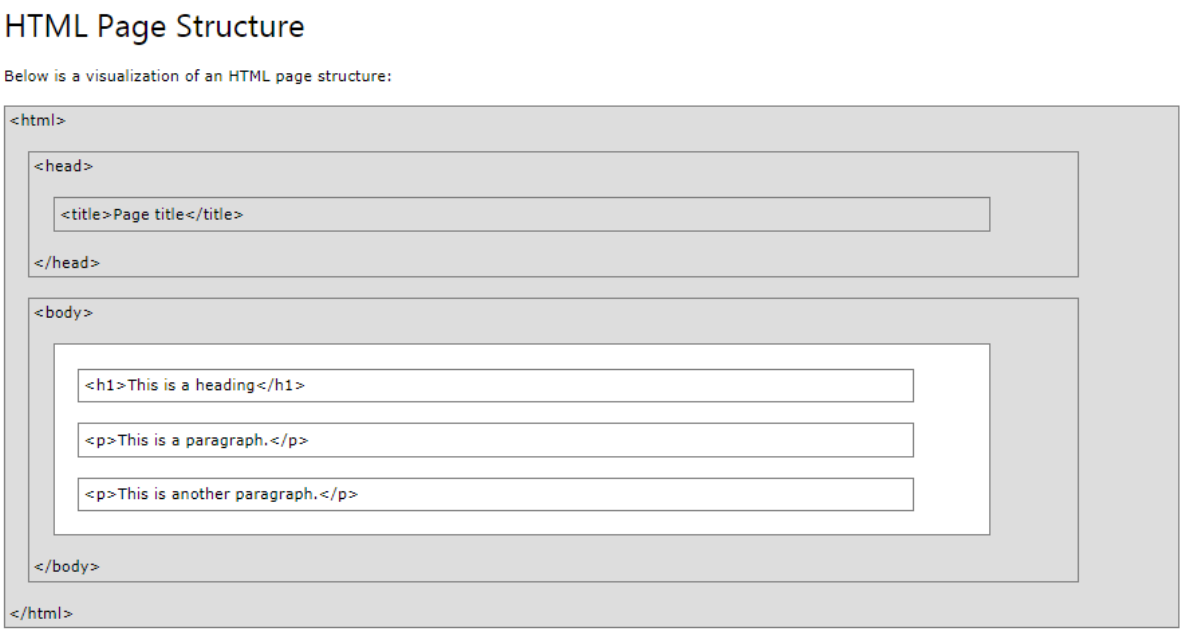
HTML

HTML is the standard markup language for creating Web pages.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```



- This is left aligned
- The following word uses a ^{superscript} typeface.
- 1989 Tim Berners-Lee invented www
- 1991 Tim Berners-Lee invented HTML

LINK Tags

The <a> tag defines a hyperlink, which is used to link from one page to another.

```
<a href="f2.html">This is a link</a>
```

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

1) How to open a link in a new browser window:

```
<a href="f2.html" target="_blank"> click here……</a>
```

2) How to use an image as a link:

```
<a href="f2.html">
<img border="0" alt="problem" src="logo_w3s.gif" width="100" height="10">
</a>
```

Table

```
<table border="1" align=”left” cellpadding=”2” cellspacing=”2” width=”30%”>

<caption><u>Students Information</u></caption>

<th><font size="20" color="red"> ROLL NUMB </th>
<th><font size="20" color="red"> PERCEN  </th>

<tr>
<td><font size="25" color="blue">  <a href="inter.html"> 8</a></td>
<td><font size="25" color="blue">80</td>
</tr>
```

```
<tr>
<td> 18</td>
<td>90</td>
</tr>

</table>
```

HTML Lists

HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

1. Ordered List or Numbered List (ol)

2. Unordered List or Bulleted List (ul)
3. Description List or Definition List (dl)

1) **<ol>**
**
Aries
**
**
Bingo
**
**
Leo
**
**
Oracle
**
**
**

2) **<ul>**
**
Aries
**
**
Bingo
**
**
Leo
**
**
Oracle
**
**
**

3)**dl> tag** defines the start of the list.
<dt> tag defines a term.
<dd> tag defines the term definition (description).

```
<dl>
  <dt>Aries</dt>
  <dd>-One of the 12 horoscope sign.</dd>
  <dt>Bingo</dt>
  <dd>-One of my evening snacks</dd>
 <dt>Leo</dt>
 <dd>-It is also an one of the 12 horoscope sign.</dd>
  <dt>Oracle</dt>
  <dd>-It is a multinational technology corporation.</dd>
</dl>
```

