

HTML (Hyper Text Markup Language): -

- i) HTML is not any programming language like C and C++. It is used for web designing for creating the interface or front-end of the website. It is used as an interpreter not compiler.
- ii) It is a static language means through the HTML we can create static website (In the static website we cannot change in the runtime)
- iii) We know that we can create static website in HTML, but though this we can also create Dynamic website (but not fully dynamic).e.g: - When cursor moves on the hyperlink at runtime the color of the hyperlink will be changed automatically.
- iv) In the HTML, we use predefined tags.
- v) **Tag:** - A tag is a specified and bounded area which is used to perform any specified task and already encoded into any browser library known as a tag. Syntax: - <>
- vi) Types of tags
 1. Paired tags(<> opening tag, </ > closing tag)
 2. Unpaired or single tag(there is only single tag contains the information regarding the closing and opening tag such as
, <hr>, <input>)

HTML Element: -

In the HTML syntax, most elements are written with a start tag and an end tag, with the content in between. Tags are composed of the name of the element

For example, a paragraph, which is represented by the `p` element, would be written as

```
<p>In the HTML syntax, most elements are written  
...</p>
```

We call HTML as a markup language, because we use a markup tag that is inbuilt tag, so we call it markup language.

HTML language is not case sensitive.

We use HTML to access the Internet through the web browser and there is no need of any server to execute the HTML page. It works on client side.

The extension of HTML page is **.html** or **.htm** (Windows operating system) but **DOS** operating system does not support .htm extension.

The HTML tag at the top of position is not compulsory, means if we don't use the HTML tag after that the page will be executed.

Example

```
<html>
```

`<body bgcolor=yellow></body>`

Attribute

Property of attribute

- `<h1>This is the maximum size</h1>`

(This is the heading tag and we can use the h1 to h6 for maintaining the size of the heading)

- By default the size of the text is **6 pixel**, we can increase or decrease by the properties.
- **Div:** - We use the div tag in HTML for making the partition of the page or represent the part of the page uniquely known as div tag. It is a paired tag in HTML.
- **Syntax:** - `<div style="background color:gray;height:300px;width:200px">
</div>`
- `` **Ordered list tag** which represent the sequence of the information.
`<ol type = I start=5>` (numbering will be started from 5 in roman way)
`abc
`
- `` **Unordered list tag** which represent information in the form of square ,
Bullet, circle or disc
- `<ul type=circle>` representing the circle in front of list of item.
`xyz
kkk
`
- `<button>click</button>` #for creating the button HTML page
`</html>`

PHP LECTURE: - 2

We can also use Dreamweaver software for making the interface of the website through the HTML tags. If we use the tags in this software, we can get automatic help through this we can easily use the tags. But if we use notepad for creating the page then there is no option for automatic help, that's why during php course work we always prefer Dreamweaver software for creating the webpage so effectively.

There are some important tags in HTML, which will be used mostly through the web designing of the website through HTML.

Username<input type=text name =t1>

Tag attributes Property Represents the ID

BY default textbox contain 20 characters but we can increase the size of the text box by size property.

Password<input type=password name =t2>

Tag attributes Property Represents the ID

The name attribute of the tag is optional. It will be used later when we pass the data in between the pages through php programming.

For creating the radio button the page we use the following syntax: -

```
<input type=radio name=s1 value=abc>Male<br>  
<input type=radio name=s1 value=xyz>Female
```

Note: - If we use the same name in both male and female category then only one will be activated at runtime and in case name of both r different then both will be selected individually.

Form Element: -

There are mainly two types of form elements: -

- 1) GET method
- 2) POST method

See the example of GET and POST methods: -

```
<html>
<head>
  <title>My first styled page</title>
  <link rel="stylesheet" href="abc.css">
</head>
```

```
<body>
<!-- Site navigation menu -->
```

<form action="abc.html" method=GET or POST>

Represents the path of
Next page.

Represents the info in URL
Of the next page through query string

Represents the info
on the next page
But not in URL

```
<ul >
  <li><a href="index.html">Home page</a>
  <li><a href="musings.html">Musings</a>
  <li><a href="town.html">My town</a>
  <li><a href="links.html">Links</a>
</ul>
<div class = "abc">
<p>Welcome to my styled page!</p>
<h1 >My first styled page</h1>
</div>
</div>
</form>
</body>
</html>
```

From the security point of view, we always prefer POST method for transferring the information on the next page.

GET method: -

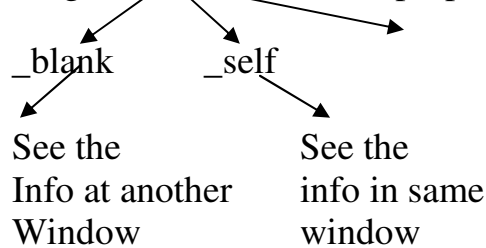
When we use GET request then everyone can see parameters values on the address bar, it is not secure, one more thing we can take or transmit less amount of information(max 40 characters).

POST method: -

When we use POST method, then no one can parameters value on the address bar and its is secure through this we can take request from small as well as large file. It is most probably used into websites.

```
<a href="abc.html" target=_blank>login</a>
```

Target attribute has three properties



So finally a journey of HTML forms starting to end in short description.

HTML :- (Hyper Text Markup Language)

Tags Used In html:-

Tag	Function
<DOCTYPE .../>	Specifies the HTML document's DTD.
<html></html>	Specifies an HTML document.
<head></head>	Specifies an HTML document's header.
<title></title>	Specifies an HTML document's title.
<meta></meta>	Specifies metadata about an HTML document.
<body></body>	Specifies an HTML document's body.

Basic Structure:-

```
<html>
<head>
<title></title>
</head>
<body>
</body>
</html>
```

Note: - The document needs is a "head", which provides information about your document, and a "body", which is the content of the document. Since HTML is nothing if not logical, the head (<head> and </head>) is on top of the body (<body> and </body>).

Tags Used In html:-

Tag	Function
<h1></h1>	Specifies a first level heading.
<h2></h2>	Specifies a second level heading.
<h3></h3>	Specifies a third level heading.
<h4></h4>	Specifies a fourth level heading.
<h5></h5>	Specifies a fifth level heading.
<h6></h6>	Specifies a sixth level heading.
<p></p>	Specifies a paragraph.
	Specifies an unordered (bulleted) list.
	Specifies an ordered (numbered) list.
	Specifies a list element.
	Emboldens text.
 	Places a line-break in text.
<i></i>	Italicises text.
<big></big>	Enlarges text.

