

Hypertext

- Hypertext એ text છે જે બીજી માહિતી સાથે જોડાય છે. Hypertext ડોક્યુમેન્ટ માંની લિંકને ક્લિક કરીને, ચુંબક અથવા જુદા જુદા content પર JUMP કરી શકે છે. તેથી Hypertext સામાન્ય રીતે વેબ પેજ સાથે associate થયેલું છે.

Hyperlink

- Hyperlink એ text અથવા image છે જેને તમે નવા ડોક્યુમેન્ટ પર jump કરવા માટે ક્લિક કરી શકો છો.
- જ્યારે તમે cursor ને hyperlink પર move કરો છો, પછી ભલે તે text અથવા image હોય, તો arrow લિંક પર pointing કરેલા small hand માં બદલાવો જોઈએ.
- જ્યારે તમે તેને ક્લિક કરો છો, ત્યારે current પેજ માં નવું પેજ open થશે.

History of HTML

- HTML નું પૂરું નામ Hypertext Markup Language છે. HTML એ વેબ પેજ બનાવવા માટે વપરાતી Language છે. "Hypertext" એ hyperlinks ને refers કરે છે કે જેમાં HTML પેજ શામેલ હોઈ શકે છે.
- "Markup language" એ page ના layout અને પેજ માં elements ને define કરવા માટે કઈ રીતે tagsનો ઉપયોગ કરવામાં આવે છે તેનો define કરે છે.
- HTML માં elements (tags) શામેલ છે જે indicate કરે છે કે વેબ બ્રાઉઝર એ ડોક્યુમેન્ટ કેવી રીતે રજૂ કરવું જોઈએ. HTML ડોક્યુમેન્ટ એ ફક્ત એક text file છે જેમાં માહિતી શામેલ હોય છે, જે published થવી જોઈએ. HTML પેજ નું collection World Wide Web બનાવે છે.
- Notepad જેવા simple text editor નો ઉપયોગ કરીને HTML પેજ બનાવી શકાય છે. બ્રાઉઝર આ text ફાઇલ (.htm / .html) ને interprets (અર્થધારના) કરે છે અને ક્લાંટ કમ્પ્યુટર પર બતાવે છે.
- HTML એક platform independent language છે જેનો ઉપયોગ કોઈપણ platform જેમ કે Windows, Linux વગેરે પર થઈ શકે છે. HTML ડોક્યુમેન્ટ્સ/પેજ ને .html extension થી સેવ કરી શકાય છે.
- 1989 માં HTML મૂળભૂત રીતે Tim Berners-Lee દ્વારા વિકસાવવામાં આવ્યું હતું. તે SGML (Standard Generalized Markup Language)માંથી લેવામાં આવ્યું છે.

HTTP (Hypertext Transfer Protocol)

- HTTP નું પૂરું નામ Hypertext Transfer Protocol છે. HTTP એ પ્રોટોકોલ વેબ પર ડેટા ટ્રાન્સફર કરવા માટે વપરાય છે.
- HTTP server-client મોડેલ નો ઉપયોગ કરે છે. જ્યારે તમે કોઈ વેબસાઇટ ને access કરો છો, ત્યારે તમારું બ્રાઉઝર તેને corresponding વેબ સર્વર ને request મોકલે છે અને તે HTTP status code સાથે responds આપે છે. જો URL valid હશે અને કનેક્શન આપવામાં આવ્યું હશે, તો સર્વર તમારા બ્રાઉઝર ને વેબપેજ અને તેને related ફાઇલ્સ મોકલશે.

Tags

- HTML ડોક્યુમેન્ટ્સ એ ટેગ્સ નામના special markup code ને combine કરીને બનાવવામાં આવે છે.
- ટેગ્સ ડોક્યુમેન્ટ્સ ના structure ને define કરે છે અને actual content ને રાખવા માટે framework provide કરે છે, જે text, images અથવા બીજી સ્પેશિયલ content હોઈ શકે છે.

- બધા HTML ટેગ્સ એ angle brackets '<' અને '>' એટલે કે Opening tag <HTML> અને closing tag </HTML> માં બંધાયેલા છે.
- ટેગ્સ એ case sensitive નથી.
- બે ટાઇપ ના ટેગ્સ છે:
 - **Container Tags:**
 - ટેગ્સ જેમાં opening અને closing થતાં બંને હોય છે i.e. <TAG> અને </TAG> ને container tags કહેવામાં આવે છે.
 - **Empty Tags:**
 - ટેગ્સ જેને ફક્ત opening છે અને કોઈ ending નથી તેને empty tags કહેવાય.

Attributes

- બધા HTML elements માં attributes હોઈ શકે છે.
- Attributes element વિશે વધારાની માહિતી આપે છે.
- Start tag માં attributes હુંમેશાં specify થાય છે.
- Attributes સામાન્ય રીતે name/value ની pairs માં આવે છે જેમ કે: name="value".

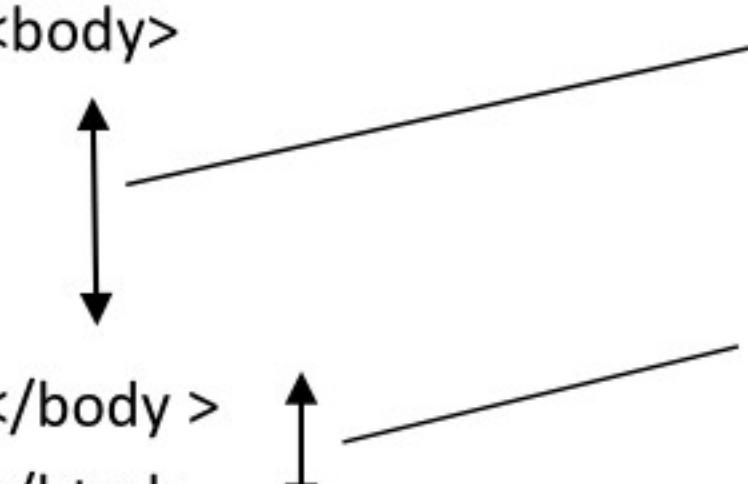
Example:

```
<body>
    <p align="left"> This is the left aligned </p>
<body>
```

Attribute names અને attribute values એ case-insensitive છે.

Structure of HTML document

```
<html>
  <head>
    <title></title >
  </head >
  <body>
    The text is added here
  </body >
  </html>
```



The text is added here
These tags must appear at
the end of the HTML page

<html>

- આ ટેગનો ઉપયોગ એ દર્શાવા માટે થાય છે કે આ એક HTML ડોક્યુમેન્ટ છે. HTML ડોક્યુમેન્ટસ આ ટેગ સાથે start અને end થવું જોઈએ.

Example:

```
<html>
```

Your title and document goes here.

</html>

<head>

- આ ટેગનો ઉપયોગ HTML ડોક્યુમેન્ટના header section ને દર્શાવા માટે થાય છે, જેમાં સામાન્ય રીતે <title> અને <meta> ટેગ્સ શામેલ હોય છે.

Example:

```
<head>
    <title> First Web Page</title>
</head>
```

<title>

- આ HTML પેજનું title સૂચવે છે. જ્યારે તમે કોઈ વેબ પેજ જુઓ છો ત્યારે title તમારા બ્રાઉઝર ના upper left corner પર display થાય છે.

Example:

```
<head>
    <title> First Web Page</title>
</head>
```

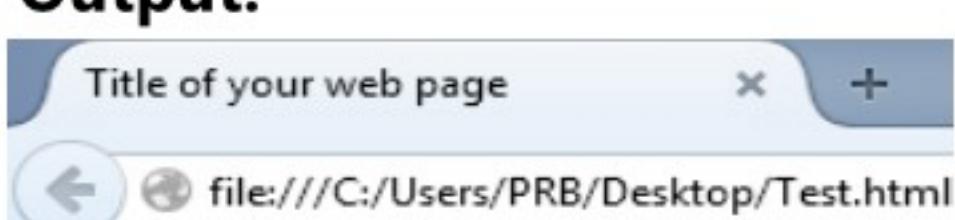
<body>

- <body> ટેગમાં ડોક્યુમેન્ટનું HTML body શામેલ છે. <Body> ટેગની અંદરની દરેક વસ્તુ બ્રાઉઝર પર display થાય છે.
- <body> ટેગમાં ધણા �attributes હોઈ શકે છે. સૌથી વધુ ઉપયોગમાં લેવાયેલા attributes નીચે list છે:
 - bgcolor:** આ સંપૂર્ણ HTML ડોક્યુમેન્ટનો background color છે, અને કાં તો તે color name દ્વારા અથવા six-digit hex code દ્વારા specify કરી શકાય છે.
 - topmargin:** બ્રાઉઝર વિન્ડો ના top માંથી margin.
 - leftmargin:** બ્રાઉઝર વિન્ડો ના left થી margin.

Example:

```
<html>
<head>
    <title>Title of your web page</title>
</head>
<body>HTML web page contents</body>
</html>
```

Output:



HTML web page contents

Basic HTML Tags

<h1> to <h6>

- <h1> થી <h6> ટેસનો ઉપયોગ HTML heading define કરવા માટે થાય છે.
- <h1> સૌથી largest headingને specify કરે છે અને <h6> smallest heading ને defines કરે છે.

Example:

```
<html>
<body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
</body>
</html>
```

Output:

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

<p>

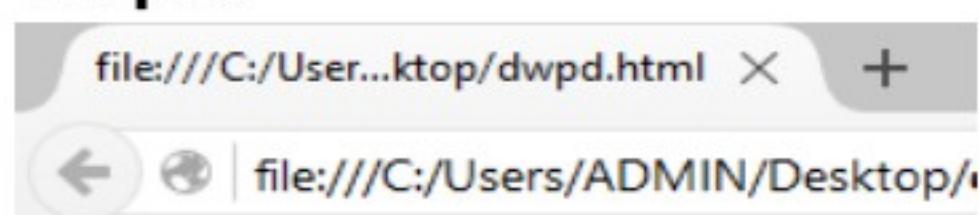
- <p> ટેગ એ text ની paragraph ને specify કરે છે.

Attribute	Value	Description
align	left, right, center, justify	Paragraph માં text alignment ને specify કરે છે.

Example:

```
<html>
<body>
    <p>This is a paragraph. </p>
</body>
</html>
```

Output:



This is a paragraph.

<hr>

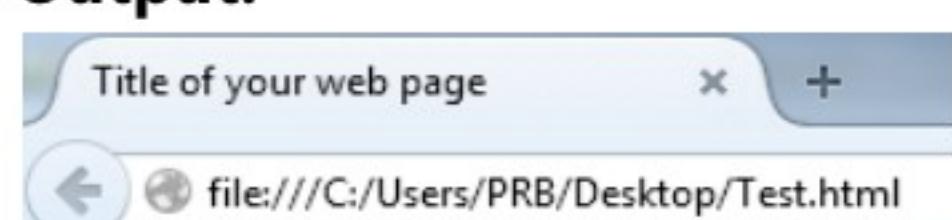
- <hr> ટેગનો ઉપયોગ horizontal line બનાવવા માટે થાય છે. આને HTML માં Horizontal Rule પણ કહેવામાં આવે છે.

Attribute	Value	Description
align	left, right, center, justify	Horizontal rule ની alignment ને specify કરે છે.
size	pixels or %	Horizontal rule ની height ને specify કરે છે.
width	pixels or %	Horizontal rule ની width ને specify કરે છે.

Example:

```
<html>
<head>
    <title>Title of your web page </title>
</head>
<body>
    HTML web page contents
    <hr align="left" width="100px">
</body>
</html>
```

Output:



HTML web page contents

- ટેગ કોઈ HTML પેજમાં image ને define કરે છે.

Attribute	Value	Description
src	URL	Image ની URL ને specify કરે છે.
height	pixels or %	Image ની height ને specify કરે છે.
width	pixels or %	Image ની width ને specify કરે છે.
alt	text	કોઈ image માટે alternate text ને specify કરે છે.
border	pixels or %	Image ની around ની border ની width specify કરે છે.

Example:

```
<html>
    <head>
        <title>Title of your web page</title>
    </head>
    <body>HTML web page contents
        
    </body>
</html>
```

Output:



<embed>

- <embed> ટેગ એક external application અથવા interactive content (a plug-in) માટે container ને define કરે છે.

Attribute	Value	Description
height	pixels or %	Embedded કરેલ contentની height ને specify કરે છે.
src	URL	Embedded કરવા માટે external file નાં address ને specify કરે છે.
type	media_type	Embedded content ની media file ને specify કરે છે.
width	pixels or %	Embedded કરેલ contentની width ને specify કરે છે.

Example:

```
<html>
    <head>
        <title>Title of your web page</title>
    </head>
    <body>
        <embed src="helloworld.swf" ></embed>
    </body>
</html>
```

<bgsound>

- HTML <bgsound> ટેગનો ઉપયોગ background માં સાઉન્ડ લે કરવા માટે થાય છે. આ ટેગ ફક્ત Internet Explorer માટે છે.

Attribute	Value	Description
loop	number	અમુક યોક્કસ સંઘા માં background માં soundtrack ફરીથી ચાલવવા માટે વપરાય છે.
src	URL	સાઉન્ડ ફાઇલનો path specify કરે છે.

Example:

```
<bgsound src="audio.mp3" loop="infinite">
```

- ટેગ એ font face, font size, અને text ના color ને specify કરે છે.

Attribute	Value	Description
color	rgb(x,x,x), #xxxxxxxx, colorname	Text ના color ને specify કરે છે.
face	font_family	Text ના font ને specify કરે છે.
size	number	Text ની size ને specify કરે છે.

Example:

```
<html>
  <head>
    <title>Title of your web page</title>
  </head>
  <body>HTML web page contents
    <font size="4" face="Arial" color="red">Hello</font>
  </body>
</html>
```

Output:

HTML web page contents **Hello**

<center>

- The <center> ટેગ નો ઉપયોગ text ને center-align કરવા માટે થાય છે.

Example:

```
<center>This is centered text.</center>
```

<marquee>

- HTML marquee scrolling piece અથવા text ના બ્લોક સેટિંગ પર આધાર રાખીને તમારા વેબપેજ ની horizontally અથવા vertically move થાય છે.

Attribute	Value	Description
width	pixels or %	આ marquee ની width ને specify કરે છે.
height	pixels or %	આ marquee ની heightને specify કરે છે.
direction	up, down, left, right	Contentને scroll કરવાની direction define કરે છે.

Example:

```
<marquee direction="up">This text will scroll from bottom to up</marquee>
```

```
<marquee direction="right">This text will scroll from left to right</marquee>
```

<a>

- <a> ટેગ hyperlink define કરે છે, જેનો ઉપયોગ એક પેજથી બીજા પેજ પર link કરવા માટે થાય છે.
- <a> element નો મહત્વનો attribute એ href attribute છે.
- default રૂપે, બધા બ્રાઉઝર માં links નીચે મુજબ દેખાશે:
 - એક unvisited link નીચે line અને blue છે.
 - એક visited link નીચે line અને purple છે.
 - એક active link નીચે line અને red છે.

Attribute	Value	Description
href	URL	<p>Link જે પેજ પર જાય છે તે URL ને specify કરે છે.</p> <p>An absolute URL – અન્ય વેબ સાઇટ પર point કરે છે. (href="http://www.example.com/default.html")</p> <p>A relative URL – કોઈ વેબસાઇટ ની અંદર ની વેબસાઇટ ની ફાઇલ ને point કરે છે. (like href= "default.htm ")</p>
target	_blank _parent _self _top framename me	Link થયેલ ડોક્યુમેન્ટ ક્યા open કરવું તે specify કરે છે.

Example:

```
<a href= "page.html" target="_blank" >Next Page</a>
```

List Tags

- ટેગ એ list item ને define કરે છે.
- The ટેગ ordered lists (), unordered lists () અને menu lists (<menu>) નો ઉપયોગ કરે છે.
- અને <menu> માં, લિસ્ટની આઇટમ્સ સામાન્ય રીતે બુલેટ પોઇન્ટ સાથે દર્શાવવામાં આવશે.
- માં, લિસ્ટ આઇટમ્સ સામાન્ય રીતે નંબરો અથવા અક્ષરો સાથે દર્શાવવામાં આવશે.

