം വിലകളിൽ ഒന്നോ അതിലധികമോ തിരഞ്ഞെടുക്കാൻ ഉപയോഗിക്കുന്നു.
റേഡിയോ	ഒരു കൂടം വിലകളിൽ നിന്ന് ഒരു വില തിരഞ്ഞെടുക്കുവാൻ ഇത് ഉപയോഗിക്കുന്നു. Name ആട്ടിബ്യൂട്ടിന് ഒരേ വിലയുള്ള ഓൺലൈൻ റേഡിയോ ബട്ടനിൽ നിന്ന് ഒന്നുമാത്രമേ ഒരുസമയത്ത് തിരഞ്ഞെടുക്കുവാൻ കഴിയും. ഉപയോകതാവ് സെലക്ഷൻ മാറ്റുമ്പോൾ നേരത്തെ തിരഞ്ഞെടുത്തത് ഡിസൈൻ ലക്ക് ചെയ്യപ്പെടും.
reset	ഒരു ഫോമിലെ രേഖപ്പെടുത്തലുകളെല്ലാം ഒഴിവാക്കി ഫോമിനെ പ്രാരംഭിച്ചാൽ ഉപയോഗിക്കുന്ന പ്രത്യേക ബട്ടൺ ആണിത്.
submit	ഫോമിൽ നൽകിയ ഡാറ്റാസർവാറിലേക്ക് സമർപ്പിക്കുന്നതിനായി ഉപയോഗിക്കുന്ന പ്രത്യേക ബട്ടണാണിത്.
button	ഫോമിൽ ഒരു ഗ്രാഫിക്കൽ ബട്ടൺ നിർമ്മിക്കുവാൻ ഉപയോഗിക്കുന്നു. ഈ ബട്ടണിൽ ക്ലിക്ക് ചെയ്ത് ഫംഷനുകളെ വിളിക്കാം.

ഡേവിൾ 5.5: Type ആട്ടിബ്യൂട്ടിന്റെ വിവിധ വിലകൾ

2. **Name:** ഇൻപുട്ട് കൺട്രോളുകളുടെ പേര് നൽകുന്നു. ഫോം സമർപ്പിക്കുമ്പോൾ ഡാറ്റായും ഡാറ്റ ഉൾക്കൊള്ളുന്ന കൺട്രോളിന്റെ പേരും സർവാറിലേക്ക് അയയ്ക്കുന്നു.
3. **Value:** ഒരു കൺട്രോളിന് തന്ത്ര വില നൽകുന്നതിന് ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു.
4. **Size:** ഇൻപുട്ട് ടെക്സ്റ്റ് ടെക്സ്റ്റ് ബോക്സ്, പാസ് വേർഡ് ടെക്സ്റ്റ് എന്നിവയുടെ വലുപ്പം നിശ്ചയിക്കുന്നതിന് ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു ഇൻപുട്ടുകളായ ടെക്സ്റ്റ്, പാസ്വേർഡ് എന്നിവയ്ക്ക് മാത്രം ഇത് ബാധകമാകും.
5. **maxlength:** ടെക്സ്റ്റ് ബോക്സ്, പാസ്വേർഡ് എന്നിവയിൽ ഉപയോകതാവിന് ഇൻപുട്ട് ചെയ്യുവാൻ സാധിക്കുന്ന പരമാവധി കൃംജിനുകളുടെ എല്ലാം നിശ്ചയിക്കുവാൻ ഈ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു. ഈ ആട്ടിബ്യൂട്ട് ടെക്സ്റ്റ് ബോക്സ്, പാസ് വേർഡ് എന്നിവയുടെ കുടെ മാത്രമേ ഉപയോഗിക്കും.



പാസ്വേർഡ് ഫൈലിലെ ഉള്ളടക്കങ്ങൾ ഉപയോകതാവിന്റെ കാഴ്ചയിൽ നിന്ന് മറയ്ക്കുന്നു. ഇതിൽ നമ്മൾ നൽകുന്ന വിവരങ്ങൾ രഹസ്യക്കോഡ് ആക്കിമാറുകയോ കൂടിക്കലർത്ഥകയോ ചെയ്യുന്നില്ല.

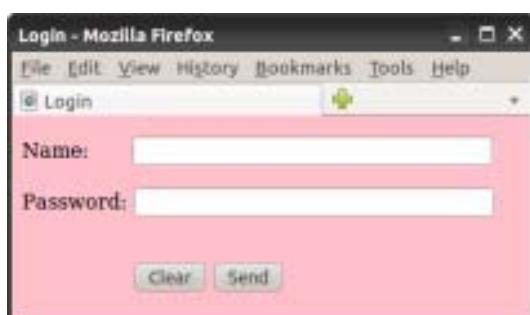


രാജാ എന്ന പേരിൽ, പാസ്വോർഡ് എന്നിവ നൽകുന്നതിനുള്ള HTML കോഡ് ഉദാഹരണം 5.18 ലും വൈബ് പേജിൽ ചിത്രം 5.24 ലും കാണിക്കുന്നു.

### ഉദാഹരണം 5.18: HTML ഫോം നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE> Login </TITLE> </HEAD>
<BODY Bgcolor= "Pink">
    <FORM Action= "login.php" Method= "post">
        <P> Name:&nbsp; &nbsp;&nbsp;
        <INPUT Type= "text" Name= "firstname" Size= "30"
               Maxlength= "25"> </P>
        <P> Password:
        <INPUT Type= "password" Name= "Psswd" Size= "30"
               Maxlength= "25"> </P> <BR>
        &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
        <INPUT Type= "reset" Value= "Clear">
        <INPUT Type= "submit" Value= "Send">
    </FORM>
</BODY>
</HTML>
```

ചിത്രം 5.24 ടെക്സ്റ്റ്, പാസ്വോർഡ് എന്നിവ ഉൾപ്പെട്ട HTML ഫോം. ഉദാഹരണം 5.18 ലെ <FORM> ടാഗിലെ Action ആഡി ബ്യൂട്ടിൽ വിലയായി login.php നൽകുന്നു. ഇതിനർമ്മം ഫോം മോഡിക്കുടി നൽകപ്പെടുന്ന യേറ്റ് login.php എന്ന ഫയലിൽ ശേഖരിച്ചിരിക്കുന്ന ആസ്ഥിക്കേഷൻിലും ഇത് പ്രോസസ് ചെയ്യപ്പെടുന്നു എന്നാണ്.



ചിത്രം 5.24: ടെക്സ്റ്റ്/പാസ്വോർഡ് എന്നിവ ഉൾപ്പെട്ട HTML ഫോം

രാജാ റേഡിയോ ബട്ടൺ (ഓപ്പ്‌ഷൻ ബട്ടൺ), ചെക്ക് ബോക്സ് എന്നിവയിലും യേറ്റ് ഇൻപുട്ട് ചെയ്യുന്ന HTML ഫോം രൂപകൽപ്പന ചെയ്യാം. ഒന്നിൽ കൂടുതൽ വിലകളിൽ നിന്ന് രാജാ വില മാത്രം തിരഞ്ഞെടുക്കാൻ റേഡിയോ ബട്ടൺം. ഒന്നിലധികം വിലകൾ തിരഞ്ഞെടുക്കുവാൻ ചെക്ക് ബോക്സും ഉപയോഗിക്കുന്നു. സാധാരണയായി രാജാ ലിസ്റ്റിലെ

അനോ അതിലധികമോ ഇനങ്ങളെ മാർക്ക് ചെയ്യുന്ന റീതിയിലാണ് ചെക്ക് ബോക്സ് ഉപയോഗിക്കുന്നത്.

ഉദാഹരണം 5.19 ത്ര നൽകിയിരിക്കുന്ന HTML കോഡ് ഒരു വ്യക്തിയുടെ ജോലിയും ഹോബികളും തമാക്രമം രേഖിയോ ബട്ടൺ, ചെക്ക്‌ബോക്സ് എന്നിവയിലും സൂചിപ്പിക്കുന്നു. ഇതിനുസ്വരൂപമായ വെബ്പേജ് ചിത്രം 5.25 ത്ര കാണിക്കുന്നു.

**ഉദാഹരണം 5.19:** രേഖിയോ ബട്ടൺ ചെക്ക്‌ബോക്സ് എന്നിവ ഉൾപ്പെടുത്തിയ ഒരു HTML ഫോം നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Checkbox Radio Button Control</TITLE> </HEAD>
<BODY Bgcolor= "#E9BEE5">
    <FORM> <BR>Sex:    &nbsp; &nbsp; &nbsp; &nbsp;
        <INPUT Type= "radio" Name= "sex" Value= "male"> Male
        <INPUT Type= "radio" Name= "sex" Value= "female"> Female
        <BR> <BR>Hobbies:
        <INPUT Type= "checkbox" Name= "Hobby" Value= "Games">
        Playing Games
        <INPUT Type= "checkbox" Name= "Hobby"
            Value="WatchingTV"> Watching TV
        <INPUT Type= "checkbox" Name= "Hobby" Value= "Reading">
        Reading
    </FORM>
</BODY>
</HTML>
```



ചിത്രം 5.25: ചെക്ക്‌ബോക്സും രേഖിയോ ബട്ടൺും ഉൾപ്പെടുത്തിയ വെബ്പേജ്.

### 5.6.3 <TEXTAREA> ടാഗ്

ഒരു ഫോമിൽ <INPUT Type = "text"> എന്ന പ്രസ്താവന ഉപയോഗിച്ച് ഒറ്റ വരി വാക്കും നൽകുന്ന റീതി നാം കണ്ടു കഴിഞ്ഞു. (ഉദാഹരണം : 5.18 ഉം ചിത്രം 5.24) നോക്കുക. എന്നാൽ ചില അവസരങ്ങളിൽ നന്നിലധികം വരികളുള്ള വാചകങ്ങൾ ഫോമിൽകൂടി നൽകേണ്ടത് ആവശ്യമായി വരും. ഒരു വ്യക്തിയുടെ തഹാൽ വിലാസം ഇതിന് ഒരു ഉദാഹരണമാണ്. ഇതിനായി <TEXTAREA> എന്ന കണ്ടയ്ക്കൽ ടാഗ് ഉപയോഗിക്കാം. ടാഗ് ജോഡികൾക്ക് മുടയിലുള്ള നന്നിലധികം വരികളുള്ള വാക്കുങ്ങൾ പ്രദർശി



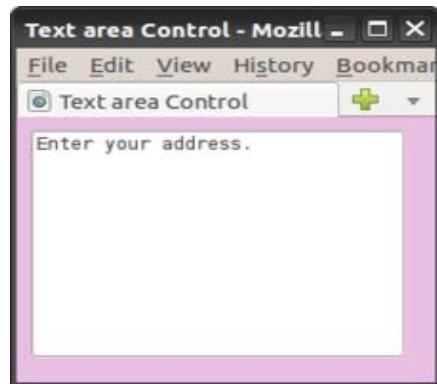
പ്ലിക്കാനുള്ള ഒക്സ്റ്റ് ഏറ്റിയായുടെ വിസ്തൃതി നിർണ്ണയിക്കുന്നത് ഈ ടാഗിന്റെ ആട്ടി ബ്യൂട്ടുകളുടെ വിലയെ ആശയിച്ചിരിക്കും. <TEXTAREA> ടാഗിന്റെ പ്രധാന ആട്ടിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.

1. **Name:** ഈ കൺട്രോളിന്റെ പേര് നൽകുന്നു.
2. **Rows:** ഈ കൺട്രോൾ വരികളുടെ എണ്ണം നിർണ്ണയിക്കുന്നു.
3. **Cols:** ഇതിലെ നിരകളുടെ എണ്ണം നിർണ്ണയിക്കുന്നു. അതായത് ഒരു വരിയിലെ കൂറക്ടറുകളുടെ എണ്ണം.

താഴെ നൽകിയിരിക്കുന്ന കോഡ് ഭാഗം പരിശീലിച്ച് <TEXTAREA> ടാഗിന്റെ ഉപയോഗം തിരിച്ചറിയുക.

```
<FORM Action= "guestbook.php" Method= "post">
<TEXTAREA Rows= "10" Cols= "30" Name= "address">
    Enter your address.
</TEXTAREA>
</FORM>
```

കോഡ് പുർത്തീകരിച്ചതിനുശേഷം വെബ് പേജ് ബൈസാർഡ് തുറക്കുന്നോൾ പാദ്യമാക്കുന്നതിന്റെ പിതൃം 5.26 ത്ര നൽകുന്നു. <TEXTAREA>, </TEXTAREA> എന്നീ ടാഗ് ജോഡികൾക്കിടയിൽ നൽകുന്ന വാക്കുങ്ങൾ കൺട്രോളിൽ പ്രദർശിപ്പിക്കുന്നു. ഉപയോകതാവ് ഏതെങ്കിലും ധാര കൺട്രോളിൽ നൽകുന്നോൾ തന്ത്ര വാക്കുത്തിന് പകരം പ്രദർശിപ്പിക്കുന്നു വെബ് പേജ്.



ചിത്രം 5.26: ഒക്സ്റ്റ് ഏറ്റിയ ഉൾക്കൊള്ളുന്നത്.

#### 5.6.4 <SELECT> ടാഗ്

ഒരു ഉപയോകതാവിന് ഒന്നോ അതിലധികമോ ഓപ്പഷനുകൾ തിരഞ്ഞെടുക്കുവാൻ കഴിയുന്ന കൺട്രോൾ ആണ്. ഈ വിവിധ ഓപ്പഷനുകൾ ദ്രോപ്പഡോണ്ട് ലിസ്റ്റ് രൂപത്തിൽ നൽകുന്നു. അതിനാൽ ഇതിന് ദ്രോപ്പഡോണ്ട് ലിസ്റ്റ് ഭോക്ക് എന്ന പേരിലും അറിയപ്പെടുന്നു. പരിമിതമായ സ്ഥലത്ത് കൂടുതൽ ഓപ്പഷനുകൾ പ്രദർശിപ്പിക്കുവാൻ സെലക്ക് ഷ്ലോക്ക് സഹായിക്കുന്നു. <SELECT> </SELECT> ടാഗിനകത്ത് നൽകുന്ന <OPTION> ടാഗ് ഉപയോഗിച്ചാണ് ലിസ്റ്റിൽ പ്രദർശിപ്പിക്കേണ്ട ഇനങ്ങൾ നൽകുന്നത്.

കണ്ണയ്ക്കൻ ടാഗായ <SELECT> </SELECT> ടാഗിന്റെ പ്രധാനപ്പെട്ട ആട്ടിബ്യൂട്ടുകൾ ചുവടെ ചേർക്കുന്നു.

1. **Name:** ഈ ആട്ടിബ്യൂട്ട് കൺട്രോളിന് പേര് നൽകുന്നു. ഈ പേര് ഇതിലെ ധാര യോക്കാപ്പും സെലവിലേക്ക് അയയ്ക്കുന്നു. ഈ പേര് ഉപയോഗിച്ച് സെലവിൽ കൺട്രോളിന്റെ പേരിനൊപ്പമുള്ള വില തിരിച്ചറിയുകയും സീക്രിക്കറ്റുകയും ചെയ്യുന്നു.

- 2 . **Size:** ഒരു സ്ക്രോളിംഗ് ലിസ്റ്റ് അവതരിപ്പിക്കുന്നതിന് ഉപയോഗിക്കാം. ഈതിന്റെ വിലയുടെ അടിസ്ഥാനത്തിൽ ഇത് ഒരു ലിസ്റ്റ് ബോക്സാണോ അബ്ലൈറ്റിൽ ഒരു ഡ്രോപ്പ്‌ഡോൾ ലിസ്റ്റ് ബോക്സാണോ എന്ന് തീരുമാനിക്കുന്നു. ഈതിന്റെ വില 1 ആണെങ്കിൽ ഡ്രോപ്പ്‌ഡോൾ ലിസ്റ്റ് ബോക്സായിരിക്കും.
- 3 . **Multiple:** ഉപയോകതാവിന് ഒന്നിൽക്കൂടുതൽ ഇനങ്ങൾ തിരഞ്ഞെടുക്കുവാൻ മൂല ആട്ടിബൃംഗം അനുവദിക്കുന്നു.

ഈ നമുക്ക് <OPTION> ടാഗിനെക്കുറിച്ച് ചർച്ച ചെയ്യാം. <SELECT> </SELECT> എന്നീ ടാഗ് ജോഡിക്കിടക്കിയിൽ നൽകുന്ന എം.റി ടാഗ് ആണെന്ന്. <OPTION> ടാഗ് സെലക്ക് ബോക്സിനുകൂടി നൽകിയിരിക്കുന്ന ഓപ്പഷനുകൾ ലിസ്റ്റ് ചെയ്യുന്നു. ഈ ടാഗിന്റെ വിവിധ ആട്ടിബൃംഗകൾ ചുവടെ ചേർക്കുന്നു.

1. **Selected:** തന്ത്ര സെലക്ഷൻ സൂചിപ്പിക്കാൻ ഉപയോഗിക്കുന്നു.
2. **Value:** <OPTION> ടാഗിന് അക്കത്തുള്ള വിലകളിൽ നിന്നും ഉള്ളടക്കത്തോടൊപ്പം വിലിനമായ വില നൽകുവാൻ ഇത് ഉപയോഗിക്കുന്നു. ഈ വില ലഭ്യമല്ലെങ്കിൽ ഇതിന്റെ ഉള്ളടക്കം വിലയായി സ്വീകരിക്കും.

ഉദാഹരണം 5.20 തേ നൽകിയിരിക്കുന്ന HTML കോഡ് ഉപയോഗിച്ച് ഒരു ഡ്രോപ്പ്‌ഡോൾ ലിസ്റ്റ് ബോക്സ് ഉൾപ്പെടുന്ന വെബ്പേജ് നിർമ്മിക്കാം. <SELECT> ടാഗിന്റെ Size ആട്ടിബൃംഗിന് ഒന്ന് (1) എന്ന വില ആണെന്നും കൂടാതെ സെലക്ക് ബോക്സിന് 4 ഓപ്പഷനുകൾ ഉൾണ്ടെന്നും പ്രത്യേകം ശ്രദ്ധിക്കുക.

#### ഉദാഹരണം 5.20: ഒരു ഡ്രോപ്പ്‌ഡോൾ ലിസ്റ്റ് ഉൾപ്പെട്ട HTML ഫോം നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE> Drop down list </TITLE> </HEAD>
<BODY Bgcolor= "#E9BEE5">
    <FORM Action= "guestbook.php" Method= "post">
        <P> Nationality:
        <SELECT Name= "Nationality" Size= "1">
            <OPTION Value= "Indian" selected> Indian
            <OPTION Value= "British"> British
            <OPTION Value= "German"> German
            <OPTION Value= "Srilankan"> Srilankan
        </SELECT>
    </FORM>
</BODY>
</HTML>
```

കോഡോ ബട്ടണിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ ലഭ്യമായ ഡ്രോപ്പ്‌ഡോൾ ലിസ്റ്റിൽ ചിത്രം 5.27-ൽ കാണിക്കുന്നു.



ചിത്രം 5.27: സെലക്ക് ബോക്സ് ഉൾപ്പെട്ട ഫോം



ഉദാഹരണം 5.20 ലെ ഫ്രോഗ്രാം കോഡിൽ `Size` ആട്ടിബ്യൂട്ടിനെ വ്യത്യാസപ്പെടുത്തിക്കൊണ്ട് സൈലക്ക് സോക്സിൽ ഉണ്ടാകുന്ന മാറ്റം നിരീക്ഷിച്ചുകൊണ്ട് ചെയ്യാം ക്കുക.

### 5.6.5 <FIELDSET> ടാഗ് ഉപയോഗിച്ച് ഫോം ഡാറ്റായെ ഒന്നിപ്പിക്കുന്നത്

ചിലപ്പോൾ പരസ്പരബന്ധമുള്ള ഡാറ്റ ഉൾക്കൊള്ളുന്ന കൺട്രോളുകളെ ഒരുമിച്ച് ചേർക്കേണ്ടത് ആവശ്യമായി വന്നേക്കാം. ഈതിനായി <FIELDSET> ടാഗ് ഉപയോഗിക്കാം. ഈ ടാഗ് ഉപയോഗിച്ച് ഫോംമിനുകൂടി പരസ്പര ബന്ധമുള്ള കൺട്രോളുകൾ അടങ്ങിയ ഉപഭാഗങ്ങൾ ഉൾപ്പെടുത്തുവാൻ സാധിക്കും. ഓരോ ഉപഭാഗത്തെയും തിരിച്ചറിയുവാൻ <LEGEND> ടാഗ് ഉപയോഗിക്കാം. <FIELDSET> ഇനങ്ങൾക്ക് ശീർഷകം നൽകുന്ന തിന് <LEGEND> ടാഗ് ഉപയോഗിക്കാം. ഈത് ഒരു കണക്കുനർ ടാഗ് ആണ്.

നമ്മൾ ഈതുവരെ ചർച്ച ചെയ്ത എല്ലാ കൺട്രോളുകളും ഉൾപ്പെടുത്തി പേര്, വയസ്സ്, അധ്യസ്തൻ, വിനോദങ്ങൾ എന്നീ വിവരങ്ങൾ സമർപ്പിക്കുന്നതിനുള്ള HTML കോഡ് ഉദാഹരണം 5.21 ലും അതിനുസ്യതമായ ചിത്രം 5.28 ലും കാണിച്ചിരിക്കുന്നു. ഉദാഹരണം 5.21 ഒരു വിദ്യാർഥിയുടെ വിവരങ്ങൾ സമർപ്പിക്കുന്നതിനുള്ള HTML കോഡ്.

#### ഉദാഹരണം 5.21: ഒരു വിദ്യാർഥിയുടെ വിവരങ്ങൾ സമർപ്പിക്കുന്ന HTML ഫോം

```
<HTML>
<HEAD> <TITLE> FormResume </TITLE> </HEAD>
<BODY Bgcolor= "#E9BEE5">
<CENTER><H3>Enter your details</H3></CENTER>
<FORM Action= "guestbook.php" Method= "get">
    Name:&nbsp;
    <INPUT Type= "text" Name= "first_name" Size= "20"
           Maxlength= "20" Value= "First Name Here"><BR><BR>
    Age:
    <INPUT Type= "text" Name= "age" Size= "3" Maxlength= "3"><BR>
    Sex: &nbsp;&nbsp;&nbsp;
    <INPUT Type= "radio" Name= "sex" Value= "male"> Male
    <INPUT Type= "radio" Name= "sex" Value= "female"> Female
    <FIELDSET>
        <LEGEND>Nationality</LEGEND>
        <SELECT Name= "Nationality" Size= "4">
            <OPTION Value= "Indian" Selected> Indian
            <OPTION Value= "British"> British
            <OPTION Value= "German"> German
            <OPTION Value= "Srilankan"> Srilankan
```



```
</SELECT><BR><BR>
Nativity:
<INPUT Type= "text" Name= "State" Size= "15"><BR><BR>
District:&nbsp;
<INPUT Type= "text" Name= "District" Size= "15">
</FIELDSET><BR>
Hobbies:
<INPUT Type= "checkbox" Name= "Hobby" Value= "games">
Playing Games
<INPUT Type= "checkbox" Name= "Hobby"
Value= "WatchingTV"> Watching TV
<INPUT Type= "checkbox" Name= "Hobby" Value= "Reading">
Reading<BR><BR>
<TEXTAREA Rows= "5" Cols= "25" Name= "address">Address
</TEXTAREA><BR><BR>
<INPUT Type= "submit" Value= "submit">
<INPUT Type= "Reset" Value= "reset">
</FORM>
</BODY>
</HTML>
```

**FormResume - Mozilla Firefox**

Name: First Name Here

Age:

Sex:  Male  Female

Nationality

Indian  
British  
German  
Srilankan

Nativity:

District:

Hobbies:  Playing Games  Watching TV  Reading

Address

submit reset

ചിത്രം 5.28: ഒരു വിദ്യാർഥിയുടെ വിവരങ്ങൾ സ്ഥാപിക്കുന്നതിനുള്ള HTML ഫോം

## 5.6.6 ഹോം സമർപ്പണം

നമ്മൾ ഫോമിലെ സബ്മിറ്റ് ബട്ടണിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ സൈക്രിച്ച് ഡാറ്റ സെർവ്വീസ് ലേക്ക് അയയ്ക്കുന്നു. ഉപയോകതാവിഭാഗം (ക്ലിയർ) കമ്പ്യൂട്ടറിൽ നിന്നും വരുന്ന വിവരങ്ങൾ സൈക്രിച്ച് വൈബ്സൈർവ്വീസിൽ ലഭ്യമായ സെർവ്വീസൈല്സ് ഫ്രോഗ്രാം ഉപയോഗിച്ച് ഫ്രോസൻ ചെയ്യുന്നു.

ഹോം സമർപ്പിക്കുന്നേയാൽ എടുക്കേണ്ട പ്രവർത്തനങ്ങളെ നിർവ്വചിക്കുന്ന അടിബൃക്കാൺ Action സാധാരണയായി സെർവ്വർജേസിലെ URL ആണ് ഈതിന്റെ വില യായി നൽകുന്നത്. ഒരു ഹോം സർവ്വറിലേക്ക് അയയ്ക്കുന്നതിനുള്ള പൊതുവായ മാർഗ്ഗം സബ്മിറ്റ് submit ബട്ടണിൽകൂടി ആണ്. *guestbook.php* ഈതിന്റെ ഉദാഹരണം 5.21 ത്ത് നൽകുന്നു.

<FORM> ടാഗിലെ Method ആട്ടിവ്യൂട്ട് ഫോം സമർപ്പിക്കുന്നതിനുള്ള HTTP മാർഗ്ഗം (get or post) പറയുന്നു.

## നിങ്ങളുടെ പുരോഗതി അറിയാം



1. വെബ്സ്പേജിലൂടെ ഇൻപുട്ട് ഡാറ്റാ നൽകുവാൻ HTML തും provides \_\_\_\_\_ ആണ് ഉപയോഗിക്കുന്നത്.
  2. ഫോമിനുകളും <FORM> ടാഗ്കളും ഡാറ്റാ ഇൻപുട്ട് ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്ന രണ്ട് ടാഗുകളും പേരെഴുതുക.
  3. റേഡിയോ ബട്ടൺ, ചെക്ക്ബോക്സ് എന്നിവ തമിലുള്ള വ്യത്യാസങ്ങൾ എഴുതുക.
  4. ഫോമിനുള്ളിൽ ഡാറ്റാ ഒരുമിപ്പിക്കുവാൻ ഉപയോഗിക്കുന്ന ടാഗ് എത്രാണ്.
  5. <FORM> ടാഗിനുള്ളിൽ ഡാറ്റാ ഇൻപുട്ട് ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്ന ടാഗിന്റെ പേര് എഴുതുക.

## 5.7 HTML 5 നും പൊതു അവലോകനം

HTML റെ പുതുക്കിയ പതിപ്പാണ് HTML 5. HTML 4.01 ന് ശേഷം വേർവ്വ് വെയ്ക്കാൻ കണ്ണസോർഷ്യവും (W3C) വെവ്വേം ഫൈലിൽ ടെക്നോളജി അപ്ലികേഷൻ ടെക്നോളജി വർക്കിങ് ഗ്രൂപ്പും (WHATWG) കൂട്ടായി വികസിപ്പിച്ചട്ടത്താണ് HTML5. ഈ തിരികെ വീഡിയോ പ്രോ ബാക്ക്, ഡ്യാഗ് ആർട്ടിക്കൾ ഫോറം തുടങ്ങിയവ കൂടിച്ചേര്ത്തിട്ടാണ്.

പ്രമുഖ ബൈൽസറ്റുകളായ ഗുഗ്ലിൽ ക്രോം, മോസില്ലു ഫയർഫോക്സ്, ആസ്റ്റ്രിൽ സഹാരി ഓപ്പോറ്റു ഇൻഡ്രോൺറ്റ് എക്സ്പ്രോസ് എന്നിവയയുടെ ഏറ്റവും പുതിയ പതിപ്പുകൾ HTML5 നെ പിന്തുണയ്ക്കുന്നു. ഫൈ ഫോണുകൾ, ഫൈപാഡ്യൂകൾ, ആൻഡ്രോഡ്യൂഡ്യീസ് ഫോണുകൾ തുടങ്ങിയവയിലുള്ള മൊബൈൽ ബൈൽസറ്റുകളും HTML5 നെ പിന്തുണയ്ക്കുന്നു. ഇപ്പോഴുള്ള ബൈൽസറ്റുകളിൽ പഴയതിനെന്നയും പ്രവർത്തിപ്പിക്കാവുന്ന രീതിയിൽ ഉൾക്കൊ

ഇളിച്ചാണ് HTML 5 രൂപകൽപ്പന ചെയ്തിരിക്കുന്നത്. HTML5 എഴു ലോഗോ ചിത്രം 5.29-ൽ HTML 5 എഴു ലോഗോ നൽകിയിരിക്കുന്നു. ചിത്രം 5.29 HTML 5 എഴു ലോഗോ.



Fig. 5.29: HTML5 logo

HTML5 ലെ പതിചയപ്പെടുത്തുന്ന ഏതാനും പുതിയ ടാഗുകൾ താഴെ നൽകുന്നു.



1. <VIDEO> യും <AUDIO> HTML ഡോക്യുമെന്റീകളിൽ ദൃശ്യ ശാവ്യ മാധ്യമ അംഗൾ വളരെ എളുപ്പം ഉൾക്കൊള്ളുകയോണ് ഈ ടാഗുകൾ സഹകര്യമൊരുക്കുന്നു.
2. <CANVAS> ചിത്രം ഗ്രാഫിക്സ് എന്നിവ വരയ്ക്കുക. ചാർട്ടുകളും ഗ്രാഫുകളും തയാറാക്കുക, നമുക്കാവശ്യമുള്ള രീതിയിൽ ഗ്രാഫിക്സ് മാറ്റിയെടുക്കുക തുടങ്ങിയവയ്ക്കുള്ള ലളിതവും ശക്തവുമായ മാർഗമാണ് ഈ ടാഗ് നൽകുന്നത്.
3. <HEADER>, <FOOTER> ഒരു ഡോക്യുമെന്റീയോ സെക്ഷൻസേയോ ഫോയർ വ്യക്തമാകുവാൻ <HEADER> ടാഗും ഫൂട്ടർ (അടിക്കോറ്റപ്പ്) വ്യക്തമാക്കുവാൻ <FOOTER> ടാഗും ഉപയോഗിക്കുന്നു. ഒരു ഫൂട്ടറിനുള്ളിൽ ഡോക്യുമെന്റീയോ രചയിതാവ് പകർപ്പുവകാശത്തെക്കുറിച്ചുള്ള വിവര അംഗൾ, ബന്ധപ്പെടുവാനുള്ള വിലാസം തുടങ്ങിയവ സൂചിപ്പിക്കുന്നു. നമുക്ക് ഒരേ ഡോക്യുമെന്റീയോ തന്നെ ധാരാളം <HEADER>, <FOOTER> ഇനങ്ങൾ ഉൾക്കൊള്ളിക്കാനാകും. ഒരു വെബ്പോജിനുള്ളിൽ ആർട്ടിക്കലുകളും സെക്ഷൻസുകളും നിർമ്മിക്കുവാൻ ഈ ടാഗുകൾ ഉപയോഗിക്കുന്നു.
4. <ARTICLE>, <SECTION> ആർട്ടിക്കലുകൾ സ്വത്തെവും വ്യത്യസ്തവുമായ വാർത്താക്കുറിപ്പ് അല്ലെങ്കിൽ സ്ക്രോൾ പോസ്റ്റ് ആണ്. ഒരു പേജിനകത്ത് വ്യത്യസ്തങ്ങളായ വിഷയ മേഖലകളാക്കുകയോ ആർട്ടിക്കലിനെ പലഭാഗങ്ങളാക്കുകയോ ചെയ്യുന്നതിന് സെക്ഷൻ ഉപയോഗിക്കുന്നു. <ARTICLE>, <SECTION> എന്നി ടാഗുകൾ ശരിയായ രീതിയിൽ ഉപയോഗിച്ചാൽ അത് സെർച്ച് എൻജിനിൽ ഈ പേജ് ലഭ്യതയ്ക്കുള്ള സാധ്യത വർദ്ധിക്കും.
5. <OUTPUT>: ഒരു സ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് നടത്തുന്ന കമ്പ്യൂട്ട്കല്യൂട്ടർ ഫലം ലഭ്യമാക്കുന്നതിന് ഈ ടാഗ് ഉപയോഗിക്കുന്നു.
6. <DETAILS>: ഫോയിൽ കൂടിക്ക ചെയ്യുന്നോൾ എക്സ്പാൻഡ് ചെയ്യുന്നതിനും കൊള്ളാപ്പൾ ചെയ്യുന്നതിനും വേണ്ട വാക്കുങ്ങളുടെ ഭാഗം നിർണ്ണയിക്കുവാൻ ഈ ടാഗ് ഉപയോഗിക്കുന്നു.
7. <FIGURE> and <FIGCAPTION>: ചിത്രങ്ങൾ ഉൾക്കൊള്ളിക്കുവാൻ <FIGURE> ടാഗും അതിന്റെ ശീർഷകം <FIGCAPTION> ടാഗും ഉപയോഗിക്കുന്നു. <FIGCAPTION> ടാഗ് <FIGURE> ടാഗിനുള്ളിലാണ് നൽകുന്നത്.
8. <PROGRESS> and <METER>: <PROGRESS>, <METER> എന്നി ടാഗുകൾ ഒരേ പോലെയാണ്. ഒരു ജോലിയുടെ പൂര്ണാഗതിയെ സൂചിപ്പിക്കാൻ <PROGRESS> ടാഗ് ഉപയോഗിക്കുന്നു. ഇത് ഫയൽ അപ്ലോഡ് പോലുള്ള പ്രവർത്തനങ്ങളുടെ പൂര്ണാഗതി കാണിക്കുന്നതിന് ഉപയോഗിക്കുന്നു. എന്നാൽ <METER> ടാഗ് ഒരു പരിധിയിലുള്ള അളവ് മാത്രം നൽകുന്നതിന് ഉപയോഗിക്കുന്നു (ഉദാഹരണത്തിന് ഹാർഡ് ഡിസ്കിന്റെ ഉപയോഗം നൽകുന്നതിന്).



## നമുക്ക് സംഗ്രഹിക്കാം

ഈ അധ്യായത്തിലുടെ HTML റെംഡർ മെച്ചപ്പെട്ട സവിശേഷത നാം പരിച്ചു കഴിഞ്ഞ് വാക്കു അശ്ര ആകർഷകമാക്കുന്നതിനുവേണ്ട വിവിധയിനം ലിറ്റസൂക്ഷൾ നമ്മൾ ചർച്ച ചെയ്തു. ഒരു വെബ് പേജിൽ ആധിയോ, വീഡിയോ, ചിത്രങ്ങൾ തുടങ്ങിയവ ഉൾപ്പെടുത്തി അതിൻ്റെ മനോഹാരിത എങ്ങനെ വർധിപ്പിക്കാം എന്ന് നാം കണ്ടുകഴിഞ്ഞു. കൂടാതെ ഒരു വെബ് പേജിൽ ഉൾപ്പെടുത്തുന്ന വിവിധതരം ഫോറ്മേറ്റ് ലിങ്കുകളെയും അവയുടെ പ്രാധാന്യത്തെക്കുറിച്ചും നാം മനസിലാക്കി. പല വിവരങ്ങളും വ്യക്തമായും എളുപ്പത്തിലും മനസിലാക്കുവാൻ കഴിയുന്ന വിധത്തിൽ രേഖപ്പെടുത്തുവാൻ ദേബിൽ ഉപയോഗിച്ചു. HTML ദേബിൽ നിർമ്മാണത്തിനാവശ്യമായ `<TABLE>`, `<TR>`, `<TH>`, `<TD>` എന്നീ ടാഗുകളും അവയുടെ ആട്ടിബൃട്ടുകളെക്കുറിച്ചും നാം ചർച്ച ചെയ്തു. ദേബിളിലെ വരികളും നിരകളും ആവശ്യാനുസരണം ഒന്നിച്ച് ചേർക്കുവാൻ യഥാക്രമം Rowspan, Colspan എന്നീ ആട്ടിബൃട്ടുകളെക്കുറിച്ചും പ്രാധാന്യം തിരിച്ച് ഒരു ബ്രൗസർ വിന്റോദയ പല ഭാഗങ്ങളാക്കി തിരിച്ച് ഒന്നിൽ കൂടുതൽ വെബ് ഡോക്യുമെന്റുകൾ ഉൾപ്പെടുത്താൻ ഒരേ സമയം `<FRAMESET>` ഉപയോഗിച്ച് നാം ഒരു വിന്റോദയയുടെ ഒരേ ഭാഗത്ത് വിവിധ വെബ് ഡോക്യുമെന്റുകൾ പ്രദർശിപ്പിക്കുന്നതിനുവേണ്ട ഒന്നെല്ലാം ശ്രദ്ധയിൽ Target ആട്ടിബൃട്ടുകളും നാം പരിശീലിച്ചു. ഒരു വെബ് ഡോക്യുമെന്റീൽ ഉപയോക്താവിൽ നിന്നും വിലകൾ സ്വീകരിച്ച് സെർവ്വറിലേക്ക് അയയ്ക്കുവാൻവേണ്ട ഫോം നിർമ്മാണം നമ്മൾ പരിച്ചു. വിവിധ തരത്തിൽ ഡാറ്റ സ്വീകരിക്കുവാൻ വേണ്ടിയുള്ള കൺട്രോളും ടെക്നോളജിക്സ് സേമാക്സ്, പാസ്വോർഡ്, റോഡിയോ ബട്ടൺ, ലിറ്റ് ഡോക്സ്, പാസ്വോർഡ്, റോഡിയോ ബട്ടൺ, ലിറ്റ് ഡോക്സ് ബട്ടണുകൾ എന്നിവ നാം പരിചയപ്പെട്ടു. ഓയിൽ സെസഡ് സ്ക്രിപ്റ്റ്, സർവർസെസഡ് സ്ക്രിപ്റ്റ് എന്നിവയുടെ കുറച്ച് കാര്യങ്ങൾ സൂചിപ്പിച്ചു. ഈ വിഷയങ്ങൾ ചർച്ചചെയ്യുന്ന അടുത്ത അധ്യായങ്ങളിലേക്കുള്ള ചവിട്ടുപടിയായി ഈ അധ്യായത്തെ കരുതാം.



## നമുക്ക് പരിശീലിക്കാം

1. താഴെ കൊടുത്തിരിക്കുന്ന വിശദാംശങ്ങളും സവിശേഷതകളും ഉൾക്കൊള്ളിച്ച് കേരളത്തിലെ ഏതെങ്കിലും ഒരു ജില്ലയുടെ വെബ് പേജ് തയാറാക്കുന്നതു HTML കോഡ് എഴുതുക.
  - ടെക്നോളജിക്സ് ടാഗുകളും അവയുടെ ആട്ടിബൃട്ടുകളും ഉപയോഗിച്ച് ഒരു ശീർഷകവും അതിനു താഴെ മുന്ന് വാചകത്തിൽ കുറയാതെയുള്ള ഒരു വണ്ണഡിക്കയിൽ ജില്ലയെക്കുറിച്ചുള്ള വിവരങ്ങളും നൽകുക.
  - ജില്ലയിലെ പ്രധാനപ്പെട്ട വിനോദസഞ്ചാര സ്ഥലങ്ങളുടെ ലിറ്റ്.
2. നിങ്ങളുടെ സ്കൂളിന്റെ വെബ്‌പേജ് രൂപകൽപ്പന ചെയ്യുവാനാവശ്യമായ HTML കോഡ് താഴെപ്പറയുന്ന വിവരങ്ങൾ/സവിശേഷതകൾ ഉൾപ്പെടുത്തി എഴുതുക.
  - ടെക്നോളജിക്സ് ടാഗുകളും അവയുടെ ആട്ടിബൃട്ടുകളും ഉപയോഗിച്ച്
  - ഒരു ശീർഷകവും അതിനുബന്ധം മുന്ന് വാചകങ്ങളിൽ കുറയാതെ നിങ്ങളുടെ സ്കൂളിന്റെ പ്രത്യേകതകൾ എടുത്തുപറയുന്ന വണ്ണഡിക്കയും.

- നിങ്ങളുടെ സ്ക്രൂളിലെ ഏതെങ്കിലും പഠനത്തോടൊപ്പമുള്ള പ്രവർത്തനങ്ങൾ ഉണ്ട്. ഉദാഹരണം : സ്ക്രൂട്ട്, ഗൈൾ, എൻ.എസ്.എസ്.
3. താഴെപ്പറയുന്ന വിവരങ്ങൾ വ്യക്തമാക്കുന്ന വെബ്പോജിന്റെ HTML കോഡ് എഴുതുക.

## Components of a Computer

### • Hardware

1. I/O Devices
2. RAM
3. Hard Disk & DVD Drive

### • Software

1. Operating System
2. Application Programs

4. കേരളത്തിന്റെ ദേശീയോസ്യവമായ ഓൺത്തിന്റെ എടുത്തുപറയത്തക്ക പ്രത്യേകതകൾ വർണ്ണിക്കുന്ന വെബ്പ് പോജിനുവേണ്ടി താഴെപ്പറയുന്ന സവിശേഷതകൾ ഉൾപ്പെടുത്താനുള്ള HTML കോഡ് എഴുതുക.
- ആകർഷകമായ ഫോൺഡിലുള്ള ശീർഷകം.
  - വള്ളംകളിയുടെ ചിത്രം പോജിന്റെ പശ്ചാത്തലമാക്കുക.
  - ഓൺത്തിനോട് അനുബന്ധിച്ച് നടത്തപ്പെടുന്ന പരമ്പരാഗത ആചാരങ്ങളായ അത്തപ്പുകളും, തുനിതുള്ളൽ, വള്ളംകളി, കുമ്മാട്ടികളും, പുലികളി എന്നിവ ഏതെങ്കിലും രണ്ടിലേക്ക് ഇന്ത്യൻ ലിക്ക് നൽകുക.
5. നിങ്ങളുടെ സ്ക്രൂളിന്റെ ഹൈഡ്രസക്കറ്ററി വിഭാഗങ്ങളുടെ വിവരങ്ങൾ അടങ്കുന്ന രണ്ട് വെബ്പ് പോജുകൾ നിർമ്മിക്കുക. അടുത്ത വെബ്പോജ് നിർമ്മിച്ച് തിരഞ്ഞെടുപ്പുകൾ ആമുഖവും രണ്ട് ലിങ്കുകളുമാണ് (ഹൈഡ്രസക്കറ്ററി) നൽകേണ്ടത്. മറ്റാരു വെബ്പോജ് നിർമ്മിച്ച് ഉപയോക്താവ് ലിങ്കിൽ ക്ലിക്ക് ചെയ്യുമ്പോൾ ത്യാക്കുകയും അവയുടെ വെബ്പോജുകൾ രണ്ടാമത്തെ ഫോറ്മുലയിൽ ലഭ്യമാക്കുന്നതിന് വേണ്ട HTML കോഡ് തയാറാക്കുക.
6. ചുവരെ ചേർക്കുന്ന ടേബിള്ടും അതിന് താഴെ കൊടുത്തിരിക്കുന്ന കേരള പോലീസിന്റെ വെബ്പോസറ്റിലേക്കുള്ള എക്സ്റ്റോണൽ ലിക്ക് എന്നിവ പ്രദർശിപ്പിക്കുന്ന വെബ്പ് പോജിന് ആവശ്യമായ HTML കോഡും തയാറാക്കുക.

## Road Accidents in Kerala during 2012 - 2014

Year	Total Number of		
	Cases	Persons Killed	Persons Injured
2012	36174	4286	41915
2013	35215	4258	40346
2014	36282	4049	41096

Data Source: [www.keralapolice.org](http://www.keralapolice.org)

7. ചുവടെ ചേർക്കുന്ന റീതിയിലുള്ള ഒരു അപേക്ഷ പോം തയാറാക്കുന്നതിനാവശ്യമായ HTML കോഡ് എഴുതുക.

**APPLICATION FOR THE BEST STUDENT AWARD**

Name:	<input type="text"/>	Sex: Male <input type="radio"/> Female <input type="radio"/>							
Class & Division:	<input style="width: 150px; height: 25px; border: 1px solid black; padding: 2px; margin-bottom: 5px;" type="button" value="Select"/>								
Total Grade Point in Class XI:	<input type="text"/>								
Average Grade Point in Termly Exams in Class XII:	<input type="text"/>								
Cocurricular Activities:									
NCC	<input type="checkbox"/>	NSS	<input type="checkbox"/>	Sports	<input type="checkbox"/>	Arts	<input type="checkbox"/>	Literary	<input type="checkbox"/>
Other Achievements:			<input style="height: 100px; width: 100%;" type="text"/>						

### നമുക്ക് വിലയിരുത്താം |||

1. ചുവടെ ചേർക്കുന്നവയിൽ തെറ്റ് കണ്ടുപിടിച്ച് തിരുത്തി എഴുതുക.
  - a. <UL Type = "i" Start = 3>
  - b. <IMG Src = "Myschool.jpg" Size = "50" >
  - c. <HTML>
 

