

CSS (Cascading Style Sheet)

මෙය Cascading Style Sheet ලෙස හඳුන්වයි. මෙමගින් html අංග වෙබ් අතරික්සුව තුළ පෙන්නුම් කළ යුතු ආකාරය අර්ථ දැක්වයි. CSS මගින් වෙබ් පිටු විවිධ ආකාරවලට හැඩගැන්වීමට ඇති හැකියාව ලබා දේ.

CSS භාවිතයේ වාසි

- වරක් ලියූ CSS ගොනුවක් html පිටු තුළ නැවත නැවත භාවිතා කළ හැකිවීම නිසා කාලය ඉතිරි කරයි.
- අඩු කේත ප්‍රමාණයක් භාවිතා කිරීම මගින් වෙබ් පිටු බාගත වීම වඩා වේගවත් වීම.
- CSS ගොනුවේ වෙනසක් සිදු කිරීම මගින් වෙබ් අඩවියේ සියලු වෙබ් පිටු ස්වයංක්‍රීයව යාවත්කාලීන කිරීමේ හැකියාව නිසා පහසුවෙන් නඩත්තු කළ හැකි වීම.
- පරිගණකය, සුහුරු දුරකථනය, tab පරිගණකය වැනි එකිනෙකට වෙනස් උපකරණවලට ගැලපෙන ලෙස html ලේඛන හැසිරවීමට පහසුකම් සැකසීම (Responsiveness)

විලාස පත්‍ර (CSS) වෙබ් පිටු සමග සම්බන්ධ කළහැකි ආකාර,

විලාස පත්‍ර වෙබ් පිටු සමග සම්බන්ධ කළහැකි ප්‍රධාන ආකාර 03 කි.

- 1) ජේලිගත විලාස පත්‍ර (Inline Style Sheet)
- 2) අභ්‍යන්තර විලාස පත්‍ර (Internal Style Sheet)
- 3) බාහිර විලාස පත්‍ර (External Style Sheet)

1) ජේලිගත විලාසපත්‍ර

මෙහි දී වෙබ් පිටුවේ බඳ කොටස තුළ CSS ගුණාංග ඇතුළත් කළයුතු උසුළන තුළට වෙන් වෙන්ව CSS ගුණාංග ඇතුළත් කිරීම සිදුකළ යුතු ය. මෙහි දී අදාළ උසුළන තුළ Style නැමැති ගුණාංගය යොදා ඉන් අනතුරුව CSS ගුණාංගයන් සහ ඊට අදාළ අගයන් ලබාදිය යුතු ය.

```

<!DOCTYPE html>
<html>

<body>
<h1 style="color:blue;">This is a heading.</h1>
<h1 style="color:red;margin-left:50px;">This is a heading.</h1>
<h1>This is a heading2.</h1>
<p>This is the first paragraph.</p>
<p style="color:green;">This is the second paragraph.</p>
</body>

</html>

```

2) අභ්‍යන්තර විලාසපත්‍ර

මෙහි දී CSS ගුණාංගය ඇතුළත් කළ යුතු උසුලන වෙබ් පිටුවේ ශීර්ෂ කොටසෙහි විස්තර කරයි. මෙහි දී එම උසුලන සඳහා අවශ්‍ය ගුණාංගයන් ශීර්ෂ කොටසෙහි <Style> </Style> උසුලන අතර ලියා දැක්විය යුතු ය.

```
<!DOCTYPE html>
<html>
<head>

<style>
body {background-color: linen;}
h1 {color: maroon;margin-left: 40px}
p{font-family:Arial;}
</style>

</head>
<body>

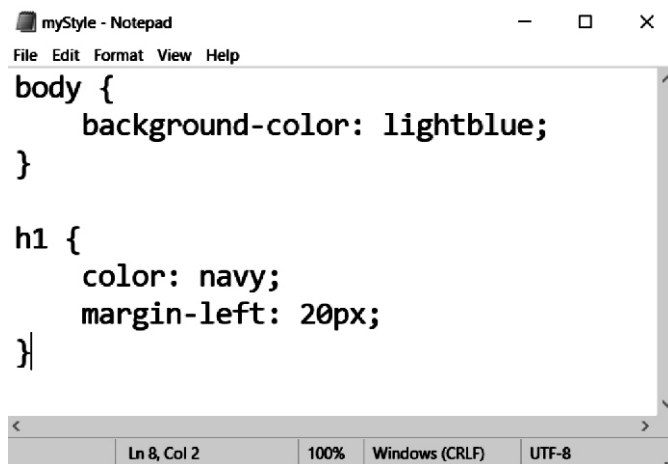
<h1>This is a heading</h1>
<h1>This is a heading2</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

3) බාහිර විලාසපත්‍ර

මෙහි දී වෙබ් පිටු වෙත සම්බන්ධ කළ යුතු CSS ගුණාංගයන් වෙනම ගොනුවක ඇතුළත් කර එය CSS දිගුව සමඟින් සුරකීමෙන් පසු අවශ්‍ය වෙබ් පිටු සමඟ සම්බන්ධතාවය ඇති කිරීම සිදු කරනු ලබයි.

පහත දැක්වෙන්නේ mystyle.css ගොනුව වෙබ් පිටුවක් වෙත සම්බන්ධ කරන ආකාරය යි.



```
myStyle - Notepad
File Edit Format View Help
body {
    background-color: lightblue;
}
h1 {
    color: navy;
    margin-left: 20px;
}
Ln 8, Col 2    100%    Windows (CRLF)    UTF-8
```

```

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>

```

Cascading Order (විලාසපත්‍ර ප්‍රමුඛතාවය)

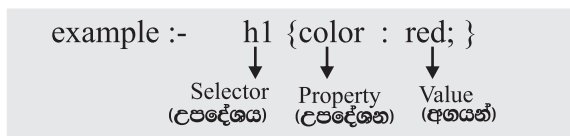
වෙබ් පිටුවක් තුළට සම්බන්ධ කරනු ලබන CSS ගුණාංගයන් බලපාන ආකාරයේ ප්‍රමුඛතාවයක් පවතී. එය Cascading Order (විලාස පත්‍ර ප්‍රමුඛතාවය) ලෙස හඳුන්වන අතර ප්‍රමුඛතාවය වැඩි සිට අඩු දක්වා පහත පරිදි පෙළගැස්විය හැකි ය.

- 01) ජේලිගත විලාස පත්‍ර
- 02) අභ්‍යන්තර විලාස පත්‍ර
- 03) බාහිර විලාස පත්‍ර
- 04) වෙබ් අතරික්සුවේ පෙරනිමිය (Browser Default)

CSS කාරක රීති (CSS Syntaxes)

■ CSS මූලිකාංගයක් Selector, Property සහ Value ලෙස ප්‍රධාන කොටස් 3 කින් සන්විත වේ.

```
Selector { Property : Value ; }
```



■ කිසියම් CSS ගුණාංගයකට අදාළ අගයන් ලබාදීමේ දී එම අගය සඳහා වචන කිහිපයක් ඇතිනම් එම අගය ද්විත්ව උද්ධෘත තුළ ඇතුළත් කළ යුතු ය.

```
h1 {font, family : "Times New Roman";}
```

■ CSS හි selector සඳහා ගුණාංග කිහිපයක් පවතියි නම් එම ගුණාංග සහ අගයන් යුගලය " ; " මගින් වෙන්කර දැක්විය යුතු ය.