HTML Image

1. < **img** src="animal.jpg" height="180" width="300" alt="animal image">

Common Image Formats

Here are the most common image file types, which are supported in all browsers (Chrome, Edge, Firefox, Safari, Opera):

Abbreviation	File Format	File Extension
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur
JPEG	Joint Photographic Expert Group image	.jpg, .jpeg, .jfif, .pjpeg, .pjp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg

HTML Comment Tags

You can add comments to your HTML source by using the following syntax:

<!-- Write your comments here -->

HTML <frame> tag

```
<head>
   <title>Frame tag</title>
</head>
  <frameset cols="25%,50%,25%">
    <frame src="frame1.html" >
    <frame src="frame2.html">
    <frame src="frame3.html">
  </frameset>
</html>
```

HTML Form

An **HTML form** is *a section of a document* which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc. .

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form −

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

1) Example text,pwd, text area

```
<body>
    <form >
      First name: <input type = "text" name = "first_name"  value=””   size=”” />
      <br>
      Last name: <input type = "text" name = "last_name" />
     Password: <input type = "password" name = "password" />
    <textarea rows = "5" cols = "50" name = "description" />
    </form>
  </body>
```

2)Example CHECKBOX, Radio

```
<input type = "checkbox" name = "maths" value = "on"> Maths
```

<input type = "checkbox" name = "physics" value = "on"> Physics

<input type = "radio" name = "subject" value = "maths"> Maths

<input type = "radio" name = "subject" value = "physics" checked=”true” > Physics

3)Dropdown / combo box

```
body>
    <form>
      <select name = "dropdown">
        <option value = "Maths" selected> Maths</option>
        <option value = "Physics">  Physics</option>
      </select>
    </form>
  </body>
```

4) File BOX

```
<form>

      <input type = "file" name = "resumes" accept = "image/*" />

    </form>
```

5)Button Controls

Sr.No	Type & Description
1	**submit** This creates a button that automatically submits a form.
2	**reset** This creates a button that automatically resets form controls to their initial values.
3	**button** This creates a button that is used to trigger a client-side script when the user clicks that

	button.
4	**image** This creates a clickable button but we can use an image as background of the button.

Example

```
<form>
    <input type = "submit" name = "submit" value = "Submit" />
    <input type = "reset" name = "reset"  value = "Reset" />
    <input type = "button" name = "ok" value = "OK" />
    <input type = "image" name = "im" src = "/html/images/logo.png" />
</form>
```

Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page.

```
<input type = "hidden" name = "pagename" value = "10" />
```

Image LINK

```
<body>
    <p>Click following link</p>
    <a href = "https://www.tutorialspoint.com" target = "_self">
      <img src = "f1.png" />
    </a>
</body>
```

4)Marque

An HTML marquee is a scrolling piece of text displayed either horizontally across or vertically down your webpage depending on the settings. This is created by using HTML <marquees> tag.

```
<body>

    <marquee width = "50%  direction=”right ">This example will take only 50%
width</marquee>

</body>
```

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

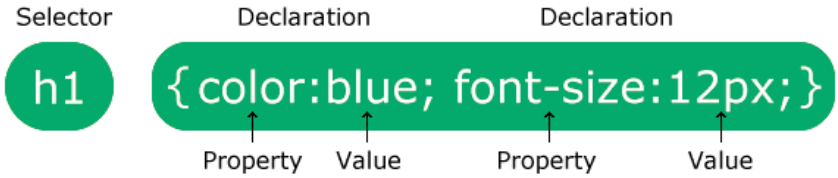
CSS Example

```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

CSS Syntax



Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

1.Internal CSS- single HTML page

- The internal style is defined inside the <style> element, inside the head section.

```
<html>
<head>
<style>


body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}



h2 {
  color:blue;
  margin-left: 60px;
}

</style>
</head>
<body>

<h1> HAVE a nice day…….</h1>

<h1> II BCA Student……..</h2>
<p>This is a paragraph.</p>

</body>
</html>
```

2.Inline CSS-Single Element

- The style attribute can contain any CSS property.

