R9N7R1

6 ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ചുള്ള ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിങ്

പ്രധാന പഠനനേട്ടങ്ങൾ

ഈ അധ്യായത്തിന്റെ പഠനം പൂർത്തിയാകുന്നതോടെ പഠിതാവ് ആർജിക്കേണ്ട പഠനനേട്ടങ്ങൾ

- ക്ലൈന്റ് വശത്തെയും സെർവർ വശത്തെയും സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ ഉപയോഗം തിരിച്ചറിയുന്നു.
- ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ ആവശ്യകത വിശദീകരിക്കുന്നു.
- ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷയായ ജാവാസ്ക്രിപ്റ്റിന്റെ പ്രാധാന്യം തിരിച്ചറിയുന്നു.
- വെബ്പേജിൽ ജാവാസ്ക്രിപ്റ്റ് ഫംഗ്ഷൻ ഉപയോഗിക്കുന്നു
- ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ ഡാറ്റ ഇനങ്ങൾ വിശദീകരിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിൽ ശരിയായ വേരിയബിളുകൾ ഉപയോഗിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിൽ അനുയോജ്യമായ കൺട്രോൾ സ്ട്രക്ച്ചറുകൾ ഉപയോഗിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിൽ അനുയോജ്യമായ അന്തർനിർമിത ഫംഗ്ഷനുകൾ ഉപയോഗിക്കുന്നു.
- ജാവാസ്ക്രിപ്റ്റിലെ ഡോക്യുമെന്റ് ഘടകങ്ങൾ ഉപയോഗിക്കുന്ന രീതി വിവരിക്കുന്നു.
- ടെക്സ്റ്റ് ബോക്സുകളിലേയും കോംബോ ബോക്സുകളിലേയും വിലകൾ കൈകാര്യം ചെയ്യുന്നതിനുള്ള ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷനുകൾ നിർമിക്കുന്നു.

മുൻ അധ്യായങ്ങളിൽ ടെക്സ്റ്റുകളും ഗ്രാഫിക്സുകളും ഉൾപ്പെടുന്ന വിവിധ തരം വെബ് പേജുകൾ നിർമിക്കുവാൻ നാം പഠിച്ചു. ഈ ഇന്റർനെറ്റ് യുഗത്തിൽ നമ്മളിൽ പലരും വിവിധ ആവശ്യങ്ങൾക്കായി അനവധി വെബ്സൈറ്റുകൾ സന്ദർശിക്കാറുണ്ട്. നമുക്ക് പരിചിതമല്ലാത്ത അനേകം സവിശേഷതകൾ ഇത്തരം വെബ് പേജുകളിൽ അടങ്ങിയിരിക്കുന്നു. അതുകൊണ്ടുതന്നെ ഇത്തരം വെബ് പേജുകൾ നിർമിക്കുന്നതിനായി സ്ക്രിപ്റ്റിംഗ് ഭാഷകളിലുള്ള അറിവ് ആവശ്യമാണ്. ക്ലൈന്റ് സൈഡിലും സെർവർ സൈഡിലും വിവിധയിനം സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ ഉപയോഗിക്കുന്നുണ്ട്. ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകളായ ജാവാസ്ക്രിപ്റ്റ്, വിബി സ്ക്രിപ്റ്റ് എന്നിവയിൽ ജാവാസ്ക്രിപ്റ്റാണ് കൂടുതലായി ഉപയോഗിക്കുന്നത്. കാരണം വിബി സ്ക്രിപ്റ്റിനെ അപേക്ഷിച്ച് ജാവാസ്ക്രിപ്റ്റിനെയാണ് കൂടുതൽ ബ്രൗസറുകളും പിന്തുണയ്ക്കുന്നത്. ഇന്റർനെറ്റിൽ അനേകം ആളുകൾ വ്യത്യസ്ത ബ്രൗസറുകൾ ഉപയോഗിക്കുന്നതിനാൽ വെബ് പേജുകൾ കഴിയുന്നത്ര എല്ലാ ബ്രൗസറുകൾക്കും അനുയോജ്യമായ രീതിയിൽ നിർമിക്കേണ്ടതുണ്ട്. ഒരു വെബ് പേജിൽ ജാവാസ്ക്രിപ്റ്റ് എങ്ങനെ ഉപയോഗിക്കാമെന്ന് ഈ അധ്യായത്തിൽ മനസ്സിലാക്കാം. നമുക്ക് C++ പരിചയമുള്ളതിനാൽ ജാവാസ്ക്രിപ്റ്റ് മനസ്സിലാക്കുവാൻ വളരെ എളുപ്പമാണ്. കാരണം C++ ന്റെ അതേ വാക്യഘടനയാണ് ജാവാസ്ക്രിപ്റ്റ് പിന്തുടരുന്നത്.

ബ്രെൻഡൻ ഇച്ച്

നെറ്റ്സ്കേപ്പ് ബ്രൗസറിനുവേണ്ടി 'ബ്രെൻഡൻ ഇച്ച്' ആണ് ജാവാസ്ക്രിപ്റ്റ് വികസിപ്പിച്ചത്. ജാവാസ്ക്രിപ്റ്റിന്റെ ആദ്യ നാമം 'മോച്ച' എന്നായിരുന്നു. 1995-ൽ നെറ്റ് സ്കേപ്പ് ബ്രൗസർ പതിപ്പ് 2.0 ത്തിൽ ഉൾപ്പെടുത്തിയപ്പോൾ ഇതിന്റെ പേര് 'ജാവാസ്ക്രിപ്റ്റ്' എന്നാക്കിമാറ്റി. ആദ്യകാലങ്ങളിൽ, നെറ്റ്സ്കേപ്പ് ബ്രൗസർ മാത്രമേ ജാവാസ്ക്രിപ്റ്റിന് പിന്തുണ നൽകിയിരുന്നുള്ളൂ. എന്നാൽ ജാവാ സ്ക്രിപ്റ്റിന്റെ വിപുലമായ പ്രചാരം കാരണം, ഇന്റർനെറ്റ് എക്സ്പ്ലോറർ 1996-ൽ ജാവാസ്ക്രിപ്റ്റിന് പിന്തുണ നൽകി. ഇപ്പോൾ, ലോകത്തിലെ മിക്ക ബ്രൗസറുകളും ജാവാസ്ക്രിപ്റ്റിനെ പിന്തുണയ്ക്കുന്നു.

6.1 ജാവസ്ക്രിപ്റ്റ് ആരംഭിക്കാം (Getting started with JavaSript)

നാലാമത്തെ അധ്യായമായ വെബ് ടെക്നോളജിയിൽ ക്ലൈന്റ് സൈഡിലും സെർവർ സൈഡിലും ഉള്ള സ്ക്രിപ്റ്റിംഗ് ഭാഷകളുടെ ഉപയോഗം നാം ചർച്ചചെയ്തു. ക്ലൈന്റ് സൈഡിലെ ഡാറ്റയുടെ സാധുത ഉറപ്പുവരുത്തുന്നതിനായി ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ ഉപയോഗിക്കുന്നു. ഇത് സെർവറിലെ നെറ്റ് വർക്ക് ട്രാഫിക്കും ജോലി ഭാരവും കുറയ്ക്കുന്നു. സെർവർ സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷകൾ സെർവറിൽ പ്രവർത്തിക്കുന്നു. ഇതിലൂടെ നിർമിക്കപ്പെട്ട വെബ് പേജ് ക്ലൈന്റ് ബ്രൗസറിലേക്ക് തിരിച്ചയക്കുന്നു. ഡാറ്റാ ബേസിന്റെ രൂപത്തിൽ വലിയ അളവ് ഡാറ്റ സെർവറിൽ സംഭരിക്കുന്നു. അതിനാൽ സെർവർ സൈഡിലെ സ്ക്രിപ്റ്റിംഗ് ഭാഷ വ്യത്യസ്ത ആവശ്യങ്ങൾക്ക് വേണ്ടി ഈ ഡാറ്റബേസുമായി സംവദിക്കേണ്ടതായി വരും. എന്നാൽ ഒരു ക്ലൈൻറ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷ സെർവർ ഡാറ്റബേസുമായി നേരിട്ട് ബന്ധപ്പെടുന്നില്ല. അതുകൊണ്ടുതന്നെ ക്ലൈന്റ് സൈഡിലും സെർവർ സൈഡിലും ഉപയോഗിക്കുന്ന ഭാഷകളും കമാൻഡുകളും വ്യത്യസ്തങ്ങളാണ്.

ഈ അധ്യായത്തിൽ ക്ലൈന്റ് വശം സ്ക്രിപ്റ്റിംഗ് ഭാഷ എന്ന നിലയിൽ നമുക്ക് ജാവാ സ്ക്രിപ്റ്റിന്റെ അടിസ്ഥാന വിവരങ്ങൾ മനസിലാക്കാം. ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് HTML പേജിലെ വിവിധ ഭാഗങ്ങളിൽ പ്രോഗ്രാമുകൾ ഉൾപ്പെടുത്താൻ കഴിയും. ഇതിനായി <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നു.

<SCRIPT> Tag

ഒരു HTML പേജിൽ സ്ക്രിപ്റ്റിംഗ് കോഡ് ഉൾപ്പെടുത്താൻ <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നു. <SCRIPT> ടാഗിലെ 'language' ആട്രിബ്യൂട്ട് ഉപയോഗിക്കുന്നത് സ്ക്രിപ്റ്റിംഗ് ഭാഷയുടെ പേര് വ്യക്തമാക്കുന്നതിനാണ്. ഇവിടെ ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗ് ഭാഷയായി ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്നതിനുവേണ്ടി <SCRIPT> ടാഗിലെ 'language' ആട്രിബ്യൂട്ടിന്റെ വില 'JavaScript' എന്ന് 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">
```

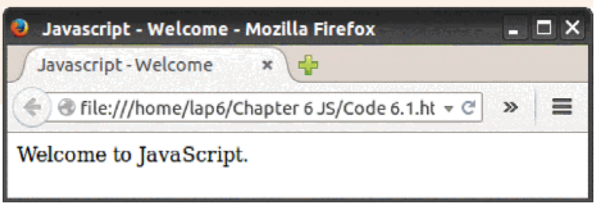
```
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
</SCRIPT>
```

ജാവാസ്ക്രിപ്റ്റിലെ ഐഡന്റിഫയറുകൾ കേസ് സെൻസിറ്റീവ് ആണ്. ജാവാസ്ക്രിപ്റ്റിൽ സാധാരണയായി ഐഡന്റിഫയറുകൾക്ക് കാമൽകേസ് (camelCase) പേരുകളാണ് ഉപയോഗിക്കുക. ഉദാഹരണത്തിന് firstName, checkData തുടങ്ങിയവ. ഒന്നിൽകൂടുതൽ വാക്കുകൾ ഒറ്റവാക്കായി ഉപയോഗിക്കുന്ന ഒരു നാമകരണ രീതിയാണ് കാമൽകേസ്. ഓരോ വാക്കിന്റെയും ആദ്യ അക്ഷരം വലിയ അക്ഷരം ഉപയോഗിക്കുന്നതിനെ 'അപ്പർകാമൽകേസ്' എന്ന് വിളിക്കുന്നു. ആദ്യത്തെ വാക്കൊഴികെ മറ്റെല്ലാ വാക്കിന്റെയും ആദ്യ അക്ഷരം വലിയ അക്ഷരം ഉപയോഗിക്കുന്നതിനെ 'ലോവർകാമൽകേസ്' എന്നു വിളിക്കുന്നു. കാമൽകേസ് രീതി വാക്കുകൾ വായിക്കുവാൻ എളുപ്പമുള്ളതാക്കുന്നു.

ഇനി നമുക്ക് താഴെകൊടുത്തിരിക്കുന്ന HTML ഫയൽ പരിഗണിക്കാം

ഉദാഹരണം 6.1 : ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് ഒരു വെബ് പേജ് നിർമ്മാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Welcome</TITLE> </HEAD>
<BODY>
   <SCRIPT Language= "JavaScript">
      document.write("Welcome to JavaScript.");
   </SCRIPT>
</BODY>
</HTML>
```



ചിത്രം 6.1: ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്ന വെബ് പേജ്

മുകളിലുള്ള കോഡ് ഏതെങ്കിലും ടെക്സ്റ്റ് എഡിറ്ററിൽ ടൈപ്പ് ചെയ്യാം. മുൻ അധ്യായങ്ങളിൽ HTML പേജുകൾ സൃഷ്ടിക്കുന്നതിന് നിങ്ങൾ ഉപയോഗിച്ചിരിക്കുന്ന ജീനി എഡിറ്ററിലും ഇത് ടൈപ്പ് ചെയ്യാവുന്നതാണ്. code6.html എന്ന പേരിൽ ഫയൽ സൂക്ഷിക്കുക. HTML പേജിൽ ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ചിട്ടുണ്ടെങ്കിലും അത് സേവ് ചെയ്തിരിക്കുന്നത്. html എക്സ്റ്റൻഷനിലാണ് എന്നത് പ്രത്യേകം ശ്രദ്ധിക്കേണ്ടതാണ്. ഇനി ഏതെങ്കിലും ഒരു ബ്രൗസറിൽ നമ്മൾ സേവ് ചെയ്ത HTML ഫയൽ തുറക്കുക. ചിത്രം 6.1 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഒരു വെബ് പേജ് ലഭിക്കും. 'document.write' എന്ന പ്രസ്താവന ചെറിയ അക്ഷരങ്ങളിലാണ് എഴുതിയിരിക്കുന്നത് എന്നതും ശ്രദ്ധിക്കുക, കാരണം ജാവാസ്ക്രിപ്റ്റ് കേസ് സെൻസിറ്റീവ് സ്ക്രിപ്റ്റിംഗ് ഭാഷയാണ്. സ്ക്രിപ്റ്റിലെ കീവേഡുകൾ എല്ലാം ചെറിയ അക്ഷരത്തിലാണ് എഴുതേണ്ടത്.

മുകളിൽ കൊടുത്തിരിക്കുന്ന HTML കോഡിൽ, document.write() എന്നത് ഒരു ജാവാസ്ക്രിപ്റ്റ് നിർദേശം ആണ്. അത് HTML പേജിന്റെ ബോഡിക്കകത്ത് ഒരു ടെക്സ്റ്റ് ഉൾ

ക്കൊള്ളിക്കുവാൻ ഉപയോഗിക്കുന്നു. അതായത്, മുകളിൽ നൽകിയിരിക്കുന്ന HTML കോഡ് നിർമിക്കുന്ന അതേ വെബ്പേജ് താഴെ നൽകിയിരിക്കുന്ന ജാവാ സ്ക്രിപ്റ്റ് ഇല്ലാത്ത HTML കോഡും നിർമ്മിക്കുന്നു.

ഉദാഹരണം 6.2 : HTML ഉപയോഗിച്ച് ഒരു വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Welcome</TITLE> </HEAD>
<BODY>
   Welcome to JavaScript.
</BODY>
</HTML>
```

മേൽ വിവരിച്ച ഉദാഹരണങ്ങൾ 6.1 ഉം 6.2 ഉം താരതമ്യം ചെയ്യാം. രണ്ടാമത്തെ HTML കോഡിൽ, "Welcome to JavaScript" എന്നത് ബോഡി ഭാഗത്തിൽ നേരിട്ട് എഴുതിയിരിക്കുന്നു. ആദ്യത്തെ കോഡിൽ "Welcome to JavaScript" എന്നത് ജാവാസ്ക്രിപ്റ്റ് രീതിയായ document.write( ) ഉപയോഗിച്ച് ബോഡി ഭാഗത്ത് ഉൾപ്പെടുത്തിയിരിക്കുന്നു.

യഥാർഥത്തിൽ 'document' എന്നത് വെബ് പേജിന്റെ ബോഡി വിഭാഗത്തെ പ്രതിനിധാനം ചെയ്യുന്നു. അതിനാൽ document.write() എന്ന ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷൻ വെബ് പേജിലെ ബോഡി ഭാഗത്തിൽ ഒരു ടെക്സ്റ്റ് ഉൾപ്പെടുത്തും. C++ പോലെ സ്ക്രിപ്റ്റിലെ ഓരോ പ്രസ്താവനയും ഒരു അർധവിരാമത്തിൽ (;) അവസാനിക്കുന്നു എന്നത് ശ്രദ്ധിക്കുക.

`<SCRIPT Language= "JavaScript">` എന്ന HTML നിർദേശം സൂചിപ്പിക്കുന്നത് തുടർന്നുവരുന്നത് ജാവാസ്ക്രിപ്റ്റ് കോഡ് ആണെന്നുള്ളതാണ്. ഇനി നമുക്ക് ജാവാ സ്ക്രിപ്റ്റിനെ ബ്രൗസർ എങ്ങനെ കൈകാര്യം ചെയ്യുന്നുവെന്ന് കാണാം. എല്ലാ ബ്രൗസറിലും ജാവാസ്ക്രിപ്റ്റ് എൻജിൻ ഉണ്ട്. പ്രവർത്തിപ്പിക്കുന്ന സമയത്ത് സ്ക്രിപ്റ്റ് കോഡിനെ ജാവാസ്ക്രിപ്റ്റ് എൻജിൻ വ്യാഖ്യാനിക്കും. ജാവാസ്ക്രിപ്റ്റ് കോഡ് നിർവഹണത്തിനുള്ള വിർച്യുൽ മെഷിനാണ് (virtual machine) ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിൻ. ബ്രൗസർ, ജാവാസ്ക്രിപ്റ്റ് കോഡ് കാണുമ്പോൾ അതിനെ സ്ക്രിപ്റ്റ് എൻജിനിലൂടെ കടത്തിവിടുന്നു. ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിൻ കോഡിനെ പ്രവർത്തിപ്പിക്കുന്നു. ഒരു HTML പേജിൽ ജാവാ സ്ക്രിപ്റ്റ് കോഡ് ഇല്ലെങ്കിൽ ബൗസറിന് ഒറ്റയ്ക്ക് തന്നെ HTML പേജ് പ്രദർശിപ്പിക്കാൻ കഴിയും. ജാവാസ്ക്രിപ്റ്റ് കോഡ് ഉണ്ടെങ്കിൽ ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിന്റെ സഹായത്തോടുകൂടി മാത്രമേ HTML പേജ് പ്രദർശിപ്പിക്കുവാൻ കഴിയൂ. ഇപ്രകാരം ജാവാസ്ക്രിപ്റ്റ് ഇല്ലാത്ത HTML ഫയലുകൾ എപ്പോഴും ജാവാസ്ക്രിപ്റ്റ് ഉള്ളവയെക്കാൾ വേഗത്തിൽ പ്രദർശിപ്പിക്കാൻ കഴിയും.

ഉദാഹരണം 6.3 ൽ ജാവാസ്ക്രിപ്റ്റ് കോഡും HTML ടാഗുകളും ഇടകലർത്തി നൽകിയിരിക്കുന്നു. ഇതിന്റെ ഔട്ട്പുട്ട് വെബ്പേജ് ചിത്രം 6.2 ൽ നൽകിയിരിക്കുന്നു.

ഉദാഹരണം 6.3: ഹെഡ്ഡിങ് ടാഗുകൾ ഉൾക്കൊള്ളുന്ന ഒരു വെബ് പേജ് നിർമ്മാണം

