

වෙබ් අඩවි සැකසීම Web Designing  
කෘතියෙන් උපුටා ගන්නා ලදී

කතෘ පුද්ගල කවිතා කොටුවේගෙදර  
ISBN 978-955-551-623-5

කතෘගේ ලිඛිත අවසරයක් නොමැතිව වාණිජ  
පරමාර්ථයෙන් මෙය පිටපත් කිරීම තහනම්.

අධ්‍යාපන කටයුතු සඳහා භාවිතා කල හැකිය.

www.kotuwagedara.com දැනුම පමනක් ඇති ඔබේ අඩවියට

ඔබේ මනාපය දෙන්න

JavaScript.

අප විසින් වෙබ් පිටුවක් සැකසීමේදී ඇතැම් ක්‍රියාවලීන් කිරීමේ හැකියාව HTML සතුනොවේ, HTML හි ඇති සීමිතකම් හේතුවෙන් වෙනත් භාෂාවක් HTML සමඟ භාවිතා කිරීමට අපට සිදුවේ, මෙහිසාම JavaScript යන පරිගණක භාෂාව වෙබ් පිටුවක සැකසීමේදී වර්ථමානයේ භාවිතා කරනු ලබයි. මෙමගින් වෙබ් පිටුව භාවිතා කරන පුද්ගලයාගේ අවශ්‍යතා අනුව වෙබ් පිටුව වෙනස් කිරීමටත්, තාර්කිකව ක්‍රියාවලීන් කිරීම හා විශේෂ ක්‍රියාවලීන් රාශියක් කිරීමේ හැකියාව මෙමගින් අපට ලැබේ. Netscape ආයතනය විසින් දියුණු කරනු ලැබූ livescripting යන භාෂාව 1995 වර්ෂයේදී Sun micro systems ආයතනය විසින් ප්‍රතිනිර්මාණය කරනු ලැබූ අතර, මෙම භාෂාවේ නාමය JavaScript. නම් විය. වෙබ් පිටු සැකසීමේදී වර්ථමානයේ අත්‍යාවශ්‍ය භාෂාවක් ලෙස මෙම භාෂාව හැදින්වේ.

JavaScript. යනු Java යන පරිගණක භාෂාව නොවේ. නමුත් එහි ඇති ගති ලක්ෂණ සමහරක් මෙහි දැකගත හැකිවේ. HTML සේ නොව මෙම භාෂාව Case sensitive ය. Case sensitive යනු මෙම භාෂාවේ ඇති විධාන Capital හෝ Simple හෝ ඒ දෙකෙහිම



මිශ්‍රණයක් වේ නම්, එම විධාන එම ආකාරයෙන්ම සටහන් කළ යුතුවේ.

### JavaScript හි භාවිතයන්

- JavaScript භාෂාව ඉතා පහසුවෙන් උගත හැකි භාෂාවක් වේ. HTML මගින් යම් කිසි ක්‍රියාවලියක් තාර්කිකව ක්‍රියාත්මක කිරීමට නොහැකි උවත් මෙම භාෂාව HTML සමඟ භාවිතා කිරීමෙන් එම හැකියාව අපට ආදේශ කොටගත හැකිවේ.
- මෙමගින් යම් කිසි ක්‍රියාවලියක් සිදුකිරීමේදී ඒ අනුව ක්‍රියාත්මක වීමට අවශ්‍ය උපදෙස් ලබාදිය හැකිය. උදාහරණයක් ලෙස වෙබ් පිටුවෙහි ඇති Button එකක් Press කළ විට යම් කිසි ක්‍රියාවලියක් සිදුවීම විය යුතු නම් ඒ සඳහා විධාන පෙලගැස්විය හැකිය.
- වෙබ් පිටුවක සටහන් කරනු ලබන දත්ත Server එක වෙත යැවීමට පෙර එම දත්ත නිවැරදි රූපයෙන් පවතිද යන්න තහවුරු කරගැනීම සඳහා මෙම භාෂාව භාවිතා කළ හැකිය. උදාහරණයක් ලෙස වෙබ් පිටුවක ඊමේල් ලිපිනයක් සටහන් කිරීමට ඇති විටක එම ඊමේල් ලිපිනයෙහි ? ලකුණ ඇත්දැයි දැනගැනීම වැනි ක්‍රියාවලීන් සඳහා මෙම භාෂාවෙහි ඇති විධාන භාවිතා කළ හැකිය.
- මෙම භාෂාව භාවිතයෙන් වෙබ් පිටුවක් භාවිතා කරනු ලබන පුද්ගලයාගේ Browser මෘදුකාංගය කුමක්ද යන්න හඳුනාගෙන ඒ අනුව වෙබ් පිටුවෙහි ඇතැම් කොටස් ස්වාධීනව හැසිරීමට උපදෙස් පෙලගැන්විය හැකිය. (ඇතැම් browser මෘදුකාංග එකම වෙබ් පිටුව නොයෙක් වෙනස් ආකාරයෙන් දර්ශණය කරනු ලබයි. මෙම වෙනස් කම්