```
<html>
<body>

<h1 style="color:blue; text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

```
</body>
</html>
```

3.External CSS

External styles are defined within the <link> element, inside the <head> section of an HTML page:

HTML FILE

```
<html>
<head>
<link rel="stylesheet" href="my.css">
</head>
<body>

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

</body>
</html>
```

"my.css"

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
    }

p{

  font-sixe=35;

}
```

Css margin, border, Div(Section of HTML page)

```
<html>
<head>
<style>

div {
  border: 3px solid red;
  margin-top: 400px;
  margin-left: 180px;
  background-color: lightblue;
}
</style>
</head>

<body>

<div>This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.



<h2>Using individual margin properties</h2>
<table>,,,

</div>
</body>
</html>
```

CSS Borders , css class

`border-style`

```
<html>
<head>
<style>
```

```
p.one {
  border-style: solid;
  border-width: 5px 20px; /* 5px top and bottom, 20px on the sides */

  border-color: red;

  text-transform: capitalize;

  text-shadow: 2px 2px;
}


p.two {

 border-style: dotted;

 border-width: 2px;

}

</style>
</head>
<body>

<h2>The border-width Property</h2>
<p>This property specifies the width of the four borders:</p>

<p class="one" >Some text for para one . this is a lockdown period.</p>

<p class="two">Some text for p2. Very much care full in this wave</p>

</body>
>/html>
```

2) Mouse over, link, list –Navigation Bar

```
<html>
<head>
<style>

ul {
```

```
  list-style-type: none;---->remove bullets
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
  border: 1px solid #555;
}

li a {
  display: block; --->whole area will be blocked
  color: red;
  padding: 8px 16px;
  text-decoration: none;
}

/* Change the link color on hover */
li a:hover {
  background-color: pink;
  color: white;
}
</style>
</head>
<body>

<h2>Vertical Navigation Bar</h2>

<ul>
  <li><a href="#b1">IBCA</a></li>
  <li><a href="#b2">II BCA</a></li>
  <li><a href="#b3">III BCA</a></li>
</ul>

</body>
</html>
```

Table & CSS

```
<html>
```

```
<head>
<style>

table {
  width: 100%;
border-collapse: collapse;
}
table, th, td {
  border: 1px solid black;
}

td {
  text-align: center;
}

</style>
</head>
<body>

<h2>Add a border to a table:</h2>

<table>
  <tr>
    <th>Roll NO</th>
    <th>Percentage </th>
  </tr>
  <tr>
    <td>01</td>
    <td>88</td>
  </tr>

  <tr>
    <td>02</td>
    <td>92</td>
  </tr>
</table>

</body>
</html>
```

Forms & CSS

```
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: none;
  background-color: #3CBC8D;
  color: white;
}



input[type=text]:focus {
  background-color: lightblue;
}

</style>
```

JS

- **JavaScript** to program the behavior of web pages
- Client-side validation
- JavaScript is *an object-based scripting language* which is lightweight and cross-platform.
- JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

Where to use JS in HTML Program

JavaScript in <head>

```
<!DOCTYPE html>
<html>

<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h1>A Web Page</h1>


<p id="demo">A Paragraph</p>


<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```

JavaScript in <body>----→o/p are same

```
<html>
<body>

<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>

</body>
</html>
```

External JavaScript

myScript.js

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

```
<html>
<body>

<h2>External JavaScript</h2>

<p id="demo">A Paragraph.</p>

<button type="button" onclick="myFunction()">Try it</button>

<p>(myFunction is stored in an external file called "myScript.js")</p>

<script src="myScript.js"></script>

</body>
```

</html>

JavaScript Display Possibilities

JavaScript can "display" data in different ways:

- Writing into an HTML element, using `innerHTML`.
- Writing into the HTML output using `document.write()`.
- Writing into an alert box, using `window.alert()`.
- Writing into the browser console, using `console.log()`.

2)Writing into the HTML output using `document.write()`.

```
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

3)Writing into an alert box, using `window.alert()`.

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

<h1>My First Web Page</h1>
```

```
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

4)Writing into the browser console, using `console.log()`.

// Log -→f12 or right click on webpage-→ inspect.