```
<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>
```

ചിത്രം 6.2: ഹെഡ്ഡിംഗ് ടാഗുകൾ അടങ്ങുന്ന വെബ് പേജ്

മുകളിൽ നൽകിയിരിക്കുന്ന HTML ടാഗുകൾക്കിടയിൽ ഒന്നിലധികം തവണ സ്ക്രിപ്റ്റിംഗ് കോഡുകൾ ഉപയോഗിച്ചിട്ടുണ്ട്. അതുപോലെ നിങ്ങൾക്ക് HTML ടാഗുകൾ ക്കിടയിൽ എത്ര തവണ വേണമെങ്കിലും സ്ക്രിപ്റ്റ് കോഡുകൾ ഉപയോഗിക്കാം. സ്ക്രിപ്റ്റ് കോഡുകൾ ഉപയോഗിക്കുന്നത് എവിടെയായിരുന്നാലും <SCRIPT>, </SCRIPT> ടാഗുകൾക്കിടയിൽ അവ എഴുതിയിരിക്കണം.

ഒരു വെബ് പേജിൽ ജാവാസ്ക്രിപ്റ്റിന്റെ പ്രവർത്തനം അനുവദിക്കുന്നതിനോ നിരാകരിക്കുന്നതിനോ ഉള്ള സൗകര്യം എല്ലാ വെബ് ബ്രൗസറുകളും ഉപയോക്താക്കൾക്ക് നൽകുന്നുണ്ട്. ജാവാസ്ക്രിപ്റ്റ് നിരാകരിക്കുന്നതിലൂടെ, ആ ബ്രൗസറിലെ ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിന്റെ പ്രവർത്തനമാണ് നിരാകരിക്കുന്നത്. മോസില്ലയിൽ, Tools -> Options -> Content -> Enabled Java Script തിരഞ്ഞെടുത്ത് ജാവാസ്ക്രിപ്റ്റ് പ്രവർത്തനക്ഷമമാക്കുകയോ പ്രവർത്തനക്ഷമമല്ലാതാക്കുകയോ ചെയ്യാം. Google Chrome Â, Tools -> Settings -> Show Advanced Settings -> Content Settings -> Do not allow any site to run JavaScript രീതി ഉപയോഗിക്കാം. ഒരു വെബ് ബ്രൗസറിൽ ജാവാസ്ക്രിപ്റ്റ് പ്രവർത്തനക്ഷമമല്ലാതാക്കിയിട്ടുണ്ടെങ്കിൽ ആ ബ്രൗസറിൽ സ്ക്രിപ്റ്റ് പ്രവർത്തിക്കുകയില്ല അതായത്, ബ്രൗസർ <SCRIPT> ... </ SCRIPT> ടാഗുകൾക്കിടയിലുള്ള ഉള്ളടക്കം അവഗണിക്കും. ഒരു ബ്രൗസറിന്റെ പ്രവർത്തനം പ്രധാനമായും അതിന്റെ സ്ക്രിപ്റ്റ് എഞ്ചിന്റെ പ്രകടനത്തെ ആശ്രയിച്ചിരിക്കുന്നു. ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കാത്ത ഒരു ഡൈനാമിക് വെബ് പേജ് കാണാൻ പ്രയാസമാണ്. മികച്ചതും, വേഗതയേറിയതും ശക്തവുമായ ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിനുകളുടെ നിർമാണത്തിനായി എല്ലാ ബ്രൗസർ നിർമ്മാതാക്കളും പരസ്പരം മത്സരിക്കുന്നു.

നിങ്ങളുടെ പുരോഗതി അറിയാം

1. <SCRIPT> ടാഗിന്റെ ഒരു ആട്രിബ്യൂട്ടിന്റെ പേര് എഴുതുക.
2. ജാവാസ്ക്രിപ്റ്റിൽ ഒരു HTML പേജിന്റെ ബോഡി ഭാഗത്തിൽ ഒരു ടെക്സ്റ്റ് പ്രിന്റ് ചെയ്യാൻ ഉപയോഗിക്കുന്ന ഫങ്ഷൻ __________ ആണ്
3. ജാവാസ്ക്രിപ്റ്റ് കോഡ് ആണ് എഴുതിയിരിക്കുന്നത് എന്ന് വ്യക്തമാക്കുന്നത് <SCRIPT> ടാഗിന്റെ language ആട്രിബ്യൂട്ടിന് നൽകിയ __________ എന്ന മൂല്യമാണ്.
4. ജാവാസ്ക്രിപ്റ്റ് എഞ്ചിന്റെ ഉപയോഗം എന്താണ്?
5. താഴെ പറയുന്ന പ്രസ്താവനകൾ ശരിയോ തെറ്റോ എന്ന് എഴുതുക.

a. ജാവ സ്ക്രിപ്റ്റ് മാത്രമാണ് ക്ലൈന്റ് വശം സ്ക്രിപ്റ്റിംഗ് ഭാഷ.

b. ഒരു HTML പേജിൽ ക്ലയന്റ് വശം സ്ക്രിപ്റ്റിംഗ് ഭാഷ ഉൾപ്പെടുത്താൻ <SCRIPT> ടാഗാണ് ഉപയോഗിക്കുന്നത്.

c. ഒരു HTML ഫയലിൽ ഒരു <SCRIPT> ടാഗ് മാത്രമേ അടങ്ങിയിരിക്കാവൂ.

d. നമുക്ക് HTML കോഡിൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് ഇടകലർ ത്തുവാൻ കഴിയും.

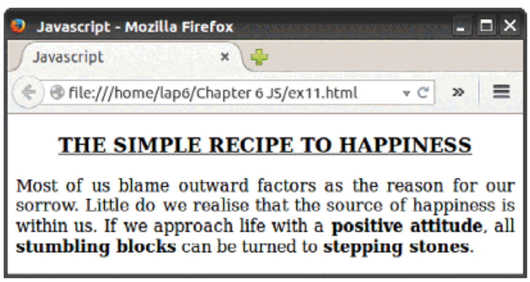
e. ജാവാസ്ക്രിപ്റ്റ് കോഡ് എല്ലായ്പ്പോഴും <SCRIPT>, </ SCRIPT> എന്നതിന് അകത്തതായിരിക്കണം.

f. ഓരോ ജാവാസ്ക്രിപ്റ്റ് പ്രസ്താവനകളും അർദ്ധവിരാമം(;) കൊണ്ട് അവസാനിക്കുന്നു.

നമുക്കു ചെയ്യാം

പേജിന്റെ ബോഡി വിഭാഗത്തിനുള്ളിൽ മാത്രം ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് താഴെ പറയുന്ന വെബ് പേജ് സൃഷ്ടിക്കാൻ ഒരു HTML കോഡ് എഴുതുക. അതായത്, ബോഡി വിഭാഗം താഴെ കാണിച്ചിരിക്കുന്നത് പോലെ ആയിരിക്കണം.

```
<BODY>
<SCRIPT Language= "JavaScript">
. . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . .
</SCRIPT>
</BODY>
```



6.2 ജാവാസ്ക്രിപ്റ്റിലെ ഫങ്ഷനുകളുടെ നിർമാണം (Creating functions in JavaScript)

C++ ലെ ഫങ്ഷനുകളെക്കുറിച്ച് നാം നേരത്തെ മനസ്സിലാക്കിയിട്ടുണ്ട്. ജാവാസ്ക്രിപ്റ്റിലും പ്രവർത്തനങ്ങൾ നിർവചിക്കപ്പെട്ടിട്ടുള്ളതും വിളിക്കപ്പെടുന്നതും C++ ലെ അതേ രീതിയിൽ തന്നെയാണ്. ഒരു പേരുള്ള നിർദ്ദേശങ്ങളുടെ കൂട്ടമാണ് ഫങ്ഷൻ. വ്യത്യസ്ത

ആവശ്യങ്ങൾക്ക് ഉപയോഗിക്കാൻ കഴിയുന്ന ധാരാളം അന്തർനിർമിത ഫങ്ഷനുകൾ ജാവാസ്ക്രിപ്റ്റിൽ ഉണ്ട്. ഇവയിൽ ചില ഫങ്ഷനുകൾ ഈ അധ്യായത്തിൽ പിന്നീട് ചർച്ചചെയ്യും. ഈ അന്തർനിർമിത ഫങ്ഷനുകൾക്ക് പുറമെ നമുക്കും ഫങ്ഷനുകൾ നിർവ്വചിക്കാം. വെബ് പ്രോഗ്രാമിന്റെ ഒരു ഭാഗം ഒന്നിൽ കൂടുതൽ തവണ പ്രവർത്തിപ്പിക്കാം എന്നതാണ് ഒരു ഫങ്ഷൻ ഉപയോഗിക്കുന്നതുകൊണ്ടുള്ള ഏറ്റവും വലിയ മെച്ചം. ഈ ഭാഗം ഒരു പ്രാവശ്യം ഫങ്ഷനായി എഴുതിയാൽ മതി. താഴെ പറയുന്ന കോഡ് നോക്കുക.

```
function print()
{
    document.write("Welcome to JavaScript.");
}
```

ഇവിടെ, ഒരു ഫങ്ഷൻ നിർവചിക്കുവാനുള്ള കീവേഡ് 'function' ഉം ഫങ്ഷന്റെ പേര് print ഉം ആണ്. ഫങ്ഷന്റെ പേര് സാധുതയുള്ള ഏതൊരു ഐഡന്റിഫയറും ആകാം. C++ ൽ ഐഡന്റിഫയറുകൾ നാമകരണം ചെയ്യുന്ന അതേ നിയമങ്ങൾ ജാവാസ്ക്രിപ്റ്റിലും ഉപയോഗിക്കുന്നു. ഇവിടെ print() ഫങ്ഷനിൽ ഒരു പ്രസ്താവന മാത്രമേ ഉള്ളൂ, ആവശ്യമെങ്കിൽ ഒന്നിലധികം പ്രസ്താവനകൾ ഒരു ഫങ്ഷനിൽ ഉൾപ്പെടുത്താവുന്നതാണ്. ഒരു ഫങ്ഷൻ നിർവചിക്കുന്നത് കൊണ്ട് അത് സ്വയം പ്രവർത്തിക്കണം എന്നില്ല, അതിനെ വിളിക്കുമ്പോൾ മാത്രമേ അത് പ്രവർത്തിക്കൂ. ഇതിനർത്ഥം ഒരു വെബ് പേജിലെ ഒരു ഫങ്ഷൻ വിളിക്കാതിരിക്കുകയാണെങ്കിൽ ആ ഫങ്ഷൻ ഒരിക്കലും പ്രവർത്തിക്കുകയില്ല. ഫങ്ഷന്റെ പേര് ഉപയോഗിച്ച് താഴെ പറയുന്ന ഫങ്ഷനെ വിളിക്കാം.

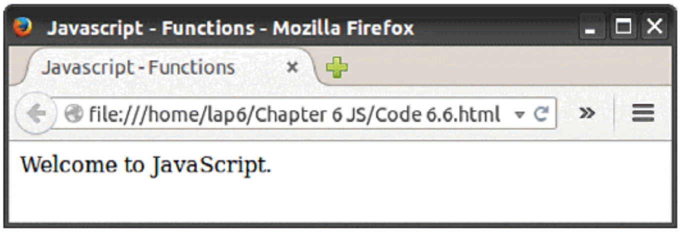
```
print();
```

ഫങ്ഷന്റെ പേരിന് ശേഷമുള്ള അർധവിരാമം ശ്രദ്ധിക്കുക. നമുക്ക് മുകളിലെ print() ഫങ്ഷൻ ഒരു HTML പേജിൽ ഉപയോഗിക്കുന്നത് എങ്ങനെ എന്ന് നോക്കാം.

ഉദാഹരണം 6.4: print() ഫംഗ്ഷൻ അടങ്ങിയ വെബ് പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - Functions</TITLE>
    <SCRIPT Language= "JavaScript">
        function print()
        {
            document.write("Welcome to JavaScript.");
        }
    </SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

HTML പേജിന്റെ ബോഡി ഭാഗത്തിൽ ഒന്നും എഴുതിയിട്ടില്ലെന്നത് ശ്രദ്ധിക്കുക. ഈ കോഡ് ബ്രൗസർ വിൻഡോയിലെ 'Welcome to JavaScript' പ്രദർശിപ്പിക്കില്ല. കാരണം ഫങ്ഷൻ നിർവചിച്ചിട്ടുണ്ടെങ്കിലും, അത് പേജിലെ ഏതെങ്കിലും സ്ഥലത്തുനിന്ന് വിളിച്ചിട്ടില്ല. അതിനാൽ, ഫങ്ഷൻ ഒരിക്കലും പ്രവർത്തിക്കുകയില്ല, സ്ക്രീനിൽ ഒന്നും പ്രദർശിപ്പിക്കുകയുമില്ല. ബ്രൗസർ ജാലകത്തിൽ "Welcome to JavaScript" എന്ന് ഔട്ട്പുട്ട് ലഭിക്കുന്നതിന് HTML പേജിന്റെ ബോഡി ഭാഗം താഴെ കാണുന്ന രീതിയിൽ പരിഷ്കരിക്കേണ്ടതുണ്ട്.



ചിത്രം 6.3: ഫങ്ഷൻ ഉപയോഗിച്ചിരിക്കുന്ന വെബ് പേജ്

```
<BODY>
<SCRIPT Language= "JavaScript">
    print();
</SCRIPT>
</BODY>
```

ഇനി നമുക്ക് ജാവാ സ്ക്രിപ്റ്റ് ഫങ്ഷന്റെ വാക്യഘടന പരിചയപ്പെടാം

```
function function_name()
{
    statements;
}
```

ഇവിടെ function_name () എന്ന വരിയിൽ ഫങ്ഷൻ ഹെഡറും, ബ്രാക്കറ്റിലുള്ള കോഡ് ഫങ്ഷൻ ബോഡി എന്നു വിളിക്കുന്നു. C++ ൽ ഫങ്ഷന് റിട്ടേൺ ടൈപ്പ് ഉണ്ടെങ്കിലും ജാവാസ്ക്രിപ്റ്റിൽ ഫങ്ഷന് റിട്ടേൺ ടൈപ്പില്ല. ജാവാസ്ക്രിപ്റ്റിലും C++ ലേതുപോലെ നിങ്ങൾക്ക് ഫങ്ഷനിൽ നിന്ന് ചില മൂല്യങ്ങൾ തിരികെ നൽകാം. ഈ അധ്യായത്തിൽ ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗത്തെക്കുറിച്ചുള്ള അടിസ്ഥാന ആശയങ്ങൾ മാത്രം നൽകാൻ ഉദ്ദേശിച്ചിട്ടുള്ളതിനാൽ, ഇത്തരം കാര്യങ്ങൾ ഇവിടെ ചർച്ച ചെയ്യുന്നില്ല. മറ്റൊരു വ്യത്യാസം C++ൽ ഒരു ഫങ്ഷൻ നിർവചിക്കാൻ function എന്ന കീവേഡ് ഉപയോഗിക്കില്ല, എന്നാൽ ജാവാസ്ക്രിപ്റ്റിൽ function എന്ന കീവേഡ് ഉപയോഗിക്കുന്നു. HTML പേജിന്റെ ഹെഡ് ഭാഗത്തിനുള്ളിൽ ഫങ്ഷൻ നിർവചിക്കപ്പെട്ടിരിക്കുന്നതായി നിങ്ങൾ ശ്രദ്ധിച്ചിരിക്കാം. ഹെഡ് ഭാഗത്തിനുള്ളിൽ തന്നെ ഫങ്ഷൻ നിർവചിക്കണം എന്ന് നിർബന്ധമില്ല, ബോഡി ഭാഗത്തും ഫങ്ഷൻ നിർവചിക്കാം.

```
<BODY>
    <SCRIPT Language= "JavaScript">
        function print()
```

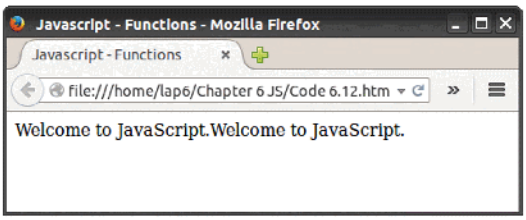
```
        {
            document.write("Welcome to JavaScript.");
        }
        print();
    </SCRIPT>
</BODY>
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന കോഡ് ഇതേ ഔട്ട്പുട്ട് നൽകുന്നത് ചിത്രം 6.3 ൽ കൊടുത്തിരിക്കുന്നു.