Attribute	Value	Description
value	number	ફક્ત લિસ્ટ માટે. લિસ્ટ આઇટમને start value તરીકે ઉલ્લેખિત કરે છે.

Example:

```
<ol>
  <li value="100">Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
  <li>Water</li>
</ol>
```

Output:

100. Coffee
101. Tea
102. Milk
103. Water

- HTML ટેગ નો ઉપયોગ unordered list બનાવવા માટે થાય છે.

Attribute	Value	Description
type	square, circle, disc	List નાં style ને specify કરે છે.

Example:

```
<ul type="square">
  <li>Cricket</li>
  <li>Football</li>
  <li>Hockey</li>
</ul>
```

Output:

- Cricket
- Football
- Hockey

- The HTML ટેગ નો ઉપયોગ ordered list બનાવવા માટે થાય છે.

Attribute	Value	Description
reversed	reversed	Specify કરે છે કે list order માં descending હોવું જોઈએ (9,8,7...).
start	number	Ordered listની start વેલ્યુ ને specify કરે છે.
type	A, a, I, i, 1	List નાં style ને specify કરે છે.

Example:

```
<ol type="1" reversed start=5>
  <li>Cricket</li>
  <li>Football</li>
  <li>Hockey</li>
```


Output:

5. Cricket
4. Football
3. Hockey

Working with Table

<table>

- <table> ટેગ HTML ટેબલ ને define કરે છે.
- એક HTML ટેબલ માં <table> element અને એક અથવા વધુ <tr>, <th>, અને <td> elements શામેલ હોય છે.
- <tr> element એ ટેબલ row ને define કરે છે, <th> element એ ટેબલ header ને define કરે છે, અને <td> element એ ટેબલ cell ને define કરે છે.

Attribute	Value	Description
align	right, left, center, justify	List ના order ને specify કરે છે. (descending).
bgcolor	rgb(x,x,x), #xxxxxxxx, colorname	List ને start કરવા માટે initial નંબર ને specify કરે છે.
cellpadding	pixels	Cell wall અને cell contentની વચ્ચે ની spaceને specify કરે છે.
cellspacing	pixels	Cells વચ્ચે ની space ને specify કરે છે.

<th>

- <th> tag HTML ટેબલ માં header cell ને define કરે છે.
- <th> element માંનો text bold અને default રૂપે centered હોય છે.

Attribute	Value	Description
bgcolor	rgb(x,x,x), #xxxxxxxx, colorname	Header cell ના background color ને specify કરે છે.
valign	top, middle, bottom	Header cell માં vertical માં content ગોઠવાયેલ છે.
align	right, left, center, justify	Header cell માં content ગોઠવાયેલ છે.
colspan	number	Header cell ના column ની સંખ્યા ને specify કરે છે.
<td>number</td> <td>Header cell ના row ની સંખ્યા specify કરે છે.</td>	number	Header cell ના row ની સંખ્યા specify કરે છે.
height	pixels or %	Header cell ની height ને specify કરે છે.
width	pixels or %	Header cell ની width ને specify કરે છે.

<td>

- The <td> ટેગ એ HTML ટેબલમાં column ને define કરે છે. Column એ દરેક row ની અંદર define થયેલ હોય છે.

Attribute	Value	Description
bgcolor	rgb(x,x,x), #xxxxxx, colorname	Cell ની background color ને specify કરે છે.
v-align	top, middle, bottom	Vertical cell માં content ને ગોઠવે છે.
align	right, left, center, justify	Cell માં content ને ગોઠવે છે.
colspan	number	આ column occupies કરેલા column ની સંખ્યા ને specify કરે છે.
rowspan	number	આ row occupies કરેલા row ની સંખ્યા ને specify કરે છે.
height	pixels or %	Cell ની height ને specify કરે છે.
width	pixels or %	Cell ની width ને specify કરે છે.

<tr>

- The <tr> ટેગ એ HTML ટેબલ માં row ને define કરે છે.

Attribute	Value	Description
align	right, left, center, justify	ટેબલ row માં content ને ગોઠવે છે.
bgcolor	rgb(x,x,x), #xxxxxx, colorname	ટેબલ row માટે background color ને specify કરે છે.
v-align	top, middle, bottom	ટેબલ row માટે content ને ગોઠવે છે.

Example 1:

```
<table border= "1">
  <tr>
    <th>Column 1 Header</th>
    <th>Column 2 Header</th>
  </tr>
  <tr>
    <td>Row 1 Col 1</td>
    <td>Row 1 Col 2</td>
  </tr>
  <tr>
    <td>Row 2 Col 1</td>
    <td>Row 2 Col 2</td>
  </tr>
</table>
```

Output:

Column 1 Header	Column 2 Header
Row 1 Col 1	Row 1 Col 2
Row 2 Col 1	Row 2 Col 2

Example 2:

```
<table border= "1">
    <tr>
        <th>Column 1 </th>
        <th>Column 2 </th>
        <th>Column 3</th>
    </tr>
    <tr>
        <td rowspan="2">Row 1 Cell 1</td>
        <td>Row 1 Cell 2</td>
        <td>Row 1 Cell 3</td>
    </tr>
    <tr>
        <td>Row 2 Cell 2</td>
        <td>Row 2 Cell 3</td>
    </tr>
    <tr>
        <td colspan="3">Row 3 Cell 1</td>
    </tr>
    </table>
</body>
```

Output:

Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

<div>

- HTML <div> ટેગનો ઉપયોગ તમારા ડોક્યુમેન્ટના સેક્શનને define કરવા માટે થાય છે. div ટેગ સાથે, તમે HTML elementના મોટાભાગના સેક્શનને એક્સાથે ગ્રૂપ કરી શકો છો અને તેમને CSS સાથે જોડી શકો છો.

Attribute	Value	Description
align	right, left, center, justify	<div> element ની અંદર content નું alignment specify કરે છે.

Example:

```
<div style="color: Red">
    <h3>This is a heading in a div element</h3>
    <p>This is some text in a div element.</p>
</div>
```

Output:

This is a heading in a div element

This is some text in a div element.

Special Character

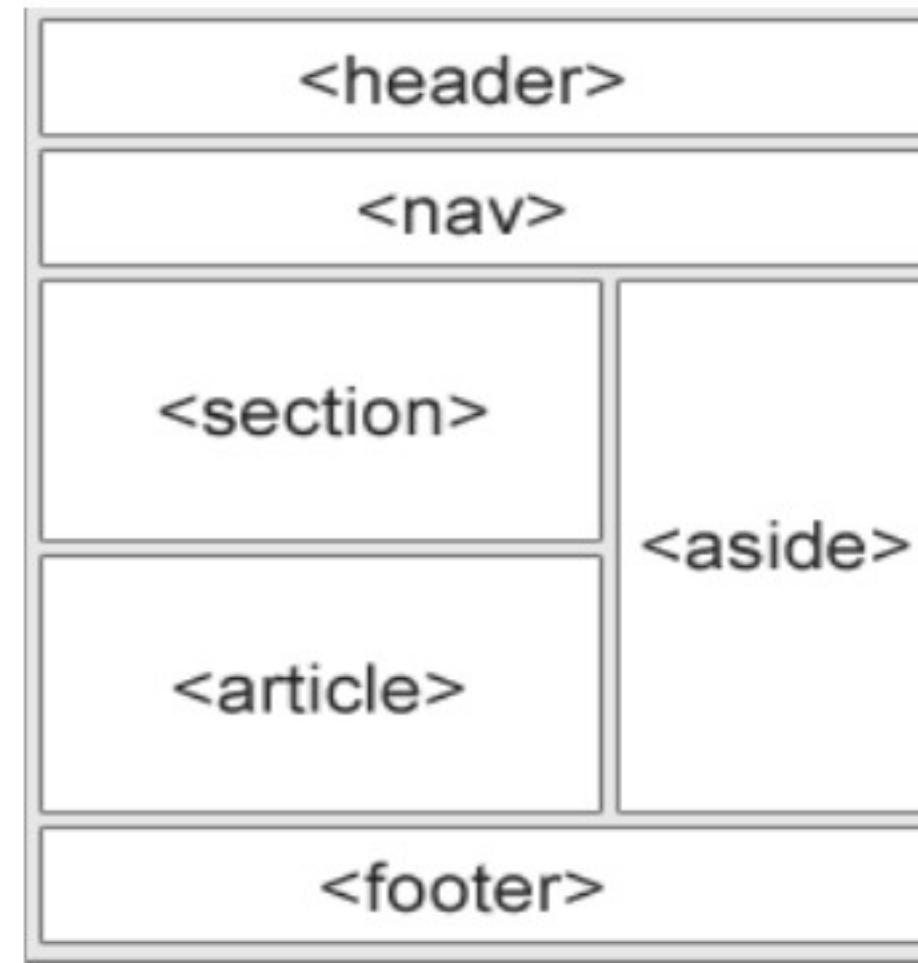
- સ્પેશિયલ characters અને સિમ્બોલ્સ એ સ્પેશિયલ character entities નો ઉપયોગ કરીને કીબોર્ડ પર મળી શકતા નથી.
- આ characters HTML માં માન્ય છે કારણ કે તેઓ ampersand (&) થી શરૂ થાય છે અને semi-colon સાથે અંત થાય છે.
- કેટલાક સ્પેશિયલ character છે:

Sr. No.	Result	Description	Entity Name
1		non-breaking space	&nbsp
2	<	less than	<
3	>	greater than	>
4	&	ampersand	&
5	¢	cent	¢
6	£	pound	£
7	¥	yen	¥
8	€	euro	€
9	©	copyright	©
10	®	registered trademark	®

New Semantic Elements in HTML5

- ધારી વેબસાઇટ માં HTML code શામેલ હોય છે જેમ કે: <div id = "nav"> <div class = "header"> <div id = "footer"> navigation, header અને footer સૂચવવા માટે.
- HTML5 વેબપેજ ના વિવિધ ભાગોને define કરવા માટે નવા semantic elements offer કરે છે.

- <article>
- <aside>
- <footer>
- <header>
- <nav>
- <section>
- <main>
- <summary>
- <time>
- <figure>



<section>

- <section> element એક ડોક્યુમેન્ટ માં એક સેક્શન define કરે છે.
- એક સેક્શન content નું ગ્રૂપ છે, સામાન્ય રીતે heading સાથે.
- વેબ સાઇટનું હોમ પેજ introduction, content અને contact information માટે સેક્શનમાં જુદું થઈ શકે છે.

<article>

- <article> element independent, self-contained content ને specify કરે છે.
- તે બાકીની વેબસાઇટ થી independently રીતે read શક્ય છે.
- જ્યાં <article> element નો ઉપયોગ કરી શકાય છે તેના ઉદાહરણો
 - Forum post
 - Blog post
 - Newspaper article

<header>

- <header> element ડોક્યુમેન્ટ અથવા સેક્શન માટે headerને specify કરે છે.
- <header> element નો introductory content માટે container તરીકે ઉપયોગ થવો જોઈએ.
- તમારી પાસે એક ડોક્યુમેન્ટમાં ધણા <header> element હોઈ શકે છે.

<footer>

- footer element ડોક્યુમેન્ટ અથવા સેક્શન માટે <footer> સૂચવે છે.
- Footer માં સામાન્ય રીતે ડોક્યુમેન્ટ ના author, copyright information, ઉપયોગની terms ની link, contact information વગેરે શામેલ હોય છે.
- તમારી પાસે એક ડોક્યુમેન્ટ માં ધણા <footer> elements હોઈ શકે છે.

<nav>

- <nav> element navigation links ના સેટને define કરે છે.
- <nav> element એ navigation links ના મોટા બ્લોક્સ માટે બનાવાયેલ છે. જો કે, કોઈ ડોક્યુમેન્ટ માંની બધી links <nav> element ની અંદર હોવી જોઈએ નહીં.

<aside>

- <aside> element તે content ને બાજુએ રાખેલી content (sidebar જેવી) માંથી કેટલોક content define કરે છે.
- <aside> content ને આસપાસના content થી સંબંધિત હોવી જોઈએ.

Example:

```
<html>
    <head>
        <title>Title of your web page</title>
    </head>
    <body>
        <header>
            <h1>This is page heading</h1>
        </header>
        <nav>
            <ul>
                <li><a href="#">Home</a></li>
                <li><a href="#">About Us</a></li>
                <li><a href="#">Contact Us</a></li>
            </ul>
        </nav>
        <article>
            <h1>This is article heading</h1>
            <p>Hello world! Hello world! Hello world!</p>
        </article>
        <aside>
            <figure>
                
            </figure>
        </aside>
        <section>
            <h1>This is a section heading</h1>
            <p>Hello world! Hello world! Hello world! </p>
        </section>
        <footer>
            <hr/>Copyright (C) 2013. All rights reserved.
        </footer>
    </body>
</html>
```

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

Output:

This is page heading

- [Home](#)
- [About Us](#)
- [Contact Us](#)

This is article heading

Hello world! Hello world! Hello world!



This is a section heading

Hello world! Hello world! Hello world!

Copyright (C) 2013. All rights reserved.

HTML5 Form Input Types

Input Type	Description	Example
number	તે ઈનપુટ ફીલ્ડ્સ માટે ઉપયોગમાં લેવાય છે જેમાં ન્યુમેરિક વેલ્ચુ હોવી જોઈએ.	Quantity (between 1 and 5): <input type="number" name="quantity" min="1" max="5">
Date	તેનો ઉપયોગ ઈનપુટ ફીલ્ડ્સ માટે થાય છે જેમાં date હોવી જોઈએ.	Birthday: <input type="date" name="bday">
color	તેનો ઉપયોગ ઈનપુટ ફીલ્ડ્સ માટે થાય છે જેમાં color હોવો જોઈએ.	Select your favorite color: <input type="color" name="favcolor">
range	તેનો ઉપયોગ ઈનપુટ ફીલ્ડ્સ માટે થાય છે. જેમાં રેન્જ ની અંદર વેલ્ચુ હોવી જોઈએ.	<input type="range" name="points" min="0" max="10">
month	તે યુઝર ને એક month અને year માટે સિલેક્ટ કરવા દે છે.	Birthday (month and year): <input type="month" name="bdaymonth">
week	તે યુઝર ને એક week અને year માટે સિલેક્ટ કરવા દે છે.	Select a week: <input type="week" name="week_year">

time	તે યુઝર ને time (no time zone) સિલેક્ટ કરવાની મંજૂરી આપે છે.	Select a time: <input type="time" name="usr_time">
datetime-local	તે યુઝરને date અને time (no time zone) સિલેક્ટ કરવાની મંજૂરી આપે છે.	Birthday (date and time): <input type="datetime-local" name="bdytime">
email	તે ઈનપુટ ફીલ્ડસ માટે વપરાય છે જેમાં ઇમેલ એડ્રેસ હોવું જોઈએ.	E-mail: <input type="email" name="email">
search	તે search ફીલ્ડસ માટે વપરાય છે (સર્ચ ફીલ્ડ રેખ્યુલર ફીલ્ડની જેમ વર્તે છે).	Search Google: <input type="search" name="googlesearch">
tel	તેનો ઉપયોગ ઈનપુટ ફીલ્ડસ માટે થાય છે જેમાં ટેલિફોન નંબર હોવો જોઈએ.	Telephone: <input type="tel" name="usrtel">
url	તે ઈનપુટ ફીલ્ડસ માટે ઉપયોગમાં લેવાય છે જેમાં URL એડ્રેસ હોવું જોઈએ.	Add your homepage: <input type="url" name="homepage">

HTML5 Form Elements

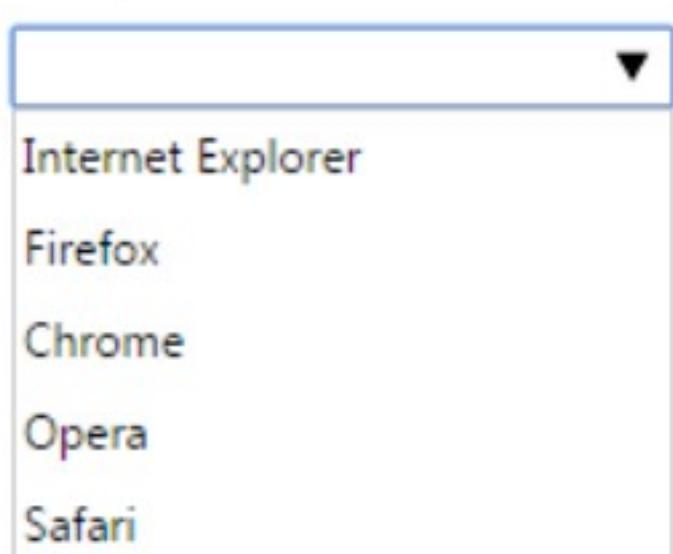
<datalist>

- <**datalist**> element એ <input> element માટે pre-defined options ના લિસ્ટ ને specify કરે છે.
- યુઝરસ ને ઈનપુટ ડેટા તરીકે pre-defined options નું drop-down list જોશે.
- <input> element ના list attribute એ <**datalist**> element ના id attribute ને refer કરવો આવશ્યક છે.