```
<HEAD><TITLE><HEAD></TITLE>
<BODY> this is the body of the HTML
document</BODY>
```

</HTML>
2. രോഹിത് HTML ഉപയോഗിച്ച് നിർമ്മിച്ച ഫേബിളിൽ ബോർഡ് കാണുവാൻ സാധിക്കുന്നില്ല. ഇതിന്റെ കാരണം എന്തായിരിക്കും?
3. ലിങ്ക് ചെയ്ത പേജ് ഒരു ഫ്രെയിമിനുള്ളിൽ പ്രദർശിപ്പിക്കുവാൻ <A> ടാഗിന്റെ \_\_\_\_\_ ആട്ടിബ്യൂട്ട് ഉപയോഗിക്കുന്നു.
4. നിങ്ങളുടെ ആത്മാർമ്മ സുഹൃത്തുകളുടെ ഒരു ലിസ്റ്റ് തയാറാക്കുവാൻ നിങ്ങളുടെ കമ്പ്യൂട്ടർ അധ്യാപകൻ ആവശ്യപ്പെടുന്നു എന്ന് കരുതുക. നിങ്ങൾ എത്ര ടാഗിനാം പരിഗണന നൽകുക. ഇതിനുവേണ്ട HTML കോഡ് ഭാഗം തയാറാക്കുക.
5. <OL> ടാഗിലെ type ആട്ടിബ്യൂട്ടിന് നൽകാവുന്ന വിലകളുടെ പേര് എഴുതുക.
6. <UL> ടാഗിന്റെ ആട്ടിബ്യൂട്ടുകൾ എഴുതുക.

7. ആഗലേയ ഭാഷയിലെ വലിയ അക്ഷരങ്ങൾ ഉപയോഗിച്ച് ക്രമനവരോട് കൂടിയ ലിംഗ് നിങ്ങൾ എങ്ങനെ നിർമ്മിക്കും.
  8. ആഗലേയ ഭാഷയിലെ ചെറിയ അക്ഷരങ്ങൾ ഒരു ലിംഗിലെ ഇനങ്ങളുടെ ക്രമന പര ആയി പ്രവർശിപ്പിക്കണം എന്ന് കരുതുക. എങ്ങനെ ഇത് സാധ്യമാക്കാം?
  9. ക്രമനവർ 6 മുതൽ ആരംഭിക്കുന്ന ഒരു ലിംഗ് നമുക്ക് എങ്ങനെ നിർമ്മിക്കാം?
  10. താഴെ കൊടുത്തിരിക്കുന്ന വിവരങ്ങൾ ഉൾക്കൊള്ളിച്ചു വെബ് പ്രോജക്റ്റിനു വേണ്ട <HTML>കോഡ് എഴുതുക.

**ABC Pvt. Ltd.**  
**Kerala**

1. Health Care
2. Baby Products
  - Toys
  - Dress
3. Ladies Wear
  - Kurthas
  - Jeans

11. വരുണ്ണ ഒരു വെബ് പേജ് നിർമ്മിക്കുന്നു. അതിൽ 'sample' എന്ന വാക്ക് ഹൈപ്പർ ലിങ്കായി പ്രദർശിപ്പിച്ച് /home/scert/Desktop-ൽ സംഭരിക്കുന്ന sample.html എന്ന ഫയലിലേക്ക് ലിങ്ക് നൽകുവാനുള്ള HTML കോഡ് ഭാഗം എഴുതുക.
  12. Noshade എന്ന ആട്ടിബ്യൂട്ടുള്ള ടാഗിന്റെ പേര് എഴുതുക.
  13. സുനിലിന് അധ്യാത്മക സ്വന്തം ആവശ്യങ്ങൾക്കായി ഒരു ഇ-മെയിൽ ലിങ്ക് ഉൾപ്പെടുത്തുന്ന വെബ്‌പേജ് നിർമ്മിക്കണം. നിങ്ങൾക്ക് ഈ-മെയിൽ ലിക്കിനുവേണ്ട പ്രോട്ടോക്കോൾ നിർദ്ദേശിക്കാമോ?
  14. സഹീറിന് അധ്യാത്മക വെബ്‌പേജിൽ നിന്നും www.gmail.com എന്ന വെബ്‌സൈറ്റിലേക്ക് ബന്ധപ്പിക്കണം. ഇതിനാവശ്യമായ ടാഗും അവയുടെ ആട്ടിബ്യൂട്ടുകളും എഴുതുക.
  15. HTML ലഭ്യമായ രണ്ട് തരം ഹൈപ്പർ ലിങ്കുകളുടെ സഭാവ സവിശേഷതകൾ പ്രതിപാറിക്കുക.
  16. Cellspacing, Cellpadding ഇവയുടെ വ്യത്യാസങ്ങൾ എഴുതുക.
  17. Text, Textarea എന്നീ കൺട്രോളുകളുടെ വ്യത്യാസങ്ങൾ എഴുതുക.
  18. <FORM> ടാഗിന്റെ പ്രധാന ആട്ടിബ്യൂട്ടുകൾ Action നും ഉം ആകുന്നു.
  19. തെറ്റോ ശരിയോ എന്ന് പറയുക.
    - a. <TABLE> ടാഗിലെ Align ആട്ടിബ്യൂട്ടിന്റെ തന്ത്തവില center ആകുന്നു.
    - b. <FRAME> ഒരു ക്ലൈഞ്ചർ ടാഗ് ആകുന്നു.
    - c. ഉപയോകതാവ് ഒരു നിശ്ചിത ഫ്രെയിമിലെ ബോർഡിന്റെ വലുപ്പ് വ്യത്യാസം വരുത്തുന്നത് തയ്യാറാൻ സ്ക്രോളിങ്ങ് ഉപയോഗിക്കുന്നു.

20. ഏത് ടാഗാംഗ് ഒരു ബേഹസർ വിന്റോദയ വിവിധ ഭാഗങ്ങളാക്കി വിഭജിക്കുന്നത്?
21. <FRAME> ടാഗിന്റെ പ്രധാന ആട്ടിബൃംഗകൾ എഴുതുക.
22. ഒരു വെബ്പോജിനെ ലംബമായ രീഡ് ഭാഗങ്ങളാക്കി തിരിക്കുന്ന ഫ്രെയിം നിർമ്മിക്കുക. നിങ്ങൾക്ക് ഇഷ്ടപ്പെട്ട ഫൂട്ട്വോൾ കളിക്കാരുടെ പേര് ഇടതുഭാഗത്തും, സെലക്ക് ചെയ്യുന്ന കളിക്കാരനെക്കുറിച്ചുള്ള ലാല്പു വിവരണം വലതുഭാഗത്തും പ്രദർശിപ്പിക്കുക.
23. ചുവടെ ചേർക്കുന്ന HTML കോഡ് പരിശോധിച്ചതിന് ശേഷം അതിന്റെ ഐട്ട്‌പുക് എന്നാകും എന്ന് എഴുതുക.

```
<HTML>
<HEAD><TITLE>A simple table</TITLE></HEAD>
<BODY>
<TABLE border="1" Cellspacing= "1" Cellpadding= "10">
  <TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD></TR>
  <TR><TD> 4 </TD><TD> 5 </TD><TD> 6 </TD></TR>
  <TR><TD> 7 </TD><TD> 8 </TD><TD> 9 </TD></TR>
</TABLE>
</BODY>
</HTML>
```

24. താഴെ കൊടുത്തിരിക്കുന്ന ടേബിൾ പ്രദർശിപ്പിക്കുവാൻ വേണ്ട HTML കോഡ് എഴുതുക.
25. നിങ്ങളുടെ ഇമെയിൽ, വിലാസം, ഹോം നമ്പർ, പാസ്വോഡ് തുടങ്ങിയവ സീക്രിക്കറ്റുവാൻ വേണ്ട HTML കോഡ് എഴുതുക.
26. നിങ്ങൾക്ക് ഇഷ്ടപ്പെട്ട മുന്ന് ബൈക്കുകളുടെ പേര് അടങ്കുന്ന ലിസ്റ്റ് ഒരു ഫ്രെയിമിലും, അതിന്റെ ചിത്രവും, വിവരങ്ങളും അടുത്ത ഫ്രെയിമിൽ ലിങ്ക് ചെയ്തും പ്രദർശിപ്പിക്കുന്നതിനുള്ള HTML കോഡ് എഴുതുക.
27. ചുവടെ ചേർക്കുന്ന HTML കോഡ് ഉപയോഗിച്ച് രാജ് നിർമ്മിച്ച വെബ് പേജിൽ പട്ടിക രൂപത്തിൽ കാണുവാൻ സാധ്യക്കുന്നില്ല. ഇതിന്റെ കാരണം കണ്ണത്തി ശരിയായ രീതിയിൽ എഴുതുക.

```
<HTML>
<HEAD><TITLE> My Page </TITLE></HEAD>
<BODY>
<TABLE><TR><TH>Roll No. </TH><TH> Name </TH></TR>
<TR><TD>1 </TD><TD> Huda </TD></TR>
<TR><TD>2 </TD><TD>Bincy</TD></TR>
</TABLE>
</BODY>
</HTML>
```

1	2	3
4	5	6
7		



## 6

# ജാവാസ്ക്രിപ്ട് ഉപയോഗിച്ചുള്ള ക്ലീറ്റ് സെസഡ് സ്ക്രിപ്റ്റിംഗ്

## പ്രധാന പഠനമേഖലകൾ

ഈ അധ്യാത്മികൾ പുർത്തീകരണത്തിനുശേഷം പഠിതാവ്

- ക്ലീറ്റ് വശ്വരതയും സെർവർ വശ്വരതയും സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ ഉപയോഗം തിരിച്ചിരിയുന്നു.
- ക്ലീ സെസസ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ ആവശ്യകത വിശദിക്കിക്കുന്നു.
- ക്ലീ സെസസ് സ്ക്രിപ്റ്റിംഗ് ഭാഷയായ ജാവാസ്ക്രിപ്റ്റിന്റെ പ്രാധാന്യം തിരിച്ചിരിയുന്നു.
- വെബ്‌പേജിൽ ജാവാസ്ക്രിപ്ട് ഫംഗഷൻ ഉപയോഗിക്കുന്നു
- ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ ധാരാ തരങ്ങൾ വിശദിക്കിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിൽ ശരിയായ വേരിയബിളുകൾ ഉപയോഗിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിൽ അനുയോജ്യമായ കൺട്രാൾ സ്റ്റ്രക്ചറുകൾ ഉപയോഗിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിൽ അനുയോജ്യമായ അന്തർ നിർബിത ഫംഗഷനുകൾ ഉപയോഗിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിലെ ഡോക്യുമെന്റ് ഘടകങ്ങൾ ഉപയോഗിക്കുന്ന രീതി വിവരിക്കുന്നു.
- ടെക്സ്റ്റ് ബോക്സുകളിലേയും കോംബോ ബോക്സുകളിലേയും വിലകൾ കൈകൊംബ ചെയ്യുന്നതിനുള്ള ജാവാസ്ക്രിപ്ട് ഫംഗഷനുകൾ നിർവ്വിക്കുന്നു.

ഒരു അധ്യാത്മികളിൽ ടെക്സ്റ്റുകളും ഗ്രാഫിക്സുകളും ഉൾപ്പെടുത്തി വിവിധ തരം വെബ് പ്രോഗ്രാഫുകൾ നിർമ്മിക്കുവാൻ നാം പഠിച്ചു. ഈ ഇൻഡസ്ട്രിയിൽ യുഗത്തിൽ നമ്മളിൽ പലരും വിവിധ ആവശ്യങ്ങൾക്കായി അനവധി വെബ്‌സെസ്റ്റുകൾ സൗഖ്യികരാനുണ്ട്. നമുക്ക് പരിചിതമല്ലാത്ത അനേകം സവിശേഷതകൾ ഇതരം വെബ് പ്രോഗ്രാഫുകളിൽ അടങ്കിയിരിക്കുന്നു. അതുകൊണ്ടു തന്നെ ഇതരം വെബ് പ്രോഗ്രാഫുകൾ നിർമ്മിക്കുന്നതിനായി സ്ക്രിപ്റ്റിംഗ് ഭാഷകളിലൂള്ള അറിവ് ആവശ്യമാണ്. ക്ലീറ്റ് സെസഡ് സെസഡിലും വിവിധയിനം സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ ഉപയോഗിക്കുന്നുണ്ട്. ക്ലീറ്റ് സെസഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകളായ ജാവാസ്ക്രിപ്റ്റ്, വി.ബി സ്ക്രിപ്റ്റ് എന്നിവയിൽ ജാവാസ്ക്രിപ്റ്റാണ് കൂടുതലായി ഉപയോഗിക്കുന്നത്. കാരണം വി.ബി സ്ക്രിപ്റ്റിനെ അപേക്ഷിച്ച് ജാവാസ്ക്രിപ്റ്റിനെ യാണ് കൂടുതൽ ബേദസറുകളും പിന്തുണയ്ക്കുന്നത്. ഈ ഇൻഡസ്ട്രിയിൽ അനേകം ആളുകൾ വ്യത്യസ്ത ബേദസറുകൾ ഉപയോഗിക്കുന്ന തിനാൽ വെബ് പ്രോഗ്രാഫുകൾ കഴിയാവുന്നതെ എല്ലാ ബേദസറുകൾക്കും അനുയോജ്യമായ രീതിയിൽ നിർമ്മിക്കേണ്ടതുണ്ട്. ഒരു വെബ് പ്രോഗ്രാഫിൽ ജാവാസ്ക്രിപ്റ്റ് എങ്കിനെ ഉപയോഗിക്കാമെന്ന് ഈ അധ്യാത്മികളിൽ മനസ്സിലാക്കാം. നമുക്ക് C++ പരിചയമുള്ളതിനാൽ ജാവാസ്ക്രിപ്റ്റ് മനസ്സിലാക്കുവാൻ വളരെ എളുപ്പമാണ്. കാരണം C++ എൻ അതേ വാക്കു ഘടനയാണ് ജാവാസ്ക്രിപ്റ്റ് പിന്തുടരുന്നത്.



 നേര്യസ്കൈപ്പ് റവോൾവറിനുകൂടി ഒരു വാർഷിക അംഗം ജാവാസ്ക്രിപ്ട് വികസിപ്പിച്ചു. ജാവാസ്ക്രിപ്പറ്റിന്റെ മുദ്രയാമാം 'ജാവ' എന്നായിരുന്നു. 1995-ൽ നേര്യസ്കൈപ്പ് റവോൾവർ പതിപ്പ് 2.0 തീരിച്ച ഉമ്മൻസ്ക്രാഫ്റ്റിയശോർ ഒരിഞ്ഞ പേര് 'ജാവാസ്ക്രിപ്പറ്റ്' എന്നാക്കിമാറ്റി. മുദ്രയാമാം നേര്യസ്കൈപ്പ് റവോൾവർ മാത്രമേ ജാവാസ്ക്രിപ്പറ്റിന് പിരുമ്പി ഏതുകിഴിരുന്നുള്ളൂ. എന്നാൽ ജാവാസ്ക്രിപ്പറ്റിന്റെ വിപുലമായ പ്രചാരം കാരണം, ഇന്ത്രൻറ്റ് ഫോൺ പോലെയുള്ള ജാവാസ്ക്രിപ്പറ്റിന് പിരുമ്പി ഏതുകി. ഒഴശാർ, ലോകത്തിൽ വികി റവോൾവറുകളും ജാവാസ്ക്രിപ്പറ്റിനെ പിരുമ്പിക്കുന്നു.

 ഭവന്നിലൻ ഇച്ച്

## 6.1 ജാവസ്ക്രിപ്പറ്റ് ആരംഭിക്കാം

നാലാമത്തെ അധ്യായമായ വെബ് ടെക്നോളജിയിൽ കൈപ്പിൽ സെസഡിലും സെർവർ സെസഡിലും ഉള്ള സ്ക്രിപ്പറ്റിംഗ് ഭാഷകളുടെ ഉപയോഗം നാം ചർച്ചചെയ്തു. കൈപ്പിൽ സെസഡിലെ ധാരായുടെ സാധൂത ഉറപ്പുവരുത്തുന്നതിനായി കൈപ്പിൽ സെസഡിപ്പറ്റിംഗ് ഭാഷകൾ ഉപയോഗിക്കുന്നു. ഈത് സെർവർിലെ നേര്യ വർക്ക് ട്രാഫിക്കും ജോലി ഭാരവും കുറയ്ക്കുന്നു. സെർവർ സെസഡി സ്ക്രിപ്പറ്റിംഗ് ഭാഷകൾ സെർവർിൽ പ്രവർത്തിക്കുന്നു. ഇതിലുടെ നിർമ്മിക്കപ്പെട്ട വെബ് പേജ് കൈപ്പിൽ ബേഹസറിലേക്ക് തിരിച്ചയക്കുന്നു. ധാരാബേസിന്റെ രൂപത്തിൽ വലിയ അളവ് ധാരാ സെർവർിൽ സംഭരിക്കുന്നു. അതിനാൽ സെർവർ സെസഡിലെ സ്ക്രിപ്പറ്റിംഗ് ഭാഷ വ്യത്യസ്ത ആവശ്യങ്ങൾക്ക് വേണ്ടി ഈ ധാരാബേസുമായി സംവദിക്കേണ്ടതായി വരും. എന്നാൽ ഒരു കൈപ്പിൽ സെസഡി സ്ക്രിപ്പറ്റിംഗ് ഭാഷ സെർവർ ധാരാബേസുമായി നേരിട്ട് ബന്ധപ്പെടുന്നില്ല. അതുകൊണ്ടു തന്നെ കൈപ്പിൽ സെസഡിലും സെർവർ സെസഡിലും ഉപയോഗിക്കുന്ന ഭാഷകളും കമാൻഡുകളും വ്യത്യസ്തങ്ങളാണ്.

ഈ അധ്യായത്തിൽ കൈപ്പിൽ വശം സ്ക്രിപ്പറ്റിംഗ് ഭാഷ എന്ന നിലയിൽ നമുക്ക് ജാവാസ്ക്രിപ്പറ്റിന്റെ അടിസ്ഥാന വിവരങ്ങൾ മനസിലാക്കാം. ജാവാസ്ക്രിപ്പറ്റ് ഉപയോഗിച്ച് HTML പേജിലെ വിവിധ ഭാഗങ്ങളിൽ പ്രോഗ്രാമുകൾ ഉൾപ്പെടുത്തുവാൻ കഴിയും. ഇതിനായി <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നു.

### <SCRIPT> Tag

ഒരു HTML പേജിൽ സ്ക്രിപ്പറ്റിംഗ് കോഡ് ഉൾപ്പെടുത്താൻ <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നു. <SCRIPT> ടാഗിലെ 'language' ആട്ടിബ്യൂക്ക് ഉപയോഗിക്കുന്നത് സ്ക്രിപ്പറ്റിംഗ് ഭാഷയുടെ പേര് വ്യക്തമാക്കുന്നതിനാണ്. ഈവിടെ കൈപ്പിൽ സെസഡി സ്ക്രിപ്പറ്റിംഗ് ഭാഷയായി ജാവാസ്ക്രിപ്പറ്റ് ഉപയോഗിക്കുന്നതിനുവേണ്ടി <SCRIPT> ടാഗിലെ 'language' ആട്ടിബ്യൂക്കിന്റെ വില 'JavaScript' എന്ന നൽകണം.

ഒരു HTML പേജിൽ <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നത് താഴെ കാണുന്നത് പോലെയാണ്.

The <SCRIPT> tag can be used in an HTML page as follows.

```
<SCRIPT Language= "JavaScript">  
.....  
.....  
</SCRIPT>
```



ജാവാസ്ക്രിപ്റ്റിലെ എല്ലാനുഭവകൾ കോം സെൻസറീം ആണ്. ജാവാസ്ക്രിപ്റ്റിൽ സാധാരണയായി എല്ലാനുഭവകൾക്ക് കാമൺകേസ് (camelCase) പേരുകളാണ് ഉപയോഗിക്കുക. ഉദാഹരണത്തിൽ firstName, checkData തുടങ്ങിയവ. ഒന്നിൽകൂടുതൽ വാക്കുകൾ ഒറ്റവാക്കായി ഉപയോഗിക്കുന്ന ഒരു ഗ്രാഫിക്കൽ കാമൺകേസ്. ദാരം വാക്കില്ലെങ്കിലും ആദ്യ അക്ഷരം വലിയ അക്ഷരം ഉപയോഗിക്കുന്നതിനെ 'അപ്രശ്രൂതകാമൺകേസ്' എന്ന് വിളിക്കുന്നു. ആദ്യത്തെ വാക്കുശിക്കുക മറ്റൊരു വാക്കില്ലെങ്കിലും ആദ്യ അക്ഷരം വലിയ അക്ഷരം ഉപയോഗിക്കുന്നതിനെ 'ലോവർകാമൺകേസ്' എന്ന് വിളിക്കുന്നു. കാമൺകേസ് രീതി വാക്കുകൾ വായിക്കുവാൻ എഴുപ്പമുണ്ടാക്കുന്നു.

ഈ നമ്മക്ക് താഴെകാടുത്തിതിക്കുന്ന HTML ഫയൽ പരിഗണിക്കാം

#### ഉദാഹരണം 6.1: ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ചു് ഒരു വെബ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Welcome</TITLE> </HEAD>
<BODY>
    <SCRIPT Language= "JavaScript">
        document.write("Welcome to JavaScript .");
    </SCRIPT>
</BODY>
</HTML>
```

മുകളിലുള്ള കോഡ് ഏതെങ്കിലും ടെക്സ്റ്റ് എഡിറ്ററിൽ ടെസ്റ്റ് ചെയ്യാം. മുൻ അധ്യായങ്ങളിൽ HTML



ചിത്രം 6.1: ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്ന വെബ് പേജ്

പേജുകൾ സൃഷ്ടിക്കുന്നതിന് നിങ്ങൾ ഉപയോഗിച്ചിരിക്കുന്ന ജീനി എഡിറ്ററിലും ഈ ടെസ്റ്റ് ചെയ്യാവുന്നതാണ്. code6.html എന്ന പേരിൽ ഫയൽ സൃഷ്ടിക്കുക. HTML പേജിൽ ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ചിട്ടുണ്ടെങ്കിലും അത് സേവ് ചെയ്തിരിക്കുന്നത്. html എക്സ്പ്ലാൻഷൻലാം എന്നത് പ്രത്യേകം ശ്രദ്ധിക്കേണ്ടതാണ്. ഈ ഏതെങ്കിലും ഒരു ബൈറ്റസറിൽ നമ്മൾ സേവ് ചെയ്ത HTML ഫയൽ തുറക്കുക. ചിത്രം 6.1 തെ കാണിച്ചിരിക്കുന്നതുപോലെ ഒരു വെബ് പേജ് ലഭിക്കും. 'document.write' എന്ന പ്രസ്താവന ചെറിയ അക്ഷരങ്ങളിലാണ് എഴുതിയിരിക്കുന്നത് എന്നതും ശ്രദ്ധിക്കുക, കാരണം ജാവാസ്ക്രിപ്റ്റ് കോം സെൻസറീം സ്ക്രിപ്റ്റിംഗ് ഓഷ്യറാണ്. സ്ക്രിപ്റ്റിലെ കീവേഡുകൾ എല്ലാം ചെറിയ അക്ഷരത്തിലാണ് എഴുതേണ്ടത്.

മുകളിൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിൽ, document.write() എന്നത് ഒരു ജാവാസ്ക്രിപ്റ്റ് നിർദ്ദേശം ആണ്. അത് HTML പേജിൽ സോഡിക്കേതത് ഒരു ടെക്സ്റ്റ് ഉൾക്കൊള്ളിക്കുവാൻ ഉപയോഗിക്കുന്നു. അതായത്, മുകളിൽ നൽകിയിരിക്കുന്ന HTML കോഡ് നിർമ്മിക്കുന്ന അതേ വെബ് പേജ് താഴെ നൽകിയിരിക്കുന്ന ജാവാ സ്ക്രിപ്റ്റ് ഇല്ലാത്ത HTML കോഡും നിർമ്മിക്കുന്നു.

```

ഉദാഹരണം 6.2: HTML ഉപയോഗിച്ച് ഒരു വെബ് പേജ് നിർമ്മാണം

<HTML>
<HEAD> <TITLE>Javascript - Welcome</TITLE> </HEAD>
<BODY>
    Welcome to JavaScript.
</BODY>
</HTML>

```

മേൽ വിവരിച്ച ഉദാഹരണങ്ങൾ 6.1 ലും 6.2 ലും താരതമ്യം ചെയ്യാം. രണ്ടാമത്തെ HTML കോഡിൽ, "Welcome to JavaScript" എന്നത് ബോധി ഭാഗത്തിൽ നേരിട്ട് എഴുതിയിരിക്കുന്നു. ആദ്യത്തെ കോഡിൽ "Welcome to JavaScript" എന്നത് ജാവാസ്ക്രീപ്പ് രീതിയായ document.write () ഉപയോഗിച്ച് ബോധി ഭാഗത്ത് ഉൾപ്പെടുത്തിയിരിക്കുന്നു.

യഥാർത്ഥത്തിൽ 'document' എന്നത് വെബ് പേജിന്റെ ബോധി വിഭാഗത്തെ പ്രതിനിധിയാം ചെയ്യുന്നു. അതിനാൽ document.write() എന്ന ജാവാസ്ക്രീപ്പ് ഫംഗഷൻ വെബ് പേജിലെ ബോധി ഭാഗത്തിൽ ഒരു ടെക്സ്റ്റ് ഉൾപ്പെടുത്തും. C++ പോലെ സ്ക്രിപ്റ്റിലെ ഓരോ പ്രസ്താവനയും ഒരു അർഭവിരാമത്തിൽ (;) അവസാനിക്കുന്നു എന്നത് ശ്രദ്ധിക്കുക.

<SCRIPT Language= "JavaScript"> എന്ന HTML നിർദ്ദേശം സൂചിപ്പിക്കുന്നത് തുടർന്നുവരുന്നത് ജാവാസ്ക്രീപ്പ് കോഡ് ആണെന്നുള്ളതാണ്. ഈനി നമുക്ക് ജാവാസ്ക്രീപ്പറിയെ ബേഹസർ എങ്ങനെ കൈകാര്യം ചെയ്യുന്നുവെന്ന് കാണാം. എല്ലാ ബേഹസറിലും ജാവാസ്ക്രീപ്പ് എൻജിൻ ഉണ്ട്. പ്രവർത്തിപ്പിക്കുന്ന സമയത്ത് സ്ക്രിപ്പ് കോഡിനെ ജാവാസ്ക്രീപ്പ് എൻജിൻ ഇന്റീർക്പ്രൈ്�ഡ് ചെയ്യും. ജാവാസ്ക്രീപ്പ് കോഡ് എക്സിക്യൂട്ട് ചെയ്യാനുള്ള വിർച്ച്യുൽ മെഷിനു് (virtual machine) ജാവാസ്ക്രീപ്പ് എഞ്ചിൻ. ബേഹസർ, ജാവാസ്ക്രീപ്പ് കോഡ് കാണുമ്പോൾ അതിനെ സ്ക്രിപ്പ് എൻജിനിലൂടെ കടത്തിവിടുന്നു. ജാവാസ്ക്രീപ്പ് എഞ്ചിൻ കോഡിനെ പ്രവർത്തിപ്പിക്കുന്നു. ഒരു HTML പേജിൽ ജാവാസ്ക്രീപ്പ് കോഡ് ഇല്ലെങ്കിൽ ബേഹസറിന് ഒറ്റയ്ക്ക് തന്നെ HTML പേജ് പ്രദർശിപ്പിക്കാൻ കഴിയും. ജാവാസ്ക്രീപ്പ് കോഡ് ഉണ്ടെങ്കിൽ ജാവാസ്ക്രീപ്പ് എഞ്ചിൻ സഹായത്തോടുകൂടി മാത്രമേ HTML പേജ് പ്രദർശിപ്പിക്കുവാൻ കഴിയും. ഇപ്പോൾ ജാവാസ്ക്രീപ്പ് ഇല്ലാത്ത HTML ഫയലുകൾ എപ്പോഴും ജാവാസ്ക്രീപ്പ് ഉള്ളവയെക്കാൾ വേഗത്തിൽ പ്രദർശിപ്പിക്കാൻ കഴിയും.