```

h1 { color : red ; margin-left : 20px ; }
හෝ
h1 { color : red ;
margin-left : 20px ;
}

```

■ CSS හි Selectors භාවිතා කරනු ලබන විධිමත් ක්‍රමවේද අනුගමනය කළ යුතුය.

CSS Selector භාවිතා කරනු ලබන විධිමත් ක්‍රමවේද

1) element selector

මෙහි දී selector ලෙස html උසුලනයේ මූලිකාංගය භාවිතා කරන අතර එම උසුලනය භාවිතාකරන සෑම අවස්ථාවකදී ම අදාළ CSS ගුණාංගයන් වෙබ් පිටුවේ යෙදීම සිදු වේ.

```
<!DOCTYPE html>
<html>
<head>
<style>
  p{text-align: center;color: red;}
</style>
</head>
<body>

<h1>Every paragraph will be affected by the style.</h1>
<p>Me too!</p>
<p>And me!</p>

</body>
</html>
```

2) Class Selector for One attributes

මෙහි දී CSS ගුණාංගයන් අර්ථ දැක්වීමෙන් පසු class නැමැති ගුණාංගය භාවිතා කරන අවස්ථාවලදී පමණක් CSS ගුණාංගයන් ක්‍රියාත්මක වීම සිදු වේ. එහෙත් එය අර්ථ දැක්වන ලද උසුලනයට පමණක් යොදාගත යුතුය.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p>This paragraph will be red and center-aligned too.</p>

</body>
</html>
```

3) Class selector

මෙහි දී අදාළ Class එක භාවිතා කරනු ලබන සෑම උපුලනයක් සමඟ ම CSS ගුණාංගයක් ක්‍රියාත්මක වීම සිදු වේ. ඕනෑම උපුලනයකට සම්බන්ධ කළ හැකිය.

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>
<p>Another Red and center-aligned paragraph.</p>
</body>
</html>
```

4) Grouping selector

මෙහි දී එකම CSS ගුණාංගයන් html මූලිකාංග කිහිපයකට එකවර භාවිතා කිරීම සිදුකළ හැකි ය.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1,h2,p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>

</body>
</html>
```

5) More Classes with one selector

මෙහි දී එකම මූලිකාංගයක් සඳහා Classes කිහිපයක් භාවිතාකරන අතර අදාළ මූලිකාංගය සමඟ පමණක් එම Classes අවශ්‍ය පරිදි භාවිතා කිරීම සිදුකළ හැකි ය.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {text-align: center;color: red;}
p.large {font-size: 300%;}
</style>
</head>
<body>

<h1 class="center">Heading 1</h1>
<p class="center">Paragraph 1</p>
<p class="center large">Paragraph 1</p>

</body>
</html>
```

6) Id selector

මෙහි දී # ලකුණ සමඟ id එකක් හඳුන්වාදීම සිදු කරයි. අදාළ CSS ගුණාංගයන් දැක්වීම සිදු කරයි. කිසියම් මූලිකාංගයක් සඳහා එම CSS ගුණාංගයන් භාවිතාකළ යුතු නම් අදාළ html මූලිකාංගයට Id නැමැති ගුණාංගය භාවිතා කරමින් එහි අගය ලෙස CSS id සඳහා ලබා දී ඇති වචනය ලබා දිය යුතු ය.

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1 id="para1">Hello World!</h1>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```

CSS විවරණයන භාවිතය (CSS Comments)

CSS කේත කොටස් විස්තර කිරීම සඳහා විවරණ භාවිතා කරනු ලැබේ. මෙම විවරණ එක් පේළියක් තුළ මෙන් ම පේළි කිහිපයක් තුළ වුව ද භාවිතා කළ හැකි ය.

විවරණ ඇතුළත් කිරීම සඳහා "/*" භාවිතා කරන අතර ඉන්පසු විවරණය සඳහන් කර "*/" මගින් අවසන් කරයි.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    color: red;
    /* This is a single-line comment */
    text-align: center;
}

/* This is
a multi-line
comment */
</style>
</head>
<body>

<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
<p>CSS comments are not shown in the output.</p>

</body>
</html>
```

CSS ගුණාංග

01) Color

මෙමගින් අක්ෂරවල වර්ණය හෝ අක්ෂරවල පසුබිම් වර්ණය අවශ්‍ය පරිදි වර්ණ ගැන්වීම සිදු කරනු ලැබේ. මේ සඳහා Color හෝ background - color නැමැති css ගුණාංග භාවිතා කරයි.

```
Ex :- h1 {color : red; }
      h1 {color : # FF 0000 ; }
      h1 {color : rgb (255,0,0); }

      h1 {background - color : red ; }
      h1 {background - color : FF0000; }
      h1 {background - color : rgb (255,0,0) ; }
```

02) Text

අක්ෂර සඳහා අවශ්‍ය හැඩ සෙවීම සිදුකිරීමට භාවිතා කරනු ලබන ගුණාංග කිහිපයක් වේ.

1) color

අක්ෂර අවශ්‍ය පරිදි වර්ණ ගැන්වීමට මෙම ගුණාංගය භාවිතා කරයි.

```
<style>
body {color : blue;}
h2 {color : green;}
</style >
```

2) letter-spacing

මෙමගින් අක්ෂර අතර පරතරය අවශ්‍ය පරිදි තීරණය කළ හැක.

```
< style >
h1 { letter-spacing : 3px ; }
h2 {letter-spacing : 3px ; }
</style >
```

3) line-height

මෙමගින් ඡේදයක පේළි අතර පරතරය අවශ්‍ය පරිදි සකසාගැනීම සිදුකළ හැක.

```
< style >
P. small {line-height : 0.7;}
P.big {line-height : 1.8;}
</style >
```

4) text-align

මෙමගින් අක්ෂර වෙබ් පිටුවේ අවශ්‍ය දිශාවකට එකෙල්ල කරගැනීම සිදුකරගත හැකිය. P මූලිකාංගය සඳහා පමණක් අමතරව Justify නැමැති අගය ලබාදිය හැකිය.

```
< style >
h1 {text-align : center ;}
h2 {text-align : left ;}
h3 {text-align : right ;}
</style >
< style >
p.a {text-align : center ;}
p.b {text-align : left ;}
p.c {text-align : right ;}
p.d {text-align : justify ;}
</style >
```

5) text-decoration

මෙමගින් අක්ෂරවල ඉහළින් ඉරක් හෝ අක්ෂර හරහා ඉරක් හෝ අක්ෂරවලට පහළින් ඉරක් ඇතිකරනු ලැබේ.

```
< style >
h1 {text-decoration : overline ;}
h2 { text-decoration : line - through ; }
h3 {text-decoration : underline ;}
</style >
```

6) text-indent

මෙමගින් අක්ෂර සඳහා අවශ්‍ය ප්‍රමාණයෙන් අනුඡේදනයන් ලබාදිය හැකි ය.

```
< style >
P {text - indent : 50px ; }
</Style >
```


7) text-transform

මෙමගින් එකවර අක්ෂර සියල්ල කැපිටල් හෝ සිම්පල් හෝ සැම වචනයක ම මූල අකුර පමණක් කැපිටල් ලෙස සකසා ගැනීම සිදුකළ හැකි ය.

```
< style >
P. uppercase {text-transform : uppercase }
P. lowercase {text-transform : lowercase; }
P. capitalize {text-transform : capitalize; }
</style>
```