www.kotuwegedara.com දැනුම පමනක් ඇති ඔබේ අඩවියට

ඔබේ මනාපය දෙන්න

පලමුව හඳුනාගත හැකිනම් සෑම වෙබ් Browser එකකම එකම ආකාරයෙන් දර්ශණය වීමට අවශ්‍ය ආකාරයෙන් පෙලගැස්වීමක් කළ හැකිය.)

### JavaScript භාෂාව HTML පිටුවකට ඇතුළත් කිරීම

JavaScript භාෂාව HTML සමඟ ආකාර දෙකකින් භාවිතා කළ හැකිය. එනම් HTML පිටුවක HTML සමඟ භාවිතා කිරීම ඉන් පලමුවන ක්‍රමයයි. මෙහිදී <script type="text/javascript"> ලෙස ආරම්භ කොට පසුව JavaScript විධාන පෙලගැස්විය හැකි අතර </script> වලින් JavaScript විධාන පෙලගැස්වීම නතර කළ හැකිය. පහත උදාහරණය පරිදි

```
<script type= "text/javascript">
```

JavaScript විධාන  
JavaScript විධාන

```
</script>
```

අනෙක් ක්‍රමය වනුනේ JavaScript විධාන වෙනමම ගොනුවක සටහන් කොට එම JavaScript විධාන ක්‍රියාත්මක වීම සඳහා HTML ගොනුවෙහි සම්බන්ධතාවයක් ඇති කිරීමයි. මෙහිදී <script යන්නට පසුව src= ලෙස සටහන් කොට ජාවා ගොනුවෙහි නම ලිවිය හැකි අතර </script> යන්න ලියා අවසන් කළ හැකිය. පහත උදාහරණය අවබෝධ කොට ගන්න. පහත උදාහරණයෙන් Test.js යන නමින් සැකසූ JavaScript ගොනුවක් ක්‍රියාත්මක කිරීමට විධාන සටහන් කරන ආකාරය දැක්වේ.

```
<script src= "Test.js"> </script>
```



05

මෙහිදී Test.js නමින් සකසාගෙන ඇති JavaScript ගොනුවෙහි ඇති විධාන ක්‍රියාත්මක වේ.

එකම JavaScript විධාන පෙලක් වෙබ්පිටු කිහිපයක ක්‍රියාත්මක විය යුතු විටක ඒ සඳහා හොඳම විසඳුම වනුයේ JavaScript විධාන ඇතුළත් කොට වෙනමම ගොනුවක තැම්පත් කොට, පසුව එම විධාන ක්‍රියාත්මක විය යුතු විට ඉහත ආකාරයෙන් එම JavaScript ගොනුව ක්‍රියාත්මක වීමට උපදෙස් දීමය. මෙහිදී මෙම JavaScript ඇතුළත් ගොනුව .Js නමින් Extension එක යොදා ගබඩා කල යුතු අතර මෙම ගොනුවෙහි ආරම්භයේ <script> යන්න නොයෙදිය යුතුය.

