CSE2007
Web Technologies

Module No. 1- Introduction to Web Design

Introduction to WWW and Internet:

Web: Collection of "*Electronic resources".* a collection of interconnected documents and other resources.*For ex:* e-journal, e-book, E-mail,

PDF document, doc document ,Blog…

World Wide Web Consortium (W3C):international body that maintains Web-related rules and frameworks.

-Maintains web standard

-A group organization

Cond.

- Network: collection of computer or other devices connected to allow data sharing. LAN(Local Area), MAN(metropolitan Area), WAN(Wide Area)
- Internet:

  - International Network

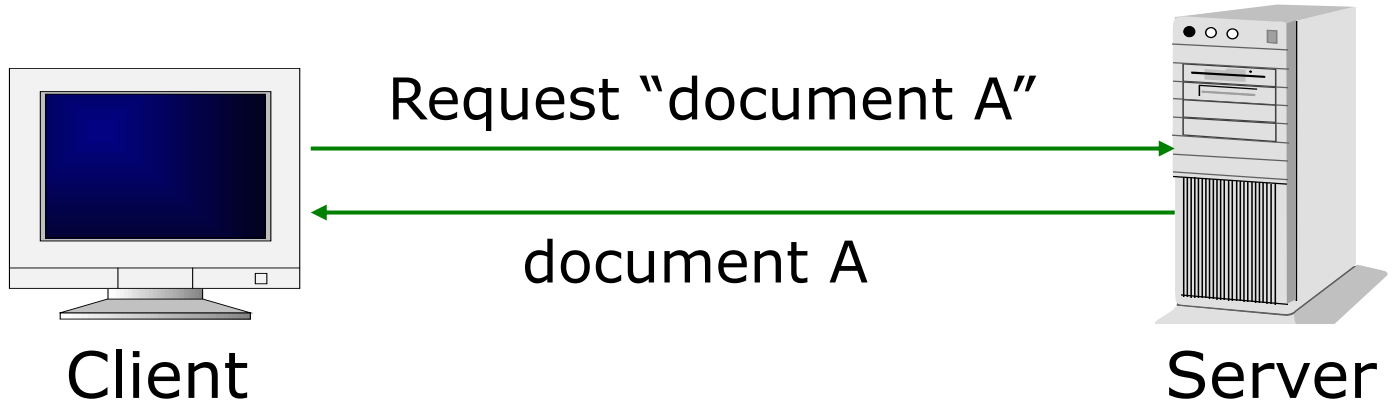
  - connection between network

Internet vs. Web

- The Internet: a inter-connected computer networks, linked by wires, cables, wireless connections, etc.
- Web: a collection of interconnected documents and other resources.
- The world wide web (WWW) is accessible via the Internet, as are many other services including email, file sharing, etc.

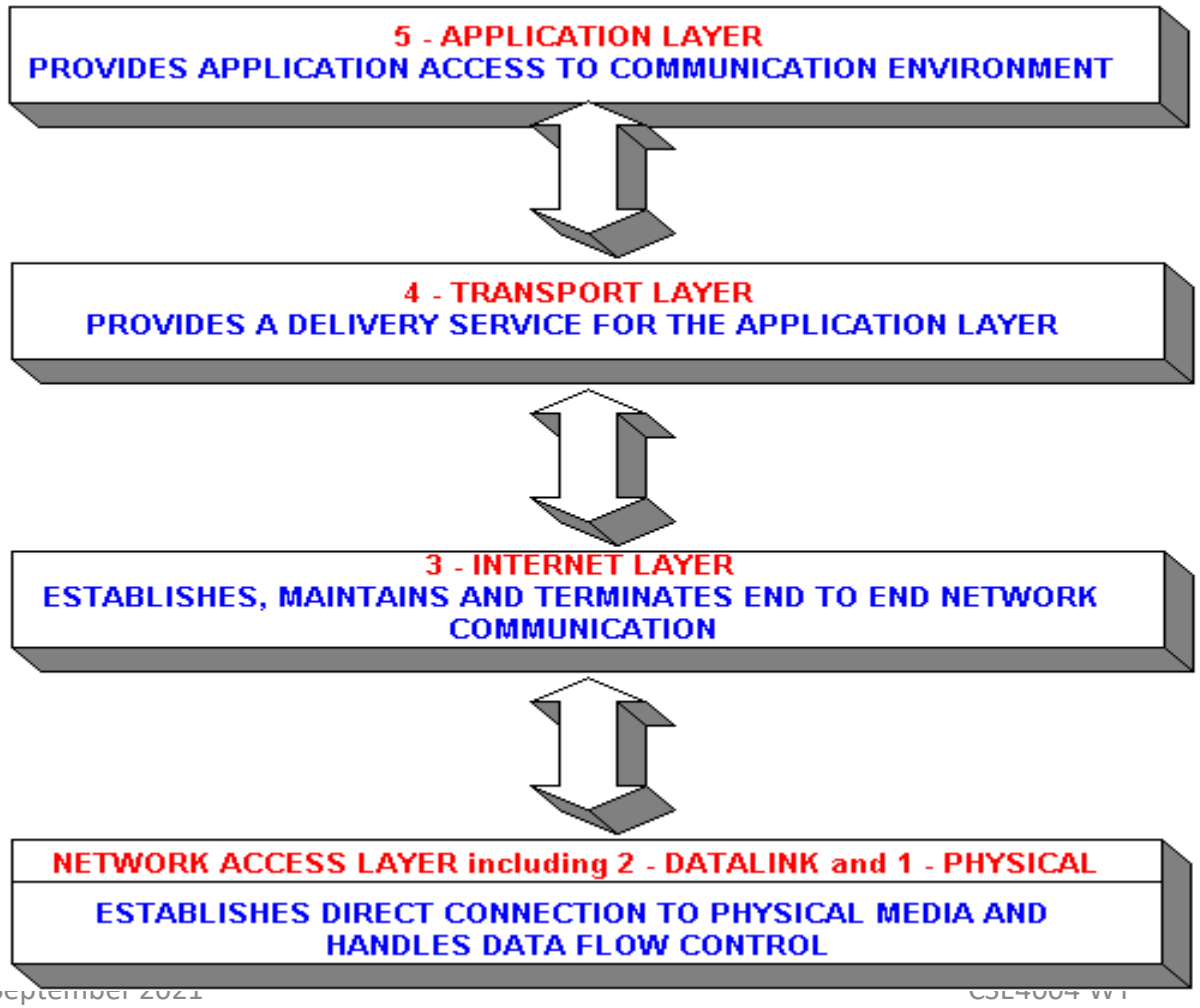
Web Essentials

- Client: web browsers, used to surf the Web
- Server systems: used to supply information to these browsers
- Computer networks: used to support the browser-server communication



- Through communication protocols
- A communication protocol is a specification of how communication between two computers will be carried out
  - IP (Internet Protocol): defines the packets that carry blocks of data from one node to another
  - TCP (Transmission Control Protocol) and UDP (User Datagram Protocol): the protocols by which one host sends data to another.
  - Other application protocols: DNS (Domain Name Service), SMTP (Simple Mail Transmission Protocol), and FTP (File Transmission Protocol)

TCP/IP Protocol Suites



HTTP, FTP, Telnet, DNS, SMTP, etc.

TCP, UDP

IP (IPv4, IPv6)

Web Site

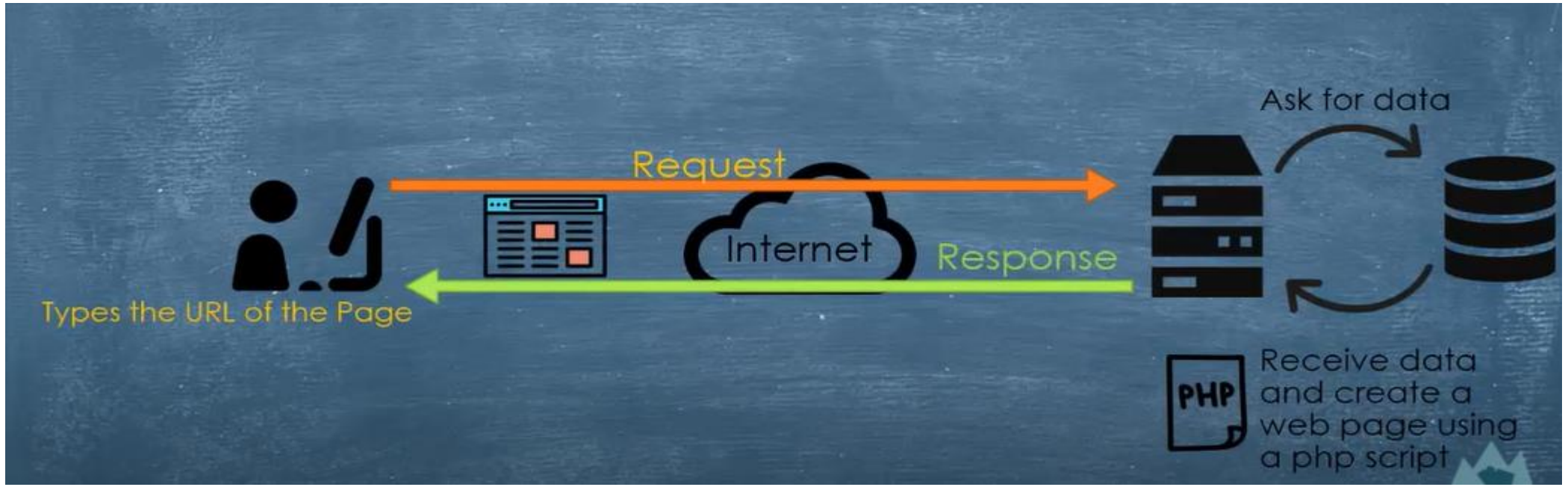
- Collection of web pages (text, video, audio..)

Two types

- Static web page
- Dynamic web page

condt


HTTP Request
Internet
Response
ypes the URL of the Page


Ask for data
Request
Internet
Response
Types the URL of the Page
PHP
Receive data
and create a
web page using
a php script

condt

Dynamic web page:

- CGI/Perl: Common Gate Way Interface (*.pl, *.cgi)
- PHP: Open source, strong database support (*.php)
- ASP: Microsoft product, uses .Net framework (*.asp)
- Java via JavaServer Pages (*.jsp)

Condt

- Static web page :
- HTML stands for HyperText Markup Language
  - It is a text file containing small markup tags that tell the Web browser how to display the page
- XHTML stands for eXtensible HyperText Markup Language
- CSS stands for Cascading Style Sheets
  - It defines how to display HTML elements

HTML

- GML-General Markup Language
- HTML- Hyper text Markup Language
- HTML 1.0,2.0….5.0
- XML+HTML=XHTML
- Developed by W3C
- Webplatform.org- web related any information.

condt

- Tags(Elements):

Two type:

- Paired tags(tag which is having opening/closing)

<html>

…

…

</html>

- Non-paired tags(tag which is having only opening tag)
-
- <hr>
- <img>

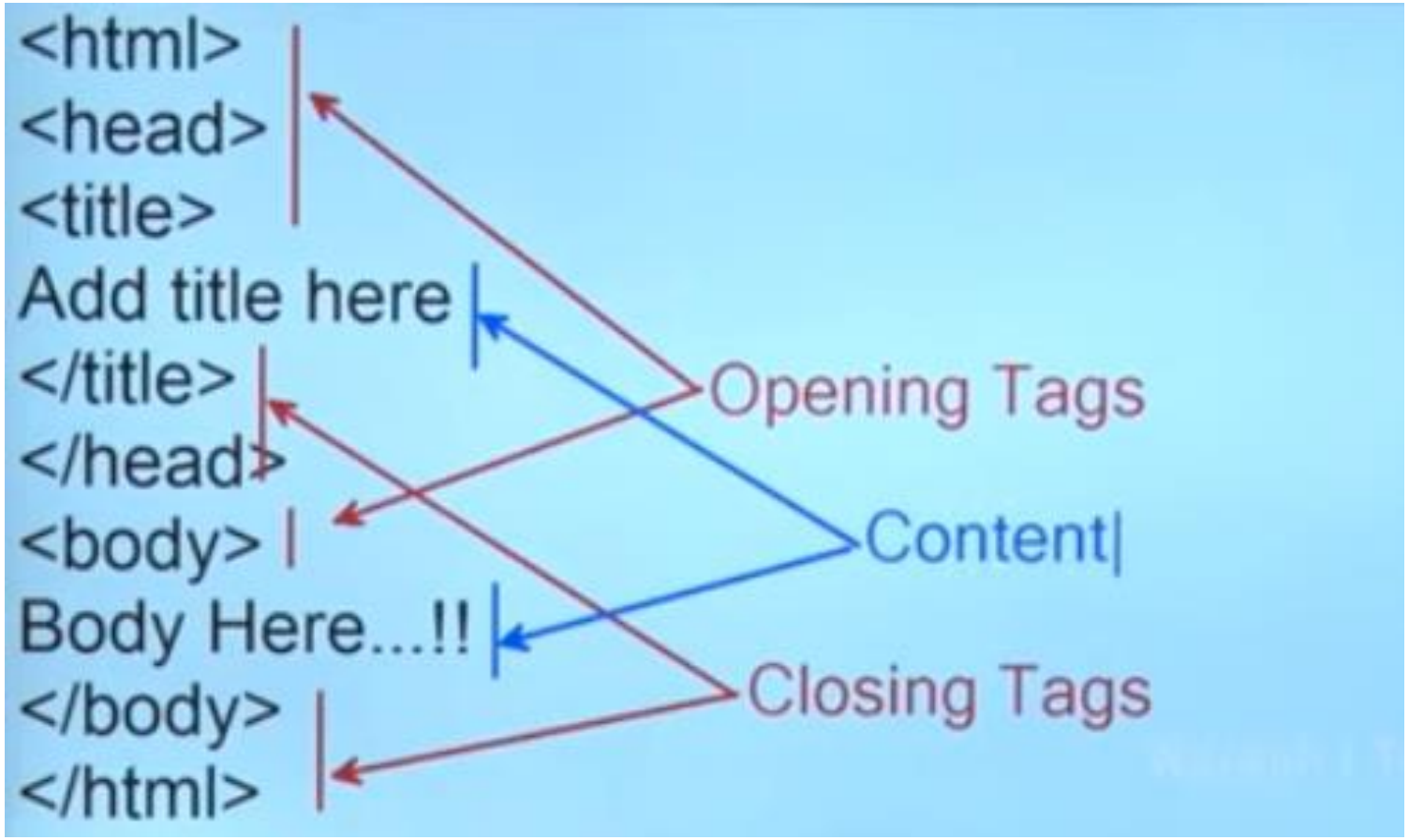
Paired/Non-Paired

<html>	begins html file	*paired*
<head>	first section of html file	*paired*
<body>	second, main section of html file	*paired*
<title>	within HEAD - title goes in top bar	*paired*
<p>	paragraph	*paired*
** **	line break	*single*
<hr>	horizontal rule	*single*
<h1> <h2>	headings	*paired*
<a href=" ">	link	*paired*
<img src=" ">	image	*single*
<pre>	preformatted - displayed like the source	*paired*
<blockquote>	separate, indented text	*paired*
<!-- -->	comments	*single*
<i>	italic physical style	*paired*
<b>	bold physical style	*paired*

condt

<tt>	teletype physical style	*paired*
****	emphasis logical style	*paired*
****	strong logical style	*paired*
<tt>	code logical style	*paired*
<cite>	citation logical style	*paired*
** **	ordered list	*paired*
** **	unordered list	*paired*
** **	list item	*can be paired*
<dl>	definition list	*paired*
<dt>	term name (left aligned)	*can be paired*
<dd>	term definition (indented)	*can be paired*
<a name=" ">	anchor	*paired*
&...;	escape sequences	*single*
<table>	table	*paired*
<tr>	table row	*paired*
<td>	table cell	*paired*

condt


<html>
<head>
<title>
Add title here
</title>
</head>
<body>
Body Here...!!
</body>
</html>
Opening Tags
Content
Closing Tags