8) word-spacing

මෙමගින් වචන අතර පරතරය අවශ්‍ය පරිදි සකසාගැනීම සිදුකළ හැකි ය.

```
<style>
h1 {word-spacing : 10px ; }
h2 {word-spacing : -5px; }
</Style>
```

03) Font

1) font-family

මෙමගින් අවශ්‍ය අක්ෂර වර්ගය තීරණය කරනු ලැබේ.

```
<style>
P. serif {font-family : "Times New Roman" ; }
P. sansserif {font-family : Arial ; }
</style>
```

2) font-size

මෙමගින් අක්ෂරවල ප්‍රමාණය අවශ්‍ය පරිදි තීරණය කළ හැකි ය.

```
<style>
h1 {font-size : 40pt ; }
h2 {font-size : 30pt ; }
P {font-size : 14pt ; }
</ style >
```

3) font-style

මෙමගින් අක්ෂරවල ඇලගැන්වීම සිදුකරනු ලබයි.

```
<style>
P. normal {font-style : normal ; }
P. italic { font-style : italic ; }
P. ob {font-style : oblique ; }
</style>
```

4) font-weight

අක්ෂර තද පැහැගැන්වීම සඳහා මෙය භාවිත කරනු ලැබේ.

```
<style>
P. normal {font-weight : normal ;}
P. light {font-weight : lighter ;}
P. thick {font-weight : bold ;}
P. thicker {font-weight : 900 ;}
</style>
```

04) List

CSS ගුණාංගයක් භාවිතා කිරීමෙන් ලැයිස්තු අවශ්‍ය පරිදි සකසාගැනීම සිදු කළ හැකි ය. මෙහි දී විශේෂයෙන් භාවිතා කරනු ලබන ගුණාංගය වන්නේ list - style - type වේ. මෙය ul, ol, li යන selectors සමඟ භාවිතා කළ හැකි ය.

```
< Style >
ul. a {list-style-type : circle ; }
ul. b {list-styl-type : square ; }
ol. c {list-style-type : upper-roman ; }
ol. d {list-style : lower-alpha ; }
</style >
```

අංක රහිත ලැයිස්තුවක් පින්තූර භාවිතා කරමින් පෙළගැසවීම සිදුකළ හැකි ය.

```
ul {list- style-image : url ('sqpurple.gif `') ; }
li {list- style-image : url ('sqpurple.gif ') ; }
```

05) Hyper Links

html ලේඛනයක පවතින අධිසන්ධාන සඳහා එහි ක්‍රියාකාරීත්වය වෙනුවෙන් CSS ගුණාංගයන් භාවිතාකළ හැක. භාවිතාකරනු ලබන CSS භාවිතයන් කිහිපයක් පහත පරිදි වේ.

```
< style >
a : link { colour : blue ; }
a : visited { color : green ; }
a : hover {color : hotpink ; }
a : active {color : red ; }
</style >
```

6) Margin

විශේෂයෙන් ඡේදයකට අදාළව හැඩසෙවීම එක්කිරීම සඳහා පහත දැක්වෙන CSS ගුණාංගයන් භාවිතාකළ හැකි ය.

```
P. ex {
border : 1px ;
border - style : solid ;
border - color : red ;
margin - top : 100px ;
margin - bottom : 100px
margin - right : 150px ;
margin - left : 80px ;
}
```

```
P. ex {
border : 1px solid red ;
margin - top : 100px ;
}
```

7. <div> භාවිතය

වර්තමානයේ වෙබ් පිටුවක් තුළ අවශ්‍ය අන්තර්ගතයන් විධිමත්ව පෙළගැස්වීම සඳහා <table> උසුලනය වෙනුවට <div> උසුලනය භාවිතා කරනු ලැබේ.

This is a Table

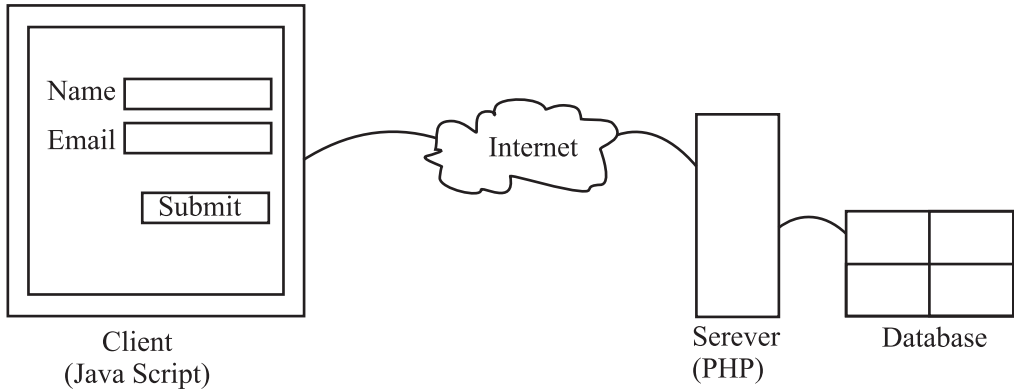
Heading 1	Heading 2	Heading 3
Row 1 Column 1	Row 1 Column 2	Row 1 Column 3
Row 2 Column 1	Row 2 Column 2	Row 2 Column 3

```
<html>
<head>
<style type="text/css">
  .Table
  {
    display: table;
  }
  .Title
  {
    display: table-caption;
    text-align: center;
    font-weight: bold;
    font-size: larger;
  }
  .Heading
  {
    display: table-row;
    font-weight: bold;
    text-align: center;
  }
  .Row
  {
    display: table-row;
  }
  .Cell
  {
    display: table-cell;
    border: solid;
    border-width: thin;
    padding-left: 5px;
    padding-right: 5px;
  }
}
```

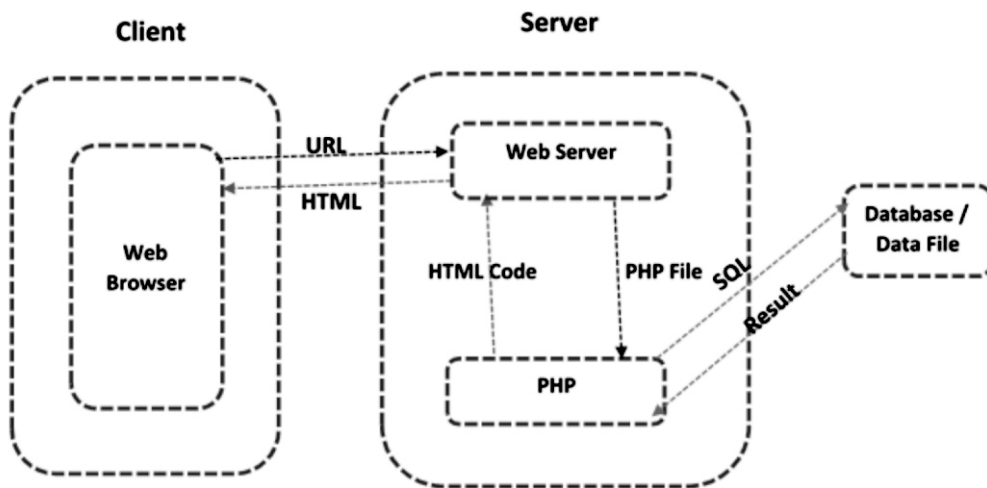
```
</style>
</head>
<body>
<div class="Table">
  <div class="Title">
    <p>This is a Table</p>
  </div>
  <div class="Heading">
    <div class="Cell">
      <p>Heading 1</p>
    </div>
    <div class="Cell">
      <p>Heading 2</p>
    </div>
    <div class="Cell">
      <p>Heading 3</p>
    </div>
  </div>
  <div class="Row">
    <div class="Cell">
      <p>Row 1 Column 1</p>
    </div>
    <div class="Cell">
      <p>Row 1 Column 2</p>
    </div>
    <div class="Cell">
      <p>Row 1 Column 3</p>
    </div>
  </div>
  <div class="Row">
    <div class="Cell">
      <p>Row 2 Column 1</p>
    </div>
    <div class="Cell">
      <p>Row 2 Column 2</p>
    </div>
    <div class="Cell">
      <p>Row 2 Column 3</p>
    </div>
  </div>
</div>
</body>
</html>
```

....PHP....