ഫങ്ഷൻ ബോഡി ഭാഗത്തിനകത്താണ് നിർവചിക്കപ്പെട്ടിരിക്കുന്നതെങ്കിൽപ്പോലും അതിന്റെ പ്രവർത്തിപ്പിക്കുന്നതിനുവേണ്ടി അതിനെ വിളിക്കണം എന്നത് ശ്രദ്ധിക്കുക. ഉദാഹരണത്തിന് താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് സ്ക്രീനിൽ ഒന്നും പ്രദർശിപ്പിക്കുന്നില്ല.

```
<BODY>
<SCRIPT Language= "JavaScript">
function print()
{
document.write("Welcome to JavaScript.");
}
</SCRIPT>
</BODY>
```

ഒരു HTML പേജിൽ എവിടെയും ഒരു ഫങ്ഷൻ നിർവചിക്കപ്പെടാമെങ്കിലും, ഹെഡ് ഭാഗത്തിൽ ഫങ്ഷൻ നിർവചനം ഉൾപ്പെടുത്തുന്നതാണ് നല്ലത്.



ചിത്രം 6.4: ബ്രേക്ക് ഇല്ലാതെ രണ്ട് പ്രിന്റ് ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്ന വെബ് പേജ്

`print()` ഫങ്ഷൻ രണ്ടു തവണ വിളിച്ചിരിക്കുന്ന താഴെ കൊടുത്തിരിക്കുന്ന കോഡ് പരിഗണിക്കുക, ഇത് ചിത്രം 6.4ൽ കാണുന്ന രീതിയിലുള്ള ഔട്ട്പുട്ട് നൽകുന്നു.

```
<BODY>
    <SCRIPT Language= "JavaScript">
        print();
        print();
    </SCRIPT>
</BODY>
```

ഔട്ട്പുട്ട് ചിത്രം 6.4 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ആയിരിക്കും. എന്തുകൊണ്ട്?

വാസ്തവത്തിൽ, നമ്മൾ രണ്ടുതവണ ഫങ്ഷനെ വിളിക്കുമ്പോൾ, അത് 'Welcome to JavaScript' എന്ന് രണ്ടുതവണ ബോഡി ഭാഗത്ത് എഴുതും. അതിനാൽ മുകളിലുള്ള കോഡിന്റെ ബോഡി ഭാഗം താഴെ എഴുതിയിരിക്കുന്നതിനു തുല്യമായിരിക്കും.

```
<BODY>
   Welcome to JavaScript.Welcome to JavaScript.
</BODY>
```

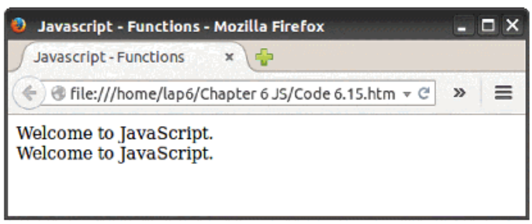
സന്ദേശം രണ്ട് വ്യത്യസ്ത വരികളിൽ പ്രദർശിപ്പിക്കുന്നതിന് നിർബന്ധമായും
 ടാഗ് ഉപയോഗിക്കണം.

```
<BODY>
   Welcome to JavaScript.<BR>Welcome to JavaScript.
</BODY>
```

ജാവാസ്ക്രിപ്റ്റ് 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 ഔട്ട്പുട്ട് ചിത്രം 6.5 ൽ കൊടുത്തിരിക്കുന്നു.



ചിത്രം 6.5: രണ്ട് പ്രിന്റ് ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്ന വെബ് പേജ്

നിങ്ങളുടെ പുരോഗതി അറിയാം

1. താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന ശരിയോ തെറ്റോ എന്ന് പറയുക.
   a. ഒരു വെബ് പേജ് ബ്രൗസറിൽ തുറക്കുമ്പോൾ ഫങ്ഷൻ സ്വമേധയാ പ്രവർത്തിക്കും.
   b. ഒരു ഫംഗ്ഷൻ സാധാരണയായി HTML പേജിന്റെ ഹെഡ്ഡ് ഭാഗത്തിൽ ഉൾപ്പെടുത്തുന്നു.
   c. ഒരു ഫംഗ്ഷൻ എത്ര തവണ വേണമെങ്കിലും വിളിക്കാം.
   d. ബോഡി ഭാഗത്തിനകത്ത് ഒരു ഫങ്ഷൻ നിർവചിച്ചിട്ടുണ്ടെങ്കിലും അത് വിളിച്ചിട്ടില്ലെങ്കിൽ പ്രവർത്തിക്കില്ല.
2. ജാവാസ്ക്രിപ്റ്റിൽ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന്റെ ഗുണദോഷങ്ങൾ എഴുതുക.

നമുക്കു ചെയ്യാം

1. താഴെക്കൊടുത്തിരിക്കുന്ന രണ്ട് HTML കോഡുകൾ (കോഡ് A ഉം കോഡ് B ഉം) പരിഗണിച്ച് ഔട്ട്പുട്ട് 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. ചിത്രം 6.6 ൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഔട്ട്പുട്ട് ലഭിക്കുന്നതിന് ഒരു HTML കോഡ് താഴെ കൊടുക്കുന്നു. കോഡിൽ `startGreen()`, `stopGreen()`, `startRed()` and `stopRed()` എന്നിങ്ങനെ നാലു ഫങ്ഷനുകൾ ഉണ്ട്. ഈ ഫങ്ഷനുകൾ HTML പേജിന്റെ ബോഡി ഭാഗത്തിൽ വ്യത്യസ്ത സ്ഥലങ്ങളിൽ നിന്നും വിളിക്കുന്നു. `stopGreen()` എന്ന ഫങ്ഷൻ ഒഴികെയുള്ള എല്ലാ ഫങ്ഷൻ നിർവചനങ്ങളുടെയും ബോഡി ശൂന്യമാണ്. നിങ്ങൾക്ക് ചിത്രത്തിൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഔട്ട്പുട്ട് ലഭിക്കുന്നതിന് മറ്റു ഫങ്ഷനുകളുടെ നിർവചനം പൂർത്തിയാക്കുക.

```
<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();
</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 ഡാറ്റ ഇനങ്ങൾ (Data types in JavaScript)

മെമ്മറി ഫലപ്രദമായി ഉപയോഗപ്പെടുത്തുന്നതിന് പ്രോഗ്രാമിങ് ഭാഷകൾ ഡാറ്റയെ വ്യത്യസ്ത വിഭാഗങ്ങളായി തരം തിരിച്ചിരിക്കുന്നു. C++ ലെ അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ int, char, float, double, void തുടങ്ങിയവയാണെന്ന് നമ്മൾ മനസ്സിലാക്കിയിട്ടുണ്ട്. ഈ ഡാറ്റ ഇനങ്ങളോടൊപ്പം, ടൈപ്പ് മോഡിഫയറുകളും C++ ൽ ഉണ്ടായിരുന്നു. അടിസ്ഥാന ഡാറ്റ ഇനങ്ങളുടെ എണ്ണം 3 ആയി പരിമിതപ്പെടുത്തിയിരിക്കുന്നതിലൂടെ ജാവാസ്ക്രിപ്റ്റ് ഇതിന്റെ സങ്കീർണത കുറയ്ക്കുന്നു. താഴെ പറയുന്നവയാണ് ജാവാസ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന മൂന്ന് അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ.

നമ്പർ (Number)

എല്ലാ നമ്പറുകളും ഈ വിഭാഗത്തിൽ വരുന്നു. പോസിറ്റീവ് സംഖ്യകളും, നെഗറ്റീവ് സംഖ്യകളും, എല്ലാ ഫ്ളോട്ട് വിലകളും (ഭിന്നസംഖ്യകൾ), പൂർണ സംഖ്യകളും നമ്പർ (number) ഡാറ്റ ഇനം ആയി പരിഗണിക്കപ്പെടും. അതിനാൽ, 27, -300, 1.89, -0.0082 എന്നിവ ജാവാസ്ക്രിപ്റ്റിലെ നമ്പർ ഡാറ്റ ഇനത്തിനുള്ള ഉദാഹരണങ്ങളാണ്.

സ്ട്രിംഗ് (String)

ഇരട്ട ഉദ്ധരണികൾക്കുള്ളിൽ ഉള്ള പ്രതീകങ്ങൾ, അക്കങ്ങൾ അല്ലെങ്കിൽ മറ്റേതെങ്കിലും ചിഹ്നങ്ങളുടെ കൂട്ടത്തെ ജാവാസ്ക്രിപ്റ്റിൽ ഒരു സ്ട്രിംഗായി പരിഗണിക്കും. ഉദാഹരണം "Kerala", "Welcome", "SCHOOL", "1234", "Mark20", "abc$" and "sanil@123".

ബൂളിയൻ (Boolean)

ഈ വിഭാഗത്തിൽ രണ്ട് മൂല്യങ്ങൾ മാത്രമേ ഉള്ളൂ, അവ true, false എന്നിവയാണ്. ഈ മൂല്യങ്ങൾ ഇരട്ട ഉദ്ധരണികളിലല്ല. ഇരട്ട ഉദ്ധരണികൾക്കുള്ളിൽ ആണെങ്കിൽ അവ സ്ട്രിംഗ് ആയിരിക്കും. ജാവാസ്ക്രിപ്റ്റ് കേസ് സെൻസിറ്റീവ് ആയതുകൊണ്ട് ബൂളിയൻ വിലകളെ പ്രതിനിധാനം ചെയ്യുന്നതിന് TRUE, FALSE എന്നിങ്ങനെ ഉപയോഗിക്കുവാൻ പാടില്ല.

6.4 ജാവാ സ്ക്രിപ്റ്റിലെ വേരിയബിളുകൾ (Variables in JavaSript)

വിലകൾ സംഭരിക്കുന്നതിന് വേരിയബിളുകൾ ഉപയോഗിക്കാമെന്ന് നമുക്കറിയാം. ജാവാ സ്ക്രിപ്റ്റിൽ **var** എന്ന കീവേഡ് ഉപയോഗിച്ചാണ് എല്ലാത്തരം വേരിയബിളുകളും പ്രഖ്യാപിക്കുന്നത്.

```
var  x;
```

ഇവിടെ 'x' എന്നത് വേരിയബിളിന്റെ പേരാണ്, ഒരു വേരിയബിളിന് C++ ലേതുപോലെ ഏതു പേരും ഉപയോഗിക്കാം. അത് ഒരു സാധുതയുള്ള ഐഡന്റിഫയർ ആയിരിക്കണമെന്നുമാത്രം. C++ ൽ വ്യത്യസ്ത തരം വേരിയബിളുകൾ പ്രഖ്യാപിക്കാൻ കീവേഡുകൾ `int`, `float`, `char` തുടങ്ങിയവ ഉപയോഗിച്ചു. പക്ഷേ, ജാവാസ്ക്രിപ്റ്റിൽ **var** എന്ന കീവേഡ്' മാത്രമാണ് എല്ലാത്തരം വേരിയബിളുകൾ പ്രഖ്യാപിക്കുന്നതിനും ഉപയോഗിക്കുന്നത്. ജാവാസ്ക്രിപ്റ്റിൽ വേരിയബിളിന് വില നൽകുമ്പോൾ മാത്രം നിർവചനം പൂർത്തിയാകുന്നു. ഒരു വേരിയബിളിന് മൂല്യം നൽകുമ്പോൾ മാത്രമാണ് വേരിയബിളിന്റെ തരം ജാവാസ്ക്രിപ്റ്റ് മനസ്സിലാക്കുന്നത്.

താഴെകൊടുത്തിരിക്കുന്ന പ്രസ്താവന പരിഗണിക്കുക.

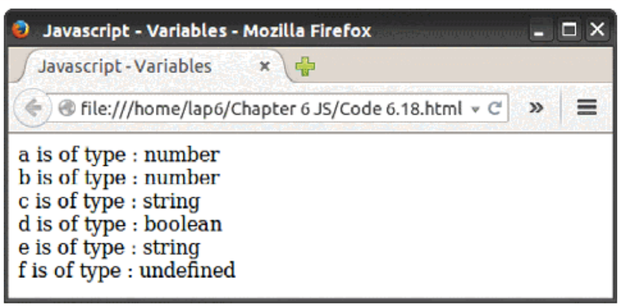
```
var x, y;
x = 25;
y = "INDIA";
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന ഉദാഹരണത്തിൽ, x എന്നത് നമ്പർ തരത്തിലും y സ്ട്രിംഗ് തരത്തിലുംപെട്ട വേരിയബിളുകൾ ആണ്. എന്നാൽ കീവേഡ് number, string എന്നൊന്നും ഇവിടെ ഉപയോഗിച്ചിട്ടില്ല. താഴെ കൊടുത്തിരിക്കുന്ന ഉദാഹരണത്തിൽ ജാവാ സ്ക്രിപ്റ്റ് എങ്ങനെയാണ് വേരിയബിളുകൾ നിർവചിക്കുന്നത് എന്ന് വിവരിക്കുന്നു.

ഉദാഹരണം 6.5 : വേരിയബിളുകളുടെ 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>
```

ഇവിടെ, `typeof()` എന്നൊരു ഫങ്ഷൻ പുതുതായി ഉപയോഗിച്ചിരിക്കുന്നു. പേര് സൂചിപ്പിക്കുന്നതുപോലെ ഒരു വേരിയബിളിന്റെ തരം കണ്ടുപിടിക്കുവാൻ ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഈ ഫങ്ഷനെ പിന്നീട് വിശദമായി പഠിക്കാം. മുകളിലുള്ള ഉദാഹരണത്തിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.6 ൽ കൊടുത്തിരിക്കുന്നു.



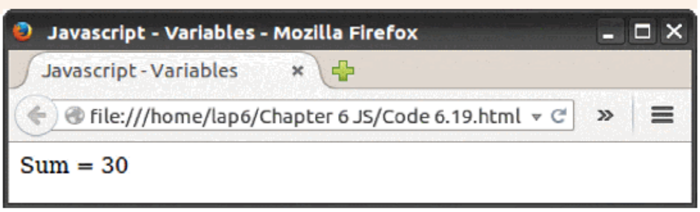
ചിത്രം 6.6: വേരിയബിളുകളുടെ തരം ഉപയോഗം പ്രദർശിപ്പിക്കുന്ന വെബ് പേജ്.

വേരിയബിൾ f പ്രഖ്യാപിച്ചിട്ടുണ്ടെങ്കിലും അതിന് ഒരു വില നൽകിയിട്ടില്ല. അതുകൊണ്ട് സ്ക്രിപ്റ്റ് എഞ്ചിന് അതിന്റെ തരം മനസ്സിലാക്കുവാനോ പ്രദർശിപ്പിക്കുവാനോ സാധിക്കു

ന്നില്ല. നിർവചിക്കാത്ത വേരിയബിളിനെ പ്രതിനിധാനം ചെയ്യുന്നതിനായി സ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന ഒരു പ്രത്യേക ഡാറ്റ ഇനമാണ് undefined. ഒരൊറ്റ var കീവേഡ് ഉപയോഗിച്ച് എത്ര വേരിയബിളുകൾ വേണെമെങ്കിലും പ്രഖ്യാപിക്കാനാവും (ഡിക്ലയർ ചെയ്യാനാവും). വേരിയബിളുകൾ കോമ (,) ഉപയോഗിച്ച് വേർതിരിക്കേണ്ടതാണ്. വേരിയബിളുകൾ ഉപയോഗിക്കുന്ന ഒരു ജാവാസ്ക്രിപ്റ്റിന്റെ പ്രവർത്തനം നമുക്ക് പരിഗണിക്കാം.

ഉദാഹരണം 6.6: രണ്ട് സംഖ്യകളുടെ തുക കണ്ടു പിടിക്കുന്നതിനുള്ള വെബ് പേജ് നിർമാണം