Condt

- Title page-> favorite Icon + Title
- Facebook-> f+Title
- Paragraph: used to divide text into multiple paragraph
  - <p>
  - paired tag
  - Attribute(align) parameters(left, right, center, jutify)
  - <p align='right'> or <p align="right">
  - default – Left align

comments

- Not execute
- Single /multi line comments
- <!- - comment -- >
- Not used in Title part
- Apply to body

condt

- HTML
- - Version information- Instruction to web browser .what version of web page is written in.
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN“ "http://www.w3.org/TR/html4/loose.dtd">
- - <!doctype html>
- lang –Specify primary language of web page .
- <html lang=“en”>

Condt (logical section)

- -head section- <Title>/<link>/<meta>/<script>/<style>
- <title >…..</title> paired
- <link> or <link/>……self tag /forcefully closed/singular/empty
- Ex: <link href="path of the resource" rel="relation to the html doc" type="type of the source">
- <link href="she.jpg" rel="icon" type="image/ico">
- Path of the source –global path (no location info)
- <link href="D:\hht\she.jpg" rel="icon" type="image/ico"> -local
- "file:///D:\hht\she.jpg (///forward slash)
- png, gif, ico,svg –recommended format(16X16) (32 X 32)

Condt(logical tag)

- <meta>- data related to search engine
- Non-paired tag
- <meta > information about your web page
- 2 attributes
- 1)name 2) content
- <meta name=“keyword” content=“html,hai”>
- <meta name =“author” content=“anu”>
- <meta name=“description” content=“hai r u”>
- <meta name=“viewport” content=“width=device-width, initial-scale=1.0”>
- User visible area of web page is varies with devices
- Setting the viewport to make your website look good on all devices

condt

- <meta charset="UTF-8">
- Unicode transformation format
- <meta name="Title" content="Oracle Java Technologies | Oracle">
- <script>
- <script type='text/javascript' language='javascript'>

  statements;

  statement;

  </script>

cond

- <style type=“text/css”>
  statement;
  statement;
- </style>

Body section(paired tag)

- <body> </body>
- Attributes parameters

  bgcolor colorname/code

  background imagepath

  text colorname/code

Attributes – adding strength to tag

Style='background-repeat:no-repeat>

Color and Hex code:

Color	Name	Hex Code	RGB Code
	White	#FFFFFF	rgb(255, 255, 255)
	Silver	#C0C0C0	rgb(192, 192, 192)
	Gray	#808080	rgb(128, 128, 128)
	Black	#000000	rgb(0, 0, 0)
	Red	#FF0000	rgb(255, 0, 0)
	Maroon	#800000	rgb(128, 0, 0)
	Yellow	#FFFF00	rgb(255, 255, 0)
	Olive	#808000	rgb(128, 128, 0)
	Lime	#00FF00	rgb(0, 255, 0)
	Green	#008000	rgb(0, 128, 0)
	Aqua	#00FFFF	rgb(0, 255, 255)
	Teal	#008080	rgb(0, 128, 128)
	Blue	#0000FF	rgb(0, 0, 255)
	Navy	#000080	rgb(0, 0, 128)
	Fuchsia	#FF00FF	rgb(255, 0, 255)
	Purple	#800080	rgb(128, 0, 128)

Condt(-body section)

- Font Tag
- Format the text-Size ,color, style
- Paired tag
- ……
- Attributes : Parameters:
- color Any color name or hexadecimal value
- Size 1 – 7
- Face arial, Tahoma, etc…

condt

- Bold
- ….
- Italic
- …..
- Strike off tag
- <s>….</s>
- Quotation
- <q>…</q> “ “
- Subscript
- <sub>…..<sub>
- Superscript
- <sup>…<sup>
- Abbreviation
- <abbr>…..</abbr> <abbr title=“ “>text </abbr> Toolt Tip

condt

- List Tag- collection of more than one item

Two type

- Ordered list (numbered) (sequence)
- Unordered list (bullet)

- apple
- Orange

condt

Unordered list

 Square, Circle, Disc, None

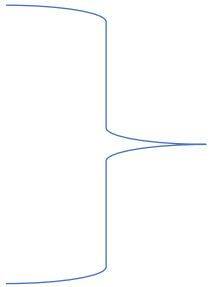
apple

orange

Anu

Hi

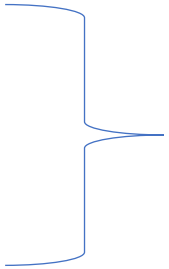
Hello



cond

- Ordered List
- Number and Alphabet 1,2,3.. /A,B,C../a,b,c…/I,II,III../i,ii,iii..

1. Anu
2. Hi
3. Hello



Stand alone-unpaired tag

-
 break the row
- <hr> create horizontal line
- <img> insert image on web page

  src=“path of image “

  alt-”Alternative” ->Text information

  width-width of image in pixel

  height-height of the image in pixel

  Align-Left/right (it wont work in HTML 5)

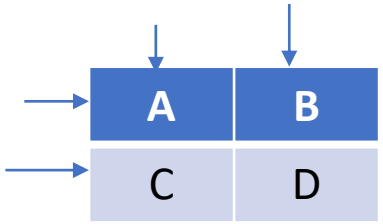
<img src=“ she.jpg” alt=“Hello” width=“300” height=“300” align=“left”>

Cond

- Anchor - used to set hyperlink
- Link –which is used to connect another web page
- <a> Tag
- -href
- - Title
- Text

condt

- Marquee
- <marquee> text </marquee> // right to left
- behavior=alternate
- <marquee behavior="alternate" bgcolor="red" height="40">
- Table
- <Table> Tag , <TR> -Row <TD> -Data



- It ll display without any border
- <caption> </caption> --→ Title of the table (center)

condt

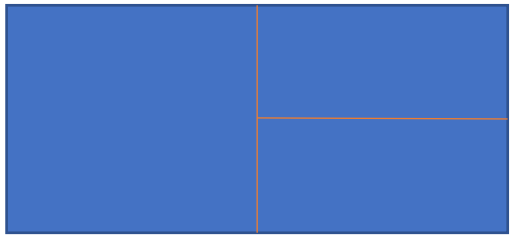
- Merge two data/ two col
- Colspan-merge two column

A	B

- <table>
- <tr>
- <td> A></td>
- <td> B></td>
- </tr>
- <tr>
- <td colspan=“2”> C </td>
- </tr>
- </table>

condt

- Rowspan-merge rows
- <table>
- <tr>
- <td rowspan="2">A</td>
- <td> B</td>
- </tr>
- <tr>
- <td> C</td>
- </tr>
- </table>
- <th>--table head



Forms

First name:

John

Last name:

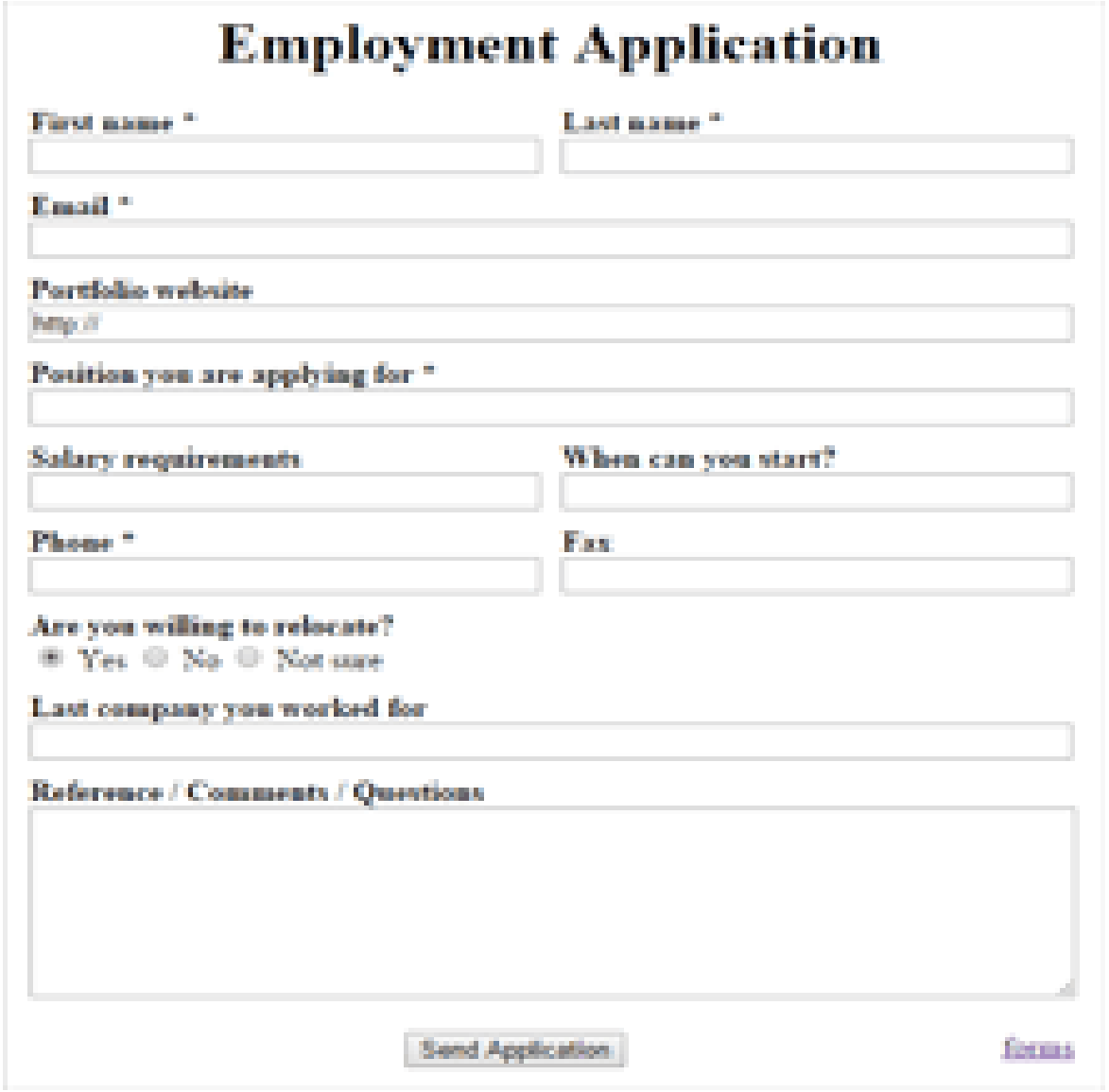
Doe

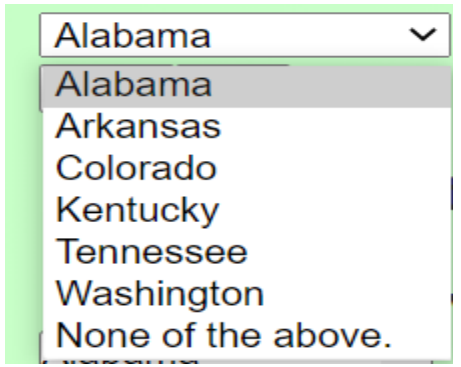
Submit

-used to collect user information

```
<form>
.
form elements
.
</form>
```

condt


Employment Application
First name *
Last name *
Email *
Portfolio website
http://
Position you are applying for *
Salary requirements
When can you start?
Phone *
Fax
Are you willing to relocate?
Yes
No
Not sure
Last company you worked for
Reference / Comments / Questions
Send Application


Alabama
Alabama
Arkansas
Colorado
Kentucky
Tennessee
Washington
None of the above.

Condt

- Text Fields and Password

- <input type=“text” name=“fna” size=“4” maxlength=“20”>

< input type=”password” name=“pwd” size=“4” maxlength=“20”

cond

- Radio button

- <input type=“radio” name=“gender” >female
- <input type=“radio” name=“gender” >male
- Checkbox-multiple input

condt

- <input type=“checkbox” name=“hobbr” >singing
- <input type=“radio” name=“hobb” >dancing
- Textarea:
- Summary about u <textarea rows=“5” cols= “5” name=area>
  </textarea>

condt

- Select- select list of element

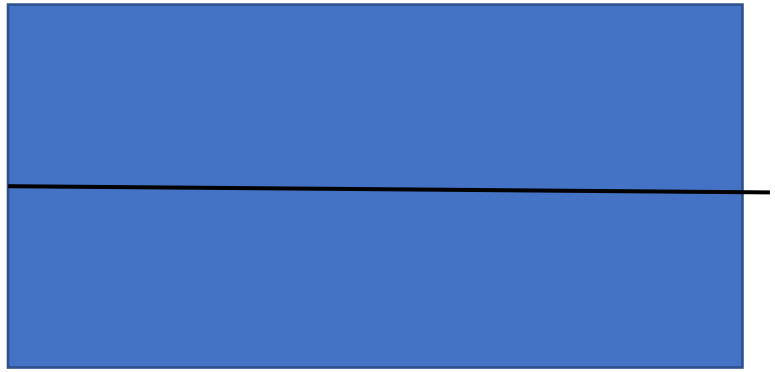
- <select size=3> size is optional // default set to one
- <option>C</option>
- <option>C++</option>
- <option>java</option>
- </select>

condt

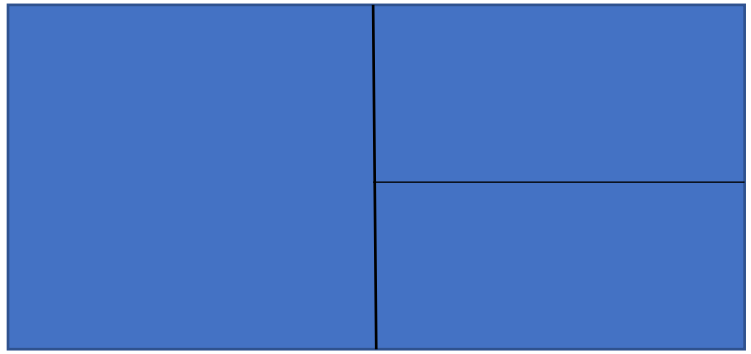
- Upload- file upload
- <input type=file name=file accept=“image”>
- <h1 style="*text*-align:*center*;">

Frames:

- Dividing web page row wise/col wise
- <frameset rows=“50%,50%”>
- <frame src=“ source”>
- <frame src=“source”>
- </frameset>



- <frameset cols=“50%,50%”
- <frame src=“src”>
- <frameset rows=“50%,50%”>
- <frame src=“src”>
- <frame src=“src”>
- </frameset>
- </frameset>



CSS-CASCADING STYLE SHEETS

- Style to web page
- Three Type:

-External CSS – Two files (.css & .html(link href="css file path") rel=stylesheet)

- writing all style in separate file . Linking file into html file

- inline CSS– In each tag(Style)attribute

- Internal CSS–.html (head tag itself) <style>……..</style>

condt

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

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

</body>
</html>
```

"mystyle.css"

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

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

Internal CSS

- 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;
}
</style>
</head>
<body>

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

</body>
</html>
```