Example:

```
<input list="browsers" name="browser">
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Output:



<keygen>

- <keygen> element નો purpose યુઝર ને ઓથેન્ટિકેટ કરવા માટે એક secure way આપે છે.
- <keygen> element ફોર્મ માં key-pair જનરેટર ફિલ્ડને specify કરે છે.
- જ્યારે ફોર્મ સબમિટ થાય છે, ત્યારે બે કી જનરેટ થાય છે, એક private અને એક public.
- Private key locally સ્ટોર થાય છે, અને public key સર્વર પર સેન્ડ કરવામાં આવે છે.
- Public key નો ઉપયોગ ભવિષ્ય માં યુઝર ને ઓથેન્ટિકેટ કરવા માટે ક્લાઇંટ સર્ટિફિકેટ જનરેટ કરવા માટે થઈ શકે છે.

Example:

```
<form action="demo_keygen.asp" method="get">
    Username: <input type="text" name="usr_name">
    Encryption: <keygen name="security">
    <input type="submit">
</form>
```

Output:

Username:

Encryption: High Grade ▾

<output>

- <output> element ના કેલક્યુલેશન ના રિઝલ્ટ રજૂ કરે છે (જેમ કે script દ્વારા perform કરવામાં આવે છે).

Attribute	Description
for	કેલક્યુલેશન ના રિઝલ્ટ અને કેલક્યુલેશન માં ઉપયોગમાં લેવાતા elements વચ્ચેની રિલેશનશીપ specify કરે છે.
form	Output element સાથે સંકળાયેલ એક અથવા વધુ ફોર્મ્સ્ સ્પેચ્યુલેટ કરે છે.
name	Output element માટે name specify કરે છે.

Example:

```
<form oninput= "result.value = parseInt(a.value) + parseInt(b.value)">
    <input type="range" name="b" value="50" /> +
    <input type="number" name="a" value="10" /> =
    <output name="result">60</output>
</form>
```

Output:

 + = 60

HTML5 Video

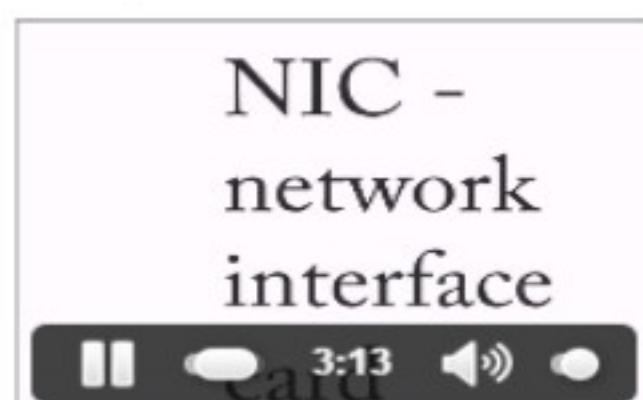
- HTML5 پહેલાં, વેબપેજ પર વિડિયો બતાવવા માટે કોઈ standard નથી.
- HTML5 પહેલાં, વિડિયો ફક્ત plug-in (જેમ કે flash) સાથે જ ખે કરી શકાય છે.
- HTML5 <video> element એ વેબપેજ માં વિડિયો ને embed કરવા માટે એક standard રીતને specify કરે છે.
- HTML Video - Media Types: MP4, WebM, Ogg.

Attribute	Description
autoplay	Specify કરે છે કે વિડિયો ready થાય ત્યારે ખે થાય છે.
src	વિડિયો file ના URL ને specify કરે છે.
controls	Specify કરે છે કે વિડિયો controls display થવું જોઈએ (જેમ કે play/pause બટન વગેરે).
loop	Specify કરે છે કે જ્યારે વિડિયો પૂરો થાય ત્યારે વિડિયો એ ફરીથી શરૂ થાય છે.
height	વિડિયો ખેયર ની height સેટ કરે છે.
width	વિડિયો ખેયર ની width સેટ કરે છે.

Example:

```
<video width="200px" controls autoplay>
  <source src="Networking Basics.mp4" type="video/mp4">
  Your browser does not support HTML5 video.
</video>
```

Output:



HTML5 Audio

- HTML5 પહેલાં, વેબપેજ પર ઓડિઓ ફાઇલ ને ખે કરવા માટે કોઈ standard નથી.
- HTML5 પહેલાં, ઓડિઓ ફાઇલ ફક્ત plug-in (જેમ કે flash) સાથે જ ખે કરી શકાય છે.
- HTML5 <audio> element વેબપેજ માં ઓડિઓ ને embed કરવા માટે standard રીતને specify કરે છે.
- HTML Audio - Media Types: MP3, Ogg, Wav.

Attribute	Description
autoplay	Specify કરે છે કે ઓડિઓ રેડી થાય ત્યારે ખે થાય છે.
src	ઓડિઓ ફાઇલ ના URL ને specify કરે છે.
controls	Specify કરે છે કે ઓડિઓ controls display થવું જોઈએ (જેમ કે play/pause બટન વગેરે)
loop	Specify કરે છે કે જ્યારે ઓડિઓ પૂરો થાય ત્યારે વિડિયો એ ફરીથી શરૂ થાય છે.

Example:

```
<audio controls autoplay>
  <source src="audio.mp3" type="audio/mp3">
    Your browser does not support the audio element.
</audio>
```

Output:



Difference between HTML 4.0 & 5.0

HTML 4.0	HTML 5.0
HTML4 मાં, DOCTYPE declaration ખૂબ જ અધરું અને લાંબુ છે.	DOCTYPE element ને ખૂબ જ સરળ બનાવમાં આવ્યું છે.
HMTL4, header, footer અને columns વગેરે જેવા સૌથી સામાન્ય વેબપેજ structure નો ઉપયોગ કરે છે	HTML5 વેબપેજને structure કરવા માટે ધણાં inbuilt elements આપવામાં આવે છે. ઉદાહરણ તરીકે, header, footer, nav, article અને section વગેરે.
HTML 4 ઓડિઓ અથવા વિડિયો directly embed કરી શકતા નથી. તે તેના માટે ફ્લેશ પ્લેયર નો ઉપયોગ કરે છે.	HTML5 ઓડિઓ અથવા વિડિયો ને ફ્લેશ પ્લેયર વગર embed કરી શકે છે.
HTML 4 એચ્યોક્સ syntax ને હેન્ડલ કરી શકતું નથી.	HTML 5 એચ્યોક્સ syntax ને હેન્ડલ કરી શકે છે.
HTML4 વેક્ટર ગ્રાફિક્સ માટે inbuilt ટેગ પૂરું પાડતું નથી.	તે inbuilt ટેગ જેવા કે canvas આપે છે જેનો ઉપયોગ વેક્ટર ગ્રાફિક્સ માટે થાય છે.
HTML4 માં, વેબસાઇટ ની visit લેનારા યુઝરસ નું જીઓગ્રાફિકલ લોકેશન મેળવવા માટે તે ખૂબ જ લાંબી કાર્ય હતું.	HTML5 માં, વેબસાઇટની visit લેનારા યુઝરના જીઓગ્રાફિકલ લોકેશન ને મેળવવાનું ખૂબ જ સરળ છે.
HTML4 માં, બ્રાઉઝર cache નો ઉપયોગ temporary સ્ટોરેજ માટે થાય છે.	આ Web SQL ડેટાબેસ માં, Application Cache અને વેબ સ્ટોરેજ ક્લાયિટ સાઇટ સ્ટોરેજ માટે ઉપલબ્ધ છે.
બધા જુના બ્રાઉઝર સાથે વર્ક કરે છે.	મોટા ભાગના મોર્ડન બ્રાઉઝર એ HTML5 specification ને supporting આપવાનું શરૂ કર્યું છે દિ.ત. Firefox, Mozilla, Opera, Chrome, Safari વગેરે.

What is CSS? And Define CSS.

- CSS એ **Cascading Style Sheets** માટે વપરાય છે.
- CSS એ define કરે છે કે **HTML elements** કેવી રીતે **display** કરવો છે.
- CSS ઘણાં બધા વર્ક ને save કરે છે જેનો અર્થ એ છે કે ઘણી styles સાથે સરખા content નો ઉપયોગ કરી તેમાં ફેરફાર કર્યા વગર તેનો ઉપયોગ થઈ શકે છે.
- Style sheets તે presentation ને પેજના content સાથે interfere કર્યા વિના specify કરવામાં capable છે.

Uses of CSS

- CSS એ **time** બચાવે છે કારણ કે CSS સાથે, આપણે ફક્ત એક જ વાર કોઈપણ element માટે content સેટિંગ ડિટેલ્સ ને specify કરવું પડશે. CSS માં જ્યારે પણ તે element આવે ત્યારે ચોક્કસ styles apply કરે છે.
- CSS માં ફાઇલ ની size ખૂબ નાની છે તેથી તમારી વેબસાઇટ **minimum loading time** લે છે.
- Element ની style બદલવા માટે, ફક્ત એક જ સ્થાને એડિટ કરવું પડે છે એટલે તેને maintain કરવો સહેલો છે.
- HTML કરતા CSS attribute નો મોટો એરે છે. તેથી CSS બનાવતી વખતે તમે વેબ ડિઝાઇન ને **flexible બનાવી શકો છો.**

Syntax of CSS

- CSS syntax ત્રણ ભાગથી બનેલી છે: **a selector, a property અને a value.**

```
• Syntax: selector {
    property: value;
}
```

- Selector સામાન્ય રીતે HTML element/tag છે જેને તમે define કરવા માંગો છો.
- Property એ attribute છે જે તમે બદલી શકો છો અને દરેક property એ વેલ્યુ મેળવી શકે છે.
- **Colon** દ્વારા property અને વેલ્યુને અલગ કરી શકાય છે.
- CSS declaration હંમેશા semicolon સાથે ends થાય છે અને declaration ગુપ્સ curly braces દ્વારા enclose હોય છે.
- નીચે આપેલા કોડનો ઉપયોગ કરીને, બધા <p> elements ગ્રીન ટેક્સ્ટ કલર સાથે center-aligned હશે.

Example: p {

```
    color: red; text-align: center;
}
```

Types of CSS

- CSS ના ત્રણ ટાઈપ છે:
 - 1) Inline style
 - 2) Embedded/Internal style

3) External/Linked style sheet

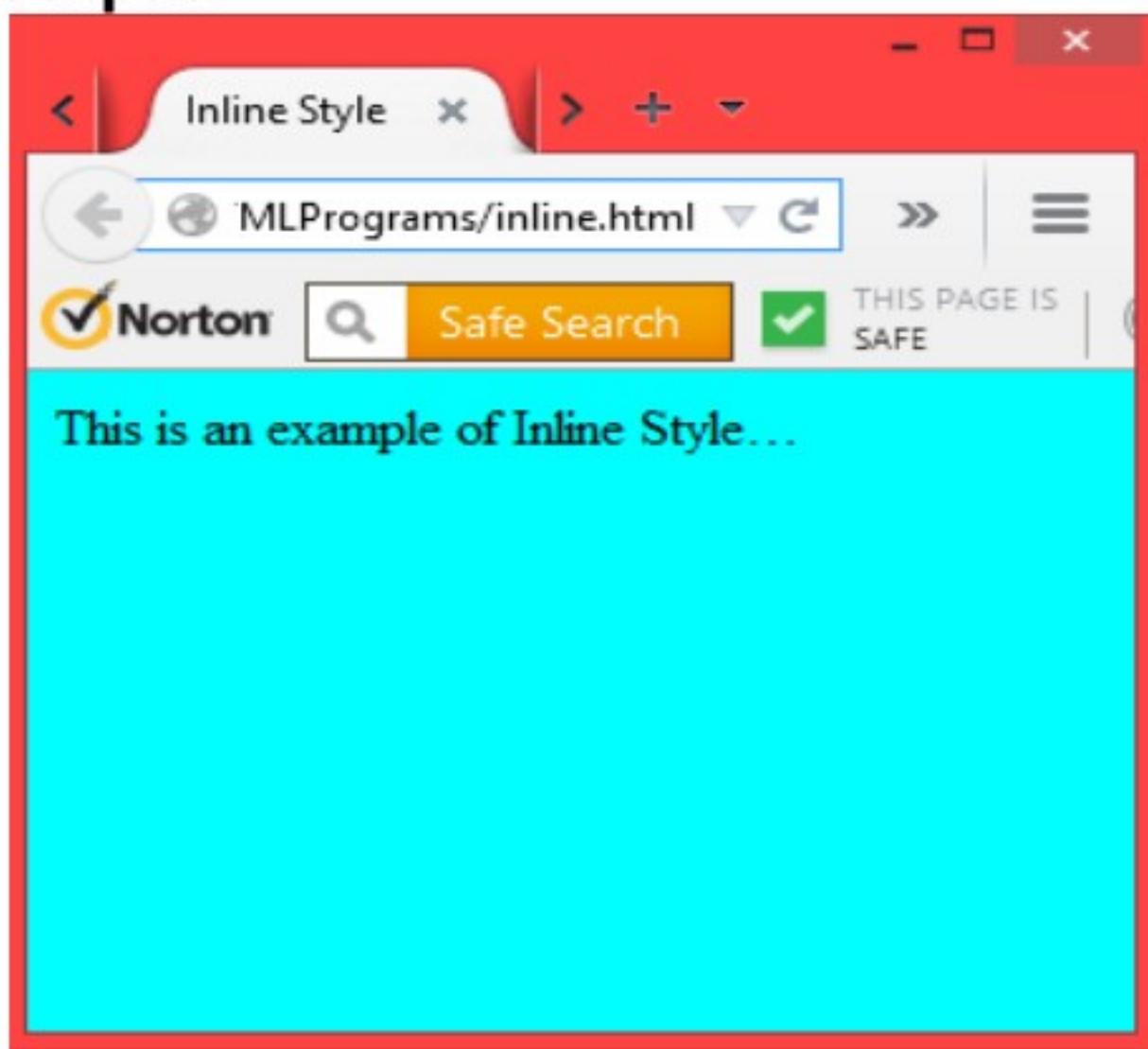
1) Inline Style

- Inline styles એ ડોક્યુમેન્ટમાં **directly** જ ટેગ માં લખાયેલી હોય છે. Inline styles ફક્ત તે ટેગ ને જ અસર કરે છે જેનો તેઓ ઉપયોગ કરે છે.
- Syntax:** <tag style="property1: value; property2: value;"> </tag>

Example: Using Inline Stylesheet