PHP, Hypertext Preprocessor යන සම්මතයෙන් දක්වනු ලැබේ. මෙය විවෘත මූලාශ්‍ර Server Site විස්තර කිරීමේ භාෂාවක් වන අතර web යෙදවුම් සංවර්ධනය සඳහා භාවිතා කරනු ලබයි. මෙම යෙදුම් සේවලාභී සේවාදායක නිර්මිතය මත පදනම්ව ගොඩනැගී ඇත.



මෙම නිර්මිතය මත පදනම් වූ යෙදුම් පද්ධතියක් සේවාදායක යෙදුම් මෘදුකාංගයකින් සේවලාභී යෙදුම් මෘදුකාංගයකින් සහ ඔවුන් අතර දත්ත සන්නිවේදනය කිරීම සඳහා වූ සන්නිවේදන මාධ්‍යයකින් සමන්විත වේ.



PHP කේතයන් html ලේඛනයක් තුළට ඇතුළත් කළ හැකි ය. එබැවින් එය HTML - embedded Web Scripting language ලෙස හඳුන්වයි. මෙම PHP භාෂාව විවිධ මෙහෙයුම් පද්ධති සමඟ සහය දක්වන අතර වර්තමානයේ පවතින බොහෝ වෙබ් සේවාදායක (Apache, IIS, Tomcat) සමඟ සහයෝගයෙන් ක්‍රියාකරයි.

බොහෝවිට PHP භාවිතා කරනු ලබන්නේ ගතික වෙබ් අඩවි සහ යෙදුම් නිර්මාණය කිරීම සඳහා වේ.

PHP ගොනුවක් හැඳින්වීම

- ◆ PHP ගොනුවක් html, CSS, Java Script සහ php කේත මගින් සමන්විත විය හැකි ය.
- ◆ PHP ගොනුවක දිගුව ".php" ලෙස ඇතුළත් කළ යුතු ය. PHP ගොනුවක් වෙබ් සේවාදායක පමණක් ක්‍රියාත්මක වන අතර එහි ප්‍රතිඵලය සරල html ලේඛනයක් ලෙස වෙබ් අතරික්සුවට ලබාදෙනු ලැබේ.

PHP මූලික කාරක රීති

- ◆ PHP කේතයක් ලේඛනයේ ඕනෑම ස්ථානයක ඇතුළත් කළ හැකිය.
- ◆ PHP කේතයක් ආරම්භ කිරීම `<?php` මගින් සිදුවන අතර අවසන් කිරීම `?>` මගින් සිදුකරනු ලබයි.

```
<? php
    // PHP code goes here
?>
```

- ◆ PHP ප්‍රකාශනයක් ";" මගින් අවසාන කළ යුතු ය.
- ◆ PHP සියළුම විචල්‍යයන් අක්‍ෂර සංවේදී වන අතර අනෙකුත් සියලුම මූලපද අක්‍ෂර සංවේදී නොවේ.

PHP කේතයක් html ලේඛනයක් තුළට ඇතුළත් කිරීම.

php කේතයක් html ලේඛනයකට ඇතුළත් කරන විට එම කේතයන් `<?php` සහ `?>` අතර ඇතුළත් කිරීම සිදුකළ යුතු ය. ඉන් අනතුරුව එම php කේත අඩංගු ලේඛනය .php දිගුව සමගින් සුරැකීම සිදුකළ යුතු ය.

```
<!DOCTYPE html>
<html>
<head>
<title>PHP</title>
</head>

<body>
<h1>My first PHP page</h1>

<?php
    echo "Hello ICT Students!";
?>

</body>
</html>
```

PHP විවරණ

php හි කේත කොටස් විස්තර කිරීම සඳහා විවරණ (comments) භාවිතා කරනු ලැබේ. මෙම විවරණ එක් පේළියක් තුළ මෙන් ම පේළි කිහිපයක් තුළ අවශ්‍ය නම් ප්‍රකාශනයක් අතරමැදි වුවද ක්‍රියාත්මක කළ හැකි ය.

```

<!DOCTYPE html>
<html>
<body>

<?php
    // This is a single-line comment

    # This is also a single-line comment

    /* This is a multiple-lines comment block
    that spans over multiple lines */

    // You can also use comments to leave out parts of a code line
    $x = 5 /* + 15 */ + 5;
    echo $x; ?>

</body>
</html>

```

PHP හඳුන්වන (PHP Identifiers)

PHP විචල්‍යයක් හෝ වැඩසටහන් නාමයක් php හඳුන්වන ලෙස හඳුන්වනු ලැබේ. මෙම හඳුන්වන නම් කිරීමේ දී අනුගමනය කළ යුතු නීති කිහිපයක් වේ.

- ◆ හඳුන්වනයක් ආරම්භ කළ යුත්තේ අක්ෂරයකින් හෝ යටිඉර මගින් පමණි.
- ◆ හඳුන්වන සඳහා භාවිතාකළ හැක්කේ අකුරු ඉලක්කම හෝ යටි ඉර පමණි.
- ◆ හඳුන්වනයක පිහිටි වචන අතර හිස් ඉඩ නොයෙදිය යුතු ය.
- ◆ හඳුන්වන අක්ෂර සංවේදී වේ.
\$age, \$AGE, \$Age (මෙම හඳුන්වන 3 එක්කොට වෙනස් විචල්‍යය 03 කි)
- ◆ හඳුන්වන ඕනෑම දිගකින් සමන්විත විය හැක.
- ◆ PHP හි පෙර අර්ථ දැක්වන ලද මූල පදයන්ට සමාන නොවිය යුතුය.

වලංගු හඳුන්වන

\$sum, \$Grade, \$Total, \$main - Counter, \$_discount

වලංගු නොවන හඳුන්වන

\$Gross Amount, \$#Address, \$paid-Amount,\$1number

PHP විචල්‍ය

ක්‍රමලේඛනයක ක්‍රියාත්මක වන විට හඳුන්වනයකට පවරන ලද අගයයන් වෙනස්වන හඳුන්වන විචල්‍ය ලෙස හඳුන්වනු ලැබේ.

```

$discount = 10;
$subject _ name = "Information and Communication Technology";
$average = 45.86;

```

PHP විචල්‍යයක පරාසය Scope of PHP Variable

PHP විචල්‍යයක් PHP ලේඛනයක ඕනෑම ස්ථානයක ඇතුළත් කළ හැකිය. මෙසේ ඇතුළත් කළ විචල්‍ය ලේඛනයේ බලපවත්වන සීමාව එම විචල්‍යයේ පරාසය ලෙස දක්වනු ලැබේ. ඒ අනුව PHP විචල්‍යයක් ආකාර 03 කින් හඳුනාගත හැකිය.