මේ වනවිට අප සරලව JavaScript HTML සමඟ යොදාගන්නා ආකාරය උගත් අතර, JavaScript භාෂාව HTML සමඟ ක්‍රියාත්මක කරන ආකාරය ප්‍රායෝගිකව හදාරමු. පහත විධාන සටහන් කොට වෙබ් පිටුවක් ආකාරයෙන් ගබඩා කරන්න. මෙහිදී HTML ගොනුවෙහි අගට HTML හෝ HTM යන නම් යොදන්න. ඔබ විසින් ඒ සඳහා FirstJava.html යන නම යොදන්න.

```
<html>
<body>
<script type="text/javascript">
document.write("Hello !")
document.write("This is My first JavaScript Practical")
</script>
</body>
</html>
```

ඉහත උදාහරණයෙහි <script type="text/javascript"> යන්නට පසුව document.write("Hello !") ලෙස සටහන් කොට ඇත්තේ

www.kotuwegedara.com දැනුම පමනක් ඇති ඔබේ අඩවියට

ඔබේ මනාපය දෙන්න

JavaScript විධානයක්ය. මෙම විධානය මගින් වෙබ් පිටුවක යම්කිසි තොරතුරක් දැක්විය හැකිය. ඉහත Hello ! යන්න වෙබ් පිටුවෙහි පෙන්වීම සඳහා භාවිතා කර ඇති document.write("") යන විධානය භාවිතා කර ඇති බව ඔබට වටහා ගත හැකිවේ. JavaScript විධානයක් අවසන් වූ පසුව එම විධානය අවසන් බව දැක්වීමට වෙනත් කිසිදු සංකේතයක් සටහන් කල යුතු නොවේ. නමුත් අපට අවසන් බව දැක්වීමට ; ලකුණ අවසානයේදී සටහන් කල හැකිය.

**විධාන ක්‍රියාත්මක වීම වැලැක්වීම (comment)**

අප විසින් ඇතැම් විධාන සටහන් කල පසුව එම විධාන තාවකාලිකව නැවැත්වීමට ඇතැම් අවස්ථාවලදී අවශ්‍ය වේ. මෙහිදී එක් විධානයක් ක්‍රියාත්මක වීම වැලැක්වීමට මෙම / ලකුණ දෙවකාවක් එම විධානයට පෙර සටහන් කල හැකිය. උදාහරණයක් ලෙස පහත විධානයට පෙර // යොදා ඇති විට මෙම විධානය ක්‍රියාත්මක නොවේ. ඉහත ඔබ සැකසූ FirstJava.html යන ගොනුවෙහි document.write("Hello !") යන්නට පෙර පහත පරිදි සටහන් කර ගබඩා කොට Browser එකෙහි ක්‍රියාත්මක වන ආකාරය අවබෝධකොට ගන්න.

```
//document.write("Hello !")
```

**විධාන කිහිපයක් ක්‍රියාත්මක වීම නැවැත්වීම**

මෙහිදී ක්‍රියාත්මක වීම වැලැක්වීම අවශ්‍ය පලමු විධානයට පෙර / \* ලකුණත් අවසාන විධානයට පසුව \*/ ලකුණත් සටහන් කිරීම මගින් විධාන කිහිපයක් ක්‍රියාත්මක වීම වැලැක්විය හැකිය.

```
<script type="text/javascript">
/* document.write("Hello !")
```



document.write("This is My first JavaScript Practical")\*/

</script>

HTML හි Head කොටසෙහි හෝ Body කොටසෙහි JavaScript ඇතුළත් කල හැකිය. මෙහිදී Body කොටසෙහි ඇතුළත් කල විට පිටුව ආරම්භයේදීම JavaScript ක්‍රියාත්මක වීම ආරම්භ කරන අතර Head කොටසෙහි ඇතුළත් කරනු ලබන JavaScript ක්‍රියාත්මක වනුයේ අප විසින් ක්‍රියාත්මක කිරීමට උපදෙස් ලබාදුන් පසුය. එසේම එකම HTML වෙබ් පිටුවක Head හා Body යන කොටස් දෙකෙහිම JavaScript ඇතුළත් කල හැකිය. පහත උදාහරණය පරිදි

```
<html>
  <head>
    <script type="text/javascript">
      Javascript විධාන
    </script>
  </head>
  <body>
    <script type="text/javascript">
      Javascript විධාන
    </script>
  </body>
</html>
```