ഉദാഹരണം 6.3 തെ ജാവാസ്ക്രീപ്പ് കോഡും HTML ടാഗുകളും ഇടകലർത്തി നൽകിയിരിക്കുന്നു. ഇതിന്റെ ഒരുപുത്ര വെബ്‌പേജ് ചിത്രം 6.2 തെ നൽകിയിരിക്കുന്നു.

**ഉദാഹരണം 6.3:** ഫൈലിൽ ടാഗുകൾ ഉൾക്കൊള്ളുന്ന ഒരു വെബ് പേജ് തിരുത്താണ്.

```
<HTML>
<HEAD> <TITLE>Javascript - Welcome</TITLE> </HEAD>
<BODY>
    <H1>
        <SCRIPT Language= "JavaScript">
            document.write("This is in H1 Head");
        </SCRIPT>
    </H1>
    <BR>
    <H2>
        <SCRIPT Language= "JavaScript">
            document.write("This is in H2 Head");
        </SCRIPT>
    </H2>
</BODY>
</HTML>
```



മുകളിൽ നൽകിയിരിക്കുന്ന HTML ടാഗുകൾക്കിടയിൽ ഓൺലൈൻ തവണ സ്ക്രിപ്റ്റിന് കോഡുകൾ ഉപയോഗിച്ചിട്ടുണ്ട്. ഫൈലം 6.2: ഫൈലിൽ ടാഗുകൾ അടങ്കുന്ന വെബ് പേജ് അതുപോലെ നിങ്ങൾക്ക് HTML ടാഗുകൾ കിടയിൽ എത്ര തവണ വേണമെങ്കിലും സ്ക്രിപ്റ്റ് കോഡുകൾ ഉപയോഗിക്കാം. സ്ക്രിപ്റ്റ് കോഡുകൾ ഉപയോഗിക്കുന്നത് എവിടെയായിരുന്നാലും <SCRIPT>, </SCRIPT> ടാഗുകൾക്കിടയിൽ അവ എഴുതിയിരിക്കണം.



ഈ വെബ് പേജിൽ ജാവാസ്ക്രിപ്റ്റ് പ്രവർത്തനം ദിനവാഴിക്കുന്നതിനോ നിരാകരിക്കുന്നതിനോ ഒരു സത്തരം മുള്ള വെബ് വൈറസുകളും ഉപയോക്താക്കൾക്ക് നേരക്കുന്നുണ്ട്. ജാവാസ്ക്രിപ്റ്റ് നിരാകരിക്കുന്നതില്ലെങ്കിൽ, ഒരു വൈറസറിലെ ജാവാസ്ക്രിപ്റ്റ് എന്നിന്ന് പ്രവർത്തനമുണ്ട് നിരാകരിക്കുന്നത്. ഫോറൈറ്റിൽ, Tools -> Options -> Content -> Enabled Java Script തിരഞ്ഞെടുത്ത് ജാവാസ്ക്രിപ്റ്റ് പ്രവർത്തനകൾ മാറ്റുകയോ പ്രവർത്തനകൾ മാറ്റുതാക്കയോ ചെയ്യാം. Google Chrome അല്ലെങ്കിൽ, Tools -> Settings -> Show Advanced Settings -> Content Settings -> Do not allow any site to run JavaScript നിൽ ഉപയോഗിക്കാം. ഈ വെബ് വൈറസറിൽ ജാവാസ്ക്രിപ്റ്റ് പ്രവർത്തനകൾ മാറ്റുതാക്കിയിട്ടുണ്ടെങ്കിൽ ഒരു വൈറസറിൽ സ്ക്രിപ്റ്റ് പ്രവർത്തിക്കുകയില്ല അതായൽ, വൈറസർ <SCRIPT> ... </SCRIPT> ടാഗുകൾക്കിടയിലുള്ള ഒരുക്കണം അവഗണിക്കും. ഒരു വൈറസറിൽ പ്രവർത്തനം പ്രധാനമായും അതിന്റെ സ്ക്രിപ്റ്റ് എന്നിന്ന് പ്രകടനം ആവശ്യിച്ചിരിക്കുന്നു. ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കാതെ ഒരു സീസ്റ്റാമീക് വെബ് പേജ് കാണാൻ പ്രധാനമാണ്. ദിക്കുന്നു, വൈറസുകൾ ശക്തവുമായ ജാവാസ്ക്രിപ്റ്റ് എന്നിന്നുകളും വികസനത്തിന്റെയി മുള്ള ഏവയുമാർ നിർമ്മാതാക്കളും പരസ്പരം മത്സ്യമാണ്.



### നിങ്ങളുടെ പുരോഗതി അറിയാം



1. <SCRIPT> ടാഗിന്റെ ഒരു ആട്ടിബ്യൂട്ടിന്റെ പേര് എഴുതുക.
2. ജാവാസ്ക്രിപ്റ്റിൽ ഒരു HTML പേജിന്റെ ബോധി ഭാഗത്തിൽ ഒരു ടെക്സ്റ്റ് പ്രിൻ്റ് ചെയ്യാൻ ഉപയോഗിക്കുന്ന ഫല്ലിംഗ് \_\_\_\_\_ ആണ്
3. ജാവാസ്ക്രിപ്റ്റ് കോഡ് ആണ് എഴുതിയിരിക്കുന്നത് എന്ന് വ്യക്തമാക്കുന്നത് <SCRIPT> ടാഗിന്റെ language ആട്ടിബ്യൂട്ടിന് നൽകിയ \_\_\_\_\_ എന്ന മുല്യമാണ്.
4. ജാവാസ്ക്രിപ്റ്റ് എന്തിന്റെ ഉപയോഗം എന്താണ്?
5. താഴെ പറയുന്ന പ്രസ്താവനകൾ ശരിയോ തെറ്റോ എന്ന് എഴുതുക.
  - a. ജാവ സ്ക്രിപ്റ്റ് മാത്രമാണ് കൈപ്പിന്റെ വശം സ്ക്രിപ്റ്റിംഗ് ഭാഷ.
  - b. ഒരു HTML പേജിൽ കൂടുതൽ വശം സ്ക്രിപ്റ്റിംഗ് ഭാഷ ഉൾപ്പെടുത്താൻ <SCRIPT> ടാഗാണ് ഉപയോഗിക്കുന്നത്.
  - c. ഒരു HTML ഫയലിൽ ഒരു <SCRIPT> ടാഗ് മാത്രമേ അടങ്കിയിരിക്കാം.
  - d. നമുക്ക് HTML കോഡിൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് ഇടകലർത്തുവാൻ കഴിയും.
  - e. ജാവാസ്ക്രിപ്റ്റ് കോഡ് എല്ലായ്പ്പോഴും <SCRIPT>, </ SCRIPT> എന്നതിന് അകത്തതായിരിക്കണം.
  - f. ഓരോ ജാവാസ്ക്രിപ്റ്റ് പ്രസ്താവനകളും അർഭവിരാമം(;) കൊണ്ട് അവസാനിക്കുന്നു.



പേജിന്റെ ബോധി വിഭാഗത്തിനുള്ളിൽ മാത്രം ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് താഴെ പറയുന്ന വെബ് പേജ് സൃഷ്ടിക്കാൻ ഒരു HTML കോഡ് എഴുതുക. അതായ്ത്, ബോധി വിഭാഗം താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ ആയിരിക്കണം.

#### നമ്മുടാം ചെയ്യാം

```
<BODY>
<SCRIPT Language= "JavaScript">
.....
.....
</SCRIPT>
</BODY>
```



### 6.2 ജാവാസ്ക്രിപ്റ്റിലെ ഫല്ലിംഗ് കൂട്ടുകളുടെ നിർമ്മാണം

C++ ലെ ഫാൻഡിംഗുകളെക്കുറിച്ച് നാം നേരത്തെ മനസ്സിലാക്കിയിട്ടുണ്ട്. ജാവാസ്ക്രിപ്റ്റിലും പ്രവർത്തനങ്ങൾ നിർവ്വചിക്കപ്പെട്ടിട്ടുള്ളതും വിളിക്കപ്പെടുന്നതും C++ ലെ അതേ രീതിയിൽ തന്നെയാണ്. ഒരു പേരുള്ള നിർദ്ദേശങ്ങളുടെ കൂട്ടമാണ് ഫല്ലിംഗ്. വ്യത്യസ്ത

ആവശ്യങ്ങൾക്ക് ഉപയോഗിക്കാൻ കഴിയുന്ന ധാരാളം അതിർത്തിൽ ഫല്ലിംഗുകൾ ജാവാസ്ക്രിപ്റ്റിൽ ഉണ്ട്. ഇവയിൽ ചില ഫല്ലിംഗുകൾ ഈ അധ്യായത്തിൽ പിന്നീട് ചർച്ചചെയ്യും. ഈ അതിർത്തിൽ ഫല്ലിംഗുകൾക്ക് പുറമെ നമ്മൾക്കും ഫംഗ്ഷനുകൾ നിർവ്വചിക്കാം. ഒവവ് പ്രോഗ്രാമിരുളി ഒരു ഭാഗം ഒന്നിൽ കൂടുതൽ തവണ പ്രവർത്തിപ്പിക്കാം എന്നതാണ് ഒരു ഫംഗ്ഷൻ ഉപയോഗിക്കുന്നതുകൊണ്ടുള്ള ഏറ്റവും വലിയ മെച്ച്. ഈ ഭാഗം ഒരു പ്രാവശ്യം ഫംഗ്ഷനായി എഴുതിയാൽ മതി. താഴെ പറയുന്ന കോഡ് നോക്കുക.

```
function print()
{
    document.write("Welcome to JavaScript.");
}
```

ഈവിടെ, ഒരു ഫംഗ്ഷൻ നിർവ്വചിക്കുവാനുള്ള കീവേൾ ‘function’ ഉം ഫംഗ്ഷൻ്റെ പേര് print ഉം ആണ്. ഫംഗ്ഷൻ്റെ പേര് സാധ്യതയുള്ള ഏതൊരു ഫല്ലിംഗിലെയറും ആകാം. C++ ത്ത് ഫല്ലിംഗിലെയറുകൾ നാമകരണം ചെയ്യുന്ന അതേ നിയമങ്ങൾ ജാവാസ്ക്രിപ്റ്റിലും ഉപയോഗിക്കുന്നു. ഈവിടെ print() ഫംഗ്ഷനിൽ ഒരു പ്രസ്താവന മാത്രമേ ഉള്ളൂ, ആവശ്യമെങ്കിൽ ഒന്നിലധികം പ്രസ്താവനകൾ ഒരു ഫല്ലിംഗനിൽ ഉൾപ്പെടുത്താ വുന്നതാണ്. ഒരു ഫംഗ്ഷൻ നിർവ്വചിക്കുന്നത് കൊണ്ട് അത് സ്വയം പ്രവർത്തിക്കണം എന്നില്ല, അതിനെ വിളിക്കുമ്പോൾ മാത്രമേ അത് പ്രവർത്തിക്കു. ഇതിനർത്ഥം ഒരു ഒവവ് പേജിലെ ഒരു ഫംഗ്ഷൻ വിളിക്കാതിരിക്കുകയാണെങ്കിൽ ആ ഫംഗ്ഷൻ ഒരിക്കലും പ്രവർത്തിക്കുകയില്ല. ഫംഗ്ഷൻ്റെ പേര് ഉപയോഗിച്ച് താഴെ പറയുന്ന ഫംഗ്ഷനെ വിളിക്കാം.

```
print();
```

ഫംഗ്ഷൻ്റെ പേരിന് ശേഷമുള്ള അർധവിരാമം ശ്രദ്ധിക്കുക. നമുക്ക് മുകളിലെ print() ഫംഗ്ഷൻ ഒരു HTML പേജിൽ ഉപയോഗിക്കുന്നത് എങ്ങനെ എന്ന് നോക്കാം.

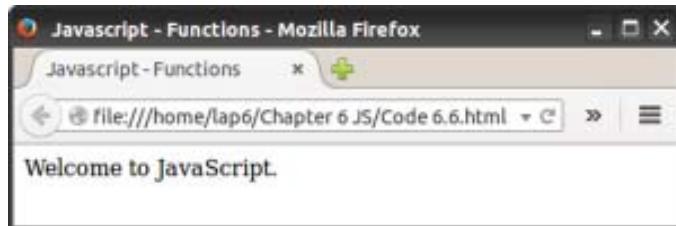
#### ഉദാഹരണം 6.4: print() ഫംഗ്ഷൻ അംദഞ്ചിയ ഒവവ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Functions</TITLE>
<SCRIPT Language= "JavaScript">
    function print()
    {
        document.write("Welcome to JavaScript.");
    }
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```



HTML പേജിന്റെ ബോധി ഭാഗത്തിൽ ഒന്നും എഴുതിയിട്ടില്ലെന്നത് ശ്രദ്ധിക്കുക. ഈ കോഡ് ബഹുസാർ വിൻഡോയിലെ 'Welcome to JavaScript' പ്രദർശിപ്പിക്കില്ല. കാരണം മംഗൾഷൻ നിർവചിച്ചിട്ടുണ്ടെങ്കിലും, അത് പേജിലെ ഏതെങ്കിലും സ്ഥലത്തുനിന്ന് വിളിച്ചിട്ടില്ല. അതിനാൽ, മംഗൾഷൻ ഒരിക്കലും പ്രവർത്തിക്കുകയില്ല, സ്ക്രീനിൽ ഒന്നും പ്രദർശിപ്പിക്കുകയുമില്ല.

ബഹുസാർ ജാലകത്തിൽ "Welcome to JavaScript" എന്ന ഒരു പുറ്റ് ലഭിക്കുന്നതിന് HTML പേജി ന്റെ ബോധി ഭാഗം താഴെ കാണുന്ന രീതിയിൽ പരിശ്കരിക്കേണ്ടതുണ്ട്.



ചിത്രം 6.3: മംഗൾഷൻ ഉപയോഗിക്കുന്ന രീതി

```
<BODY>
<SCRIPT Language= "JavaScript">
    print();
</SCRIPT>
</BODY>
```

ഈ നമ്മക്ക് ജാവാ സ്ക്രിപ്റ്റ് മംഗൾഷൻ വാക്യാലടന പരിചയപ്പെടാം

```
function function_name()
{
    statements;
}
```

ഈവിടെ 'function\_name ()' എന്ന വരിയിൽ മംഗൾഷൻ ഹെഡറും, ബോക്കറ്റിലുള്ള കോഡും ചേർത്ത് മംഗൾഷൻ ബോധി എന്നു വിളിക്കുന്നു. C++ ത്ര മംഗൾഷൻ റിട്ടേൺ ദെപ്പ് ഉണ്ടെങ്കിലും ജാവാസ്ക്രിപ്റ്റിൽ മംഗൾഷൻ റിട്ടേൺ ദെപ്പില്ല. ജാവാസ്ക്രിപ്റ്റിലും C++ ലേതുപോലെ നിങ്ങൾക്ക് മംഗൾഷനിൽ നിന്ന് ചില മുല്യങ്ങൾ തിരികെ നൽകാം. ഈ അധ്യായത്തിൽ ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗത്തെക്കുറിച്ചുള്ള അടിസ്ഥാന ആശയങ്ങൾ മാത്രം നൽകാൻ ഉദ്ദേശിച്ചിട്ടുള്ളതിനാൽ, ഇത്തരം കാര്യങ്ങൾ ഇവിടെ ചർച്ച ചെയ്യുന്നില്ല. മറ്റാരു വ്യത്യാസം C++ ത്ര ഒരു മംഗൾഷൻ നിർവ്വചക്രാണ് function എന്ന കീവേയ് ഉപയോഗിക്കില്ല, എന്നാൽ ജാവാസ്ക്രിപ്റ്റ് റിട്ടേൺ എന്ന കീവേയ് ഉപയോഗിക്കുന്നു. HTML പേജിന്റെ ഫോഡ് ഭാഗത്തിനുള്ളിൽ മംഗൾഷൻ നിർവചിക്കപ്പെട്ടിരിക്കുന്നതായി നിങ്ങൾ ശ്രദ്ധിച്ചിരിക്കാം. ഫോഡ് ഭാഗത്തിനുള്ളിൽ തന്നെ മംഗൾഷൻ നിർവചിക്കണം എന്ന് നിർബന്ധമില്ല, ബോധി ഭാഗത്തും മംഗൾഷൻ നിർവചിക്കാം.

```
<BODY>
<SCRIPT Language= "JavaScript">
    function print()
```



```
{  
    document.write("Welcome to JavaScript.");  
}  
print();  
</SCRIPT>  
</BODY>
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന കോഡ് ഈതേ ഒരു പുതിയ നൽകുന്നത് ചിത്രം 6.3 തെക്കാടുത്തിരിക്കുന്നു.

ഹംഗൾ ബോധി ഭാഗത്തിനകത്താണ് നിർവചിക്കപ്പെട്ടിരിക്കുന്നതെങ്കിൽപ്പോലും അതിന്റെ പ്രവർത്തിപ്പിക്കുന്നതിനുവേണ്ടി അതിനെ വിളിക്കണം എന്നത് ശ്രദ്ധിക്കുക. ഉദാഹരണത്തിന് താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് സ്കൈനിൽ ഓനും പ്രദർശിപ്പിക്കുന്നില്ല.

```
<BODY>  
<SCRIPT Language= "JavaScript">  
function print()  
{  
    document.write("Welcome to JavaScript.");  
}  
</SCRIPT>  
</BODY>
```

ഒരു HTML പേജിൽ എവിടെയും ഒരു ഹംഗൾ ഷൾ നിർവചിക്കപ്പെടാമെങ്കിലും, ഫോഡ് ഭാഗത്തിൽ ഹംഗൾ നിർവചനം ഉൾപ്പെടുത്തുന്നതാണ് ലഭ്യ.

print() ഹംഗൾ റണ്ടു തവണ വിളിച്ചിരിക്കുന്ന താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് പരിഗണിക്കുക, ഈത് ചിത്രം 6.4യും കാണുന്ന രീതിയിലുള്ള ഒരു പുതിയ നൽകുന്നു.

```
<BODY>  
<SCRIPT Language= "JavaScript">  
    print();  
    print();  
</SCRIPT>  
</BODY>
```

ഒരു പുതിയ ചിത്രം 6.4 തെക്കാടുത്തിരിക്കുന്നതുപോലെ ആയിരിക്കും. എന്തുകൊണ്ട്?

വാസ്തവത്തിൽ, നമ്മൾ റണ്ടുതവണ ഹംഗൾ നേരിട്ടെന്നോശ്ര, അത് 'Welcome to JavaScript' എന്ന റണ്ടുതവണ ബോധി ഭാഗത്ത് എഴുതും. അതിനാൽ മുകളിലുള്ള കോഡിന്റെ ബോധി ഭാഗം താഴെ എഴുതിയിരിക്കുന്നതിനു തുല്യമായിരിക്കും.



ചിത്രം 6.4: ഭേദക്കുള്ളാതെ ഞേർപ്പിയ്ക്കുന്ന ഹംഗൾ ഷൾ ഉപയോഗിക്കുന്ന ബഹുപോജ്

Java വാസ്കൈപ്പ് ഉപയോഗിച്ചുള്ള ശൈലി സെവ്യ സ്ക്രിപ്റ്റ്

```

<BODY>
    Welcome to JavaScript.Welcome to JavaScript.
</BODY>
സന്ദേശം രണ്ട് വ്യത്യസ്ത വർകളിൽ പ്രദർശിപ്പിക്കുന്നതിന് നിർബന്ധമായും <BR>
ഒരു ഉപയോഗിക്കണം.
```

```

<BODY>
    Welcome to JavaScript.<BR>Welcome to JavaScript.
</BODY>
```

ജാവാസ്ക്രീപ്പ് ഉപയോഗിച്ച് ഇതേ റിസൾട്ട് ലഭിക്കുന്നതിന് താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ മാറ്റം വരുത്താം.

```

<SCRIPT Language= "JavaScript">
    function print()
    {
        document.write("Welcome to JavaScript.<BR>");
    }
</SCRIPT>
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന HTML കോഡിന്റെ ഒരു പൂര്ണ ഫിത്തോ 6.5 ടെക്നോളജിക്കുന്നു.

ചിത്രം 6.3: റണ്ട് ഫീറ്റ് മാർഷഗുകൾ ഉപയോഗിക്കുന്ന ഏബ്ലെപ്പജ്ഞ്

### നിങ്ങളുടെ പുരോഗതി അറിയാം



1. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന ശരിയോ തെറ്റോ എന്ന് പറയുക.
  - a. ഒരു ഏബ്ലെപ്പജ്ഞ് ബൈറ്റസറിൽ തുറക്കുന്നോൾ ഫംഗ്ഷൻ സമേധയാ പ്രവർത്തിക്കും.
  - b. ഒരു ഫംഗ്ഷൻ സാധാരണയായി HTML പേജിന്റെ ഫോള്ഡ് ഭാഗത്തിൽ ഉൾപ്പെടുത്തുന്നു.
  - c. ഒരു ഫംഗ്ഷൻ എത്ര തവണ വേണമെങ്കിലും വിളിക്കാം.
  - d. ഭോധി ഭാഗത്തിനകത്ത് ഒരു ഫംഗ്ഷൻ നിർവ്വചിച്ചിട്ടുണ്ടെങ്കിലും അത് വിളിച്ചിട്ടില്ലെങ്കിൽ പ്രവർത്തിക്കില്ല.
2. ജാവാസ്ക്രീപ്പിൽ ഫംഗ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന്റെ ഗുണങ്ങളെപ്പറ്റി എഴുതുക.



നമ്മകൾ മെച്ചാം

1. താഴെക്കൊടുത്തിരിക്കുന്ന രേഖ എം കോഡ് A ഉം കോഡ് B ഉം) പരിഗണിച്ച് ഒരുപുട്ട് എഴുതുക.

കോഡ് A

```
<HTML>
<HEAD>
<SCRIPT Language= "JavaScript">
    function print()
    {
        document.write("Welcome to JavaScript");
    }
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    print();
    document.write("<BR>");
    print();
</SCRIPT>
</BODY>
</HTML>
```

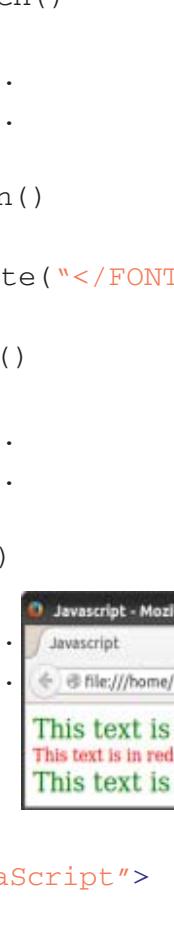
കോഡ് B

```
<HTML>
<HEAD>
<SCRIPT Language= "JavaScript">
function print()
{
    document.write("Welcome to JavaScript");
}
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    print();
</SCRIPT>
<BR>
<SCRIPT Language= "JavaScript">
    document.write("<BR>");
    print();
</SCRIPT>
</BODY>
</HTML>
```



2. ඩිග්‍රම 6.6 හේ කාඩ්‍රියිකුනු තුපොල ගැටුපුද් ලබා කුනු තියේ නෑ  
HTML කොයි තාഴේ කෙටුවකුණු. කොයි ත්‍යාම් startGreen(),  
stopGreen(), startRed() and stopRed() අඟිජීව පොයි ගැහැතියේ  
හස්සුක්‍රියා මූල්‍ය නිශ්චිත නෑ. stopGreen() අඟිජීව පොයි  
සියලුම පොයි නිශ්චිත නෑ. මෙයි පොයි නිර්වචනය නෑ නිස් නෑ.  
නිස් නිස්

```
<HTML>
<HEAD>
<SCRIPT Language= "JavaScript">
    function startGreen()
    {
        .....
        .....
    }
    function stopGreen()
    {
        document.write("</FONT>");
    }
    function startRed()
    {
        .....
        .....
    }
    function stopRed()
    {
        .....
        .....
    }
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    startGreen();
</SCRIPT>
This is in Green colour with size 5
<SCRIPT Language= "JavaScript">
    stopGreen();
```



The screenshot shows a Mozilla Firefox window titled 'Javascript - Mozilla Firefox'. The address bar shows 'file:///home/lap6/Chapt'. The page content displays several text blocks: 'This text is in green' in green, 'This text is in red colour with size 5' in red, and 'This text is in green' again in green. The text 'This text is in red colour with size 5' is larger than the other text blocks.





```
</SCRIPT>
<SCRIPT Language= "JavaScript">
    startRed();
</SCRIPT>
This is in Red colour with size 3
<SCRIPT Language= "JavaScript">
    stopRed();
</SCRIPT>
<SCRIPT Language= "JavaScript">
    startGreen();
</SCRIPT>
This is in Green colour with size 5
<SCRIPT Language= "JavaScript">
    stopGreen();
</SCRIPT>
</BODY>
</HTML>
```

### 6.3 ജാവാസ്ക്രിപ്റ്റിലെ ഡാറ്റാ ഇനങ്ങൾ

മെമ്മറി ഫലപ്രദമായി ഉപയോഗപ്പെടുത്തുന്നതിന് പ്രോഗ്രാമിംഗ് ഭാഷകൾ ഡാറ്റയെ വ്യത്യസ്ത വിഭാഗങ്ങളായി തരം തിരിച്ചിത്തിക്കുന്നു. C++ ലെ അടിസ്ഥാന ഡേറ്റാ തരങ്ങൾ int, char, float, double, void തുടങ്ങിയവയാണെന്ന് നമ്മൾ മനസ്സിലാക്കിയിട്ടുണ്ട്. ഈ ഡാറ്റ തരങ്ങളൊപ്പും, ഒരു മോഡിഫയറുകളും C++ ലെ ഉണ്ടായിരുന്നു. അടിസ്ഥാന ഡാറ്റ തരങ്ങളുടെ എണ്ണം 3 ആയി പരിമിതപ്പെടുത്തിയിരിക്കുന്നതിലൂടെ ജാവാസ്ക്രിപ്റ്റ് ഇതിന്റെ സകീറ്റുറ്റ് കുറയ്ക്കുന്നു. താഴെ പറയുന്നവയാണ് ജാവാസ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന മൂന്ന് അടിസ്ഥാന ഡാറ്റ തരങ്ങൾ.

#### 6.3.1 നമ്പർ (Number)

എല്ലാ നമ്പറുകളും ഈ വിഭാഗത്തിൽ വരുന്നു. പോസിറ്റീവ് സംവ്യക്തി, നെഗറ്റീവ് സംവ്യക്തി, എല്ലാ ഫ്ലോട്ട് വിലകളും (ഭിന്നസംവ്യക്തി), പൂർണ്ണ സംവ്യക്തി നമ്പർ (number) ഡാറ്റ തരം ആയി പരിഗണിക്കപ്പെടും. അതിനാൽ, 27, -300, 1.89, -0.0082 എന്നിവ ജാവാസ്ക്രിപ്റ്റിലെ നമ്പർ ഡാറ്റ തരത്തിനുള്ള ഉദാഹരണങ്ങളാണ്.

#### 6.3.2 സ്ട്രിംഗ് (String)

ഈടു ഉദാഹരണികൾക്കുള്ളിൽ ഉള്ള പ്രതീകങ്ങൾ, അക്കങ്ങൾ അല്ലെങ്കിൽ മറ്റേതെങ്കിലും ചിഹ്നങ്ങളുടെ കൂടുതൽ ജാവാസ്ക്രിപ്റ്റിൽ ഒരു സ്ട്രിംഗായി പരിഗണിക്കും. ഉദാഹരണം “Kerala”, “Welcome”, “SCHOOL”, “1234”, “Mark20”, “abc\$” and “sanil@123”.

#### 6.3.3 ബൗളിയൻ (Boolean)



ഈ വിഭാഗത്തിൽ രണ്ട് മുല്യങ്ങൾ മാത്രമേ ഉള്ളൂ, അവ true, false എന്നിവയാണ്. ഈ മുല്യങ്ങൾ ഇരട്ട ഉദഘരണികളിലാണ്. ഇരട്ട ഉദഘരണികൾക്കുള്ളിൽ ആണെങ്കിൽ അവ സ്റ്റ്രിംഗ് ആയിരിക്കും. ജാവാസ്ക്രിപ്റ്റ് കേസ് സെൻസറീം ആയതുകൊണ്ട് ബുളിയൻ വിലകളെ പ്രതിനിധാനം ചെയ്യുന്നതിന് TRUE, FALSE എന്നിങ്ങനെ ഉപയോഗി കുവാൻ പാടില്ല.

#### 6.4 ജാവാ സ്ക്രിപ്റ്റിലെ വൈറിയബിള്യുകൾ

വിലകൾ സംഭരിക്കുന്നതിന് വേറിയബിള്യുകൾ ഉപയോഗിക്കാമെന്ന് നമുക്കറിയാം. ജാവാസ്ക്രിപ്റ്റിൽ var എന്ന കീവോൾ ഉപയോഗിച്ചാണ് എല്ലാ തരം വേറിയബിള്യുകളും പ്രവൃത്തിക്കുന്നത്.

```
var x;
```

ഇവിടെ 'x' എന്നത് വേറിയബിളിന്റെ പേരാണ്, ഒരു വേറിയബിളിന് C++ ലേതുപോലെ ഏതു പേരും ഉപയോഗിക്കാം. അത് ഒരു സാധുതയുള്ള എയർഫീഫയർ ആയിരിക്കണമെന്നുമാത്രം. C++ ത്ര വ്യത്യസ്ത തരം വേറിയബിള്യുകൾ പ്രവൃത്തിക്കാൻ കീവോൾകൾ int, float, char തുടങ്ങിയവ ഉപയോഗിച്ചു. പക്ഷേ, ജാവാസ്ക്രിപ്റ്റിൽ var എന്ന കീവോൾ' മാത്രമാണ് എല്ലാത്തരം വേറിയബിള്യുകൾ പ്രവൃത്തിക്കുന്നതിനും ഉപയോഗിക്കുന്നത്. ജാവാസ്ക്രിപ്റ്റിൽ വേറിയബിളിന് വില നൽകുന്നോൾ മാത്രം നിർവ്വചനം പൂർത്തിയാക്കുന്നു. ഒരു വേറിയബിളിന് മുല്യം നൽകുന്നോൾ മാത്രമാണ് വേറിയബിളിന്റെ തരം ജാവാസ്ക്രിപ്റ്റ് മനസ്സിലാക്കുന്നത്.

താഴെകാടുത്തിരിക്കുന്ന പ്രസ്താവന പരിശീലനിക്കുക.

```
var x, y;
x = 25;
y = "INDIA";
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന ഉദാഹരണത്തിൽ, x എന്നത് നമ്പർ തരത്തിലും y സ്റ്റ്രിംഗ് തരത്തിലും പെട്ട വേറിയബിള്യുകൾ ആണ്. എന്നാൽ കീവോൾ number, string എന്നാണും ഇവിടെ ഉപയോഗിച്ചിട്ടില്ല. താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണത്തിൽ ജാവാ സ്ക്രിപ്റ്റ് എങ്ങനെയാണ് വേറിയബിള്യുകൾ നിർവ്വചിക്കുന്നത് എന്ന് വിവരിക്കുന്നു.