```
<html>
<body>

<h2>Activate Debugging</h2>

<p>F12 on your keybord will activate debugging.</p>
<p>Then select "Console" in the debugger menu.</p>
<p>Then click Run again.</p>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

Variables

```
<script>

var data=200;//gloabal variable

function a(){
document.writeln(data);
}
```

```
function b(){
document.writeln(data);
}
</script>
```

Functions

```
<html>
<body>
<script>
functiongetcube(number){
alert(number*number*number);
}
</script>
<form>
<input type="button" value="click" onclick="getcube(4)"/>
</form>
</body>
</html>
```

Common HTML Events

Here is a list of some common HTML events:

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

Form events:

Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

Example for local variable, onblur, onfocus, function (backcolor)

```
<html>
<head>
<script>

function f1()
{
var t1=document.getElementById("t1");

t1.style.background="grey";
}

function f11()
{
var t2=document.getElementById("t2");
t2.style.background="grey";

}

</script>
</head>


<body>

<input type="text" id="t1" onfocus="f1()"   onblur="f2()">
```

```
<br><br>
<input type="text" id="t2" onfocus="f11()"  onblur="f2()">

</body>

</html>
```

2) Example for log.write (), onclick

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

function my()
{
console.log("hai.  button clicked.....");
}


</script>
</head>


<body>
<h1>A Web Page</h1>

<p id="demo">  A Paragraph  is about to demo HTML usage  for our BCA Students</p>
<p>
we all in online class </p>

<input type="button" onclick="my()"  value="ok"></button>
</body>
</html>
```

3)) Example for alert () , multiple function (Capital Case)

```
<html>
<head>Javascript Intro
<script>

var tt;

function f1()
{
tt=document.getElementById("t1");
alert("Give Name in Capital");
tt.value="";

}

function f2()
{
var v=tt.value;
var ss=v.toUpperCase();
tt.value=ss;

}
</script>
</head>


<body>
<h2> Enter something here</h2>

<input type="text" id="t1" onclick="f1()" onblur="f2()"/>

</body>
</html>
```

4) Example for onsubmit event

```
<html>
<head>
<script>

function f1()
{
var x=document.getElementById("t1");
var y=document.getElementById("t2");

 var a=x.value;
var b=y.value;

if((a.length==0) || (b<0) || (b>100))
    alert('wromg data submission');
  else
 alert('Successfull data submission');


}

</script>
</head>


<body>
<form onsubmit="f1()">
Roll No
<input type="text" id="t1">
<br>
Percentage
<input type="text" id="t2">


<input type="submit" value="ok" />
</form>
</body>
</html>
```

5)Example for onchange event

```
<html>
<body>
<p>Select one of the favourite fruit  </p>

<select id="fruit" onchange="f1()">
<option value="Chico">Chico
<option value="Apple">Apple
<option value="Guava">Guava
<option value="Pineapple">Pineapple
</select>
<p id="para">"SELECTED Fruit:"</p>


<script>
function f1() {
var x = document.getElementById("fruit");
var v=x.value;
document.getElementById("para").innerHTML = "Select from the given fruit list " + v;
}
</script>


</body>
</html>
```

What is PHP

PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side.

PHP is well suited for web development.

PHP was created by Rasmus Lerdorf in 1994 but appeared in the market in 1995.

- PHP stands for Hypertext Preprocessor.
- PHP is an interpreted language, i.e., there is no need for compilation.
- PHP is faster than other scripting languages, for example, ASP and JSP.
- PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
- PHP can be embedded into HTML.
- PHP is an object-oriented language.
- PHP is an open-source scripting language.
- PHP is simple and easy to learn language.

To install PHP, we will suggest you to install AMP (Apache, MySQL, PHP) software stack. It is available for all operating systems. There are many AMP options available in the market that are given below:

- WAMP for Windows
- LAMP for Linux
- MAMP for Mac
- SAMP for Solaris
- FAMP for FreeBSD
- XAMPP (Cross, Apache, MySQL, PHP, Perl) for Cross Platform: It includes some other components too such as FileZilla, OpenSSL, Webalizer, Mercury Mail, etc.
-

PHP Case Sensitivity

In PHP, keyword (e.g., echo, if, else, while), functions, user-defined functions, classes are not case-sensitive. However, all variable names are case-sensitive.

How to run PHP programs in XAMPP

```
<body>
<?php
echo "<h2>Hello First PHP</h2>";
```

1. print "Hello by PHP print "; ----------→slower

```
?>
</body>
</html>
```

PHP print: printing variable value

```
1. <?php
2. $msg="Hello in PHP";
3. print "Message is: $msg";
4. print “hai”.$msg;
5. ?>
```

PHP Variables

In PHP, a variable is declared using a $ sign followed by the variable name. Here, some important points to know about variables:

- As PHP is a loosely typed language, so we do not need to declare the data types of the variables. It automatically analyzes the values and makes conversions to its correct datatype.
- After declaring a variable, it can be reused throughout the code.