- ◆ local (ස්ථානීය විචල්‍යය)
- ◆ global (විශ්වීය විචල්‍යය)
- ◆ static (ස්ථිතික විචල්‍යය)

Local Variable

PHP කාර්යබද්ධ ක්‍රියා පටිපාටියක් හෙවත් ශ්‍රිතයක් තුළ අර්ථ දක්වනු ලබන විචල්‍ය එම ශ්‍රිතය තුළ පමණක් භාවිතා කළ හැකි අතර එවැනි විචල්‍ය ස්ථානීය විචල්‍ය ලෙස හඳුන්වයි.

```
<!DOCTYPE html>
<html>
<body>

<?php
function LocalTest() {
    $x = 5; // local scope
}

LocalTest ();

echo $x; // using x outside the function will generate an error
?>

</body>
</html>
```

x නැමැති ස්ථානීය විචල්‍ය ශ්‍රිතයෙන් පරිහානිරව භාවිතා කිරීම නිසා දෝෂ පණිවුඩයක් ජනනය වේ.

```
<!DOCTYPE html>
<html>
<body>

<?php
function LocalTest () {
    $x = 5; // local scope
    echo $x;
}

LocalTest ();
?>

</body>
</html>
```

මෙහි එම දෝෂ පණිවුඩය ජනනය නොවේ.

Global Variable

ශ්‍රිතයකින් පරිභාහිරව අර්ථ දැක්වනු ලබන විචල්‍ය global variable ලෙස හඳුන්වනු ලැබේ. මෙවැනි විචල්‍ය ශ්‍රිතයක් තුළ මෙන් ම ශ්‍රිතයකින් පරිභාහිරව භාවිතා කිරීම සිදු කළ හැකි ය.

එහි දී ශ්‍රිතයක් තුළ එවැනි විචල්‍ය භාවිතා කරන්නේ නම් එය global විචල්‍ය ලෙස විශේෂයෙන් සඳහන් කිරීම කළ යුතු ය.

```
(i) <!DOCTYPE html>
<html>
<body>

<?php
    $a = 10; $b = 15; /* global scope */
    function GlobalTest(){
        echo $a; /* reference to local scope variable */
    }

    GlobalTest ();

?>

</body>
</html>

(ii) <!DOCTYPE html>
<html>
<body>

<?php
    $a = 10;
    $b = 15;

    function GlobalTest (){
        global $a, $b;
        echo $a;
    }

    GlobalTest ();
?>

</body>
</html>
```

Static Variable

සාමාන්‍යයෙන් ශ්‍රිතයක් ක්‍රියාත්මක වී අවසන් වූ පසු එයට සම්බන්ධ ආදාන විචල්‍ය මකාදැමීම සිදු කරයි. කෙසේ වෙතත් ශ්‍රිතය ක්‍රියාත්මක වී අවසන් වූ පසුවත් එම විචල්‍ය මතකයේ රඳවා තබාගත හැකිය. එවැනි විචල්‍ය ස්ථිතික විචල්‍ය හෙවත් Static Variable ලෙස හඳුන්වයි.

```
<!DOCTYPE html>
<html>
<body>

<?php
    function StaticTest() {
        static $x = 0;
        echo $x;
        $x++;
    }

    StaticTest ();
    StaticTest ();
    StaticTest ();
?>

</body>
</html>
```

Data Type (දත්ත වර්ග)

කිසියම් දත්තයක පරිගණකයේ ගබඩා කිරීමට අවශ්‍ය ඉඩ ප්‍රමාණය සහ ඒ මත සිදුකළ හැකි කර්මයන් මත එම දත්ත වර්ගීකරණය කරනු ලබයි. ඒ අනුව php හි සම්මත දත්ත වර්ග කිහිපයකි.

- String
- Integer
- Float
- Boolean
- Array

String

ගණනය කිරීම සඳහා භාවිතා නොකරන අක්ෂර දත්තයන් මෙයට අයත් වේ.

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x = "Hello world!";
    $y = 'hello world!';
    echo $x;
    echo "<br>";
    echo $y;
?>

</body>
</html>
```

Integer

ධන සහ ඍණ පූර්ණ සංඛ්‍යා මෙයට අයත් වේ. -2, 147, 483, 648 සිට 2, 147, 483, 647 දක්වා අයත් වේ.

```
<!DOCTYPE html>
<html>
<body>

<?php
$a = 1234;
var_dump ($a);
echo "<br>";

$b = -123; // a negative number
var_dump ($b);
echo "<br>";

$c = 0123; // octal number
var_dump ($c);
echo "<br>";

$d= 0x1A; // hexadecimal number
var_dump ($d);
echo "<br>";
```

```

    $e = 0b11111111; // binary number (equivalent to 255 decimal)
    var_dump ($e);
    echo "<br>";
    ?>

</body>
</html>

```

Float දශම හෝ භාග සංඛ්‍යා මෙම දත්ත වර්ගයට අයත් වේ.

```

<!DOCTYPE html>
<html>
<body>

<?php
    $x = 10.365;
    var_dump($x);
    echo "<br>";

    $y = 10.2e3;
    var_dump($y);

?>

</body>
</html>

```

Boolean ද්වීමය අවස්ථාවක් පෙන්වීමකදී අවස්ථාවක දී භාවිතා කරනු ලබන අතර True හෝ False යන ද්වීමය අගයක් අයත් වේ.

```

<!DOCTYPE html>
<html>
<body>

<?php
    $x = true;
    var_dump($x);
    echo "<br>";

    $y = false;
    var_dump($y);

?>

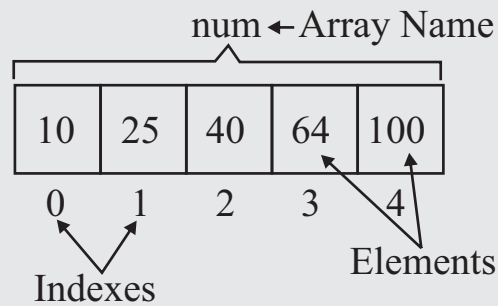
</body>
</html>

```

Array

තනි විචල්‍ය නාමයක් තුළ අගයන් කිහිපයක් ගබඩා කිරීම සඳහා අරාවක් භාවිතා කරනු ලැබේ. මෙම අරාවක තැන්පත් කරනු ලබන අගයන් යතුරක් හෙවත් අනුක්‍රමික අංකයක් මගින් හඳුනා ගැනීම සිදුකළ හැකි ය. සාමාන්‍යයෙන් අරාවක් තුළ එකම වර්ගයේ දත්තයන් ඇතුළත් කරනු ලබයි.