```
var x, y;
x = 25;
y = "INDIA";
```

ഉദാഹരണം 6.5: വൈറിയബിള്യുകളുടെ ഉപയോഗം വിവരിക്കുന്നു.

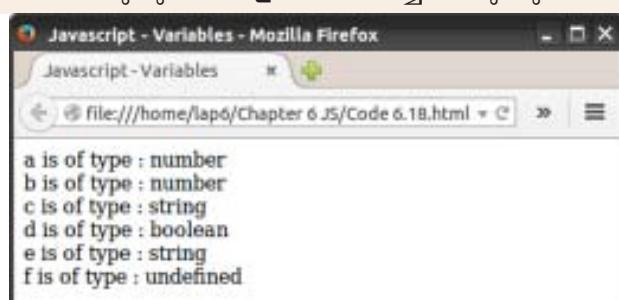
```

<HTML>
<HEAD> <TITLE>Javascript - Variables</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    var a, b, c, d, e, f;
    a = 25;
    b = 18.5;
    c = "INDIA";
    d = true;
    e = "true";
    document.write("a is of type : ");
    document.write(typeof(a));
    document.write("<BR>b is of type : ");
    document.write(typeof(b));
    document.write("<BR>c is of type : ");
    document.write(typeof(c));
    document.write("<BR>d is of type : ");
    document.write(typeof(d));
    document.write("<BR>e is of type : ");
    document.write(typeof(e));
    document.write("<BR>f is of type : ");
    document.write(typeof(f));
</SCRIPT>
</BODY>
</HTML>

```

ഈവിടെ, `typeof()` എന്നൊരു ഫലങ്ങൾ പുതുതായി ഉപയോഗി ചെയ്യുന്നു. പേര് സൂചിപ്പിക്കുന്ന തുപ്പോലെ ഒരു വേരിയബിളിഞ്ച് തരം കണ്ണുപിടിക്കുവാൻ ഈ ഫലങ്ങൾ ഉപയോഗിയ്ക്കുന്നു. ഈ ഫലങ്ങൾ പെരും വിവരങ്ങൾ നൽകുന്നുണ്ട്. അതുകൊംബന്നും മുകളിലൂള്ള ഉദാഹരണം തിരിഞ്ചെന്ന് ഒരുപോഴും ചിത്രം 6.6. തീ കൊടുത്തിരിക്കുന്നു.

വേരിയബിൾ f പ്രവ്യാഹിച്ചിട്ടുണ്ടെങ്കിലും അതിന് ഒരു വില നൽകിയിട്ടില്ല. അതുകൊണ്ട് സ്ക്രിപ്റ്റ് എന്നീന് അതിഞ്ചെന്ന് തരം മനസ്സിലാക്കുവാനോ പ്രദർശിപ്പിക്കുവാനോ സാധിക്കുന്നില്ല. നിർവ്വചിക്കാത്ത വേരിയബിളിനെ പ്രതിനിധാനം ചെയ്യുന്നതിനായി സ്ക്രിപ്റ്റിൽ



ചിത്രം 6.3: വേരിയബിളിഞ്ചുള്ളേം തരം ഉപയോഗം

പ്രദർശിപ്പിക്കുന്ന ഫലം പോജിഷൻ

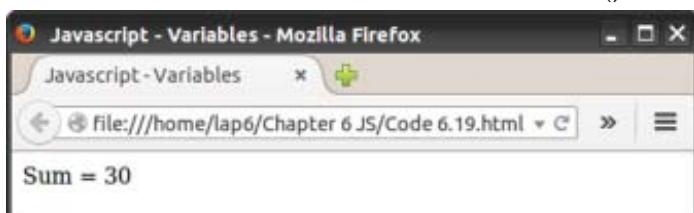


ഉപയോഗിക്കുന്ന ഒരു പ്രത്യേക ഡാറ്റാടെപ്പാണ് undefined. ഒരോറു var കീവേഴ്സ് ഉപയോഗിച്ച് എത്ര വേരിയബിള്ളുകൾ വേണ്ടുമെങ്കിലും പ്രവ്യാഹിക്കാനാകും (ഡിക്ഷയർ ചെയ്യാനാകും). വേരിയബിള്ളുകൾ കോമ (,) ഉപയോഗിച്ച് വേർത്തിരിക്കേണ്ടതാണ്. വേരിയബിള്ളുകൾ ഉപയോഗിക്കുന്ന ഒരു ജാവാസ്ക്രിപ്റ്റിൻ്റെ പ്രവർത്തനം നമുക്ക് പരിഗണിക്കാം.

ഉദാഹരണം 6.5: തണ്ട് സംഖ്യകളുടെ തുക കണ്ടു പിടിക്കുന്നതിനുള്ള വൈബ് പ്രൈം നിർമ്മാണം

```
<HTML>
<HEAD><TITLE>Javascript - Variables</TITLE>
<SCRIPT Language= "JavaScript">
function add()
{
    var m, n, sum;
    m = 20;
    n = 10;
    sum = m + n;
    document.write("Sum = ");
    document.write(sum);
}
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    add();
</SCRIPT>
</BODY>
</HTML>
```

മുകളിൽ പറഞ്ഞ ഉദാഹരണ തതിൽ, ബോധി ഭാഗത്തിൽ ഒരു ജാവാസ്ക്രിപ്റ്റ് പ്രൈം താവന മാത്രമേ അടങ്ങിയിട്ടുള്ളൂ, അത് add() എന്ന ഫംഗണ്ടെ വിളിക്കുന്നു. ഫംഗണ്ടെ പ്രവർത്തനത്തിൽ എൻ ഫലം ചിത്രം 6.7 തീ കാണിച്ചിരിയ്ക്കുന്നു.



ചിത്രം 6.7: രണ്ട് സംഖ്യകളുടെ തുക കണ്ടുപിടിക്കുന്നതിനുള്ള വൈബ് പ്രൈം

## 6.5 ജാവാസ്ക്രിപ്റ്റിലെ ഓഷ്മറ്ററുകൾ

ജാവാ സ്ക്രിപ്റ്റിലെ മിക്കവാറും എല്ലാ ഓപ്പറേറ്ററുകളും C++ ലേതിന് സമാനമാണ് അവ എത്തല്ലാമാണെന്ന് നോക്കാം

### 6.5.1 അരിതമാറ്റിക് ഓപ്പറേറ്ററൾ

ജാവാസ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന അരിതമാറ്റിക് ഓപ്പറേറ്ററുകൾ പട്ടിക 6.1 കാണിക്കുന്നു.

ഓപ്പറേഷൻ	വിവരങ്ങൾ	ഉദാഹരണം	y യുടെ മൂല്യം	ഉത്തരം (x)
+	സങ്കലനം	$x = y + 10$	15	25
-	വ്യവകലനം	$x = y - 10$	15	5
*	ഗുണനം	$x = y * 3$	15	45
/	ഹരണം	$x = y / 2$	15	7.5
%	മൊഡൂലസ്	$x = y \% 2$	15	1
ഹരണത്തിലെ ശിൽഘ്യം				
++	വർദ്ധനവ്	$x = ++y$	15	16
		Increment	$x = y++$	15
--	കുറവ്	$x = --y$	15	14
		Decrement	$x = y--$	15

മുകളിലുള്ള പട്ടികയിലെ ഫലം അനുസരിച്ച് ഓപ്പറേറ്ററുകളും C++ തുല്യമായി പ്രവർത്തിക്കുന്നതും പ്രവർത്തിക്കുന്നതും എന്ന് കാണാം.

### 6.5.2 അസൈൻമെന്റ് ഓപ്പറേറ്ററൾ

പട്ടിക 6.2 തുല്യമായി ഓപ്പറേറ്ററുകളുടെ അസൈൻമെന്റ് ഓപ്പറേറ്ററുകളുടെ ഉപയോഗം പട്ടിക കാണിച്ചിരിക്കുന്നു.

ഓപ്പറേറ്റർ	വിവരങ്ങൾ	ഉദാഹരണം	a യുടെ വലിബന്ധം	b യുടെ വലിബന്ധം	ഉത്തരം
			a	b	(a)
=	അസൈൻമെന്റ്	$a = b$	10	3	3
+=	സങ്കലനവും അസൈൻമെന്റും	$a += b$	10	3	13
-=	വ്യവകലനവും അസൈൻമെന്റും	$a -= b$	10	3	7
*=	ഗുണനവും അസൈൻമെന്റും	$a *= b$	10	3	30
/=	ഹരണവും അസൈൻമെന്റും	$a /= b$	10	3	3.33
%=	മൊഡൂലസവും അസൈൻമെന്റും	$a \%= b$	10	3	1

പട്ടിക 6.2: ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ അസൈൻമെന്റ് ഓപ്പറേറ്ററൾ



മുകളിലൂള്ള പട്ടികയിൽ നിന്ന് ഓരോ ഓപ്പറേറ്ററുകളുടെയും പ്രവർത്തനം മനസ്സിലാക്കുക വളരെ എളുപ്പമാണ്. a യുടെയും b യുടെയും വിലകളുപയോഗിച്ച് ‘ഉദാഹരണം’ എന്ന കോളത്തിലെ പ്രസ്താവന പ്രവർത്തിപ്പിച്ച ശേഷമുള്ള ഫലം ‘ഉത്തരം’ എന്ന കോളത്തിൽ കാണിച്ചിരിക്കുന്നു.

### 6.5.3 റിലേഷണൽ ഓപ്പറേറ്റർ (താരതമ്യ ഓപ്പറേറ്റർ) (Relational/Comparison operator)

ജാവാസ്ക്രിപ്റ്റിലെ ഉപയോഗിക്കുന്ന വിവിധ തരം താരതമ്യ ഓപ്പറേറ്ററുകൾ പട്ടിക 6.3 ത്ത് കാണിച്ചിരിക്കുന്നു.

ഓപ്പറേറ്റർ	വിവരണം	ഉദാഹരണം	a യുടെ വില	b യുടെ വില	ഉത്തരം (a)
<code>==</code>	തുല്യം	<code>a==b</code>	10	3	false
<code>!=</code>	തുല്യമല്ല	<code>a != b</code>	10	3	true
<code>&lt;</code>	കുറവ്	<code>a &lt; b</code>	10	3	false
<code>&lt;=</code>	കുറവോ അല്ലെങ്കിൽ തുല്യമോ	<code>a &lt;= b</code>	10	3	false
<code>&gt;</code>	കുടുതൽ	<code>a &gt; b</code>	10	3	true
<code>&gt;=</code>	കുടുതലോ അല്ലെങ്കിൽ തുല്യമോ	<code>a &gt;= b</code>	10	3	true

പട്ടിക 6.3: റിലേഷണൽ ഓപ്പറേറ്റർ

അരു താരതമ്യ/റിലേഷണൽ ഓപ്പറേഷൻ ഫലം ശരിയോ തെറ്റോ എന്ന് പട്ടികയിൽ നിന്ന് വ്യക്തമാണ്. ഈ ഓപ്പറേറ്ററുകൾ ഒരു വരുത്തും ഉള്ള മുല്യങ്ങൾ താരതമ്യം ചെയ്യുകയും അതിനുസരിച്ച് ഫലം നൽകുകയും ചെയ്യുന്നു.

### 6.5.4 ലോജിക്കൽ ഓപ്പറേറ്റർ

ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ ലോജിക്കൽ ഓപ്പറേറ്ററുകളുടെ പട്ടിക 6.4 ത്ത് ഉദാഹരണത്തോടൊപ്പം കാണിച്ചിരിക്കുന്നു.

ഓപ്പറേറ്റർ	വിവരണം	ഉദാഹരണം	a യുടെ വില	b യുടെ വില	ഉത്തരം (a)
<code>&amp;&amp;</code>	ആശ്ചേരം	<code>a &amp;&amp; b</code>	true	false	false
<code>  </code>	ഓർ	<code>a    b</code>	true	false	true
<code>!</code>	നോട്ട്	<code>! a</code>	true		false

പട്ടിക 6.4: ലോജിക്കൽ ഓപ്പറേറ്റർ

മുകളിൽ കൊടുത്തിരിക്കുന്ന പട്ടികയിൽ നിന്ന്, ഈ ഓപ്പറേറ്ററുകൾ എല്ലാം തന്നെ C++ ലേതിന് സമാനമാണെന്നത് വളരെ വ്യക്തമാണ്. വിവിധ ആവശ്യങ്ങൾക്കായി മറ്റ് ധാരാളം ഓപ്പറേറ്ററുകൾ ജാവാസ്ക്രിപ്റ്റ് നൽകുന്നു. ജാവാസ്ക്രിപ്റ്റിനുണ്ട്

നമ്മുടെ ചർച്ച ഈ അധ്യായത്തിൽ മാത്രം ഒരു അംഗങ്ങിൽക്കുന്നതിനാൽ അത്തരം ഓപ്പറേറ്ററുകളെക്കുറിച്ചുള്ള ചർച്ച ഈ പുസ്തകത്തിന്റെ പരിധിക്കു പുറത്താണ്. കൂടാതെ, മുകളിൽ വിവരിച്ച ഓപ്പറേറ്ററുകൾ ഒരു തുടക്കക്കാരനു വേണ്ട എല്ലാ പ്രവർത്തനങ്ങളും നടത്തുന്നതിന് പര്യാപ്തമാണ്. എന്നിരുന്നാലും, താഴെ പറയുന്ന സ്ട്രിംഗ് ഓപ്പറേറ്റർ വിവിധ സാഹചര്യങ്ങളിൽ നിങ്ങൾക്ക് ഉപയോഗപ്രദമാകും. ഈ ഓപ്പറേറ്റർ C++ ലെ ലഭ്യമല്ല.

### 6.5.5 സ്ട്രിംഗ് അഡിഷൻ ഓപ്പറേറ്റർ (+)

രണ്ട് സംഖ്യകൾ കൂടുന്നതിന് + ഓപ്പറേറ്റർ ഉപയോഗിക്കാമെന്ന് നമുക്ക് അറിയാം. അതെ ഓപ്പറേറ്ററായ '+' രണ്ട് സ്ട്രിംഗുകൾ കൂടിച്ചേർക്കുന്നതിനും ഉപയോഗിക്കാം. കൺകാക്കേഷൻ (concatenation) എന്നതിനർമ്മം രണ്ട് സ്ട്രിംഗുകൾ കൂടിചേർക്കുക എന്നാണ്.

ഉദാഹരണം

```
var x, y;
x = "A good beginning ";
y = "makes a good ending.";
z = x + y;
```

+ ഓപ്പറേറ്റർ രണ്ട് സ്ട്രിംഗുകൾ കൂടിചേർക്കും, അങ്ങനെ വേറിയബിൾ z എൻ്റെ മൂല്യം 'A good beginning makes a good ending' എന്നാകും. ഒരേ ഓപ്പറേറ്റർ വ്യത്യസ്ത ഓപ്പറേറ്റുകളുടെ തരം മനസ്സിലാക്കിയാണ് പ്രവർത്തിക്കുന്നത്. ഓപ്പറേറ്റർ സംഖ്യ ആണെങ്കിൽ, അവയുടെ തുക കണക്കുപിടിക്കും. മറിച്ച് സ്ട്രിംഗുകൾ ആണെങ്കിൽ അവ കൂടിചേർക്കും. ഈ, താഴെ പറയുന്നതിനുസരിച്ചു കോഡിൽ z എൻ്റെ മൂല്യം പ്രവചിക്കുക?

```
var x, y;
x = "23";
y = 5;
z = x + y;
```

ഇതിന്റെ ഉത്തരം 235 ആണ്. ഓപ്പറേറ്റുകളിൽ ഏതെങ്കിലും ഒരെണ്ണം സ്ട്രിങ്ങാണെങ്കിൽ + ഓപ്പറേറ്റർ എല്ലാ ഓപ്പറേറ്റുകളെയും സ്ട്രിങ്ങായി കരുതി കൂടിചേർക്കുന്നു. അങ്ങനെ z എൻ്റെ വിലയായി 235 ലഭിക്കുന്നു. x എൻ്റെയും y യുടെയും തുക സംഖ്യാരൂപത്തിൽ കാണണമെന്നുകരുതുക. ഇതിനായി താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ പ്രസ്താവന മാറ്റിയെഴുതാം.

```
z = Number(x) + y;
```

മുകളിൽ പറയുന്നതിനുസരിച്ചു പ്രസ്താവന പ്രവർത്തിച്ചു ശേഷം വേറിയബിൾ z എൻ്റെ വില 28 ആയിരിക്കും. Number() എന്നത് ജാവാ സ്ക്രിപ്റ്റ് ലൈ ഒരു ഫംഗ്ഷൻ ആണ്, അത് സ്ട്രിംഗ് തരം ഡാറ്റയെ നന്ദിയായി മാറ്റുന്നു. ഈ അധ്യായത്തിലെ ചില ഉദാഹരണങ്ങളിൽ ഈ ഫംഗ്ഷൻ ഉപയോഗിക്കുന്നുണ്ട്.



### നിങ്ങളുടെ പ്രവേശനത്തി അറിയാം



- നമ്പർ, സ്ക്രിപ്റ്റ്, \_\_\_\_\_ എന്നിവയാണ് ജാവാസ്ക്രിപ്റ്റിലെ അടിസ്ഥാന ധാരാ തരങ്ങൾ.
- 'true' ഒരു \_\_\_\_\_ തരം ധാരാധാരാണ്.
- 'false' ഒരു \_\_\_\_\_ തരം ധാരാധാരാണ്.
- ജാവാസ്ക്രിപ്റ്റിൽ ഒരു വേരിയബിൾ പ്രവൃത്താപിക്കാൻ വേണ്ട കീവേഡ് \_\_\_\_\_ ആണ്.
- ജാവാസ്ക്രിപ്റ്റിൽ ധാരാ തരം അറിയാൻ ഉപയോഗിക്കുന്നതിനുള്ള ഫഞ്ചണൾ \_\_\_\_\_ ആണ്.
- % ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നത് എന്തിനാണ്?
- ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ എവ.

## 6.6 ജാവാസ്ക്രിപ്റ്റിലെ കണർമ്മഭാൾ സ്ക്രിപ്റ്റുകൾ

ഒരു പ്രോഗ്രാമിന്റെ പ്രവർത്തനത്തിന്റെ നിയന്ത്രണത്തിന് കൺട്രോൾ സ്ട്രക്ചൗകൾ ഉപയോഗിക്കുന്നു. C++ തും പഠിച്ചിട്ടുള്ള എല്ലാ കൺട്രോൾ സ്ട്രക്ചൗകളും ജാവാസ്ക്രിപ്റ്റിലും വ്യത്യാസമില്ലാതെ ഉപയോഗിക്കാവുന്നതാണ്. ജാവാ സ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന വിവിധ കൺട്രോൾ സ്ട്രക്ചൗകൾ ഉദാഹരണസഹിതം നമ്മുകൾ പരിശോധിക്കാം.

### 6.6.1 if

എല്ലാ പ്രോഗ്രാമിന്റെ ഭാഷയിലും ഏറ്റവുമധികം ഉപയോഗിക്കപ്പെടുന്ന കൺട്രോൾ സ്ട്രക്ചൗരിാണിത്. ചില വ്യവസ്ഥകൾ അടിസ്ഥാനമാക്കി ഒരു പ്രസ്താവന അല്ലെങ്കിൽ ഒരു കൂടുതൽ പ്രസ്താവനകൾ പ്രവർത്തിപ്പിക്കാൻ ഇത് ഉപയോഗിക്കുന്നു. പട്ടിക 6.5 തും നൽകിയിരിക്കുന്നതുപോലെ രണ്ട് രീതിയിൽ ഇത് ഉപയോഗിക്കാം

if എണ്ണ വാക്യാലാട്ട	if ..... എണ്ണ വാക്യാലാട്ട
<pre>if (test_expression) {     statements; }</pre>	<pre>if (test_expression) {     statements; } else {     statements; }</pre>

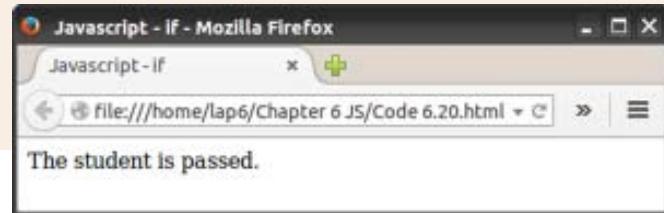
പട്ടിക 6.5: if, else ഓട്ടക്കുറിയ if പ്രസ്താവനയുടെ വാക്യാലാട്ട

if, if .... else പ്രസ്താവനയുടെ ഉപയോഗം കാണിക്കുന്ന ഒരു ഉദാഹരണം നോക്കാം.

ഉദാഹരണം 6.12: ഒരു വിദ്യാർഥി ജയിച്ചോ ഇല്ലയോ എന്നു പരിശോധിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ് നിർമ്മാണം.

**ഉദാഹരണം 6.7:** തണ്ട് സംവൃക്കളുടെ തുക കണക്ക് പിലിക്കുന്നതിനുള്ള വൈദിക്ഷണികൾ നിർമ്മാണം.

```
<HTML>
<HEAD> <TITLE>Javascript - if</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    var score;
    score = 35;
    if (score < 30)
    {
        document.write("The student is failed.");
    }
    else
    {
        document.write("The student is passed.");
    }
</SCRIPT>
</BODY>
</HTML>
```



മുകളിൽ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം കോഡ് if - else പ്രസ്താവന ചെയ്യുന്നതാണ്. പ്രോഗ്രാം മിക്ക് ഒരു പ്രവർത്തിപ്പിക്കുന്നതിനായി തിരഞ്ഞെടുക്കാം. സിച്ചിന്റെ വാക്യാലടന്നു.

**ചിത്രം 6.8: if പ്രസ്താവന വൈദിക്ഷണികൾ**

### 6.6.2 സിച്ച് (switch)

സിച്ച് ഒരു 'മൾട്ടി ബ്രോഖിൽ' പ്രസ്താവനയാണ്. ഇതുപയോഗിച്ച് ഒരു എക്സ്പ്രഷൻ വില അടിസ്ഥാനമാക്കി വ്യത്യസ്ത പ്രസ്താവന കോഡുകൾ പ്രവർത്തിപ്പിക്കുന്നതിനായി തിരഞ്ഞെടുക്കാം. സിച്ചിന്റെ വാക്യാലടന്നു.

```
switch (പരിശോധന പ്രയോഗങ്ങൾ)
{
    case value1:
        പ്രസ്താവനകൾ;
        break;
    case value2:
        പ്രസ്താവനകൾ;
        break;
    .....
    .....
```

+ ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ചുള്ള ശൈലീ സെഡ് സ്ക്രിപ്റ്റ്

```

default:
    പ്രസ്താവനകൾ;
}

എക്സ്പ്രഷൻ വില അടിസ്ഥാനമാക്കി ഉചിതമായ കേസ് ഭാഗം പ്രവർത്തിക്കുന്നു.
ഇവിടെ എക്സ്പ്രഷൻ ഒരു വേരിയബിളിന്റെ പേര് ആവാം. നൽകുന്ന നമ്പറിന്
അനുയോജ്യമായ റിബാൾ പ്രിൻ്റ് ചെയ്യുന്ന ഒരു HTML കോഡ് താഴെ കൊടുത്തിരിക്കുന്നു.

ഉദാഹരണം 6.8: നാലുകുന്ന നമ്പറിന് അനുയോജ്യമായ ഓവലം പ്രിൻ്റ്
        ചെയ്യുന്ന ടൈപ്പ് പേജ് നിർമ്മിക്കുന്നതിനുള്ള HTML കോഡ്

<HTML>
<HEAD> <TITLE>Javascript - switch</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    var d;
    d = 3;
    switch(d)
    {
        case 1:
            document.write("Sunday");
            break;
        case 2:
            document.write("Monday");
            break;
        case 3:
            document.write("Tuesday");
            break;
        case 4:
            document.write("Wednesday");
            break;
        case 5:
            document.write("Thursday");
            break;
        case 6:
            document.write("Friday");
            break;
        case 7:
            document.write("Saturday");
            break;
        default:
            document.write("Invalid Day");
    }
</SCRIPT>
</BODY>
</HTML>
```

ഈ കോഡിന്റെ ഒരു പുന്ന ചിത്രം 6.9 തേ നൽകിയിരിക്കുന്നു.

### 6.6.3 for ലൂപ്പ്

ഒരു കുടം നിർദ്ദേശങ്ങൾ ആവർത്തിച്ച് നടപ്പിലാക്കുവാൻ for ലൂപ്പ് ഉപയോഗിക്കുന്നു. for ലൂപ്പിന്റെ ആവർത്തനം നിയന്ത്രിക്കുന്നതിനായി ഒരു ലൂപ്പ് വേരിയബിൾ ഉപയോഗിയ്ക്കുന്നു.

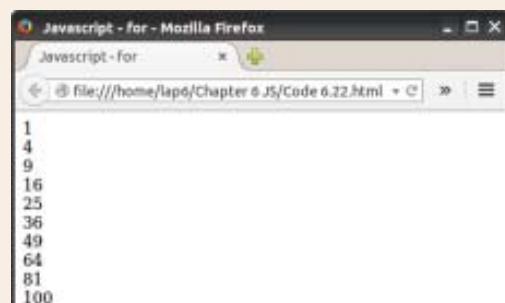
for ലൂപ്പിന്റെ വാക്യാലടം

```
for(initialisation; test_expression; update_statement)
{
    പ്രസ്താവനകൾ;
}
```

initialisation ലൂപ്പ് വേരിയബിളിന് ആദ്യ വില നൽകുവാൻ ഉപയോഗിക്കുന്നു. test\_expression ലൂപ്പ് തുടരണോ വേണ്ടയോ എന്ന പരിശോധനാനുള്ള വ്യവസ്ഥയും update\_statement ലൂപ്പ് വേരിയബിളിന്റെ വില വർധിപ്പിക്കുകയോ കുറയ്ക്കുകയോ ചെയ്യുന്നതിനുമുള്ള പ്രസ്താവനയുമാണ്. താഴെ പറയുന്ന ഉദാഹരണം ലൂപ്പിന്റെ പ്രവർത്തനങ്ങൾ വിശദീകരിക്കുന്നു.

**ഉദാഹരണം 6.9:** അരബ്യത്തെ 10 നമ്പറുകളുടെ വർഗ്ഗങ്ങൾ പ്രദർശിപ്പിക്കുന്ന തിരഞ്ഞെടുപ്പ് ഒരു വൈബ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Javascript - for</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    var i, s;
    for (i=1; i<=10; i++)
    {
        s = i*i;
        document.write(s);
        document.write(" <BR> ");
    }
</SCRIPT>
</BODY>
</HTML>
```



**ചിത്രം 6.10:** വേദിയബിളുകളുടെ തരം ഉപയോഗം പ്രദർശിപ്പിക്കുന്ന വൈബ് പേജ്.

ഇതിന്റെ ഒരു പുന്ന ചിത്രം 6.10 തേ കൊടുത്തിരിക്കുന്നു.

ചിത്രം 6.10: ഒരു പുന്ന 'Square of 1 is 1' എന്ന ലഭിക്കുന്നതിന് മുകളിലുള്ള കോഡ് നമുക്ക് പരിഷ്കരിക്കാം.

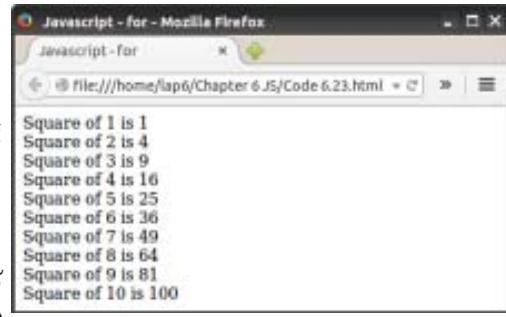


```

for (i=1; i<=10; i++)
{
    s = i*i;
    document.write("Square of "
    " + i + " is " + s);
    document.write(" <BR> ");
}

```

ഒരു പൂട്ട് നിർമ്മിക്കുന്നതിന് സ്ട്രിംഗ് കൂട്ടിച്ചേർക്കുന്നതിനുള്ള (സ്ട്രിംഗ് അധിഷ്ഠണ) ഓപ്പറേറ്റർ ആയി + ഉപയോഗിക്കുന്നു.



ചിത്രം 6.11: നമ്മുകളുടെ വർദ്ധം പ്രവർത്തിപ്പിക്കുന്നതിനുള്ള ബൈബ്‌പ്ലേജ്.

#### 6.6.4 while ലൂപ്പ്

ഒരു വ്യവസ്ഥയുടെ അടിസ്ഥാനത്തിൽ ഒരു കുടം പ്രസ്താവനകൾ ആവർത്തിച്ച് പ്രവർത്തിപ്പിക്കുന്നതിനാണ് ലൂപ്പ് എന്ന് പറയുന്നത്.

```

while (test_expression)
{
    statements;
}

```

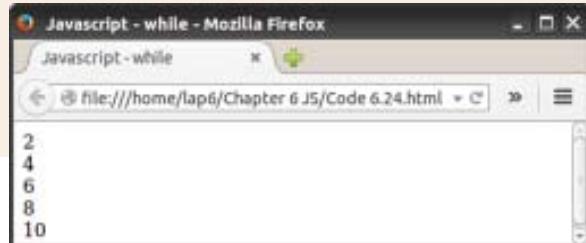
ഇവിടെ test\_expression എന്നത് ഒരു വ്യവസ്ഥയാണ്. വ്യവസ്ഥ ശരിയായിരിക്കുന്നിട്ടേതാണോ ലൂപ്പിനുള്ളിലെ പ്രസ്താവനകൾ തുടർച്ചയായായി പ്രവർത്തിച്ചുകൊണ്ടിരിക്കും. താഴെക്കാണുന്ന ഉദാഹരണം 10 വരെയുള്ള ഇട സംഖ്യകൾ പ്രദർശിപ്പിക്കുന്നു.

**ഉദാഹരണം 6.10:** 10 വരെയുള്ള ഇട സംഖ്യകൾ പ്രദർശിപ്പിക്കുന്നതിനുള്ള വൈഡോജ് നിർമ്മാണം.

```

<HTML>
<HEAD> <TITLE>Javascript - while</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
    var i;
    i = 2;
    while (i<=10)
    {
        document.write(i);
        document.write(" <BR> ");
        i += 2;
    }
</ SCRIPT>
</ BODY>
</ HTML>

```



ചിത്രം 6.12: while ലൂപ്പിലെ ഉപയോഗത്തെ വ്യക്തമാക്കുന്ന ബൈബ്‌പ്ലേജ്

തന്നിരിക്കുന്ന കോഡിംഗ് ഓട്ടപുട്ട് ചിത്രം 6.12 തേ നൽകിയിരിക്കുന്നു.

ജാവാസ്ക്രിപ്റ്റിൽ C++ ലേതിന് സമാനമായ do while ലൂപ്പ് ലഭ്യമാണ്. മിക്കവാറും എല്ലാ ജോലികളും for ലൂപ്പിലും while ലൂപ്പിലും ചെയ്യാമെന്നതിനാൽ ഈ അധ്യായത്തിൽ നമൾ മറ്റ് ലൂപ്പുകൾ ചർച്ച ചെയ്യുന്നില്ല.



ഉദാഹരണം 6.14 തേ document.write ("<BR>"); എന്ന പ്രസ്താവന ഒഴിവാക്കുകയാണെങ്കിൽ എന്ത് ഓട്ടപുട്ട് ലഭിക്കും;

### നമ്മൾ വെയ്ക്കും

#### നിങ്ങളുടെ പുരോഗതി അറിയാം



1. ഒരു കുട്ടി പ്രസ്താവനകൾ തിരഞ്ഞെടുത്തത് പ്രവർത്തിപ്പിക്കുന്നതിന് \_\_\_\_\_ ഉം \_\_\_\_\_ ഉം കൺട്രോൾ സ്ട്രക്ചറുകൾ ഉപയോഗിക്കുന്നു.
2. ജാവാസ്ക്രിപ്റ്റിലെ ലൂപ്പിങ് പ്രസ്താവനകൾക്ക് ഉദാഹരണങ്ങൾ എഴുതുക.
3. \_\_\_\_\_ ഒരു മൾട്ടി ബ്രോഡ്കിസ്റ്റ് പ്രസ്താവനയാണ്.
4. ശരിയോ തെറ്റോ എന്നെഴുതുക.
  - a. break പ്രസ്താവന switch ബ്ലോക്കിനുള്ളിൽ ഉപയോഗിക്കുന്നു.
  - b. switch, പ്രസ്താവന ഉപയോഗിക്കുകയാണെങ്കിൽ, ചുരുങ്ങിയത് ഒരു തവണയെങ്കിലും അതിൽ break പ്രസ്താവന ഉപയോഗിച്ചിരിക്കും.
  - c. if-else ഉപയോഗിച്ച് എഴുതിയ പ്രോഗ്രാമുകൾ എല്ലാം സിച്ച് പ്രസ്താവന ഉപയോഗിച്ച് മാറ്റി എഴുതാം.
  - d. switch പ്രസ്താവന ഉപയോഗിച്ച് എഴുതിയ പ്രോഗ്രാമുകൾ എല്ലാം if-else ഉപയോഗിച്ച് മാറ്റി എഴുതാം.
5. for ലൂപ്പും while ലൂപ്പും തമ്മിലുള്ള വ്യത്യാസം എന്താണ്?

#### 6.7 അന്തരാ നിർമ്മിത ഫംഷൻകൾ (Built-in function)

ജാവാസ്ക്രിപ്റ്റിൽ ധാരാളം അന്തർന്നിർമ്മിത ഫംഷൻകൾ ലഭ്യമാണ്. ഫംഗ്ഷനുകളെ മെത്രേയ്യ് എന്നും വിളിക്കുന്നു. സാധാരണയായി ഉപയോഗിക്കുന്ന ഏതാനും ഫംഗ്ഷനുകളെക്കുറിച്ച് ചർച്ച ചെയ്യാം.

##### a. alert()

സ്ക്രീനിൽ ഒരു സന്ദേശം പ്രദർശിപ്പിക്കാൻ ഇള ഫംഷൻ ഉപയോഗിക്കുന്നു.

ഉദാഹരണത്തിന് :

```
alert("Welcome to JavaScript");
```

Welcome to JavaScript

OK

ചിത്രം 6.3: സന്ദേശാലക്ഷം



എന്ന പ്രസ്താവന ഉപയോഗിച്ച് ചിത്രം 6.13 ത്ത് കാണിച്ചിരിക്കുന്ന സന്ദേശജാലകം ബൈബസറിൽ കാണിക്കാം. ഡാറ്റയുടെ സാധുത ഉറപ്പുവരുത്തുന്ന സമയത്ത് ഉപയോകതാവിന് സന്ദേശം നൽകുന്നതിന് ഈ ഫംശൻ ഉപയോഗിക്കുന്നു.

### b. isNaN()

തനിരിക്കുന്ന വില ഒരു സംവ്യയാണോ അല്ലയോ എന്നു പരിശോധിക്കുന്നതിന് ഈ ഫംശൻ ഉപയോഗിക്കുന്നു. ഈ ഫംശനിൽ, NaN എന്നത് നോട് ഏ നമ്പർ എന്ന സൂചിപ്പിക്കുന്നു. നൽകിയിരിക്കുന്ന വില ഒരു സംവ്യയല്ലെങ്കിൽ ഫംശൻ true എന്ന തിരികെ നൽകുന്നു. ഉദാഹരണത്തിന്,

1. isNaN("welcome");
2. isNaN("A123");
3. isNaN("Score50");
4. isNaN("A");

താഴെ പറയുന്ന പ്രസ്താവനകൾ false എന്ന വില തിരികെ നൽകുന്നു.

1. isNaN("13");
2. isNaN(13);
3. isNaN("13.5");
4. isNaN("0.123");

താഴെ പറയുന്ന പ്രസ്താവന ചിത്രം 6.14 ത്ത് കാണിച്ചിരിക്കുന്ന സന്ദേശജാലകം ബൈബസറിൽ പ്രദർശിപ്പിക്കുന്നു.

```
alert(isNaN("A"));
```

ഡാറ്റയുടെ സാധുകരണത്തിനായി ഈ ഫംശൻ വളരെ ഉപയോഗപ്രദമാണ്. ഉദാഹരണത്തിന്, ഒരു വിദ്യാർഥിയുടെ പ്രായം നൽകുന്നതിന് വെബ് പേജിൽ ഒരു ടെക്സ്റ്റ് ബോക്സ് ഉണ്ട് എന്ന് കരുതുക, അബവഭത്തിൽ അതിൽ സംവ്യക്ക് പകരം ഒരു ക്യാരക്ക് നൽകിയേക്കാം. അപ്പോൾ ഈ ഫംശൻ, ഇൻപുട്ട് നൽകിയത് ഒരു നമ്പർ ആണോയെന്ന് പരിശോധിക്കുകയും ഈത് ഒരു സംവ്യയല്ലെങ്കിൽ, alert() ഫംശൻ ഉപയോഗിച്ച് ഒരു സന്ദേശം പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു.



ചിത്രം 6.14: NaN() ഫംശൻ പുരുത്വം

### c. toUpperCase()

ഈ ഫംഗ്ഷൻ തനിരിക്കുന്ന സ്ട്രീംഗിനെ അപുർ കേസാക്കി മാറ്റുന്നു. താഴെ തനിരിക്കുന്ന ഉദാഹരണം നോക്കുക.

```
var x, y;
x = "JavaScript";
y = x.toUpperCase();
alert(y);
```

ഈ കോഡിന്റെ ഒരു പുന്ന ചിത്രം 6.15 ത്ത് കാണിച്ചിരിക്കുന്നു. `toUpperCase()` എന്ന ഫലം ഷഡിംഗ് വേറിയവിൽ x എന്ന പേരിനോട് ചേർത്ത് വിളിക്കുന്നു. അതായത് `toUpperCase()` എന്നത് സ്ക്രിപ്റ്റിനു അപേക്ഷാക്രമി തിരികെ നൽകുന്നു. ജാവസ്ക്രിപ്റ്റ് കേസ് സൈൻസിറ്റിവ് ഭാഷയാണ്. അതിനാൽ കോഡിൽ ഫലം ഷഡിംഗ് എന്നു കേസിലാണോ അതേ രീതിയിൽ തന്നെ ഉപയോഗിക്കണം.

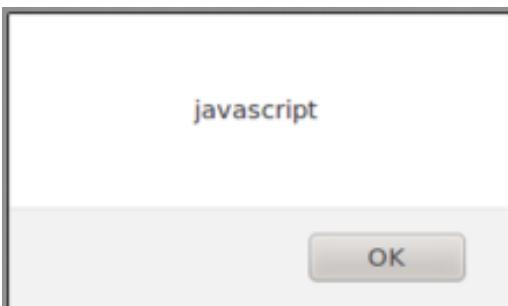
ചിത്രം 6.15: `LowerCase()` എൻ ഒരു പുന്ന

#### d. `toLowerCase()`

ഈ ഫലം ഷഡിംഗിനു ലോവർ കേസാക്രമി തിരികെ നൽകുന്നു. താഴെ നൽകിയിരിക്കുന്ന ഉദാഹരണം ശ്രദ്ധിക്കുക

```
var x, y;
x = "JavaScript";
y = x.toLowerCase();
alert(y);
```

മുകളിലെ കോഡിന്റെ ഒരു പുന്ന ചിത്രം 6.16 ത്ത് കാണിച്ചിരിക്കുന്നു. സ്ക്രിപ്റ്റിൽ എല്ലാ ക്യാരക്കുകളും ഇപ്പോൾ ചെറിയ അക്ഷരത്തിലാണ് അതിനാൽ `toLowerCase()` അതേ സ്ക്രിപ്റ്റ് തിരികെ നൽകുന്നു.

ചിത്രം 6.16: `toLowerCase()` എൻ ഒരു പുന്ന

#### e. `charAt()`

ഈ ഫലം ഷഡിംഗ് രേഖയുക സ്ഥാനത്തുള്ള ക്യാരക്കൾ തിരികെ നൽകുന്നു. `charAt(0)` എന്നത് സ്ക്രിപ്റ്റിലെ ആദ്യത്തെ അക്ഷരം തിരികെ നൽകുന്നു. `charAt(1)` സ്ക്രിപ്റ്റിലെ രണ്ടാമത്തെ അക്ഷരം തിരികെ നൽകുന്നു. ഉദാഹരണം നോക്കുക.

```
var x;
x = "JavaScript";
y = x.charAt(4);
alert(y);
```

ഇതിന്റെ ഒരു പുന്ന ചിത്രം 6.17 ത്ത് നൽകിയിരിക്കുന്നു. വേറിയവിൽ x ലെ സ്ക്രിപ്റ്റിന്റെ അഞ്ചാമത്തെ അക്ഷരം 'S' ആയതിനാൽ പ്രൗഢാർ ജാലകത്തിൽ B അക്ഷരം കാണിക്കുന്നു.