```
<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>
```



ചിത്രം 6.7: രണ്ടു സംഖ്യകളുടെ തുക കണ്ടുപിടിക്കുന്നതിനുള്ള വെബ്പേജ്

മുകളിൽ പറഞ്ഞ ഉദാഹരണത്തിൽ, ബോഡി ഭാഗത്തിൽ ഒരു ജാവാസ്ക്രിപ്റ്റ് പ്രസ്താവന മാത്രമേ അടങ്ങിയിട്ടുള്ളൂ, അത് add() എന്ന ഫങ്ഷനെ വിളിക്കുന്നു. ഫങ്ഷൻ പ്രവർത്തനത്തിന്റെ ഫലം ചിത്രം 6.7 ൽ കാണിച്ചിരിയ്ക്കുന്നു.

6.5 ജാവാസ്ക്രിപ്റ്റിലെ ഓപ്പറേറ്ററുകൾ (Operators in JavaScript)

ജാവാ സ്ക്രിപ്റ്റിലെ മിക്കവാറും എല്ലാ ഓപ്പറേറ്ററുകളും C++ ലേതിന് സമാനമാണ്. അവ ഏതെല്ലാമാണെന്ന് നോക്കാം

6.5.1 അരിത്തമാറ്റിക് ഓപ്പറേറ്ററുകൾ (Arithmetic Operatiors)

ജാവാസ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന അരിത്തമാറ്റിക് ഓപ്പറേറ്ററുകൾ പട്ടിക 6.1 കാണിക്കുന്നു.

ഓപ്പറേഷൻ	വിവരണം	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
%	മോഡുലസ് ഹരണത്തിലെ ശിഷ്ടം	x = y % 2	15	1
++	വർദ്ധനവ് Increment	x = ++y x = y++	15 15	16 15
--	കുറവ് Decrement	x = --y x = y--	15 15	14 15

പട്ടിക 6.1: അരിത്തമാറ്റിക് ഓപ്പറേറ്ററുകൾ

മുകളിലുള്ള പട്ടികയിലെ എല്ലാ അരിത്തമാറ്റിക് ഓപ്പറേററുകളും C++ ൽ പ്രവർത്തിക്കുന്നപോലെതന്നെ പ്രവർത്തിക്കുന്നുവെന്ന് കാണാം.

6.5.2 വിലനൽകൽ ഓപ്പറേറ്ററുകൾ (Assignment Operators)

ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ വിലനൽകൽ ഓപ്പറേറ്ററുകളുടെ ഉപയോഗരീതി പട്ടിക 6.2ൽ കാണിച്ചിരിക്കുന്നു.

ഓപ്പറേറ്റർ	വിവരണം	ഉദാഹരണം	a യുടെ വില	b യുടെ വില	ഉത്തരം (a)
=	വില നൽകുന്നു	a = b	10	3	3
+=	സങ്കലനത്തിനുശേഷം വില നൽകുന്നു	a+=b	10	3	13
-=	വ്യവകലനത്തിനുശേഷം വില നൽകുന്നു	a-=b	10	3	7
*=	ഗുണനത്തിനുശേഷം വില നൽകുന്നു	a*=b	10	3	30
/=	ഹരണത്തിനുശേഷം വില നൽകുന്നു	a/=b	10	3	3.33
%=	മോഡുലസ് പ്രവർത്തനത്തിനുശേഷം വില നൽകുന്നു	a%=b	10	3	1

പട്ടിക 6.2: ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ വിലനൽകൽ ഓപ്പറേറ്ററുകൾ

മുകളിലുള്ള 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)

ജാവാസ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന വിവിധ തരം താരതമ്യ ഓപ്പറേറ്ററുകൾ പട്ടിക 6.3 ൽ കാണിച്ചിരിക്കുന്നു.

ഓപ്പറേറ്റർ	വിവരണം	ഉദാഹരണം	a യുടെ വില	b യുടെ വില	ഉത്തരം (a)
==	തുല്യം	a==b	10	3	false
!=	തുല്യമല്ല	a!=b	10	3	true
<	കുറവ്	a<b	10	3	false
<=	കുറവോ അല്ലെങ്കിൽ തുല്യമോ	a<=b	10	3	false
>	കൂടുതൽ	a>b	10	3	true
>=	കൂടുതലോ അല്ലെങ്കിൽ തുല്യമോ	a>=b	10	3	true

പട്ടിക 6.3: റിലേഷണൽ ഓപ്പറേറ്റർ

ഒരു താരതമ്യ/റിലേഷണൽ ഓപ്പറേഷന്റെ ഫലം ശരിയോ തെറ്റോ എന്ന് പട്ടികയിൽ നിന്ന് വ്യക്തമാണ്. ഈ ഓപ്പറേറ്ററുകൾ രണ്ട് വശത്തും ഉള്ള മൂല്യങ്ങൾ താരതമ്യം ചെയ്യുകയും അതിനനുസരിച്ച് ഫലം നൽകുകയും ചെയ്യുന്നു.

6.5.4 ലോജിക്കൽ ഓപ്പറേറ്റർ (Logical Operators)

ജാവാസ്ക്രിപ്റ്റിലെ വിവിധ ലോജിക്കൽ ഓപ്പറേറ്ററുകളുടെ പട്ടിക 6.4 ൽ ഉദാഹരണത്തോടൊപ്പം കാണിച്ചിരിക്കുന്നു.

ഓപ്പറേറ്റർ	വിവരണം	ഉദാഹരണം	a യുടെ വില	b യുടെ വില	ഉത്തരം (a)
&&	ആന്റ്	a && b	true	false	false
||	ഓർ	a || b	true	false	true
!	നോട്ട്	!a	true		false

പട്ടിക 6.4: ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ

മുകളിൽ കൊടുത്തിരിക്കുന്ന പട്ടികയിൽ നിന്ന്, ഈ ഓപ്പറേറ്ററുകൾ എല്ലാം തന്നെ C++ ലേതിന് സമാനമാണെന്നത് വളരെ വ്യക്തമാണ്. വിവിധ ആവശ്യങ്ങൾക്കായി മറ്റ് ധാരാളം ഓപ്പറേറ്ററുകൾ ജാവാസ്ക്രിപ്റ്റ് നൽകുന്നു. ജാവാസ്ക്രിപ്റ്റിനെക്കുറിച്ചുള്ള നമ്മുടെ ചർച്ച ഈ അധ്യായത്തിൽ മാത്രം ഒതുങ്ങിയിരിക്കുന്നതിനാൽ 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++ ൽ ലഭ്യമല്ല.

6.5.5 സ്ട്രിംഗ് അഡീഷൻ ഓപ്പറേറ്റർ (+) (String addition operator (+))

രണ്ട് സംഖ്യകൾ കൂട്ടുന്നതിന് + ഓപ്പറേറ്റർ ഉപയോഗിക്കാമെന്ന് നമുക്ക് അറിയാം. അതേ ഓപ്പറേറ്ററായ '+' രണ്ട് സ്ട്രിംഗുകൾ കൂട്ടിച്ചേർക്കുന്നതിനും ഉപയോഗിക്കാം. കൺകാറ്റ് നേഷൻ (concatenation) എന്നതിനർഥം രണ്ട് സ്ട്രിംഗുകൾ കൂട്ടിചേർക്കുക എന്നാണ്.

ഉദാഹരണം

```
var x, y;
x = "A good beginning ";
y = "makes a good ending.";
z = x + y;
```

+ ഓപ്പറേറ്റർ രണ്ട് സ്ട്രിംഗുകൾ കൂട്ടിച്ചേർക്കും, അങ്ങനെ വേരിയബിൾ z ന്റെ മൂല്യം 'A good beginning makes a good ending' എന്നാകും. ഒരേ ഓപ്പറേറ്റർ വ്യത്യസ്ത ഓപ്പറന്റുകളുടെ തരം മനസ്സിലാക്കിയാണ് പ്രവർത്തിക്കുന്നത്. ഓപ്പറന്റുകൾ സംഖ്യ ആണെങ്കിൽ, അവയുടെ തുക കണ്ടുപിടിക്കും. മറിച്ച് സ്ട്രിംഗുകൾ ആണെങ്കിൽ അവ കൂട്ടിച്ചേർക്കും. ഇനി, താഴെ പറഞ്ഞിരിക്കുന്ന കോഡിൽ z ന്റെ മൂല്യം പ്രവചിക്കുക?

```
var  x, y;
x = "23";
y = 5;
z = x + y;
```

ഇതിന്റെ ഉത്തരം 235 ആണ്. ഓപ്പറന്റുകളിൽ ഏതെങ്കിലും ഒരെണ്ണം സ്ട്രിങ്ങാണെങ്കിൽ + ഓപ്പറേറ്റർ എല്ലാ ഓപ്പറന്റുകളെയും സ്ട്രിങ്ങായി കരുതി കൂട്ടിച്ചേർക്കുന്നു. അങ്ങനെ z ന്റെ വിലയായി 235 ലഭിക്കുന്നു. x ന്റെയും y യുടെയും തുക സംഖ്യാരൂപത്തിൽ കാണണമെന്നുകരുതുക. ഇതിനായി താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ പ്രസ്താവന മാറ്റിയെഴുതാം.

```
z = Number(x) + y;
```

മുകളിൽ പറഞ്ഞിരിക്കുന്ന പ്രസ്താവന പ്രവർത്തിച്ച ശേഷം വേരിയബിൾ z ന്റെ വില 28 ആയിരിക്കും. Number() എന്നത് ജാവാ സ്ക്രിപ്റ്റിലെ ഒരു ഫംഗ്ഷൻ ആണ്, അത് സ്ട്രിംഗ് തരം ഡാറ്റയെ നമ്പറായി മാറ്റുന്നു. ഈ അധ്യായത്തിലെ ചില ഉദാഹരണങ്ങളിൽ ഈ ഫംഗ്ഷൻ ഉപയോഗിക്കുന്നുണ്ട്.

നിങ്ങളുടെ പുരോഗതി അറിയാം

1. നമ്പർ, സ്ട്രിംഗ്, ______ എന്നിവയാണ് ജാവാസ്ക്രിപ്റ്റിലെ അടിസ്ഥാന ഡാറ്റ ഇനങ്ങൾ.
2. 'true' ഒരു ______ ഇനം ഡാറ്റയാണ്.
3. 'false' ഒരു ______ ഇനം ഡാറ്റയാണ്.
4. ജാവാസ്ക്രിപ്റ്റിൽ ഒരു വേരിയബിൾ പ്രഖ്യാപിക്കാൻ വേണ്ട കീവേഡ് ______ ആണ്.
5. ജാവാസ്ക്രിപ്റ്റിൽ ഡാറ്റ ഇനം അറിയാൻ ഉപയോഗിക്കുന്നതിനുള്ള ഫങ്ഷൻ ______ ആണ്.
6. % ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നത് എന്തിനാണ്?
7. ലോജിക്കൽ ഓപ്പറേറ്ററുകൾ ഏവ?

6.6 ജാവാസ്ക്രിപ്റ്റിലെ കൺട്രോൾ സ്ട്രക്ചറുകൾ (Control structures in JavaScript)

ഒരു പ്രോഗ്രാമിന്റെ പ്രവർത്തനത്തിന്റെ നിയന്ത്രണത്തിന് കൺട്രോൾ സ്ട്രക്ചറുകൾ ഉപയോഗിക്കുന്നു. C++ ൽ പഠിച്ചിട്ടുള്ള എല്ലാ കൺട്രോൾ സ്ട്രക്ചറുകളും ജാവാസ്ക്രിപ്റ്റിലും വ്യത്യാസമില്ലാതെ ഉപയോഗിക്കാവുന്നതാണ്. ജാവാ സ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്ന വിവിധ കൺട്രോൾ സ്ട്രക്ചറുകൾ ഉദാഹരണസഹിതം നമുക്ക് പരിശോധിക്കാം.

6.6.1 if

എല്ലാ പ്രോഗ്രാമിങ് ഭാഷയിലും ഏറ്റവുമധികം ഉപയോഗിക്കപ്പെടുന്ന കൺട്രോൾ സ്ട്രക്ചറാണിത്. ചില വ്യവസ്ഥകൾ അടിസ്ഥാനമാക്കി ഒരു പ്രസ്താവന അല്ലെങ്കിൽ ഒരു കൂട്ടം പ്രസ്താവനകൾ പ്രവർത്തിപ്പിക്കാൻ ഇത് ഉപയോഗിക്കുന്നു. പട്ടിക 6.5 ൽ നൽകിയിരിക്കുന്നതുപോലെ രണ്ട് രീതിയിൽ ഇത് ഉപയോഗിക്കാം

if ന്റെ വാക്യഘടന	if ...... ന്റെ വാക്യഘടന

`if (test_expression)`
`{`
`    statements;`
`}`
	`if (test_expression)`
	`{`
	`    statements;`
	`}`
	`else`
	`{`
	`    statements;`
	`}`

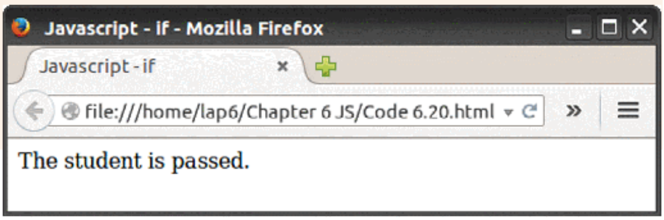
പട്ടിക 6.5: if, else ഓടുകൂടിയ if പ്രസ്താവനയുടെ വാക്യഘടന

if, if .... else പ്രസ്താവനയുടെ ഉപയോഗം കാണിക്കുന്ന ഒരു ഉദാഹരണം നോക്കാം.

ഉദാഹരണം 6.12: ഒരു വിദ്യാർഥി ജയിച്ചോ ഇല്ലയോ എന്നു പരിശോധിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ് നിർമാണം.

ഉദാഹരണം 6.7: രണ്ട് സംഖ്യകളുടെ തുക കണ്ടു പിടിക്കുന്നതിനുള്ള വെബ് പേജ് നിർമാണം

```
<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>
```



ചിത്രം 6.8: if പ്രസ്താവന വ്യക്തമാക്കുന്ന വെബ് പേജ്

മുകളിൽ കൊടുത്തിരിക്കുന്ന പ്രോഗ്രാം കോഡ് if - else പ്രസ്താവന ഉപയോഗിക്കുന്ന വിധം കാണിക്കുന്നു. പ്രോഗ്രാമിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.8 ൽ നൽകിയിരിക്കുന്നു. നിങ്ങൾ 30 ൽ താഴെയുള്ള സംഖ്യ score ന്റെ വിലയായി നൽകി ഫലത്തിൽ വരുന്ന മാറ്റം ശ്രദ്ധിക്കുക.

6.6.2 സ്വിച്ച് (switch)

സ്വിച്ച് ഒരു 'മൾട്ടി ബ്രാഞ്ചിങ്' പ്രസ്താവനയാണ്. ഇതുപയോഗിച്ച് ഒരു എക്സ്പ്രഷന്റെ വില അടിസ്ഥാനമാക്കി വ്യത്യസ്ത പ്രോഗ്രാം കോഡുകൾ പ്രവർത്തിപ്പിക്കുന്നതിനായി തിരഞ്ഞെടുക്കാം. സ്വിച്ചിന്റെ വാക്യഘടന

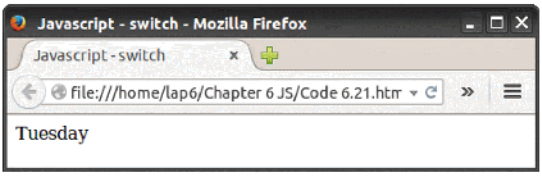
```
switch (പരിശോധന പ്രയോഗങ്ങൾ)
{
    case value1:
               പ്രസ്താവനകൾ;
               break;
    case value2:
               പ്രസ്താവനകൾ;
               break;
    . . . . . . . . . . . . . . . .
    . . . . . . . . . . . . . . . .
    default:
               പ്രസ്താവനകൾ;
}
```

എക്സ്പ്രഷന്റെ വില അടിസ്ഥാനമാക്കി ഉചിതമായ കേസ് ഭാഗം പ്രവർത്തിക്കുന്നു. ഇവിടെ എക്സ്പ്രഷൻ ഒരു വേരിയബിളിന്റെ പേര് ആവാം. നൽകുന്ന നമ്പറിന് അനുയോജ്യമായ ദിവസം പ്രിന്റ് ചെയ്യുന്ന ഒരു HTML കോഡ് താഴെ കൊടുത്തിരിക്കുന്നു.

ഉദാഹരണം 6.8: നൽകുന്ന നമ്പറിന് അനുയോജ്യമായ ദിവസം പ്രിന്റ് ചെയ്യുന്ന വെബ് പേജ് നിർമിക്കുന്നതിനുള്ള 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 ഔട്ട്പുട്ട് ചിത്രം 6.9 ൽ നൽകിയിരിക്കുന്നു.



ചിത്രം 6.9: സ്വിച്ച് പ്രസ്താവനയുടെ ഉപയോഗം വ്യക്തമാക്കുന്ന വെബ് പേജ്

6.6.3 for ലൂപ്പ് (for loop)

ഒരു കൂട്ടം നിർദ്ദേശങ്ങൾ ആവർത്തിച്ച് നടപ്പാക്കാൻ for ലൂപ്പ് ഉപയോഗിക്കുന്നു. for ലൂപ്പിന്റെ ആവർത്തനം നിയന്ത്രിക്കുന്നതിനായി ഒരു ലൂപ്പ് വേരിയബിൾ ഉപയോഗിയ്ക്കുന്നു.

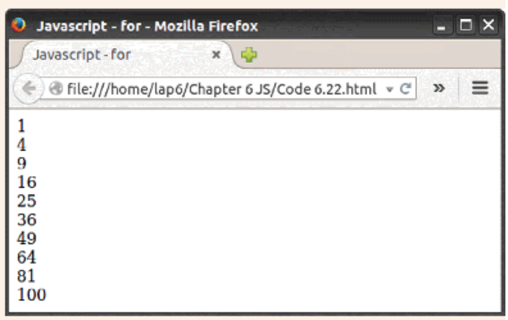
for ലൂപ്പിന്റെ വാക്യഘടന

```
    for(initialisation; test_expression; update_statement)
    {
        പ്രസ്താവനകൾ;
    }
```

initialisation ലൂപ്പ് വേരിയബിളിന് ആദ്യ വില നൽകുവാൻ ഉപയോഗിക്കുന്നു. test_expression ലൂപ്പ് തുടരണോ വേണ്ടയോ എന്ന് പരിശോധിക്കാനുള്ള വ്യവസ്ഥയും update_statement ലൂപ്പ് വേരിയബിളിന്റെ വില വർധിപ്പിക്കുകയോ കുറയ്ക്കുകയോ ചെയ്യുന്നതിനുമുള്ള പ്രസ്താവനയുമാണ്. താഴെ പറയുന്ന ഉദാഹരണം ലൂപ്പിന്റെ പ്രവർത്തനങ്ങൾ വിശദീകരിക്കുന്നു.

ഉദാഹരണം 6.9: ആദ്യത്തെ 10 നമ്പറുകളുടെ വർഗങ്ങൾ പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ് നിർമാണം

```
<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>
```

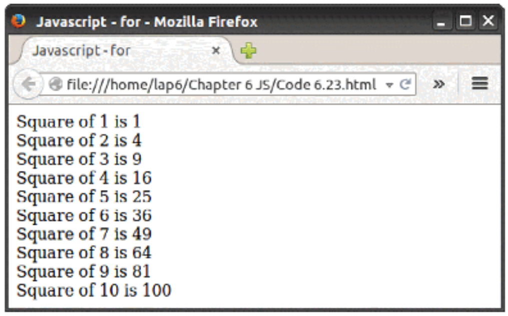


ചിത്രം 6.10: വേരിയബിളുകളുടെ തരം ഉപയോഗം പ്രദർശിപ്പിക്കുന്ന വെബ് പേജ്.

ഇതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.10 ൽ കൊടുത്തിരിക്കുന്നു.

ചിത്രം 6.11: ഔട്ട്പുട്ട് 'Square of 1 is 1' എന്ന് ലഭിക്കുന്നതിന് മുകളിലുള്ള കോഡ് നമുക്ക് പരിഷ്കരിക്കാം.