Inline CSS

- To use inline styles, add the style attribute to the relevant 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>
```

Two CSS in one webpage

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
```

```
h1 {
  color: navy;
}
```

CSS :color

- background color background-color:red
- Foreground color:red

- CSS :Align
- Text Alignment :text-align=center/left/right

- CSS : margin
- Margin-left:40px
- Margin-right:40px Margin:40px
- Margin-down:40px
- Margin-top:40px

condt

- CSS : padding
- padding-left:40px
- padding-right:40px
- padding-down:40px
- padding-top:40px

condt

- CSS: Display:
- Display: None // hide the element/tag in web page
- Display: block //display the element in a separate block
- Display: inline //display the element with other content
- Display: inline-block // merge with other element and leave some space above
- Display :inherit //inherit the propriety of parent property

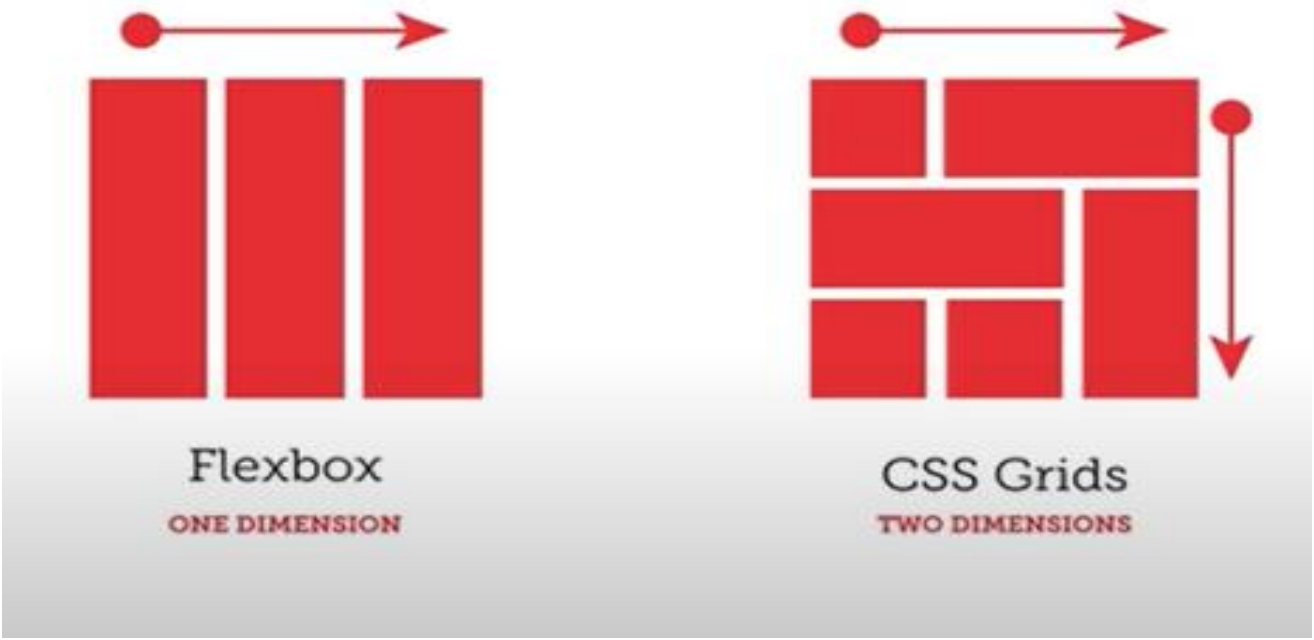
condt

- CSS : Background:
- Background-color:colorname
- Background-image=url(“”)
- Background-repeat: no-repeat/repeat
- Background-position:left/right/center/left top/left bottom/ right top/right bottom

condt

- {border-style: dotted;}
  {border-style: dashed;}
  {border-style: solid;}
  {border-style: double;}

LAYOUT



Module-2 JAVASCRIPT

- keep in <head> section -<script> tag
- Keep in <body>section-<script> tag
- Keep in outside HTML tag
- Client side scripting language
- Add programming to web page
- Add functionality
- Visual Studio code-Browser(check JavaScript is enable or not)

Condt

- 2 Ways to add Java Script

---Embed in HTML

---External JS File (programname.js)

condt

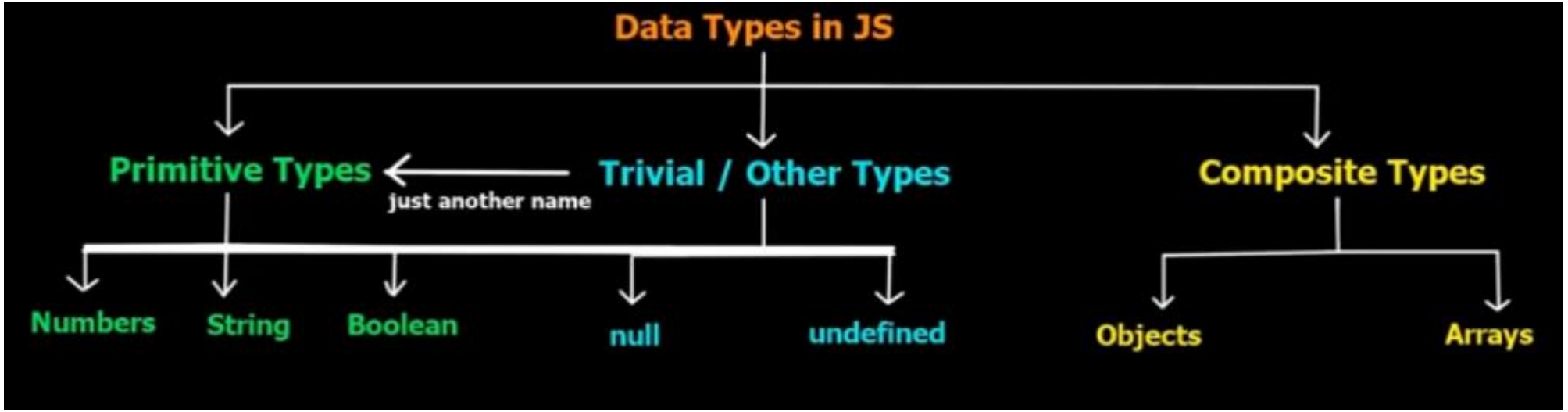
```
1   <html>
2       <head>
3           <title>My title</title>
4           <script type="text/javascript">
5               document.write("<h1>Heading 1</h1
6           </script>
7
8       </head>
9
10      <body>
11
12      </body>
13
14  </html>
15
```

condt

```
1   <html>
2       <head>
3           <title>My title</title>
4
5
6       </head>
7
8       <body>
9
10      </body>
11
12  </html>
13  <script type="text/javascript">
14      document.write("<h1>Heading 3</h1>");
15  </script>
```

Variables

- Variable- used to store data and information



- var a; int a;
- var a=“sssh”;

condt

- Using var
- Using let
- Using const

```
var person = "John Doe", carName = "Volvo", price = 200;

Var person="John Doe",
carName = "Volvo",
price = 200;
```

condt

Variables defined with let cannot be Redeclared.
Variables defined with let must be Declared before use.
Variables defined with let have Block Scope.

condt

```
<html>
    <head>
        <title>Variables & Datatypes</title>
        <script type="text/javascript">

            // int x = 5;

            // JS is LOOSELY typed.
            // JS is DYNAMICALLY TYPED
            var num = 16;  // Number
            var Name = "Tanmay Sakpal";    // String
            var flag = true; // boolean


            document.write(num); // printing on the browser
        </script>
    </head>
    <body>

    </body>
</html>
```

- Int a, int b;
- Int a,b;

- var a,b;
- a=10.1;

condt

```
<html>
<head>
<script>
var n=10;
var n1="sheela";
var n3=false;
//document.write(n1);
alert(n1);
</script>
</head>
</html>
```

Array

- Three Ways:

1.Literal:

- Var Arrayname=[“rav”,”anu”,”redd”];
- Var Arrayname=[1,2,3];

2. Regular

Var Arrayname=new Array();

Arrayname[0]=“rav”;

Arrayname[1]=“anu”;

Arrayname[2]=“redd”;

2: Regular:

```
var myNames=new Array();
myNames[0]="Ravi";
myNames[1]="Smith";
myNames[2]="Raju";
```

Condt

- 3.condensed
- Var Myarray=new Array(“rav”,”anu”,”hhh”);

Object

JavaScript objects are written with curly braces {}

- Variable have multiple property.
- Var humen {

name: hhhh,

age: 21,

address: hhgghh,

}

Operators
Arithmetic operator:

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

Assignment operator

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
**=	x **= y	x = x ** y

condt

JavaScript String Operators

The + operator can also be used to add (concatenate) strings

```
Var a=10;
Var  str=20;
Var st2=“hhh”
Document.write(10+20+”hhh”);   30hhh
```

condt

- Var x=4+4;
- Var y=“5”+5;
- Var z=“she”+”she1”
- System.out.println(x); //8
- System.out.println(y); //55
- System.out.println(z); //sheshe1

Functions

- A JavaScript function is a block of code designed to perform a particular task.
- We can reuse code again (reusability)
- Function Functionname()
- {
- }

Function m1() {…….}

m1();

condt

- Function without parameters

```
<html>
<script>
function m1()
{
var a=3,b=6;
document.write(a+b);
}
m1();
</script>
<body>
</body>
</html>
```

condt

- Function with parameters

```
<html>
<script>
function m1(a,b)
{
document.write(a+b);
}
m1("sheela","J");
</script>
<body>
</body>
</html>
```

condt

- Function with return

```
<html>
<script>
function m1(a,b)
{
var c=a+b;
return c;
}
var x=m1(24,23);
document.write(x);
</script>
<body>
</body>
</html>
```

Objects

- JS is object oriented programming
- Var car = {

  name:maruthi,

  Model:Mar1,

  price:20000, methodname: function() { ….}

}

- Document.write(car.name)
- Document.write(car.Model)
- Document.write(car.price)

Events

- JavaScript- create dynamic web page
- Event means action
- Combination with function

Mouse Event

Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

condt

Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

Onclick()

- When user click on element
- In HTML:
- <element onclick=“some functionname”>

Ondoubleclick-> ondblclick()

- When user double click on element
- <element ondblclick=“some functionname”>

condt

```
<html>
<script>
function m2()
{
document.getElementById("ii").innerHTML=Date();
}
</script>
<body>
<p id="ii"></p>
<button onclick="m2()"> click on buttton </button>
</body>
</html>
```

condt

- Onload-it occurs when an object has been loaded
- -body
- <button onload=“ “ >

```
<html>
<script>
function m1()
{
alert("hhhh");
}
</script>
<body onload="m1()">
hi how hhj
</body>
</html>
```

condt

```
<html>
<script>
function m1()
{
alert("hhhh");
}
</script>
<body>
<img src="baby.jpg" height="300" width="300" onload="m1()">
</body>
</html>
```

condt

- Onerror Event:It is triggered ,if an error while loading external file (image or documents)
- <element onerror=“sourcecode”>

OnMouseover and onmouseout Event:

It occurs when a user moves the mouse pointer over an the element.

<element onmouseover=“sourcecode”>

condt

```
function m1(x)
{
x.style.height="300";
x.style.width="300";
}
function m2(x)
{
x.style.height="30";
x.style.width="30";
}
</script>
<body>
<img src="baby.jpg"  onmouseover="m1(this)" onmouseout="m2(this)"> hai eveny hhhhh</p
</body>
</html
```

condt

- Onselect Event-it occurs, when an user to select content of element

```
<html>
<script>
function m1()
{
alert("should not possible to select");
}
</script>
<body>
Name <input type="text" onselect="m1()">
password <input type="password" onselect="m1()">
</body>
</html>
```

condt

- Onblur and Onfocus Event
- Onblur event: It occurs when an object loses focus.
- Used in form (user leaves a element in form)

```
function m1()
{
var z=document.getElementById("gg");
z.value=z.value.toUpperCase();
}
function m2()
{
var z=document.getElementById("g1");
z.value=z.value.toLowerCase();
}
</script>
<body>
Name <input type="text" id="gg" onblur="m1()">
lastname <input type="text" id="g1" onblur="m2()">
</body>
</html>
```

condt

```
<html>
<script>
function m1(x)
{
x.style.backgroundColor="blue";
x.style.color="yellow";
}
function m2(x)
{
x.style.backgroundColor="yellow";
x.style.color="blue";
}
</script>
<body>
Name <input type="text" onblur="m1(this)">
lastname <input type="text"  onfocus="m2(this)">
</body>
</html>
```

condt

- Onsubmit Event
- In form
- <form onsubmit=“ sourcecode”>

condt

```
<html>
<script>
function check()
{
alert("submitted");
}
</script>
<body>
<form  onsubmit="check()">
name <input type="text" >
<input type="submit" value="submit" >
</form>
</body>
</html>
```

Looping statement (control statement)

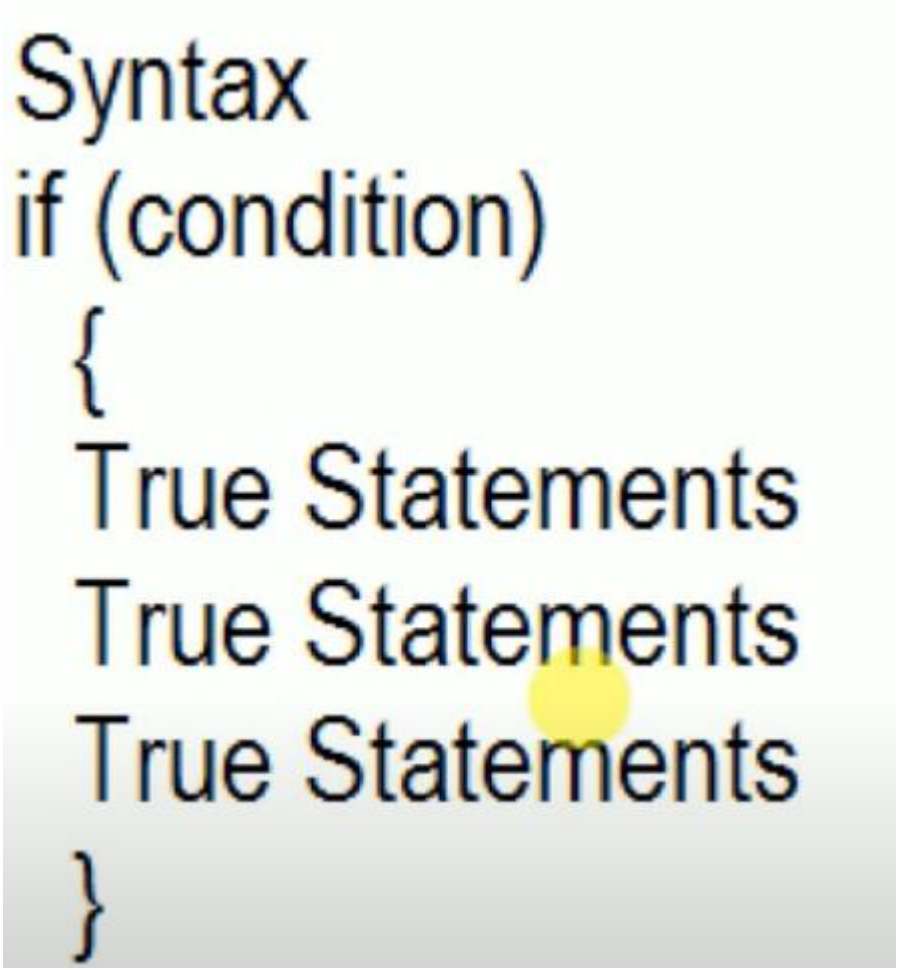
- Document.write(“ffff”);
- Prompt(“ “)
- If(Boolean condition)

  {

  ……

  …..