### f. length പ്രോപ്പറ്റി

ഹണ്പചനുകൾ കൂടാതെ, പ്രോഗ്രാമർമ്മാർക്ക് ഉപയോഗപ്രദമായ ചില പ്രോപ്പറ്റികളും സ്ട്രിങ് വേരിയബിളിനോടൊപ്പം ഉപയോഗിക്കാം. length പ്രോപ്പറ്റി സ്ട്രിംഗിന്റെ നീളം തിരികെ നൽകുന്നു. സ്ട്രിംഗിലെ ക്യാരട്ടുകളുടെ എണ്ണമാണ് നീളം എന്നതു കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. ഉദാഹരണത്തിന്

ചിത്രം 6.17: charAt () റിസൾട്ട്

```
var x, n;  
x = "JavaScript";  
n = x.length;  
alert(n);
```

ഈവിടെ length പ്രോപ്പറ്റി എങ്ങനെ വിളിക്കുന്നു എന്ന നോക്കാം. x.length എന്ന രൂപത്തിൽ x എന്ന വേരിയബിളിന്റെ പേര് ചേർത്തതാണ് ഈ പ്രോപ്പറ്റി വിളിക്കുന്നത്. മുകളിലെ കോഡിന്റെ ഒരു പുരുഷ് ചിത്രം 6.18 തെ കാണിച്ചിരിക്കുന്നു. ഹണ്പചനും പ്രോപ്പറ്റിയും തമ്മിലുള്ള പ്രധാന വ്യത്യാസം, ഹണ്പചനോ ടോപ്പ് () ഭ്രാക്കറ്റ് ഉപയോഗിക്കുന്നു (ചില സന്ദർഭങ്ങളിൽ ഭ്രാക്കറ്റിൽ പരാമീറ്ററുകളും) എന്നാൽ പ്രോപ്പറ്റിയോടൊപ്പ് () ഭ്രാക്കറ്റിന്റെ ആവശ്യം ഇല്ല എന്നതാണ്.



ചിത്രം 6.18: length പ്രോപ്പറ്റിയുടെ റിസൾട്ട്

### നിങ്ങളുടെ പ്രവേശനത്തി അറിയാം

താഴെപ്പറയുന്നവയിൽ വേരിയബിൾ y യുടെ വില എഴുതുക.



1. x = "welcome";
2. y = x.length;
3. x = "WELCOME";
4. y = x.toLowerCase();
5. x = "Welcome";
6. y = x.toUpperCase();
7. x = "welcome";
8. y = x.toLowerCase();
9. x = "welcome";
10. y = isNaN(x);
11. x = "welcome";
12. y = charAt(3);

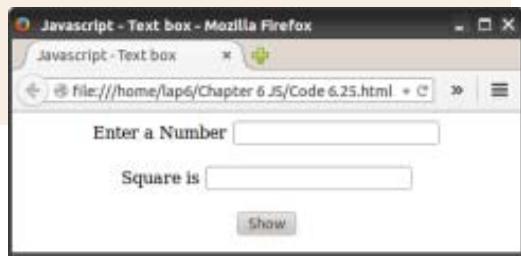
### 6.8 ജാവാസ്ക്രിപ്റ്റിലുള്ള ടെക്നോളജീകൾ വിലക്കൾ<sup>1</sup> ഉപയോഗിക്കുന്നവിധം

മുൻ അധ്യായത്തിൽ ഒരു വെബ് പേജിൽ ടെക്നോളജീ ബോക്സ്, ചെക്ക്കോക്സ്, റേഡിയോ ബോക്സ്, സബ്മിറ്റ് ബോക്സ് തുടങ്ങിയ വിവിധ കൺട്രോളൗകൾ (controls) എങ്ങനെ ഉപയോഗിക്കാം എന്ന് പറിച്ചു. ഈ തരം കൺട്രോളൗകൾ ജാവാസ്ക്രിപ്റ്റിൽ സഹായത്തിൽ എങ്ങിനെ ഉപയോഗിക്കാം എന്നാണ് ഇവിടെ ചർച്ചചെയ്യുന്നത്. ഈ അധ്യായത്തിൽ നമ്മൾ ചർച്ച ചെയ്ത ഒരു പ്രോഗ്രാമിലും ഉപയോകതാവിൽ നിന്ന് ധാരാളായവിധി ഇൻഫോറ്മേഷൻ സൈറ്റുകളിലും പ്രോസസ്സിലും ആവശ്യമായ ഡാറ്റ നേരിട്ട് പ്രോഗ്രാം കോഡിൽ തന്നെ നൽകുകയാണ് ചെയ്തത്. ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് ഒരു വെബ് പേജ് ഘടകങ്ങളെ എങ്ങനെ സൈറ്റുകളും എന്ന് നോക്കാം. ഈ പാരഭേഗത്തിനുശേഷം നമുക്ക് ശരിയായ ഒരു ഇൻററാക്ടീവ് വെബ് പേജ് രൂപപ്പെടുത്തുവാൻ കഴിയും. അതായത്, ഉപയോകതാവ് ടെക്നോളജീ ബോക്സ് സൈറ്റിൽ വില നൽകാനും അതിനേൽക്കേ ചില പ്രോസസിലും നടത്തി റിസൾട്ട് മറ്റാരു ടെക്നോളജീ ബോക്സ് സൈറ്റിൽ പ്രദർശിപ്പിക്കാനും കഴിയും. ഈ കാര്യങ്ങൾ മനസ്സിലാക്കുന്നതിന് താഴെ കൊടുത്തിരിക്കുന്ന HTML കോഡ് ശ്രദ്ധാപൂർവ്വം വായിക്കുക.

**ഉദാഹരണം 6.11:** ഒരു വെബ് ഫോം (web form) പ്രദർശിപ്പിക്കുന്ന ഒരു വെബ് പേജ് നിർമ്മാണം.

```
<HTML>
<HEAD> <TITLE>Javascript - Text box</TITLE> </HEAD>
<BODY>
    <FORM Name= "frmSquare">
        <CENTER>
            Enter a Number
            <INPUT Type= "text" Name= "txtNum">
            <BR><BR>
            Square is
            <INPUT Type= "text" Name= "txtSqr">
            <BR><BR>
            <INPUT Type= "button" Value= "Show">
        </CENTER>
    </FORM>
</BODY>
</HTML>
```

ഈ കോഡിൽ റിസൾട്ട്, ചിത്രം 6.19 തോന്തരം കൊടുത്തിരിക്കുന്നു. ഈ വെബ് ഫോം നിന്ന് frmSquare എന്നും, ടെക്നോളജീ ബോക്സുകൾക്ക് txtNum, txtSqr എന്നും ദിക്കാക്കാം.



ചിത്രം 6.19: വെബ് ഫോം പ്രദർശിപ്പിക്കുന്ന ഒരു വെബ് പേജ്

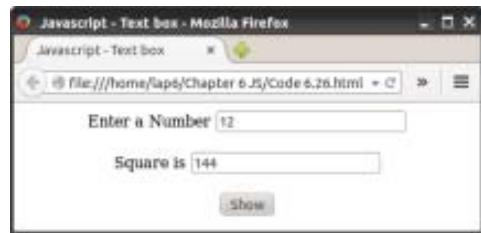


എന്നും പേരുകൾ നൽകിയിരിക്കുന്നു. ജാവസ്ക്രീപ്റ്റിൽ ഇത്തരം ഘടകങ്ങൾക്ക് ഉപയോഗിക്കുന്നതിന് അവയ്ക്ക് പേര് നൽകേണ്ടത് വളരെ പ്രധാനമാണ്. ഒരു വെബ് പേജ് ഘടകത്തിന് നമ്മൾ പേര് നൽകുന്നില്ലെങ്കിൽ, ജാവസ്ക്രീപ്റ്റിന് ഇവയെ ഉപയോഗിക്കാൻ കഴിയില്ല. Show എന്ന് എഴുതിയിരിക്കുന്ന സബ്മിറ്റ് ബട്ടന് പേര് നൽകിയിട്ടില്ലെന്നത് ശ്രദ്ധിക്കുക. കാരണം, ഈ ബട്ടൺ ജാവസ്ക്രീപ്റ്റിൽ നിന്ന് പരാമർശിക്കാൻ കഴിയില്ല.

ഈ നമ്മുകൾ ഈ പ്രോഗ്രാം കോഡിൽ കൂടിച്ചു മറ്റൊക്കെന്നും വരുത്തി നോക്കാം.

**ഉദാഹരണം 6.12:** നൽകുന്ന സംവ്യയ്യുടെ വരുഗം (സ്ക്രിപ്റ്റിനും തീരുമാളിക്കുന്ന ഒരു വെബ് പേജ് നിർമ്മാണം.

```
<HTML>
<HEAD> <TITLE>Javascript - Text box</TITLE>
<SCRIPT Language= "JavaScript">
    function showSquare()
    {
        var num, ans;
        num = document frmSquare.txtNum.value;
        ans = num * num;
        document frmSquare.txtSqr.value = ans;
    }
</SCRIPT>
</HEAD>
<BODY>
<FORM Name= "frmSquare">
<CENTER>
    Enter a Number
    <INPUT Type= "text" Name= "txtNum">
    <BR><BR>
    Square is
    <INPUT Type= "text" Name= "txtSqr">
    <BR><BR>
    <INPUT Type= "button" Value= "Show"
           onClick= "showSquare() ">
</CENTER>
</FORM>
</BODY>
</HTML>
```



ചിത്രം 6.20: ഒരു സംഖ്യയുടെ വർഗ്ഗം പ്രദർശിപ്പിക്കുന്ന തിരുന്നേ വെബ്‌പേജ്

മുകളിലുള്ള കോഡ് ശ്രദ്ധാപൂർവ്വം പരിശോധിച്ച് കുടിച്ചുവരിക്കുള്ളവ ശ്രദ്ധിക്കുക (ഉദാഹരണം 6.17). വെബ് പേജിന്റെ ഫോം ഭാഗത്ത് showSquare() എന്ന ഫലങ്ങൾ നിർവ്വചിച്ചിരിക്കുന്നു. താഴെ കോടുത്തിരിക്കുന്ന രീതിയിൽ ഈ ഫലങ്ങൾ വിളിക്കുന്നു.

```
<INPUT Type= "button" Value= "Show" onClick= "showSquare()">
onclick= "
```

എന്നത് ഇൻപുട്ട് ടാഗിന്റെ ഉള്ളിലാണ് എഴുതിയിട്ടുള്ളത്. ഉപയോകതാവ് ഈ ബട്ടൺ ക്ലിക്കുചെയ്യുമ്പോൾ, showSquare() എന്ന ഫലങ്ങൾ പ്രവർത്തിക്കുന്നു. ബട്ടനിൽ ക്ലിക്കുചെയ്തുകൊണ്ട് ഒരു ഫലങ്ങൾ എങ്ങനെ വിളിക്കാം എന്ന് മനസ്സിലായായാലോ.

ഫലങ്ങൾ നിർവ്വചനത്തിലെ താഴെപ്പറയുന്ന വരി ശ്രദ്ധിക്കുക

```
num = document.frmSquare.txtNum.value;
document.frmSquare.txtNum.value നോക്കുക. ഇവിടെ 'document' എന്നത് വെബ് പേജിന്റെ ബോധി ഭാഗത്തെ സൂചിപ്പിക്കുന്നു. frmSquare എന്നത് body ഭാഗത്ത് നൽകിയിട്ടുള്ള ഫോമിന്റെ പേരാണ്. 'txtNum' എന്നത് 'frmSquare'- ലെ ഉള്ള ടെക്റ്റ് ബോക്സിന്റെ പേരാണ്, value എന്നത് ടെക്റ്റ് ബോക്സിലെ ഉള്ളടക്കത്തെ പരാമർശിക്കുന്നു. അതായത് document.frmSquare.txtNum.value എന്നത് ഡോക്യുമെന്റിലെ frmSquare എന്ന ഫോമിലെ txtNum റെറ്റ് വിലയാണ്. മുകളിൽ പറഞ്ഞ വരി ടെക്റ്റ് ബോക്സിലെ വില വേരിയബിൽ num-ൽ ലഭ്യമാക്കും.
```

ഈ പരയുന്ന വരയുടെ അർമ്മം മനസ്സിലാക്കാൻ ശ്രമിക്കുക.

```
document.frmSquare.txtSqr.value = ans;
മുകളിലുള്ള വരി രണ്ടാമത്തെ ടെക്റ്റ് ബോക്സിൽ ans എന്ന വേരിയബിളിന്റെ വില കാണിക്കുന്നു. ബട്ടനിൽ ക്ലിക്കുചെയ്യുമ്പോൾ വെബ്‌പേജിലെ ആദ്യത്തെ ടെക്റ്റ് ബോക്സിൽ നൽകിയിരിക്കുന്ന സംഖ്യയുടെ വർഗ്ഗം രണ്ടാമത്തെ ടെക്റ്റ് ബോക്സിൽ കാണിക്കുന്നു. ഉപയോകതാവിന് ആദ്യത്തെ ടെക്റ്റ് ബോക്സിൽ ആത്മ സംഖ്യയും ടെപ്പുചെയ്യാം, അതിന്റെ വർഗ്ഗം കാണുന്നതിന് show ബട്ടൺ ക്ലിക്കുചെയ്യുക. പ്രവർത്തന സമയത്തുള്ള വെബ് പേജിന്റെ സ്ക്രീൻ ഷേഖര് ചിത്രം 6.20 ലെ കോടുത്തിരിക്കുന്നു
```

```
<INPUT Type= "button" Value= "Show" onClick= "showSquare()">
എന്നത്
<INPUT Type= "button" Value= "Show"
onMouseEnter= "showSquare()">
```

എന്നാക്കിയാൽ എന്ന സംഭവിക്കുമെന്ന് നോക്കാം. ബട്ടനുമുകളിലുടെ മഹാ പോയിന്റും നീക്കുമ്പോൾ ഫലങ്ങൾ പ്രവർത്തിപ്പിക്കുന്നതിനായി വിളിക്കുന്നു. ഫലങ്ങൾ പ്രവർത്തനത്തിനുവേണ്ടി നിങ്ങൾ ബട്ടണിൽ ക്ലിക്ക് ചെയ്യേണ്ടതില്ല. ഒരു വെബ് പേജിലെ



ചില ബന്ധനകൾ മുകളിലൂടെ മുൻ പോയിൻ്റർ നീക്കുന്നോൾ, ബന്ധങ്ങൾ നിരു മാറ്റുന്നത് നിങ്ങൾ ശ്രദ്ധിച്ചിരിക്കാം. ബന്ധങ്ങൾ നിരു മാറ്റുന്നതിനുള്ള ഒരു ജാവാസ്ക്രിപ്റ്റ് ഫണ്ട്ഷൻ എഴുതി അത് `onMouseEnter` എന്ന ഇവർത്തിലൂടെ വിളിച്ച് ഇത് സാധ്യമാക്കാം.

```
<INPUT Type="button" Value="Show" onClick="showSquare()>
```

`onMouseEnter`, `onClick`, `onMouseEnter`, `onMouseLeave`, `onKeyDown`, `onKeyUp` തുടങ്ങിയവ സാധാരണയായി ഉപയോഗിക്കുന്ന ചില ഇവർത്തികളാണ്.

സാധാരണ ഉപയോഗിക്കുന്ന ചില ഇവർത്തികളും അവയുടെ വിവരങ്ങളും പട്ടിക 6.6 തുടർത്തിരിക്കുന്നു.

ഇവർത്തി	വിവരണം
<code>onClick</code>	ഉപയോകതാവ് ഒരു ബെജക്റ്റിൽ ക്ലിക്കുചെയ്യുന്നോൾ സംഭവിക്കുന്നു
<code>onMouseEnter</code>	മുൻ പോയിൻ്റർ ഒരു ബെജക്റ്റിലേക്ക് നീങ്ങുന്നോൾ സംഭവിക്കുന്നു
<code>onMouseLeave</code>	മുൻ പോയിൻ്റർ ഒരു ബെജക്റ്റിൽ നിന്നും മാറുകയാണൊക്കിൽ സംഭവിക്കുന്നു
<code>onKeyDown</code>	ഉപയോകതാവ് കീബോർഡിലെ ഒരു കീ അമർത്തുന്നോൾ സംഭവിക്കുന്നു
<code>onKeyUp</code>	ഉപയോകതാവ് കീബോർഡിലെ ഒരു കീയിൽ നിന്ന് വിടുന്നോൾ സംഭവിക്കുന്നു

പട്ടിക 6.6: സാധാരണ ഉപയോഗിക്കുന്ന ജാവാസ്ക്രിപ്റ്റ് ഇവർത്തികൾ

ഒരു വെബ്പോജിൽ രണ്ട് വ്യത്യസ്ത ടെക്നോളജികളുടെ ഉപയോകതാവിന് രണ്ട് സംഖ്യ നൽകാനും, ഒരു ബന്ധനിൽ അമർത്തുന്നോൾ ഈ സംഖ്യകളുടെ തുക മുന്നാമത്തെ ടെക്നോളജികളുടെ കാണിക്കുന്നതിനുമുള്ള ഒരു വെബ്പോജികളിൽ നിർമ്മിക്കാം. കോഡ്, ഒരു പുറകു എന്നിവ ചിത്രം 6.21 തുടർത്തിരിക്കുന്നു.

**ഉദാഹരണം 6.13:** രണ്ട് സംഖ്യകളുടെ തുക കാണിക്കുന്ന വെബ്സൈറ്റ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Sum</TITLE>
<SCRIPT Language= "JavaScript">
    function showSum()
    {
        var num1, num2, ans;
        num1 = document.frmSum.txtNum1.value;
        num2 = document.frmSum.txtNum2.value;
        ans = num1 + num2;
        document.frmSum.txtSum.value = ans;
    }

```



```
</SCRIPT>
</HEAD>
<BODY>
<FORM Name= "frmSum">
    <CENTER>
        Enter a Number 1
        <INPUT Type= "text" Name= "txtNum1">
        <BR><BR>
        Enter a Number 2
        <INPUT Type= "text" Name= "txtNum2">
        <BR><BR>
        The sum is
        <INPUT Type= "text" Name= "txtSum">
        <BR><BR>
        <INPUT Type= "button" Value= "Show" onClick= "showSum( )">
    </CENTER>
</FORM>
</BODY>
</HTML>
```

ഈ പ്രോഗ്രാമിന് ഇൻപുട്ടായി 10, 20 എന്നീ സംവ്യൂകൾ നൽകിയപ്പോൾ റിസൾട്ടായി 1020 എന്നാണ് കിട്ടിയത് (ചിത്രം 6.21 ലെ കാണിച്ചിരിക്കുന്നു). സ്ട്രിങ്ങുകൾ കൂടിചേർക്കുന്നതിന് + ഓപ്പറേറ്റർ ഉപയോഗി ക്കുന്നു എന്നതിനാലാണിത്. നമ്മൾ ഒരു സ്ട്രിങ്ങുകൾ “10” + “20” ചേർക്കുന്നോൾ ഉത്തരം ‘1020’ ആണ്ടോ. ഒരു, ഒക്ലൂൾ വോക്സിൽ തന്ത്രവില എല്ലായ്പ്പോഴും സ്ട്രിങ്ങായിരിക്കും അതിനാൽ ഒക്ലൂൾ വോക്സിലെ ഉള്ളടക്കം ഒരു നമ്പർ ആയിരുന്നാലും ഒരു വേറിയബിളിലേക്ക് നൽകുന്നോൾ അതിനെ സ്ട്രിംഗ് ആയി മാത്രമേ പരിഗണിക്കും. ഒരു സംവ്യൂകളുടെ തുക കിട്ടുന്നതിന് ഫംഗഷൻ showSum() താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ പരിഷ്കരിക്കാം.

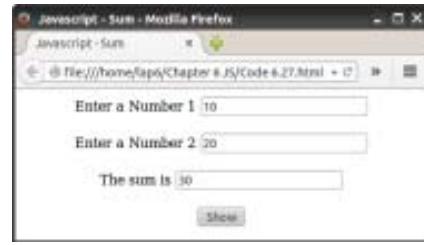
```
function showSum()
{
    var num1, num2, ans;
    num1 = Number(document.frmSum.txtNum1.value);
    num2 = Number(document.frmSum.txtNum2.value);
    ans = num1 + num2;
    document.frmSum.txtSum.value = ans;
}
```



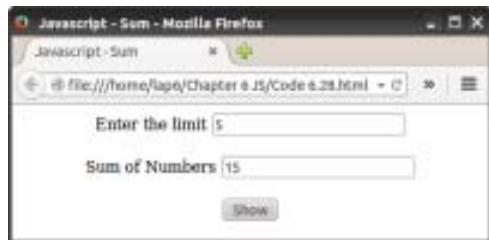
ഇവിടെ Number() എന്ന ഫലങ്ങൾ , ഡാറ്റയെ  
സംഖ്യയായി മാറ്റി ആ സംഖ്യ psum1 എന്ന  
വേദിയബില്ലിലേക്ക് നൽകുന്നു. ഇവിടെ psum1, psum2  
എന്നിവയെ നമ്പർ തരം ഡാറ്റയായി കണക്കാക്കു  
കയും അതുവഴി ശരിയായ തുക കിട്ടുകയും ചെയ്യും.  
ഉദാഹരണം 6.13 ലെ showSum() എന്ന ഫലങ്ങൾക്കിൽ  
മാറ്റം വരുത്തിയാൽ ചിത്രം 6.22 ലെ കാണിച്ചിരിക്കുന്ന  
വെബ് പേജ് ഒരുപുട്ടായി ലഭിക്കുന്നു. തന്നിരിക്കുന്ന  
പരിധിവരെയുള്ള സംഖ്യകളുടെ തുക പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ്  
നിർമ്മിക്കാം. ഈതിൽ പരിധി ഉപയോകതാവിന് ഒക്ടസ്റ്റ് ബോക്സിലുടെ നൽകാൻ  
കഴിയണം.

**ഉദാഹരണം 6.14:** തന്നിരിക്കുന്ന പരിധിവരെയുള്ള സംഖ്യകളുടെ തുക  
കാണിക്കുന്നതിനുള്ള വെബ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Sum</TITLE>
      <SCRIPT Language= "JavaScript">
          function sumLimit()
          {
              var sum = 0, i, limit;
              limit = Number(document.frmSum.txtLimit.value);
              for(i = 1; i <= limit; i++)
                  sum += i;
              document.frmSum.txtSum.value = sum;
          }
      </SCRIPT>
</HEAD>
<BODY>
    <FORM Name= "frmSum">
        <CENTER>
            Enter the limit
            <INPUT Type= "text" Name= "txtLimit">
            <BR><BR>
            Sum of Numbers
            <INPUT Type= "text" Name= "txtSum">
            <BR><BR>
            <INPUT Type= "button" Value= "Show"
                   onClick= "sumLimit()">
        </CENTER>
    </FORM>
</BODY>
</HTML>
```



ചിത്രം 6.22: ഒൻസംഖ്യകളുടെ തുക  
കാണിക്കുന്ന വെബ് പേജ്



ചിത്രം 6.23: നൽകിയ പരിധിവരുളയുള്ള സംഖ്യകളുടെ തുക കാണിക്കുന്ന ബെബ്ലീപ്പ്

മുകളിലെ കോഡിന്റെ ഒരു പുതിയ ചിത്രം 6.23 തോന്തരം നൽകിയിരിക്കുന്നു. ഈ വെബ്സൈറ്റ് പേജിൽ ആദ്യത്തെ ടെക്സ്റ്റ് ഫോക്സിൽ ഉപയോകതാവും പരിധി നൽകണം. അതിനി ശേഷം **show** ബട്ടണിൽ ക്ലിക്കുചെയ്യുമ്പോൾ നൽകിയ പരിധി വരെയുള്ള സംഖ്യകളുടെ തുക രണ്ടാമത്തെ ടെക്സ്റ്റ് ഫോക്സിൽ കാണിക്കുന്നു. ആദ്യത്തെ ടെക്സ്റ്റ് ഫോക്സിൽ ഉപയോകതാവും പരിധി നൽകണില്ല. പരിധി ഇല്ലെങ്കിൽ തുക പുജ്യമായിരിക്കും കാരണം ആദ്യത്തെ ടെക്സ്റ്റ് ഫോക്സിൽ ശുന്നമാകുമ്പോൾ Number() ഫലംഷൻ ശുന്ന വിലയെ പുജ്യമായി മാറുന്നു. അതായത് document.frmSum.txtLimit.value എൻ വില പുജ്യമായിരിക്കും. അങ്ങനെ ലൂപ്പ് പ്രവർത്തിക്കാതിരിക്കുകയും sum എന്ന വേരിയബിലിന്റെ ആദ്യ വിലയായ പുജ്യം രണ്ടാമത്തെ ടെക്സ്റ്റ് ഫോക്സിൽ പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. ജാവാസ്ക്രിപ്ട് റിക്കല്യൂം ഒരു തെറ്റ് ഉണ്ടെന്ന് കാണിച്ച് തരില്ല. സ്ക്രിപ്ട് എഞ്ചിന് എന്നെങ്കിലും നിർദ്ദേശം പ്രാവർത്തികമാക്കാൻ കഴിയാത്തപ്പോൾ, അത് ഫലംഷൻിലെ ആ വരിയും ബാക്കി ഭാഗങ്ങളും അവഗണിക്കും. താഴെ കൊടുത്തിരിക്കുന്ന sumLimit() ഫലംഷൻ പരിശോധിക്കാം.

```
function sumLimit()
{
    var sum = 0, i, limit;
    if (document.frmSum.txtLimit.value == "")
    {
        alert("Please enter the limit!");
        return;
    }
    limit = Number(document.frmSum.txtLimit.value);
    for(i=1; i<=limit; i++)
        sum += i;
    document.frmSum.txtSum.value = sum;
}
```

പരിധി നൽകാതെ നിങ്ങൾ show ബട്ടണിൽ ക്ലിക്കുചെയ്താൽ, ഈ കോഡ് പരിധി നൽകാൻ ഓർമ്മപ്പെടുത്തുന്ന ഒരു സന്ദേശം കാണിക്കും. ബാക്കി നിർദ്ദേശങ്ങൾ ഒഴിവാക്കിക്കൊണ്ട് ഫലംഷൻിൽ നിന്ന് പുറത്തുകടക്കുവാൻ return പ്രസ്താവന ഉപയോഗിക്കുന്നു. ഇത് C++ തോന്തരം ഉപയോഗിക്കുന്ന റിട്ടോൺ പ്രസ്താവനയ്ക്ക് സമാനമാണ്. ഇനി നമ്മൾ ടെക്സ്റ്റ് ഫോക്സിൽ നൽകിയ പരിധി അക്ഷരമാന്മാരും സംഖ്യയാണോ എന്ന് പരിശോധിക്കുന്നതിനായി isNaN() എന്ന ഫലംഷൻ ഉപയോഗിക്കാം. ഉപയോകതാവും ശരിയായ ധാര നൽകിയിട്ടുണ്ടോ എന്ന് പരിശോധിക്കുന്നതിന് ഇത് സഹായിക്കുന്നു.



```
function sumLimit()
{
    var sum = 0, i, limit;
    if (document.frmSum.txtLimit.value == "") {
        alert("Please enter the limit!");
        return;
    }
    if (isNaN(document.frmSum.txtLimit.value)) {
        alert("Please enter a number as the limit!");
        return;
    }
    limit = Number(document.frmSum.txtLimit.value);
    for(i = 1; i <= limit; i++)
        sum += i;
    document.frmSum.txtSum.value = sum;
}
```

ഈക്കുന്ത് ഭാഗത്തെ ഡാറ്റയുടെ സാധൃത വിലയിരുത്തുന്നതിനായി ജാവ സ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്നു. ആവശ്യമുള്ള എല്ലാ ഡാറ്റയും നൽകിയിട്ടുണ്ടാ, നൽകിയ ഡാറ്റ ശരിയായ ഫലനയിലാണോ തുടങ്ങിയവ പരിശോധിക്കാനായി ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കാം. ഏതെങ്കിലും പോരായ്മ ഉണ്ടെങ്കിൽ അത് ഓർമ്മിപ്പിക്കുന്ന ഒരു സന്ദേശം കാണിക്കുവാൻ കഴിയും. താഴെക്കാടുത്തിരിക്കുന്ന ഉദാഹരണത്തിൽ ജാവാസ്ക്രിപ്റ്റിലെ ഒരു ഡ്രോപ്പ്യൂണൾ ലിസ്റ്റ് ഉപയോഗിക്കുന്നു ഈ ഡ്രോപ്പ്യൂണൾ ലിസ്റ്റ് ഉപയോഗിച്ച് ഒരു സംസ്ഥാനം തെരഞ്ഞെടുക്കാനും show ബട്ടണിൽ ക്ലിക്കേച്ചുവോൾ, തെരഞ്ഞെടുത്ത സംസ്ഥാനത്തിൽ തലസ്ഥാനം ഒരു ടെക്റ്റ് ബോക്സിൽ പ്രദർശിപ്പിക്കാനുമുള്ള ഒരു വെബ്പോജ്യ നൽകിയിരിക്കുന്നു. ഇതിന്റെ ഒരുപുത്ര ചിത്രം 6.24 ലെ കൊടുത്തിരിക്കുന്നു.

**ഉദാഹരണം 6.15:** ഒരു സംസ്ഥാനത്തിന്റെ തലസ്ഥാനം പ്രദർശിപ്പിക്കുന്ന തിരുത്തു ഒരു വെബ്പോജ്യ നിർമ്മാണം.

```
<HTML>
<HEAD> <TITLE>Javascript - switch</TITLE>
<SCRIPT Language= "JavaScript">
function capital()
{
    var n, answer;
    n = document.frmCapital.cboState.selectedIndex;
    switch (n)
    {
```

കമ്പ്യൂട്ടർ സയൻസ് - 12

```

case 0:
    answer = "Thiruvananthapuram";
    break;
case 1:
    answer = "Bengaluru";
    break;
case 2:
    answer = "Chennai";
    break;
case 3:
    answer = "Mumbai";
    break;
}
document frmCapital.txtCapital.value = answer;
}
</SCRIPT>
</HEAD>
<BODY>
<FORM Name= "frmCapital">
<CENTER> State
<SELECT Size= 1 Name= "cboState">
<OPTION>Kerala</OPTION>
<OPTION>Karnataka</OPTION>
<OPTION>Tamilnadu</OPTION>
<OPTION>Maharashtra</OPTION>
</SELECT>
<BR><BR>
Capital
<INPUT Type= "text" Name= "txtCapital">
<BR><BR>
<INPUT Type= "button" Value= "Show" onClick= "capital()">
</CENTER>
</FORM>
</BODY>
</HTML>
```



ചിത്രം 6.24: സിസ്റ്റമാന്തരിക്ക് തലമുറക്ക്  
പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ് പ്രോഗ്രാം



ഈ പ്രോഗ്രാമിലെ താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന പരിഗണിക്കുക.

```
document frmCapital.cboState.selectedIndex;
```

ഇവിടെ 'cboState' എന്നത് ഡ്രോപ്പ്‌ഡോൾ ലിസ്റ്റിന്റെ പ്രതിഫലം. 'selectedIndex' എന്നത് ഡ്രോപ്പ്‌ഡോൾ ലിസ്റ്റിന്റെ സൂചിക കാണിക്കുന്നു. ഒന്നാമത്തെ ഈനും തിരഞ്ഞെടുത്താൽ സൂചിക 0-ഉം രണ്ടാമത്തെ ഈനും തിരഞ്ഞെടുത്താൽ, സൂചിക 1 ഉം ആയി കണക്കാക്കുന്നു. മുകളിലെ വരി തിരഞ്ഞെടുത്ത വേറിയബിളിന്റെ ഇൻഡക്സ് വേറിയബിൾ റ തുണ്ടാക്കുന്നു. വിദ്യാർഥിയുടെ പേരും പ്രായവും നൽകാൻ ഉപയോകതാവിനെ അനുവദിക്കുന്ന വെബ് പേജാണ് താഴെ കൊടുത്തിരിക്കുന്നത്. പേരിൽ കുറഞ്ഞത് 5 അക്ഷരങ്ങൾ ഉണ്ടായിരിക്കണം. പ്രായം 15 മുതൽ 20 വരെ ഉള്ള ഒരു സംഖ്യയായിരിക്കണം.

**ഉദാഹരണം 6.16:** പെതിരുർധ്യം പ്രായത്തിരുർധ്യം സാധ്യത പതിശായിക്കുന്ന വെബ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Validation</TITLE>
<SCRIPT Language= "JavaScript">
function checkData()
{
    var T_name, T_age, N_age;
    T_name = document frmValid.txtName.value;
    if (T_name == "")
    {
        alert("Please enter name!");
        return;
    }
    if (T_name.length < 5)
    {
        alert("Name must contain at least 5 characters!");
        return;
    }
    T_age = document frmValid.txtAge.value;
    if (T_age == "")
    {
        alert("Please enter age!");
        return;
    }
    if (isNaN(T_age))
    {
        alert("Please enter a number as the age!");
        return;
    }
}
```



```
N_age = Number(T_age);
if (N_age < 15 || N_age > 20)
{
    alert("The age must be between 15 and 20!");
    return;
}
}
</SCRIPT>
</HEAD>
<BODY>
<FORM Name= "frmValid">
<CENTER>Name
<INPUT Type= "text" Name= "txtName">
<BR><BR>
Age
<INPUT Type= "text" Name= "txtAge">
<BR><BR>
<INPUT Type= "button" Value= "Save"
onClick= "checkData()">
</CENTER>
</FORM>
</BODY>
</HTML>
```

HTML കോഡിന്റെ ഒരു പാർസ്സ് ചിത്രം 6.25 തേ  
കൊടുത്തിരിക്കുന്നു. ഡാറ്റയുടെ എല്ലാ  
സാധ്യതകളും ഇത് പരിശോധിക്കുന്നു. ചിത്രം 6.25: ലൂപ്പിന്റെ ഉപയോഗം വിശദിക്കിക്കുന്ന ബോർഡ്  
അടിസ്ഥാനം, പേരിന്റെ ഫീൽഡിൽ ഒരു വില  
ഉണ്ടാ എന്ന് പരിശോധിക്കുന്നു. പിന്നീട് പേരിന്റെ ദൈർଘ്യം കുറഞ്ഞത് 5 അക്ഷരങ്ങൾ  
ഉണ്ടാ എന്ന് പരിശോധിക്കുന്നു. ശേഷം വയസ്സ് നൽകിയിട്ടുണ്ടാ എന്ന്  
പരിശോധിക്കുന്നു പിന്നീട് നൽകിയ ഡാറ്റ സംവ്യൂത ആണെന്ന് അല്ലെന്നു എന്ന്  
പരിശോധിക്കുന്നു. അന്തിമമായി പ്രായപരിധി 15 മുതൽ 20 വരെ ആണെന്ന് എന്ന്  
പരിശോധിക്കുന്നു.



#### 6.9 സ്ക്രിപ്റ്റുകൾ വൈബോജിതർ ടെച്നോളജിക്കളിനുള്ള വഴികൾ

പല രീതിയിൽ HTML കോഡുകൾക്കുള്ളിൽ സ്ക്രിപ്റ്റുകൾ ഉൾപ്പെടുത്താവുന്നതാണ്.  
മുൻ ഉദാഹരണങ്ങളിൽ നമ്മൾ ജാവാസ്ക്രിപ്റ്റ് വൈബ് പേജിന്റെ ഫോറമാണ്  
ഉൾപ്പെടുത്തിയിരുന്നത്. ഇതിന് പുറമേ, സ്ക്രിപ്റ്റുകൾ <BODY> ടാഗിലോ അല്ലെങ്കിൽ  
ഒരു ബാഹ്യ ഫയലിലോ നൽകാവുന്നതാണ്.



ഇന്നി വെബ്പോജുകളിൽ സ്കീപ്പറുകൾ കൂടിച്ചേർക്കുന്നതിനുള്ള വ്യത്യസ്ത രീതികൾ പരിചയപ്പെട്ടാം.

### 6.9.1 <BODY> ടാഗിന് ഉള്ളടിൽ

ഇരു അധ്യായത്തിന്റെ തുടക്കത്തിൽ <BODY> ടാഗിന് ഉള്ളിൽ സ്ക്രിപ്റ്റുകൾ സ്ഥാപിക്കുന്നത് നാം ചർച്ചചെയ്തു. വെബ്പേജിന്റെ ഉള്ളടക്കം ബോസറിൽ പ്രവർഷിപ്പിക്കുന്നോൾ സ്ക്രിപ്റ്റുകളും പ്രവർത്തിക്കും. വെബ്പേജ് ഡോക്യുമെന്റിന്റെ ആരംഭം മുതൽക്കാണ് പ്രവർഷിപ്പിക്കുക. ഒരു സ്ക്രിപ്റ്റ് കോഡ് കാണുന്നോൾ അത് പ്രവർത്തിക്കുകയും തുടർന്ന് വെബ്പേജിന്റെ ബാക്കി ഭാഗം പ്രവർത്തിക്കുകയും ചെയ്യുന്നു.

രു ഉദാഹരണം ഉപയോഗിച്ച് ഈ രീതി മനസ്സിലാക്കാം. ഒരു വിദ്യാർമ്മിയുടെ റിസൾട്ട് ലഭിക്കുന്നതിനുള്ള ഒരു വൈബ് പേജാണ് താഴെ കൊടുത്തിരിക്കുന്നത്. ടെക്ന്റ് ബോക്സിൽ ഉപയോക്താവ് ഒരു രജിസ്റ്റർ നമ്പർ നൽകണം. **Get Result** ബട്ടനിൽ ക്ലിക്കു ചെയ്യുമ്പോൾ രജിസ്റ്റർ നമ്പർ ബോക്സിൽ ഫോറെക്ഷില്ഡ് ഡാറ്റ ഉണ്ടാക്കുന്നതിൽ പരിശോധിക്കാനുള്ള ജാവാസ്ക്രിപ്റ്റ് ഫംഗ്ശൻ പ്രവർത്തിക്കണം. ടെക്ന്റ് ബോക്സിൽ വില ഉണ്ടെങ്കിൽ അത് ഫ്രീ അക്കമുള്ളു ഒരു സംഖ്യ തന്നെ ആയിരിക്കണം. ഈ കാര്യങ്ങൾ ഫംഗ്ശൻ പരിശോധിക്കണം. വൈബ് പേജിന്റെ ഒരുപുട്ട് ചിത്രം 6.26 ത്തെ നൽകിയിരിക്കുന്നു.

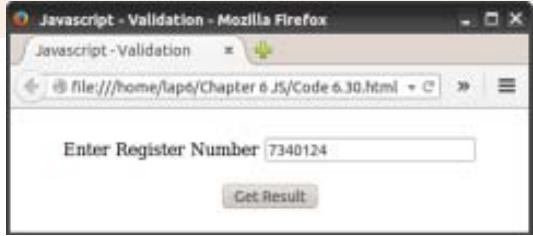
**ഉദാഹരണം 6.17:** സാധുതാപതിശോധനയ്ക്ക് മശേഷം തജിസ്റ്റർ നമ്പർ സ്വീകാര്യക്രമം ദാഖലം ചെയ്യാൻ വേണ്ടും.