```
    for (i=1; i<=10; i++)
    {
        s = i*i;
        document.write("Square
of " + i + " is " + s);
        document.write("<BR>");
    }
```



ചിത്രം 6.11: നമ്പറുകളുടെ വർഗ്ഗം പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ് പേജ്.

ഔട്ട്പുട്ട് നിർമ്മിക്കുന്നതിന് സ്ട്രിംഗ് കൂട്ടി ചേർക്കുന്നതിനുള്ള (സ്ട്രിംഗ് അഡീഷൻ) ഓപ്പറേറ്റർ ആയി + ഉപയോഗിക്കുന്നു.

6.6.4 while ലൂപ്പ്

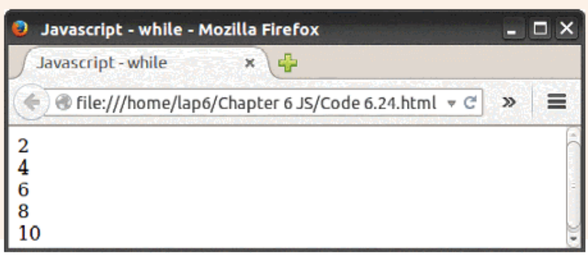
ഒരു വ്യവസ്ഥയുടെ അടിസ്ഥാനത്തിൽ ഒരു കൂട്ടം പ്രസ്താവനകൾ ആവർത്തിച്ച് പ്രവർത്തിപ്പിക്കുന്നതിനാണ് ലൂപ്പ് എന്ന്പറയുന്നത്.

```
    while (test_expression)
    {
        statements;
    }
```

ഇവിടെ test_expression എന്നത് ഒരു വ്യവസ്ഥയാണ്. വ്യവസ്ഥ ശരിയായിരിക്കുന്നിടത്തോളം ലൂപ്പിനുള്ളിലെ പ്രസ്താവനകൾ തുടർച്ചയായി പ്രവർത്തിച്ചുകൊണ്ടിരിക്കും. താഴെക്കാണുന്ന ഉദാഹരണം 10 വരെയുള്ള ഇരട്ട സംഖ്യകൾ പ്രദർശിപ്പിക്കുന്നു.

ഉദാഹരണം 6.10: പത്തു വരെയുള്ള ഇരട്ട സംഖ്യകൾ പ്രദർശിപ്പിക്കുന്നത്തിനുള്ള വെബ്പേജ് നിർമാണം

```
<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 ലൂപ്പിന്റെ ഉപയോഗത്തെ വ്യക്തമാക്കുന്ന വെബ് പേജ്

തന്നിരിക്കുന്ന കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.12 ൽ നൽകിയിരിക്കുന്നു.

ജാവാസ്ക്രിപ്റ്റിൽ C++ ലേതിന് സമാനമായ do while ലൂപ്പ് ലഭ്യമാണ്. മിക്കവാറും എല്ലാ ജോലികളും for ലൂപ്പിലും while ലൂപ്പിലും ചെയ്യാമെന്നതിനാൽ ഈ അധ്യായത്തിൽ നമ്മൾ മറ്റ് ലൂപ്പുകൾ ചർച്ച ചെയ്യുന്നില്ല.

നമുക്കു ചെയ്യാം

ഉദാഹരണം 6.10 ൽ document.write("<BR>"); എന്ന പ്രസ്താവന ഒഴിവാക്കുകയാണെങ്കിൽ എന്ത് ഔട്ട്പുട്ട് ലഭിക്കും;

നിങ്ങളുടെ പുരോഗതി അറിയാം

1. ഒരു കൂട്ടം പ്രസ്താവനകൾ തിരഞ്ഞെടുത്ത് പ്രവർത്തിപ്പിക്കുന്നതിന് __________ ഉം ________ ഉം കൺട്രോൾ സ്ട്രക്ചറുകൾ ഉപയോഗിക്കുന്നു.
2. ജാവാസ്ക്രിപ്റ്റിലെ ലൂപ്പിങ് പ്രസ്താവനകൾക്ക് ഉദാഹരണങ്ങൾ എഴുതുക.
3. ____________ ഒരു മൾട്ടി ബ്രാഞ്ചിങ് പ്രസ്താവനയാണ്.
4. ശരിയോ തെറ്റോ എന്നെഴുതുക.
   a. break പ്രസ്താവന switch ബ്ലോക്കിനുള്ളിൽ ഉപയോഗിക്കുന്നു.
   b. switch, പ്രസ്താവന ഉപയോഗിക്കുകയാണെങ്കിൽ, ചുരുങ്ങിയത് ഒരു തവണയെങ്കിലും അതിൽ break പ്രസ്താവന ഉപയോഗിച്ചിരിക്കണം.
   c. if-else ഉപയോഗിച്ച് എഴുതിയ പ്രോഗ്രാമുകൾ എല്ലാം സ്വിച്ച് 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 അന്തർ നിർമിത ഫങ്ഷനുകൾ (Built-in function)

ജാവാസ്ക്രിപ്റ്റിൽ ധാരാളം അന്തർനിർമിത ഫങ്ഷനുകൾ ലഭ്യമാണ്. ഫങ്ഷനുകളെ മെത്തേഡ് എന്നും വിളിക്കുന്നു. സാധാരണയായി ഉപയോഗിക്കുന്ന ഏതാനും ഫങ്ഷനുകളെക്കുറിച്ച് ചർച്ച ചെയ്യാം.

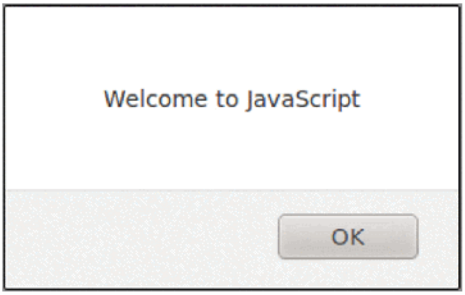
a. alert()

സ്ക്രീനിൽ ഒരു സന്ദേശം പ്രദർശിപ്പിക്കാൻ ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു.

ഉദാഹരണത്തിന് :

```
alert("Welcome to JavaScript");
```

എന്ന പ്രസ്താവന ഉപയോഗിച്ച് ചിത്രം 6.13 ൽ കാണിച്ചിരിക്കുന്ന സന്ദേശ ജാലകം ബ്രൗസറിൽ കാണിക്കാം. ഡാറ്റയുടെ സാധുത ഉറപ്പുവരുത്തുന്ന സമയത്ത് ഉപയോക്താവിന് സന്ദേശം നൽകുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു.



ചിത്രം 6.13: സന്ദേശജാലകം

b. isNaN()

തന്നിരിക്കുന്ന വില ഒരു സംഖ്യയാണോ അല്ലയോ എന്നു പരിശോധിക്കുന്നതിന് ഈ ഫങ്ഷൻ ഉപയോഗിക്കുന്നു. ഈ ഫങ്ഷനിൽ, NaN എന്നത് നോട്ട് എ നമ്പർ എന്ന് സൂചിപ്പിക്കുന്നു. നൽകിയിരിക്കുന്ന വില ഒരു സംഖ്യയല്ലെങ്കിൽ ഫങ്ഷൻ true എന്ന് തിരികെ നൽകുന്നു. ഉദാഹരണത്തിന്,

1. `isNaN("welcome");`
2. `isNaN("A123");`
3. `isNaN("Score50");`
4. `isNaN("A");`

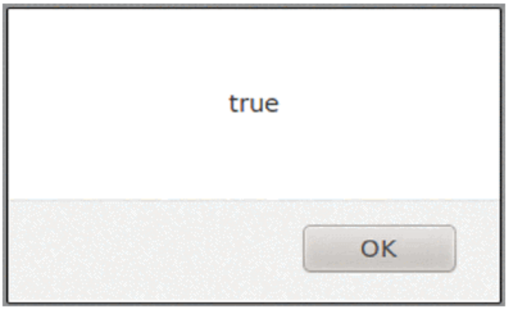
താഴെ പറയുന്ന പ്രസ്താവനകൾ false എന്ന വില തിരികെ നൽകുന്നു.

1. `isNaN("13");`
2. `isNaN(13);`
3. `isNaN("13.5");`
4. `isNaN("0.123");`

താഴെ പറയുന്ന പ്രസ്താവന ചിത്രം 6.14 ൽ കാണിച്ചിരിക്കുന്ന സന്ദേശ ജാലകം ബ്രൗസറിൽ പ്രദർശിപ്പിക്കുന്നു.

```
alert(isNaN("A"));
```

ഡാറ്റയുടെ സാധൂകരണത്തിനായി ഈ ഫങ്ഷൻ വളരെ ഉപയോഗപ്രദമാണ്. ഉദാഹരണത്തിന്, ഒരു വിദ്യാർഥിയുടെ പ്രായം നൽകുന്നതിന് വെബ് പേജിൽ ഒരു ടെക്സ്റ്റ് ബോക്സ് ഉണ്ട് എന്ന് കരുതുക, അബദ്ധത്തിൽ അതിൽ സംഖ്യക്ക് പകരം ഒരു ക്യാരക്ടർ നൽകിയേക്കാം. അപ്പോൾ ഈ ഫങ്ഷൻ, ഇൻപുട്ട് നൽകിയത് ഒരു നമ്പർ ആണോയെന്ന് പരിശോധിക്കുകയും ഇത് ഒരു സംഖ്യയല്ലെങ്കിൽ, alert() ഫങ്ഷൻ ഉപയോഗിച്ച് ഒരു സന്ദേശം പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു.



ചിത്രം 6.14: isNaN() ന്റെ ഔട്ട്പുട്ട്

c. toUpperCase()

ഈ ഫങ്ഷൻ തന്നിരിക്കുന്ന സ്ട്രിംഗിനെ അപ്പർ കേസാക്കി മാറ്റുന്നു. താഴെ തന്നിരിക്കുന്ന ഉദാഹരണം നോക്കുക.

```
var x, y;
x = "JavaScript";
y = x.toUpperCase();
alert(y);
```

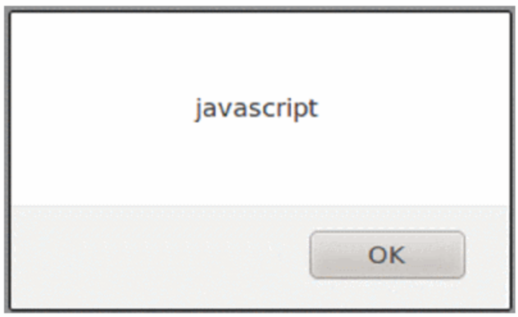
ചിത്രം 6.15: LowerCase() ന്റെ ഔട്ട്പുട്ട്

ഈ കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.15 ൽ കാണിച്ചിരിക്കുന്നു. toUpperCase() എന്ന ഫങ്ഷൻ സ്ട്രിംഗ് വേരിയബിൾ x ന്റെ പേരിനോട് ചേർത്ത് വിളിക്കുന്നു. അതായത് toUpperCase() എന്നത് സ്ട്രിംഗിനെ വലിയ അക്ഷരമാക്കി തിരികെ നൽകുന്നു. ജാവസ്ക്രിപ്റ്റ് കേസ് സെൻസിറ്റീവ് ഭാഷയാണ്. അതിനാൽ കോഡിൽ ഫങ്ഷൻ ഏതു തരം അക്ഷരത്തിലാണോ അതേ രീതിയിൽ തന്നെ ഉപയോഗിക്കണം.

d. toLowerCase()

ഈ ഫങ്ഷൻ തന്നിരിക്കുന്ന സ്ട്രിംഗിനെ ചെറിയ അക്ഷരമാക്കി തിരികെ നൽകുന്നു. താഴെ നൽകിയിരിക്കുന്ന ഉദാഹരണം ശ്രദ്ധിക്കുക

```
var x, y;
x = "JavaScript";
y = x.toLowerCase();
alert(y);
```



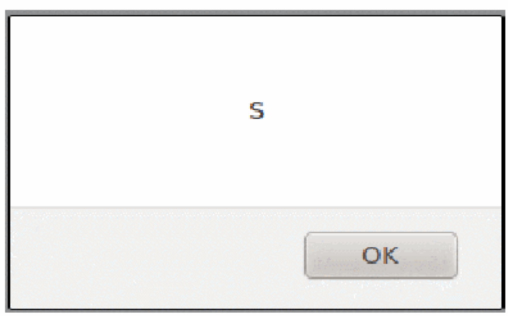
ചിത്രം 6.16: toLowerCase() ന്റെ ഔട്ട്പുട്ട്

മുകളിലെ കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.16ൽ കാണിച്ചിരിക്കുന്നു. സ്ട്രിംഗിൽ എല്ലാ ക്യാരക്ടറുകളും ഇപ്പോൾ ചെറിയ അക്ഷരത്തിലാണ് അതിനാൽ toLowerCase() അതേ സ്ട്രിംഗ് തിരികെ നൽകുന്നു.

e. charAt()

ഈ ഫങ്ഷൻ ഒരു പ്രത്യേക സ്ഥാനത്തുള്ള ക്യാരക്ടർ തിരികെ നൽകുന്നു. charAt(0) എന്നത് സ്ട്രിംഗിലെ ആദ്യത്തെ അക്ഷരം തിരികെ നൽകുന്നു. charAt(1) സ്ട്രിംഗിലെ രണ്ടാമത്തെ അക്ഷരം തിരികെ നൽകുന്നു. ഉദാഹരണം നോക്കുക.

```
var x;
x = "JavaScript";
y = x.charAt(4);
alert(y);
```



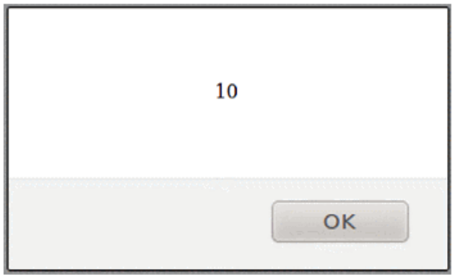
ചിത്രം 6.17: charAt() ന്റെ റിസൾട്ട്

ഇതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.17 ൽ നൽകിയിരിക്കുന്നു. വേരിയബിൾ x ലെ സ്ട്രിംഗിന്റെ അഞ്ചാമത്തെ അക്ഷരം 'S' ആയതിനാൽ ബ്രൗസർ ജാലകത്തിൽ B അക്ഷരം കാണിക്കുന്നു.

f. length പ്രോപ്പർട്ടി

ഫങ്ഷനുകൾ കൂടാതെ, പ്രോഗ്രാമർക്ക് ഉപയോഗപ്രദമായ ചില പ്രോപ്പർട്ടികളും സ്ട്രിംഗ് വേരിയബിളിനോടൊപ്പം ഉപയോഗി ക്കാം. length പ്രോപ്പർട്ടി സ്ട്രിംഗിന്റെ നീളം തിരികെ നൽകുന്നു. സ്ട്രിംഗിലെ ക്യാരക്ടറുകളുടെ എണ്ണമാണ് നീളം എന്നതു കൊണ്ട് ഉദ്ദേശിക്കുന്നത്. ഉദാഹരണത്തിന്

```
var x, n;
x = "JavaScript";
n = x.length;
alert(n);
```



ചിത്രം 6.18: length പ്രോപ്പർട്ടിയുടെ റിസൾട്ട്

ഇവിടെ length പ്രോപ്പർട്ടി എങ്ങനെ വിളിക്കുന്നു എന്ന് നോക്കാം. x.length എന്ന രൂപത്തിൽ x എന്ന വേരിയബിളിന്റെ പേര് ചേർത്തതാണ് ഈ പ്രോപ്പർട്ടി വിളിക്കുന്നത്. മുകളിലെ കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.18 ൽ കാണിച്ചിരിക്കുന്നു. ഫങ്ഷനും പ്രോപ്പർട്ടിയും തമ്മിലുള്ള പ്രധാന വ്യത്യാസം, ഫങ്ഷനോടൊപ്പം () ബ്രാക്കറ്റ് ഉപയോഗിക്കുന്നു (ചില സന്ദർഭങ്ങളിൽ ബ്രാക്കറ്റിൽ പരാമീറ്ററുകളും) എന്നാൽ പ്രോപ്പർട്ടിയോടൊപ്പം () ബ്രാക്കറ്റിന്റെ ആവശ്യം ഇല്ല എന്നതാണ്.

നിങ്ങളുടെ പുരോഗതി അറിയാം

താഴെപ്പറയുന്നവയിൽ വേരിയബിൾ y യുടെ വില എഴുതുക.