  }



condt

- If …else

```
Syntax
if (condition)
  {
  True Block Statements
  True Block Statements
  }
else
  {
   False Block Statements
   False Block Statements
   }
```

condt

Syntax

```
if (condition1)
{
code to be executed if condition1 is true
}
else if (condition2)
{
code to be executed if condition2 is true
}
else
{
code to be executed if neither
condition1 nor condition2 is true
}
```

condt

- Switch:

```
Syntax
switch(n)
{
case 1:
  execute code block 1
  break;
case 2:
  execute code block 2
  break;
default:
  code to be executed if
n is different from case 1 and 2
}
```

condt

- Looping statement:
- **while** - loops through a block of code while a condition is true
- **do...while** - loops through a block of code once, and then repeats the loop while a condition is true
- **for** - run statements a specified number of times

condt

- **while**
- The while statement will execute a block of code while a condition is true..

```
while (condition)
{
    code to be executed
}
```

do...while

The do...while statement will execute a block of code once, and then it will repeat the loop while a condition is true

```
do
{
    code to be executed
}
while (condition)
```

condt

- **for**
- The for statement will execute a block of code a specified number of times

```
for (initialization; condition; increment)
{
    code to be executed
}
```

Condt

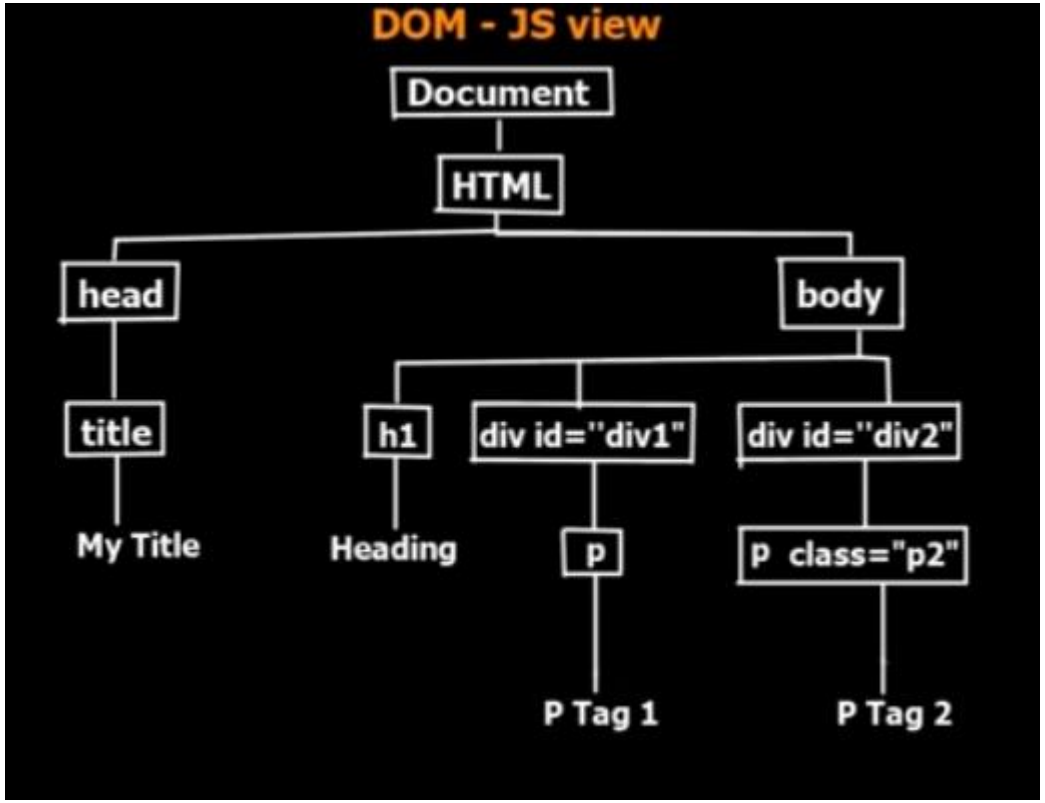
```
<html>
<script>
for(i=1;i<6;i++)
{
document.write("<h" +i+ ">This");
document.write("</h" +i+ ">");
}
</script>
</html>
```

DOM(Document Object Model)

- How does java script understand the HTML

HTML Document

```
<html>
  <head>
    <title>My Title </title>
  </head>
  <body>
    <h1>Heading</h1>
    <div id="div1">
      <p>P Tag 1</p>
    </div>
    <div id="div2">
      <p class="p2">P Tag 2</p>
    </div>
  </body>
</html>
```



FORM Validation

```
<html>
<script>
function m1()
{
var x=document.getElementById("t1");
var y=document.getElementById("t2");
if(x.value ==""||y.value =="")
{
alert("text and pwd tab is empty");
return false;
}
else
{
return true;
}
}
</script>
<form onsubmit="return m1()" action="whi.html">
<body>
Name <input type="text" id="t1"/><br><br>
password<input type="password" id="t2"/><br><br>
submit<button type="submit" id="t3" /><br><br>
</form>
```

Module 3 PHP

- Server side programming **Practical extraction and Report Language (Perl) ,Active Server Pages (ASP) .Java server pages (JSP), Ruby**
- Embedded HTML with PHP
- Interact with database
- Secure
- Open source
- Encrypt data
- Dynamic web page
- Software :Xampp and sublime

Php code format:

```
<?
        echo “Welcome to PHP”
?>
```

- echo ‘hai’;
- print “hai”;
- Comment :
- //
- #
- /*…….*/

Variable Names

- **Start with a dollar sign ($) followed by a letter or underscore, followed by any number of letters, numbers, or underscores**
- **Case matters**

Valid	In-valid
$abc = 12;	abc = 12;
$total = 0;	$2php = 0;
$largest_so_far = 0;	$bad-punc = 0;

Strings

- “ “ or ‘ ‘
- **Concatenation is the "." not "+“**

  String literals can use single quotes or double quotes.

  The backslash (\) is used as an “escape” character.
- ** **
- **New Line
  echo “…..
”**

Ternary Operator

- **The ternary operator comes from C. It allows conditional expressions. It is like a one-line if-then-else. Like all “contraction” syntaxes, we must use it carefully.**

```
$www = 123;
$msg = $www > 100 ? "Large" : "Small" ;
echo "First: $msg \n";
$msg = ( $www % 2 == 0 ) ? "Even" : "Odd";
echo "Second: $msg \n";
$msg = ( $www % 2 ) ? "Odd" : "Even";
echo "Third: $msg \n";
```

First: Large

Second: Odd

Third: Odd

Operators

Increment / Decrement (++ --)

String concatenation (.)

Equality (== !=)

Ternary (? :)

Side-effect Assignment (+= -= .= etc.)

Ignore the rarely-used bitwise operators (>> << ^ | &)

Conditional statement

- Simple if
- If ..else
- If ..elseif…else ladder
- Switch statement

condt

- Simple if :

```
if (condition) {
  code to be executed if condition is true;
}
```

```php
<?php
$t = date("H");

if ($t < "20") {
  echo "Have a good day!";
}
?>
```

condt

- If ..else

```php
<?php
$t = date("H");

if ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>
```

condt

```php
<?php
$t = date("H");

if ($t < "10") {
  echo "Have a good morning!";
} elseif ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>
```

condt

```php
<?php
$favcolor = "red";

switch ($favcolor) {
  case "red":
    echo "Your favorite color is red!";
    break;
  case "blue":
    echo "Your favorite color is blue!";
    break;
  case "green":
    echo "Your favorite color is green!";
    break;
  default:
    echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

PHP Loops

•while - loops through a block of code as long as the specified condition is true
•do...while - loops through a block of code once, and then repeats the loop as long as the specified condition is true
•for - loops through a block of code a specified number of times
•foreach - loops through a block of code for each element in an array

- $x = 1; - Initialize the loop counter ($x), and set the start value to 1
- $x <= 5 - Continue the loop as long as $x is less than or equal to 5
- $x++; - Increase the loop counter value by 1 for each iteration

condt

```php
<?php
$x = 0;

while($x <= 100) {
  echo "The number is: $x <br>";
  $x+=10;
}
?>
```

condt

```php
<?php
$x = 1;

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

condt

```php
<?php
for ($x = 0; $x <= 10; $x++) {
  echo "The number is: $x <br>";
}
?>
```

```php
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
  echo "$value <br>";
}
?>
```

condt

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

<?php
for ($x = 0; $x < 10; $x++) {
  if ($x == 4) {
    break;
  }
  echo "The number is: $x <br>";
}
?>

</body>
</html>
```

The number is: 0

The number is: 1

The number is: 2

The number is: 3

condt

```
<!DOCTYPE html>
<html>
<body>
<?php
for ($x = 0; $x < 10; $x++) {
  if ($x == 4) {
    continue;
  }
  echo "The number is: $x <br>";
}
?>

</body>
</html>
```

The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9

Array-

1. **Indexed arrays** - Arrays with a numeric index

array();

0 1 2

```
$a = array(10,20,30);
```

```php
<?php
echo"hai";
$ss = array(1,2,"fff");
echo $ss[2]."<br>";
echo $ss[1]."<br>";
echo $ss[0]."<br>";
?>
```

condt

```php
$ss = array(1,2,"fff");
/*echo $ss[2]."<br>";
echo $ss[1]."<br>";
echo $ss[0]."<br>"; */
print_r($ss);
?>
```

Print_r- to print or display information stored in a variable.

condt

```php
$ss = array(1,2,"fff");
/*echo $ss[2]."<br>";
echo $ss[1]."<br>";
echo $ss[0]."<br>"; */
echo "<pre>";
print_r($ss);
echo "</pre>";
echo "Zero ".$ss[0]."<br>"."1st ".$ss[1]."<br>"."2nd ".$ss[2]."<br>";

?>
```

condt

```php
$ss = array(1,2,"fff");
echo "<ul>";
for($i = 0; $i <= 2; $i++)
{
  echo "<li>". $ss[$i] ."</li>";
}
echo "</ul>";
?>
```

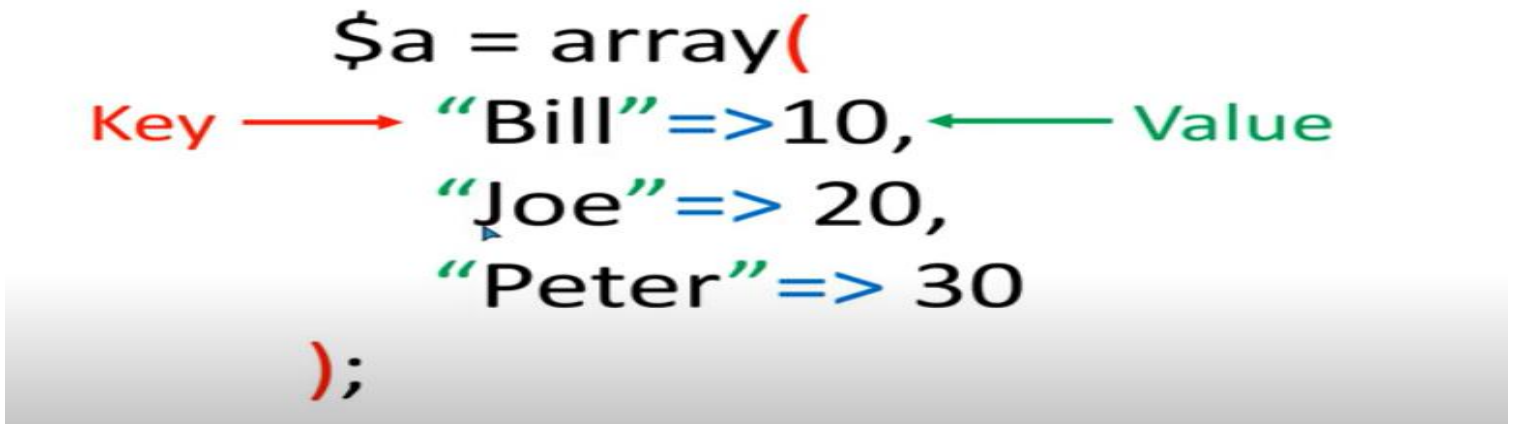
Length of Array :Count()

condt

2.Associative arrays - Arrays with named keys

Array with Key & Value

$ss=array(“ggg”=>20,”fff”=>30, “hhh”=>40);



condt

```php
<?php

$age = array(
        "bill" => 25,
        "steve" => 28,
        "elon" => 22
      );

echo $age["bill"] . "<br>";
echo $age["steve"] . "<br>";
echo $age["elon"] . "<br>";
?>
```

```php
<?php

$age = [
        "bill" => 25,
        "steve" => 28,
        "elon" => 22
      ];

print_r($age);

echo $age["bill"] . "<br>";
echo $age["steve"] . "<br>";
echo $age["elon"] . "<br>";
?>
```

condt

```php
<?php

$age = [
        "bill" => "25",
        "steve" => 28,
        "elon" => 22
      ];

$age["elon"] = 50;

echo "<pre>";
var_dump($age);
echo "</pre>";
```

```
array(3) {
  ["bill"]=>
  string(2) "25"
  ["steve"]=>
  int(28)
  ["elon"]=>
  int(50)
}
```

condt

```php
$s1=["gg"=>"10", "jj"=>20, "cc"=>34];
$d=array_keys($s1);
for($j=0; $j<count($d); $j++)
{
  echo $d[$j];
}
?>
```

```php
$dd=array("k1"=>10,"k2"=>100,"k3"=>110);
$ff=array_keys($dd);
$gg=array_values($dd);
for($i=0; $i<count($ff); $i++)
{
  echo $gg[$i];
}
```

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

foreach($age as $x => $x_value) {
  echo "Key=" . $x . ", Value=" . $x_value;
  echo "<br>";
}
?>
```

condt

- **3.Multidimensional arrays** - Arrays containing one or more arrays

Emp No.	Name	Designation	Salary
1	Krishana	Manager	50000
2	Salman	Salesman	20000
3	Mohan	Computer Operator	12000
4	Amir	Driver	5000

```
$emp = array (
    array(1,"Krishana","Manager",50000),
    array(2,"Salman","Salesman",20000),
    array(3,"Mohan","Computer Operator",12000),
    array(4,"Amir","Driver",5000)
);
```

condt

```
$emp = [
0   [ 1 ,"Krishana“ , "Manager“ , 50000 ],
1   [ 2 , "Salman“ , "Salesman“ , 20000 ],
2   [ 3 , "Mohan“ , "Computer Operator“ , 12000 ],
3   [ 4 , "Amir“ , "Driver“ , 5000 ]
];
```

condt

```
$emp = [
      0          1              2              3
0   [ 1 ,"Krishana“ , "Manager“ , 50000 ],
1   [ 2 , "Salman“ , "Salesman“ , 20000 ],
2   [ 3 , "Mohan“ , "Computer Operator“ , 12000 ],
3   [ 4 , "Amir“ , "Driver“ , 5000 ]
];
```

```
echo $emp[1][1];
```

condt

```
for($row = 0; $row < 4; $row++){
  for($col = 0; $col < 4; $col++){
    echo $emp[$row][$col] . " ";
  }
```

Function:

```
function functionName(){ ← Function Definition
   Statement
}

functionName(); ← Calling a Function
```

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

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

condt

```
function functionName(parameter1, parameter2){
    Statement
}
```

```
functionName(argument1, argument2);
```

Parameter Passing

```php
<?php

function hello($name){
  echo "Hello $name.<br>";
}

hello("Yahoo Baba");


?>
```

Hello Yahoo Baba.

condt

```php
<?php

function hello($fname,$lname){
  echo "Hello $fname $lname.<br>";
}

hello("Yahoo","Baba");



?>
```

Hello Yahoo Baba.