```
<HTML>
<HEAD> <TITLE>Javascript - Validation</TITLE> </HEAD>
<BODY>
<FORM Name= "frmValid">
<SCRIPT Language= "JavaScript">
    function checkData()
    {
        var rno;
        rno = document.frmValid.txtRegno.value;
        if (rno == " ")
        {
            alert("Please enter Register No.");
            return;
        }
        if (isNaN(rno))
        {
            alert("Invalid Register No.");
            return;
        }
        if (rno.length < 7)
        {
            alert("Register No. must be 7 digits long");
            return;
        }
    }
</SCRIPT>
</FORM>
</BODY>
</HTML>
```



```
        alert("The Register No. must have 7 digits");
        return;
    }
}
</SCRIPT>
<CENTER>
<BR>Enter Register Number
<INPUT Type= "text" Name= "txtRegno">
<BR><BR>
<INPUT Type= "button" Value= "Get Result"
onClick= "checkData()">
</CENTER>
</FORM>
</BODY>
</HTML>
```

സ്ക്രീപ്റ്റ <BODY> ടാഗിനുള്ളിൽ അവസാനഭാഗത്തും ആകാം.  
<BODY>ടാഗിൽ അല്ലെങ്കിൽ <HEAD> ടാഗിൽ സ്ക്രീപ്റ്റുകൾ എഴുതിയിട്ടുണ്ടെങ്കിൽ അവ HTML കോഡിനൊപ്പം ബൈറ്റസറിലെ തത്ത്വകയും ചെയ്യും. വെബ് പേജ് പ്രദർശിപ്പിക്കുന്നത് വെകുവാൻ ഇത് കാരണമാകുന്നു. സ്ക്രീപ്റ്റുകൾ </BODY> ടാഗിന് മുമ്പായി കൊടുക്കുകയാണെങ്കിൽ വെബ്‌പേജിലെ ഉള്ളടക്കങ്ങൾ അക്ഷരങ്ങൾ, ചിത്രങ്ങൾ പോലുള്ളവ സ്ക്രീനിൽ വേഗത്തിൽ ദൃശ്യമാകും. എന്നാൽ വെബ് പേജ് ലഭ്യമാക്കുന്നതിനിടയിൽ സ്ക്രീപ്റ്റുകൾ പ്രവർത്തിക്കണമെന്നില്ല.



ചിത്രം 6.26: രജിസ്ട്രേഷൻ സൈറ്റ് സ്വീകരിക്കാനുള്ള ഫോം പ്രദർശിപ്പിക്കുന്ന വെബ്‌പേജ്

### 6.9.2 <HEAD> ടാഗിന് ഉള്ളിൽ

സാധാരണ രീതിയിൽ വെബ് പേജിന്റെ ഫോം ഭാഗത്താണ് സ്ക്രീപ്റ്റുകൾ ഉൾപ്പെടുത്തുന്നത്. നാം ചർച്ച ചെയ്ത എല്ലാ ഉദാഹരണങ്ങളിലും അങ്ങനെ തന്നെയാണ് ചെയ്തിരിക്കുന്നത്. ഇതിന് പ്രധാന കാരണം മിക്ക HTML പേജുകളുടെയും ഭോഡി ഭാഗത്തിനുള്ളിൽ ഉള്ളടക്കത്തെ സുചിപ്പിക്കുന്ന ഒരു വലിയ അളവ് ടെക്നോളജിക്കൾ വെബ്‌പേജിൽ മാറ്റം വരുത്തുന്നോ ഇത് രൂപകല്പനചെയ്യുന്നയാൾക്ക് ആശയക്കുഴപ്പം സൃഷ്ടിക്കും. ഭോഡി ഭാഗത്തിന് മുൻപായി ഒരു വെബ് പേജിന്റെ ഫോം ഭാഗം ലോഡ് ചെയ്യുന്നു എന്നത് ഇതിന്റെ നേട്ടമാണ്. അതിനാൽ, ഭോഡി ഭാഗത്ത് വിളിക്കപ്പെടുന്ന ഫംഗ്ശൻമുകളുടെ നിർവ്വചനം നേരേത്തെ തന്നെ മെമ്പറിയിൽ ലഭ്യമാക്കിയിരിക്കും. ഉദാഹരണം 6.17 ലേ



### 6.9.3 ബാഹ്യ ജാവാസ്ക്രിപ്റ്റ് ഫയൽ

HTML ഡോക്യുമെന്റിലെ എല്ലാ സ്ക്രിപ്റ്റുകളും ഒരു ബാഹ്യ ഫയലിലേക്ക് ശേർത്തിക്കുകയും ഡോക്യുമെന്റിൽ നിന്ന് ആ ഫയലിലേക്ക് ഒരു ലിങ്ക് സ്ഥാപിക്കുകയും ചെയ്യാം. ഈ ഫയൽ ‘.js’ എന്ന എക്സിഞ്ചനോടുകൂടി സൂക്ഷിക്കുക. ബാഹ്യ ഫയലുകളിൽ ജാവാസ്ക്രിപ്റ്റ് സ്ഥാപിക്കുന്നത് കൊണ്ട് പില ഗുണങ്ങളുണ്ട്. ഒരേ സ്ക്രിപ്റ്റ് ഒന്നിലധികം HTML പേജുകളിലേക്ക് ഒരു മുഴുവൻ വൈബ് സെറ്റിലേക്ക് ആവർത്തിക്കപ്പെടുവോൾ ഇത് ഉപയോഗപ്രദമാണ്. ഈ HTML ഉം ജാവാസ്ക്രിപ്റ്റും വേർത്തിരിച്ച് കാണാൻ സഹായിക്കുന്നു. ഈ HTML കോഡിനെയും ജാവാസ്ക്രിപ്റ്റിനെയും വേർത്തിരിക്കുവാനും ഇവ രണ്ടിനെയും എളുപ്പത്തിൽ വായിക്കാനും നിലനിർത്താനും സഹായിക്കുകയും ചെയ്യുന്നു. ജാവാസ്ക്രിപ്റ്റ് ബാഹ്യ ഫയലുകളിൽ സംഭരിക്കുന്നത് പേജ് വേഗത്തിൽ ലഭ്യമാക്കാൻ സഹായിക്കും.

മുകളിൽ പറഞ്ഞിരിക്കുന്ന ഉദാഹരണം 6.17 ലേഖ ജാവാസ്ക്രിപ്റ്റ് കോഡ് “check.js” എന്ന പേരിൽ ഒരു പ്രത്യേക ഫയൽ ആയി സൂക്ഷിക്കുന്നു. ഈ ഫയലിന്റെ ഉള്ളടക്കം താഴെ കാണിച്ച് രീതിയിലാണ്.

```
function checkData()
{
    var rno;
    rno = document frmValid.txtRegno.value;
    if (rno == "")
    {
        alert("Please enter Register No.");
        return;
    }
    if (isNaN(rno))
    {
        alert("Invalid Register No.");
        return;
    }
    if (rno.length < 7)
    {
        alert("The Register No. must have 7 digits");
        return;
    }
}
```

ഈ ഫയലിൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് മാത്രമേ ഉള്ളു എന്നും <SCRIPT> ടാഗ് ഉപയോഗിച്ചിട്ടില്ല എന്നും ശ്രദ്ധിക്കുക. HTML ഫയലിനുള്ളിൽ മാത്രമാണ് <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നത്. <SCRIPT> ടാഗ് ഉപയോഗിച്ച് ജാവാസ്ക്രിപ്റ്റ് ഫയലിനെ HTML ഫയലുമായി ലിങ്ക് ചെയ്യാവുന്നതാണ്. Type ആട്ടിബ്യൂട്ട് സൂചിപ്പിക്കുന്നത്

ലിങ്കുചെയ്തിരിക്കുന്ന ഫയൽ ഒരു ജാവാസ്ക്രിപ്റ്റ് ഫയൽ ആണെന്നതാണ്. Src ആട്ടിബ്യൂട്ട് ബാഹ്യ ജാവാസ്ക്രിപ്റ്റ് ഫയലിന്റെ സ്ഥാനവും ഫയലിന്റെ പേരും വ്യക്തമാക്കുന്നു. പരിഷ്കരിച്ച HTML കോഡ് താഴെ കൊടുത്തിരിക്കുന്നു.

```
<HTML>
<HEAD><TITLE>Javascript - Validation</TITLE>
<SCRIPT Type= "text/JavaScript" Src= "checkdata.js">
</SCRIPT>
</HEAD>
<BODY>
<FORM Name= "frmValid">
<CENTER>
<BR>Enter Register Number
<INPUT Type= "text" Name= "txtRegno">
<BR><BR>
<INPUT Type= "button" Value= "Get Result"
onClick= "checkData()">
</CENTER>
</FORM>
</BODY>
</HTML>
```

<SCRIPT> ടാഗിൽ Src ആട്ടിബ്യൂട്ട് ഉണ്ടാക്കിയിൽ അതിലെ ഉള്ളടക്കം അവഗണിക്കപ്പെടും. അതായത്, നിങ്ങൾക്ക് ഒറ്റ <SCRIPT> ടാഗുപയോഗിച്ച് ബാഹ്യ ഫയൽ ചേർക്കാനും ഒരു കോഡ് നടപ്പാക്കാനും ഒരുമിച്ച് സാധ്യമല്ല. ഇതിനായി രണ്ട് വ്യത്യസ്ത <SCRIPT> ടാഗുകൾ ആവശ്യമാണ്. ഇവയിൽ ഒന്ന് Src ഉപയോഗിച്ച് ഒരു ബാഹ്യ ഫയൽ ചേർക്കാനും മറ്റൊന്ന് കോഡ് ചേർക്കാനും ഉപയോഗിക്കാം

### വിജ്ഞാന പ്രവേശനത്തി അറിയാം



1. ജാവാസ്ക്രിപ്റ്റിൽ onMouseEnter ഇവർഗ്ഗ് എപ്പോഴാണ് പ്രവർത്തിക്കുക?
2. ഭ്രാവപ്പഡാൻ ലിസ്റ്റിൽ <SELECT> നിന്നും തിരഞ്ഞെടുത്ത ഇന്നത്തിന്റെ സുചിക ലഭിക്കാനുള്ള പ്രോപ്പറ്റി അണ്.
3. ജാവാസ്ക്രിപ്റ്റിലെ onKeyDown, onKeyUp എന്നിവ താരതമ്യം ചെയ്യുക.
4. Number() ഫംഗിൾ ഉപയോഗിക്കുന്നത് എന്തിനാണ്?
5. ജാവാസ്ക്രിപ്റ്റ് കോഡ് </BODY> ടാഗിന് മുൻപായി ഉപയോഗിക്കുന്നതിന്റെ ശുണ്ട് എന്ത്?



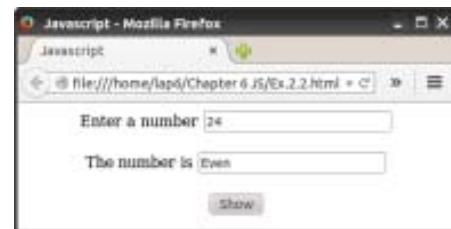
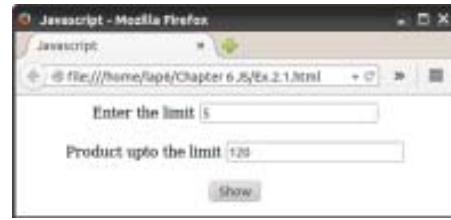
## നമ്മക്ക് സംഗ്രഹിക്കാം

ഡാറ്റയുടെ സാധ്യകരണത്തിനായി ഉപയോഗിക്കുന്ന ഒരു ഫേസ്റ്റ്ലൈൻ വശം സ്ക്രിപ്റ്റിംഗ് ഭാഷയായിട്ടാണ് ജാവാസ്ക്രിപ്റ്റിനെ ഈ അധ്യായത്തിൽ അവതരിപ്പിക്കുന്നത്. ജാവാസ്ക്രിപ്റ്റ് കോഡ് HTML ത്തെ ഉൾപ്പെടുത്തുന്നതിന് ഉപയോഗിക്കുന്ന ടാഗുകളും ജാവാസ്ക്രിപ്റ്റിലെ പ്രധാന ഫം‌ഷനുകളും ഈവിടെ വിശദികരിച്ചിരിക്കുന്നു. ജാവാസ്ക്രിപ്റ്റിലെ ഡാറ്റാരാഞ്ചും വേരിയബിളിങ്സ് ഉപയോഗവും വിശദമായി പർച്ചേചയ്ക്കുന്നു. ഓപ്പറേറ്ററുകളുടെയും കൺട്രോൾ സ്ട്രക്ചറുകളുടെയും ഉപയോഗം C++-ലേതിന് സമാനമാണ്. വിവിധ അന്തർ നിർമ്മിത ഫം‌ഷനുകളും ജാവാസ്ക്രിപ്റ്റിലെ ഇവഎകളും ഉദാഹരണസഹിതം വിശദികരിച്ചിരിക്കുന്നു. HTML പേജിൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് ഉൾപ്പെടുത്തുന്ന വ്യത്യസ്ത രീതികൾ വിശദമായി പ്രതിപാദിച്ചിരിക്കുന്നു.



## നമ്മക്ക് പരിശീലിക്കാം

1. താഴെ കാണുന്ന സ്ക്രീൻ പ്രദർശിപ്പിക്കുന്നതിന് ഒരു വെബ് പേജ് തയാരാക്കുക. ഉപയോകതാവിന് ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഒരു നമ്പർ നൽകാം. Show ബട്ടൺ ക്ലിക്കുചെയ്യുമ്പോൾ രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ 1 മുതൽ നൽകിയിട്ടുള്ള പരിധി വരെയുള്ള എല്ലാ സംഖ്യകളുടെയും ഗുണനഫലം പ്രദർശിപ്പിക്കണം.
2. താഴെ കാണുന്ന സ്ക്രീൻ പ്രദർശിപ്പിക്കുന്നതിന് ഒരു വെബ് പേജ് തയാരാക്കുക. ഉപയോകതാവിന് ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഒരു നമ്പർ നൽകാം. Show ബട്ടൺ ക്ലിക്കുചെയ്യുമ്പോൾ രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഒറ്റ സംഖ്യയോ ഇരട്ട സംഖ്യയോ എന്നതിന് അനുസൃതമായി Odd അല്ലെങ്കിൽ Even എന്ന പ്രദർശിപ്പിക്കണം.
3. താഴെക്കൊടുത്തിരിക്കുന്ന സ്ക്രീൻ പ്രദർശിപ്പിക്കുന്നതിന് ഒരു വെബ് പേജ് തയാരാക്കുക. ഉപയോകതാവിന് ടെക്സ്റ്റ് ബോക്സിൽ ഒരു പ്രായം നൽകാം. ടെക്സ്റ്റ് ബോക്സിൽ ഒരു സംഖ്യക്ക് പകരം ഉപയോകതാവ് അക്ഷരമാണ് നൽകുന്നതെങ്കിൽ, Show ബട്ടൺ ക്ലിക്കുചെയ്യുമ്പോൾ ഒരു സംഖ്യയോ ഇരട്ട സംഖ്യയോ എന്നതിന് അനുസൃതമായി Odd അല്ലെങ്കിൽ Even എന്ന പ്രദർശിപ്പിക്കണം.



കീക്കുചെയ്യുന്നോൾ "Invalid Age" എന്ന ഒരു സന്ദേശം പ്രദർശിപ്പിക്കണം അല്ലെങ്കിൽ "Correct Data" എന്ന സന്ദേശം പ്രദർശിപ്പിക്കണം.

4. ചിത്രത്തിൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഒരു ലോഗിൻ പേജ് തയാറാക്കുക. പാസ് വേഡ് നൽകുന്നതിനായി പാസ് വേഡ് ടെക്റ്റ് ബോക്സും യുസർനെയിം ഒരു ടെക്റ്റ് ബോക്സും ഉണ്ടായിരിക്കണം. യുസർനെയിൽ കുറത്തത് 4 മുതൽ പാസ് വേഡിൽ കുറത്തത് 6 മുതൽ കൂട്ടുകൾ ഉണ്ടായിരിക്കണം. പാസ് വേഡിലെ ആദ്യത്തെ റണ്ക് കൂട്ടുകൾ അക്കങ്ങൾ ആയിരിക്കണം. show ബട്ടണിൽ കീക്കുചെയ്യുന്നോൾ സാധ്യവായ ധാരാളം ബോക്സുകളിൽ നൽകിയിട്ടുള്ളതെങ്കിൽ "Correct Data" എന്നും അല്ലെങ്കിൽ "Wrong Data" എന്നുമുള്ള സന്ദേശം പ്രദർശിപ്പിക്കണം.
5. താഴെ കാണുന്ന ചിത്രത്തിലെ പോലെ ലഭിതമായ ഒരു കാൽക്കുലേറ്റർ അടങ്കിയ ഒരു വേഡ് പേജ് വികസിപ്പിക്കുക. റണ്ക് സംഖ്യകൾ നൽകാൻ റണ്ക് ടെക്റ്റ് ബോക്സുകൾ ഉണ്ടായിരിക്കണം. സങ്കലനം, വ്യവകലനം, ഗുണനം, ഹരണം എന്നിവയ് കാണി 4 ബട്ടണുകൾ ഉണ്ടായിരിക്കണം. ബട്ടനിൽ കീക്കുചെയ്യുന്നോൾ മുന്നാമത്തെ ടെക്റ്റ് ബോക്സിൽ ഉത്തരം കാണിക്കണം.

### നമ്മക്ക് വിലയിരുത്താം |||

1. താഴെപ്പറയുന്നവയിൽ വേറിയബിൾ z എന്ന വേറിയബിളിൽ വില എഴുതുക
  - a. 

```
var x, y, z;
x = 5;
y = 3;
z = ++x - y--;
```
  - b. 

```
var x, y, z;
x = "12";
y = 13;
z = x + y;
```



- c. 

```
var x, y, z;
x = 20;
y = 8;
x %= y;
z = x++;
```
- d. 

```
var x, y, z;
x = 1;
y = 4;
z = !(x < y);
```
- e. 

```
var x, y, z;
x = 5;
y = 6 ;
z = (x > y) || (y % 2 == 0);
```
2. താഴെകൊടുത്തിരിക്കുന്നവയുടെ ഒരുപ്പുക് കണ്ടത്തുക.
- a. 

```
<HTML>
<BODY>
<SCRIPT Language= "JavaScript">
var i;
for (i = 10; i >= 1; i--)
    document.write(i + "<BR>");
</SCRIPT>
</BODY>
</HTML>
```
- b. 

```
<HTML>
<BODY>
<SCRIPT Language= "JavaScript">
var i, s = 0;
for (i = 1; i <= 100; i += 2)
    s += i;
document.write("Sum = " + s);
</SCRIPT>
</BODY>
</HTML>
```
- c. 

```
<HTML>
<BODY>
<SCRIPT Language= "JavaScript">
var n, s = 0;
n = 0;
```



```
while (n <= 50)
{
    s = s + n;
    n = n + 5;
}
document.write("Sum = " + s);
</SCRIPT>
</BODY>
</HTML>

d. <HTML>
<BODY>
<SCRIPT Language= "JavaScript">
var n, f = 1;
n = 5;
while ( n > 0)
{
    f = f * n;
    n--;
}
document.write("Product = " + f);
</SCRIPT>
</BODY>
</HTML>
```

3. താഴെ കൊടുത്തിരിക്കുന്ന html കോഡ് ശ്രദ്ധിക്കുക

```
<FORM Name= "frmStud">
<INPUT Type= "text" Name= "studentName">
</FORM>
```

ഒക്ടൂൺ ബോക്സിൽരെ വില ന എന്ന വേരിയബിളിൽ സംഭരിക്കുന്നതിനായി വിട്ടാണ് പൂർപ്പിക്കുക.

```
var n;
n = .....;
```

4. നിങ്ങൾ checkData () എന്ന പേരിൽ ഒരു ജാവാസ്ക്രിപ്റ്റ് ഫലം ഷൻ എഴുതിയിട്ടുണ്ടെന്ന് കരുതുക. മുൻ പോയിറ്റർ ബട്ടനുമുകളിലൂടെ നീക്കുമ്പോൾ മാത്രമേ ഫലം ഷൻ പ്രവർത്തിക്കേണ്ടതുള്ളൂ. അതിനുവേണ്ടി താഴെക്കൊടുത്തിരിക്കുന്ന കോഡ്‌എഞ്ചനെ പൂർത്തീകരിക്കാം.