```
<html>
  <head>
    <title>Inline Style</title>
  </head>
  <body bgcolor="#ffffff">
    <p style="color:blue; font-family:arial;"> This is an example of Inline Style...
  </p>
  </body>
</html>
```

Output:



2) Embedded/Internal Style

- Internal/embedded CSS નો ઉપયોગ કરતી વખતે, તમારે નવો <style> ટેગ એ <head> ટેગ ની અંદર ઉમેરવામાં આવે છે.
- Embedded કરેલી styles ફક્ત તે જ પેજ પરના ટેગને અસર કરે છે જેમાં તેઓ embedded કરેલા છે.
- Syntax:**

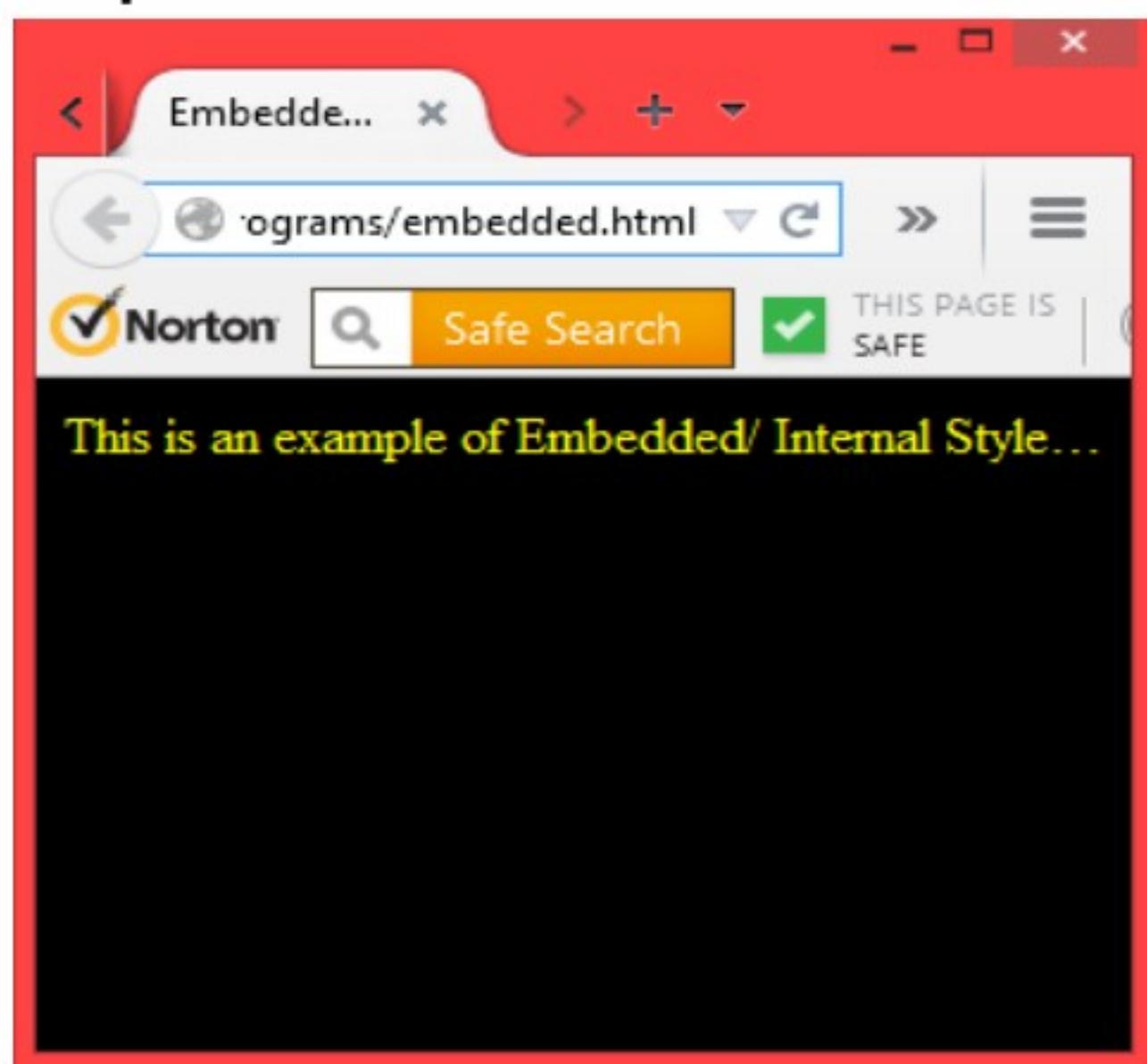
```
<style type="text/css">
  Selector {property: value;}
  Selector {property: value;}
```

.....
</style>

Example: Embedding Stylesheet

```
<html>
    <head>
        <title>Embedded Style</title>
        <style>
            p {color: yellow;}
            body {background-color: black;}
        </style>
    </head>
    <body>
        <p> This is an example of Embedded/ Internal Style... </p>
    </body>
</html>
```

Output:



3) External/Linked Style Sheet

- External styles એવી styles છે જે જુદા ડોક્યુમેન્ટ માં લખાયેલી હોય છે અને પછી વિવિધ વેબ ડોક્યુમેન્ટ સાથે **attached** હોય છે. External style sheets સાથે જોડાયેલા કોઈપણ ડોક્યુમેન્ટને તે અસર કરી શકે છે.
- External CSS ફાઇલ કે જે ફક્ત CSS કોડ ધરાવે છે અને ".css" **file extension** થી save કરવામાં આવે છે. આ ફાઇલનો ઉપયોગ HTML ડોક્યુમેન્ટ માં <style> ટેગ બદલે <link> ટેગ નો ઉપયોગ કરીને થાય છે.
- ઉદાહરણ તરીકે, નીચે આપેલ style.css ફાઇલ બનાવીએ.

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

- ```
p
{ color: blue; }

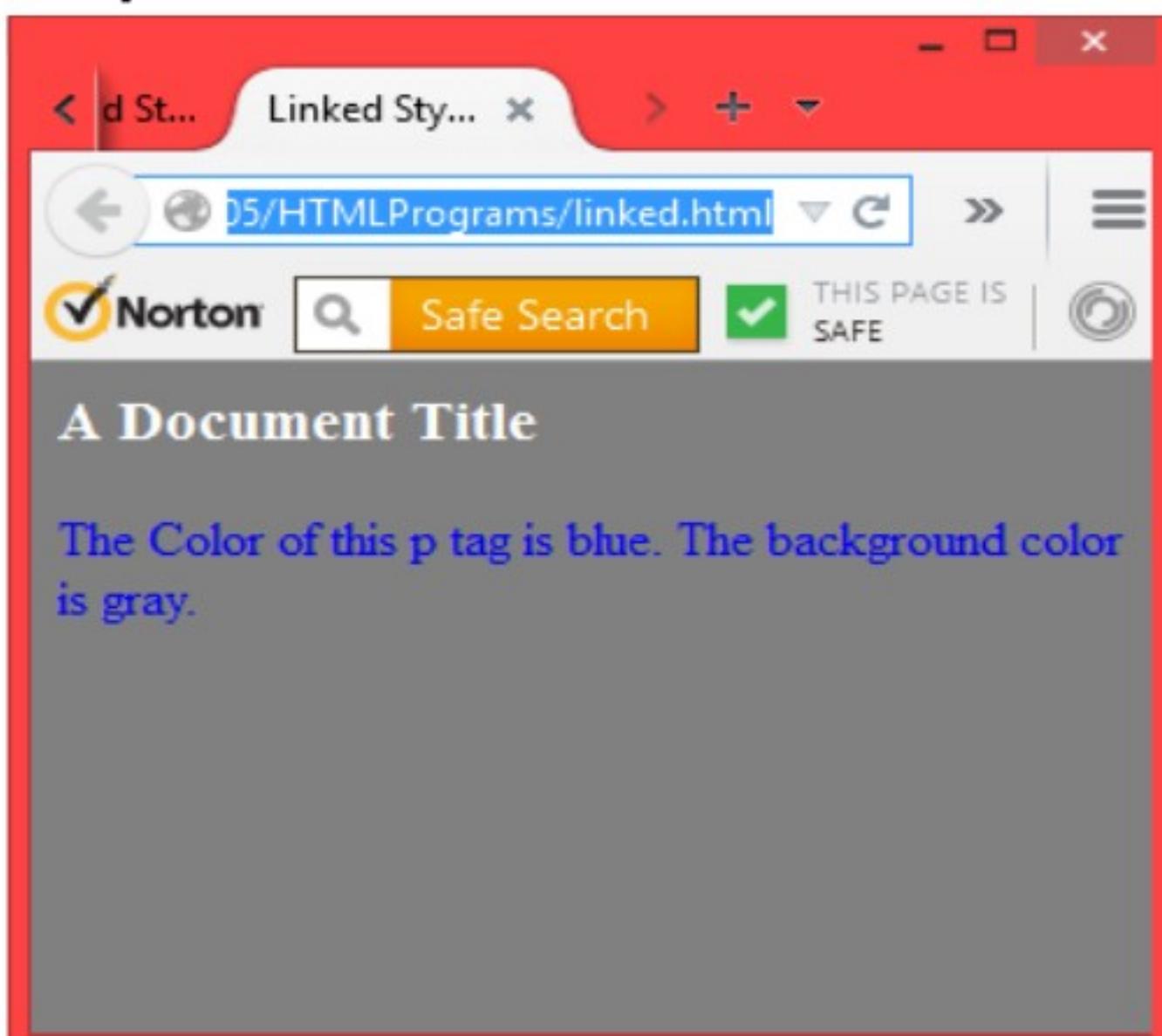
h3
{ color: white; }

• હવે નીચે એક નવું HTML વેબપેજ બનાવો અને style.css સાથે તેને લિંક કરો જેમ કે નીચે ઉદાહરણમાં બતાવ્યું છે.
```

### Example: Using Linked Stylesheet

```
<html>
<head>
 <title>Linked Style Sheets</title>
 <link rel="stylesheet" href="style.css" type="text/css"/>
</head>
<body>
 <h3> A Document Title </h3>
 <p> The Color of this p tag is blue. The background color is gray. </p>
</body>
</html>
```

### Output:



## Making use of ids and classes with Example or CSS Selector

- **ID Selector**
  - ચોકકસે element શોધવા માટે “**id**” **selector** એ HTML ટેગ ના id attribute નો ઉપયોગ કરે છે.
  - **ID એ unique** છે. દરેક element માં માત્ર એક જ ID હોઈ શકે છે અને દરેક પેજ માં ફક્ત એક જ element હોય શકે છે તે id સાથે.

- કોઈ ચોક્કસ id સાથે element શોધવા માટે, id ના નામ પહેલા **hash(#)** character લખો.
- **Class Selector**
  - “**class**” selector એ ચોક્કસ class સાથે element શોધે છે. class selector એ HTML class ના attribute નો ઉપયોગ કરે છે.
  - **Classes એ unique** નથી. એક જ �class નો મલિટપલ elements પર ઉપયોગ કરી શકીએ છીએ અને તે જ �element પર મલિટપલ class નો ઉપયોગ કરી શકીએ છીએ.
  - કોઈ ચોક્કસ class સાથે elements શોધવા માટે, class ના નામ પહેલા, **period(.)** character લખો.

**Example:**

```
<html>
<head>
<style>
#intro{
 color: yellow;
}
#intro1{
 color: hotpink;
}
.main{
 background-color: #013375;
 color: white;
 width:25%;
}
</style>
</head>
<body>
<div class="main">
 <p id="intro">Welcome to DWSL material webpage.</p>
 <p id="intro1">Learn the use of selector ID and Class.</p>
 <p>Thanking you.</p>
</div>
</body>
</html>
```

## Output:

Welcome to DWSL material webpage.

Learn the use of selector ID and Class.

Thanking you.

## The CSS Box Model

- બ્ધા HTML elements બોક્સ તરીકે માનવામાં આવે છે. CSS માં, "box model" શબ્દ નો ઉપયોગ કિાણ અને લેઆઉટ વિશે વાત કરતી વખતે થાય છે.
- CSS બોક્સ મોડેલ એ એવું બોક્સ છે જે HTML બોક્સ ની around wrap આપે છે, અને તેમાં margins, borders, padding, અને actual content શામેલ છે.
- બોક્સ મોડેલ elements ની આસપાસ બોર્ડર ઉમેરવા અને elements વચ્ચેની જગ્યા define કરવા દે છે.
  - Content** - બોક્સનો content, જ્યાં text અને images દેખાઈ છે.
  - Padding** - content ની આસપાસ નો એરિયા padding હોય છે.
  - Border** - બોર્ડર કે જે padding અને margin ની વચ્ચે હોય છે.
  - Margin** - બોર્ડરની બહાર નો એરિયા margin હોય છે.



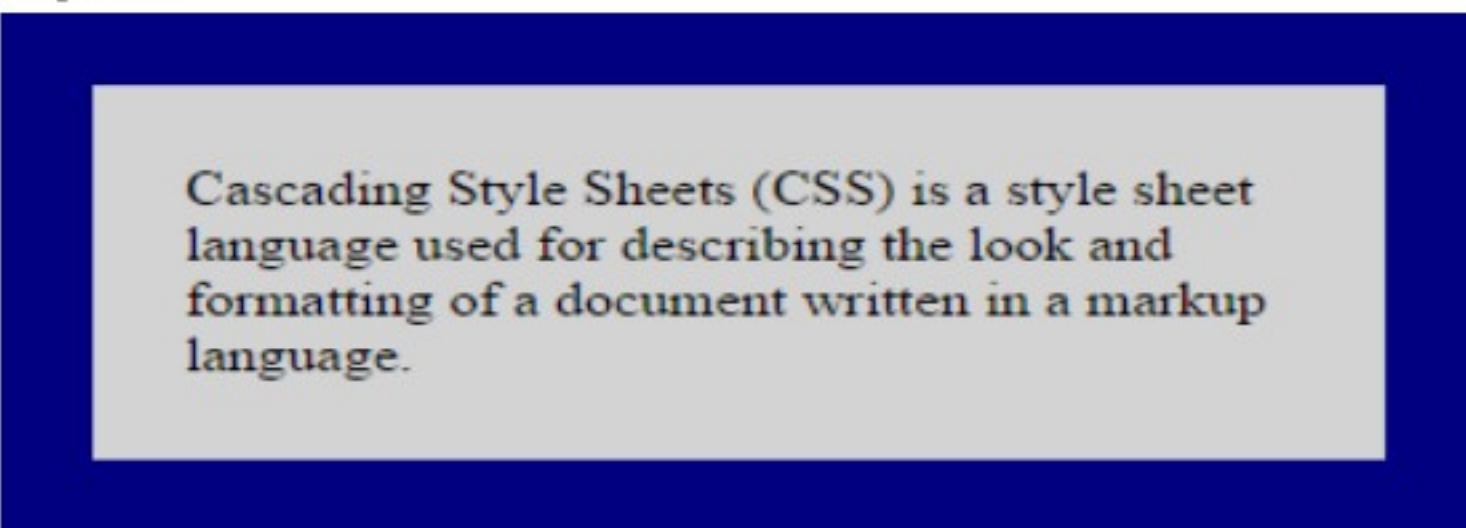
## Example: Using Linked Stylesheet

```
<html>
<head>
<style>
div
{
 background-color: lightgrey;
 width: 300px;
 padding: 25px;
 border: 25px solid navy;
 margin: 25px;
}
</style>
</head>
<body>
<div>
```

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language.

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

## Output:



## CSS Margin

- CSS margin પ્રોપર્ટીસ elements ની આસપાસની જગ્યાને define કરે છે. Margin એ element (બૉક્સની બહાર) ની આસપાસની જગ્યા clear કરે છે. Margin માં background color હોતો નથી, અને તે completely transparent છે.

Value	Description
auto	બ્રાઉઝર એ margin નું કેલક્યુલેશન કરે છે.
length	Margin ને px, pt, cm માં specify કરી શકાય છે. Default value 0px છે.
%	Containing element ની width ના percent માં margin ને specify કરે છે.
inherit	Specify કરે છે કે margin એ inherit થી parent હોવું જોઈએ.

**Example:** paragraph ટેગ માટે margin apply કરી રહ્યા છીએ.

```
p {
 margin-top: 100px;
 margin-bottom: 100px;
 margin-right: 150px;
 margin-left: 150px;
}
```

- Margin પ્રોપર્ટી ની એક થી ચાર વેલ્યુ હોય શકે છે.
  - margin: 25px 50px 75px 100px;** // **top** margin 25px, **right** margin 50px, **bottom** margin 75px અને **left** margin 100px
  - margin: 25px 50px 75px;** // **top** margin 25px, **right** અને **left** margins 50px અને **bottom** margin 75px
  - margin: 25px 50px;** // **top** અને **bottom** margins 25px અને **right** અને **left** margins 50px
  - margin: 25px;** // બધી બાજુ margins 25px

## CSS Padding

- CSS padding प्रोपर्टीस એ element બોર્ડર અને element content વચ્ચેની space ને define કરે છે. Padding એ element ના content (સીમાની અંદર) ની આસપાસનો area clear કરે છે. Padding એ element ના background color ને અસર થાય છે.

Value	Description
length	Fix paddingને define કરે છે (pixels, pt, em, વગેરે.)
%	Containing element % માં padding define કરે છે.

**Example:** paragraph ટેગ માટે padding apply કરી રહ્યા છીએ.