<small></small> Makes text smaller.
 Emphasises text.
 Emphasises text, with more emphasis than the em tag.
 Subscripts text.
 Superscripts text.
 Specifies enclosed text as deleted.
 <ins></ins> Specifies enclosed text as inserted.
 <kbd></kbd> The KBD tag defines text that should be entered by a user as in a keyboard. Most browsers display this text in a Monospaced font

<a> Anchor tag
 <caption> </caption> Defines the table caption
 <cite> </cite> Defines the title of work
 <datalist></datalist> specifies a list of pre-defined options for an <input> element
 <link></link> Specifies the connectivity b/w the document and resources
 <meta></meta> provides the metadata of the html document
 <iframe></iframe> specifies web document within the web document
 <select> </select> <select> element is used to create a drop-down list.
 <option></option> specifies the options in the list
 <fieldset></fieldset> specifies the form element in group
 <marquee></marquee> specifies the moving text and images
 <legend></legend> defines the caption of group defined by <fieldset>
 <object></object> defines the embedded object in HTML document
 <param></param> pass the parameter to plug-in that embedded in <object> tag

Special Character Tags:-

Tag	character	Meaning
&	&	ampersand
 		non-breaking space
£	£	pound sterling character
¢	¢	cent character
©	©	copyright character
®	®	registered character
°	°	degree character

±	±	plus or minus character
×	x	multiplication character
÷	÷	division character
<	<	less-than sign
>	>	greater-than sign

Assignment No: - 2

We will create a login page along with registration navigation, and when we press the registration hyperlink then we get the registration in another window. On the right side of the login page we want set of four images.

Php lecture: - 3

CSS (Cascading Style Sheet)

Introduction: -

A style is just the representation of the document on the interface. It contains the settings for the **color, size, type and weight (Boldness) and style (italic, normal) of the text on your document.**

What are the cascading Style Sheets: -

Cascading style sheets are like Dreamweaver templates. They allow creating files that defined how text will look on the entire site. **Css files** can even define the appearance of the page layouts.

Sometimes, styles can be defined in the <HEAD> section of an HTML document. There are mainly two ways for creating the CSS.

Style sheets are saved completely **independent files** and we can **import** the file or can say that uploaded the file on HTML elements, which is the part of website and when we execute the website, then the line of code which created in CSS that tells the browser to go get that style sheet in the same way that pages refer browsers to graphics and other pages.

You can create or implement the style sheet in **between the HTML elements** but that style sheet will only work for the particular page not for all pages of the website.

Anatomy of a Style: -

All the style have the same basic parts, through this we can create the style sheets with full of functionalities that will affect the HTML documents.

SELECTOR: - The *selector* is simply the HTML element that is linked to a particular style or you can create your own selector.

```
P { text-align: 3em }
```

Such as: - `h1 { color: purple }`

In the example given, any text that falls between `<h1>` tags would be made purple.

Multiple style declarations for a single selector may be separated by a semicolon:

```
Selector {property1:value1; property2:value2}
```

Class Selectors

A simple selector can have different classes, (**A class is the combination of different styles which would be used for representing the pages more attractive**) thus allowing the same element to have different styles. For example, an author may wish to display code in a different color depending on its language:

```
code.html { color: #191970 }
```

```
code.css { color: #4b0082 }
```

In the above example has created two classes, **css** and **html** for use with HTML's **CODE** element. The **CLASS** attribute is used in HTML to indicate the class of an element, *e.g.*,

```
<P CLASS=warning>Only one class is allowed per selector.
```

```
For example, code.html.proprietary is invalid.</p>
```

Classes may also be declared without an associated element:

```
.note { font-size: small ;}
```

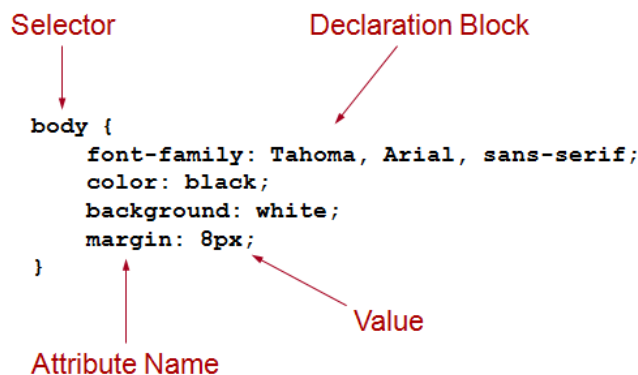
DECLARATION: - This is tells the part between the curly brackets that the browser how to render parts of the page to which the selector has been applied. Properties or Arguments

A *property or Arguments* is assigned to a selector in order to manipulate its style. Examples of properties include color, margin, and font.

Values

The declaration *value* is an assignment that a property receives. For example, the property **color** could receive the value **red**.

CSS Rule



Grouping

In order to decrease repetitious statements within style sheets, *grouping* of selectors and declarations is allowed. For example, all of the headings in a document could be given identical declarations through a grouping:

```
H1, H2, H3, H4, H5, H6 {
  color: red;
  font-family: sans-serif }
```

Style Sheet Syntax: -

Every statement must have a **selector** and **declaration**. The declaration comes immediately after the selector and is contained by a pair of curly brackets { }
H1 {font: Arial, Impact, Sans Serif}

The declaration is one or more properties separated by **semicolons**. That is
H1 {Font-family: Arial; Color: blue}

Some property or argument can take a number of values, as in the font-family example above. The values in the list should be separated by comma and a space.

For example, the following code segment defines the color and font-size properties for H1 and H2 elements: -

```
<HEAD>  
<TITLE>CSS TEMPLATE</TITLE>  
<STYLE SHEET = "text/css">  
H1 {font-size: x-large; color: red}  
H2 {font-size: large; color: blue}  
</STYLE>  
<H1>Simple CSS template</H1>  
</HEAD>
```

In the above example you can create **the style in the HTML elements** with Selector and property, with the syntax of CSS.

Comments

Comments are denoted within style sheets with the same conventions that are used in C programming. A sample CSS1 comment would be in the format:

```
/* COMMENTS CANNOT BE NESTED */
```

```
H1 {color: red} /* red really red!! */
```

Ways for creating the Style Sheet: -

Style sheet information is the textual in the nature and is included either within the HTML document itself or in a separate text file referred to by a HTML document. These three types of styles are embedded, inline and linked or external.