Condt (default values)

```php
function m1($n1,$n2)
{
  echo $n1,$n2;
}
 m1("sheela","Dr");
```

```php
function m1($n1="hai",$n2="ff")
{
  echo $n1,$n2."<br>";
}
 m1("sheela","Dr");
 m1();
```

sheelaDr
haiff

Function with return

```
function functionName(parameter1, parameter2){
    Statements
    return value;
}

$a = functionName(argument1, argument2);
```

condt

```php
<?php

function hello($fname="First",$lname="Last")
  $v = "$fname $lname";

  return $v;
}

echo hello("Yahoo","Baba");

?>
```

condt

```php
<?php

function hello($fname="First",$lname="Last")
  $v = "$fname $lname";

  return $v;
}

$name = hello("Yahoo","Baba");

echo "Hello $name";
?>
```

condt

```php
<?php

function sum($math, $eng, $sc){
  $s = $math + $eng + $sc;

  return $s;
}

$total= sum(55,65,88);

echo $total;
?>
```

Regular Expression

- Used to extract information from data
- $Pattern=“/ /”

For Ex: $Pattern=“/data/”

(or)

=“#data#”

- Preg_match_all ($pattern, $variable (where I have to search));

condt

```php
<?php

$pattern="/f41/";
$str="galaxy f41";
echo preg_match_all($pattern,$str);

?>
```

```
1
```

condt

```php
$str="sheela baby sheela sheela";
 $patt="#sheela#";
 echo preg_match_all($patt,$str);

 $pat="#Sheela#i";
 echo preg_match_all($pat,$str);
```

condt

Function	Description
preg_match()	Returns 1 if the pattern was found in the string and 0 if not
preg_match_all()	Returns the number of times the pattern was found in the string, which may also be 0
preg_replace()	Returns a new string where matched patterns have been replaced with another string

condt

Using preg_match(): The `preg_match()` function will tell you whether a string contains matches of a pattern

```php
<?php
$str = "Visit W3Schools";
$pattern = "/w3schools/i";
echo preg_match($pattern, $str); // Outputs 1
?>
```

Condt

Using preg_match_all()

The `preg_match_all()` function will tell you how many matches were found for a pattern in a string.

```php
<?php
$str = "The rain in SPAIN falls mainly on the plains.";
$pattern = "/ain/i";
echo preg_match_all($pattern, $str); // Outputs 4
?>
```

condt

Using preg_replace()

The `preg_replace()` function will replace all of the matches of the pattern in a string with another string.

```php
<?php
$str = "Visit Microsoft!";
$pattern = "/microsoft/i";
echo preg_replace($pattern, "W3Schools", $str); // Outputs "Visit W3Schools!"
?>
```

condt

- Pattern design
- []→character class- Brackets are used to find a range of characters:
- $pat="/f41/";
- $pat="/[f41]/";
- $pat="/[a-z]/";
- $pat="/[a-z A-Z]/";

condt

Expression	Description
[abc]	Find one character from the options between the brackets
[^abc]	Find any character NOT between the brackets
[0-9]	Find one character from the range 0 to 9

condt

- [ ] ? ….0 or 1 time
- []*……0 or more time
- []+…….1 or more time
- []{n}……n number of times
- []{y, z}…… occurs at least y times, but not less than z times

Condt

- Subpattern-()
- $pat="/(abc|xy)/";

```php
$str="abcc ab xyz";
 $patt="#sheela#";
 //echo preg_match_all($patt,$str);

 $pat="#[a-zA-Z]#";
 //echo preg_replace($pat,"poshi",$str);
// echo preg_match_all($pat,$str);

  $pat1= "#(ab|xy)#";
  echo preg_match_all($pat1,$str);
```

condt

Metacharacter	Description
|	Find a match for any one of the patterns separated by | as in: cat|dog|fish
^	Finds a match as the beginning of a string as in: ^Hello
$	Finds a match at the end of the string as in: World$
\d	Find a digit
\s	Find a whitespace character
\b	Find a match at the beginning of a word like this: \bWORD, or at the end of a word like this: WORD\b
\uxxxx	Find the Unicode character specified by the hexadecimal number xxxx

condt

```php
$str="Ab";
 $patt="#^Ab#";
 echo preg_match_all($patt,$str,$ar);
 echo "<br>";
 echo "<pre>";
 print_r($ar);
 echo "</pre>";
```

```
Array
(
    [0] => Array
        (
            [0] => Ab
        )

)
```

cond

```php
$str="Abf";
 $patt="#\d#";
 echo preg_match_all($patt,$str,$ar);
 echo "<br>";
 echo "<pre>";
 print_r($ar);
 echo "</pre>";
```

```
Array
(
    [0] => Array
        (
        )

)
```

```php
$str="Abf 33";
 $patt="#\d#";
 echo preg_match_all($patt,$str,$ar);
 echo "<br>";
 echo "<pre>";
 print_r($ar);
 echo "</pre>";
```

```
Array
(
    [0] => Array
        (
            [0] => 3
            [1] => 3
        )

)
```

condt

```php
$str="Abf 33";
 $patt="#\s#";
 echo preg_match_all($patt,$str,$ar);
 echo "<br>";
 echo "<pre>";
 print_r($ar);
 echo "</pre>";
```

```
Array
(
    [0] => Array
        (
            [0] =>
        )

)
```

Quantifiers

```php
$str="bb 33";
 $patt="#[b]{3}#";
 echo preg_match_all($patt,$str,$ar);
 echo "<br>";
 echo "<pre>";
 print_r($ar);
 echo "</pre>";
```

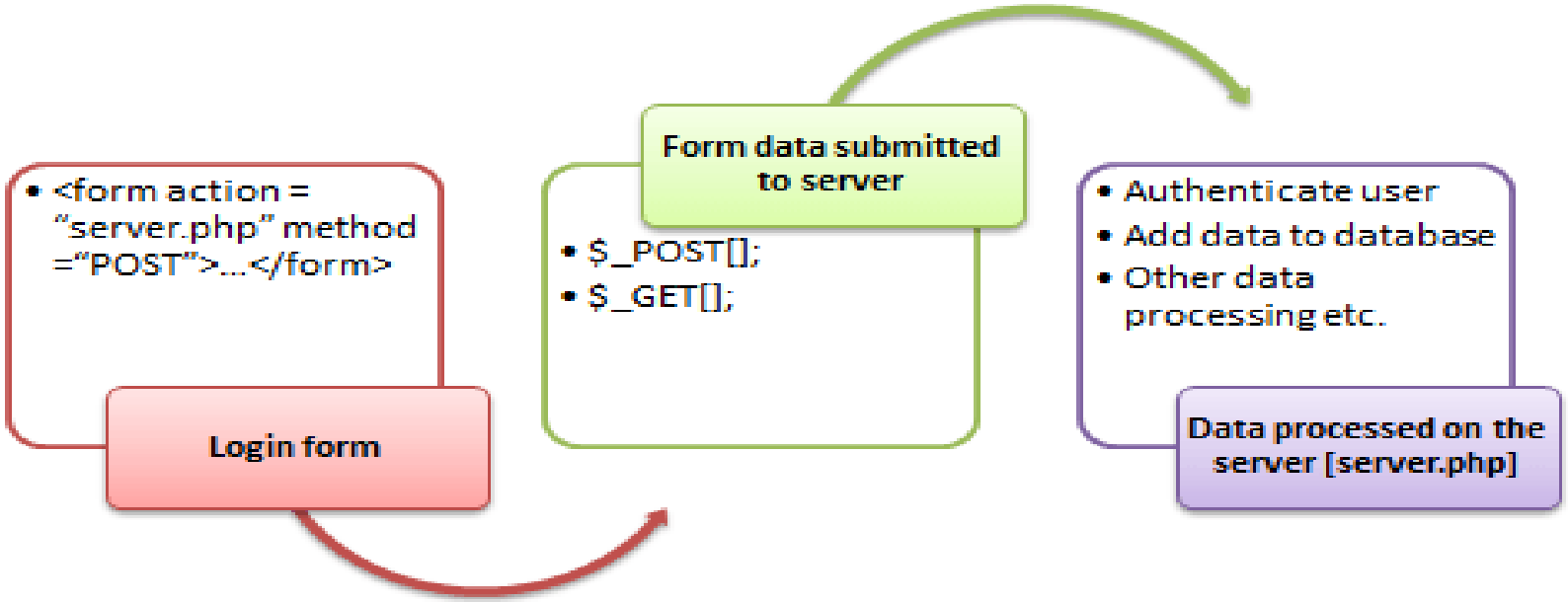
```
Array
(
    [0] => Array
        (
        )

)
```

Form Get() & POST() method

- **Forms are used to get input from the user and submit it to the web server for processing.**
- There are two ways the browser client can send information to the web server.
- The GET Method
- The POST Method

condt


• <form action = "server.php" method ="POST">...</form>
Login form
Form data submitted to server
• $_POST[];
• $_GET[];
• Authenticate user
• Add data to database
• Other data processing etc.
Data processed on the server [server.php]

condt

- **PHP provides many predefined variables or built-in variables to all scripts**
  - **$GLOBALS: References all variables available in global scope.**
  - **$_SERVER: Server and execution environment information.**
  - **$_GET: HTTP GET variables.**
  - **$_POST: HTTP POST variables.**
  - **$_FILES: HTTP File Upload variables.**
  - **$_REQUEST: HTTP Request variables.**
  - **$_SESSION: Session variables.**
  - **$_ENV: Environment variables.**
  - **$_COOKIE: HTTP Cookies.**

condt

```php
<?php
$a=4;
$b=5;
function m1()
{
    $GLOBALS['c']=$GLOBALS['a']+$GLOBALS['b'];
    echo $GLOBALS['c'];
}

m1();
echo $GLOBALS['c'];
//print_r($GLOBALS);
```

$_SERVER

is an array containing information such as headers, paths, and script locations

```
echo"<pre>";
print_r($_SERVER);
echo"</pre>";
```

$_GET,$_POST

- The PHP provides **$_GET** associative array to access all the sent information using GET method.
- The PHP provides **$_POST** associative array to access all the sent information using POST method.
- The PHP $_REQUEST variable contains the contents of both $_GET, $_POST,

GET VS POST

- GET utilizes URL to send the data to the server using name/value pairs
- GET has a limit of 2048 characters
- GET transports data with the URL, so its visible and not advisable for sensitive information
- Get is really useful when users wants to bookmark the result to reference the specific page later.
- POST is secure and no data displayed in URL
- POST form submissions cannot be bookmarked

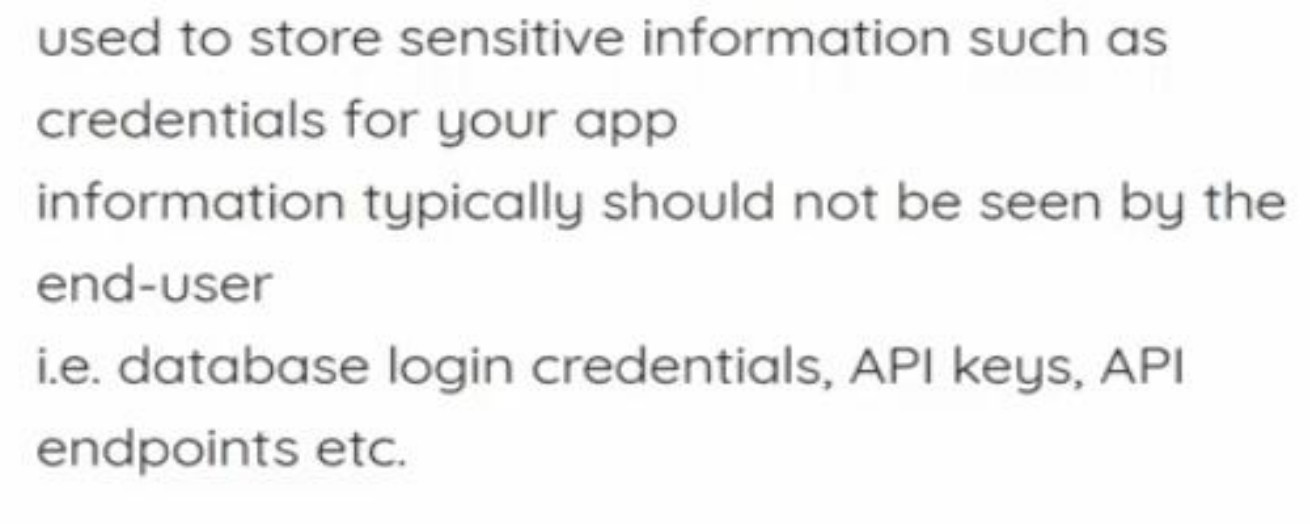
Condt

- $_FILES — HTTP File Upload variables
- Associative array containing items uploaded via HTTP POST method.
- **$_FILES['file']['name']** - The original name of the file to be uploaded.
- **$_FILES['file']['type']** - The mime type of the file.
- **$_FILES['file']['size']** - The size, in bytes, of the uploaded file.

condt

- **$_ENV: Environment variables.**

$_ENV is another super global associative array in PHP. It stores environment variables available to current script.