```
$ num = array (10, 25, 40, 64, 100) ;
```



```
(i) <!DOCTYPE html>
<html>
<body>

<?php
    $cars = array("Volvo","BMW","Toyota");
    print_r($cars);
    echo "<br>";

    $color_codes = array (
        "Red" => "#ff0000",
        "Green" => "#00ff00",
        "Blue" => "#0000ff" );
    print_r($color_codes);

?>

</body>
</html>
```

Output

```
Array ([0] => volvo [1] BNW [2] => Toyota)
Array ([Red] => # ff0000 [Green] => # 00FF00 {Blue} => # 0000F
```

```
(ii) <!DOCTYPE html>
<html>
<body>

<?php
$num = array(12,34,56);
$sum=$num[0]+$num[1]+$num[2];
echo $sum;

?>

</body>
</html>
```

PHP ක්‍රියාකාරක

PHP හි භාවිතාකරනු ලබන ක්‍රියාකාරක 5 ක් වේ.

- Arithmetic Operators
- Assignment Operators
- Comparison Operators
- Increment / Decrement Operators
- Logical Operators

1. (Arithmetic Operators)

සාමාන්‍ය ගණිත ක්‍රියාකාරක වෙනුවෙන් භාවිතා කරන ක්‍රියාකාරක මෙයට අයත් වේ.

Operator	Name	Example
+	Addition	$5 + 6 = 11$
-	Subtraction	$8 - 6 = 2$
*	Multiplication	$5 * 2 = 10$
/	Division	$10 / 4 = 2.5$
%	Modulus	$15 \% 4 = 3$
**	Exponentiation	$2 ** 5 = 32$

```

<!DOCTYPE html>
<html>
<body>

<?php
    $x = 5;
    $y= 2;

    echo $x+$y;
    echo "<br>";
    echo $x-$y;
    echo "<br>";
    echo $x*$y;
    echo "<br>";
    echo $x/$y;
    echo "<br>";
    echo $x%$y;
    echo "<br>";
    echo $x**$y;
?>

</body>
</html>

```

2. Assignment Operators

Operator	Name	Example	Same as
=	Assign	\$x = \$y	\$x = \$y
+=	Add and assign	\$x += \$y	\$x = \$x + \$y
-=	Subtract and assign	\$x -= \$y	\$x = \$x - \$y
*=	Multiply and assign	\$x *= \$y	\$x = \$x*\$y
/=	Divids and assign	\$x /= \$y	\$x = \$x / \$y
%=	Divide and assign	\$x%= \$y	\$x = \$x % \$y

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x = 10;
    $y = 5;

    echo $x; // Outputs: 10

    $x += $y;
    echo $x; // Outputs: 15

    $x = 8;
    $y = 5;
    $x -= $y;
    echo $x; // Outputs: 3

    $x = 8;
    $y = 3;
    $x *= $y;
    echo $x; // Outputs: 24

    $x = 17;
    $y = 5;
    $x /= $y;
    echo $x; // Outputs: 3.4

    $x = 8;
    $x %= 3;
    echo $x; // Outputs: 2
?>

</body>
```


3. Comparison Operators

Operator	Name	Example
==	equal	\$x == \$y
===	Identical	\$x === \$y
!=	Not equal	\$x != \$y
<>	Not equal	\$x <> = \$y
!==	Not identical	\$x !== \$y
>	Greater than	\$x > \$y
<	Less than	\$x < \$y
>=	Greater than or equal to	\$x >= \$y
<=	Less than or equal to	\$x <= \$y

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x = 25;
    $y = 35;
    $z = "25";

    var_dump($x == $z); // Outputs: boolean true
    var_dump($x === $z); // Outputs: boolean false
    var_dump($x != $y); // Outputs: boolean true
    var_dump($x !== $z); // Outputs: boolean true
    var_dump($x < $y); // Outputs: boolean true
    var_dump($x > $y); // Outputs: boolean false
    var_dump($x <= $y); // Outputs: boolean true
    var_dump($x >= $y); // Outputs: boolean false
?>

</body>
</html>
```

4. Increment / Decrement Operator

Operator	Name	Description
++ \$x	Pre - increment	x හි අගය වැඩිකර ඉන් අනතුරුව ප්‍රතිදානය කරයි
\$x ++	Post - increment	x ප්‍රතිදානය කිරීමෙන් පසු එකකින් වැඩි කරයි.
-- \$x	Pre - Decrement	x හි අගය 1 කින් අඩුකර ප්‍රතිදානය කරයි
\$x --	Post - Decrement	x ප්‍රතිදානය කර 1 කින් අඩු කරයි.

```

<!DOCTYPE html>
<html>
<body>

<?php
    $x = 10;
    echo ++$x; // Outputs: 11
    echo "<br>";
    echo $x; // Outputs: 11
    echo "<br>";
    echo "<br>";

    $x = 10;
    echo $x++; // Outputs: 10
    echo "<br>";
    echo $x; // Outputs: 11
    echo "<br>";
    echo "<br>";

    $x = 10;
    echo --$x; // Outputs: 9
    echo "<br>";
    echo $x; // Outputs: 9
    echo "<br>";
    echo "<br>";

    $x = 10;
    echo $x--; // Outputs: 10
    echo "<br>";
    echo $x; // Outputs: 9
    echo "<br>";
    echo "<br>";
?>

</body>
</html>

```

5. Logical Operators

Operator	Name
and	And
or	Or
xor	Xor
&&	And
	Or
!	Not

And

```
ex. 01 :- <!DOCTYPE html>
<html>
<body>

<?php
    $x=3;
    $y=5;
    if (($x>0) and ($y<2)){
        echo "Logical true";
    }else{
        echo "Logical false";
    }
?>

</body>
</html>
```

```
ex. 02 :- <!DOCTYPE html>
<html>
<body>

<?php
    $x=3;
    $y=5;
    if (($x>0) && ($y<6)){
        echo "Logical true";
    }else{
        echo "Logical false";
    }
?>

</body>
</html>
```

Or

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x=3;
    $y=5;
    if (($x>0) or ($y<2)){
        echo "Logical true";
    }else{
        echo "Logical false";
    }
?>

</body>
</html>
```

OR (II)

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x=3;
    $y=5;
    if (($x<0) || ($y>6)){
        echo "Logical true";
    }else{
        echo "Logical false";
    }
?>

</body>
</html>
```

XOR

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x=3;
    $y=5;
    if (($x>0) xor ($y<6)){
        echo "Logical true";
    }else{
        echo "Logical false";
    }
?>

</body>
</html>
```

NOT

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x=3;
    $y=5;
    if (!($x>0)){
        echo "Logical true";
    }else{
        echo "Logical false";
    }

?>

</body>
</html>
```

පාලන ආකෘති (Control Structures)

PHP හි භාවිතා කරනු ලබන පාලන ආකෘති පහත පරිදි සාරාංශ කර දැක්විය හැකිය.

1) කොන්දේසි සහිත ප්‍රකාශන (වරණය)

01. if statement
02. if.....else statement
03. if.....elseif.....else statement
04. switch statement

2) පුනර්කරණ පාලනය (Control Loops)

01. While
02. Do....while
03. for
04. foreach

වරණය

01. if Statement

කිසියම් කොන්දේසියක් සත්‍ය වන අවස්ථාවේ දී පමණක් සිදුකළ යුතු ප්‍රකාශන භාවිතයේ දී යොදාගැනේ.

```

<!DOCTYPE html>
<html>
<body>

<?php
    $marks = 78;
    if ($marks > 50) {
        echo " You are pass the exam";
    }
?>

</body>
</html>

```

02. if....else Statement

එක් කොන්දේසියක් මත එම කොන්දේසිය සත්‍ය වන අවස්ථාවේ හා අසත්‍ය වන අවස්ථාවේ දී භාවිතා කරයි.

```

<!DOCTYPE html>
<html>
<body>

<?php
    $marks = 45;
    if ($marks > 50) {
        echo "You are pass the exam";
    }else{
        echo "Sorry! you are fail the exam";
    }
?>

</body>
</html>

```

03. if.....elseif.....else statement

ඔහු කොන්දේසි භාවිතා කරනු ලබන අවස්ථාවන්හි දී භාවිතා කරයි.