Embedded Styles: -

Embedded styles are placed within the head element and use the style element. Embedded styles affect every instance of the element specified.

```
<html>
<head>
  <title></title>
  <style type="text/css">
    body
    {
      font-family: "Angsana New";
      text-decoration: underline;
      font-style: normal;
      font-variant: small-caps;
      text-transform: lowercase;
      font-weight: 200;
      font-size: large;
    }
  </style>
</head>
<body>
  <p>How we use the Css within the HTML</p>
  <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</body>
</html>
```

If you create the class or can say method with user defined name, then you have to use the dot operator for accessing the properties on a particular HTML tag. Such AS: -

```
<html>
<head>
  <title> </title>
  <style type="text/css">
    .abc
    {
      font-family: "Times New Roman", Times, serif;
      font-size: x-large;
      font-style: normal;
    }
    .navbar
    {
      font-family: ADMUI3Lg;
      border-style: double;
    }
  </style>
</head>
<body>
  <div class="abc">
    <p>This is a paragraph with class abc</p>
  </div>
  <div class="navbar">
    <p>This is a paragraph with class navbar</p>
  </div>
</body>
</html>
```

```

border-width: medium;
text-align: justify;
white-space: normal;
}
</style>
</head>
<body>

<div class = "abc">
<h1> The most important part of the css </h1>
<p>This is the way for creating the css </p>
</div>

<div class = "navbar">
<a href = "default.aspx">home page</a>
</div>
</body>
</html>

```

Inline Styles: -

Inline styles, as you know, they are placed inside elements within the body of an HTML document. Using the attribute "style" any element can be affected, for example: -

```

<h1 style = "align-text: center">
<p style = "background-color: red">
<a href = www.google.com style = "text – decoration: none";>

```

In the three cases above you can see the attribute "style" used with the

Heading (h1)

Paragraph(p)

And anchor(a) elements

In all cases, the style affects only that specific instance of the element. Such as style of h1 would affect only h1 element and p would affect only p element not h1.

External Style Sheets

Once you created the embedded styles, you can create external or linked style sheets. You can move the style information which you embedded in HTML

document transferred into new file. Save the file as a text document with the .CSS extension. Such as abc.css.

In the external style sheets, save your file with extension “.CSS”.

Here we use a special tag is called <LINK>tag. The <LINK> tag goes in the head of any HTML document that you would like to apply your external sheet to. We use the <LINK> tag to refer to the style sheet which you just created on different notepad with the extension of “.CSS”.

<head>

<link rel="stylesheet" href="copy of xyz.css">

</head>

The basic Cascade (means through an example how we can attach the CSS with HTML element)

There is a basic rule of the cascade the external style sheet with the HTML elements.

Suppose you have an external style sheet (**abc.css**), which states the following text, So open your notepad window and type the following text: -

Body

```
{  
padding-left: 11em;  
font-family: Georgia, "Times New Roman", Times;  
color: red;  
background-color: blue  
}
```

h1 (H1 is the HTML element. So there is no need of. (Dot) to access the properties of attributes

```
{  
color: green;  
background-color: purple  
font-size: 15px  
}
```

```

a:hover
{
text-decoration:underline;
font-size:large;
background-color:red
}
.abc
{
padding:1em;
margin: 1em;
border-bottom: 1px
}

```

Note: - (Creation of method or class with attribute and its properties, here we use. (Dot) before the name of method or class and we can access the class anywhere on the HTML according to the Customer's need. Notice that in the

```

<div class = "abc">
<p> here we are defining the paragraph</p>
<h1> represents the size of heading</h1>
</div>

```

Means every tag which represented in div tag will automatically affected. If we attach the class or method with the div tag with following syntax

```

P.green
{
font-style:normal;
text-decoration:underline;
font-size:small;
background-color:red
}

```

Here it means, here we already defined a SELECTOR that is <P> tag followed by a (.), followed by the class name or method.

So now will see the complete HTML PAGE after cascading the style sheet.


```
<html>
<head>
  <title>My first styled page</title>
  <link rel="stylesheet" href="abc.css">
</head>

<body>
<!-- Site navigation menu -->
<ul >
  <li><a href="index.html">Home page</a>
  <li><a href="musings.html">Musings</a>
  <li><a href="town.html">My town</a>
  <li><a href="links.html">Links</a>
</ul>
<div class = "abc">
<p>Welcome to my styled page!</p>
<h1 >My first styled page</h1>
</div>
</div>
</body>
</html>
```

Assignment No: - 3

Create a music site (static site) with the help of HTML tags and CSS. There must be in proper format with all navigations, images, header, and footer. Along with this have to create audio, video, wallpapers, about us and contact us pages with same header and footer

musicworld.com

audio

video

wallpapers

about us

contact us

// to set the position of the div we use float

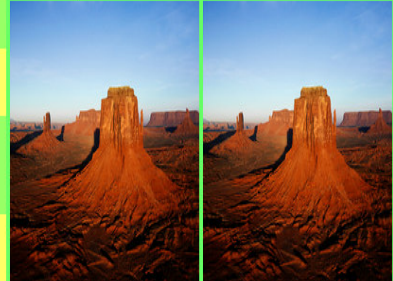
Music:

The best way to feel better in every aspect
through this we can listen various types of songs

Video:

The best way to feel better in every aspect
through this we can represent the nature and feelings on screen

// to set the position of the div we use float



Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

Special for links are that they can be styled differently depending on what state they are in.

The four possible links states are:

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited

- a:hover - a link when the user mouse over it
- a:active - a link the moment it is clicked

Text Decoration

Represents the text with links in HTML

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouse over it
- a:active - a link the moment it is clicked

Background Color

The background-color property specifies the background color for links:

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouse over it
- a:active - a link the moment it is clicked

Styling Text

Text Color

The color property is used to set the color of the text.

Text Alignment

The **text-align** property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

Text Decoration

The text-decoration property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

Border style

none: Defines no border

dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

Double: Defines two borders. The widths of the two borders are the same as the border-width value

groove: Defines a 3D grooved border. The effect depends on the border-color value

ridge: Defines a 3D ridged border. The effect depends on the border-color value

inset: Defines a 3D inset border. The effect depends on the border-color value

outset: Defines a 3D outset border. The effect depends on the border-color value

Border Color

The border-color property is used to set the color of the border.

Border - Individual sides

In CSS it is possible to specify different borders for different sides:

```
border-top-style:dotted;
```

```
border-right-style:solid;
```

```
border-bottom-style:dotted;
```

```
border-left-style:solid;
```

Introduction to JavaScript: -

JavaScript is used to make any form validation on the HTML page. It is a client side scripting. It makes the execution fast of the HTML design on the client side.

JavaScript uses inbuilt functions, events and objects to perform any functions on the HTML page.