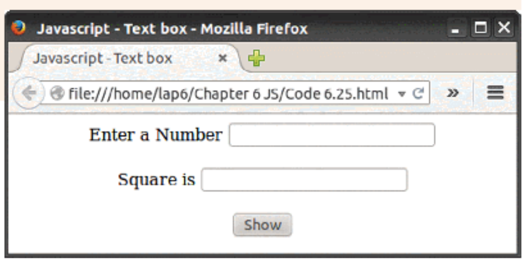
1. ```
   x = "welcome";
   y = x.length;
   ```
2. ```
   x = "WELCOME";
   y = x.toLowerCase();
   ```
3. ```
   x = "Welcome";
   y = x.toUpperCase();
   ```
4. ```
   x = "welcome";
   y = x.toLowerCase();
   ```
5. ```
   x = "welcome";
   y = isNaN(x);
   ```
6. ```
   x = "welcome";
   y = charAt(3);
   ```

6.8 ജാവാസ്ക്രിപ്റ്റിലൂടെ ടെക്സ്റ്റ് ബോക്സിലെ വിലകൾ സ്വീകരിക്കുന്നവിധം (Accessing values in a textbox using JavaScript)

മുൻ അധ്യായത്തിൽ ഒരു വെബ് പേജിൽ ടെക്സ്റ്റ് ബോക്സ്, ചെക്ക്ബോക്സ്, റേഡിയോ ബട്ടൺ, സബ്മിറ്റ് ബട്ടൺ തുടങ്ങിയ വിവിധ കൺട്രോളുകൾ (controls) എങ്ങനെ ഉപയോഗിക്കാം എന്ന് പഠിച്ചു. ഇത്തരം കൺട്രോളുകളെ ജാവാസ്ക്രിപ്റ്റിന്റെ സഹായത്താൽ എങ്ങിനെ ഉപയോഗിക്കാം എന്നാണ് ഇവിടെ ചർച്ചചെയ്യുന്നത്. ഈ അധ്യായത്തിൽ നമ്മൾ ചർച്ച ചെയ്ത ഒരു പ്രോഗ്രാമിലും ഉപയോക്താവിൽ നിന്ന് യാതൊരുവിധ ഇൻപുട്ടും സ്വീകരിച്ചിട്ടില്ല. പ്രോസസ്സിംഗിന് ആവശ്യമായ ഡാറ്റ നേരിട്ട് പ്രോഗ്രാം കോഡിൽ തന്നെ നൽകുകയാണ് ചെയ്തത്. ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിച്ച് ഒരു വെബ് പേജ് ഘടകങ്ങളെ എങ്ങനെ സ്വീകരിക്കുന്നു എന്ന് നോക്കാം. ഈ പാഠഭാഗത്തിനുശേഷം നമുക്ക് ശരിയായ ഒരു ഇന്ററാക്ടീവ് വെബ് പേജ് രൂപപ്പെടുത്തുവാൻ കഴിയും. അതായത്, ഉപയോക്താവ് ടെക്സ്റ്റ് ബോക്സിൽ വില നൽകാനും അതിന്മേൽ ചില പ്രോസസിംഗ് നടത്തി ഫലം മറ്റൊരു ടെക്സ്റ്റ് ബോക്സിൽ പ്രദർശിപ്പിക്കാനും കഴിയും. ഈ കാര്യങ്ങൾ മനസ്സിലാക്കുന്നതിന് താഴെ കൊടുത്തിരിക്കുന്ന HTML കോഡ് ശ്രദ്ധാപൂർവം വായിക്കുക.

ഉദാഹരണം 6.11: ഒരു വെബ് ഫോം (web form) പ്രദർശിപ്പിക്കുന്ന ഒരു വെബ് പേജ് നിർമാണം

```
<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>
```



ചിത്രം 6.19: വെബ് ഫോം പ്രദർശിപ്പിക്കുന്ന ഒരു വെബ് പേജ്

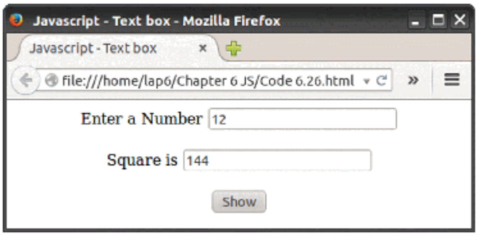
ഈ കോഡിന്റെ പ്രവർത്തന ഫലം ചിത്രം 6.19 ൽ കൊടുത്തിരിക്കുന്നു. ഇവിടെ ഫോമിന് frmSquare എന്നും, ടെക്സ്റ്റ് ബോക്സുകൾക്ക് txtNum, txtSquare എന്നും

പേരുകൾ നൽകിയിരിക്കുന്നു. ജാവസ്ക്രിപ്റ്റിൽ ഇത്തരം ഘടകങ്ങൾക്ക് ഉപയോഗിക്കുന്നതിന് അവയ്ക്ക് പേര് നൽകേണ്ടത് വളരെ പ്രധാനമാണ്. ഒരു വെബ് പേജ് ഘടകത്തിന് നമ്മൾ പേര് നൽകുന്നില്ലെങ്കിൽ, ജാവസ്ക്രിപ്റ്റിന് അവയെ ഉപയോഗിക്കാൻ കഴിയില്ല. Show എന്ന് എഴുതിയിരിക്കുന്ന സബ്മിറ്റ് ബട്ടന് പേര് നൽകിയിട്ടില്ലെന്നത് ശ്രദ്ധിക്കുക. കാരണം, ഈ ബട്ടൺ ജാവസ്ക്രിപ്റ്റിൽ നിന്ന് പരാമർശിക്കാൻ കഴിയില്ല.

ഇനി നമുക്ക് ഈ പ്രോഗ്രാം കോഡിൽ കുറച്ചു മാറ്റം വരുത്തി നോക്കാം.

ഉദാഹരണം 6.12: നൽകുന്ന സംഖ്യയുടെ വർഗം (സ്ക്വയർ) പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ് നിർമാണം.

```
<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 വർഗം പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ് പേജ്

മുകളിലുള്ള കോഡ് ശ്രദ്ധാപൂർവം പരിശോധിച്ച് കൂട്ടിച്ചേർത്തിട്ടുള്ളവ ശ്രദ്ധിക്കുക (ഉദാഹരണം 6.17). വെബ് പേജിന്റെ ഹെഡ് ഭാഗത്ത് `showSquare()` എന്ന ഫങ്ഷൻ നിർവചിച്ചിരിക്കുന്നു. താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ ഈ ഫങ്ഷനെ വിളിക്കുന്നു.

```
<INPUT Type= "button" Value=
"Show" onClick= "showSquare()">
```

```
onclick= ""
```

എന്നത് ഇൻപുട്ട് ടാഗിന്റെ ഉള്ളിലാണ് എഴുതിയിട്ടുള്ളത്. ഉപയോക്താവ് ഈ ബട്ടൺ ക്ലിക്കുചെയ്യുമ്പോൾ, `showSquare()` എന്ന ഫങ്ഷൻ പ്രവർത്തിക്കുന്നു. ബട്ടനിൽ ക്ലിക്കുചെയ്തുകൊണ്ട് ഒരു ഫങ്ഷൻ എങ്ങനെ വിളിക്കാം എന്ന് മനസ്സിലായായല്ലോ.

ഫങ്ഷൻ നിർവചനത്തിലെ താഴെപ്പറയുന്ന വരി ശ്രദ്ധിക്കുക

```
num = document.frmSquare.txtNum.value;
```

`document.frmSquare.txtNum.value` നോക്കുക. ഇവിടെ 'document' എന്നത് വെബ് പേജിന്റെ ബോഡി ഭാഗത്തെ സൂചിപ്പിക്കുന്നു. `frmSquare` എന്നത് body ഭാഗത്ത് നൽകിയിട്ടുള്ള ഫോമിന്റെ പേരാണ്. 'txtNum' എന്നത് 'frmSquare'- ൽ ഉള്ള ടെക്സ്റ്റ് ബോക്സിന്റെ പേരാണ്, `value` എന്നത് ടെക്സ്റ്റ് ബോക്സിലെ ഉള്ളടക്കത്തെ പരാമർശിക്കുന്നു. അതായത് `document.frmSquare.txtNum.value` എന്നത് ഡോക്യുമെന്റിലെ `frmSquare` എന്ന ഫോമിലെ `txtNum` ന്റെ വിലയാണ്. മുകളിൽ പറഞ്ഞ വരി ടെക്സ്റ്റ് ബോക്സിലെ വില വേരിയബിൾ `num`-ൽ ലഭ്യമാക്കും.

ഇനി പറയുന്ന വരിയുടെ അർഥം മനസ്സിലാക്കാൻ ശ്രമിക്കുക.

```
document.frmSquare.txtSqr.value = ans;
```

മുകളിലുള്ള വരി രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ `ans` എന്ന വേരിയബിളിന്റെ വില കാണിക്കുന്നു. ബട്ടനിൽ ക്ലിക്കുചെയ്യുമ്പോൾ വെബ്പേജിലെ ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ നൽകിയിരിക്കുന്ന സംഖ്യയുടെ വർഗം രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ കാണിക്കുന്നു. ഉപയോക്താവിന് ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഏത് സംഖ്യയും ടൈപ്പു ചെയ്യാം, അതിന്റെ വർഗ്ഗം കാണുന്നതിന് show ബട്ടൺ ക്ലിക്കുചെയ്യുക. പ്രവർത്തന സമയത്തുള്ള വെബ് പേജിന്റെ സ്ക്രീൻ ഷോട്ട് ചിത്രം 6.20 ൽ കൊടുത്തിരിക്കുന്നു

```
<INPUT Type= "button" Value= "Show" onClick= "showSquare()">
```

എന്നത്

```
<INPUT Type= "button" Value= "Show"
                    onMouseEnter= "showSquare()">
```

എന്നാക്കിയാൽ എന്ത് സംഭവിക്കുമെന്ന് നോക്കാം. ബട്ടനു മുകളിലൂടെ മൗസ് പോയിന്റർ നീക്കുമ്പോൾ ഫങ്ഷൻ പ്രവർത്തിപ്പിക്കുന്നതിനായി വിളിക്കുന്നു. ഫങ്ഷൻ പ്രവർത്തനത്തിനുവേണ്ടി നിങ്ങൾ ബട്ടണിൽ ക്ലിക്ക് ചെയ്യേണ്ടതില്ല. ഒരു വെബ് പേജിലെ ചില

ബട്ടനു മുകളിലൂടെ മൗസ് പോയിന്റർ നീക്കുമ്പോൾ, ബട്ടന്റെ നിറം മാറുന്നത് നിങ്ങൾ ശ്രദ്ധിച്ചിരിക്കാം. ബട്ടന്റെ നിറം മാറുന്നതിനുള്ള ഒരു ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷൻ എഴുതി അത് `onMouseEnter` എന്ന ഇവന്റിലൂടെ വിളിച്ച് ഇത് സാധ്യമാക്കാം.

<INPUT Type= "button" Value= "Show"onClick= "showSquare()">

`onMouseEnter`, `onClick`, `onMouseEnter`, `onMouseLeave`, `onKeyDown`, `onKeyUp` തുടങ്ങിയവ സാധാരണയായി ഉപയോഗിക്കുന്ന ചില ഇവന്റുകളാണ്.

സാധാരണ ഉപയോഗിക്കുന്ന ചില ജാവ സ്ക്രിപ്റ്റ് ഇവന്റുകളും അവയുടെ വിവരണങ്ങളും പട്ടിക 6.6 ൽ ചേർത്തിരിക്കുന്നു.

ഇവന്റ്	വിവരണം
onClick	ഉപയോക്താവ് ഒരു ഒബ്ജക്റ്റിൽ ക്ലിക്കുചെയ്യുമ്പോൾ സംഭവിക്കുന്നു
onMouseEnter	മൗസ് പോയിന്റർ ഒരു ഒബ്ജക്റ്റിലേക്ക് നീങ്ങുമ്പോൾ സംഭവിക്കുന്നു
onMouseLeave	മൗസ് പോയിന്റർ ഒരു ഒബ്ജക്റ്റിൽ നിന്നും മാറുകയാണെങ്കിൽ സംഭവിക്കുന്നു
onKeyDown	ഉപയോക്താവ് കീബോർഡിലെ ഒരു കീ അമർത്തുമ്പോൾ സംഭവിക്കുന്നു
onKeyUp	ഉപയോക്താവ് കീബോർഡിലെ ഒരു കീയിൽ നിന്ന് വിടുമ്പോൾ സംഭവിക്കുന്നു

പട്ടിക 6.6: സാധാരണ ഉപയോഗിക്കുന്ന ജാവാസ്ക്രിപ്റ്റ് ഇവന്റുകൾ

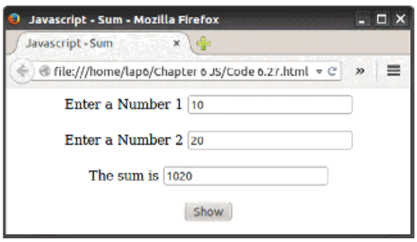
ഒരു വെബ്പേജിൽ രണ്ട് വ്യത്യസ്ത ടെക്സ്റ്റ് ബോക്സുകളിൽ ഉപയോക്താവിന് രണ്ട് സംഖ്യ നൽകാനും, ഒരു ബട്ടണിൽ അമർത്തുമ്പോൾ ഈ സംഖ്യകളുടെ തുക മൂന്നാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ കാണിക്കുന്നതിനുമുള്ള ഒരു വെബ്പേജ് നിർമ്മിക്കാം. കോഡ്, ഔട്ട്പുട്ട് എന്നിവ ചിത്രം 6.21 ൽ കൊടുത്തിരിക്കുന്നു.

ഉദാഹരണം 6.13: രണ്ട് സംഖ്യകളുടെ തുക കാണിക്കുന്ന വെബ് പേജ് നിർമാണം

```
<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>
```

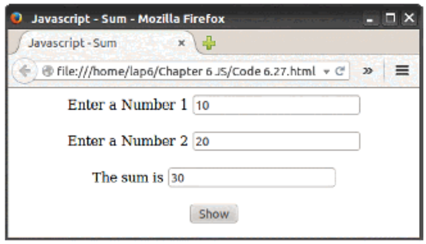
ഈ പ്രോഗ്രാമിന് ഇൻപുട്ടായി 10, 20 എന്നീ സംഖ്യകൾ നൽകിയപ്പോൾ റിസൾട്ടായി 1020 എന്നാണ് കിട്ടിയത് (ചിത്രം 6.21 ൽ കാണിച്ചിരിക്കുന്നു). സ്ട്രിങ്ങുകൾ കൂട്ടിച്ചേർക്കുന്നതിന് + ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നു എന്നതിനാലാണിത്. നമ്മൾ രണ്ട് സ്ട്രിങ്ങുകൾ "10"+"20" ചേർക്കുമ്പോൾ ഉത്തരം '1020' ആണല്ലോ. ഒരു, ടെക്സ്റ്റ് ബോക്സിന്റെ തനതുവില എല്ലായ്പ്പോഴും സ്ട്രിങ്ങായിരിക്കും അതിനാൽ ടെക്സ്റ്റ് ബോക്സിലെ ഉള്ളടക്കം ഒരു നമ്പർ ആയിരുന്നാലും ഒരു വേരിയബിളിലേക്ക് നൽകുമ്പോൾ അതിനെ സ്ട്രിംഗ് ആയി മാത്രമേ പരിഗണിക്കൂ. രണ്ട് സംഖ്യകളുടെ തുക കിട്ടുന്നതിന് ഫങ്ഷൻ showSum() താഴെ കൊടുത്തിരിക്കുന്ന രീതിയിൽ പരിഷ്കരിക്കാം.



ചിത്രം 6.21: ഓപ്പറേറ്റർ ഉപയോഗത്തെ ചിത്രീകരിക്കുന്നതിന് വെബ് പേജ്

```
    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;
    }
```

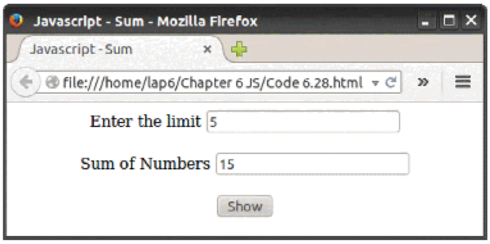
ഇവിടെ Number() എന്ന ഫങ്ഷൻ, ഡാറ്റയെ സംഖ്യയായി മാറ്റി ആ സംഖ്യ num1 എന്ന വേരിയബിളിലേക്ക് നൽകുന്നു. ഇവിടെ num1, num2 എന്നിവയെ നമ്പർ തരം ഡാറ്റയായി കണക്കാക്കുകയും അതുവഴി ശരിയായ തുക കിട്ടുകയും ചെയ്യും. ഉദാഹരണം 6.13 ലെ showSum() എന്ന ഫങ്ഷനിൽ മാറ്റം വരുത്തിയാൽ ചിത്രം 6.22 ൽ കാണിച്ചിരിക്കുന്ന വെബ് പേജ് ഔട്ട്പുട്ടായി ലഭിക്കും. തന്നിരിക്കുന്ന പരിധി വരെയുള്ള സംഖ്യകളുടെ തുക പ്രദർശിപ്പിക്കുന്നതിനുള്ള ഒരു വെബ് പേജ് നിർമിക്കാം. ഇതിൽ പരിധി ഉപയോക്താവിന് ടെക്സ്റ്റ് ബോക്സിലൂടെ നൽകാൻ കഴിയണം.



ചിത്രം 6.22: രണ്ട് സംഖ്യകളുടെ തുക കാണിക്കുന്ന വെബ് പേജ്

ഉദാഹരണം 6.14: തന്നിരിക്കുന്ന പരിധിവരെയുള്ള സംഖ്യകളുടെ തുക കാണിക്കുന്നതിനുള്ള വെബ് പേജ് നിർമാണം

```
<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 6.23: നൽകിയ പരിധിവരെയുള്ള സംഖ്യകളുടെ തുക കാണിക്കുന്ന വെബ് പേജ്

മുകളിലെ കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.23 ൽ നൽകിയിരിക്കുന്നു. ഈ വെബ് പേജിൽ ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഉപയോക്താവ് പരിധി നൽകണം. അതിനു ശേഷം **show** ബട്ടണിൽ ക്ലിക്കുചെയ്യുമ്പോൾ നൽകിയ പരിധി വരെയുള്ള സംഖ്യകളുടെ തുക രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ കാണിക്കുന്നു. ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഉപയോക്താവ് പരിധി നൽകാതെ **show** ബട്ടൺ ക്ലിക്കുചെയ്താൽ അത് ഒരു സന്ദേശവും പ്രദർശിപ്പിക്കുന്നില്ല. പരിധി ഇല്ലെങ്കിൽ തുക പൂജ്യമായിരിക്കും കാരണം ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സ് ശൂന്യമാകുമ്പോൾ `Number()` ഫങ്ഷൻ ശൂന്യ വിലയെ പൂജ്യമായി മാറ്റുന്നു. അതായത് `document.frmSum.txtLimit.value` ന്റെ വില പൂജ്യമായിരിക്കും. അങ്ങനെ ലൂപ്പ് പ്രവർത്തിക്കാതിരിക്കുകയും `sum` എന്ന വേരിയബിളിന്റെ ആദ്യ വിലയായ പൂജ്യം രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ പ്രദർശിപ്പിക്കുകയും ചെയ്യുന്നു. ജാവാസ്ക്രിപ്റ്റ് ഒരിക്കലും ഒരു തെറ്റ് ഉണ്ടെന്ന് കാണിച്ച് തരില്ല. സ്ക്രിപ്റ്റ് എഞ്ചിന് എന്തെങ്കിലും നിർദേശം പ്രാവർത്തികമാക്കാൻ കഴിയാത്തപ്പോൾ, അത് ഫങ്ഷനിലെ ആ വരിയും ബാക്കി ഭാഗങ്ങളും അവഗണിക്കും.