```
<INPUT Type= "button" ..... = "checkData ()">
```



5. <SCRIPT> ടാഗും അതിന്റെ ആട്ടിബ്യൂട്ടുകളും വിശദമാക്കുക.
6. ജാവാസ്ക്രീപ്പറ്റിൽ ഒരു അന്തർനിർമ്മിത ഫല്ഷൻ നിർവചിക്കുന്നതിനുള്ള വാക്യാലടന എഴുതുക.
7. താഴെക്കൊടുത്തിരിക്കുന്ന വിലകൾ ജാവാസ്ക്രീപ്പറ്റിലെ അനുയോജ്യമായ ഡാറ്റാ തരങ്ങൾ ഉപയോഗിച്ച് വേർത്തിരിക്കുക.  
“Welcome”, “123”, “true”, 67.4, .98, false, “hello”
8. ജാവാസ്ക്രീപ്പറ്റിൽ undefined ഡാറ്റ തരം എന്നതുകൊണ്ട് എന്താണ് അർഥമാക്കുന്നത്?
9. ജാവാസ്ക്രീപ്പറ്റിലെ ഓപ്പറേറ്റുകൾ വിശദീകരിക്കുക.
10. താഴെപ്പറയുന്നവ ചെയ്യുന്നതിനുള്ള ജാവാസ്ക്രീപ്പറ്റ് ഫല്ഷൻകളുടെ പേര് എഴുതുക
  - a. N എന്ന വേരിയബിളിൽ സംഖ്യ ആണോ എന്ന് പരിശോധിക്കാൻ.
  - b. "scert" എന്ന സ്ക്രിപ്റ്റ് വലിയ അക്ഷരങ്ങളായി മാറ്റാൻ.
  - c. "HTML" എന്ന സ്ക്രിപ്റ്റ് ചെറിയ അക്ഷരങ്ങളായി മാറ്റാൻ.
  - d. "Welcome to functions" എന്ന സന്ദേശം പ്രദർശിപ്പിക്കാൻ.
  - e. "Computer" എന്ന സ്ക്രിപ്റ്റിലെ മുന്നാമത്തെ ക്യാരക്ടർ പ്രദർശിപ്പിക്കാൻ.
11. 'Computer' എന്ന സ്ക്രിപ്റ്റിന്റെ നീളം പ്രദർശിപ്പിക്കുന്നതിനുള്ള ജാവാസ്ക്രീപ്പറ്റ് കോഡ് എഴുതുക.
12. ഒരു വെബ് പേജിൽ ഒരു ബട്ടൺ ഉണ്ട്. താഴെപ്പറയുന്ന ഇവർഗ്ഗുകൾക്കുനുസരിച്ച് Message() എന്ന ഫല്ഷൻ പ്രവർത്തിപ്പിക്കാനുള്ള HTML കോഡ് എഴുതുക.
  - a. ഉപയോകതാവ് ബട്ടണിൽ മുന്ന് ക്ലിക്ക് ചെയ്യുന്നോൾ.
  - b. ഉപയോകതാവ് ബട്ടണിനുമേൽ മുന്ന് പോയിറ്റു നീക്കുന്നോൾ.
13. ഒരു HTML പേജിൽ ഹൈ ഭാഗത്ത് ജാവാസ്ക്രീപ്പറ്റ് കോഡ് എഴുതുന്നതിന്റെ ഗുണങ്ങൾ എവ?
14. ഒരു വിഷയത്തിന്റെ മാർക്ക് നൽകുവാൻ ഒരു ടെക്റ്റസ്റ്റ് ബോക്സ് അടങ്ങിയിരിക്കുന്ന ഒരു വെബ് പേജ് രൂപകൽപ്പന ചെയ്യുക.
  - a. ഈ വെബ് പേജിനുവേണ്ടിയുള്ള HTML കോഡ് എഴുതുക.
  - b. ഈ ടെക്റ്റസ്റ്റ് ബോക്സിലെ വിലയുടെ സാധൂകരണത്തിന് ഒരു ബാഹ്യ ജാവാസ്ക്രീപ്പറ്റ് ഫയൽ നിർമ്മിച്ച് അത് HTML ഡോക്യുമെന്റുമായി ബന്ധിപ്പിക്കുക. വിലയിരുത്തുന്നതിനുള്ള സൂചകങ്ങൾ
    - (i) അത് ശുന്നമായിരിക്കരുത്
    - (ii) അത് ഒരു സംഖ്യയായിരിക്കണം
    - (iii) 0-നും 60-നും ഇടയിലായിരിക്കണം മാർക്ക്.
  - c. സ്ക്രീപ്പറ്റ് ഒരു ബാഹ്യ ഫയൽ അയി എഴുതുന്നതിന്റെ ഗുണപരമായങ്ങൾ എഴുതുക.



## 7

# വെബ് ഹോസ്റ്റിംഗ്

## പ്രധാന പട്ട നേട്ടങ്ങൾ



ഈ അദ്ദീയം പുർത്തിയാകുമ്പോൾ പറിതാവ്:

- വെബ്സൈറ്റ് ഉപയോഗം വെബ് ഹോസ്റ്റിംഗ് ആശയം എന്നിവ വിവരിക്കും.
- വ്യത്യസ്ത തരത്തിലുള്ള ഹോസ്റ്റിംഗ് രീതി മനസിലാക്കും.
- ഹോസ്റ്റിംഗ് സ്ഥലം വാങ്ങുന്നതിനുള്ള മാർഗ്ഗം വിശദമാക്കും.
- ദഡാമെൻ നെയിം രജിസ്ട്രർ ചെയ്യുകയും എച്ച്.റി.പി. ഷൈറ്റ് സോഫ്റ്റ് വെയർ ഉപയോഗിച്ച് ഹോസ്റ്റ് ചെയ്യും.
- സൗജന്യ ഹോസ്റ്റിംഗ് എൻ സവിശേഷ തകൾ വിശദമാക്കും.
- കണ്ടറ്റ് മാനേജ്മെന്റ് സംവിധാനത്തിനുള്ള ഉപയോഗം തിരിച്ചിരിയും.
- റിസ്പോൺസിവ് വെബ് രൂപ കൽപ്പനയുടെ ആവശ്യകത വിവരിക്കും.

വെബ് പേജുകൾ എങ്ങനെ നിർമ്മിക്കാമെന്ന് മുൻ അഭ്യാസങ്ങളിൽ നിന്ന് നാം പറിച്ചുവാലോ? നിരവധി വെബ്പോജുകൾ ചേർന്ന് നിർമ്മിക്കപ്പെടുന്ന വെബ്സൈറ്റ് ഒരു സ്ഥാപനത്തിന്റെയോ ഉല്പന്നത്തിന്റെയോ സേവനത്തിന്റെയോ വിവരങ്ങൾ നൽകുന്നവയായിരിക്കും. നമ്മുടെ സ്കൂളിന്റെ ഒരു വെബ്സൈറ്റ് HTML ഉപയോഗിച്ച് നിർമ്മിച്ച എന്നിരിക്കും. ഈ വെബ്സൈറ്റ് ഇൻറെന്ററിൽ എങ്ങനെ ലഭ്യമാക്കാം? അങ്ങനെ ലഭ്യമാക്കണമെക്കിൽ ഈ വെബ്പോജുകളെ ഇൻറെന്ററുമായി ബന്ധമുള്ള ഒരു വെബ്സൈറ്റ് വാൻഡേറ്റിനാണ്. വെബ്ഹോസ്റ്റിംഗിനെപ്പറ്റിയുള്ള അവലോകനം, വിവിധതരം വെബ്ഹോസ്റ്റിംഗുകൾ അതിന്റെ പ്രത്യേകതകൾ തുടങ്ങിയവ ഈ അധ്യായത്തിൽ പ്രതിപാദിക്കുന്നു. കൂടാതെ ഒരു വെബ്സൈറ്റിനെ തിരിച്ചിരിയാനുള്ള ദഡാമെൻപേര് (ദഡാമെൻ നെയിം) എങ്ങനെ തെരഞ്ഞെടുക്കാം എന്നും അത് എങ്ങനെ രജിസ്ട്രർ ചെയ്യാം എന്നും ഇവിടെ ചർച്ചചെയ്യുന്നു. വിവിധ FTP CLIENT സോഫ്റ്റ് വെയർ ഉപയോഗിച്ച് നമ്മുടെ കമ്പ്യൂട്ടറിൽ നിന്ന് സൈറ്റ് ലൈഡ്യൽകൾ ഫയലുകൾ (വെബ്പോജുകൾ, ചിത്രങ്ങൾ മുതലായവ) എങ്ങനെ മാറ്റാമെന്നും ഇതിൽ വിശദമാക്കുന്നു. ഈ അദ്ദീയം പറിച്ചുകഴിയുമ്പോൾ ഒരു ദഡാമെൻ പേര് രജിസ്ട്രർ ചെയ്യുവാനും വെബ്സൈറ്റ് ഹോസ്റ്റ് ചെയ്യുവാനും പഠിതാവിന്നകഴിയും.

## 7.1 വെബ് ഹോസ്റ്റിംഗ് (Web hosting)

നമ്മുടെ സ്കൂളിനായി നിരവധി പേജുകളുള്ള ഒരു വെബ്സൈറ്റ് എങ്ങനെ നിർമ്മിക്കുമെന്ന് മുൻ അദ്ദൂയായങ്ങളിൽ നാം പതിച്ചിട്ടുണ്ട്. ഏതെങ്കിലും ഒരു ടെക്നോളജി അല്ലെങ്കിൽ ഒരു വെബ്സൈറ്റെന്ന് ടുൾ ഉപയോഗിച്ച് നമ്മുടെ വെബ്സൈറ്റ് നിർമ്മിക്കാം. ഹോം പേജിൽ സ്കൂളിലെ സാക്ഷ്യങ്ങൾ, കോഴ്സുകൾ മേൽവിലാസം മറ്റ് വെബ് പേജുകളിലേയ്ക്കുള്ള ലിങ്കുകൾ എന്നിവ ഉൾപ്പെടുത്തി ആകർഷണീയമാക്കാം.

നമ്മുടെ കബ്യൂട്ടറിൽ സ്കൂളിൽ വെബ്സൈറ്റ് നിർമ്മാണം പൂർത്തിയാക്കിയ ശേഷം അത് ഇള്ളർന്നറ്റിൽ ലഭ്യമാക്കണം. അതിനായി നമ്മുടെ കൈവശമുള്ള വെബ്സൈറ്റ് എത്തെങ്കിലും വെബ്സൈറ്റും പുറത്തിയാക്കിയ ശേഷം അപ്പലോഡ് ചെയ്തെങ്കിൽ മാത്രമേ ലോകത്തുള്ള എല്ലാവർക്കും അത് വീക്ഷിക്കാനാവു. വെബ്സൈറ്റ് സുക്ഷിക്കുവാനായി വെബ്സൈറ്റും കുറച്ച് സ്ഥലം വാടകയ്ക്ക് എടുക്കുകയോ അല്ലെങ്കിൽ നമ്മുടെതായ ഒരുവെബ്സൈറ്റും ക്രമീകരിക്കുകയോ ചെയ്യാം. ഒരു വെബ്സൈറ്റവർ ക്രമീകരിക്കുക എന്നത് വളരെ ചെലവേറിയ ഒരു പ്രക്രിയയാണ്. ആയതിനാൽ നിലവിലുള്ള ഒരു വെബ്സൈറ്റവരിൽ നാലു സ്ഥലം വാടകയ്ക്കെടുക്കുന്നതായിരിക്കും ഉചിതം.

വെബ്ഹോസ്റ്റിംഗ് എന്നാൽ ഇള്ളർന്നറ്റിൽ ലഭ്യമാക്കേണ്ട വെബ്സൈറ്റിലെ ഫയലുകളെ വെബ്സൈറ്റവരിൽ സുക്ഷിക്കുകയും അതിനാവശ്യമായ സേവനം നൽകുകയും ചെയ്യുന്ന പ്രക്രിയയാണ്. വെബ്ഹോസ്റ്റിംഗ് സേവനങ്ങൾ നൽകുന്ന കമ്പനികളെ വെബ്ഹോസ്റ്റ് എന്നുവിളിക്കും. വെബ് സേർവ്വറുകൾ വെബ്ഹോസ്റ്റുടെ ഉടമസ്ഥതയിലും നിയന്ത്രണത്തിലും ഉള്ളതായിരിക്കും. ഇത്തരം വെബ്സൈറ്റവരുകൾ തടസ്സില്ലാതെ ഇള്ളർന്നറ്റിലെ വന്നം നൽകുവാനും PHP, JAVA, ASP.NET തുടങ്ങിയ പ്രോഗ്രാമിംഗ് ഭാഷകൾക്ക് ശേറ്റ് സേവന് പിൻബലം ആവശ്യമുള്ളപ്പോൾ നൽകാനും ഇത്തരം സേർവ്വറുകൾക്ക് കഴിയും.

### 7.1.1 വിവിധ തരം വെബ് ഹോസ്റ്റിംഗുകൾ (Types of web hosting)

HTML, ചിത്രങ്ങൾ എന്നിവ അടങ്കിയ നമ്മുടെ സ്കൂൾ വെബ്സൈറ്റിന് 4 എം.ബി. സ്ഥലമാണ് വെബ്സൈറ്റവരിൽ വേണ്ടത് എന്ന് കരുതുക. എന്നാൽ വെബ്ഹോസ്റ്റിംഗ് 10 എം.ബി., 20 എം.ബി തുടങ്ങിയ പാക്കേജുകളാണ് ഈന്ന് സാധാരണയായി ലഭ്യമാക്കുന്നത്. അപ്പോൾ നമ്മുടെ ആവശ്യത്തിനുസരിച്ച് എറ്റവും അനുയോജ്യമായ ഒരു പാക്കേജ് തെരഞ്ഞെടുക്കേണ്ടതുണ്ട്. നമ്മുടെ വെബ്സൈറ്റിൽ എത്ര ആളുകൾ സന്ദർശിക്കുമെന്നത് തെരഞ്ഞെടുപ്പിന് ഒരു മാനദണ്ഡമാണ്. നമ്മുടെ വെബ്സൈറ്റിൽ ധാറ്റാബേസും സ്ക്രിപ്റ്റുകളും ഉണ്ട് എങ്കിൽ അതിന് ആവശ്യമായ സേവനം ലഭ്യമാകുന്ന വെബ്ഹോസ്റ്റുകളെയാണ് തിരഞ്ഞെടുക്കേണ്ടത്. വെബ്ഹോസ്റ്റുകൾ വിവിധതരത്തിലുള്ള ഹോസ്റ്റിംഗ് പാക്കേജുകൾ ലഭ്യമാക്കുന്നു. ഷൈയേർഡ് ഹോസ്റ്റിംഗ്, വെർച്ചൽ പ്രൈവറ്റ് ഹോസ്റ്റിംഗ്, ഡെബിക്കറ്റീവ് ഹോസ്റ്റിംഗ് എന്നിങ്ങനെയാണ് അവയെ തരംതിരിച്ചിരിക്കുന്നത്.

**a. ഷൈയേർഡ് ഹോസ്റ്റിംഗ് (Shared hosting):** വെബ് ഹോസ്റ്റിംഗിലെ സർവ്വസാധാരണമായ ഹോസ്റ്റിംഗ് രീതിയാണ് ഷൈയേർഡ് വെബ്ഹോസ്റ്റിംഗ്. പല വെബ്സൈറ്റുകൾ ഒരു വെബ്സൈറ്റവരിൽ സുക്ഷിക്കുകയും സേർവ്വറിൽ RAM, CPU എന്നിവ പങ്കിടുകയും ചെയ്യുന്നതുകാണോ ഇതിനെ ഷൈയേർഡ് ഹോസ്റ്റിംഗ് എന്നു വിളിക്കുന്നത്. ഇത് ഉയർന്ന ബാൻഡ് വിധത്തും ഉയർന്ന സംഭരണസ്ഥലവും ആവശ്യമുള്ള വെബ്സൈ



റൂക്കൾക്ക് അനുയോജ്യമല്ല. ഷൈയേർഡ് ഹോസ്റ്റിംഗ് സെർവ്വേഴ്സ് കുറവുള്ള വെബ്സൈറ്റുകൾക്കും കുടുതൽ മുഖ്യമാണ്. അതുകൊണ്ട് തന്നെ ഇതുവളരെ വ്യാപകമായി ഉപയോഗിക്കുന്നു. സെർവ്വർ നവീകരണവും സോഫ്റ്റ്‌വെയർ ഇൻസ്റ്റാൾ ചെയ്യുന്നോൾ ഉണ്ടാകുന്ന സുരക്ഷാപ്രൈറ്റ് ഹോസ്റ്റിംഗ് കമ്പിനു തന്നെ പരിഹരിക്കും. നിരവധി വെബ്സൈറ്റുകൾ ഇതിന്റെ ബാൻഡ് വില്ക്ക് പകിടുന്നതുകൊണ്ട് ഏതെങ്കിലും ഒരു വെബ്സൈറ്റുകൾ ഇതിന്റെ ബാൻഡ് വില്ക്ക് പകിടുന്നതുകൊണ്ട് ഏതെങ്കിലും ഒരു വെബ്സൈറ്റുകളുടെയും പ്രവർത്തനം മനഗതിയിലാകും എന്നതാണ് ഇതിന്റെ പ്രധാനപ്പെട്ട നൃന്തരം.

**b. ഡെഡിക്കേറ്റെഡ് ഹോസ്റ്റിംഗ് (Dedicated hosting):** ഉപഭോക്താവ് ഒരു വെബ്സൈറ്റും അനുബന്ധം സൗകര്യങ്ങളും പൂർണ്ണമായും വാടകയ്ക്ക് എടുക്കുന്നതിനെ ഡെഡിക്കേറ്റെഡ് ഹോസ്റ്റിംഗ് എന്ന് വിളിക്കുന്നു. ഈ വെബ്സൈറ്റുകൾ മറ്റ് വെബ്സൈറ്റുകളുമായി വിഭാഗങ്ങൾ പകിടുന്നില്ല. വലിയ സ്ഥാപനങ്ങളും സർക്കാർ വകുപ്പുകളും ധാരാളം സെർവ്വേഴ്സ് ഇള്ളതുകൊണ്ട് ഡെഡിക്കേറ്റെഡ് വെബ്സൈറ്റുകളും കമ്പിനു തുടർന്നെല്ലാം ഇവിടെ ഹാർഡ്‌വെയറും സോഫ്റ്റ്‌വെയറും തെരഞ്ഞെടുക്കുന്നതിനുള്ള സ്വാതന്ത്ര്യവും വെബ്സൈറ്റുകൾക്ക് പൂർണ്ണ നിയന്ത്രണവും ഉപയോക്താവിനാണ്. ഡെഡിക്കേറ്റെഡ് സെർവ്വറിന്റെ കാര്യക്ഷമത ഏപ്പോഴും ഉയർന്ന നിലവാരത്തിലുള്ളതും, എന്നാൽ വളരെ ചെലവേറിയതാണ്. ഡെഡിക്കേറ്റെഡ് സെർവ്വറുകൾ യേറ്റാ സെറ്റിംഗുകളിൽ ആയാൽ ഇന്ത്രോനെറ്റ് കമ്പക്ഷനും വൈദ്യുതിയും വിദർശ സേവനവും ഏപ്പോഴും ലഭ്യമാകും. ഈ മൂലം ഇന്ത്രോനെറ്റ് കമ്പക്ഷനും വൈദ്യുതി സംബന്ധമായുള്ള സൗകര്യങ്ങളും ക്രമീകരിക്കുന്നതിനുള്ള ചെലവ് ഉപഭോക്താവിന് സ്വന്തമായി ക്രമീകരിക്കുന്നതിനേക്കാൾ കുറിച്ചായിരിക്കും. ബാൻഡ് വില്ക്ക് പക്കുവയ്ക്കുപ്പോരുത്തതിനാൽ വളരെ വേഗത്തിൽ വെബ്സൈറ്റുകളെ ഉപയോഗിക്കുവാൻ ഉപയോക്താവിന് സാധ്യമാകും. ഉപയോക്താവ് സേവനങ്ങളാണ് ഭൗതിക സൗകര്യത്തിനുള്ളിൽ വെബ്സൈറ്റുകൾ പ്രവർത്തിക്കുവാൻ ക്രമീകരണം ഒരുക്കുകയാണെങ്കിൽ അതിനെ കോ ലോക്കേഷൻ എന്നുവിളിക്കും.

**c. വെർച്ചുൽ പ്രൈവറ്റ് സെർവ്വർ (Virtual Private Server):** വെർച്ചുൽ പ്രൈവറ്റ് സെർവ്വറും എന്നാൽ വെർച്ചുൽ ടെക്നോളജി ഉപയോഗിച്ച് പലതായി വിഭജിക്കപ്പെട്ട ഒരു വെബ്സൈറ്റുകളാണ്. ഓരോ VPS ഉം ഡെഡിക്കേറ്റെഡ് സെർവ്വർ പോലെ പ്രവർത്തിക്കും. അതിനായി പ്രത്യേക ഓപ്പറേറ്റിംഗ് സിസ്റ്റം വെബ്സൈറ്റുകൾ സോഫ്റ്റ്‌വെയറും പാക്കേജുകളായ ഇ-മെയിൽ, ഡോക്യുമെന്റേഷൻ തുടങ്ങിയവയും അതിൽ സ്ഥാപിച്ചിരിക്കും. ഷൈയേർഡ് ഹോസ്റ്റിംഗ് സെർവ്വറും പോലെ ഇവയെ കരുതേണ്ടതില്ല. കാരണം VPS-ൽ ഒരു നിശ്ചിത തശ്ശമാനം RAM ഓരോ വെർച്ചുൽ വെബ്സൈറ്റുകളുമായി മാറ്റിവച്ചിരിക്കുന്നു. ഓരോ VPS-ഉം പൂർണ്ണ സ്വാതന്ത്ര്യമുള്ള വെബ്സൈറ്റുകളായി പ്രവർത്തിക്കും. അതായത് ഒരു പ്രത്യേക ഭൗതിക സെർവ്വറിൽ പ്രവർത്തിക്കുന്നതുപോലെ VPS-ൾ ഉപഭോക്താവിന് സോഫ്റ്റ്‌വെയർ ഇൻസ്റ്റാൾ ചെയ്യാനും അവരെ ക്രമീകരിക്കുവാനും പൂർണ്ണ അവകാശം ഉണ്ടായിരിക്കും. മറ്റ് വെർച്ചുൽസൈറ്റുകളെ ബാധിക്കാതെ പുനരാരംഭിക്കാൻ ഓരോ VPS-നും അവകാശമുണ്ട്. VPS-ഹോസ്റ്റിംഗ് സെർവ്വറിലെ ഓരോ വെബ്സൈറ്റുകളും തന്നെ ബാൻഡ് വില്ക്ക് പക്കുവയ്ക്കുന്നതുമുണ്ട്. ഇതുമുലം സെർവ്വർ ഷൈയർ ചെയ്യപ്പെട്ടാലും ഡെഡിക്കേറ്റെഡ് ഹോസ്റ്റിംഗ് സെർവ്വറുകൾക്ക് വെർച്ചുൽ പ്രവർത്തിക്കുന്നതുമുണ്ട്.

കേരുവ് ഹോസ്റ്റിംഗിൽ നേടുന്ന കൈവരിക്കാനാകും. ഷയേർവ് ഹോസ്റ്റിംഗിനേക്കാളും സവിശേഷതകൾ ആവശ്യമുള്ള വെബ്സൈറ്റുകൾക്കാണ് ഈത്തരം ഹോസ്റ്റിംഗ് യോജിക്കുന്നത്. പക്ഷെ ഡെഡിക്കേറ്റേറ്റ് ഹോസ്റ്റിംഗിൽ എല്ലാ പ്രത്യേകതകളും ഈവിടെ ലഭ്യമല്ല. ഡെഡിക്കേറ്റേറ്റ് ഹോസ്റ്റിംഗിൽ ഒരുമിക്ക സർവീസുകളും മിതമായ നിരക്കിൽ നൽകുവാൻ VPS-ഹോസ്റ്റിംഗിന് കഴിയും.

അറിയപ്പെടുന്ന ചില സെർവർ വെർച്ച്ചിലെസൈഷൻ സോഫ്റ്റ് വെയറുകളാണ് VM വെയർ, വെർച്ച്ചൽവോക്സ്, ഫ്രീ VPS, യൂസർ മോഡ് ലിനേക്സ്, മെമ്പ്രോ സോഫ്റ്റ് ഹെപ്പർ വി തുടങ്ങിയവ.

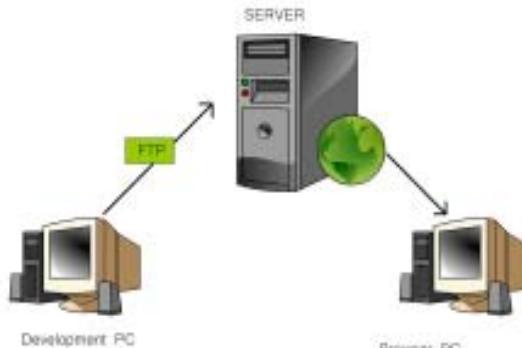
വിവിധതരത്തിലുള്ള ഹോസ്റ്റിംഗ് പാക്കേജുകളെപ്പറ്റിയുള്ള പ്രതീകാത്മക വിവരങ്ങൾ ചിത്രം 7.1 തോന്തരിക്കുന്നു.

Shared Hosting	Dedicated Hosting	VPS Hosting
Several websites	Single website	VPS 1 VPS 2 VPS 3
Resources	Resources	Resources
Physical Web Server	Physical Web Server	Physical Web Server

ചിത്രം 7.1 : വെബ്സൈറ്റേസനിൽ തരങ്ങൾ

## 7.1.2 ഹോസ്റ്റിംഗ് സ്ഥലം വാങ്ങുക (Buying hosting space)

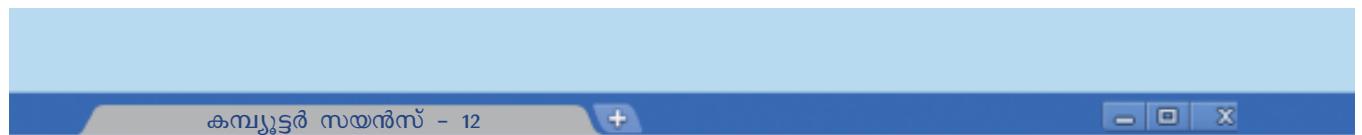
സ്കൂളിന് വേണ്ടി നിർമ്മിച്ച വെബ്സൈറ്റ് ഇപ്പോൾ നമ്മുടെ കമ്പ്യൂട്ടറിലെ ഒരു ഹോസ്റ്റിംഗിൽ സൂക്ഷിച്ചിരിക്കുകയാണ്. ഈ ഇൻഡൈന്റീൽ ലഭ്യമാകാൻ ഇല്ല എങ്കിലും വെബ്സൈറ്റുവിലേയ്ക്ക് പകർത്തണം. ചിത്രം 7.2 തോന്തരിക്കുന്നു.



ചിത്രം 7.2 : ഇൻഡൈന്റീൽ ഒരു വെബ്സൈറ്റ് ലഭ്യമാക്കുന്നു



ചിത്രം 7.3 : ഹോസ്റ്റിംഗ് ഓപ്പഷൻസ്



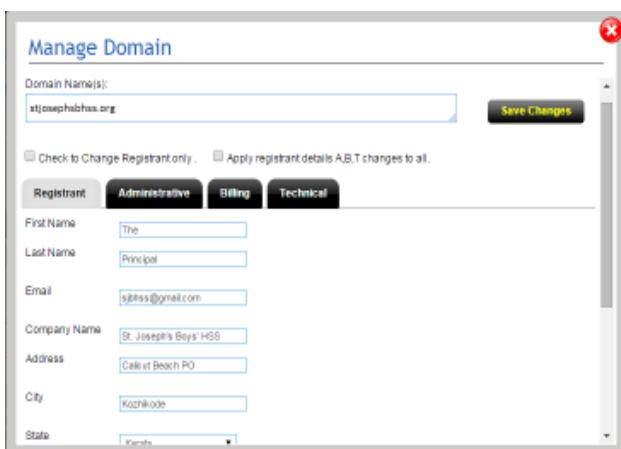
ശ്രദ്ധാർ സഹായം സേവന ഭാതാവിൽ നിന്ന് വാങ്ങണം. ഹോസ്റ്റിംഗ് സഹായം വാങ്ങുമ്പോൾ നിരവധി കാര്യങ്ങൾ പരിഗണിക്കേണ്ടതായുണ്ട്. ഒന്നാമതായി നമ്മുടെ വൈബ്ലേസറിന് എത്രസഹായം ആവശ്യമുണ്ടെന്ന് കണ്ടെത്തണം. വൈബ്ലേജിൽ പ്രോഗ്രാംിംഗ് കണ്ടെഴുപ്പ് ഉണ്ടായിരിക്കും. വൈബ്ലേജിൽ വൈബ്ലേസറിൽ ഈ സേവന നൽകുന്ന സാങ്കേതിക വിദ്യ ഉണ്ടായിരിക്കും. വൈബ്ലേജിൽ നിർമ്മാണത്തിനായി ഉപയോഗിച്ചിരിക്കുന്ന പ്രോഗ്രാം, വിഭേദാസ് ഹോസ്റ്റിംഗ്കുന്നുണ്ടോ ലിംക് ഹോസ്റ്റിംഗ്കുന്നുണ്ടോ അനുയോജ്യമെന്ന് കണ്ടെത്തണം. അതിനുശേഷം വിവരങ്ങൾ സേർവ്വറോ ലിംക് സേർവ്വറോ തെരഞ്ഞെടുക്കുണ്ടോ. ചിത്രം 7.3 ഈ വ്യക്തമാക്കുന്നു. വൈബ്ലേസറിൽ HTML കോഡ് മാത്രമാണ് ഉപയോഗിച്ചിരിക്കുന്നതെങ്കിൽ ഏത് തരം സേർവ്വരായാലും മതിയാകും. എന്നാൽ യേറ്റാ ബൈയ്സ്, ഈമെയിൽ എന്നിവയ്ക്ക് ആവശ്യമായ സാങ്കേതിക സൗകര്യം ആവശ്യമുണ്ടെങ്കിൽ അത് കൂടി പരിഗണിച്ച് വേണം വൈബ്ലേസറിനെ തിരഞ്ഞെടുക്കാൻ.

### 7.1.3 ഡോമേണ് നെയിം റജിസ്റ്റർ ചെയ്യുക (Domain name registration)

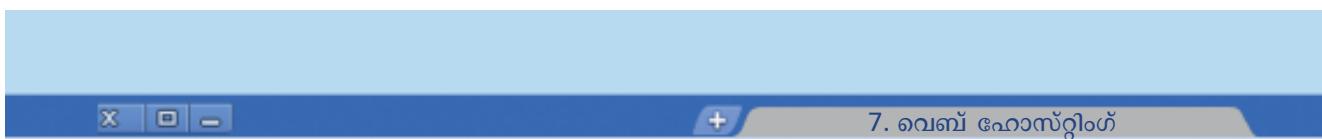
സ്കൂൾ വൈബ്ലേസറിന് ആവശ്യമായ ഹോസ്റ്റിംഗ് സഹായം നാം വാങ്ങിക്കണ്ടു. ഇൻഡ്രെനേറ്റിൽ ഓരോ വൈബ്ലേസറും തിരിച്ചറിയുന്നതിനായി പ്രത്യേകം URL ആവശ്യമാണ്. ഇതിനായി അനുയോജ്യമായ ഒരു ഡോമേണ് നാമം സ്കൂൾ വൈബ്ലേസറിനായി നാം തെരഞ്ഞെടുക്കുണ്ടോ. ഇൻഡ്രെനേറ്റിൽ ഒരു വൈബ്ലേസറിനെ തിരിച്ചറിയാൻ സഹായിക്കുന്നതാണ് ഡോമേണ് നാമം. മിക്ക വൈബ്ലേസറിൽ കമ്പനികളും ഡോമേണ് നെയിം റജിസ്റ്റർ ചെയ്യുവാനുള്ള സേവനം നൽകുന്നുണ്ട്. നമ്മുടെ വൈബ്ലേസറിന് അനുയോജ്യമായ ഒരു ഡോമേണ് നാമം തിരഞ്ഞെടുത്തു തുടർന്നും ഇവ ലഭ്യമാണോ എന്നും മറ്റാരെ കിലും ഇതേ പേരിൽ റജിസ്റ്റർ ചെയ്തിട്ടുണ്ടോ എന്നും പരിശോധിക്കുണ്ടോ. www.whois.net പോലെയുള്ള സൈറ്റുകളും വൈബ്ലേസറിലും കമ്പനികളുടെ വൈബ്ലേസറിനുകളും ഇത്തരം പരിശോധനയ്ക്ക് നമ്മുടെ സഹായിക്കുന്നു. റജിസ്റ്റർ ചെയ്തപ്പെട്ട എല്ലാ ഡോമേണ് നാമം അളുതെയും യേറ്റാ ബൈയ്സ് പരിശോധിച്ച് ഇവ മറുപടി നൽകുന്നു.



ചിത്രം 7.4 : Domain name registration search result



ചിത്രം 7.5 : WHOIS വിവരങ്ങൾ



ഡോമേന്മെന്റ് നാമം ലഭ്യമാണെങ്കിൽ പേര്, ICANN-ലോ whois.net ഷേറ്റാബെയ്സിൽ പേര്, അധിക സ്ലീസ്, ഫോൺ നമ്പർ, ഇ-മെയിൽ എന്നിവ ചിത്രം 7.5 തുടർച്ചാ ചീടു ഇതു പോലെ രേഖപ്പെടുത്തുന്നു.

ഈ വിവരങ്ങൾ രജിസ്ട്രർ ചെയ്യുന്ന ആളിൽ ഇഷ്ടകാനുസരണം പൊതുവായോ വ്യക്തിപരമായോ സുക്ഷിക്കുന്നതാണ്. വാർഷിക വരിസംഖ്യ ഓൺലൈൻഗായി അടച്ചുനമ്പുടെ പേരിൽ ഡോമേന്മെന്റ് രജിസ്ട്രർ ചെയ്യാവുന്നതാണ്. വാങ്ങുന്നതിൽ ഷേഖ്രിംഗ് കാർട്ട് ചിത്രം 7.6 തുടർച്ചാ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.

ഇപ്പോൾ നമ്മൾ വെബ്സൈറ്റുകളും സ്കൂൾ വെബ്സൈറ്റുകളും സ്ഥലം വാങ്ങി ഡോമേന്റ് നാമം രജിസ്ട്രർ ചെയ്തു. ഏതെങ്കിലും ഉപയോകതാവും നമ്പുടെ ഡോമേന്റ് (stjosephbhss.org) എന്ന് ബ്രൗസർ വിൻഡോയിൽ രേഖപ്പെടുത്തിയാൽ നമ്മൾ വെബ്സൈറ്റുകളിൽ സുക്ഷിച്ചിരിക്കുന്ന വെബ്പേജും ദൃശ്യമാകും. DNS നമ്പുടെ സെർവ്വീസ് IP അധികം ബ്രൗസറിൽ ലഭ്യമാക്കുകയും ആ പേരുകളും ഉപയോഗിച്ചു നമ്പുടെ വെബ്പേജും സുക്ഷിച്ചിരിക്കുന്ന വെബ്സൈറ്റുമായി ബന്ധമുണ്ടാക്കുകയും ചെയ്യുന്നതുമുള്ളാണ്. ഈ സാധ്യമാകുന്നത് വെബ്സൈറ്റുമായി ബന്ധപ്പെട്ടിട്ടുള്ള ഡോമേന്റ് നേരിലെ ഏഎ.പി. അധികാരിയും സുക്ഷിച്ചിരിക്കുന്നതിനും എ.റിക്കോർഡ് ഉപയോഗിക്കുന്നത്. ഡോമേനിന്റെ കൺട്രോൾ പാനലിൽ ലോറ്റ് ഇൻ ചെയ്തതിനുശേഷം എ.റിക്കോർഡ് പുതുക്കാവുന്നതാണ്. ഒരു ഡോമേനെന്നുകൂടിച്ചുള്ള വിവരങ്ങൾ whois search തോന്തരം ലഭ്യമാകും.

വെബ്സൈറ്റുകളിൽ IP അധികാരിയും സുചിപ്പിക്കുന്നതിനായി ചിത്രം 7.7 തുടർച്ചാ കാണിച്ചിരിക്കുന്നതുപോലെ A റിക്കോർഡ് ക്രമീകരിക്കാം. ഇതിന് ശേഷം ഡി.എൻ.എസ്. സെർവ്വർ നാം നൽകിയ വെബ്സൈറ്റ് ഇതിനും വെബ്സൈറ്റുകളിലേയ്ക്ക് ബ്രൗസർ വഴി ബന്ധപ്പെടാനുള്ള IP

അ ട റ സ  
ന ത ക കു .  
ന മ മ കു .  
വെബ്സൈറ്റ്  
ഇ റ ക കു .  
വെബ്സൈറ്റുകൾ  
ലേയ്ക്ക് വളരെ  
വേഗം ബന്ധപ്പെടാനാകും.

	NAME	TYPE	CONTENT	PRIORITY	Save Changes
<input checked="" type="checkbox"/>	stjosephbhss.org	A	118.67.244.3		<button>Save Changes</button>
<input checked="" type="checkbox"/>	www.stjosephbhss.org		118.67.244.3		<button>Save Changes</button>

ചിത്രം 7.7 : A റിക്കോർഡ് മാറ്റം



രജിസ്റ്റർ ഡെബാമെൻ ഉടമസ്ഥത എവിടെ എപ്പോൾ രജിസ്റ്റർ ചെയ്തു, കാലാവധി തുടങ്ങിയ വിവരങ്ങൾ ഇതിൽ ഉൾപ്പെടുന്നു. ഒരു പ്രത്യേക ഡെബാമെൻ പേര് ലഭ്യമാണോ എന്നും എന്ന് കണ്ണെത്തുന്നതിന് ഇത് ഉപയോഗിക്കും. [www.kerala.gov](http://www.kerala.gov). എന്നത് WHOIS റിസൈറ്റ് കമ്പനിയോ അഥവാ അത് താഴെ പറയുന്ന വിവരങ്ങൾ ലഭ്യമാക്കും.