### Variable

අප විසින් යම්කිසි දත්තයක් ගබඩාකොට ඒ අනුව යම්කිසි ක්‍රියාවලියක් සිදුකිරීමට අපේක්ෂා කරනු ලබයි නම්, මෙම දත්ත තාවකාලිකව ගබඩාකොට තැබීම සඳහා අපට තාවකාලික ගබඩා ස්ථාන සකසා ගැනීමට සිදුවේ. මෙම තාවකාලික ගබඩා ස්ථාන variable නමින් හඳුන්වනු ලැබේ. මෙහිදී මෙම variable වල

www.kotuwegedara.com දැනුම පමනක් ඇති මගේ අඩවියට

මගේ මනාපය දෙන්න

ගබඩා කරනු ලබන දත්ත ක්‍රියාත්මක වන අවස්ථාවේදී වෙනස් කිරීමට හා එම දත්ත පදනම් කොට ගනිමින් නොයෙක් ක්‍රියාවලීන් සඳහා මෙහෙයවිය හැකිය.

### Variable සකසා ගැනීම

අප විසින් variable එකක් සැකසීම ආරම්භ කරන බව දැන්වීම සඳහා අප අපේක්ෂා කරනු ලබන variable නාමයට පෙර var යන විධානය සටහන් කොට එම variable නාමයට පසුව සමාන කිරීමේ ලකුණ යොදා මෙම variable එකෙහි ගබඩා විය යුත්තේ කුමක්ද යන්න පහත රූපයේ පරිදි සටහන් කල හැකිය.

```
var strname = ගබඩා විය යුතු දත්තය
var strname = "Saman Kumara"
var caltotal = 500
```

එසේම var යන්න නොයොදා පහත රූපයේ පරිදිද variable



05

```
strname = ගබඩා විය යුතු දත්තය
strname = "Saman Kumara"
caltotal = 500
```

අප විසින් සකසනු ලබන variable එකක් අප විසින් සකසනු ලබන ස්ථානය මත ආකාර දෙකකින් ක්‍රියාත්මක වේ. ඉන් පලමු ආකාරය වනුයේ local variables යන්නය. function එකක් තුළ variable එකක් සැකසනු ලබයි නම් (ඉදිරියේදී විස්තරවේ) එම variable එක භාවිතා කල හැක්කේ එම function එක තුළ පමනක් වන අතර වෙනස් ක්‍රියාවලියක් සඳහා මෙම variables එක භාවිතා කල නොහැකි නමුත් function එකකින් පිට variable එකක්

සැකසුවේ නම් මෙම variable එක ඕනෑම ස්ථානයකදී භාවිතා කල හැකිය. මේවා Global variable කියා හඳුන්වයි.

```
<html>
<body>
  var name = "Saman"
  document.write(name)
  document.write("<h1>" + name + "</h1>")
</body>
</html>
```

**තාර්කික ක්‍රියාවලීන් කිරීම**

තාර්කික ක්‍රියාවලීන් කිරීම සඳහා අපට If හා Case යන තර්කනයන් භාවිතා කල හැකිය. සාමාන්‍යයෙන් යම් කිසි තර්කනයකට එකඟවේ නම් යම් කිසි ක්‍රියාවලියක් කරවීමට හා එකඟ නොවේ නම් විය යුතු ක්‍රියාවලිය කුමක්ද ආදී ලෙස තර්කනයන් අප විසින් භාවිතා කරනු ලබයි. මෙහිදී තාර්කනයන් සැකසීම සඳහා භාවිතා කරනු ලබන If හා Case යන තර්කනයන් හි රූපය අවබෝධ කොට ගනිමු.

```
if (condition (තර්කනයනය))
{
  If තර්කනයනයට එකඟවේ නම්
  සිදුවිය යුතු ක්‍රියාවලිය.
}
```

www.kotuwegedara.com දැනුම පමනක් ඇති ඔබේ අඩවියට

ඔබේ මනාපය දෙන්න

අප විසින් තර්කනයක් ගොඩනැගීමේදී If යන විධානය භාවිතා කොට වරහන් වලට පසුව අපගේ තර්කණය සටහන් කොට වරහන් වසා දැමීමෙන් පසුව ඉහත දැක්වෙන ආකාරයෙන් සඟල වරහනකින් පසුව විය යුතු ක්‍රියාවලිය සටහන් කල හැකිය. පසුව සඟල වරහන වසා දැමීමෙන් මෙම තර්කනය අවසන් කල හැකිය. තර්කණයක් ගොඩ නැගීමේදී තාර්කික සංකේත භාවිතා කලයුතු වන අතර පහත සංකේත ඒ සඳහා භාවිතා කල හැකිය.