```
<!DOCTYPE html>
<html>
<body>

<?php
    $marks = 49;
    if ($marks >= 75) {
        echo "A";
    }elseif ($marks >= 65) {
        echo "B";
    } elseif ($marks >= 50) {
        echo "C";
    } elseif ($marks >= 35) {
        echo "S";
    } else{
        echo "W";
    }
?>

</body>
</html>
```


Switch Statement

කේවල විචල්‍යයකට බහු කොන්දේසි පවතින විට භාවිතා කරයි.

```
<!DOCTYPE html>
<html>
<body>
<?php
    $today = "Tue";
    switch ($today) {
        case "Mon":
            echo "Today is Monday. Wake up early in the morning.";
            break;
        case "Tue":
            echo "Today is Tuesday. Do your homework.";
            break;
        case "Wed":
            echo "Today is Wednesday. Play a game";
            break;
        case "Thu":
            echo "Today is Thursday. Its movie time";
            break;
        case "Fri":
            echo "Today is Friday. Do some rest.";
            break;
        default:
            echo "No information available for that day.";
    }
?>

</body>
</html>
```

පුනර්කරන පාලන ව්‍යුහය

php හි පුනර්කරණ පාලන ව්‍යුහ 04 ක් හඳුනාගත හැකිය.

- while
- do....while
- for
- for each

While

කිසියම් කොන්දේසියක් සත්‍යයව පවතින තුරු ප්‍රකාශනයක් හෝ ප්‍රකාශන කිහිපයක් නැවත නැවත සිදු කිරීම සඳහා while පාලන ව්‍යුහය භාවිතා කරනු ලබයි.

```
ex. 01 :- <!DOCTYPE html>
<html>
<body>

<?php
    $x = 0;
    while($x <= 5) {
        echo "The number is: $x <br>";
        $x++;
    }
?>

</body>
</html>
```

කිසිවිටෙකත් අවසාන නොවන While පුනර්කරණය

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x = 1;
    while($x >= 0) {
        echo "The number is: $x <br>";
        $x++;
    }
    echo "End.....";
?>

</body>
</html>
```

කිසිවිටෙකත් ආරම්භ නොවන While පුනර්කරණය

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x = 0;
    while($x >= 5) {
        echo "The number is: $x <br>";
        $x++;
    }
    echo "End....";
?>

</body>
</html>
```

do...while

do while පුනර්කරණයේ දී පළමු අවස්ථාවේදී ම කොන්දේසි අසත්‍ය වුවත් පුනර්කරණය එක් වතාවක් හෝ සිදු වේ. එයට හේතුව වන්නේ Do While පුනර්කරණයේ දී පුනර්කරණයට අදාළ ප්‍රකාශන ක්‍රියාත්මක වීමෙන් පසු කොන්දේසි පරීක්ෂාකර බැලීමයි.

```
<!DOCTYPE html>
<html>
<body>

<?php
    $x = 2;
    do {
        echo " $x <br>";
        $x +=2;
    } while ($x <= 10);
?>

</body>
</html>
```

For

කිසියම් ප්‍රකාශනයක් හෝ ප්‍රකාශන කිහිපයක් ක්‍රියාත්මක විය යුතු වාර ගණන නිශ්චිතව දන්නා විට for පුනර්කරණ පාලන ව්‍යුහය භාවිතා කෙරේ.

```

1) <!DOCTYPE html>
   <html>
   <body>

   <?php
     for ($x = 1; $x <= 10; $x++) {
       echo $x. " , ";
     }
   ?>

   </body>
   </html>

```

```

2) <!DOCTYPE html>
   <html>
   <body>

   <?php
     for ($x = 1; $x <= 10; $x+=2) {
       echo $x. " , ";
     }
   ?>

   </body>
   </html>

```

Foreach

for each පාලන ව්‍යුහය අරාවක් (array) සමඟ පමණක් භාවිත කළ හැකිය. මෙමගින් අරාවක ඇති මූලිකාංග එකිනෙක එහි අවසානය දක්වා ප්‍රතිදානය කිරීමේ

```

<!DOCTYPE html>
<html>
<body>

<?php
  $foods = array("Bread", "Chocolate", "Biscuit", "Ice-Cream");
  foreach ($foods as $value) {
    echo "$value <br>";
  }
?>

</body>
</html>

```

PHP ශ්‍රිත

php සතුව දැනටමත් නිර්මාණය කර පවතින ශ්‍රිත 1000 කට අධික ප්‍රමාණයක් පවතී. පරිශීලකයාගේ අවශ්‍යතාවය පරිදි එම ශ්‍රිත php ක්‍රමලේඛ තුළ භාවිතා කළ හැකිය. එමෙන් ම php මගින් පරිශීලකයාට තමන්ගේ අවශ්‍යතාවය පරිදි ශ්‍රිත නිර්මාණය කිරීමට අවස්ථාව පවතී.

මෙසේ ශ්‍රිත නිර්මාණය කිරීමෙන් පසු එම ශ්‍රිත නැවත නැවත භාවිතා කිරීමේ හැකියාව ද පවතී.

ශ්‍රිතයන් නිර්මාණය කිරීමට අවශ්‍ය කාරක රීති,

Syntax

```
Function Function_Name (Parameters){  
Code to be executed ;  
}
```

Example :- 1. පරාමිතින් නොමැතිව සරලව අර්ථ දැක්වෙන ශ්‍රිතයක් පහත දැක්වේ.

```
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
function writeMessage() {  
    echo "Hello ICT Students!";  
}  
  
writeMessage(); // call the function  
>  
  
</body>  
</html>
```

Example :- 2. පරාමිතින් යොදාගෙන නිර්මාණය කරනු ලබන ශ්‍රිතයන් පහත දැක්වේ.

```
1) <!DOCTYPE html>  
<html>  
<body>  
  
<?php  
function ShowName($fname) {  
    echo "Your name is $fname <br>";  
}  
  
ShowName("Jack");  
ShowName("Peter");  
ShowName("Smith");  
>  
  
</body>  
</html>
```

```

2) <!DOCTYPE html>
<html>
<body>

<?php
function getSum($num1 , $num2) {
    $sum = $num1 + $num2;
    echo "Sum of two numbers $num1 and $num2 is : $sum ";
}

getSum(25 , 56);
?>

</body>
</html>

```

පුරුදු පරාමිතීන් භාවිතා කිරීමෙන් ශ්‍රිත ගොඩනැගීම

```

<!DOCTYPE html>
<html>
<body>

<?php
function setAge($name , $age = 25) {
    echo "$name 's age is $age years old. <br>";
}

setAge("Peter" , 30);
setAge("Jack"); // will use the default value of 25
setAge("John" , 28);
?>

</body>
</html>

```

ශ්‍රීතයකින් අගයක් ප්‍රතිදානය කරන ආකාරය දැක්වෙන php ක්‍රමලේඛය පහත දැක්වේ.