```php
<?php
$str="hello string";
$x=200;
$y=44.6;
echo "string is: $str <br/>";
echo "integer is: $x <br/>";
echo "float is: $y <br/>";
?>
```

PHP Variable Scope

The scope of a variable is defined as its range in the program under which it can be accessed. In other words, "The scope of a variable is the portion of the program within which it is defined and can be accessed."

PHP has three types of variable scopes:

1. Local variable
2. Global variable
3. Static variable

Local variable

The variables that are declared within a function are called local variables for that function. These local variables have their scope only in that particular function in which they are declared.

```php
<?php
    function lo()
    {
        $num = 45;  //local variable
        echo "Local variable declared inside the function is: ". $num;
    }
    lo();
?>
```

Global variable

The global variables are the variables that are declared outside the function. These variables can be accessed anywhere in the program.

```php
<?php
   $name = " Sharma";       //Global Variable
   function glo()
   {

      echo "Variable inside the function: ". $name;
      echo "</br>";
   }


   glo();
   echo "Variable outside the function: ". $name;
?>
```

Static variable

It is a feature of PHP to delete the variable, once it completes its execution and memory is freed. Sometimes we need to store a variable even after completion of function execution.

```php
function static_var()
 {
    static $num1 ;       //static variable
    $num2 ;       //Non-static variable
    //increment in non-static variable
    $num1++;
    //increment in static variable
    $num2++;
    echo "Static: " .$num1 ."</br>";
    echo "Non-static: " .$num2 ."</br>";
 }

//first function call
```

```
14.     static_var();
15.
16.     //second function call
17.     static_var();
18. ?>
```

PHP Data Types

Variables can store data of different types, and different data types can do different things.

PHP supports the following data types:

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

Create a PHP Constant

To create a constant, use the `define()` function.

Syntax

```
define(name, value)
```

```
<?php
define("GREETING", "Welcome");


?>
```

PHP Operators

Operators are used to perform operations on variables and values.

PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators
- Conditional assignment operators

Create an Array in PHP

In PHP, the `array()` function is used to create an array:

```
array();
```

In PHP, there are three types of arrays:

- Indexed arrays - Arrays with a numeric index
- Associative arrays - Arrays with named keys
- Multidimensional arrays - Arrays containing one or more arrays

1) Indexed arrays

```
<html>
<body>

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

</body>
</html>
```

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
$arrlength = count($cars);

for($x = 0; $x < $arrlength; $x++) {
  echo $cars[$x];
  echo "<br>";
}
?>
```

PHP Associative Arrays

Associative arrays are arrays that use named keys that you assign to them.

There are two ways to create an associative array:

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

or:

```
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";
```

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
echo "Peter is " . $age['Peter'] . " years old.";
?>
```

When to use GET?

Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL). GET also has limits on the amount of information to send. The limitation is about 2000 characters. However, because the variables are displayed in the URL, it is possible to bookmark the page. This can be useful in some cases.

GET may be used for sending non-sensitive data.

Note: GET should NEVER be used for sending passwords or other sensitive information!

When to use POST?

Information sent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send.

Moreover POST supports advanced functionality such as support for multi-part binary input while uploading files to server.

PHP User Defined Functions

Besides the built-in PHP functions, it is possible to create your own functions.

- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute automatically when a page loads.
- A function will be executed by a call to the function.

Create a User Defined Function in PHP

A user-defined function declaration starts with the word `function`:

Syntax

```
function functionName() {
  code to be executed;
}
```

example:

```
<?php
function f1() {
  echo "Hello world!";
}

f1(); // call the function
?>
```

HP Function Arguments

Information can be passed to functions through arguments. An argument is just like a variable.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

```php
<?php
function f1($fname) {
  echo "$fname Refsnes.<br>";
}

f1("Jani");
f1("Hege");
f1("Stale");?>
```

PHP Functions - Returning values

To let a function return a value, use the `return` statement:

```php
<?php


function sum($x,  $y) {
  $z = $x + $y;
  return $z;
}

echo "5 + 10 = " . sum(5, 10) . "<br>";
echo "7 + 13 = " . sum(7, 13) . "<br>";

?>
```

PHP Default Argument Value

The following example shows how to use a default parameter. If we call the function setHeight() without arguments it takes the default value as argument:

```php
<?php


 function setHeight(int $minheight = 50) {
  echo "The height is : $minheight <br>";
}

setHeight(350);
setHeight(); // will use the default value of 50
setHeight(135);
setHeight(80);
?>
```

PHP Recursive Function

we call current function within function. It is also known as recursion.