```
p {
 padding-top: 100px;
 padding-bottom: 100px;
 padding-right: 150px;
 padding-left: 150px;
}
```

- Padding property ના એક થી ચાર વેલ્યુ હોય શકે છે.
  - padding: 25px 50px 75px 100px;** //top padding 25px, right padding 50px, bottom padding 75px અને left padding 100px
  - padding: 25px 50px 75px;** //top padding 25px, right અને left paddings 50px , bottom padding 75px
  - padding: 25px 50px;** //top અને bottom paddings 25px , right અને left paddings 50px
  - padding: 25px;** // બધી બાજુ paddings 25px

## CSS Text

### All CSS Text Properties

Property	Description	Example
color	તેનો ઉપયોગ ટેક્સ્ટ નો કલર સેટ કરવા માટે થાય છે. CSS સાથે, કલર દ્વારા specify થાય છે: a <b>HEX</b> વેલ્યુ જેમ કે "#ff00ff", a <b>RGB</b> વેલ્યુ જેમ કે "rgb(255,0,0)", a <b>Color name</b> જેમ કે "black"	body { color: black; }  h1 { color: #ff00ff; }
text-align	તેનો ઉપયોગ ટેક્સ્ટ ના horizontal alignment ને specify કરવા માટે થાય છે. Text ને <b>centered</b> કરી શકાય છે, અથવા <b>left or right aligned</b> કરી શકાય છે અથવા <b>justified</b> કરી શકાય છે.	p.main { text-align: justify; }  h1 { text-align: right; }

text-decoration	<p>તેનો ઉપયોગ ટેક્સ્ટ માંથી decoration ને સેટ અથવા દૂર કરવા માટે થાય છે. તે મોટાભાગે design purposes માટે લિંક નીચે underlines દૂર કરવા માટે વપરાય છે. આ પ્રોપર્ટી નીચેની વેલ્યુ ધરાવે છે:</p> <ul style="list-style-type: none"> <li>(1) none: એક normal ટેક્સ્ટ ને define કરે છે. આ દ્વારા default છે.</li> <li>(2) underline: ટેક્સ્ટની નીચે એક line define કરે છે.</li> <li>(3) overline: ટેક્સ્ટની ઉપરની line define કરે છે.</li> <li>(4) line-through: ટેક્સ્ટ line through ને define કરે છે.</li> </ul>	<pre>a { text-decoration: none; }  h1 { text-decoration: underline; }</pre>
text-transform	<p>તેનો ઉપયોગ ટેક્સ્ટ ના capitalization ને કંટ્રોલ કરવા માટે થાય છે. આ પ્રોપર્ટી ટેક્સ્ટ માં uppercase અને lowercase ટેક્સ્ટ નો ઉલ્લેખ કરે છે. આ પ્રોપર્ટી નીચેની વેલ્યુ ધરાવે છે:</p> <ul style="list-style-type: none"> <li>(1) none : કોઈ capitalization નહીં. Text renders કરે છે.</li> <li>(2) capitalize: દરેક word ના first character ને uppercase માં transforms કરે છે.</li> <li>(3) uppercase: બધા character ને uppercase માં transforms કરે છે.</li> <li>(4) lowercase: બધા character ને lowercase માં transforms કરે છે.</li> </ul>	<pre>p.uppercase { text-transform: uppercase; }  p.lowercase { text-transform: lowercase; }  p.capitalize { text-transform: capitalize; }</pre>
text-indent	તે text-block માં પહેલી લાઇન ના indentation ને specify કરવા માટે ઉપયોગમાં લેવાય છે.	<pre>p { text-indent: 50px; }</pre>
text-shadow	<p>આ પ્રોપર્ટી ટેક્સ્ટ પર shadow ઉમેરે છે. આ પ્રોપર્ટી નીચેની વેલ્યુસ ધરાવે છે:</p> <ul style="list-style-type: none"> <li>(1) h-shadow(Required): horizontal shadow ની position.</li> <li>(2) v-shadow(Required): vertical shadow ની position.</li> </ul>	<pre>h1 { text-shadow: 2px 2px 8px #FF0000; }</pre>

	(3) blur-radius(Optional): blur radius. Default વેલ્યુ 0 છે. (4) color(Optional): shadow ની color. (5) none: Default વેલ્યુ. Shadow નહોં	
letter-spacing	આ પ્રોપ્ટી ટેક્સ્ટ ના characters વચ્ચેની જગ્યા ને વધારે છે અથવા ઘટાડે કરે છે.	h1 { letter-spacing: 2px; } h2 { letter-spacing: -3px; }
line-height	આ પ્રોપ્ટી એ line ની height ને specify કરે છે.	p.small { line-height: 70%; }

## CSS Font

### All CSS Font Properties

Property	Description	Example
font	એક declaration માં બધી font પ્રોપ્ટીસ ને સેટ કરે છે.	p { font: italic bold 12px/30px Georgia, serif; }
font-family	ટેક્સ્ટ માટે font family ને specify કરે છે. Font ના બે ટાઇપ ના font ના name છે: (1) <b>family-name</b> જેમ કે "times", "courier", "arial" વગેરે. (2) <b>generic-family</b> જેવા કે "serif", "sans-serif", વગેરે.	p.serif { font-family: "Times New Roman", Times, serif; }
font-size	ટેક્સ્ટ ના font size નો specify કરે છે. font-size માટેના પ્રોપ્ટી વેલ્યુ <b>medium</b> , <b>xx-small</b> , <b>x-small</b> , <b>small</b> , <b>large</b> , <b>x-large</b> , <b>xx-large</b> , <b>smaller</b> , <b>larger</b> , <b>length</b> , <b>initial</b> અને <b>inherit</b> છે.	h1 { font-size:150%; } div { font-size:larger; }
font-style	ટેક્સ્ટ માટે font ની style specify કરે છે. આ પ્રોપ્ટી પાસે ત્રણ વેલ્યુસ છે: (1) <b>normal</b> : આ સામાન્ય ટેક્સ્ટ બતાવે છે. (2) <b>italic</b> : આ ટેક્સ્ટ italics બતાવે છે. (3) <b>oblique</b> : આ ટેક્સ્ટ italic ની જેમ જ છે પરંતુ ઓછો સપોર્ટ થાય છે.	p.normal { font-style: normal; }

font-variant	આ ટેક્સ્ટ માં small caps font display થાય છે. તેની પાસે ચાર પ્રોપર્ટી વેલ્યુ છે. (1) <b>normal</b> : સામાન્ય font બતાવે છે. (2) <b>small-caps</b> : small-caps font બતાવે છે. (3) <b>initial</b> : આ પ્રોપર્ટી default વેલ્યુ ને સેટ કરે છે. (4) <b>inherit</b> : Parent element માથી આ પ્રોપર્ટી inherits થાય છે.	p.normal { font-variant: normal; }  p.small { font-variant: small-caps; }
font-weight	આ પ્રોપર્ટી સેટ કરે છે કે ટેક્સ્ટના characters ને કેટલા thick અથવા thin બતાવા છે. તેની પાસે નીચેની પ્રોપર્ટી વેલ્યુ છે: <b>normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, initial</b> અને <b>inherit</b> .	p.normal { font-weight: normal; }  p.thick { font-weight: bold; }

## CSS Link

- કોઈપણ CSS પ્રોપર્ટી (જેમ કે color, font-family, background, વગેરે) સાથે links style નો ઉપયોગ કરી શકાય છે. આ ઉપરાંત, તેઓ ક્યા સ્ટેટ માં છે તેના આધારે links નો અલગ રીતે ઉપયોગ કરી શકાય છે.
- તેમાં ચાર links છે:
  - a:link**: unvisited link.
  - a:visited**: યુઝરએ visit લીધેલ link.
  - a:hover**: જ્યારે યુઝર તેના પર માઉસ લે છે.
  - a:active**: તે ક્લિક થાય તે moment એ link થાય છે.

### Example:

/* unvisited link */ a:link { color: #FF0000; }	/* visited link */ a:visited { color: #00FF00; }	/* mouse over link */ a:hover { color: #FF00FF; }	/* selected link */ a:active { color: #0000FF; }
-------------------------------------------------------------	--------------------------------------------------------------	---------------------------------------------------------------	--------------------------------------------------------------

## CSS Background

Property	Description	Example
background	એક declaration માં બધી background પ્રોપર્ટીસ સેટ કરે છે.	background: #00ff00 url("smiley.gif")
background-attachment	Background image ને ફિક્સ કરે છે બાકીના પેજ સાથે scrolls કરે છે.	background-attachment: fixed;
background-color	Element ની background color ને સેટ કરે છે.	background-color: yellow;
background-image	Element ની background image ને સેટ કરે છે.	background-image: url("paper.gif");
background-position	Background image ની starting position ને સેટ કરે છે.	background-position: center;
background-repeat	Background image ને કેવી રીતે repeat કરવામાં આવશે તે સેટ કરે છે.	background-repeat: no-repeat;

## CSS Border Properties

### Border Style

- Border-style પ્રોપર્ટી એ specify કરે છે કે બોર્ડર કઈ styleની display કરવી. Border – style વેલ્યુ: none: Defines no border

dotted: Dotted બોર્ડર define કરે છે.

dashed: Dashed બોર્ડર define કરે છે.

solid: Solid બોર્ડર define કરે છે.

double: બે બોર્ડર define કરે છે. બે બોર્ડર ની width એ border-width વેલ્યુ ની જેમ જ હોય છે.

groove: 3D grooved બોર્ડર define કરે છે. Effect એ border-color વેલ્યુ પર depend છે.

ridge: 3D ridged બોર્ડર define કરે છે. Effect એ border-color વેલ્યુ પર depend છે.

outset: 3D outset બોર્ડર define કરે છે. Effect એ border-color વેલ્યુ પર depend છે.

Inset: 3D inset border ने define करे छે. Effect એ border-color વેલ્યુ પર depend છે.

## Border Width

- Border-width પ્રોપર્ટી નો ઉપયોગ બોર્ડરની width ને સેટ કરવા માટે થાય છે.
- Width pixelsમાં સેટ છે, અથવા ત્રણ pre-defined વેલ્યુ માંથી એકનો ઉપયોગ કરી શકાય છે: thin, medium અથવા thick.

**Example:** border-width: medium;

## Border Color

- Border-color પ્રોપર્ટી નો ઉપયોગ બોર્ડર નો color સેટ કરવા માટે થાય છે. Color દ્વારા સેટ કરી શકાય છે.
  - name - color નું name specify કરે, જેમ કે "red"
  - RGB - RGB વેલ્યુ ને specify કરે જેમ કે "rgb(255,0,0)"
  - Hex - hex વેલ્યુ ને specify કરે જેમ કે "#ff0000"

**Example:** border-color: red;

## The Position Property

- Position પ્રોપર્ટી element માટે ઉપયોગમાં લેવાતી position મેથડને specify કરે છે.
- ચાર જુદી જુદી પ્રોપર્ટી values છે: static, relative, fixed, absolute.
- Elements પછી top, bottom, left, અને right પ્રોપર્ટીસ ઉપયોગ કરીને position લે છે. જોકે, આ પ્રોપર્ટીસ વર્ક કરશે નહીં જ્યાં સુધી position પ્રોપર્ટી સેટ થાશે નહીં ત્યાં સુધી.

### Position: static

- HTML elements default રૂપે static position લે છે.
- Static position એ elementsની top, bottom, left અને right પ્રોપર્ટીસ થી અસર થતી નથી.
- Static position એ કોઈ ખાસ position નથી; તે પેજ ના normal flow માં જ છે.

### Position: fixed

- એક element ને position fixed આપવામાં આવે છે. જેનો અર્થ એ છે કે તે પેજને scroll કરવામાં આવે તો પણ તે જ જગ્યા પર રહે છે. Element ને position આપવા માટે top, right, bottom, અને left પ્રોપર્ટીસ નો ઉપયોગ કરવામાં આવે છે.
- કોઈ fixed element પેજમાં કોઈ જગ્યા છોડતું નથી જ્યાં તે સામાન્ય રીતે located થયેલ હોય છે.

**Example:**

```
<html>
<head>
 <title>Title of your web page</title>
 <style>
 div.static
 {
```

```

 position: static;
 border: 3px solid #8AC007;
 }
 div.fixed
 {
 position: fixed;
 bottom: 0;
 right: 0;
 width: 300px;
 border: 3px solid #8AC007;
 }
</style>
</head>
<body>
 <div class="static"> This div element has position: static; </div>
 <div class="fixed"> This div element has position: fixed; </div>
</body>
</html>

```

### Output:

This div element has position: static;

This div element has position: fixed;

### Position: relative

- એક element ની position relative છે જે normal position છે.
- Relatively-position એ આપેલ element ની top, right, bottom, અને left બાજુની પ્રોપર્ટીસ સેટ કરવાથી તે તેની normal position થી દૂર કરવામાં આવશે. Element દ્વારા બાકી કોઈપણ જગ્યામાં fit થવા માટે અન્ય content ને adjusted કરવામાં આવશે નહીં.

## Position: absolute

- Absolute પ્રોપર્ટીસ એ વેબપેજ પર બે દિવ ને overlap કરવા માટે વક્ત કરી શકે છે. Absolute position એ પહેલા parent element ની relative માં static હોય છે જેમાં static સિવાયની position હોય છે.
- Absolutely position એ elements ને normal flow માંથી દૂર કરવામાં આવે છે. Absolutely position element એ બીજા element ને overlap કરી શકે છે.

### Example:

```
<html>
<head>
 <style>
 div.relative
 {
 position: relative;
 width: 400px;
 height: 200px;
 border: 3px solid #8AC007;
 }
 div.absolute
 {
 position: absolute;
 top: 80px;
 right: 0;
 width: 200px;
 height: 100px;
 border: 3px solid #8AC007;
 }
 </style>
</head>
<body>
 <div class="relative">This element has position: relative;
 <div class="absolute">This element has position: absolute;</div>
 </div>
</body>
</html>
```

## Output:

This div element has position: relative;

This div element has position: absolute;

## Overlapping Elements

- જ્યારે elements ગોઠવાય છે, ત્યારે તે બીજા elements ને overlap કરી શકે છે.
- Z-index પ્રોપર્ટી element ના stack order ને specify કરે છે (કે જે element ને બીજા ની આગળ, અથવા પાછળ મુકવામાં આવે છે).
- Element માં positive અથવા negative stack order હોઈ શકે છે.

## Example:

```
<html>
<head>
 <style>
 img {
 position: absolute; left: 0px; top: 0px; z-index: -1;
 }
 </style>
</head>
<body>

 <p>Because the image has a z-index of -1, it will be placed behind the text. </p>
</body>
</html>
```

## Output:



Because the image has a z-index of -1,  
it will be placed behind the text.

## Float Properties

- Float property એ specify કરે છે કે કોઈ બોક્સ (અથવા element) ને float કરવું કે નહીં.
- Syntax:** float: none|left|right|initial|inherit;

### Example:

```
<html>
<head>
 <style>
 img {
 float: left;
 }
 </style>
</head>
<body>
 <p>

 Cascading Style Sheets (CSS) is a style sheet language used for describing the look and for
 mattig of a document written in a markup language.
 </p>
</body>
</html>
```

### Output:



Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language.

## Difference between CSS2 and CSS3

- CSS3 મોડ્યુલ નામના ધણા જુદા જુદા ડોક્યુમેન્ટ્સ માં વહેંચાયેલું છે. દરેક મોડ્યુલ નવી capability ઉમેરે છે અથવા CSS2 માં defined features ને extends કરે છે. Original CSS2 ની recommendation publication સમયે CSS3 પર વર્ક ચાલુ થયું.
- CSS3 માં મોડ્યુલાઇઝનને કારણે, દરેક મોડ્યુલોમાં વિવિધ stability હોય છે અને તે અલગ status માં હોય છે.
- CSS3 version CSS2 કરતા ધણા બધા બ્રાઉઝર્સને સપોર્ટ કરે છે.
- CSS3 માં બીજા features ઉમેરાયેલા જેમ કે નવા combinator, નવા CSS selectors, નવા pseudo-elements અને નવી style properties.

## Opacity (Transparency)

- Opacity પ્રોપર્ટી element માટે opacity લેવલ સેટ કરે છે. Transparency માટે CSS3 પ્રોપર્ટી opacity છે.
- IE9, Firefox, Chrome, Opera અને Safari transparency માટે પ્રોપર્ટી ની opacityનો ઉપયોગ કરે છે. Opacity પ્રોપર્ટી 0.0 - 1.0 વેલ્ચુ ને લઈ શકે છે. Lower વેલ્ચુ એ element ને વધુ પારદર્શક બનાવે છે.

- **Transparency-level** એ **opacity-level** ને બતાવે છે, જ્યાં 1 એ transparent નથી, 0.5 એ 50% દેખાઈ છે, અને 0 completely transparent છે.
- **Syntax:** opacity: number

**Example:**