- == සමාන
- === අගයන් ගෙන් පමනක් නොව දත්ත වර්ගයෙන්ද සමාන
- != අසමාන
- > විශාල
- < කුඩා
- >= විශාල හෝ සමාන
- <= කුඩා හෝ සමාන

පරිගණකයෙහි සටහන් වී ඇති වේලාව හඳුනාගෙන දහවල් 11 ට පෙර වේලාවක් නම් Good morning දර්ශණය කරමු.

```
<script type="text/javascript">
  var d=new Date()
  var time=d.getHours()

  if (time<11)
  {
    document.write("<b>Good morning</b>")
  }
```



</script>

දහවල් 12 ට Lunch-time! යැයි දර්ශනය කරමු.

```
<script type="text/javascript">
  var d=new Date()
  var time=d.getHours()
  if (time==12)
  {
    document.write("<b>Lunch-time!</b>")
  }
</script>
```

if හා else තර්කනය භාවිතය

```
if (condition (තර්කනයනය))
{
  If තර්කනයනයට එකඟවේ නම් සිදුවිය යුතු ක්‍රියාවලිය.
}
else
{
  එකඟවේ නම් සිදුවිය යුතු ක්‍රියාවලිය else යන්නට පසුව
}
```

පරිගණකයෙහි සටහන් වී ඇති වේලාව හඳුනාගෙන දහවල් 11 ට පෙර වේලාවක් නම් Good morning ලෙසත්, එසේ නොවේ නම් Nise day ලෙසත් දක්වමු

```
var d = new Date()
var time = d.getHours()
```

```
if (time < 10)
{
  document.write("Good morning!")
}
else
{
  document.write("Nise day!")
}
```

ඉහත IF හා Else යොදාගනිමින් තර්කයකට එකඟ නම් හෝ නොවේ නම් විය යුතු ක්‍රියාවලිය තීරණය කල අතර තර්කන පදනම්කොට ගනිමින් විය යුතු ක්‍රියාවලීන් බොහොමයක් ඇති විටක පහත Else If යන්න භාවිතා කල හැකිය. Else If භාවිතා කරනු ලබන ආකාරය අවබෝධ කොට ගන්න.

```
if (condition1(පලමු තර්කනයනය))
{
  If තර්කනයනයට එකඟවේ නම් සිදුවිය යුතු ක්‍රියාවලිය.
}
else if (condition1(දෙවන තර්කනයනය))
{
  දෙවන else if තර්කනයනයට එකඟවේ නම් සිදුවිය යුතු ක්‍රියාවලිය.
}
else
{
  පලමු හා දෙවන තර්කනයනයට එකඟවේ නම් සිදුවිය යුතු ක්‍රියාවලිය.
}
```

පහත උදාහරණය මගින් උදෑසන 10 ට අඩු වේලාවන් හිදී Good morning ලෙසත් 10 සිට හතර දක්වා Good Day ලෙසත් ඉහත

www.kotuwegedara.com දැනුම පමනක් ඇති ඔබේ අඩවියට ඔබේ මනාපය දෙන්න



තර්ක දෙකටම එකඟ නොවෙන විට Hello World ලෙසත් දැක්වීම පහත පරිදි කල හැකිය.

```
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time<10)
{
document.write("<b>Good morning</b>")
}
else if (time>10 && time<16)
{
document.write("<b>Good day</b>")
}
else
{
document.write("<b>Hello World!</b>")
}
</script>
```