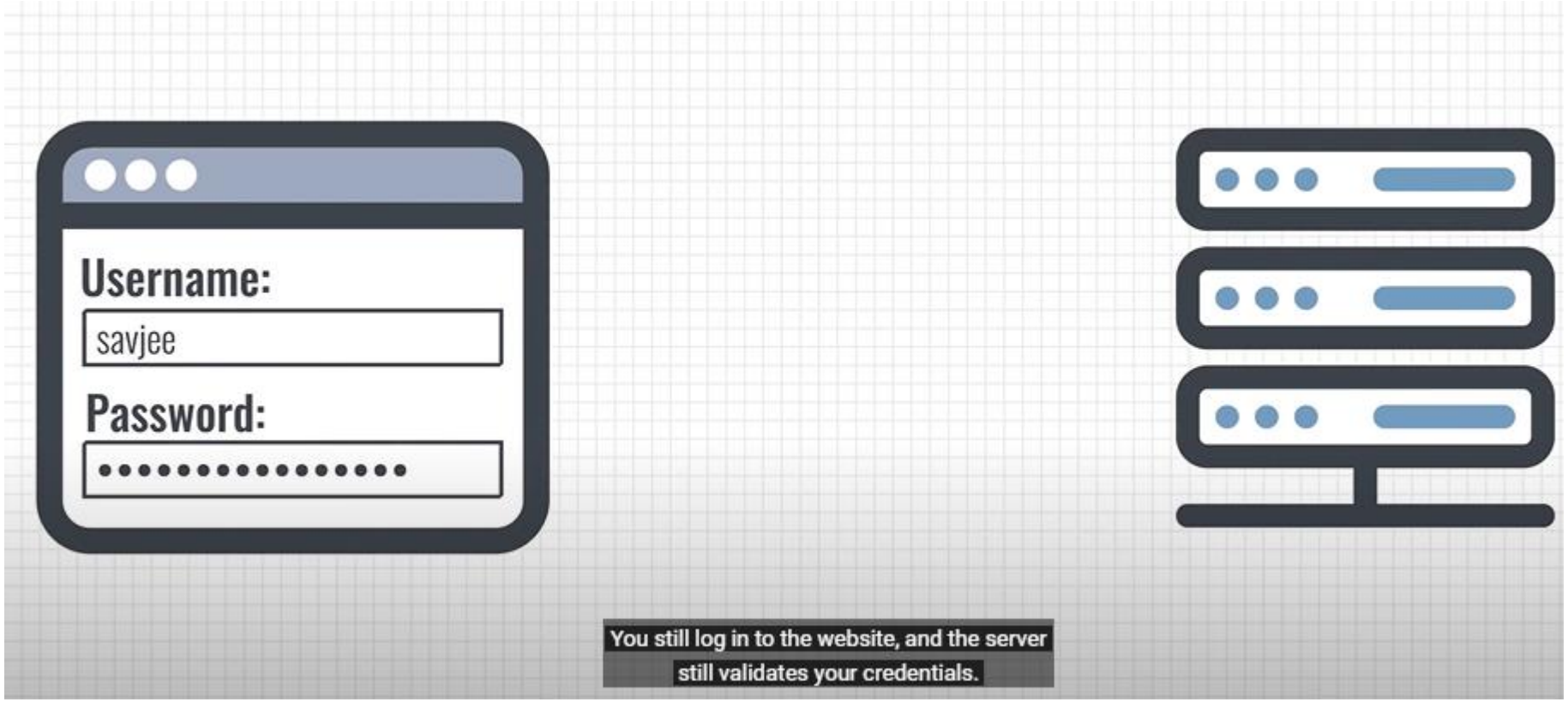
condt

- Provides information about execution environment
  -Types of web browser
- -Types of server
- -Details of HTTP connection

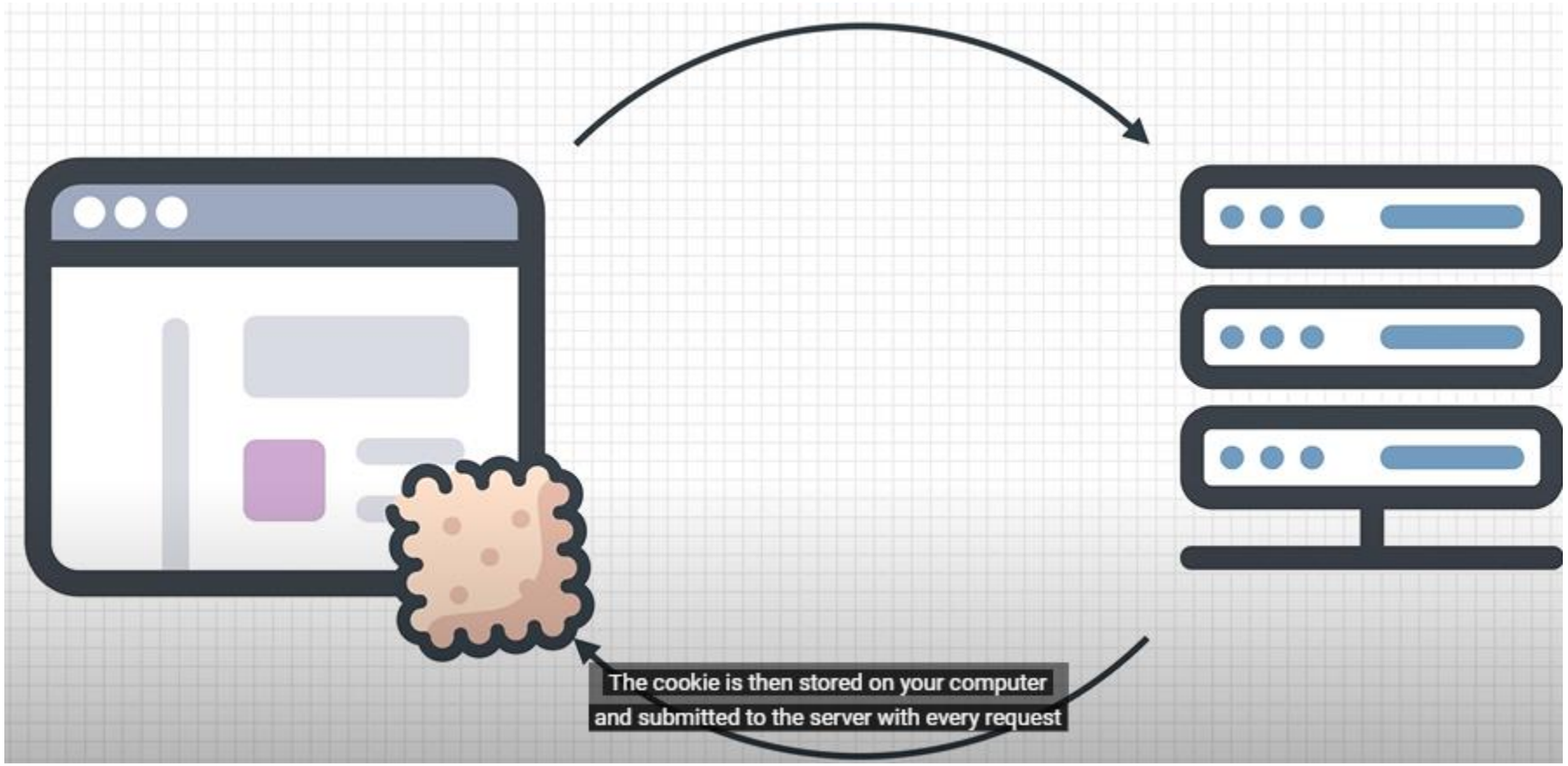
$_COOKIE

- For accessing a cookie value, the **PHP $_COOKIE super global** variable is used.
- It is an associative array that contains a record of all the cookies values sent by the browser in the current request.
- A cookie is a small text file that lets you store a small amount of data (nearly 4KB) on the user's computer.
- They are typically used to keeping track of information such as username that the site can retrieve to personalize the page when user visit the website next time.

condt


Username:
savjee
Password:
You still log in to the website, and the server
still validates your credentials.

condt


The cookie is then stored on your computer
and submitted to the server with every request

$_Session variable

- An associative array containing session variables available to the current script.
- associative array containing session **variables** available to the current script.

Difference between Session Vs Cookie

- Cookies are client-side files that contain user information, whereas Sessions are server-side files that contain user information.
- Cookie is not dependent on session, but Session is dependent on Cookie.
- Cookie expires depending on the lifetime you set for it, while a Session ends when a user closes his/her browser.
- The maximum cookie size is 4KB whereas in session, you can store as much data as you like.
- Cookie does not have a function named unsetcookie() while in Session you can use Session_destroy(); which is used to destroy all registered data or to unset some

Validating form

Module 4 :Cookie in PHP

- Cookie are text files stored in client browser.
- A cookie is a small file that the server embeds on the user's computer.
- Each time the same computer requests a page with a browser, it will send the cookie too.

Create Cookies With PHP

A cookie is created with the `setcookie()` function

`setcookie(name, value, expire, path, domain, secure, httponly);`
Only the *name* parameter is required. All other parameters are optional.

setCookie() – setCookie() is used to set/create/sent cookies. This function must appear before the <html> tag.
Syntax: - setCookie(name, value, expire, path, domain, secure, httponly);
Ex: - setCookie("username", "geeky");
Where,
Name – This is the name of the cookie.
Value – This sets the value of cookie. This value is stored on the clients computer.
name and value is required to set cookie.

condt

Optional Cookies Attribute:-

expire
Path
domain
Secure
httponly

Whenever you omit the optional cookie fields, the browser fills them in automatically with reasonable defaults.

condt

expire

It describes the time when cookie will be expire. If this parameter is not set or set 0 then cookie will automatically expire when the Web Browser is closed.

Type of cookies: -

- Session Cookies – Cookies that are set without the expire field are called session cookies. It is destroyed when the user quits the browser.
- Persistent Cookies – The browser keeps it up until their expiration date is reached.

```
Ex:- setCookie( "username", "geeky", time()+60*60*24*10);        // 10 days
```

condt

Deleting Cookies

A cookie is deleted by setting a cookie with the same name (and domain and path, if they were set) with an expiration date in the past.

Session in PHP

- **session_start()** function is used to begin a new session.It als creates a new session ID for the user
- <?php

- session_start();

- ?>

Destroying Certain Session Data:

```
unset($_SESSION["Rollnumber"]);
```

Destroying Complete Session:

The **session_destroy()** function is used to completely destroy a session. The session_destroy() function does not require any argument.

condt

- <?php
- session_start();
- session_destroy();
- ?>

callback function

- A callback function (often referred to as just "callback") is a function which is passed as an argument into another function.
- To use a function as a callback function, pass a string containing the name of the function as the argument of another function

condt

```php
<?php
function m1()
{
echo "hai";
}
function m2($v)
{
    $v(); // call m1() function
}
m2("m1");
```

condt

Anonymous Functions

Anonymous functions, also known as closures or Lambda, allow the creation of functions which have no specified name.

- Can be stored in a Variable
- Can be Returned in a Function
- Can be pass in a Function

Syntax: -

```
function ( ) {
block of statement;
};
```

Condt(can be stored in Variable)

```php
// Anonymous Function
$a = function(){
        echo "Anonymous Function";
    };
$a();
```

Global variable not possible to access

```php
<?php
    // Anonymous Function
    $y = 34; // global variable
    $a = function(){
            echo "Anonymous Function $y";
        };
    $a();
?>
```

```php
<?php
    // Anonymous Function
    $y = 34; // global variable
    $a = function($p){
            echo "Anonymous Function $p";
        };
    $a($y);
?>
```

“use” Keyword

```php
$a=34;
$d=function() use ($a)
{
    echo "hai".$a;
};
m2("m1");
$d();
```

Filter

- This PHP filters is used to validate and filter data coming from insecure sources, like user input.

Function	Description
filter_has_var()	Checks whether a variable of a specified input type exist
filter_id()	Returns the filter ID of a specified filter name
filter_input()	Gets an external variable (e.g. from form input) and optionally filters it
filter_input_array()	Gets external variables (e.g. from form input) and optionally filters them
filter_list()	Returns a list of all supported filter names
filter_var()	Filters a variable with a specified filter
filter_var_array()	Gets multiple variables and filter them

Filter_var() For EMAIL 274

```php
$str1="sheela.nitt@gmail.com";
echo filter_var($str1,FILTER_VALIDATE_EMAIL);
```

FILTER_VALIDATE_URL- 273
FILTER_VALIDATE_IP- 275

condt

- **Validate filter constants**
  - **FILTER_VALIDATE_BOOLEAN -** Validates a boolean
  - **FILTER_VALIDATE_INT -** Validates an integer
  - **FILTER_VALIDATE_FLOAT -** Validates a float
  - **FILTER_VALIDATE_REGEXP -** Validates a regular expression
  - **FILTER_VALIDATE_IP -** Validates an IP address
  - **FILTER_VALIDATE_EMAIL -** Validates an e-mail address
  - **FILTER_VALIDATE_URL -** Validates an URL

condt

- Validating data = Determine if the data is in proper form.
- Sanitizing data = Remove any illegal character from the data.

condt

- Filter_has_var():

- Check if the input variable "email" is sent to the PHP page, through the "get" method
- The filter_has_var() function checks whether a variable of a specified input type exist.

- `filter_has_var(`*`type, variable`*`)`

condt

- INPUT_GET
- INPUT_POST
- INPUT_COOKIE
- …..

filter_id()-return filter ID

- <!DOCTYPE html>
- <html>
- <body>

274

- <?php
- echo(filter_id("validate_email"));
- ?>

- </body>
- </html>

Filter_input()

- filter_input() function gets an external variable (e.g. from form input) and optionally filters it.

- `filter_input(`*`type, variable, filter`*`)`

- `filter_input(INPUT_GET, "email", FILTER_VALIDATE_EMAIL)`

filter_input_array() Function

- Gets external variables and optionally filters them.

condt

$d=array(“name”,”age”,”email”);

```
<?php
$filters = array
 (
 "name" => array
  (
  "filter"=>FILTER_CALLBACK,
  "flags"=>FILTER_FORCE_ARRAY,
  "options"=>"ucwords"
  ),
 "age" => array
  (
  "filter"=>FILTER_VALIDATE_INT,
  "options"=>array
   (
   "min_range"=>1,
   "max_range"=>120
   )
  ),
 "email"=> FILTER_VALIDATE_EMAIL,
 );
print_r(filter_input_array(INPUT_POST, $filters));
?>
```

PHP filter_list() Function

- The filter_list() function returns a list of all the supported filter names.

- **Sanitize filter constants**
  - **FILTER_SANITIZE_EMAIL - Removes all illegal characters from an e-mail address**
  - **FILTER_SANITIZE_ENCODED - Removes/Encodes special characters**
  - **FILTER_SANITIZE_MAGIC_QUOTES - Apply addslashes() function**
  - **FILTER_SANITIZE_NUMBER_FLOAT - Remove all characters, except digits, +- and optionally ., eE**
  - **FILTER_SANITIZE_NUMBER_INT - Removes all characters except digits and + –**

- **Sanitize filter constants**
  - **FILTER_SANITIZE_SPECIAL_CHARS - Removes special characters**
  - **FILTER_SANITIZE_FULL_SPECIAL_CHARS - Encoding quotes can be disabled by using FILTER_FLAG_NO_ENCODE_QUOTES.**
  - **FILTER_SANITIZE_STRING - Removes tags/special characters from a string**
  - **FILTER_SANITIZE_STRIPPED - Alias of FILTER_SANITIZE_STRING**
  - **FILTER_SANITIZE_URL - Removes all illegal character from s URL**

condt

- The FILTER_SANITIZE_EMAIL filter removes all illegal characters from an email address.

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

<?php
// Variable to check
$email = "john(.doe)@exa//mple.com";

// Remove all illegal characters from email
$email = filter_var($email, FILTER_SANITIZE_EMAIL);

echo $email;
?>

</body>
</html>
```

john.doe@example.com

FILTER_SANITIZE_ENCODED filter removes or encodes special characters.

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

<?php
// Variable to check
$url="https://www.sheelaÅÅ.com";

// Encode characters
$url = filter_var($url, FILTER_SANITIZE_ENCODED);
echo $url;
?>

</body>
</html>
```

https%3A%2F%2Fwww.sheela%C3%85%C3%85.com

condt

- FILTER_FLAG_STRIP_LOW - Remove characters with ASCII value < 32
- FILTER_FLAG_STRIP_HIGH - Remove characters with ASCII value > 127
- FILTER_FLAG_ENCODE_LOW - Encode characters with ASCII value < 32
- FILTER_FLAG_ENCODE_HIGH - Encode characters with ASCII value > 127

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

<?php
// Variable to check
$url="https://www.sheelaÅÅ.com";

// Encode characters and remove all characters with ASCII value > 127
$url = filter_var($url, FILTER_SANITIZE_ENCODED, FILTER_FLAG_STRIP_HIGH);
echo $url;
?>

</body>
</html>
```

https%3A%2F%2Fwww.sheela.com

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

<?php
// Variable to check
$str = "<h1>sheelaÆØÅ!</h1>";

// Remove HTML tags and all characters with ASCII value > 127
$newstr = filter_var($str, FILTER_SANITIZE_STRING,
FILTER_FLAG_STRIP_HIGH);
echo $newstr;
?>

</body>
</html>
```

condt

- It will both remove all HTML tags, and all characters with ASCII value > 127, from the string.

```php
$str = "<h1>Hello WorldÆØÅ!</h1>";

$newstr = filter_var($str, FILTER_SANITIZE_STRING,FILTER_FLAG_STRIP_HIGH);
echo $newstr;
?>
```

FILTER_SANITIZE_MAGIC_QUOTES Filter

- This filter sets backslashes in front of predefined characters.