Events: - when through the action when we change the internal state of anything is known as events

OR

According to action when we get any reaction is known as events, whenever we press the button of the fan, after pressing the button the fan will be rotated, means whenever we perform the action on button then we get the reaction or can say change the internal state of fan s known as event.

Event Source: - Through the source we generates the events such as button, mouse etc.....

Note: - Java and JavaScript both are different languages.

Q) Is JavaScript Case Sensitive or not?

Answer: - it's neither fully case sensitive nor not case sensitive.

It's (60% case sensitive + 40% non case sensitive)

There are mainly 18 popular events in Java Script and that will be not case sensitive.

Functions are inbuilt in the JavaScript and that will be case sensitive.

Names of 18 events in JavaScript (remember this: - starts always with on keyword)

- | | |
|----------------|---------------|
| 1) onclick | 10) onkeydown |
| 2) ondblclick | 11) onblur |
| 3) onmouseover | 12) onload |
| 4) onmouseout | 13) onunload |

- | | |
|----------------|--------------|
| 5) onfocus | 14) onresize |
| 6) onchange | 15) onsubmit |
| 7) onmouseup | 16) onreset |
| 8) onmousedown | 17) onselect |
| 9) onkeyup | 18) onerror |

JavaScript also uses in built functions like: -

(Remember this: - These are totally case sensitive)

- | | |
|------------------|-------------------------------|
| 1) alert() | 9) Math.random() |
| 2) Confirm() | 10) Math.round() |
| 3) Prompt() | 11) document.getElementById() |
| 4) Date() | 12) document.write() |
| 5) moveBy() | 13) getHours() |
| 6) indexOf() | 14) getMinutes() |
| 7) toUpperCase() | 15) getSeconds() |
| 8) toLowerCase() | 16) window.open() |
| | 17) window.close() |
| | 18) location.reload() |
| | 19) setTimeout() |

JavaScript supports Inbuilt objects like: -

- 1) document
- 2) event
- 3) navigator
- 4) window
- 5) location

Embedding JavaScript in HTML

Loading Scripts: -

Script with src

```
<script src="my-script.js" type="text/javascript"></script>
```

– Purpose

- To define functions, objects, and variables. Define functions, objects, and variables.
- Functions will later be triggered by buttons, other user events, inline script tags with body content, etc.

Script with body content

Syntax: -

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

OR

```
<script language=javascript> //latest syntax for representing the JS in body
```

Purpose

- 1) To directly invoke code that will run as page loads – E.g., to output HTML content built by JavaScript
- 2) Don't use this approach for defining functions or for doing this approach for defining functions or for doing things that could be done in external files.

- 1) Example of **alert ()** function: - To create a pop up window for representing the alert message.

Syntax: -

```
<input type=text onclick="alert('hello')" value=clickme>
```

Inbuilt event

Not case sensitive

Inbuilt function

case sensitive

Note: - When we create the JavaScript with the body of HTML, then it will save with the extension of .html or .htm

When we execute the programming in JavaScript. We have to block the security of Internet Explorer; because it created in VB script and it is the product of Microsoft and does not support JavaScript.

- For creating double click, we have to use : - **ondblclick="alert('hello')"**

Onload and onunload events

Whenever the page will be loaded the onload event will be executed and at the time of unloading of the page, onunload event occurs.

Syntax: -

```
<body onlaod="alert('loading')">
</body>
```

Q) How and with which constraints we create the function in JavaScript?

Answer: - Here in JavaScript we never use int, float, void etc any type of variables and return type for declaring or in defining the function.

Syntax: -

Function show ()

We use function
Instead of int, float,
Void etc.

Represents the name
of the function(custom defiend)

Q) How we multiply, add, divide and subtract two numbers?

```
<script language=javascript>
function add()
{
var a=t1.value; //we use var instead of int,float or void and it is an optional
var b=t2.value;
t3.value = a*b;
}
</script>
```

Enter a:<input type=text name=t1>


```
Enter b:<input type=text name=t2><br>
Result<input type=text name=t3><br>
<input type=button onclick="add()" value=click>
```

Here we call the method

Note: - Through this program we can perform multiplication, divide but not addition for this, we have to convert the string into int, because textbox always Accept string value.

Whenever we wanna see the exception details, then we have to enable the properties in browser (Internet Explorer)

Tools→Internet Options→advanced→choose →display a notification about any script error and disable script debugging (other)

→To represent the current date and Time on the website

We use JavaScript's Date () function, that would be inbuilt function and its case sensitive.

Syntax: -

```
<Script language=javascript>
    document.write(Date());
</script>
```

To represent the dynamic popup dialog box in JavaScript

We prefer **prompt** function.

```
<script language=javascript>
function show()
{
var b=prompt("enter name");
alert(b);
}
</script>
```

```
<input type=button onclick="show()" value=clickme>
```

Q) How we create the stopwatch or digital watch in JavaScript?

Answer: -

```
<script language=javascript>
Function timer()
{
var d=new Date();
h=d.getHours();
m=d.getMinutes();
s=d.getSeconds();
}
```

Operator to allocate the memory

Inbuilt functions

```
document.getElementById('aa').innerHTML=h+":"+m+":"+s;
setTimeout('timer()',1000);
}
</script>
<body onload="timer()">
<div id=aa align=center>Welcome
</div>
</body>
```

to fetch the id of current item

To find the length of the string: -

```
<script language=javascript>
var s="welcome";
document.write(s.length);
OR
Document.write(s.toUpperCase())
</script>
```

Property

To create the random number in JavaScript: -

Syntax: - document.write(Math.random());

To know the current version and name of the browser: -

```
<script language=javascript>
var x=navigator.appName;
object
var y=navigator.appVersion;
```

```
document.write(x);           Fetch the current version of the browser.
document.write("<br>");
document.write(y);
</script>
```

➔ Roll Over image through the JavaScript: -

```
<html>
<title>rollover images</title>
<script language=javascript>

function show1()
{
t1.value="welcome";
}
function show2()           Create the array and used for replacing the image
{
document.images[1].src="c.jpg";
}
function show3()
{
t1.value="welcome to 3rd";
}
function show4()
{
document.images[2].src="b.jpg";
}
</script>






<textarea name=t1>
```

```
</textarea>
```

```
</html>
```

➔ Create water mark in web controls through JavaScript?

```
<html>
<title>
info in textbox</title>
<table>
<tr>
<td><input type=text value="enter username" onfocus="this.value=' '"></td>
</tr>
</table>
```

➔ Set or reset function in JavaScript used with web controls?

```
<html>
<title>
reset and send confirmation</title>

<script language=javascript>
function allowreset()
{
confirm("Are you sure to reset the information?");
}
function allowsend()
{
confirm("Are you sure for sending the information?");
}
</script>

<body>
<form action="mailto:san@gmail.com" onreset="allowreset()"
onsubmit="allowsend()">
```

Enter the name<input type=text name=tt1 value="enter username"
onfocus="this.value=' '>

Enter the password<input type=password name=tt2 value="enter password"
onfocus="this.value=' '>

Enter the address<input type=text name=tt3 value="enter address"
onfocus="this.value=' '>

```
<input type=submit name=sub1>  
<input type=reset>  
</body>  
</html>
```

We will focus on some another important assignments on JavaScript in Regular Classes.