താഴെ കൊടുത്തിരിക്കുന്ന `sumLimit()` ഫങ്ഷൻ 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 നിർദേശങ്ങൾ ഒഴിവാക്കിക്കൊണ്ട് ഫങ്ഷനിൽ നിന്ന് പുറത്തുകടക്കുവാൻ `return` പ്രസ്താവന ഉപയോഗിക്കുന്നു. ഇത് C++ ൽ ഉപയോഗിക്കുന്ന റിട്ടേൺ പ്രസ്താവനയ്ക്ക് സമാനമാണ്. ഇനി നമുക്ക് ടെക്സ്റ്റ് ബോക്സിൽ നൽകിയ പരിധി അക്ഷരമാണോ സംഖ്യയാണോ എന്ന് പരിശോധിക്കുന്നതിനായി `isNaN()` എന്ന ഫങ്ഷൻ ഉപയോഗിക്കാം. ഉപയോക്താവ് ശരിയായ ഡാറ്റ നൽകിയിട്ടുണ്ടോ എന്ന് പരിശോധിക്കുന്നതിന് ഇത് സഹായിക്കുന്നു.

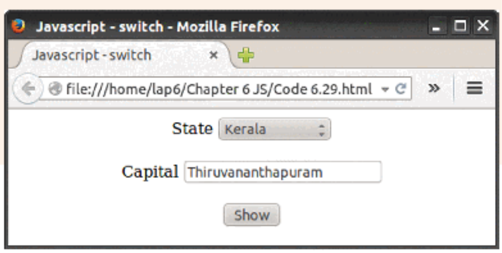
```
function sumLimit()
{
    var sum = 0, i, limit;
    if (document.frmSum.txtLimit.value == "")
    {
        alert("Please enter the limit!");
        return;
    }
    if (isNaN(document.frmSum.txtLimit.value))
    {
        alert("Please enter a number as the limit!");
        return;
    }
    limit = Number(document.frmSum.txtLimit.value);
    for(i = 1; i <= limit; i++)
        sum += i;
    document.frmSum.txtSum.value = sum;
}
```

ക്ലൈന്റ് ഭാഗത്തെ ഡാറ്റയുടെ സാധുത വിലയിരുത്തുന്നതിനായി ജാവ സ്ക്രിപ്റ്റ് ഉപയോഗിക്കുന്നു. ആവശ്യമുള്ള എല്ലാ ഡാറ്റയും നൽകിയിട്ടുണ്ടോ, നൽകിയ ഡാറ്റ ശരിയായ ഘടനയിലാണോ തുടങ്ങിയവ പരിശോധിക്കാനായി ജാവാസ്ക്രിപ്റ്റ് ഉപയോഗിക്കാം. ഏതെങ്കിലും പോരായ്മ ഉണ്ടെങ്കിൽ അത് ഓർമ്മിപ്പിക്കുന്ന ഒരു സന്ദേശം കാണിക്കുവാൻ കഴിയും. താഴെക്കൊടുത്തിരിക്കുന്ന ഉദാഹരണത്തിൽ ജാവാസ്ക്രിപ്റ്റിലെ ഒരു ഡ്രോപ്പ് ഡൗൺ ലിസ്റ്റ് ഉപയോഗിക്കുന്നു ഈ ഡ്രോപ്പ് ഡൌൺ ലിസ്റ്റ് ഉപയോഗിച്ച് ഒരു സംസ്ഥാനം തെരഞ്ഞെടുക്കാനും show ബട്ടണിൽ ക്ലിക്കുചെയ്യുമ്പോൾ, തെരഞ്ഞെടുത്ത സംസ്ഥാനത്തിന്റെ തലസ്ഥാനം ഒരു ടെക്സ്റ്റ് ബോക്സിൽ പ്രദർശിപ്പിക്കാനുമുള്ള ഒരു വെബ്പേജ് നൽകിയിരിക്കുന്നു. ഇതിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.24 ൽ കൊടുത്തിരിക്കുന്നു.

ഉദാഹരണം 6.15: ഒരു സംസ്ഥാനത്തിന്റെ തലസ്ഥാനം പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ്പേജ് നിർമാണം

```
<HTML>
<HEAD> <TITLE>Javascript - switch</TITLE>
<SCRIPT Language= "JavaScript">
function capital()
{
    var n, answer;
    n = document.frmCapital.cboState.selectedIndex;
    switch (n)
    {
```

```
    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: സംസ്ഥാനത്തിന്റെ തലസ്ഥാനം പ്രദർശിപ്പിക്കുന്നതിനുള്ള വെബ് പേജ്

ഈ പ്രോഗ്രാമിലെ താഴെ കൊടുത്തിരിക്കുന്ന പ്രസ്താവന പരിഗണിക്കുക.

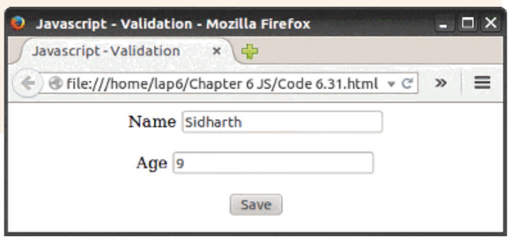
```
document.frmCapital.cboState.selectedIndex;
```

ഇവിടെ 'cboState' എന്നത് ഡ്രോപ്ഡൌൺ ലിസ്റ്റിന്റെ പേരാണ്. 'selectedIndex' എന്നത് ഡ്രോപ് ഡൌൺ ലിസ്റ്റിന്റെ സൂചിക കാണിക്കുന്നു. ഒന്നാമത്തെ ഇനം തിരഞ്ഞെടുത്താൽ സൂചിക 0-ഉം രണ്ടാമത്തെ ഇനം തിരഞ്ഞെടുത്താൽ, സൂചിക 1 ഉം ആയി കണക്കാക്കുന്നു. മുകളിലെ വരി തിരഞ്ഞെടുത്ത വേരിയബിളിന്റെ ഇൻഡക്സ് വേരിയബിൾ n ൽ നൽകുന്നു. വിദ്യാർഥിയുടെ പേരും പ്രായവും നൽകാൻ ഉപയോക്താവിനെ അനുവദിക്കുന്ന വെബ് പേജാണ് താഴെ കൊടുത്തിരിക്കുന്നത്. പേരിൽ കുറഞ്ഞത് 5 അക്ഷരങ്ങൾ ഉണ്ടായിരിക്കണം. പ്രായം 15 മുതൽ 20 വരെ ഉള്ള ഒരു സംഖ്യയായിരിക്കണം.

ഉദാഹരണം 6.16: പേരിന്റെയും പ്രായത്തിന്റെയും സാധുത പരിശോധിക്കുന്ന വെബ് പേജ് നിർമാണം

```
<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>
```



ചിത്രം 6.25: ലൂഷിന്റെ ഉപയോഗം വിശദീകരിക്കുന്ന വെബ് പേജ്

HTML കോഡിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.25 ൽ കൊടുത്തിരിക്കുന്നു. ഡാറ്റയുടെ എല്ലാ സാധുതകളും ഇത് പരിശോധിക്കുന്നു. ആദ്യം, പേരിന്റെ ഫീൽഡിൽ ഒരു വില ഉണ്ടോ എന്ന് പരിശോധിക്കുന്നു. പിന്നീട് പേരിന്റെ ദൈർഘ്യം കുറഞ്ഞത് 5 അക്ഷരങ്ങൾ ഉണ്ടോ എന്ന് പരിശോധിക്കുന്നു. ശേഷം വയസ്സ് നൽകിയിട്ടുണ്ടോ എന്ന് പരിശോധിക്കുന്നു. പിന്നീട് നൽകിയ ഡാറ്റ സംഖ്യ ആണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നു. അന്തിമമായി പ്രായപരിധി 15 മുതൽ 20 വരെ ആണോ എന്ന് പരിശോധിക്കുന്നു.

6.9 സ്ക്രിപ്റ്റുകൾ വെബ്പേജിൽ ചേർക്കുന്നതിനുള്ള വഴികൾ (Ways to add scripts to a webpage)

പല രീതിയിൽ HTML കോഡുകൾക്കുള്ളിൽ സ്ക്രിപ്റ്റുകൾ ഉൾപ്പെടുത്താവുന്നതാണ്. മുൻ ഉദാഹരണങ്ങളിൽ നമ്മൾ ജാവാസ്ക്രിപ്റ്റ് വെബ് പേജിന്റെ ഹെഡ് ഭാഗത്താണ് ഉൾപ്പെടുത്തിയിരുന്നത്. ഇതിന് പുറമേ, സ്ക്രിപ്റ്റുകൾ <BODY> ടാഗിലോ അല്ലെങ്കിൽ ഒരു ബാഹ്യ ഫയലിലോ നൽകാവുന്നതാണ്.

ഇനി വെബ്പേജുകളിൽ സ്ക്രിപ്റ്റുകൾ കൂട്ടിച്ചേർക്കുന്നതിനുള്ള വ്യത്യസ്ത രീതികൾ പരിചയപ്പെടാം.

6.9.1 <BODY> ടാഗിന് ഉള്ളിൽ (Inside <BODY>)

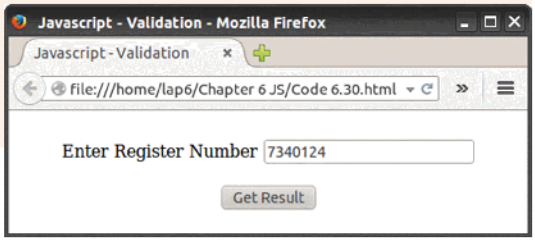
ഈ അധ്യായത്തിന്റെ തുടക്കത്തിൽ <BODY> ടാഗിന് ഉള്ളിൽ സ്ക്രിപ്റ്റുകൾ സ്ഥാപിക്കുന്നത് നാം ചർച്ചചെയ്തു. വെബ്പേജിന്റെ ഉള്ളടക്കം ബ്രൗസറിൽ പ്രദർശിപ്പിക്കുമ്പോൾ സ്ക്രിപ്റ്റുകളും പ്രവർത്തിക്കും. വെബ് പേജ് ഡോക്യുമെന്റിന്റെ ആരംഭം മുതൽക്കാണ് പ്രദർശിപ്പിക്കുക. ഒരു സ്ക്രിപ്റ്റ് കോഡ് കാണുമ്പോൾ അത് പ്രവർത്തിക്കുകയും തുടർന്ന് വെബ് പേജിന്റെ ബാക്കി ഭാഗം പ്രവർത്തിക്കുകയും ചെയ്യുന്നു.

ഒരു ഉദാഹരണം ഉപയോഗിച്ച് ഈ രീതി മനസ്സിലാക്കാം. ഒരു വിദ്യാർഥിയുടെ റിസൾട്ട് ലഭിക്കുന്നതിനുള്ള ഒരു വെബ് പേജാണ് താഴെ കൊടുത്തിരിക്കുന്നത്. ടെക്സ്റ്റ് ബോക്സിൽ ഉപയോക്താവ് ഒരു രജിസ്റ്റർ നമ്പർ നൽകണം. **Get Result** ബട്ടനിൽ ക്ലിക്കു ചെയ്യുമ്പോൾ രജിസ്റ്റർ നമ്പർ ബോക്സിൽ ഏതെങ്കിലും ഡാറ്റ ഉണ്ടോ എന്ന് പരിശോധിക്കാനുള്ള ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷൻ പ്രവർത്തിക്കണം. ടെക്സ്റ്റ് ബോക്സിൽ വില ഉണ്ടെങ്കിൽ അത് ഏഴ് അക്കമുള്ള ഒരു സംഖ്യ തന്നെ ആയിരിക്കണം. ഈ കാര്യങ്ങൾ ഫങ്ഷൻ പരിശോധിക്കണം. വെബ് പേജിന്റെ ഔട്ട്പുട്ട് ചിത്രം 6.26 ൽ നൽകിയിരിക്കുന്നു.

ഉദാഹരണം 6.17: സാധുതാപരിശോധനയ്ക്കു ശേഷം രജിസ്റ്റർ നമ്പർ സ്വീകരിക്കുന്ന വെബ് പേജ് നിർമാണം