- The predefined characters are:
- single quote (')
- double quote (")
- backslash (\)
- NULL

-
```
<?php
$var="Peter's here!";

var_dump(filter_var($var,
FILTER_SANITIZE_MAGIC_QUOTES));
?>
```

- string(14) "Peter\'s here!"

condt

- **FILTER_FLAG_STRIP_LOW- Strips characters that have a numerical value <32.**
- **FILTER_FLAG_STRIP_HIGH- Strips characters that have a numerical value >127.**
- **FILTER_FLAG_EMAIL_UNICODE - Allows the local part of the email address to contain Unicode characters.**
- **FILTER_FLAG_PATH_REQUIRED - Requires the URL to contain a path part.**

- **Other filter constants**
  - **FILTER_UNSAFE_RAW - Do nothing, optionally strip/encode special characters**
  - **FILTER_CALLBACK - Call a user-defined function to filter data**

Error and Exception Handling

- Error- Wrong in the program
- Kind of Error
- Waring error- It does not stop the script running (ex reading content from external source)
- Notice error- same like warning (to access undefined variable)
- Parse error(syntax)- misused or missing symbol {},;…
- Fatal error-stop the execution

condt

- Run time- Exception
- it is an abnormal event that rises during the execution of a program and disturb the normal flow of exception
- Example -divide by zero

  $b=2/0;

  echo “hai”;
- Try
- catch
- throw

Condt.

- Try- exception should be in try block.

  -if the exception triggers , an exception is thrown
- Throw-This is how you trigger an exception. Each "throw" must have at least one "catch“
- Catch-A "catch" block retrieves an exception and creates an object containing the exception information
- $n=2, $s=10;
- $d=$s/$n;

condt

- When an exception is thrown, the code following it will not be executed, and PHP will try to find the matching "catch" block.
- If an exception is not caught, a fatal error will be issued with an "Uncaught Exception" message.

PHP File Handling: open, read, create and upload.

Default fun: touch(filename);
unlink(filename);
mkdir(dir name);

PHP has several functions for creating, reading, uploading, and editing files.

```
fopen("web.txt", "r")
```

- Used to open file
- 2 argument
- Filename , mode (read only, read&write, Write only)

condt

```php
$d=fopen("ff.txt","a");
fwrite($d,"hai hello welcome to VIT AP");
fclose($d);
?>
```

condt

- **“w” –** Opens a file for write only. If file not exist then new file is created and if file already exists then contents of file is erased.
- **“r” –** File is opened for read only.
- **“a” –** File is opened for write only. File pointer points to end of file. Existing data in file is preserved.
- **“w+” –** Opens file for read and write. If file not exist then new file is created and if file already exists then contents of file is erased.
- **“r+” –** File is opened for read/write.
- **“a+” –** File is opened for write/read. File pointer points to end of file. Existing data in file is preserved. If file is not there then new file is created.
- **“x” –** New file is created for write only.

condt

- fread() -reads from an open file.
- `fread($myfile,filesize("web.txt"));`
- fclose()- used to close an open file
-
```php
<?php
$myfile = fopen("web.txt", "r");
// some code to be executed....
fclose($myfile);
?>
```

```php
$d=fopen("ff.txt","r");
while(!feof($d))
{
    echo fgets($d);
}
?>
```

condt

- Read Single Line - fgets()
- function is used to read a single line from a file.
-
```
<?php
$myfile =
fopen("webdictionary.txt", "r") or die("Unable to
open file!");
echo fgets($myfile);
fclose($myfile);
?>
```

condt

- Read Single Character - fgetc()
- to read a single character from a file
- Write to File - fwrite()
- name of the file to write to and the second parameter is the string to be written

condt

```php
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable
to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

- File Size: Filesize()
- File exits: file_exists()

JSON – JavaScript object Notation

- It is a format to store and transfer (exchange data)
- Filename extension .json
- Media type: application/json
- Language independent
- Support for all browser

data is written inside the curly brace	{ json data }
Data is represented as key value pairs.	{ key : value}
Key should be enclosed in double quotes	{ "name": "ram"}
Key and Value should be separated by colon(:)	{ "name": "ram"}
Comma is used to separate data.	{ "name": "ram" , "age":25}
Square brackets hold arrays.	{"marks" : [95,85,74,65] }
Curly braces hold objects	

condt

1. Number
2. String
3. Boolean
4. Array
5. Object
6. null

```
{
  "name": "Ram",
  "gender": "male",
  "age":25,
  "ismarried":false,
  "marks":[95,78,25,88,85],
  "average":74.0,
  "hobbies":null,
  "address":{
              "street":"cherry road",
              "city":"Salem",
              "pincode":636007
            }
}
```

condt

```
{
  "student": [
    {
      "rollno": "ae1001",
      "name": "Ram",
      "gender": "male"
    },
    {
      "rollno": "ae1002",
      "name": "sara",
      "gender": "female"
    }
  ]
}
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
    <student>
        <rollno>ae1001</rollno>
        <name>ram</name>
        <gender>male</gender>
    </student>
    <student>
        <rollno>ae1002</rollno>
        <name>sara</name>
        <gender>female</gender>
    </student>
</root>
```

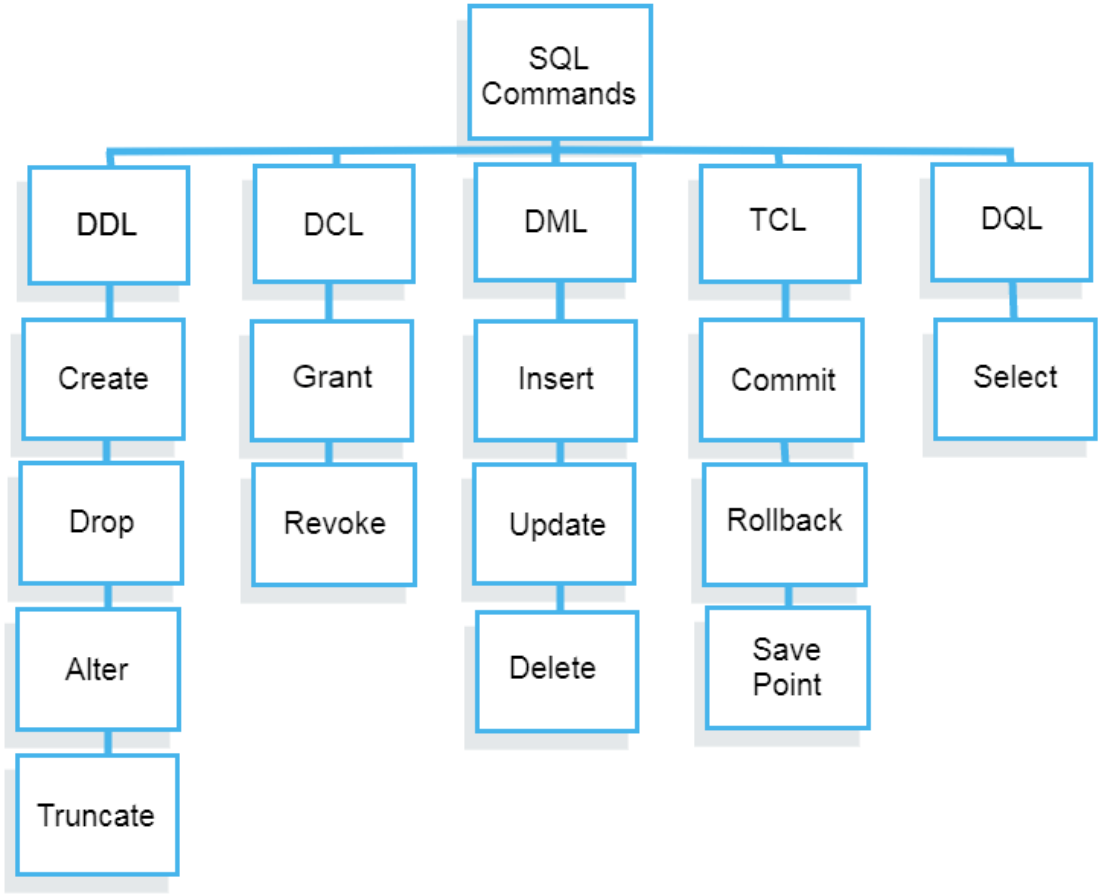
MODULE-5

MYSQL:

- Database: It is a collection of information that is organized
- Easy accessed, managed, update
- Open database
- Used for small and big business

- https://www.youtube.com/watch?v=QIOeXe3ZDSg
- mysql -u root –p/ pass_12345
- Use databse

Condt.


SQL Commands
DDL
DCL
DML
TCL
DQL
Create
Drop
Alter
Truncate
Grant
Revoke
Insert
Update
Delete
Commit
Rollback
Save Point
Select

Data Definition Language (DDL)

- Create:

```
mysql> create table students(id int,name varchar(15), phono int);
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> desc students;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int         | YES  |     | NULL    |       |
| name  | varchar(15) | YES  |     | NULL    |       |
| phono | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

condt

- Alter:
- Alters command allows you to alter the structure of the database.
- add a new column in the table

```
mysql> alter table students add(address varchar(25));
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc students;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | int         | YES  |     | NULL    |       |
| name    | varchar(15) | YES  |     | NULL    |       |
| phono   | int         | YES  |     | NULL    |       |
| address | varchar(25) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

condt

- To drop an column in the table:

```
mysql> alter table students drop column name;
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc students;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | int         | YES  |     | NULL    |       |
| phono   | int         | YES  |     | NULL    |       |
| address | varchar(25) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

condt

- Rename table and column:

```
mysql> alter table students rename to students_info;
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> alter table students_info rename column id to studentID;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc students_info;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| studentID | int         | YES  |     | NULL    |       |
| phono     | int         | YES  |     | NULL    |       |
| address   | varchar(25) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

condt

- **TRUNCATE:**
- This command used to delete all the rows from the table and free the space containing the table
- we can’t delete the single row as here WHERE clause is not used

condt

- To remove specific rows, use `DELETE`.
- To remove all rows from a large table and leave the table structure, use `TRUNCATE TABLE`. It's faster than `DELETE`.
- To remove an entire table, including its structure and data, use `DROP TABLE`

Data Manipulation Language

- Data Manipulation Language (DML) allows you to modify the
- database instance by inserting, modifying, and deleting its data.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.
- INSERT:
- This command is used to insert data into the row of a table.

```
mysql> insert into students(Rollno,Sname,pno)values(12,'sheela',908990890);
Query OK, 1 row affected (0.04 sec)
```

Condt.

- **UPDATE:** This command is used to update or modify the value of a column in the table.

```
mysql> insert into students(Rollno,Sname,pno)values(13,'Anu',90890000);
Query OK, 1 row affected (0.03 sec)

mysql> insert into students(Rollno,Sname,pno)values(14,'Priya',999990000);
Query OK, 1 row affected (0.02 sec)

mysql> select * from students;
+--------+--------+-----------+
| Rollno | Sname  | pno       |
+--------+--------+-----------+
|     12 | sheela | 908990890 |
|     13 | Anu    |  90890000 |
|     14 | Priya  | 999990000 |
+--------+--------+-----------+
```

condt

```
mysql> insert into students values(15,"varun",899997899);
Query OK, 1 row affected (0.03 sec)

mysql> select * from students;
+--------+--------+-----------+
| Rollno | Sname  | pno       |
+--------+--------+-----------+
|     12 | sheela | 908990890 |
|     13 | Anu    |  90890000 |
|     14 | Priya  | 999990000 |
|     15 | varun  | 899997899 |
+--------+--------+-----------+
```

```
mysql> update students set Sname="sundar" where Rollno=15;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

update

```
mysql> update students set Sname="sundar" where Rollno=15;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Without “Where”- it will update all Sname as “sundar” Rollno as “55”

Delete:

The DELETE statement is used to delete existing records in a table.

```
DELETE FROM table_name WHERE condition
```

```
mysql> delete from students where Rollno=55;
Query OK, 1 row affected (0.03 sec)

mysql> select * from students;
+--------+--------+
| Rollno | Sname  |
+--------+--------+
|     12 | sheela |
|     14 | Priya  |
|     15 | sundar |
+--------+--------+
3 rows in set (0.00 sec)
```

DCL

- Data Control Language.
- 1. GRANT
- 2. REVOKE

condt

```
mysql> create user ssb identified by '@123';
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> create user sheela1 identified by 'pass_12345';
Query OK, 0 rows affected (0.04 sec)

mysql> grant select, update,insert on students to sheela1;
Query OK, 0 rows affected (0.01 sec)
```

Cndt

```
mysql> quit
Bye

C:\mysql\bin>mysql -u sheela1 -p
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.23 MySQL Community Server - GPL
```

```
mysql> delete from student where id=1;
ERROR 1142 (42000): DELETE command denied to user 'ssb'@'localhost' for table 'student'
mysql> quit
Bye
```

MODULE :06

- Model View Controller
- It is architecture Pattern
- Kind of Design pattern
- It is consists of 3 part
  - Model
- View
- -Controller

Angular JS-Google

- React Native…..Facebook
- Ionic-Mobile app
- Single page web application
- -nodejs
- Node –v
- Ng version –Check Angular installed or not
- How to create new project

  ng new my(appname)-app [do u want routing]… y

condt

- Automatically creating template for project
- Cmd-ng serve or ng s compiled successfully
- Open browser window type **localhost:4200**

- **ng add @angular/material**

Angular JS

- AngularJS Extends HTML
- AngularJS extends HTML with **ng-directives**.
- The **ng-app** directive defines an AngularJS application.
- The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.
- The **ng-bind** directive binds application data to the HTML view.

- The **ng-init** directive initializes AngularJS application variables.
-

```
<div ng-app="" ng-init="firstName='John'">

<p>The name is <span ng-
bind="firstName"></span></p>

</div>
```

condt

- ```
  <div data-ng-app="" data-ng-
  init="firstName='John'">

  The name is <span data-ng-
  bind="firstName">

  ```
  ```

Condt

- AngularJS expressions are written inside double braces: **{{ expression }}**.

- ```
  <!DOCTYPE html>
  <html>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
  <body>

    My first expression: {{ 5 + 5 }}

  ```

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

condt

```
<!DOCTYPE html>
<html ng-app>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1
<body >
<div >
10+20={{ 10+20 }}
</div>
<div>
10+20={{10+20}}
</div>
</html>
```

Ondt(object with properties)

```
<!doctype html>
<html ng-app>
<head>
    <script src="Scripts/angular.min.js"></script>
</head>
<body>
    <div>
        {{ {name: 'David', age:'30'}.name }}
    </div>
    <div>
        40 + 50 = {{ 40 + 50 }}
    </div>
</body>
</html>
```

condt

```
<!doctype html>
<html ng-app>
<head>
    <script src="Scripts/angular.min.js"></script>
</head>
<body>
    <div>
        {{ ['David', 'Pam', 'Sara'][2] }}
    </div>
    <div>
        40 + 50 = {{ 40 + 50 }}
    </div>
</body>
</html>
```