Concepts of

Form Validation in JavaScript: -

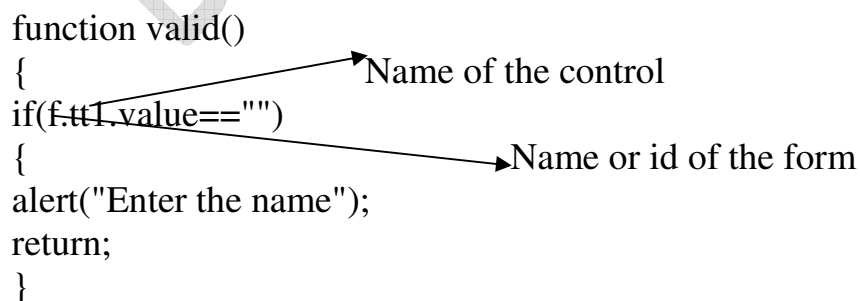
If we put some kind of restrictions on web form for controlling or maintaining the information at a particular level at client side with JavaScript validation known as Form Validation.

Case 1) if we don't enter the username or address etc on web form

Answer:-

```
<script language=javascript>
```

```
function valid()  
{  
  if(f.tt1.value=="")  
  {  
    alert("Enter the name");  
    return;  
  }  
}
```



```
if(f.tt2.value=="")
{
alert("Enter the address");
return;
}
```

we can't execute next condition until this statement will be executed.

```
if(f.tt3.value=="")
{
alert("Enter the username");
return;
}
```

Case 2) if we don't choose any option through the combo box then

```
if(!f.k1.selectedIndex)
{
alert("select the state");
return;
}
```

Property of combo box

Inbuilt function

Case 3) If we match the password with the confirm password through JavaScript (its dynamic language)

```
if(f.tt4.value!=f.tt5.value)
{
alert("password don't match");
return;
}
```

Case 4) If we want that password must be integer then, which code will helpful for us

```
if((f.tt4.value)!=parseInt(f.tt4.value))
{
alert("Password must be integer")
return;
}
```

In built function for converting string into int

Case 5) If we have to choose any of one option through the combo box then

```
if((!f.s1[0].checked)&&!f.s1[1].checked)
{
alert("choose the gender");
return;
}
```

Over all summary of all the possibilities in one program...see carefully..

```
<script language=javascript>
```

```
function valid()
```

```
{
if(f.tt1.value=="")
{
alert("Enter the name");
return;
}
```

```
if(f.tt2.value=="")
{
alert("Enter the address");
return;
}
```

```
if(!f.k1.selectedIndex)
{
alert("select the state");
return;
}
```

```
if(f.tt3.value=="")
{
alert("Enter the username");
return;
}
```

```
if(f.tt4.value=="")
{
```



```
alert("Enter the password");
return;
}
if(f.tt5.value=="")
{
alert("Enter the confirm password");
return;
}
```

```
if((f.tt4.value)!=parseInt(f.tt4.value))
{
alert("Password must be integer")
return;
}
if(f.tt4.value!=f.tt5.value)
{
alert("password don't match");
return;
}
```

```
if((!f.s1[0].checked)&&(!f.s1[1].checked))
{
alert("choose the gender");
return;
}
```

```
if(f.tt6.value=="")
{
alert("Submit the query");
return;
}
f.submit()
}
```

```
</script>
```

```
<form name=f>
<body bgcolor=#66FF66>
<table class=xyz>
<tr>
```

```
<td>name</td>
<td><input type=text name=tt1></td>
</tr>
<tr>
<td>address</td>
<td><input type=text name=tt2></td>
</tr>
<td>state</td>
<td><select name=k1 width=80px>
  <option>haryana</option>
  <option>punjab</option>
  <option>UP</option>
</select>
</td>
</tr>
<tr>
<td>username</td>
<td><input type=text name=tt3></td>
</tr>
<tr>
<td>password</td>
<td><input type=password name=tt4></td>
</tr>
<tr>
<td>Confirm password</td>
<td><input type=password name=tt5 OnFocus="this.value=' '></td>
</tr>
<tr>
<td>email</td>
<td><input type=text name=tt7></td>
</tr>
<tr>
<td></td>
<td><input type=radio name=s1 value=male>male
<input type=radio name=s1 value=female>female</td>
</tr>
<tr>
<td>query</td><td><textarea name=tt6>
</textarea></td>
```

```
</tr>
```

```
</tr>
```

```
<tr>
```

```
<td></td>
```

```
</tr>
```

```
<tr>
```

```
<td></td>
```

```
<td><input type=button class=kbc name=btn1 onclick="valid()"
value=submit></td>
```

```
</tr>
```

call the function here
Onclick event

```
</table>
```

Introducing the concepts of conditional statements, loops, nested loops, super loops, and strings with JavaScript.

➔ W.A.P to find out the greatest number?

```
<html>
```

```
<title>if statement in c</title>
```

```
<body>
```

```
<script language="javascript">
```

```
var a=30;
```

```
var b=20;
```

```
// In JavaScript we don't use any data types  
// with variables, we use VAR.
```

```
if(a>b)
```

```
{
```

```
document.write("A is greater");
```

```
}
```

```
else
```

```
{
```

```
document.write("B is greater");
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Assignment: -

- ➔ W.A.P to find out the highest number in three numbers?
 - ➔ W.A.P to sum of even numbers in between 1 to 20?
 - ➔ W.A.P to print out the Fibonacci series?
-

- ➔ W.A.P to print out the reverse of the number?