```
<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("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>
```



ചിത്രം 6.26: രജിസ്റ്റർ നമ്പർ സ്വീകരിക്കാനുള്ള ഫോം പ്രദർശിപ്പിക്കുന്ന വെബ് പേജ്

സ്ക്രിപ്റ്റ് <BODY> ടാഗിനുള്ളിൽ അവസാനഭാഗത്തും ആകാം. <BODY> ടാഗിൽ അല്ലെങ്കിൽ <HEAD> ടാഗിൽ സ്ക്രിപ്റ്റുകൾ എഴുതിയിട്ടുണ്ടെങ്കിൽ അവ HTML കോഡിനൊപ്പം ബ്രൗസറിലെത്തുകയും ചെയ്യും. വെബ്പേജ് പ്രദർശിപ്പിക്കുന്നത് വൈകുവാൻ ഇത് കാരണമാകുന്നു. സ്ക്രിപ്റ്റുകൾ </BODY> ടാഗിന് മുമ്പായി കൊടുക്കുകയാണെങ്കിൽ വെബ്പേജിലെ ഉള്ളടക്കങ്ങൾ അക്ഷരങ്ങൾ, ചിത്രങ്ങൾ പോലുള്ളവ സ്ക്രീനിൽ വേഗത്തിൽ ദൃശ്യമാകും. എന്നാൽ വെബ് പേജ് ലഭ്യമാക്കുന്നതിനിടയിൽ സ്ക്രിപ്റ്റുകൾ പ്രവർത്തിക്കണമെന്നില്ല.

6.9.2 <HEAD> ടാഗിന് ഉള്ളിൽ (Inside <HEAD>)

സാധാരണ രീതിയിൽ വെബ് പേജിന്റെ ഹെഡ് ഭാഗത്താണ് സ്ക്രിപ്റ്റുകൾ ഉൾപ്പെടുത്തുന്നത്. നാം ചർച്ച ചെയ്ത എല്ലാ ഉദാഹരണങ്ങളിലും അങ്ങനെ തന്നെയാണ് ചെയ്തിരിക്കുന്നത്. ഇതിന് പ്രധാന കാരണം മിക്ക HTML പേജുകളുടെയും ബോഡി ഭാഗത്തിനുള്ളിൽ ഉള്ളടക്കത്തെ സൂചിപ്പിക്കുന്ന ഒരു വലിയ അളവ് ടെക്സ്റ്റ് ദൃശ്യമാക്കേണ്ടതായിട്ടുണ്ട്. ഫങ്ഷൻ നിർവചനം കൂടി ഇവിടെ ചേർക്കുകയാണെങ്കിൽ വെബ്പേജിൽ മാറ്റം വരുത്തുമ്പോൾ ഇത് രൂപകല്പനചെയ്യുന്നയാൾക്ക് ആശയക്കുഴപ്പം സൃഷ്ടിക്കും. ബോഡി ഭാഗത്തിന് മുൻപായി ഒരു വെബ് പേജിന്റെ ഹെഡ് ഭാഗം ലോഡ് ചെയ്യുന്നു എന്നത് ഇതിന്റെ നേട്ടമാണ്. അതിനാൽ, ബോഡി ഭാഗത്ത് വിളിക്കപ്പെടുന്ന ഫങ്ഷനുകളുടെ നിർവചനം നേരത്തെ തന്നെ മെമ്മറിയിൽ ലഭ്യമാക്കിയിരിക്കും. ഉദാഹരണം 6.17 ൽ <SCRIPT> നും </SCRIPT> നും ഇടയിലുള്ള കോഡ് HTML കോഡിന്റെ ഹെഡ് ഭാഗത്തേക്ക് മാറ്റി ഉപയോഗിക്കാവുന്നതാണ്.

6.9.3 ബാഹ്യ ജാവാസ്ക്രിപ്റ്റ് ഫയൽ (External JavaSript file)

HTML ഡോക്യുമെന്റിലെ എല്ലാ സ്ക്രിപ്റ്റുകളും ഒരു ബാഹ്യ ഫയലിലേക്ക് ശേഖരിക്കുകയും ഡോക്യുമെന്റിൽ നിന്ന് ആ ഫയലിലേക്ക് ഒരു ലിങ്ക് സ്ഥാപിക്കുകയും ചെയ്യാം. ഈ ഫയൽ '.js' എന്ന എക്സ്റ്റൻഷനോടുകൂടി സൂക്ഷിക്കുക. ബാഹ്യ ഫയലുകളിൽ ജാവാസ്ക്രിപ്റ്റ് സ്ഥാപിക്കുന്നത് കൊണ്ട് ചില ഗുണങ്ങളുണ്ട്. ഒരേ സ്ക്രിപ്റ്റ് ഒന്നിലധികം HTML പേജുകളിലോ ഒരു മുഴുവൻ വെബ് സൈറ്റിലോ ആവർത്തിക്കപ്പെടുമ്പോൾ ഇത് ഉപയോഗപ്രദമാണ്. ഇത് HTML ഉം ജാവാസ്ക്രിപ്റ്റും വേർതിരിച്ച് കാണാൻ സഹായിക്കുന്നു. ഇത് HTML കോഡിനെയും ജാവാസ്ക്രിപ്റ്റിനെയും വേർതിരിക്കുവാനും ഇവ രണ്ടിനെയും എളുപ്പത്തിൽ വായിക്കാനും നിലനിർത്താനും സഹായിക്കുകയും ചെയ്യുന്നു. ജാവാസ്ക്രിപ്റ്റ് ബാഹ്യ ഫയലുകളിൽ സംഭരിക്കുന്നത് പേജ് വേഗത്തിൽ ലഭ്യമാക്കാൻ സഹായിക്കും.

മുകളിൽ പറഞ്ഞിരിക്കുന്ന ഉദാഹരണം 6.17 ൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് "check.js" എന്ന പേരിൽ ഒരു പ്രത്യേക ഫയൽ ആയി സൂക്ഷിക്കുന്നു. ഈ ഫയലിന്റെ ഉള്ളടക്കം താഴെ കാണിച്ച 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 ഉള്ളൂ എന്നും <SCRIPT> ടാഗ് ഉപയോഗിച്ചിട്ടില്ല എന്നും ശ്രദ്ധിക്കുക. HTML ഫയലിനുള്ളിൽ മാത്രമാണ് <SCRIPT> ടാഗ് ഉപയോഗിക്കുന്നത്. <SCRIPT> ടാഗ് ഉപയോഗിച്ച് ജാവാസ്ക്രിപ്റ്റ് ഫയലിനെ HTML ഫയലുമായി ലിങ്ക് ചെയ്യാവുന്നതാണ്. `Type` ആട്രിബ്യൂട്ട് സൂചിപ്പിക്കുന്നത് ലിങ്കു

ചെയ്തിരിക്കുന്ന ഫയൽ ഒരു ജാവാസ്ക്രിപ്റ്റ് ഫയൽ ആണെന്നതാണ്. Src ആട്രിബ്യൂട്ട് 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 ആട്രിബ്യൂട്ട് ഉണ്ടെങ്കിൽ 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

നിങ്ങളുടെ പുരോഗതി അറിയാം

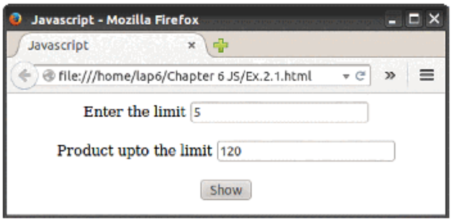
1. ജാവാസ്ക്രിപ്റ്റിൽ onMouseEnter ഇവന്റ് എപ്പോഴാണ് പ്രവർത്തിക്കുക?
2. ഡ്രോപ്ഡൗൺ ലിസ്റ്റിൽ <SELECT> നിന്നും തിരഞ്ഞെടുത്ത ഇനത്തിന്റെ സൂചിക ലഭിക്കാനുള്ള പ്രോപ്പർട്ടി ________ ആണ്.
3. ജാവാസ്ക്രിപ്റ്റിലെ onKeyDown, onKeyUp എന്നിവ താരതമ്യം ചെയ്യുക.
4. Number() ഫങ്ഷൻ ഉപയോഗിക്കുന്നത് എന്തിനാണ്?
5. ജാവാസ്ക്രിപ്റ്റ് കോഡ് </BODY> ടാഗിന് മുൻപായി ഉപയോഗിക്കുന്നതിന്റെ ഗുണം എന്ത്?

നമുക്കു സംഗ്രഹിക്കാം

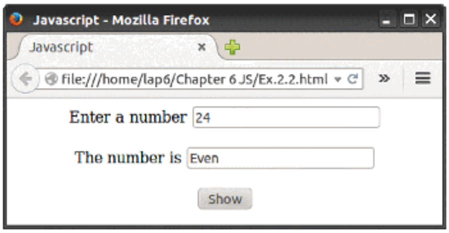
ഡാറ്റയുടെ സാധൂകരണത്തിനായി ഉപയോഗിക്കുന്ന ഒരു ക്ലൈന്റ് ഭാഗം സ്ക്രിപ്റ്റിംഗ് ഭാഷയായിട്ടാണ് ജാവാസ്ക്രിപ്റ്റിനെ ഈ അധ്യായത്തിൽ അവതരിപ്പിക്കുന്നത്. ജാവാ സ്ക്രിപ്റ്റ് കോഡ് HTML ൽ ഉൾപ്പെടുത്തുന്നതിന് ഉപയോഗിക്കുന്ന ടാഗുകളും ജാവാ സ്ക്രിപ്റ്റിലെ പ്രധാന ഫങ്ഷനുകളും ഇവിടെ വിശദീകരിച്ചിരിക്കുന്നു. ജാവാസ്ക്രിപ്റ്റിലെ ഡാറ്റാതരങ്ങളും വേരിയബിളിന്റെ ഉപയോഗവും വിശദമായി ചർച്ചചെയ്യുന്നു. ഓപ്പറേറ്ററുകളുടെയും കൺട്രോൾ സ്ട്രക്ചറുകളുടെയും ഉപയോഗം C++ലേതിന് സമാനമാണ്. വിവിധ അന്തർ നിർമ്മിത ഫങ്ഷനുകളും ജാവാസ്ക്രിപ്റ്റിലെ ഇവന്റുകളും ഉദാഹരണ സഹിതം വിശദീകരിച്ചിരിക്കുന്നു. HTML പേജിൽ ജാവാസ്ക്രിപ്റ്റ് കോഡ് ഉൾപ്പെടുത്തുന്ന വ്യത്യസ്ത രീതികൾ വിശദമായി പ്രതിപാദിച്ചിരിക്കുന്നു.

നമുക്കു പരിശീലിക്കാം

1. താഴെ കാണുന്ന സ്ക്രീൻ പ്രദർശിപ്പിക്കുന്നതിന് ഒരു വെബ് പേജ് നിർമിക്കുക. ഉപയോക്താവിന് ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഒരു നമ്പർ നൽകാം. Show ബട്ടൺ ക്ലിക്കുചെയ്യുമ്പോൾ രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ 1 മുതൽ നൽകിയിട്ടുള്ള പരിധി വരെയുള്ള എല്ലാ സംഖ്യകളുടെയും ഗുണനഫലം പ്രദർശിപ്പിക്കണം.



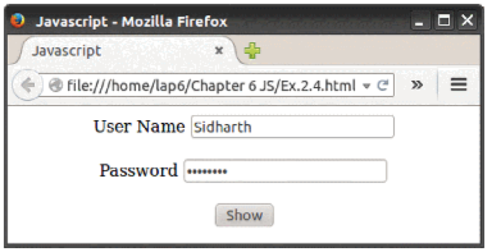
2. താഴെ കാണുന്ന സ്ക്രീൻ പ്രദർശിപ്പിക്കുന്നതിന് ഒരു വെബ് പേജ് നിർമിക്കുക. ഉപയോക്താവിന് ആദ്യത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഒരു നമ്പർ നൽകാം. Show ബട്ടണിൽ ക്ലിക്കുചെയ്യുമ്പോൾ രണ്ടാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഒറ്റ സംഖ്യയോ ഇരട്ട സംഖ്യയോ എന്നതിന് അനുസൃതമായി Odd അല്ലെങ്കിൽ Even എന്ന് പ്രദർശിപ്പിക്കണം.



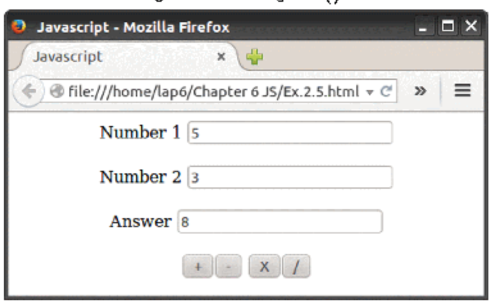
3. താഴെക്കൊടുത്തിരിക്കുന്ന സ്ക്രീൻ പ്രദർശിപ്പിക്കുന്നതിന് ഒരു വെബ് പേജ് തയാറാക്കുക. ഉപയോക്താവിന് ടെക്സ്റ്റ് ബോക്സിൽ ഒരു പ്രായം നൽകാം. ടെക്സ്റ്റ് ബോക്സിൽ ഒരു സംഖ്യക്ക് പകരം ഉപയോക്താവ് അക്ഷരമാണ് നൽകുന്നതെങ്കിൽ, Show ബട്ടണിൽ ക്ലിക്കുചെയ്യുമ്പോൾ 'Invalid Age" എന്ന ഒരു

സന്ദേശം പ്രദർശിപ്പിക്കണം അല്ലെങ്കിൽ 'Correct Data" എന്ന സന്ദേശം പ്രദർശിപ്പിക്കണം.

4. ചിത്രത്തിൽ കാണിച്ചിരിക്കുന്നതുപോലെ ഒരു ലോഗിൻ പേജ് നിർമിക്കുക. പാസ് വേഡ് നൽകുന്നതിനായി പാസ് വേഡ് ടെക്സ്റ്റ് ബോക്സും യൂസർനെയിം നൽകാനായി ഒരു ടെക്സ്റ്റ് ബോക്സും ഉണ്ടായിരിക്കണം. യൂസർനെയിമിൽ കുറഞ്ഞത് 4 ഉം പാസ് വേഡിൽ കുറഞ്ഞത് 6 ഉം ക്യാരക്ടറുകൾ ഉണ്ടായിരിക്കണം. പാസ്‌വേഡിലെ ആദ്യത്തെ രണ്ട് ക്യാരക്ടറുകൾ അക്കങ്ങൾ ആയിരിക്കണം. show ബട്ടണിൽ ക്ലിക്കു ചെയ്യുമ്പോൾ സാധുവായ ഡാറ്റയാണ് ബോക്സുകളിൽ നൽകിയിട്ടുള്ളതെങ്കിൽ 'Correct Data" എന്നും അല്ലെങ്കിൽ 'Wrong Data" എന്നുമുള്ള സന്ദേശം പ്രദർശിപ്പിക്കണം.



5. താഴെ കാണുന്ന ചിത്രത്തിലെ പോലെ ലളിതമായ ഒരു കാൽക്കുലേറ്റർ അടങ്ങിയ ഒരു വെബ് പേജ് നിർമിക്കുക. രണ്ട് സംഖ്യകൾ നൽകാൻ രണ്ടു ടെക്സ്റ്റ് ബോക്സുകൾ ഉണ്ടായിരിക്കണം. സങ്കലനം, വ്യവകലനം, ഗുണനം, ഹരണം എന്നിവയ്ക്കായി 4 ബട്ടണുകൾ ഉണ്ടായിരിക്കണം. ബട്ടനിൽ ക്ലിക്കുചെയ്യുമ്പോൾ മൂന്നാമത്തെ ടെക്സ്റ്റ് ബോക്സിൽ ഉത്തരം കാണിക്കണം.



നമുക്കു വിലയിരുത്താം

1. താഴെപ്പറയുന്നവയിൽ വേരിയബിൾ z എന്ന വേരിയബിളിന്റെ 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 ഔട്ട്പുട്ട് കണ്ടെത്തുക.

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>
```

ടെക്സ്റ്റ് ബോക്സിൻറെ വില n എന്ന വേരിയബിളിൽ സംഭരിക്കുന്നതിനായി വിട്ടഭാഗം പൂരിപ്പിക്കുക.

```
var n;
n = ............................;
```

4. നിങ്ങൾ checkData () എന്ന് പേരിൽ ഒരു ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷൻ എഴുതിയിട്ടുണ്ടെന്ന് കരുതുക. മൗസ് പോയിൻറർ ബട്ടനുമുകളിലൂടെ നീക്കുമ്പോൾ മാത്രമേ ഫങ്ഷൻ പ്രവർത്തിക്കേണ്ടതുള്ളൂ. അതിനുവേണ്ടി താഴെക്കൊടുത്തിരിക്കുന്ന കോഡ് എങ്ങനെ പൂർത്തീകരിക്കാം.

```
<INPUT Type= "button" .......... = "checkData()">
```

5. <SCRIPT> ടാഗും അതിന്റെ ആട്രിബ്യൂട്ടുകളും വിശദമാക്കുക.
6. ജാവാസ്ക്രിപ്റ്റിൽ ഒരു അന്തർനിർമിത ഫങ്ഷൻ നിർവചിക്കുന്നതിനുള്ള വാക്യഘടന എഴുതുക.
7. താഴെക്കൊടുത്തിരിക്കുന്ന വിലകൾ ജാവാസ്ക്രിപ്റ്റിലെ അനുയോജ്യമായ ഡാറ്റാ ഇനങ്ങൾ ഉപയോഗിച്ച് വേർതിരിക്കുക.
   "Welcome", "123", "true", 67.4, .98, false, "hello"
8. ജാവാസ്ക്രിപ്റ്റിൽ undefined ഡാറ്റ ഇനം എന്നതുകൊണ്ട് എന്താണ് അർഥമാക്കുന്നത്?
9. ജാവാസ്ക്രിപ്റ്റിലെ ഓപ്പറേറ്ററുകൾ വിശദീകരിക്കുക.
10. താഴെപ്പറയുന്നവ ചെയ്യുന്നതിനുള്ള ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷനുകളുടെ പേര് എഴുതുക
    a. N എന്ന വേരിയബിളിൽ സംഖ്യ ആണോ എന്ന് പരിശോധിക്കാൻ.
    b. "scert" എന്ന സ്ട്രിംഗ് വലിയ അക്ഷരങ്ങളായി മാറ്റാൻ.
    c. "HTML" എന്ന സ്ട്രിംഗ് ചെറിയ അക്ഷരങ്ങളായി മാറ്റാൻ.
    d. "Welcome to functions" എന്ന സന്ദേശം പ്രദർശിപ്പിക്കാൻ.
    e. "Computer" എന്ന സ്ട്രിംഗിലെ മൂന്നാമത്തെ ക്യാരക്ടർ പ്രദർശിപ്പിക്കാൻ.
11. 'Computer" എന്ന സ്ട്രിങിന്റെ നീളം പ്രദർശിപ്പിക്കുന്നതിനുള്ള ജാവാസ്ക്രിപ്റ്റ് കോഡ് എഴുതുക.
12. ഒരു വെബ് പേജിൽ ഒരു ബട്ടൺ ഉണ്ട്. താഴെപ്പറയുന്ന ഇവന്റുകൾക്കനുസരിച്ച് Message( ) എന്ന ഫങ്ഷൻ പ്രവർത്തിപ്പിക്കാനുള്ള HTML കോഡ് എഴുതുക.
    a. ഉപയോക്താവ് ബട്ടണിൽ മൗസ് ക്ലിക്ക് ചെയ്യുമ്പോൾ.
    b. ഉപയോക്താവ് ബട്ടണിനുമേൽ മൗസ് പോയിന്റർ നീക്കുമ്പോൾ.
13. ഒരു HTML പേജിൽ ഹെഡ് ഭാഗത്ത് ജാവാസ്ക്രിപ്റ്റ് കോഡ് എഴുതുന്നതിന്റെ ഗുണങ്ങൾ ഏവ?
14. ഒരു വിഷയത്തിന്റെ മാർക്ക് നൽകുവാൻ ഒരു ടെക്സ്റ്റ് ബോക്സ് അടങ്ങിയിരിക്കുന്ന ഒരു വെബ് പേജ് രൂപകൽപന ചെയ്യുക.
    a. ഈ വെബ് പേജിനുവേണ്ടിയുള്ള HTML കോഡ് എഴുതുക.
    b. ഈ ടെക്സ്റ്റ് ബോക്സിലെ വിലയുടെ സാധൂകരണത്തിന് ഒരു ബാഹ്യ ജാവാസ്ക്രിപ്റ്റ് ഫയൽ നിർമിച്ച് അത് HTML ഡോക്യുമെന്റുമായി ബന്ധിപ്പിക്കുക. വിലയിരുത്തുന്നതിനുള്ള സൂചകങ്ങൾ
       (i) അത് ശൂന്യമായിരിക്കരുത്
       (ii) അത് ഒരു സംഖ്യയായിരിക്കണം
       (iii) 0-നും 60-നും ഇടയിലായിരിക്കണം മാർക്ക്.
    c. സ്ക്രിപ്റ്റ് ഒരു ബാഹ്യ ഫയൽ ആയി എഴുതുന്നതിന്റെ ഗുണഫലങ്ങൾ എഴുതുക.