```
<html>
<head>
<style>
 img { opacity: 0.4; }
</style>
</head>
<body>

</body>
</html>
```

**Output:**

**Original Image:**



**Transparent Image:**



## Box-shadow (Shadows)

- Box-shadow પ્રોપર્ટી ડિઝાઇનરને color, size, blur અને offset માટેની વેચ્યુ ને specify કરીને, box elements, text, etc પર multiple drop shadows (outer અથવા inner) ને સરળતાથી implement માં મૂકવાની મંજૂરી આપે છે. Box-shadowની પ્રોપર્ટી shadows એ comma-separated નું list છે, દરેક 2-4 length values, optional color અને optional inset keyword specify થાય છે. Omitted lengths 0 છે.
- **Syntax:** box-shadow: none|h-shadow v-shadow blur spread color |inset|initial|inherit;

**Example:**

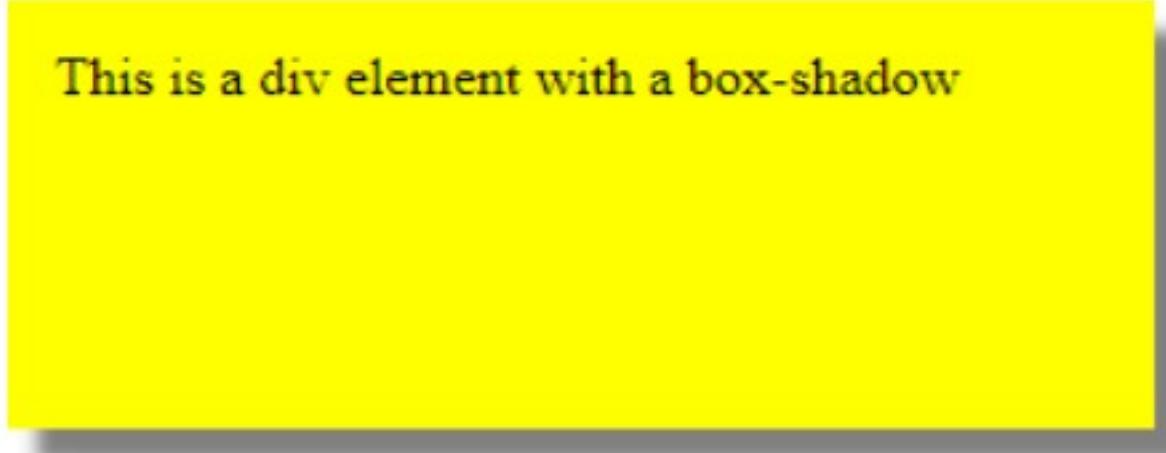
```
<html>
<head>
<style>
 div
 {
 width: 300px;
 height: 100px;
```

```

padding: 15px;
background-color: yellow;
box-shadow: 10px 10px 5px grey;
}
</style>
<body>
<div>This is a div element with a box-shadow</div>
</body>
</html>

```

**Output:**



This is a div element with a box-shadow

## Border-radius (Rounded Corners)

- Border-radius પ્રોપર્ટી ચાર બોર્ડર - \* - \* - radius પ્રોપર્ટી ને સેટ કરવા માટે એક shorthand property છે. આ પ્રોપર્ટી તમને elements માં rounded બોર્ડરસ ઉમેરવા દે છે.
- Syntax:** border-radius: 1-4 length|% / 1-4 length|%|initial|inherit;

**Example:**

```

<html>
<head>
<style>
#BorderRadiusDemo {
 border: 2px solid #000000;
 padding: 10px 40px;
 background: #ababab;
 width: 300px;
 border-radius: 10px;
}
</style>
</head>
<body>
<div id="BorderRadiusDemo"> This is the demo example of the css property border- radius.
</div>
</body>
</html>

```

## Output:

This is the demo example of the css property border-radius.

## Gradients

- **Gradient એ colors ની sequence છે.** Simple gradients એ foreground થી background color તરફનો flow છે, પરંતુ gradients માં ધણા બધા colors શામેલ હોઈ શકે છે. CSS3 gradients તમને બે અથવા વધુ specified colors વચ્ચે smooth transitions display કરવા દે છે.
- CSS3 gradientsનો ઉપયોગ કરીને તમે ડાઉનલોડ ટાઇમ અને bandwidth નો ઉપયોગ ઓછો કરી શકો છો. વધારામાં, element સાથે gradient નો ઉપયોગ કરવાથી જ્યારે તમે zoom કરો છો ત્યારે વધારે સારું દેખાઈ છે, કારણ કે **browser** દ્વારા gradient ને જનરેટ કરવામાં આવે છે.
- **CSS3 બે ટાઇપ ના gradient ને define કરે છે:** (1) **Linear Gradient** અને (2) **Radial Gradient**.

### 1) Linear Gradients

- Linear Gradient જે સીધી લાઇન માં એક color થી બીજી color માં flow જાય છે.
- Linear gradient બનાવવા માટે તમારે ઓછામાં ઓછા બે color stops define કરવા આવશ્યક છે. Color stops તે color છે જે તમે smooth transitions કરવા માંગો છો. તમે gradient effect સાથે starting point અને direction (અથવા an angle) પણ સેટ કરી શકો છો.
- **Syntax:** background: linear-gradient(direction, color-stop1, color-stop2, ...);

#### Example: Linear Gradients

```
<html>
<head>
<style>
div
{
 height: 200px;
 width: 200px;
 background: linear-gradient(to right, red, blue);
}
</style>
</head>
<body>
 <div id="ligrad">Linear Gradient - Left to Right</div>
</body>
</html>
```

## Output:



## 2) Radial Gradients

- Radial Gradient જ્યાં એક color થોકસ point એ concentrate હોય છે, અને બીજા color તે point થી આગળ દેખાય છે. By default, shape ellipse છે, size સૌથી દૂર-ખૂણા (farthest-corner) છે, અને position center છે. Radial gradient બનાવવા માટે તમારે ઓછામાં ઓછા બે color stops પણ define કરવા આવશ્યક છે.
- **Syntax:** background: radial-gradient(shape size at position, start-color, ..., last-color);

**Example:** Radial Gradients

```
<html>
<head>
 <style>
 div
 {
 height: 150px;
 width: 200px;
 background: radial-gradient(circle, pink, yellow, green);
 }
 </style>
</head>
<body>
 <div id="Radgrad">Radial Gradient - Circle Shape</div>
</body>
</html>
```

## Output:



## Transition Animation

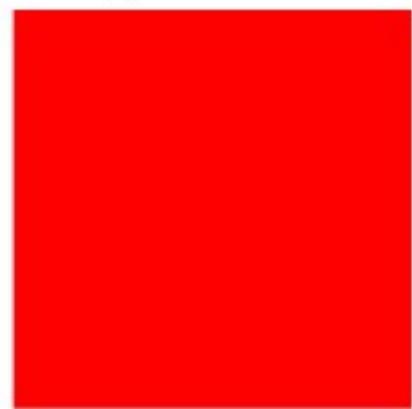
- **CSS3 transitions** તમને આપેલ સમયમાં, પ્રોપર્ટી વેલ્યુસ ને સરળતાથી (એક વેલ્યુથી બીજામાં) બદલી શકાય છે.
- **Transition attribute** ધણા �parameters લે છે:
  - **transition-property:** આ ટેગ દ્વારા animation નો ટાઇપ define થાય છે. Default વેલ્યુ બધી છે, પરંતુ અન્ય types, color, length, width, percentage, opacity અને number સહિત વર્ક કરવા માટે expected છે.
  - **transition-duration:** એનિમેશનની લંબાઈ define કરે છે.
  - **transition-timing-function:** જો તમે એનિમેશન ને constant speed એ આપવા માંગતા હો, તો linear નો ઉપયોગ કરો.
  - **transition-delay:** જો તમે બીજુ ટાઇમ વેલ્યુ શામેલ કરો છો, તો તે વેલ્યુ જેટલું delay કરવામાં આવશે. Delay પછી animation શરૂ થશે નહીં.

### Example:

```
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background: red;
 transition-property: width;
 transition-duration: 2s;
 transition-delay: 2s;
 transition-timing-function: linear
}
div:hover {
 width: 300px;
}
</style>
</head>
<body>
 <div></div>
</body>
</html>
```

## Output:

### Original Box



### After Hover Effect



## Transformations

- **Transformation** એ એક effect છે જે element ને shape, size અને position ને બદલી આપે છે.
- **Transform property** element પર 2D અથવા 3D transformation લાગુ કરે છે. આ પ્રોપર્ટીસ તમને rotate, scale, move, skewX, skewY અને skew, વગેરે, element માટે allow કરે છે.
- જ્યારે તમે transform ને element માં apply કરો છો, ત્યારે તમારે transformation ના ટાઇપને describe કરવા માટે નીચેનામાંના એક અથવા વધુ parameter ને લાગુ કરવાની જરૂર છે:
  - **Translate:** translate() મેથડએ elementને તેની current position (X-axis અને Y-axis માટેના parameter અનુસાર) થી ખસેડે છે.
  - **rotate:** rotate() મેથડ આપેલ degree અનુસાર element ને clockwise અથવા counter-clockwise ફેરવે છે. Negative વેલ્યુ નો ઉપયોગ કરીને element ને counter-clockwise ફેરવશે.
  - **scale:** scale() મેથડ elementની size ને વધારે છે અથવા ધરાડે કરે છે (width અને height માટેના parameters અનુસાર).
  - **skew:** skew() મેથડ આપેલ angles દ્વારા X અને Y-axis સાથે element ને skews કરે છે.
  - **skewX:** skewX() મેથડ આપેલ angles દ્વારા X -axis સાથે element ને skews કરે છે.
  - **skewY:** skewY() મેથડ આપેલ angles દ્વારા Y -axis સાથે element ને skews કરે છે.
  - **matrix:** matrix() મેથડ માં બધી 2D transform મેથડ એક સાથે જોડાય છે. Matrix() મેથડ એ છ parameters લે છે, જેમાં મેથેમેટીકસ ફંક્શન રામેલ છે, જે rotate, scale, move (translate) અને skew elements allow કરે છે.
- **Syntax :** transform: none|transform-functions|initial|inherit;

### Example:

### In CSS:

```
#box1, #box2, #box3, #box4, #box5, #box6{
 width: 75px;
 height: 40px;
```

```

border: 3px solid black;
background-color: yellow; }

#box2 {
 transform: translate(100px,15px); }

#box3 {
 transform: rotate(80deg); }

#box4 {
 transform: scale(2) translate(100px,15px); }

#box5 {
 transform: skew(20deg,10deg);}

#box6 {
 transform: matrix(1, -0.3, 0, 1, 0, 0); }

```

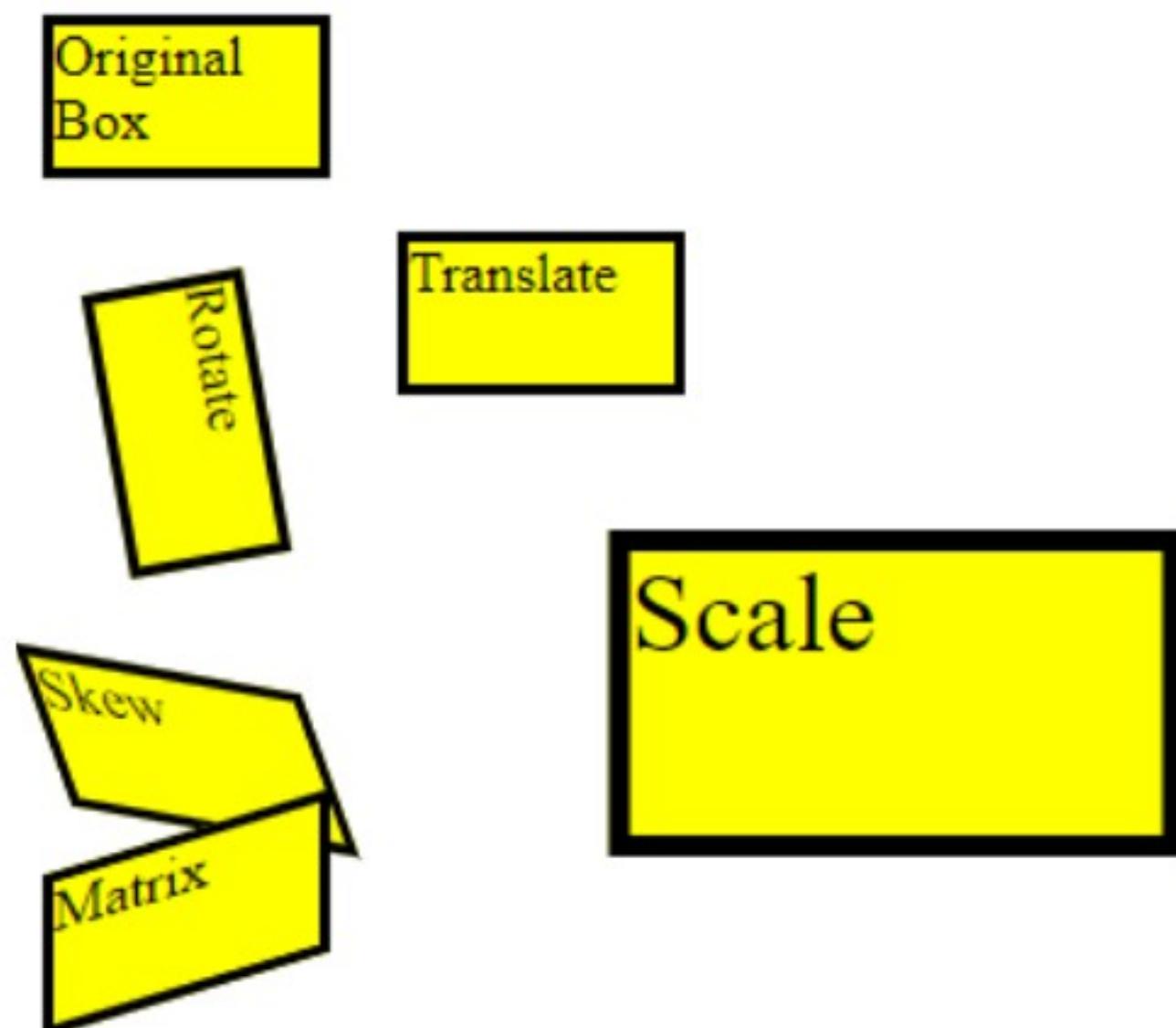
### In Body Tag:

```

<div id="box1">Original Box</div>
<div id="box2">Translate</div>
<div id="box3">Rotate</div>
<div id="box4">Scale</div>
<div id="box5">Skew</div>
<div id="box6">Matrix</div>

```

### Output:



### Animations

- એનિમેશન element ને ધીમે ધીમે એક style થી બીજું style માં બદલવાની મંજૂરી આપે છે.
- તમે ઇચ્�ો તેટલી CSS પ્રોપર્ટીસ તમે બદલી શકો છો.

- CSS3 એનિમેશનનો ઉપયોગ કરવા માટે, તમારે એનિમેશન માટે પહેલા કેટલાક keyframes ને specify કરવા આવશ્યક છે.
- Keyframes અમુક સમયે ચોક્કસ element ધરાવતી style ધરાવે છે.

## Key Frames

- **@keyframes** rule એ એનિમેશન કોડ ને specify કરે છે. એનિમેશનએ CSS styles ને એક સેટ થી બીજામાં ધીમે ધીમે change કરીને create કરવામાં આવે છે.
- એનિમેશન દરમિયાન, તમે ધણી વાર CSS styles નો સેટ બદલી શકો છો.
- જ્યારે style માં change percent માં થશે, અથવા "from" અને "to" keywords સાથે specify થશે, જે 0% અને 100% જેટલું જ છે. **0%** એ એનિમેશન ની શાસ્ત્રાત છે, એનિમેશન એ **100%** પૂર્ણ થાય છે
- સારા બ્રાઉઝર સપોર્ટ માટે, તમારે હંમેશાં 0% અને 100% selectors બંને define કરવા જોઈએ.

**Syntax:** @keyframes animationname {