```
<script language=javascript>
var n=1476
while(n>0)
{
document.write(n%10);
n=parseInt(n/10);

}
</script>
```

Assignment: -

- ➔ W.A.P to find out the character is vowel or not?
-

Switch Statements

We tested a single variable that came from a drop-down list. A different picture was displayed on screen, depending on the value inside of the variable. **A long list of if and else ... if statements were used.** A better option, if you have only one variable to test, is to use something called a switch statement. To see how switch statements work, study the following code:

```
<html>
```


```
<title>switch statement</title>
<script type="text/javascript">
```

```
function show(a)
{
  var i=parseInt(a);
  switch(i)
  {
    case 1:
    document.write("a");
    break;

    case 2:
    document.write("b");
    break;

    case 3:
    document.write("c");
    break;

    default:
    document.write("none")
  }
}
</script>
<form name="f">
Enter n<input type="text" name="t1" />
<input type="button" onClick="show(f.t1.value)" value="click" />
</form>
```



Note: - In the code above, we place the direct text will be converted into Integer into the variable called **I**. Now through that, we check the different cases, if we get the case then through the break statement, we will jump out from the loop and display the output, so that we can display the correct answer.

Assignment: -

→ W.A.P in which input any character and find out it will be vowel or not?

Switch example: -

```
<title>swith example</title>
<script language=javascript>
```

```
function show(i,a,b)
{
var x=parseInt(i);
var y=parseInt(a);
var z=parseInt(b);
```

```
switch(x)
{
```

```
case 1:
```

```
    c=y+z;
    f.tt4.value=c;
    break;
```

```
case 2:
```

```
    c=y*z;
    f.tt4.value=c;
    break;
```

```
default:
```

```
    document.write("not found");
}
}
</script>
```

```
<body>
<form name=f>
enter i<input type=text name=tt1><br>
enter a<input type=text name=tt2><br>
enter b<input type=text name=tt3><br>
result<input type=text name=tt4>
```

```
<input type=button onClick="show(f.tt1.value,f.tt2.value,f.tt3.value)"
value=click>
</form>
</body>
</html>
```

Concept of Arrays with JavaScript.

As we already know about the concept of array, here we just go through the definition, scope and syntax of array in brief.

Definition: -

Array is the collection of same data types or similar data types; we use subscript for representing the arrays. We can't take different kind of data types in array for this we have to use the concept of structure, we will discuss on later.

Syntax of Array with JavaScript

```
<script language=javascript>
function display()
{
var quote=new Array(4);
quote[0]=5;
quote[1]="B.Sc multimedia";
quote[2]="MCA";
quote[3]="Computer Science";

var y=0;
for(y=0;y<4;y++)
{
alert(quote[y]);
}
}
</script>
<body>
<a href="javascript:display()">Click Here</a>
</body>
```

To allocate the memory to objects

Inbuilt function

absence of event we can use name of JavaScript

Entry control loops(For loop → **initialization+condition+increment or decrement**)

Example: - 1

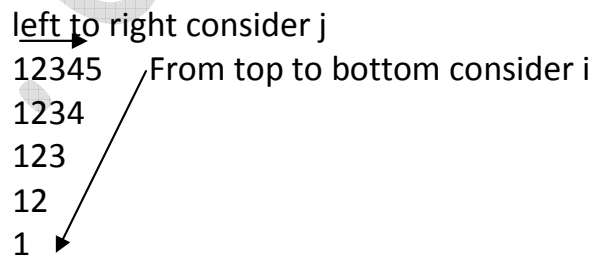
```
<html>
<title>simple for loop</title>
<body>
<script language="javascript">
for(i=1;i<=10;i++)
{
document.write("<br>" +i);
}
</script>
</body>
</html>
```

Assignments: -

- ➔ **W.A.P to print out the sum of 10 natural numbers.**
 - ➔ **W.A.P to find out the factorial of a number?**
 - ➔ **W.A.P to find the power of the number?**
-

Nested For Loops: -

- ➔ **W.A.P. to print the peramid like as: -**



Input: -

```
<html>
<title>peramid</title>
<script language=javascript>
```



```

for(i=5;i>=1;i--)
{
for(j=1;j<=i;j++)
{
document.write(j);
}
document.write("<br>");
}

</script>
</html>

```

Note: - Here we use nested for loop The I loop will operate the J loop and finally the J loop will be executed.

Assignments: -

➔ **W.A.P to print out the various peramid with nested loops.**

18161412	18	1	5
181614	1816	01	54
1816	181614	101	543
18	18161412	0101	5432
		10101	54321

JavaScript Query

J query: -

Its is the advanced version of JavaScript, which is used to make special effects in website such as animation ,calendar, Image gallery, slide show, zooming of the images, show effects, progress bar, slider etc.

Syntax: -

<Script>

```
$(document).ready(function());  
</script>
```

Used for applying the methods or functions

Note: -

- ➔ Here \$ represents the J query
- ➔ Document is the HTML element such as <p>, <div>, <button> etc.
- ➔ Functions are the J query inbuilt functions which should be applied on the HTML tags

Such as: - \$("p").hide() → will be used for hide the paragraph

Note: -

To use the J query function we must attach the J query.js file with HTML page by using the following tag on the HTML page.

```
<script src="jquery.js" type="text/javascript"></script>
```

The extension of the page that will be used with J query saved by **.html**

Example: -1 Hide the content through the J query?

```
<script type="text/javascript" src="C:\Users\Sandy\Desktop\jquery.js"></script>
```

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

```
$(document).ready(function()
```

```
{
```

```
$("#button").click(function()
```

```
{
```

```
$(".abc").hide();
```

```
});
```

```
});
```

```
</script>
```

```
<h2 class="abc">Heading</h2>
```

```
<button type="button">click me</button>
```

➔ **inbuilt function of J query**

Example: - 2 Hide the current content with J query?

```
<html>
<title>this with J query</title>
<script type="text/javascript" src="C:\Users\Sandy\Desktop\jquery.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
$("p").click(function()
{
$(this).hide();
});
});
</script>
<p>click</p>
<p>click1</p>
<p>click3</p>
</html>
```

Example: - 3 Use animate function or method of Jquery.

Output: -



Input: -

```
<html>
<title>animate function in Jquery</title>
<script type="text/javascript" src="C:\Users\Sandy\Desktop\jquery.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
$("button").click(function()
{
```

```
$("#div").animate({left:"100px"},"slow");
$("#div").animate({fontSize:"5em"},"slow");
});
});
</script>
```

```
<button>click</button>
<div style="background-color:red;height:100px;width:300px;position:relative">
<table>
<tr><td>username<input type=text name=tt1></td></tr>
<tr><td>password<input type=text name=tt1></td></tr>
<tr><td><input type=submit name=sub></td></tr>
</table>
</div>
</html>
```

Assignment: -

➔ How to create the calendar through jquery?

(Prepare in regular classes)

Output: -

```
PrevNext
August 2012
Su Mo Tu We Th Fr Sa
    1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

PHP

<http://www.utexas.edu/learn/forms/action.html>

Php was developed by Rasmus Lerdorf in 1994. PHP is the server-side scripting. PHP is a platform Independent language. We can use PHP on any operating system like window, Linux etc.