ඉහත දැක්වූ උදාහරණයෙහි (time>10 && time<16) ලෙස කොටසක් දැකගත හැකිවිය. මෙහිදී වේලාව දහයට වැඩි හා සවස හතරට අඩු වේලාවන් හි දී යන තර්කනය ඇතුළත් වේ. මෙහිදී && යන සංකේතය භාවිතා කොට ඇත්තේ තර්කයන් දෙකක් ඇති අතර මේ දෙකටම එකඟ වීම අනිවාර්යය යන්න දැක්වීමට හා එම තර්කයන් දෙක සම්බන්ධ කිරීමටය. මෙසේ තර්කදෙකක් හෝ ඊට වැඩි සංඛ්‍යාවක් තාර්කික එකතු කිරීම සඳහා පහත පරිදි සංකේත භාවිතා වේ

www.kotuwegedara.com දැනුම පමනක් ඇති ඔබේ අඩවියට

ඔබේ මනාපය දෙන්න

- && (And) තර්කනයන් දෙකම සමාන විය යුතු බව
- || (Or) තර්කනයන් දෙකින් එකක් හෝ සමාන වීම
- ! (Not) දෙකම අසමාන විය යුතු බව

If තර්කනය සමඟ Else යන්න භාවිතා කොට අපට ක්‍රියාවලීන් කිහිපයක් සිදුකර ගැනීමේ හැකියාව ඇතත්, If සමඟ Else යන්න භාවිතා කොට තාර්කික ක්‍රියාවලීන් සිදු කිරීමේ දී ඇතැම් විට එය ලිවීමේදී අපහැදිලිතාවයක් පෙන්වුම් කරනු ලබයි. මෙවැනි අවස්ථාවලදී Case තර්කන ආකෘතිය භාවිතා කිරීම වඩා සරලවේ. පහත දැක්වෙන Case තර්කන ආකෘතිය හඳුනාගන්න

```
switch(n)
{
case 1:
case තර්කනයට එකඟ නම් විය යුතු ක්‍රියාව
break (පහල ඇති විධාන ක්‍රියාත්මක වීම වැලැක්වීම)
case 2:
case තර්කනයට එකඟ නම් විය යුතු ක්‍රියාව
break (පහල ඇති විධාන ක්‍රියාත්මක වීම වැලැක්වීම)
default:
ඉහත case තර්කනයන්ට නොගැලපෙන අවස්ථාවේ සිදුවිය යුතු ක්‍රියාවලිය.
}
```

මෙහිදී ඉහත ආකෘතියේ n සඳහා Variable එකක් ආදේශ කල හැකිය. Case යන විධානයට පසුව එම Variable එකෙහි අගය යම් අගයක් වේ නම් සිදුවිය යුතු ක්‍රියාවලියද break යන විධානය



යොදා ඊට පසුව යෙදෙන case විධානය ක්‍රියාත්මක වීම වැලැක්වීම සිදුකරනු ලබයි. එසේම case යටතේ නොමැති ක්‍රියාවලියක් අගයක් සඳහා යම් කිසි ක්‍රියාවලියක් විය යුතුවනම් එය default: යටතේ පෙලගැස්විය හැකිය.

Case තර්කනයෙහි ක්‍රියාවලිය හඳුනාගැනීමට ඉහත දැක්වෙන උදාහරණය අවබෝධ කොට ගනිමු.

මෙහිදී මෙම උදාහරණය ක්‍රියාත්මක වන අවස්ථාවේදී පරිගණකයෙහි ඇති දිනය හඳුනාගෙන ඒ අනුව විධාන සුභපැතුම් තුනක් ක්‍රියාත්මක වන අතර එම අවස්ථා තුනටම අදාල නොවන

```

<html>
<head>
</head>
<body>
<script type="text/javascript">
var d=new Date()
theDay=d.getDay()

switch (theDay)
{
case 5:
document.write("Finally Friday")
break

case 6:
document.write("Super Saturday")
break

case 0:
document.write("Sleepy Sunday")
break

default:
document.write("I'm looking forward to this weekend!")
}
</script>
</body>
</html>

```

www.kotuwegedara.com දැනුම පමනක් ඇති ඔබේ අඩවියට

ඔබේ මනාපය දෙන්න

දිනයක් උවහොත් වෙනත් සුභපැතුමක්ද ක්‍රියාත්මක වේ. මෙම උදාහරණය ක්‍රියාත්මක කොට Case තර්කනය නිවරදිව හඳුනාගන්න.