```
<!DOCTYPE html>
<html>
<body>

<?php
function getAverage($num1 , $num2, $num3) {
    $avg = ($num1 + $num2 + $num3)/3;
    return $avg;
}

echo "Average value is : " . getAverage(75, 30, 56);
?>

</body>
</html>
```

දත්ත සම්බන්ධතා සහ සම්බන්ධතා කටයුතු කිරීම

Database Connectivity

php කේතයක් දත්ත සම්බන්ධතා සමඟ සම්බන්ධ කරන විට පළමුවෙන් ම සිදුකළ යුත්තේ Mysql හා සම්බන්ධ වීමයි. ඒ සඳහා පහත පරිදි එම සම්බන්ධතාවය ඇතිකළ හැකි ය.

```
$conn = mysqli_connect ($servername, $username, $password)
හෝ
$conn = mysqli_connect ("localhost", "root", "1234")
```

mysql වලට සම්බන්ධ වීමෙන් අනතුරුව දත්ත සම්බන්ධතා නිර්මාණය කිරීමෙන් පසු දත්ත සම්බන්ධතා ක්‍රියාකාරකම් සිදු කිරීමට දත්ත සම්බන්ධතා සමඟ සම්බන්ධතාව ඇති කළ යුතු ය.

```
$conn = mysqli_connect($servername, $username, $password, $database);

<?php
$servername = "localhost";
$username = "root";
$password = "1234";
$dbname= "school";

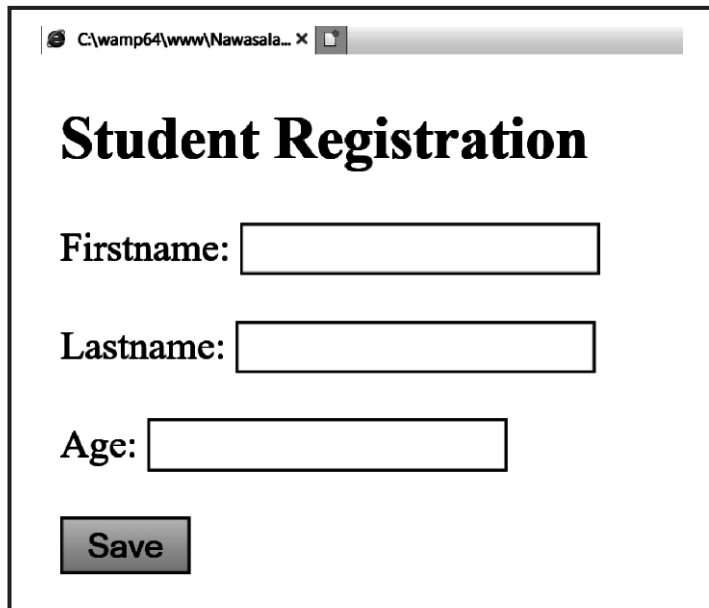
// Create connection

$conn =mysqli_connect($servername, $username, $password, $database);

// Check connection

if ($conn === false) {
    die("Error : Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

වෙබ් පෝරමයක් මගින් දැන්වූ සමුදායට සම්බන්ධ වීම.



The screenshot shows a web browser window with the address bar displaying 'C:\wamp64\www\Nawasala...'. The main content of the page is a registration form titled 'Student Registration'. The form consists of three text input fields: 'Firstname:', 'Lastname:', and 'Age:'. Below these fields is a 'Save' button.

Submit Query

```
<!DOCTYPE html>
<html>
<head>
</head>

<body>
<h2>Student Registration</h2>

<form action="insert.php" method="post">
Firstname: <input type="text" name="firstname" /> <br/><br/>
Lastname: <input type="text" name="lastname" /> <br/><br/>
Age: <input type="text" name="age" /><br/><br/>
<input type="submit" value="Save"/>
</form>

</body>

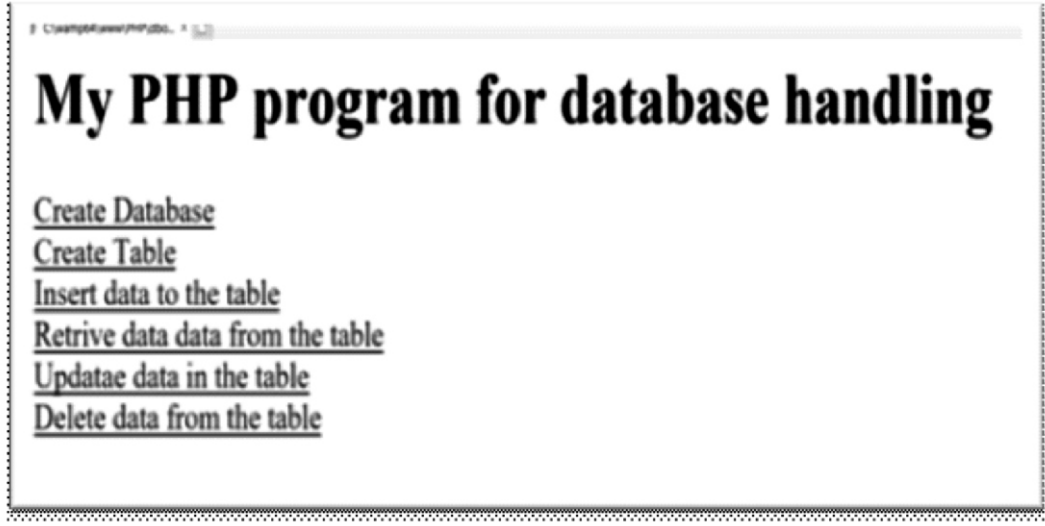
</html>
```



```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$sql="INSERT INTO Persons (FirstName, LastName, Age)
VALUES ('$_POST[firstname]', '$_POST[lastname]', '$_POST[age]')";
if (!mysql_query($sql,$con))
{
    die('Error: ' . mysql_error());
}
echo "1 record added";
mysql_close($con);
?>
```

Practical PHP – Database connection

- Create the following html document



```
<html>
<body>
<h1>My PHP program for database handling</h1>

<a href="dbc.php">Create Database</a><br/>
<a href="tablecr.php">Create Table</a><br/>
<a href="dataen.php">Insert data to the table</a><br/>
<a href="retriveda.php">Retrive data data from the table</a><br/>
<a href="updateda.php">Updatae data in the table</a><br/>
<a href="deleteda.php">Delete data from the table</a><br/>
</body>
</html>
```

1.Create database

```
<html>
<body>
</body>
<?php
$servername = "localhost";
$username = "root";
$password = "";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
echo "Database created successfully";
}else {
echo "Error creating database: " . $conn->error;
}

$conn->close();
?>
</html>
```

2.Create table

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50))";

if ($conn->query($sql) === TRUE) {
echo "Table MyGuests created successfully";
}else {
echo "Error creating table: " . $conn->error;
}

$conn->close();
?>
```

3. Inserting data into the table

```
<html>
<body>
</body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com)";

if ($conn->query($sql) === TRUE) {
echo "New record created successfully";
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
</html>
```

4. Selecting data into the table

```
<html>
<body>
</body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
echo "0 results";
}
$conn->close();
?>
</html>
```

5. Updating data

```
<html>
<body>
</body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

$sql = "UPDATE MyGuests SET lastname='Wickramage' WHERE id=1";

if ($conn->query($sql) === TRUE) {
echo "Record updated successfully";
}else {
echo "Error updating record: " . $conn->error;
}

$conn->close();
?>
</html>
```

6. Deleting data

```
<html>
<body>
</body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=1";

if ($conn->query($sql) === TRUE) {
echo "Record deleted successfully";
} else {
echo "Error deleting record: " . $conn->error;
}

$conn->close();
?>
</html>
```