PHP is used to make the website not for software. One more important aspect regarding PHP, which makes different identification of PHP in market that is PHP is an open source and it always executes on the Local Host on your computer.

OR

PHP is nothing just embedding the concepts of C, C++, JAVA.

In PHP, there are the contribution of **C (70%)** and **C++ & Java (30%)** syntax for making the website.

C → (operators, functions, conditional statement, arrays, nested loops)

C++ → OOPS concepts → Classes, Object, Inheritance, Constructor, Destructor, Polymorphism.

JAVA → Interface, AJAX, Collections, Wrapper Classes.

Web Designing → HTML and JAVASCRIPT etc

What is PHP?

- PHP stands for : - **H**ypertext **P**reprocessor
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use

Here we use different server for different operating system such as: - for window we use the following server like: - **WAMP Server**, XAMP Server, Apache Server

and **IIS (Internet Information Services)** Server, and for LINUX operating system we would like to prefer LAMP Server.

In windows operating system we prefer WAMP Server for executing any PHP statements. We use the following versions of the PHP, see below: -

<p
--
-- } Version 4.0 of PHP, this is the first version for creating and executing the.
?> } PHP statements.

<%
--
-- } We call the second version (8.1) in the world of PHP for supporting and .
%> } executing the statements

<Script>
--
-- } So called version (5.1), with new capabilities in PHP world.
</script>

<?php
--
-- } **Here! The latest version (5.3.0) in PHP for supporting the PHP**
?> } **statements.**

PHP Variables

Rules for PHP variables:

- A variable starts with the \$ sign, followed by the name of the variable
- A variable name must begin with a letter or the underscore character
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- A variable name should not contain spaces
- Variable names are case sensitive (**\$y and \$Y are two different variables**)

String Variables in PHP

String variables are used for values that contain characters.

The PHP Concatenation Operator

There is only one string operator in PHP.

The concatenation operator (.) Is used to join two string values together

The PHP strlen() function

Sometimes it is useful to know the **length** of a string value.

The strlen() function returns the length of a string, in characters.

Significant of PHP tags

This tag gives you the direct output on the client browser, like: -

A simple program with php tags for sum of two numbers.

```
<?php
$a=10;
$b=20;
$c=$a+$b;

Echo"sum=", $c;
?>
```

Note: - We also use Printf("sum=", \$c;) syntax instead of echo.

Here one most important arise here, what are the differences between print, printf and echo in PHP?

Answer: -

Print: -

Print is constructing not a function, which is used to display a single statement on the browser.

Echo: -

Echo is also a construct not a function and it is used to display more than one statement sperating with commas.

Syntax: -

Echo "hello", "
", "welcome";

Printf(): -

It is function not a construct and it is used to display a single statement and we can use format parameters are: -

%d	Represents the Integer
%f	Represents the float
%c	Represents the character
%b	Represents the binary format
%s	Represents the string
%lf	Represents the long float
%ld	Represents the long Integer

PHP uses total 9 super variables: -

Variables in PHP are represented by a dollar sign followed by the name of the variable. The variable name is case-sensitive

\$_GET[]
\$_POST[]
\$_REQUEST[]
\$_SERVER[]
\$_SESSION[]
\$_COOKIE[]
\$_PHP[]
\$_FILES[]
\$_ENV[]

The \$_GET Variable

The predefined **\$_GET variable** is used to collect data in a form with method="get"

Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send.

Step:-1 Create a form in HTML with **get method**

Step: -2 Create abc.php , we use the \$_GET variable or container to collect the data from the HTML form

Function: -

It is nothing just the block of statements which are used to execute them one by one, when we make a call to function.

Syntax: -

Function fname (parameters)

```
{  
-----  
-----  
}
```

There are various ways for calling the function for executing them: -

1. Simple call
2. Call by value

Benefits of function: -

- 1) It avoids the repetition of code.
- 2) It is less time consuming.
- 3) It reduces the complexity.

Simple program which represents the simple **function call**: -

```
<?php
function show()
{
echo "hello";
}
show();
?>
```

```
<?php
function add($a,$b)
{
echo $a+$b;
}
add(3,5);
?>
```

What is a Cookie?

Steps for creating and define the scope of cookies

setcookie() : - It is used to set the cookies

isset(): - It is used to find out that cookies has been set or not

Concepts of OOPS (Object Oriented Programming)

It is a process through which we can divide the complicated program in small blocks or modules and allocate the partition memory for each data member and member function. We can make template on our interest by using the modules.

Class: -

It is the collection of data member and member function.

Object: -

It is a data item which contains the instance of the class.

Syntax: -

Object name=new class name();

➔ \$a = new abc();

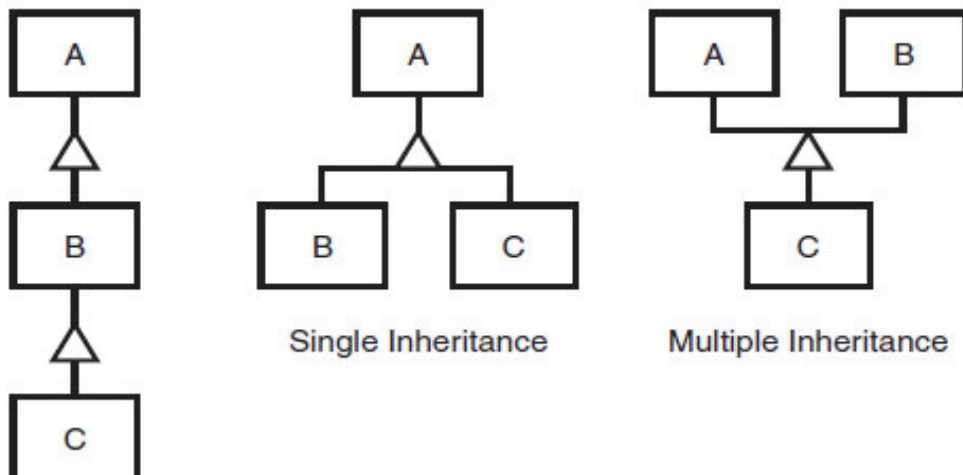
```
<?php
class student
{
var $name;
var $rollno;
var $marks;
function record($nam,$roll,$ma)
{
$this->name=$nam;
$this->rollno=$roll;
$this->marks=$ma;
}
function show()
{
echo "name=", $this->name, "<br>";
echo "rollno=", $this->rollno, "<br>";
echo "marks=", $this->marks, "<br>";
}
}
$s=new student();
$s->record("aman",10,70);
$s->show();
?>
```

Assignment: -

W.A.P with banking feature includes deposit and withdraws?