Popup Boxes

JavaScript භාවිතා කිරීමේදී වෙබ් පිටුවක් තුළින් යම් කිසි තොරතුරු දැන්වීමටත් වෙබ් පිටුව භාවිතා කරනු ලබන පුද්ගලයාගේ අදහස් විමසීම හා ඔහුගෙන් තොරතුරු ලබාගැනීම වැනි ක්‍රියාවලීන් සඳහා popup boxes වර්ග තුනක් භාවිතා කරනු ලබයි මේවායේ භාවිතයන් උදාහරණ මගින් අවබෝධ කොට ගනිමු.

Alert Box



යම් කිසි තොරතුරක් වෙබ් පිටුවක් නිරීක්ෂණය කරනු ලබන පුද්ගලයා වෙත දැන්වීම සඳහා මෙම AlertBox යන්න භාවිතා කල හැකි අතර ඒ සඳහා පහත පරිදි විධාන පෙලගැස්විය හැකිය.

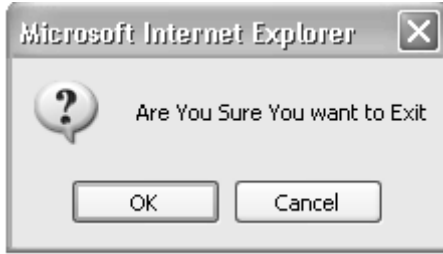


```
alert("දර්ශණය කල යුත්ත")
```

ඉහත රූපයේ දැක්වෙනුයේ alert("Error on It") ලෙස සටහන් කල පසුව දැක්වෙන ආකාරයයි.

Confirm Box

යම් කිසි තොරතුරක් අප වෙතින් ලබාගෙන ඒ අනුව ක්‍රියාත්මක වීම සඳහා මෙම Confirm Box යන්න භාවිතා කල හැකිය. මෙහිදී



අප වෙත Ok හා Cancel ලෙස බොත්තම් දෙකක් ලැබෙන අතර Ok යන්න තෝරාගත් පසුව true යන්නත් Cancel යන්න තෝරාගත් විට false යන්නත් ලබාගත හැකි

අතර ඒ අනුව වෙනස් පියවරක් වෙත යා හැකිවේ. මේ සඳහා පහත පරිදි විධානය සටහන් කල හැකිය.

confirm("දර්ශණය කරනු ලබන විස්තරය " )

ඉහත රූපයේ confirm("Are You Sure You want to Exit") ලෙස සටහන් කල පසුව දර්ශණය වන ආකාරය දැක්වේ.

**Prompt box**



ඇතැම් වෙබ් අඩවිවලදී User ගෙන් නොයෙකුත් ප්‍රශ්න විමසනු ලබන අතර මෙම තොරතුරු අපෙන් ලබාගනුයේ පහත රූපයේ දැක්වෙන

ආකාරයේ කවුලුවකිනි. මෙම කවුලුවෙහි අපට යතුරු පුවරුව මගින් එම කවුලුවෙන් විමසනු ලබන ප්‍රශ්නයට අවශ්‍ය පිලිතුර අපට සටහන් කිරීමට කොටසක් දැකිය හැකිවේ. මෙම කවුලුව Prompt Box ලෙස හඳුන්වනු ලබන අතර Prompt Box එකක් සඳහා අවශ්‍ය විධානය පහත පරිදි දැක්විය හැකිය.

www.kotuwegedara.com දැනුම පමනක් ඇති ඔබේ අඩවියට

ඔබේ මනාපය දෙන්න

```
prompt("ඇසිය යුතු ප්‍රශ්නය","")
<script type= "text/javascript">
    prompt("Enter Your Name","")
</script>
```

මෙසේ Prompt Box එකක් මගින් විමසනු ලබන ප්‍රශ්නයක් වෙත User ලබාදුන් පිලිතුර අනුව ක්‍රියාත්මක වීම සඳහා තාර්කික ප්‍රශ්න ක්‍රියාත්මක හැකිය. ඉහතදී දැක්වෙන උදාහරණයන් හිදී Prompt Box මගින් User සටහන් කරනු ලබන දත්ත පදනම් කොට වෙනස් වූ ක්‍රියාවලීන් සිදුකරනු ලබන ආකාරය විස්තරවේ.