```
<?php
function display($number) {
    if($number<=5){
     echo "$number <br/>";
     display($number+1);
    }
}

display(1);
?>
```

1. Program for sum of two values

```
<html>
<body>
<form name="sqfrm" method="post" action="http://localhost/bca20/f1.php">
  <p> Give 2 VALUE <p>
  <input type="text" name="t1"><br><br>

 <input type="text"  name="t2"><br>
  <input type="submit" value="Submit">
</form>

</body>

</html>
```

---

f1.php

```
<?php

$a=$_POST['t1'];
$b=$_POST['t2'];
$sq= intval($a)+ intval($b);

echo "OUTPUT=".$sq;

?>
```

++

2) DEMO for FOR in PHP

```
<html>
<body>
<form name="sqfrm" method="post" action="http://localhost/bca20/gr.php">
  <p> Give 2 VALUE <p>
  <input type="text" name="t1"><br><br>
  <input type="submit" value="Submit">
</form>

</body>

</html>
```

----------gr.php----------

```
<?php

$a=$_POST['t1'];

$n=intval($a);

for($i=1; $i<=$n;$i++)
  echo "have a NICE DAY <br>";
?>
```

++

3) DEMO for usage of HTML in PHP

```
<html>
<body>
<form name="sqfrm" method="post" action="http://localhost/bca20/tabex.php">
 <p> Give N VALUE <p>
 <input type="text" name="t1"><br><br>
 <input type="submit" value="Submit">
</form>

</body>
</html>
```

----------tabex.php----------

```
<?php

$a=$_POST['t1'];
$n=intval($a);

echo "<table border=2>";
echo "<th> NUMBER <th> CUBE";


for($i=1; $i<=$n;$i++)
{
echo "<tr><td>". $i;
echo "<td>". $i*$i*$i;
}

echo "</table>";

?>
```

++

4)HTML FORM CONTROLS & PHP

```
<html>

<body>

<form name="sample" method="post" action="http://localhost/bca20/frmser.php">

Give Your Name
<br><input type="text" name="t1">

<br> Give Yr Gender
<br><input type="radio" name="gen" value="male">MALE
<br><input type="radio" name="gen" value="female">FEMALE

<br> Choose your value Added COURSE
<br> <input type="checkbox" name="c1" value="ST">SW TESTING
<br> <input type="checkbox" name="c2" value="DEV">DevOp
<br> <input type="checkbox" name="c3" value="CLOUD">Cloud

<br> <input type="submit" >

</form>
</body>
</html>
```

---

frmser.php

```
<?php

$na=$_POST['t1'];
$gn=$_POST['gen'];
$cc1=$_POST['c1'];
$cc2=$_POST['c2'];
$cc3=$_POST['c3'];


$credit=0;
if($cc1=="ST")
  $credit=$credit+4;
if($cc2=="DEV")
  $credit=$credit+5;
```

```
if($cc3=="CLOUD")
  $credit=$credit+10;
echo $na. $gn. $cc1.$cc2. $cc3. "<br>TOTAL Credit". $credit;
?>
```

++

5) Simple QUIZ

```
<html>
<body>


<form name="quiz" method="post" action="http://localhost/bca20/examser.php">

<p>

1)What is the size of byte variable?

<br>
<br><input type="radio"  name="qs1" value="a">A - 8 bit
<br><input type="radio"  name="qs1" value="b">B - 16 bit
<br><input type="radio"  name="qs1" value="c">C - 32 bit
<br><input type="radio" name="qs1" value="d">D - 64 bit

<p>

2) What kind of variables a class can consist of?
<br>

<br><input type="radio" id="q2" name="qs2" value="a">A - class variables, instance variables
<br><input type="radio" id="q2" name="qs2" value="b">B - class variables, local variables, instance
variable
<br><input type="radio" id="q2" name="qs2" value="c">C - class variables
<br><input type="radio" id="q2" name="qs2" value="d">D - class variables, local variables
<p>

3) What invokes a thread's run() method?
<br>

<br><input type="radio" id="q3" name="qs3" value="a">A - JVM invokes the thread's run() method
when the thread is initially executed.
```

```
<br><input type="radio" id="q3" name="qs3" value="b">B - Main application running the thread.
<br><input type="radio" id="q3" name="qs3" value="c">C - start() method of the thread class.
<br><input type="radio" id="q3" name="qs3" value="d">D - None of the above.