```

Domain ID:D8944-AFIN
Domain Name:KERALA.GOV.IN
Created On:31-Dec-2003 05:00:00 UTC
Last Updated On:16-Jul-2014 11:37:59 UTC
Expiration Date:31-Dec-2016 05:00:00 UTC
Sponsoring Registrar:National Informatics Centre (R12-AFIN)
Status:OK
Registrant ID:R-R03120114034
Registrant Name:Government of Kerala
Registrant Organization:
Registrant Street1:Chief Minister's Office
Registrant Street2:
Registrant Street3:
Registrant City:Government Secretariate, Trivandrum, 69500
Registrant State/Province:Kerala
Registrant Postal Code:695001
Registrant Country:IN

```



### നമുക്ക് ചെയ്യാം

ജനപ്രീതിയാർജ്ജിച്ച വെബ്പോള്സുകളുടെയും അവയുടെ ഒരു വർഷത്തെ ഏറ്റവും കുറത്തെ അളവിലുള്ള ഹോസ്റ്റിംഗ് സൗലത്തിനായുള്ള വിലകളും താരതമ്യം ചെയ്യുന്ന ഫേബിൾ തയ്യാറാക്കുക. വിശ്വാസിക്കും, ലിനോക്സ് വെബ്പോള്സീറ്റിനും ഉള്ള വിലകൾ പ്രത്യേകമായി തയ്യാറാക്കുക.

ജനകീയമായ വെബ്പോള്സ് സേവനങ്ങളാക്കുന്ന .org, .com ഡെബാമെൻ റജിസ്ട്രേഷൻകൾക്ക് ഒരു വർഷത്തെ വിലവിവരപ്പടിക താരതമ്യം ചെയ്യുന്ന ഫേബിൾ തയ്യാറാക്കുക.

[www.dhsekerala.gov.in](http://www.dhsekerala.gov.in) എന്ന വെബ്പേസൈറ്റിൽ WHOIS വിവരങ്ങൾ [www.scert.kerala.gov.in](http://www.scert.kerala.gov.in), [www.whois.net](http://www.whois.net) തുടങ്ങിയ വെബ്പോള്സുകളിൽ അവയുടെ പാർട്ട് തയ്യാറാക്കുക.

### നിങ്ങളുടെ പുരോഗതി അറിയാൻ



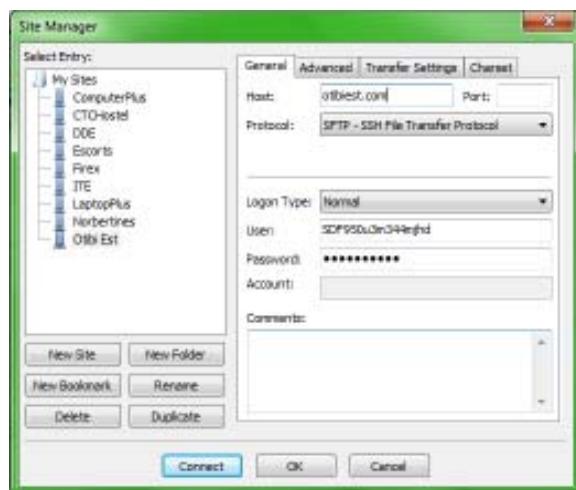
1. വെബ്പോള്സീറ്റ് സേവനം നൽകുന്ന കമ്പനികളെ \_\_\_\_\_ എന്നു പറയുന്നു.
2. വെബ്പോള്സീറ്റ് തരം നിശ്ചയിക്കുന്ന ഘടകങ്ങൾ തയ്യാറാക്കുക.
3. വി.പി.എസ്. എന്ത്
  - a. Virtual Premium Service
  - b. Virtual Private Service
  - c. Virtual Premium Server
  - d. Virtual Private Server
4. കോ-ലോക്കേഷൻ എന്നാലെന്ത്?
5. WHOIS വിവരങ്ങൾ ഉൾക്കൊള്ളുന്നതെന്നാകെ?
6. ഒരു ഡെബാമെൻ നെയ്തിമിന് A റിക്കോർഡ് പ്രാധാന്യമുള്ളതാകുന്ന തെന്തുകൊണ്ട്?

### 7.1.4 എഫ്.റി.പി. കൈസ്റ്റ്‌സോഫ്റ്റ്‌വെയർ (FTP client software)

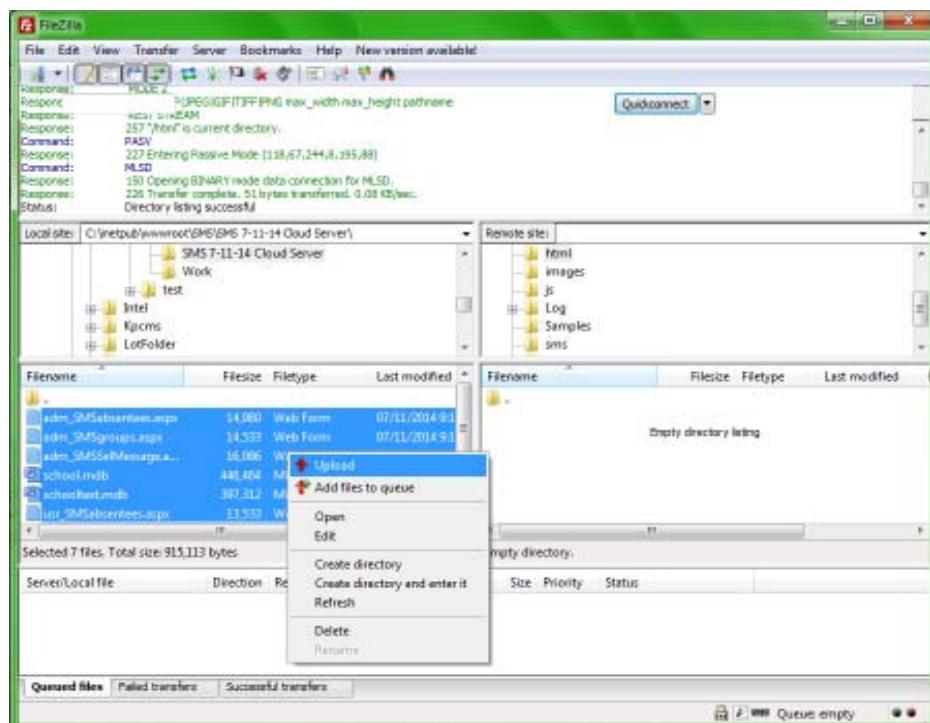
ഹോസ്റ്റിംഗിനായി സെർവ്വർ സ്ഥലവും ഡാബേമെൻ നെയിമും വാങ്ങിയശേഷം നമ്മുടെ കമ്പ്യൂട്ടറിൽ സൂക്ഷിച്ചിരിക്കുന്ന വെബ് സെറ്റിംഗ് ഫയലുകളെ വെബ് സെർവ്വറിലേയ്ക്ക് പകർത്തേണ്ട തായുണ്ട്. ഇതിനായി എഫ്.റി.പി. കൈസ്റ്റ്‌സോഫ്റ്റ്‌വെയർ ഉപയോഗിക്കുന്നു.

FTP യെ കൂടിച്ച് കമ്പ്യൂട്ടർ നേര്ത്തവർക്ക് എന്ന അഭ്യാസത്തിൽ 11-ാം ക്ലാസിൽ നാം ചർച്ച ചെയ്തു. ഇരുപ്പും നേര്ത്തവർക്ക് സൂക്ഷിച്ചിരിക്കുന്ന ഒരു കമ്പ്യൂട്ടറിൽ നിന്ന് മറ്റാരു കമ്പ്യൂട്ടറിലേയ്ക്ക് ഫയൽ നീക്കുന്നതിന് FTP ഉപയോഗിക്കുന്നു.

FTP CLIENT സോഫ്റ്റ്‌വെയർ വിഭൂത സെർവ്വറുമായി ബന്ധപ്പെട്ടുകയും അതിന്റെ കമ്പ്യൂട്ടറിലെ ഫയലുകൾ സെർവ്വർ കമ്പ്യൂട്ടറിലേയ്ക്ക് മാറ്റുകയും ചെയ്യുന്നു. FTP സെർവ്വറിൽ മാത്രം ബന്ധം ഉണ്ടാക്കുവാൻ FTP കൈസ്റ്റ്‌സോഫ്റ്റ്‌വെയറിന് യുസർനെയിൽ



ചിത്രം 7.8 : FTP സോഫ്റ്റ്‌വെയറിൽ ലോഗിൻ പ്രോസ്സ്



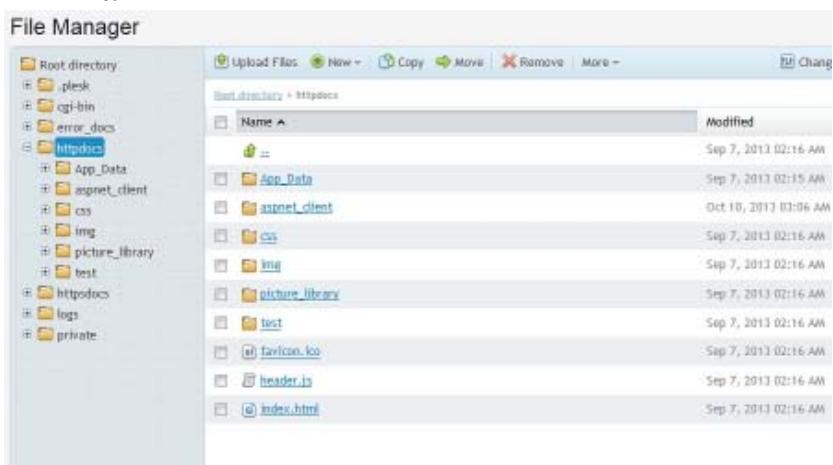
ചിത്രം 7.9 : FTP കൈസ്റ്റ്‌സോഫ്റ്റ്‌വെയറിൽ IDE

പാസ്വേർഡ് ദോഷമെന്ന് കൈയിലുണ്ടാക്കുന്നതിൽ ഒരു പ്രധാന കാരണമാണ്. ചിത്രം 7.8 തോറുന്നതിൽ കമ്പ്യൂട്ടർ സൈറ്റിൽ മാനേജർ ഡാബ്ലുഎൽ ഫോർമേറ്റിൽ ഉള്ള ലഭ്യമാക്കുന്നതിൽ സുരക്ഷിതമല്ലാത്ത ഒരു പ്ലാറ്റ്‌ഫോർമിൽ ടെക്നോളജിക്കൽ രൂപത്തിൽ ഉള്ള തുംബനാമവും പാസ്വേർഡും സെർവ്വിലേയ്ക്ക് എത്തുക്കും. എന്നാൽ ഇപ്പോൾ SSH, FTP പ്രോട്ടോക്കോൾ ഉള്ള വിവരങ്ങളെ മറ്റൊരു രൂപത്തിലാക്കി വിവരത്തെന്നും ചെയ്യുന്നതിൽ ഏപ്ര.റ്റി.പി.യുടെ സഹായത്തോടെ അയയ്ക്കുകയും ചെയ്യുന്നു. SFTP സെക്രേഷൻ പ്രോട്ടോക്കോൾ ഉപയോഗിച്ച് സുരക്ഷിതമായി മയൽനീക്കത്തിന് SFTP സഹായിക്കുകയും ചെയ്യും.

ചിത്രം 7.9 തോറുന്നതിൽ പ്രകാരം FTP കൈറ്റ് സോഫ്റ്റ് വെയറിംഗ് ആധികാരിക്കുന്ന ഉറപ്പിച്ച് കഴിത്താൽ അതിന്റെ IDE-ചിത്രം 7.9 ലേതുപോലെ പ്രത്യുക്ഷമാക്കും. ഈ ചിത്രത്തിൽ ഇടതുവശത്ത് പ്രദർശിപ്പിച്ചിരിക്കുന്ന ഫോർമേറ്റ് നമ്മുടെ കമ്പ്യൂട്ടറിലേതും വലതുവശതെന്ന് വെബ്സൈറ്റ് കമ്പ്യൂട്ടറിലേതും ആകുന്നു. മെനു ഉപയോഗിച്ചോ ഡ്രാഗ് ചെയ്തോ ഇടതുവശത്തുള്ള മയലിനെ വലതുവശത്തുള്ള ജാലകത്തിലേക്കാക്കി വെബ്സൈറ്റ് വെബ്സൈറ്റ് അപ്ലോഡ് ചെയ്യാം. ഇങ്ങനെ നമ്മുടെ കമ്പ്യൂട്ടറിലെ മയലുകൾ വെബ്സൈറ്റ് പകർത്താം. പ്രമുഖമായ ചില FTP സോഫ്റ്റ് വെയറുകളാണ് മയൽസില്ല, കൂടുതൽ FTP., സ്ഥാർട്ട് FTP. തുടങ്ങിയവ.

ചില വെബ്സൈറ്റുകൾ

ന്റീസിന്റെ കമ്പനികൾ  
അപ്ലാറ്റ്‌വെബ് ഡാബ്ലുഎൽ  
ചെയ്യുന്ന തിന്ന്  
അവരുടെ തായ്  
കൺട്രോൾ പാന  
ലൂകൾ സജ്ജീകരി  
ചീട്ടുണ്ട്. ഈ പുറത്ത് നിന്നുള്ള  
FTP കൈറ്റ്  
സോഫ്റ്റ് വെയർ  
വഴി മയൽ  
അപ്ലാറ്റ്‌വെബ് ഡാബ്ലുഎൽ  
ചെയ്യുവാൻ അനുവദിക്കുകയില്ല. ചിത്രം 7.10 ഇത്തരത്തിലുള്ള ഒരു കൺട്രോൾ പാനൽ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.



ചിത്രം 7.10 : ഫോറ്റോസിന്റെ സ്ഥാപനങ്ങൾ ലഭ്യമാക്കുന്ന FTP കൺട്രോൾ പാനൽ

## 7.2 സൗജന്യ ഹോസ്റ്റിംഗ് (Free hosting)

വെബ്സൈറ്റുകൾ സൗകര്യം സൗജന്യമായി നൽകുന്നതാണ് സൗജന്യ ഹോസ്റ്റിംഗ്. ഇവിടെ സർവ്വീസ് പ്രോവൈദ്ദർ ഉള്ള വെബ്സൈറ്റുകളിൽ ചില പരസ്യങ്ങൾ നൽകി ഇതിന്റെ ചെലവ് കുറയ്ക്കുന്നു. സൗജന്യവെബ്സൈറ്റുകൾ സെറ്റുകൾ നമ്മുടെ വെബ്സൈറ്റിലേ മയലുകളെ സെർവ്വിസ് സുരക്ഷിക്കുന്നു. എന്നാലിതിന് ചില നിയന്ത്രണങ്ങളുണ്ട്. മയലിന്റെ വലുപ്പം നിയന്ത്രണവിധേയമായായിരിക്കും (5 MB വരെ) ശമ്പദവും വീഡി

## 7. വെബ് ഹോസ്റ്റിംഗ്

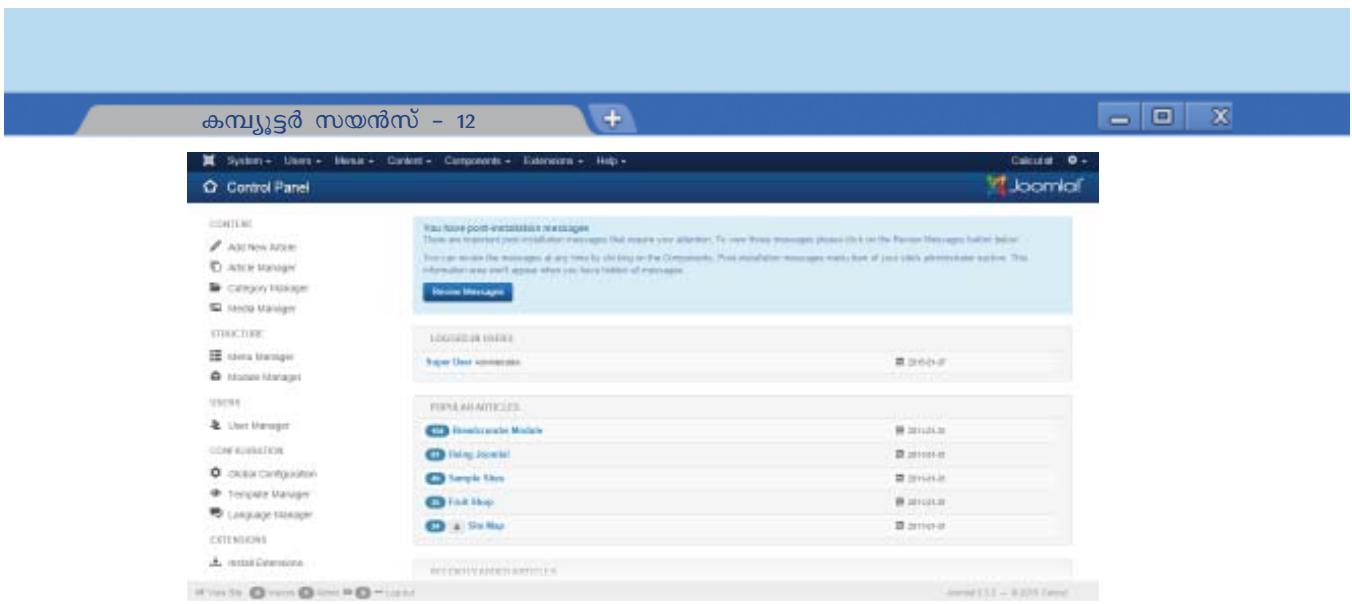
യോധും ചിലതിൽ അനുവദനീയമല്ല. ചില വെബ്സൈറ്റുകൾ അതിന്റെ ഒരു ടെംപ്ലറ്റ് അടിസ്ഥാനമാക്കി വെബ്സൈറ്റ് രൂപ കൽപ്പന ചെയ്യുവാൻ മാത്രമേ അനുവദിക്കുകയുള്ള തു. മറ്റ് ഫയലുകൾ വെബ്സൈറ്റുവിൽ അപ്ലോഡ് ചെയ്യുവാനും അനുവദിക്കില്ല. ചില സൗജന്യവെബ്ഹോസ്റ്റിംഗ് കമ്പനികൾ ദിശാമെന്ന് നേരിം രജിസ്ട്രേഷനും അനുവദിക്കുന്നു. ഈ വെബ്സൈറ്റുകൾ ഫയലുകളെ അപ്ലോഡ് ചെയ്യുന്നതിനുള്ള കൺട്രോൾ പാനലുകൾ ചിത്രം 7.10 ത്തിലേതുപോലെ നൽകുന്നു. ചില വെബ്സൈറ്റുകൾ മുൻ നിശ്ചയിച്ച ഘടനയിൽ മാത്രമേ വെബ്സൈറ്റ് നിർമ്മിക്കുവാൻ അനുവദിക്കുകയുള്ളതു. സൗജന്യ വെബ്ഹോസ്റ്റിംഗ് സൗകര്യം അവരുടേതായ ഒരു സബ്സയാമെന്നോ അല്ലെങ്കിൽ ഒരു ഡയറക്ടറി സർവീസോ ഉപയോഗിച്ചാണ് നമ്മുടെ വെബ്സൈറ്റിനെ ലഭ്യമാക്കുവാൻ സഹായിക്കുന്നത്. ഒരേ ആഴയങ്ങളുള്ളവരുമായി വിവരങ്ങൾ പകിടാനും, ലാഡേ പ്ലാറ്റ്ഫോർമ്മുകളും പ്രസ്ഥാനങ്ങൾക്കും അധിക ചെലവ് വഹിക്കാനാകാത്തവർക്കും സൗജന്യ വെബ്ഹോസ്റ്റിംഗ് സഹായകരമാണ്. വളരെ ചെലവ് കുറഞ്ഞ രീതിയിൽ വെബ്ഹോസ്റ്റിംഗ് സൗജന്യവെബ്ഹോസ്റ്റിംഗ് സൗകര്യം നിലവിൽ വന്നതോടുകൂടി സൗജന്യ വെബ്ഹോസ്റ്റിംഗ് കുറഞ്ഞവരും. sites.google.com, yola.com എന്നിവ സൗജന്യ വെബ്ഹോസ്റ്റിംഗ് സർവീസുകളാണ്. ഹയർസൈക്കൽസിൽ അഖ്യാപകർക്കും വിദ്യാർത്ഥികൾക്കും ഏറെ പ്രയോജനകരമായ വിവരങ്ങൾ നൽകുന്ന hsslive.in എന്ന വെബ്സൈറ്റിന് സൗജന്യ ഹോസ്റ്റിംഗ് ആണ് ഉപയോഗിച്ചിരിക്കുന്നത്. ചിത്രം 7.11 തു ഈ പ്രദർശിപ്പിച്ചിരിക്കുന്നു.



ചിത്രം 7.11 : സൗജന്യ വെബ്ഹോസ്റ്റിംഗ്

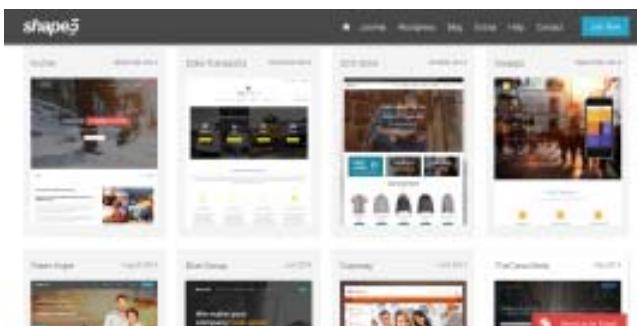
### 7.3 കണക്ക് മാനേജ്മെന്റ് സിസ്റ്റം (Content Management System)

വെബ്സൈറ്റുകൾ നിർമ്മിക്കുന്നതിനും നിയന്ത്രിക്കുന്നതിനും അവയിലേയ്ക്ക് പുതിയ വിവരങ്ങൾ ഉൾപ്പെടുത്തുന്നതിനും സാധ്യമായ ഒരു ഇൻററെന്റ് അധിഷ്ഠിതമായ സോഫ്റ്റ്‌വെയർ സംവിധാനമാണ് കണക്ക് മാനേജ്മെന്റ് സിസ്റ്റം. മിക്ക CMS കളും അവയുടെ ഒന്നേറ്റാഗിക വെബ്സൈറ്റിൽ നിന്ന് സൗജന്യമായി ധനാദാലോധ്യ ചെയ്യാം. ഈ ഫയലുകളെ നമ്മുടെ വെബ്സൈറ്റിലെ ഹോസ്റ്റിംഗ് സമലതേതയ്ക്ക് കോപ്പി ചെയ്യാം. അതി



**ചിത്രം 7.12 :** ഇന്ത്യൻ ഉപയോഗിച്ച് വെബ്സൈറ്റ് പ്രവർത്തന സജ്ജമാക്കുന്നു

നുശേഷം ചിത്രം 7.12 ലെപോലെ  
അതിനെ പ്രവർത്തനസജ്ജമാക്കു  
കയും ചെയ്യാം. വെബ്സൈറ്റുക  
ളിൽ CMS കളായി മുൻപ് രൂപക  
ല്പന ചെയ്ത് വച്ചിട്ടുള്ള ചില  
ടെംപ്ലറ്റുകൾ ലഭ്യമാക്കും. ചിത്രം  
7.13 ത്ത് ഇവയുടെ ഉദാഹരണം  
നൽകിയിരിക്കുന്നു. ഉപയോക്താ  
വിന് ഈ ടെംപ്ലറ്റുകൾ ഒന്ന് തെര  
ഞ്ഞടക്കത്ത് നമ്മുടെ വെബ്സൈ  
റ്റിൽ അപ്പലോഡ് ചെയ്യാം. അതി  
വിവരങ്ങളും നമ്മുടെ ഇഷ്ടപ്രക



**ചിത്രം 7.13:** ജൂമ്പയിൽ ലഭിക്കുന്ന ടെംപ്പേറേച്ചർ



ചിത്രം 7.14 : ഇന്തു CMS ഉപയോഗിച്ച് വികസിപ്പിച്ച ഒരു ബൈബിൾപേജ്

മാൻ. ചില വ്യക്തികൾ CMS. നിശ്ചിത തുകയ്ക്ക് നമ്മുടെ വെബ്സൈറ്റിന് അനുയോജ്യമാം വിധം ക്രമപ്പെടുത്തി നൽകും. ഒരു ഷൈയേർഷ്യ വെബ്സൈറ്റ് വെബ്സൈറ്റ് ഹോസ്റ്റ് ചെയ്യുന്നതെങ്കിൽ അതിൽ CMS ഹയലുകൾ ഉൾപ്പെടുത്താനാകുമോ എന്ന് ശ്രദ്ധിക്കണം.

സാമ്പത്തികലാഭം ഉള്ളതുകൊണ്ട് ധാരാളം സംഘടനകളും ബ്ലോഗറുകളും അവരുടെ വെബ്സൈറ്റിനായി സി.എം.എസ്. ഉപയോഗിക്കുന്നു. പ്രമുഖമായ ചില C M S സോഫ്റ്റ്‌വെയറുകളാണ് വേഡ്പ്രൈസ്, ദ്രോപ്സ്റ്റ്, ജൂംല എന്നിവ ചിത്രം 1.4ൽ ജൂംല ഉപയോഗിച്ച് നിർമ്മിച്ച മോട്ടോർ വെഹിക്കിൾവകുപ്പിന്റെ വെബ്സൈറ്റ് കാണാനാകും.

#### **7.4 റെസ്പോൺസീവ് വെബ്സൈറ്റ് പകൽപ്പന (Responsive web design)**

ഈ ഇൻറൈറ്റേറുമായി ബന്ധപ്പെട്ടുവാൻ കമ്പ്യൂട്ടർ, ലാപ്ടോപ്പ്, ടാബ്ലെറ്റ്, മൊബൈൽഫോൺ തുടങ്ങിയ ധാരാളം ഉപകരണങ്ങൾ ഉണ്ട്. പല ഉപകരണങ്ങൾക്കും വ്യത്യസ്ത സ്ക്രീൻവലൂപ്പമാണുള്ളത്. പണ്ട് നിർമ്മിച്ചിട്ടുള്ള വെബ്പേജുകൾ ദേശക്കും ടോപ്പിനും ലാപ്ടോപ്പിനും അനുയോജ്യമാംവിധം ഉള്ളതാണ്. ഈ വെബ്സൈറ്റുകൾ ടാബ്ലെറ്റും, മൊബൈൽഫോൺും വഴി കാണുവാൻ പ്രയാസമാണ്. അല്ലെങ്കിൽ ഉപഭോക്താവായ ഈ വെബ്പേജിന്റെ ഓരോ ഭാഗവും കാണുവാൻ സ്ക്രോൾബാർ വഴങ്ങളിലേയ്ക്ക് നീകിലി നോക്കണം. മുൻകാലത്ത് മൊബൈലിവഴി വെബ്സൈറ്റ് കാണുവാൻ പ്രത്യേക വെബ്സൈറ്റുകൾ നിർമ്മിച്ചിരുന്നു. ഈവ മൊബൈലിന്റെ സ്ക്രീൻ വലുപ്പത്തിന് അനുയോജ്യമാംവിധമാണ് നിർമ്മിച്ചിരുന്നത്. എന്നാൽ ഒരു സ്ഥാപനത്തിന് രണ്ട് വെബ്സൈറ്റുകൾ കൈകാര്യം ചെയ്യേണ്ടിവരുന്നത് പ്രയാസമാണ്. എന്നാൽ വെബ്പേജിന്റെ വലുപ്പ് സ്ക്രീനിന്റെ വലുപ്പത്തിന് അനുസരിച്ച് മാറുന്ന രീതിയിൽ രൂപപ്പെടുത്തിയാൽ അത് കുടുതൽ പ്രയോജനകരമാകും. ഇത്തരത്തിലുള്ള വെബ്പോജ് രൂപകൽപ്പനയെ റെസ്പോൺസീവ് വെബ്സൈറ്റ് പകൽപ്പന എന്നു പറയും. വിവിധ ഉപകരണങ്ങളിൽ ഒരു റെസ്പോൺസീവ് വെബ്സൈറ്റ് എങ്ങനെ ദ്രുത്യാനകുമെന്ന് ചിത്രം 7.15 തുടർന്ന് കാണിച്ചിരിക്കുന്നു.

എല്ലാ ഉപകരണങ്ങളിലും പ്രവർത്തിക്കുത്തക വിധമാണ് റെസ്പോൺസീവ് വെബ്ഡിസൈനിങ്ങിൽ വെബ്സൈറ്റിനെ രൂപപ്പെടുത്തുന്നത്. അതുകൊണ്ട് തന്നെ സ്ക്രീൻ വലുപ്പമോ ഉപകരണമേരെതന്നോ ഇവിടെ പരിശീലനപ്പെടുന്നില്ല. റെസ്പോൺസീവ് വെബ്ഡിസൈനിംഗ് എന്ന ആശയം ഒരു സത്ത്ര ഡിസൈനറും ഇമാൻ മർക്കോട്ടിന്റെതാണ്. സ്ഥിരമായി മാറിക്കൊണ്ടിരിക്കുന്ന ഒരു വെബ്സൈറ്റ് രീതിയെപ്പറ്റി അദ്ദേഹം വിവരിക്കുന്നു. ഈ വെബ്സൈറ്റ് രീതിയിൽ ഫ്ലക്സിബിൾ ഗ്രിഡ് ലേബോട്ടും ഫ്ലക്സിബിൾ ഇമേജും മീഡിയക്കൊണ്ട് ഉപയോഗിച്ച് പ്രാവർത്തികമാക്കിയിരിക്കുന്നു. ഫ്ലക്സിബിൾ ഗ്രിഡ് ലേബോട്ട്, സൗകര്യം ഉപയോഗിച്ച് ഉപകരണത്തിന്റെ വലുപ്പം പരിഗണിച്ച് വെബ്പോജിനെ അതിന് അനുസൃത



Fig. 7.15 : Responsive web design

മാതി സജീകരിക്കുകയും ചെയ്യും. മീഡിയക്കാർ ഓരോ ഉപകരണത്തിനും അനുസരിച്ച് ഇളം വ്യത്യസ്ത ശൈലിയിൽ ലഭ്യമാക്കും. ഒരു വെബ്പേജിലെ തിരഞ്ഞീൻ മെനു മൊബൈലിൽ ഫോപ്പഡാണ് മെനു ആയി പരിവർത്തനം ചെയ്യപ്പെടും. CSS ഹയലിനുള്ളിലെ മീഡിയക്കാറിയാണ് ഈതിന് സഹായിക്കുന്നത്. ഈവിടെ പ്രദർശനത്തിന്റെ വലുപ്പം മാറുന്നതിന് അനുസരിച്ച് വെബ്സൈറ്റിന്റെ വലുപ്പവും ഭാവവും മാറിക്കൊണ്ടിരിക്കും.



**ഇന്ത്യൻ കീകൾ ടീം അംഗങ്ങളുടെ പേര്, വയസ്സ്, നേടിയ റണ്ടുകളുടെ എണ്ണം, എടുത്ത വികരുകളുടെ എണ്ണം എന്നിവ ഉൾക്കൊള്ളുന്ന ഒരു വെബ്സൈറ്റ് രൂപകൽപ്പന ചെയ്ത് ട്രേഡ് ഹോസ്റ്റിംഗ് സേവനങ്ങൾ ഉപയോഗിച്ച് ഈതിനെ ഹോസ്റ്റ് ചെയ്യുക.**

**നമുക്ക് ചെയ്യാം** ജനകീയമായ CMS ലഭ്യമാക്കുന്നതിലൂടെ ഒരു ലിസ്റ്റ് തയ്യാറാക്കുന്നതിനോടൊപ്പം അവരുടെ സവിശേഷതകളും തയ്യാറാക്കുക.

നിങ്ങൾ തയ്യാറാക്കിയ സ്കൂൾ വെബ്സൈറ്റുകളുടെ ഹയലുകൾ ഏതെങ്കിലും FTP സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് ലഭ്യമായ വെബ്ഹോസ്റ്റിംഗ് സ്ഥലത്ത് അപ്ലോഡ് ചെയ്യുക.



## നമുക്ക് സംഗ്രഹിക്കാം

ഒരു വെബ്സൈറ്റ് നിർമ്മിച്ചതിനുശേഷം അനുയോജ്യമായ ഒരു ഹോസ്റ്റിംഗ് റീതി ഉപയോഗിച്ച് ഇരുപ്പേരെന്തീൽ ലഭ്യമാക്കാം. ചെറിയ സ്ഥാപനങ്ങൾ തങ്ങളുടെ വെബ്സൈറ്റിന് ഷൈയർലീഡ് ഹോസ്റ്റിംഗ് ഉപയോഗിക്കും. ഉയർന്ന ട്രാഫിക്കുള്ളതും ഉയർന്ന സുരക്ഷ വേണ്ടതുമായ വെബ്സൈറ്റുകൾക്ക് വി.പി.എസ്. ഹോസ്റ്റിംഗ് ഉപയോഗിക്കും. എന്നാൽ വലിയ സ്ഥാപനങ്ങളുടെ വെബ്സൈറ്റുകളിൽ നിരന്തരം ഉയർന്ന ട്രാഫിക്കുകൾ ഉണ്ടാകുന്നതുകൊണ്ട് ഡെബിക്കേറ്റർ ഹോസ്റ്റിംഗ് ഉപയോഗിക്കും. വെബ്ഹോസ്റ്റിംഗിന് ആവശ്യമായ സ്ഥലം വാങ്ങിയതിനുശേഷം FTP സോഫ്റ്റ്‌വെയർ ഉപയോഗിച്ച് നമ്മുടെ വെബ്സൈറ്റ് ഹയലുകളെ വെബ്സൈറ്റ്‌വിലേയ്ക്ക് മാറ്റാം അതുകൊമുഖ്യ ഡെബിക്കേറ്റർ നേയിം സേവന ദാദാവ് വഴി രജിസ്ട്രർ ചെയ്യണം. വെബ്സൈറ്റ്‌വിലെ നമ്മുടെ ഡെബിക്കേറ്റർ പോയിന്തു ചെയ്യുവാൻ A റിക്കോർഡ് സജീവമാക്കണം. സൗജന്യനിരക്കിൽ വെബ്സൈറ്റിനെ ലഭ്യമാക്കുന്ന വെബ്സൈറ്റ്‌വിലുകൾ ലഭ്യമാണ്. കണ്ണറ്റ് മാനേജ്മെന്റ് സംവിധാനം ചില ഉപകരണങ്ങളും ചില അടിസ്ഥാന സുരക്ഷിത സൗകര്യങ്ങളും ഒരുക്കി സാങ്കേതികവിദ്യകളിൽ അറിയപ്പെടുന്നതു ആളുകൾക്കുപോലും വെബ്സൈറ്റുകൾ നിർമ്മിക്കുവാനുള്ള സാധ്യത നൽകുന്നു. വ്യത്യസ്ത ഉപകരണങ്ങൾ ഉപയോഗിച്ച് ബേബിൾ ചെയ്യുന്നതുകൊണ്ട് സ്ക്രൈപ്റ്റിംഗ് വലുപ്പമനുസരിച്ച് വെബ്സൈറ്റുകൾ ലഭ്യമാക്കുക എന്നത് പ്രായാനുത്തേരീയ കാണേണ്ടിയിരിക്കുന്നു. ഈതിനായി റിസ്പോൺസീവ് വെബ്സൈറ്റുകൾ ഉപയോഗിക്കുന്നു.

### നമുക്ക് പ്രവർത്തനങ്ങളിലേർപ്പോം

1. വെബ് ഹോസ്റ്റിംഗ് എന്നതുകൊണ്ട് എന്താണ് നിങ്ങൾ അർഥമാക്കുന്നത്? പലതര തിലുള്ള വെബ് ഹോസ്റ്റിംഗിനെ കുറിച്ച് വിശദീകരിക്കുക.
2. നഗരത്തിലുള്ള ഒരു സൂപ്രമാർക്കട്ട് സ്ഥാപനം അവരുടെ വ്യാപാരം ഓൺ ലൈൻ ആക്കുവാൻ ഉദ്ദേശിക്കുന്നു. വെബ് സേസ്റ്റ് വഴി സാധനങ്ങൾ വിൽക്കാനും പണ്മിട പാടുകൾ ഓൺലൈൻ വഴി നടത്തുവാനും ആസുത്രണം ചെയ്യുന്നു. എങ്കിൽ
  - a. എത്ര തരത്തിലുള്ള ഹോസ്റ്റിംഗ് ആണ് ഈ വെബ് പേജിന് അനുയോജ്യമാ കൂട്.
  - b. അതി തിരഞ്ഞെടുക്കുവാനുള്ള കാരണവും വിശദമാക്കുക.
3. എമിൽ സ്വന്തം മരുന്നു കടയുടെ വെബ് സേസ്റ്റിനായി വെബ് ഹോസ്റ്റിംഗ് സഹലം വാങ്ങുവാൻ ആഗ്രഹിക്കുന്നു. വെബ് സർവ്വിസ് ഹോസ്റ്റിംഗിനായി സഹലം വാങ്ങു പോൾ പരിശീലനം പ്രത്യേകതകൾ പട്ടികപ്പെടുത്തുക.
4. ഒരു ഡോക്യുമെന്റ് നാമത്രേതയും വെബ് സർവ്വിസ് സംഭരിച്ചിട്ടുള്ള വെബ് സേസ്റ്റും തമിൽ എങ്ങനെയാണ് പരസ്പരം ബന്ധിപ്പിക്കുന്നത്?
5. FTP സോഫ്റ്റ് വെയറിൽ SFTP ചട്ടം (പോട്ടോകോൾ) ഉപയോഗിക്കുന്നത് കൊണ്ടുള്ള മേമ്പയന്താണ്?
6. സ്വന്തം കൂടുംബത്തിനായി ഒരു വെബ് സേസ്റ്റ് ഹോസ്റ്റ് ചെയ്യുവാൻ രാജു ആഗ്രഹിക്കുന്നു. സൗജന്യ വെബ് സേസ്റ്റ് ഹോസ്റ്റിംഗ് കമ്പനികൾ നൽകുന്ന സംവിധാനങ്ങൾ എന്താക്കേയാണ്?
7. CMS എന്നാൽ എന്താണ്? CMS ഏത് പ്രത്യേകതകൾ എന്താക്കും? ഉദാഹരണങ്ങൾ നൽകുക.
8. ഇന്നത്തെ കാലത്ത് വെബ് സേസ്റ്റിന്റെ രൂപകൽപ്പനയിൽ റിസ്പോൺസീവ് വെബ് സേസ്റ്റ് രൂപകൽപ്പനയുടെ ആവശ്യകത വിശദീകരിക്കുക.
9. എങ്ങനെയാണ് റിസ്പോൺസീവ് വെബ് പേജ് രൂപകൽപ്പന നടപ്പിലാക്കുന്നത്?

## പദ്രോവോ

Abstract Data Types	അബ്സ്ട്രക്ട് ഡാട്യൂപ്പസ്
Access	ഉപയോഗിക്കുക
AccessSpecifier	ആക്സസ് സ്പെസിഫൈർ
Activity	പ്രവർത്തനം
Address	വിലാസം
Allocate	നീക്കി വയ്ക്കുക
Annual Registration Fees	വാർഷിക വരിസംഖ്യ
Architecture	രൂപൊലാറ്റന
Argument	ആർഗ്യൂമെന്റ്
Array	അരേ
Artificial Intelligence	നിർമ്മിത ബൃഹി
Assignment statement	വില നൽകൽ പ്രസ്താവന
Authentication	അയികാരിക്കൽ
Base Class	ബേസ് ക്ലാസ്
Base number	ആധാരസംഖ്യ
Bottom Up	ബേംടം അപ്
Builtin	അന്തർനിർമ്മിത
Circular Queue	വൃത്തക്കു
Class	ക്ലാസ്
Compile Time Polymorphism	കംപയിൽ ദെണ പോളിമോർഫിസം
Compound Datastructure	സകീറ്റണമായ ഡാറ്റാ സ്റ്റോക്ഷർ
Computing Discipline	കമ്പ്യൂട്ടിങ് വിജ്ഞാനശാഖ
Computing machine	കമ്പ്യൂട്ടിങ് യന്ത്രം
Content	ഉള്ളടക്കം
Copy	പകർത്തുക
Data Abstraction	ഡാറ്റാ അബ്സ്ട്രക്ട്
Data Elements	ഡാറ്റാ അംഗങ്ങൾ
Data Type	ഡാറ്റ ഇനം
Datastructure	ഡാറ്റാ സ്റ്റോക്ഷർ
Declaration	പ്രവ്യാപനം
Declaration statement	പ്രവ്യാപന പ്രസ്താവന
Declare	പ്രവ്യാപിക്കുക.
Default value	തനതുവില
Definition	നിർവചനം
Deletion	നീക്കംചെയ്യുക
Derived Class	ഡിരേവ്യു ക്ലാസ്
Derived data type	രൂപീകൃത ഡാറ്റ ഇനം
Design	രൂപ കല്പന

Develop	നിർമ്മാണം
Display	പ്രദർശിപ്പിക്കുക
Domain Name	ഡേബാമെൻഡ് പേര് , ഡേബാമെൻഡ് നാമം
Double Linked List	ധാരിച്ച ലിക്ക് ലിസ്റ്റ്
Drawback	നൃസന്ധി
Dynamic Datastructure	അസ്ഥിര ഡാറ്റാ സ്റ്റ്രക്ചർ
Encrypt	രൂപമാറ്റം വരുത്തുക
Execute	പ്രവർത്തിപ്പിക്കുക
Expense	ചെലവ്
Expression	പദ്ധതിയോഗം
Floating Point number	അസ്ഥിര ഭാംഗം സംഖ്യ
Function	ഫംഷൻ
Fundamental data type	അടിസ്ഥാന ഡാറ്റ ഇനം
Global	ഗ്ലോബൽ
Header	ഹൈഡർ
Hierachial Classification	ക്രമപ്രകാരമുള്ള വർഗീകരണം
Hierachial Inheritance	ഹൈറോകിയർ ഇൻഹരിറ്റൻസ്
Hover Craft	ഹോവർ ക്രാഫ്റ്റ്
Implementation	പ്രവർത്തനതലം
Initialisation statement	പ്രാരംഭ വില നൽകൽ പ്രസ്താവന
Initialise	പ്രാരംഭ വില നൽകൽ
Input	ഇൻപുട്ട്
Inserte	ഇൻസ്രൈട്ടുത്തുക
Instance	ഇൻസ്റ്റൻസ്
Integrtiy	സമഗ്രത
Interface	സൗഖ്യകമുഖ്യം
Keyword	കീവേഴ്സ്
Linear	രേഖീയം
Link List	ലിങ്ക് ലിസ്റ്റ്
Maintainabilty	പരിപാലനം
Member	മെമ്പർ
Member Function	മെമ്പർ ഫംഷൻ
Memory address	മെമ്മറി വിലാസം
Memory location	മെമ്മറിസ്ഥാനം
Memory size	മെമ്മറി വലുപ്പം
Merging	ലയിക്കുക
Modular	മോഡ്യൂലാർ
Modularitiy	മോഡ്യൂലാരിറ്റി
Module	വസ്യം

Multi level Inheritance	മൾടിലൈവൽ ഇൻഹെറിറ്റൻസ്
Multiple Inheritance	മൾടിപിൾ ഇൻഹെറിറ്റൻസ്
Nested	നെസ്റ്റഡ്
Node	നോഡ്
Non liner	നോൺലൈൻ
Non Profit	നോൺപ്രോഫിറ്റ്
Null Pointer	നൂളിൽപ്പിനർ
Object Oriented	ഓബ്ജക്ട് ഓറിയെന്റെഡ്
Objects	ഓബ്ജക്ടുകൾ
Operations	പ്രവർത്തനങ്ങൾ
Operator	ഓപ്പറേറ്റർ
Operator Overloading	ഓപ്പറേറ്റർ ഓവർലോഡിംഗ്
Output	ഇട്ട്‌പുക്ക്
Performance	കാര്യക്ഷമത
Pointer	പോയിന്റർ
Positional value numbers ystem	സ്ഥാനീയ സംവ്യാന സ്വന്ധായം
postfix	പോസ്റ്റ്‌ഫിക്സ്
prefix	പ്രൈംറിക്സ്
Primary Memory	പ്രാഥമിക മെമ്മറി
Print Buffer	പ്രിൻ്റ് ബഫർ
Problem solving	പ്രശ്ന സൊള്വിംഗ് നിർധാരണം ചെയ്യുക
Procedure	നടപടിക്രമം
Process	പ്രോസസ്
Programming Paradigm	പ്രോഗ്രാമിംഗ് മാതൃക
Push	പൂഷ്
Queue	ക്യൂ
Queue Overflow	ക്യൂ ഓവർഫെല്ലോ
Queue Underflow	ക്യൂ അംബർഫെല്ലോ
Refer	സൂചിപ്പിക്കുക
Remote Server	വിദുത സൈറ്റ്‌വർ
Restart	പുനരാരംഭിക്കാൻ
Reusability	പുനരുപയോഗം
Run time Polymorphism	റിസ് ഫെഡ് പോളിമാർപ്പിംഗ്
R-Value	R വില
Screen size	സ്ക്രീൻ വലുപ്പം
Search	തിരയൽ
Secondary Memory	ഡിതീയ മെമ്മറി
Secure	സുരക്ഷിതം
Segment	ശകളം
Self referential	സ്വയം സൂചിത

Self Referential tStructures	സ്വയം സുചിത സ്ട്രക്ചർ
Server Update	സൈറ്റ്‌വർ നവീകരണം
Set up	ക്രമീകരിക്കുക
Simple Datastructure	പ്രാഥമികമായ ഡാറ്റാ സ്റ്റ്രക്ചർ
Single Inheritance	സിംഗിൾ ഇൻഹെരിറ്റൻസ്
Singly Linked List	സിംഗിൾ ലിങ്ക്ഡ് ലിസ്റ്റ്
Software	സോഫ്റ്റ്‌വെയർ
Sorting	ക്രമപ്പെടുത്തൽ
Stack	സ്റ്റാക്ക്
Stack Overflow	സ്റ്റാക്ക് ഓവർഫോള്റ്
Stack Underflow	സ്റ്റാക്ക് അൻഡർഫോള്റ്
Statement	പ്രസ്താവന
Static Datastructure	സ്ഥിര ഡാറ്റാ സ്റ്റ്രക്ചർ
Storage Space	സംഭരണ സ്ഥലം
Store	സുക്ഷിക്കുക
Structure	സ്റ്റ്രക്ചർ
Symbol	ചിഹ്നം
Symbolic	പ്രതീകാത്മക
Syntax	വാക്യ ഘടന
Template	ഘടന
Theory of computing	കമ്പ്യൂട്ടിംഗ് സിഭാന്തം
Top Down	ടോപ്പ് ഡാൻസ്
Traversal	കടന്നുപോകൽ
Trigonometry	ത്രികോൺമിതി
Type modifier	ടൈപ്പ് മോഡിഫയർ
User defined	ഉപയോക്തൃനിർവ്വചിത
User Friendly	ഉപയോക്തൃ സൗഹ്യദാം
Value	മൂല്യം
Value	വില
Water Vehicle	ജല വാഹനം
Write	നല്കകൽ



## பாரம்பரை/நுங்கம் (References)

- Stroustrup, B. (2013). *The C++ Programming Language*. New Delhi : Addison-Wesley Professional
- Lafore, R. (2009). *Object-Oriented Programming in C++*. Chennai : Sams Publishing
- Balagurusamy, E. (2008). *Object Oriented Programming with C++*. New Delhi: Tata McGraw-Hill Education
- Sharma, A. K. (2011). *Data structure using C*. New Delhi : Pearson Education India
- Srivastava, S. K. & Srivastava, D. (2011). *Data Structures Through C in Depth*. New Delhi : BPB Publications
- Powel, T. A. (2010). *The Complete Reference : HTML & XHTML*. New Delhi: OSborne/Tata MC Graw-Hill
- Lloyd, I. (2008). *The Ultimate HTML Reference*. Melbourne : Sitepoint
- Holzner, S. (2000). *HTML Black Book*. New Delhi : DreamTech Press
- Frain, B. (2012). *Responsive Web Design with HTML5 and CSS3*. Mumbai : Packt Publishing
- Powel, T. A. & Schenider, F. (2008). *The Complete Reference JavaScript*. New Delhi : Tata McGraw-Hill
- Zakas, N. C. (2012). *Professional JavaScript for Web Developers*. Birmingham : Wrox