JavaScript වෙබ් පිටුවක් තුළ ඇති JavaScript විධාන එම වෙබ් පිටුව Browser එකක දර්ශණය වන විටම ක්‍රියාත්මක වීම සාමාන්‍යයෙන් සිදුවේ. නමුත් ඇතැම් විධාන ක්‍රියාත්මක වීම අපට අවශ්‍ය විටක පමනක් සිදුවිය යුතුය. මේ සඳහා ඇති හොඳම විකල්පය වනුයේ Function යන පහසුකම් වේ.



**Function එකක් යනු කුමක්ද**

Function එකක් යනු JavaScript එකක් හෝ කිහිපයක් පෙලගස්වා යම් කිසි අවශ්‍යතාවයක් උදෙසා සකසා ගනු ලැබූ කුඩා වැඩසටහනක් ලෙස හඳුනා ගත හැකිය. නමුත් මෙම Function එකක් ක්‍රියාත්මක වීම සඳහා අප විසින් එම Function එක යම් නිෂ්චිත අවස්ථාවක ක්‍රියාත්මක වන ලෙස උපදෙස් ලබාදිය යුතුය. එසේ නොමැතිව ඉබේ ක්‍රියාත්මක වීමක් සිදු නොවේ.

යම්කිසි අවස්ථාවක Function එකක් ක්‍රියාත්මක වීම තීරණය කල හැකි අතර මෙම අවස්ථාව Event එකක් ලෙස හඳුන්වයි.

උදාහරණයක් ලෙස යම් කිසි Button එකක් Click කල පසුව අප විසින් සකසා ඇති Function එකක් ක්‍රියාත්මක විය යුතුයැයි අපට තීරණය කල හැකිය. මෙය Function එකක් Event එකකට අනුව ක්‍රියාත්මක කිරීමක් වේ. Function එකක් සාමාන්‍යයෙන් සකසනුයේ <Head> tag එක තුළ වන අතර එය ඕනෑම විටක ක්‍රියාත්මක කල හැකිය. පහත දැක්වෙන උදාහරණය අවබෝධ කොට ගන්න මෙහි **displaymessage** නමින් Function එකක් සකසා ඇති අතර වෙබ් පිටුවෙහි ඇති එක් Button එකක් Click කල පසුව ඉහත Function එක ක්‍රියාත්මක වේ.

```

<html>
<head>
<script type="text/javascript">
function displaymessage()
    {
        alert("Hello World!")
    }
</script>
</head>
<body>
<form>
<input type="button" value="Click me!"
onclick="displaymessage()" >
</form>
</body>
</html>

```

ඉහත උදාහරණයෙහි Button එකක් click කල පසුව අප විසින් සකසා ගනු ලැබූ Function එකක් ක්‍රියාත්මක වීම සඳහා උපදෙස් පෙලගස්වා ඇත. Function එකක් සැකසීමේදී Function එකෙහි

www.kotuwegedara.com දැනුම පමනක් ඇති ඔබේ අඩවියට ඔබේ මනාපය දෙන්න

රූපය හා එහි ඇතුළත් විය හැකි කොටස් පිලිබදව මිලගට අවධානය යොමු කරමු.

**function function** නාමය (*variable 1 ,variable 2,.....*)

```

{
Function එක තුළ අන්තර්ගත විධාන කොටස්
}

```

අප Function එකක් සැකසීමට යාමේදී පලමුව Function යන වචනයද පසුව ඒ සඳහා භාවිතා කරනු ලබන නාමයද පසුව වරහන් තුළ Function එකක භාවිතා වන තාවකාලික මතක ස්ථාන variable වල නම් සටහන් කිරීමෙන් පසුව සගල වරහන් තුළ අපගේ Function එකෙහි ක්‍රියාකාරීත්වයට අදාල විධාන පෙලගැස්විය හැකිය. යම් ලෙසකින් Variable කිසිවක් අප විසින් භාවිතා කරනු නොලැබේ. නමුත් Function නාමයට පසුව වරහන් ලකුණ ආරම්භකොට එය වසා දැමිය යුතුවේ.

මෙහිදී function එකෙහි නාමය අනිවාර්යයෙන්ම Simple අක්ෂර වලින් පමනක් සමන් විය යුතුවේ.