<br>

<br><br>
<input type="submit">

</form>
</body>
</html>

----------examser.php----------
<?php

$q1=$_POST['qs1'];

$q2=$_POST['qs2'];

$q3=$_POST['qs3'];

$sc=0;

if($q1=="a")
  $sc=$sc+10;

if($q2=="b")
  $sc=$sc+10;

if($q3=="c")
  $sc=$sc+10;


echo "<br> <h1> <u> <i>your score=". $sc;
?>



++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

1) DEMO for CIRFIM() function in JS

```
<html>


<head>
<script>


function my()
 {
  var cc = confirm("Do u Like to change Back Color");

  if(cc==true)
  document.body.style.backgroundColor="cornsilk";

 }
</script>

<body >
<button onclick="my()"> Change Color</button>
</body>
</html>
```

++

2) SHOW PASSWORD FILED in JS

```
<html>
<head>

<script>
function f1()
{
var ch=document.getElementById("ch");

        if(ch.checked==true)
        {
        var pw=document.getElementById("pw");
        pw.setAttribute("type", "text");
        }

        else
        {var pw=document.getElementById("pw");
```

```
            pw.setAttribute("type", "password"); }

            }

</script>
</head>


<body>
GIVE USER NAME

<input type="text" id="un">
<br>
<br>
Give password<input type="password" id="pw">
<br>
<br>
<input type="checkbox" id="ch" onclick="f1()">Show Password
</body>
</html>
```

++

3) Functions in JS

```
<html>
<script>

function f1(no)
 {
   var n1=Math.floor(no);

   return(n1);
 }

var cc=f1(45.67);
alert('op'+cc);
</script>


<body >

<p id="para1">hello</p>
```

```
</body>
```

++

4. DEMO for JS BUILT IN FUNCTIONS

```
<html>

<script>
function my()
 {
 var cc = prompt("Enter data");


  var cc=cc.toUpperCase();
  alert("caps"+cc);


 var x1=cc.length;

 var pp=Math.pow(2,6);

var ii=parseInt(cc);

 alert("op"+pp+ x1);



 }
</script>

<body >

<button onclick="my()"> Change Color</button>

</body>

</html>
```

++

The count() function returns the number of elements in an array.

```
echo count($cars);
```

The sort() function sorts an **indexed array** in ascending order.

The rsort() function to sort an **indexed array** in descending order.

The asort() function to sort an **associative array** in ascending order, according to the **value.**

the arsort() function to sort an **associative array** in descending order, according to the value.

the krsort() function to sort an **associative array** in descending order, according to the **key**.

the ksort() function to sort an associative array in ascending order, according to the key.

```
in_array(search value, array)
```

```
array_product(array)
```

The array_pop() function deletes the last element of an array.

```
array_pop($a);
```

The array_push() function inserts one or more elements to the end of an array.

```
array_push(array, value1, value2, ...)
```

```
array_slice(array, start, length);
```

he array_search() function search an array for a value and returns the key.

```
array_search(value, array)
```

```
array_values(array)
```

array_values(*array*)

```
<?php
$array1 = array("color" => "red", 2, 4);
$array2 = array("a", "b", "color" => "green", "shape" => "trapezoid", 4);
$result = array_merge($array1, $array2);
print_r($result);
?>
```

EXAMPLE:

```
<?php

$rno=array("rama","ravi", "raja");

$mk=array(56,78,89,99,40,80);

print('Array values');


rsort($mk);

$l=count($mk);

for ($i=0; $i<$l; $i++)
    echo $mk[$i]."<br>";



$s=in_array(500, $mk);

$pro=array_sum($mk);


array_pop($mk);

array_push($mk, 600,400,450);
print_r($mk);


$x=array_slice($mk,1,4);
print_r($mk);


$emp=array("Peter"=>"85", "Ben"=>"37", "Joe"=>"43");

krsort($emp);
print_r($emp);

$vv= array_search("37",$emp);
$va=array_values($emp);


?>
```