Module Creation/controller(anonymous function)

How to create a module

Use the angular object's module() method to create a module.

```
var myApp = angular.module("myModule", []);
```

```
var myApp = angular.module("myModule", []);

var myController = function ($scope) {
    $scope.message = "AngularJS Tutorial";
};

myApp.controller("myController", myController);
```

condt

```
//Create the module
var myApp = angular.module("myModule", []);

// Creating the controller and registering with the module all done in one line
myApp.controller("myController", function ($scope) {
    $scope.message = "AngularJS Tutorial";
});
```

Binding with object

```
var myApp = angular.module("myModule", []);

myApp.controller("myController", function ($scope) {
    var employee = {
        firstName: "Daivd",
        lastName: "Hastings",
        gender: "Male"
    };

    $scope.employee = employee;
});
```

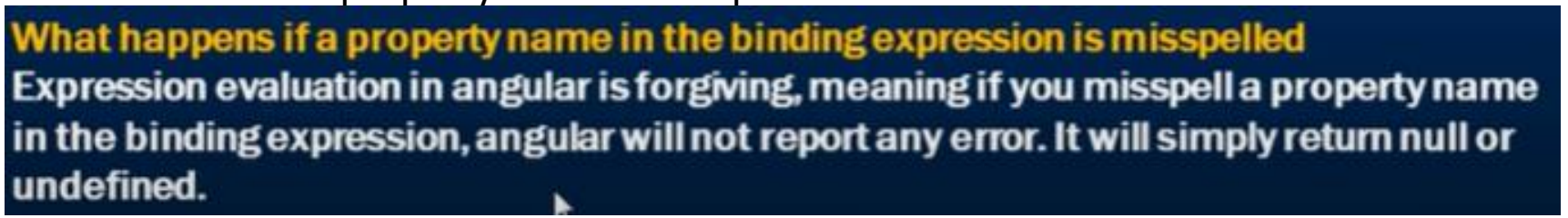
condt

```
<!doctype html>
<html ng-app="myModule">
<head>
    <script src="Scripts/angular.min.js"></script>
    <script src="Scripts/Script.js"></script>
</head>
<body>
    <div ng-controller="myController">
        <div>
            First Name : {{ employee.firstName }}
        </div>
        <div>
            Last Name : {{ employee.lastName }}
        </div>
        <div>
            Gender : {{ employee.gender}}
        </div>
    </div>
</body>
</html>
```

condt

- Case 1: if the controller name is misspelled:

Case 2: if the property name is misspelled :



condt

- Method Chaining: we can write all in one line (module, controller, registration in single line)

```
var myApp = angular
                .module("myModule", [])
                .controller("myController", function ($scope) {
                    var employee = {
                        firstName: "Daivd",
                        lastName: "Hastings",
                        gender: "Male"
                    };

                    $scope.employee = employee;
                });
```

Ng-src

```
<!doctype html>
<html ng-app="myModule">
<head>
    <script src="Scripts/angular.js"></script>
    <script src="Scripts/Script.js"></script>
</head>
<body>
    <div ng-controller="myController">
        <div>
            Name : {{ country.name }}
        </div>
        <div>
            Capital : {{ country.capital }}
        </div>
        <div>
            <img ng-src="{{ country.flag }}"
                 alt="{{ country.name + ' Flag' }}"
                 style="height:100px; width:200px" />
        </div>
    </div>
</body>
</html>
```

condt

```
var myApp = angular
                .module("myModule", [])
                .controller("myController", function ($scope) {
                    var country = {
                        name: "USA",
                        capital: "Washington, D.C.",
                        flag: "/Images/usa-flag.png"
                    };

                    $scope.country = country;
                });
```

Ng-repeat –used on an array of objects

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">
</script>
<body>

<div ng-app="" ng-init="names=['Jani','Hege','Kai']">
  <p>Looping with ng-repeat:</p>
  <ul>
    <li ng-repeat="x in names">
      {{ x }}
    </li>
  </ul>
</div>

</body>
</html>
```

Looping with ng-repeat:

- Jani
- Hege
- Kai

condt

- Ng-app –defines the root element of an Angular JS

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">
</script>
<body>

<div ng-app="" ng-init="names=[
{name:'Jani',country:'Norway'},
{name:'Hege',country:'Sweden'},
{name:'Kai',country:'Denmark'}]">

<p>Looping with objects:</p>
<ul>
  <li ng-repeat="x in names">
  {{ x.name + ', ' + x.country }}</li>
</ul>

</div>

</body>
</html>
```

Looping with objects:

- Jani, Norway
- Hege, Sweden
- Kai, Denmark

condt

```
<!DOCTYPE html>
<html>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
<body>
<div ng-app="myApp" ng-controller="myCtrl">
<h1>{{carname}}</h1>
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.carname = "Volvo";
});
</script>
<p>The property "carname" was made in the controller, and can be referred to in the view by usi
</p>
</body>
</html>
```

condt

- $Scope is a object used to bind the HTML and Java Script.

Validate User Input

- The `ng-model` directive can provide type validation for application data (number, e-mail, required):

Input controls provides data-binding by using the `ng-model` directive

```
<input type="text" ng-model="firstname">
```

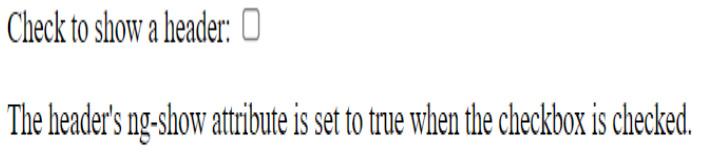
Cont

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">
</script>
<body>

<div ng-app="">
  <form>
    Check to show a header:
    <input type="checkbox" ng-model="myVar">
  </form>
  <h1 ng-show="myVar">My Header</h1>
</div>

<p>The header's ng-show attribute is set to true when the checkbox is checked.</p>

</body>
</html>
```



```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">
</script>
<body ng-app="">

<form>
  Pick a topic:
  <input type="radio" ng-model="myVar" value="dogs">Dogs
  <input type="radio" ng-model="myVar" value="tuts">Tutorials
  <input type="radio" ng-model="myVar" value="cars">Cars
</form>

<div ng-switch="myVar">
  <div ng-switch-when="dogs">
     <h1>Dogs</h1>
     <p>Welcome to a world of dogs.</p>
  </div>
  <div ng-switch-when="tuts">
     <h1>Tutorials</h1>
     <p>Learn from examples.</p>
  </div>
  <div ng-switch-when="cars">
     <h1>Cars</h1>
     <p>Read about cars.</p>
  </div>
</div>
```

Select

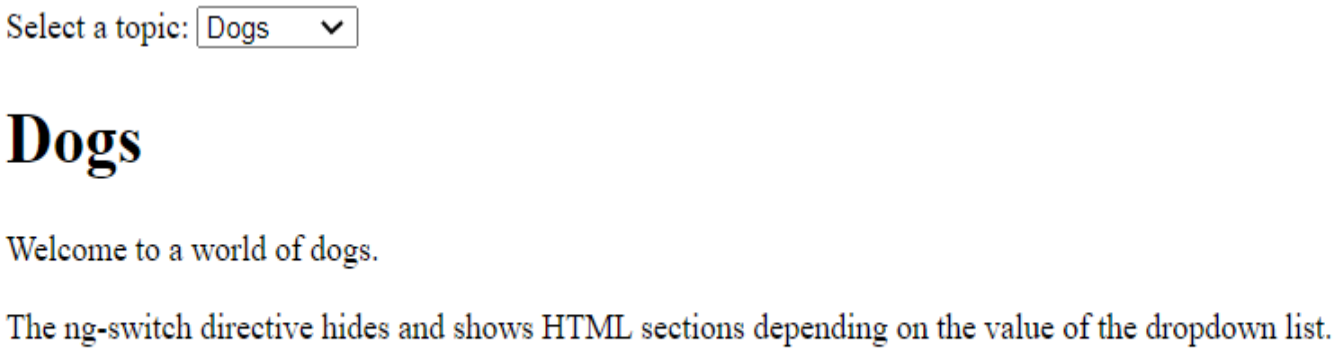
```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body ng-app="">

<form>
  Select a topic:
  <select ng-model="myVar">
    <option value="">
    <option value="dogs">Dogs
    <option value="tuts">Tutorials
    <option value="cars">Cars
  </select>
</form>

<div ng-switch="myVar">
  <div ng-switch-when="dogs">
     <h1>Dogs</h1>
     <p>Welcome to a world of dogs.</p>
  </div>
  <div ng-switch-when="tuts">
     <h1>Tutorials</h1>
     <p>Learn from examples.</p>
  </div>
  <div ng-switch-when="cars">
     <h1>Cars</h1>
     <p>Read about cars.</p>
  </div>
</div>

<p>The ng-switch directive hides and shows HTML sections depending on the value of the dropdown
list.</p>

</body>
</html>
```



Form

First Name:

John

Last Name:

Doe

RESET

Required:

Required
Use the HTML5 attribute `required` to specify that the input field must be filled out:

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body ng-app="">

<p>Try writing in the input field:</p>

<form name="myForm">
<input name="myInput" ng-model="myInput" required>
</form>

<p>The input's valid state is:</p>
<h1>{{myForm.myInput.$valid}}</h1>

</body>
</html>
```

Condt

E-mail
Use the HTML5 type `email` to specify that the value must be an e-mail:

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body ng-app="">

<p>Try writing an E-mail address in the input field:</p>

<form name="myForm">
<input type="email" name="myInput" ng-model="myInput">
</form>

<p>The input's valid state is:</p>
<h1>{{myForm.myInput.$valid}}</h1>
<p>Note that the state of the input field is "true" before you start writing in it, even if it does
not contain an e-mail address.</p>

</body>
</html>
```

condt

Input fields have the following states:

- `$untouched` The field has not been touched yet
- `$touched` The field has been touched
- `$pristine` The field has not been modified yet
- `$dirty` The field has been modified
- `$invalid` The field content is not valid
- `$valid` The field content is valid

Forms have the following states:

- `$pristine` No fields have been modified yet
- `$dirty` One or more have been modified
- `$invalid` The form content is not valid
- `$valid` The form content is valid
- `$submitted` The form is submitted

They are all properties of the form, and are either `true` or `false`.

Content Management System (Word Press)

- **The Content Management System (CMS) is a software which stores all the data such as text, photos, music, documents, etc. and is made available on your website.**
- **It helps in editing, publishing and modifying the content of the website.**
- **WordPress is an open source Content Management System (CMS), which allows the users to build dynamic websites and blog.**
- **WordPress is the most popular blogging system on the web and allows updating, customizing and managing the website from its back-end CMS and components.**

Condt.

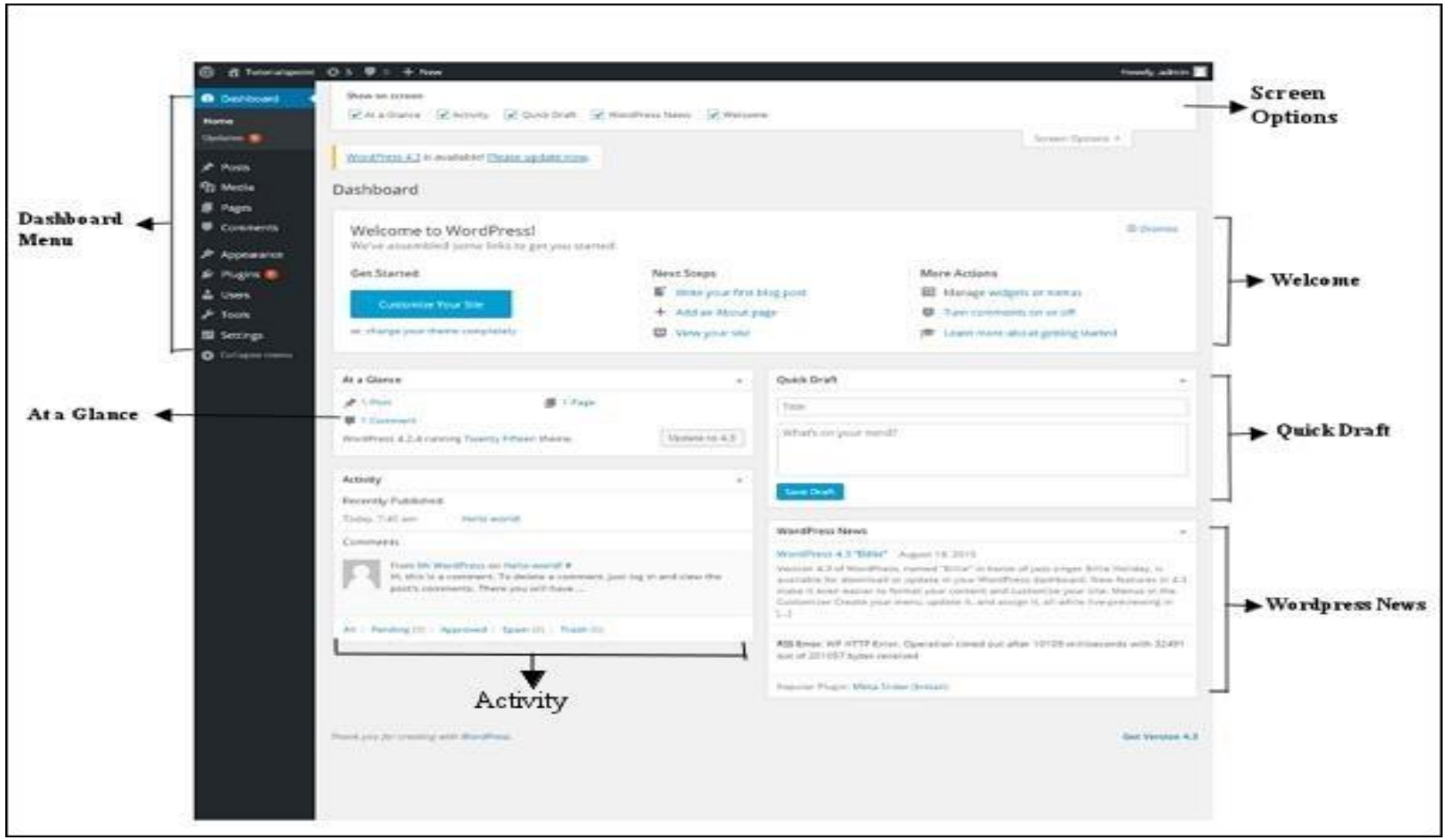
- It is software , used to manage to creation and modification of digital content.

- WordPress
- Joomla
- magento

ondt

- **User Management**
- **Media Management**
- **Theme System**
- **Extend with Plugins**
- **Search Engine Optimization**
- **Multilingual**
- **Importers**

Condt.


Screen Options
Dashboard Menu
Welcome
At a Glance
Quick Draft
Wordpress News
Activity

ASP.NET

- Dynamic web application

-Desktop application- install in every computer

-Mobile Application- designed for mobile

-Web Application- installed in centralized server

Microsoft:

ASP- VB Script

ASP.NET- Any.Net Language / C#/VB.Net

PHP-PHP Script

JSP- Java

SERVERLET.....Java

Condt

- Building Application:
- - HTML-Display
- -CSS- Style
- - JavaScript/J Query – client side script
- - ASP.Net with C#- Server side script
- - Web Server- hold data
- (IIS, Apche……) Internet Information Service