Inheritance: -

Inheritance allows us to create hierarchy between the classes using subclasses. A subclass inherits attributes and function from its super class. With inheritance you can build on and add to the existing classes. From the simple base class, you can drive more complex and specialized classes as the need arises. This makes your code more reusable, which is one of the most important advantages of object-oriented approach.



Multi level Inheritance

There are mainly two types of inheritance supported in PHP

1) Single level

Super class

Base Class

2) Multi Level

Super class

Intermediate class

Base class

W.A.P of single level Inheritance with PHP tags?

```
<?php
```

```
class deptt  
{  
var $dept_name;  
var $dept_loc;
```

```
function depart($dep,$loc)
{
$this->dept_name=$dep;
$this->dept_loc=$loc;
}
function show()
{
echo"dept_name=",$this->dept_name,"<br>";
echo"dept_loc=",$this->dept_loc,"<br>";
}
}
class emp extends deptt
{
var $sal;
function empsal($a)
{
$this->sal=$a;
}
function show()
{
echo"sal=",$this->sal,"<br>";
deptt::depart();
deptt::show();
}
}
$ss=new emp();
$ss->depart("OOPS","Chandigarh");
$ss->empsal(30000);
$ss->show();
?>
```

Assignment: -

W.A.P with the concept of Inheritance create country (base class) and state (derived class) and explain how we derived the properties of base class with single Inheritance.

Multiple Inheritances: -

Some OO languages support Multiple Inheritance but PHP does not. This means each class can only inherit from one parent class. No restriction how many children can share a single parent.

But through the concept of Interface we can use the Multiple Inheritance in PHP, but through the classes it's not possible.

Interface: - This concept generated from JAVA. It's act like a class for multiple inheritance in PHP. All the functions presents in Interface are by default abstract (Those functions which have no body are known as abstract function).

Syntax: -

```
Interface abc
{
    Function set();
}
```

Note: - When class uses the Interface then it creates all functions into itself which will be declared in Interface and will provide the body to the function in class part or can say that we will define the Interface in class.

We use IMPLEMENTS as a keyword to use Interface.

Syntax: -

```
Class xyz implements abc
```

We can implement more than one Interface at a time.

A simple example, which represents the concept of Interface

```
<?php
interface abc
{
function set($n);
function show();
```



```
$this->empno=$eno;
$this->ename=$enam;
$this->salary=$esal;
}
function show()
{
echo "empno=",$this->empno,"<br>";
echo "ename=",$this->ename,"<br>";
echo "salary=",$this->salary,"<br>";
}
function display()
{
echo "hello";
}
}
}
$s1=new person();
$s1->set(10,"aman",20000);
$s1->show();
$s1->display();
?>
```

State Management: -

State Management: -

Definition: -

To manage the state of **stateless** pages, we use the State Management technique.

There are **two types** of State Management: -

- 1) Client side State Management
- 2) Server Side State Management

Client Side State Management: -

- 1) Client side management is used to use the client resources.

- 2) It's **faster** as compared to server side state management.
- 3) **Not secure** compared to server side state management.

There are **two types** of client side state management: -

- 1) Cookies
- 2) Query String

Server Side State Management: -

- 1) It is used to use the server resources only.
- 2) **Slower** as compared to client side state management
- 3) **More secure** as compared to client side management.

There are **two types** of server side state management: -
(For retaining the data)

- 1) Applications
- 2) Session

Cookies: -

Key Terms or properties regarding cookies: -

- 1) To store data in text files and that text file reside on the client machine.
- 2) We can store only string values not object.
- 3) **Scope:** - Multiple pages.
- 4) By **default cookies are temporary** and temporary cookie stored in browser only.
- 5) But we can **create permanent cookies**, we can specify expiry date and time according to the user.
- 6) By default cookies are URL specific means gmail is not supported in yahoo)
- 7) Per site 20 cookies can created in one website
- 8) Initial size of cookie is 50 bytes.
- 9) Maximum size of cookie is 4096 bytes.

When you create a cookie, using the function setcookie, you must specify three arguments. These arguments are setcookie(name, value, expiration):

- 1) **name:** The name of your cookie. You will use this name to later retrieve your cookie, so don't forget it!
- 2) **value:** The value that is stored in your cookie. Common values are username(string) and last visit(date).
- 3) **expiration:** The date when the cookie will expire and be deleted. If you do not set this expiration date, then it will be treated as a session cookie and be removed when the browser is restarted.

Step1) Create a simple login page with following HTML tags: -

```
<html>
<title>cookie</title>
<form action="result.php">
username:<input type="text" name="t1" /><br />
password:<input type="text" name="t2" /><br />
  <input type="submit" value="submit" />
</form>
</html>
<a href="view1.php">welcome</a>
```

Note: - If we want to create page more attractive, we can implement CSS and JavaScript as per the requirement of the customer.

Step2) Here we create cookies with super global variable (setcookie) with Php.

```
<?php
$nm=$_GET['t1'];
setcookie("user","$nm",time()+10);
?>
<h1>welcome Mr.<?php echo $nm; ?>
</h1>
<hr />
<a href="view1.php"> view the view1</a>
<a href="visit.php"> visit</a>
```

Step 3) here we check the validity, accessibility and reliability.

```

<?php
if(isset($_COOKIE["user"]));
{
echo "<h1>welcome...in view1" .$_COOKIE["user"];
}
?>
<hr />
system and technology<hr />
<a href="view2.php">visit the view2</a>

```

Step 4) Create one more page of Php here we check the status of cookies.

```

<?php
if(isset($_COOKIE["user"])) {
echo "<h1>welcome.....in view2" .$_COOKIE["user"];
}
else {
echo "<h1>cookie expried";
return; }
?>
<a href="logout.php">logout</a>

```

Step 5) At the last step create logout option.

```

<?php
$var="";
setcookie("user",$var,time()+10);
Include("login.html");
?>

```

We can create sub key for cookies

Drawback: -

- 1) Size limitation
- 2) User can manipulate the cookies.
- 3) User can enable or disable the cookies.

Basic code for creating cookies with the support of super variable in Php: -