```
 keyframes-selector {
 css-styles;
 }
```

```
}
```

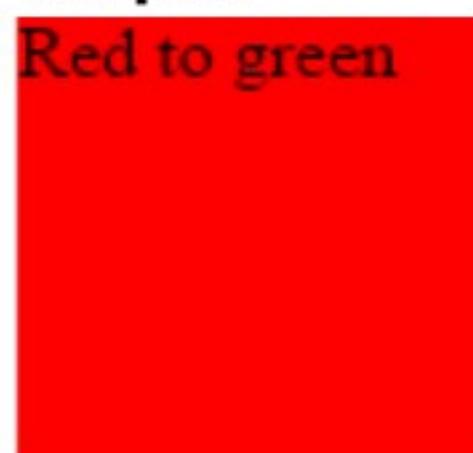
### Example:

```
<html>
<head>
<style>
div
{
 width: 100px;
 height: 100px;
 background-color: red;
 animation-name: example;
 animation-duration: 4s;
}
@keyframes example
{
 from
 {
 background-color: red;
 }
 to
 {

```

```
background-color: green;
}
}
</style>
</head>
<body>
 <div>Red to green</div>
</body>
</html>
```

**Output:**



### Introduction to PHP

- PHP એ server scripting language છે અને dynamic અને interactive વેબપેજ બનાવવા માટેનું એક પાવરફૂલ ટૂલ છે.
- PHP એ "**PHP: Hypertext Preprocessor**" માટે વપરાય છે.
- PHP એ વધારે ઉપયોગ માં લેવાતી, open source scripting language છે.
- PHP એ scripts સર્વર પર execute કરવામાં આવે છે.
- PHP એ ડાઉનલોડ અને ઉપયોગ માટે ફી છે.

### A Brief History of PHP

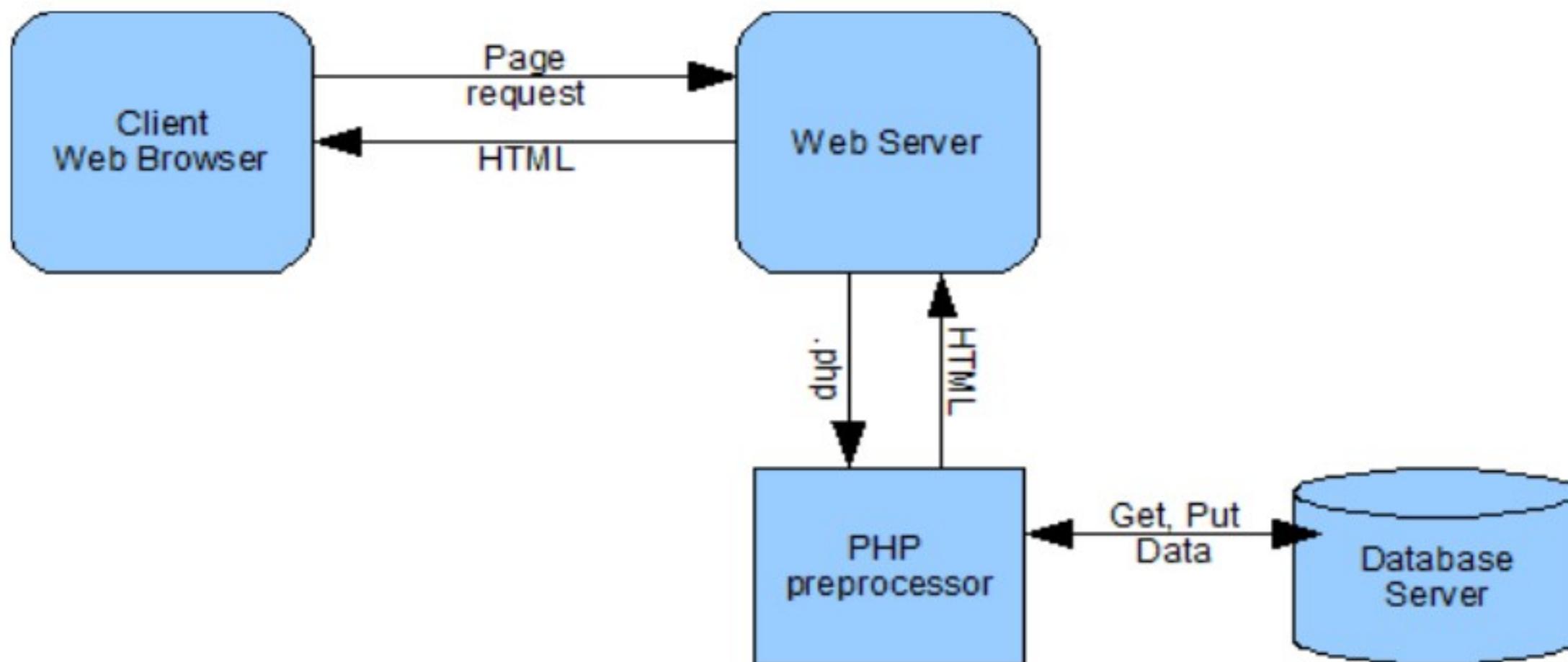
- PHP, મૂળ્યાંપે 1994 માં રેમસસ લોર્ડિઝ દ્વારા બનાવવામાં આવ્યું હતું. પ્રોગ્રામર રેમસ લોર્ડિઝ શસ્ત્રાત્માં C scripts નો સેટ બનાવ્યો હતો જેને તેણે "Personal Home Page Tools" તરીકે ઓળખાવ્યો હતો, જેથી તે પોતાના પર્સનલ હોમપેજ ને maintain કરી શકે. Scripts એ તેના resume ને display કરવા અને તેના વેબપેજ ના ટ્રાફિકને રેકૉર્ડ કરવા જેવા કાર્યો કર્યા હતા.
- આને Form Interpreter (PHP/ FI) તરીકે ઓળખાતા પેકેજને સમાવવા માટે extend કરવામાં આવી હતી. જ્યારે PHP, મૂળ્યાંપે "Personal Home Page" માટે હતું, તે હવે "PHP: Hypertext Preprocessor" તરીકે ઓળખાય છે.
- 1997 માં ઝીવ સુરસ્કી અને એન્ડી ગુટમેન સાથે, રેમસસે PHP ને ફરીથી લઘ્યું અને જૂન 1998 માં PHP, version 3.0 release કર્યું. આ version પછી PHP ખૂબ લોકપ્રિય બન્યું.
- May 2000 માં PHP નું version 4.0 લોચ કરવામાં આવ્યું હતું. આ version માં session handling, output buffering અને વિવિધ પ્રકારના વેબ સર્વર પ્લેટફોર્મ માટે સપોર્ટ શામેલ છે.
- 2004 માં object oriented programming concept સાથે PHP, 5.0 version release કરવામાં આવ્યું.

### Open Source

- સામાન્ય રીતે, open source એવા કોઈપણ પ્રોગ્રામને refer કરે છે જેનો source code ઉપયોગ અથવા મોડિફિકેશન માટે ઉપલબ્ધ કરાયો છે. Open source સોફ્ટવેર સામાન્ય રીતે public collaboration તરીકે ડેવલોપ કરવામાં આવે છે અને freely available બને છે. તેનો અર્થ એ છે કે કોઈ પણ લાઇસન્સ લીધા વગર ઉપયોગ કરી શકાય છે.
- Open Source એ Open Source Initiative (OSI) દ્વારા સંચાલિત certification mark છે. સોફ્ટવેરના ડેવલોપર જે freely રીતે share કરવામાં આવ્યાં છે અને સંભાવિત રૂપે સુધારેલ છે અને બીજા દ્વારા ફરીથી distribute કરવામાં આવે છે તે Open Source trademark નો ઉપયોગ કરી શકે છે જો તેમનું distribution એ OSI ની Open Source Definition ને અનુસૂપ હોય. Summary માટે, distribution terms ની definition model ની આવશ્યકતા છે:
  - Distributed સોફ્ટવેર કોઈ પણ પ્રકારના restriction વગર ફરીથી distribute કરવું જરૂરી છે.
  - Source code હોવો જરૂરી છે (જેથી receiving party તેને improve કરવામાં અથવા modify કરવામાં able હશે).
  - Open Source નું ઉદાહરણ: Linux, Apache, MySQL, PHP.

### How PHP Works?

- PHP એ તમારા બ્રાઉઝર અને વેબ સર્વર વચ્ચે હોય છે. જ્યારે તમે તમારા બ્રાઉઝરમાં કોઈ PHP, વેબસાઇટનું URL ટાઇપ કરો છો, ત્યારે તમારું બ્રાઉઝર એ વેબ સર્વરને request મોકલે છે. ત્યારબાદ વેબ સર્વર તે પેજ પર PHPની script ને send કરે છે. PHP, મોડ્યુલ script ને execute કરે છે જે પછી તમે HTMLના સ્વરૂપમાં રિઝલ્ટ ને તમારા બ્રાઉઝર પર જોશો જે તમે સ્ક્રીન પર જોયું છો.



### Relationship between Apache, MySQL and PHP (AMP Module)

- AMP એ Apache, MySQL અને PHP માટે છે.

#### PHP

- PHP એ server side scripting language છે જે ડાયનેમિક વેબસાઇટ બનાવવા માટે ડિઝાઇન કરવામાં આવી હતી.
- PHP script એ વેબ સર્વર પર ચાલે છે.
- PHP એ loosely type language છે કારણ કે તે આપમેળે ડેટા ટાઇપ ના વેરીએબલ ને convert કરે છે.
- PHP એ powerful scripting language છે જે PHP, ઇન્સ્ટોલ કરેલા કોઈપણ કમ્પ્યુટરની કમાંડ લાઇન માં રન કરી શકાય છે.
- જો કે ડાયનેમિક વેબસાઇટ બનાવવા માટે ફક્ત PHP, પૂરતું નથી.

#### Apache

- વેબ સાઇટ પર PHP નો ઉપયોગ કરવા માટે, તમારે એવા સર્વરની જરૂર છે જે PHP scripts પર process કરી શકે. Apache એ એક ફી વેબ સર્વર છે, જે એકવાર કમ્પ્યુટર પર ઇન્સ્ટોલ થાય છે અને પછી ડેવલોપર લોકલ રીતે PHP scripts નું invaluable કરવા દે છે. આ તમારા લોકલ ડેવલોપમેન્ટ environment નો invaluable piece બનાવે છે.
- ખધા વેબ સર્વરોની જેમ, Apache HTTP request ને accepts કરે છે અને HTTP response આપે છે.
- Apache એ Apache Software Foundation દ્વારા distribute open source ફી સોફ્ટવેર છે.

#### MySQL

- વધુમાં, ડાયનેમિક વેબસાઇટ એ સ્ટોર થયેલી માહિતી પર dependent છે જે ઝડપથી અને સરળતાથી સુધારી શકાય છે; ડાયનેમિક સાઇટઘને static HTML સાઇટ વચ્ચે આ મહત્વ નો તફાવત છે.

- જો કે, PHP એ ડેટા સ્ટોર કરવા માટે એક સિમ્પલ અને efficient way આપતું નથી. જ્યાં MySQL જેવી relational database management system ચાલે છે.
- PHP, તેના માટે મૂળ સપોર્ટ provide કરે છે અને ડેટાબેઝ free, open-source પ્રોજેક્ટ છે.
- MySQL એ relational database management system (RDBMS) છે. આવશ્યક રીતે, આનો અર્થ એ છે કે MySQL એ યુઝર ને ડેટાના વિવિધ ભાગોને ગોઠવવા માટે rows અને columns નો ઉપયોગ કરીને ટેબલ-આધારિત structure માં માહિતી સ્ટોર કરવાનું allow કરે છે.

### Creating and Saving a PHP File

- સર્વર પર એક PHP, script execute કરવામાં આવે છે, અને plain HTML રિઝલ્ટ ને બ્રાઉઝર પર પાછિ sent કરવામાં આવે છે.
- PHP script **<? php** થી start થાય છે અને **?>** end થાય છે.
- ડોક્યુમેન્ટ માં કોઈપણ જગ્યા એ PHP, script મૂકી શકાય છે.
- Syntax:**

- ```
<?php
  // PHP code goes here
?>
```
- PHP, ફાઇલો માટે default ફાઇલ extension ".php" છે.
 - એક PHP ફાઇલમાં સામાન્ય રીતે HTML ટેગ્સ, અને કેટલાક PHP, scripting code હોય છે.
 - નીચે, PHP script સાથે એક સિમ્પલ PHP ફાઇલનું એક ઉદાહરણ છે જેમાં "echo" એ વેબપેજ પર built-in PHP ફંક્શનનો ઉપયોગ કરે છે "Hello World!" ટેક્સ્ટને આઉટપુટ કરવા માટે.

```
<html>
<head>
  <title>Unit-2</title>
</head>
<body>
  <?php
    echo "Hello World";
  ?>
</body>
</html>
```

Output:

Hello World

PHP Start and End Tags

- Opening અને closing ટેગ્સ ની ચાર જુદી જુદી pairs છે જેનો ઉપયોગ php માં કરી શકાય છે.
- નીચે, અમારી પાસે એક simple PHP, script નું ઉદાહરણ છે જે બ્રાઉઝરને "Hello World" ટેક્સ્ટ મોકલે છે:

1. Standard PHP tags

```
<?php
    echo "Hello World";
?>
```

2. Short tags

```
<?
    echo "Hello World";
?>
```

- Short tags ફક્ત ત્યારે available હોય છે જ્યારે તેઓ short_open_tag php.ini configuration ફાઈલ directive દ્વારા enabled હોય છે, અથવા જો php enable-short-tags ઓપ્શન સાથે configure હોય.

3. Script tags

```
<script language="php">
    echo "Hello World";
</script>
```

- PHP 7.0.0 પછી આ syntax દ્વારા કરવામાં આવી.

4. ASP style tags

```
<%
    echo "Hello World";
%>
```

- PHP 7.0.0 પછી આ syntax દ્વારા કરવામાં આવી.

Comments in PHP

- PHP એ C, C++, Java languages જેવી comment ને પણ સપોર્ટ કરે છે.
- PHP, બે પ્રકારની comments ને supports આપે છે:

Single line comment

- Single line comment // સિમ્બોલ સાથે start થાય છે.
<?php // આ single line comment છે ?>
- જો તમારી પાસે single line comment છે, તો બીજો ઓપ્શન # sign નો ઉપયોગ કરી શકાય છે.
<?php # આ single line comment છે ?>

Multiline comment

- Multiline comment સિમ્બોલ /* સાથે start થઈ અને સિમ્બોલ */ સાથે end થઈ.
- તેથી, multiline comment એક line થી શરૂ થાય છે અને એકથી વધુ line માં expand થઈ શકે છે.
<?php /* આ multiline Comment */ ?> છે.

PHP Echo and Print Statements

- તે બંનેનો ઉપયોગ સ્ક્રીન પર આઉટપુટ ડેટા મેળવા માટે છે.

Echo

- Echo નો સૌથી સરળ ઉપયોગ string ને argument તરીકે print કરવા માટે છે, ઉદાહરણ તરીકે:
echo "Hello World";

Or equivalently:

echo ("Hello World");

- તમે unparenthesized version પર multiple arguments પણ આપી શકો છો, commas દ્વારા separated:
echo "Hello", "World";
- parenthesized version, જોકે, multiple arguments accept કરશો નહોં: echo("Hello", "World");

Input	Output
echo "Hello World";	Hello World
echo("Hello World");	Hello World
echo "Hello", "World";	Hello World
echo("Hello", "World");	Invalid Argument

Print

- Command print એ echo ની સમાન છે, બે મહત્વપૂર્ણ તફાવતો સાથે:
 - Print માત્ર એક argument ને accept શકે છે.
 - Print એ વેલ્યુ રિટર્ન કરે છે, જે દર્શાવે છે કે print statement success થયું છે કે કેમ.
- Printing દ્વારા return આવતી વેલ્યુ 1 હશે તો printing successful થયું હોય તો જો 0 હોય તો unsuccessful હોય છે.
- ઉદાહરણ તરીકે, નીચેની બે લાઇન exactly સરખી વસ્તુને print કરશો:

```
print("3.14159 "); //print a string
print(3.14159); //print a number
```

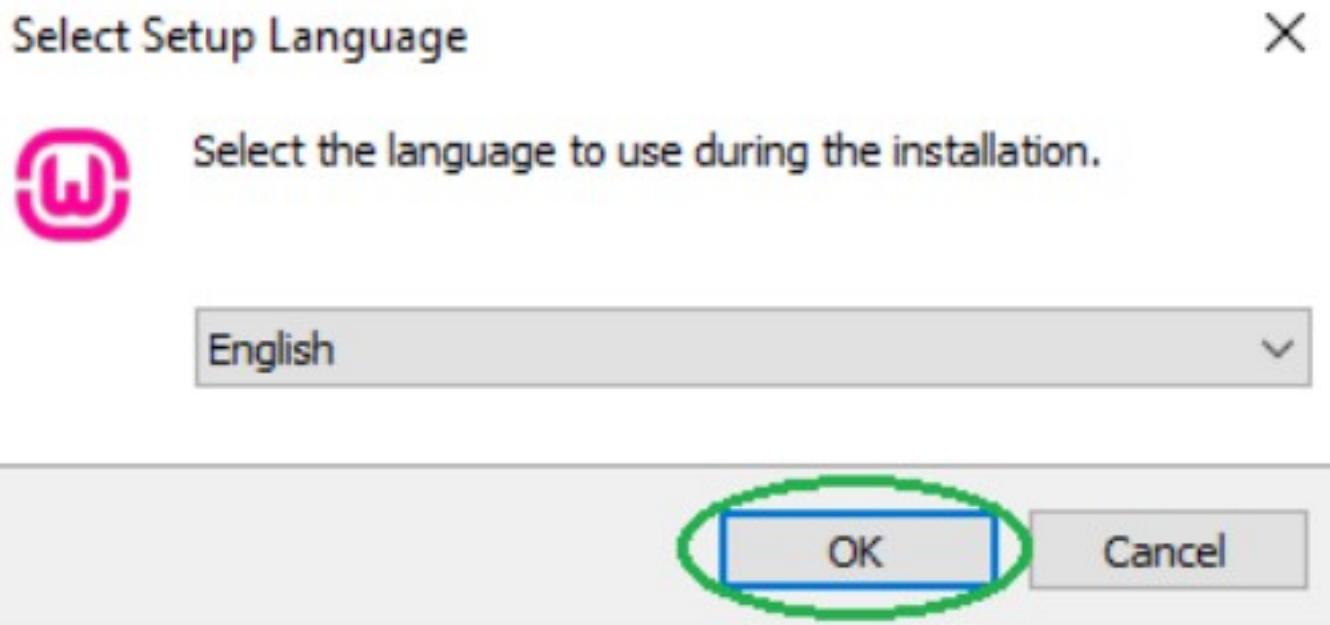
Difference between Echo and Print

ECHO	PRINT
તે multiple expressions ને accept કરી શકે છે.	તે multiple expressions ને accept કરી શકે નહોં.
તે print કરતાં ઝડપી છે.	તે echo કરતાં ધીમું છે.
તે multiple string ને (,) તરીકે અલગ કરી શકે છે.	તે multiple arguments ને pass કરી શકતું નથી.
તે આઉટપુટ display કરવા માટે ઉપયોગમાં લેવાતું એક statement છે અને તેને parentheses echo સાથે અથવા parentheses echo વગર વાપરી શકાય છે.	તે આઉટપુટ display કરવા માટે ઉપયોગમાં લેવાતું એક statement છે અને તેને parentheses print() સાથે અથવા parentheses print વગર વાપરી શકાય છે.
તે કોઈપણ વેલ્યુ return કરતું નથી.	તે એક વેલ્યુ return કરે છે.

Installation of WAMP Server

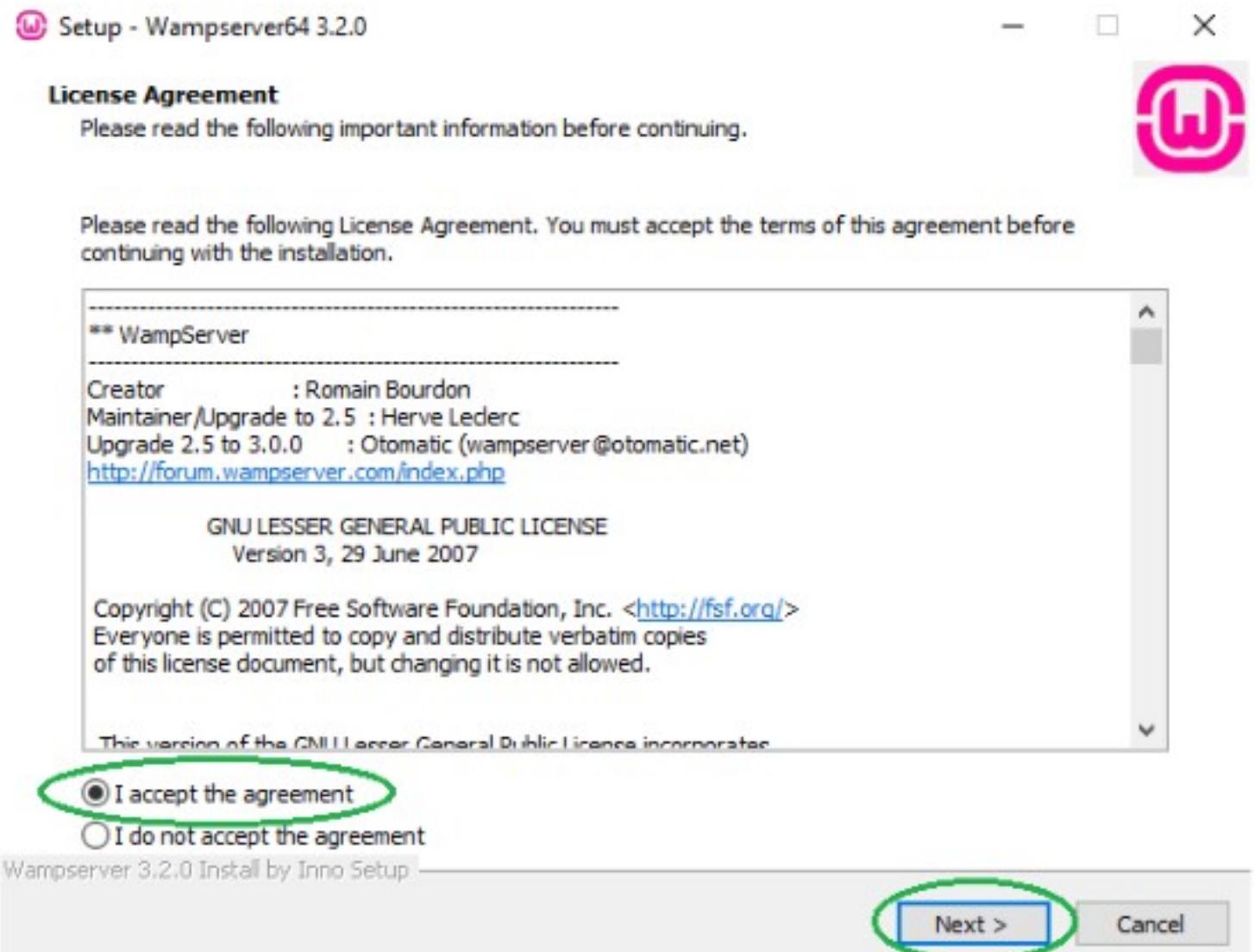
- **Step-1 : Download the WAMP Server**
- **Step-2 : Select Language**

WAMP मां तमे जे लेंगवेज वापरवा मांगो छो ते पसंद करवा माटे तमने पूछवामां आवशे. नीये आवता dropdown box ना ऐरो पर क्लिक करो, लिस्टमां तमारी लेंगवेज पसंद करो, पछी इन्स्टोलेशन प्रोसेस चालु राखवा माटे Ok पर क्लिक करो.



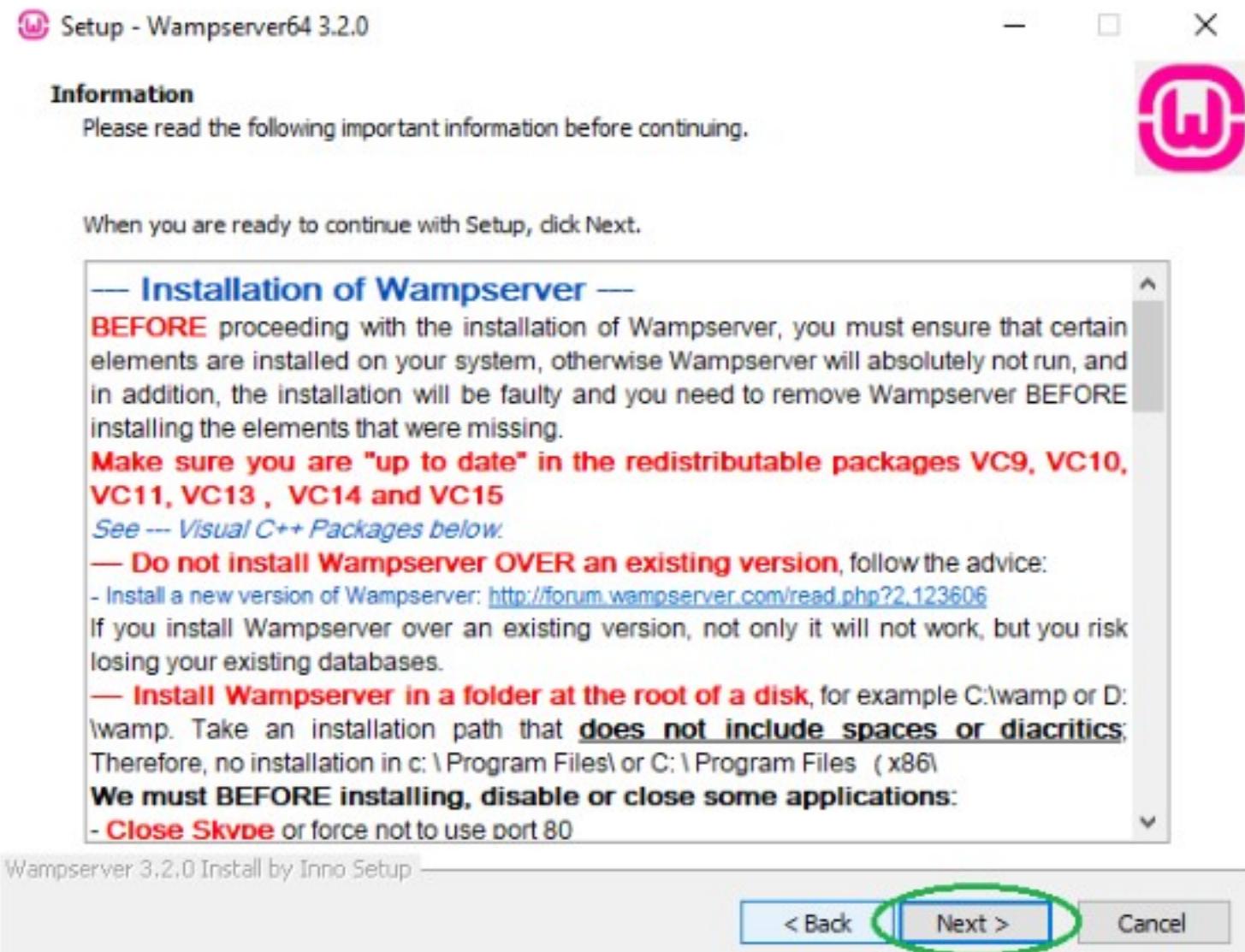
- **Step-3: License Agreement**

अही घतावेल स्क्रीन ए License Agreement नी छे. Agreement वांचो, I accept the agreement तेनी पासेनु रेडियो बटन क्लिक करो पछी इन्स्टोलेशन चालु राखवा माटे Next पर क्लिक करो.



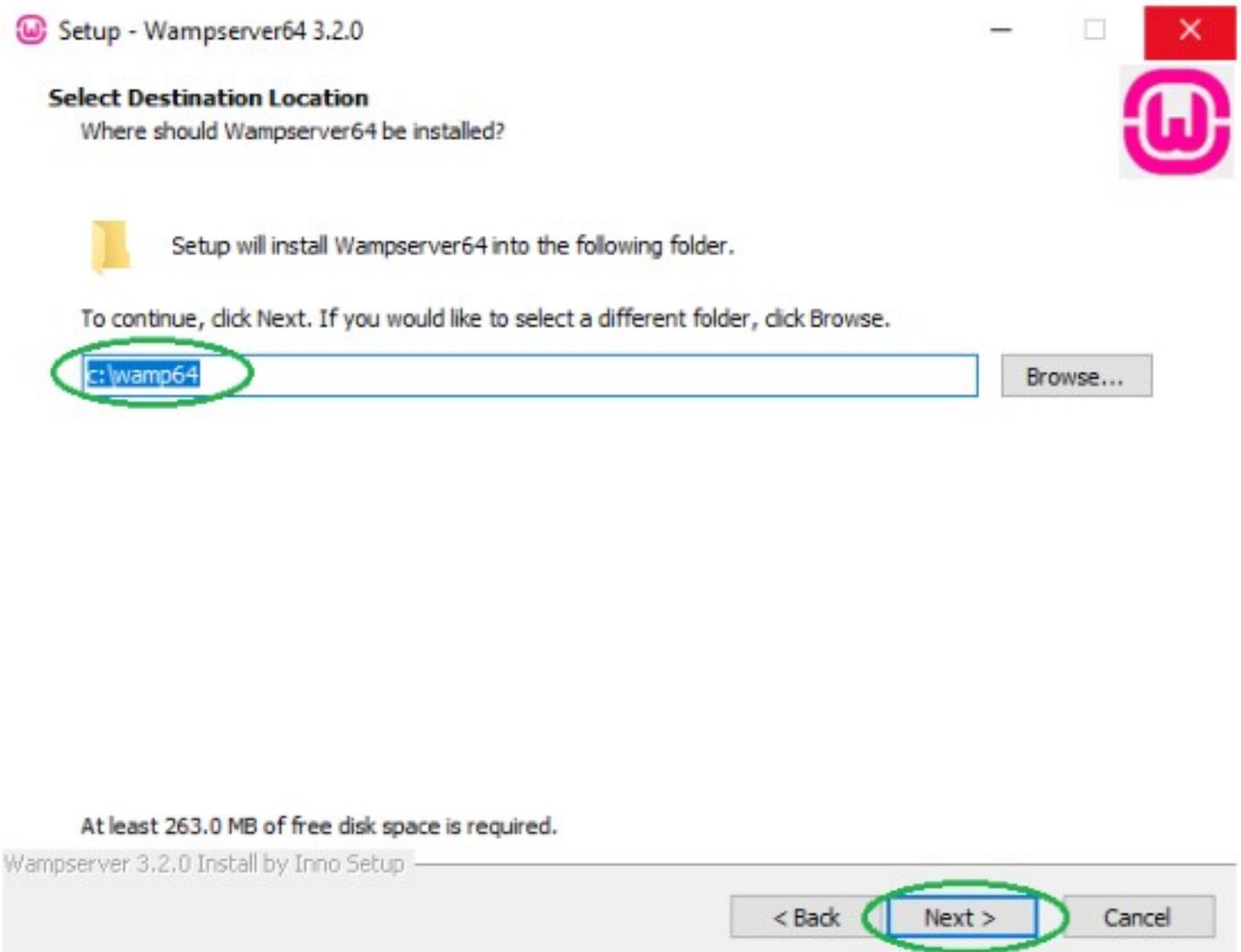
- **Step-4: Please read the following**

आपेल महत्वपूर्ण माहिती वांचो, पछी इन्स्टोलेशन चालु राखवा माटे Next पर क्लिक करो.



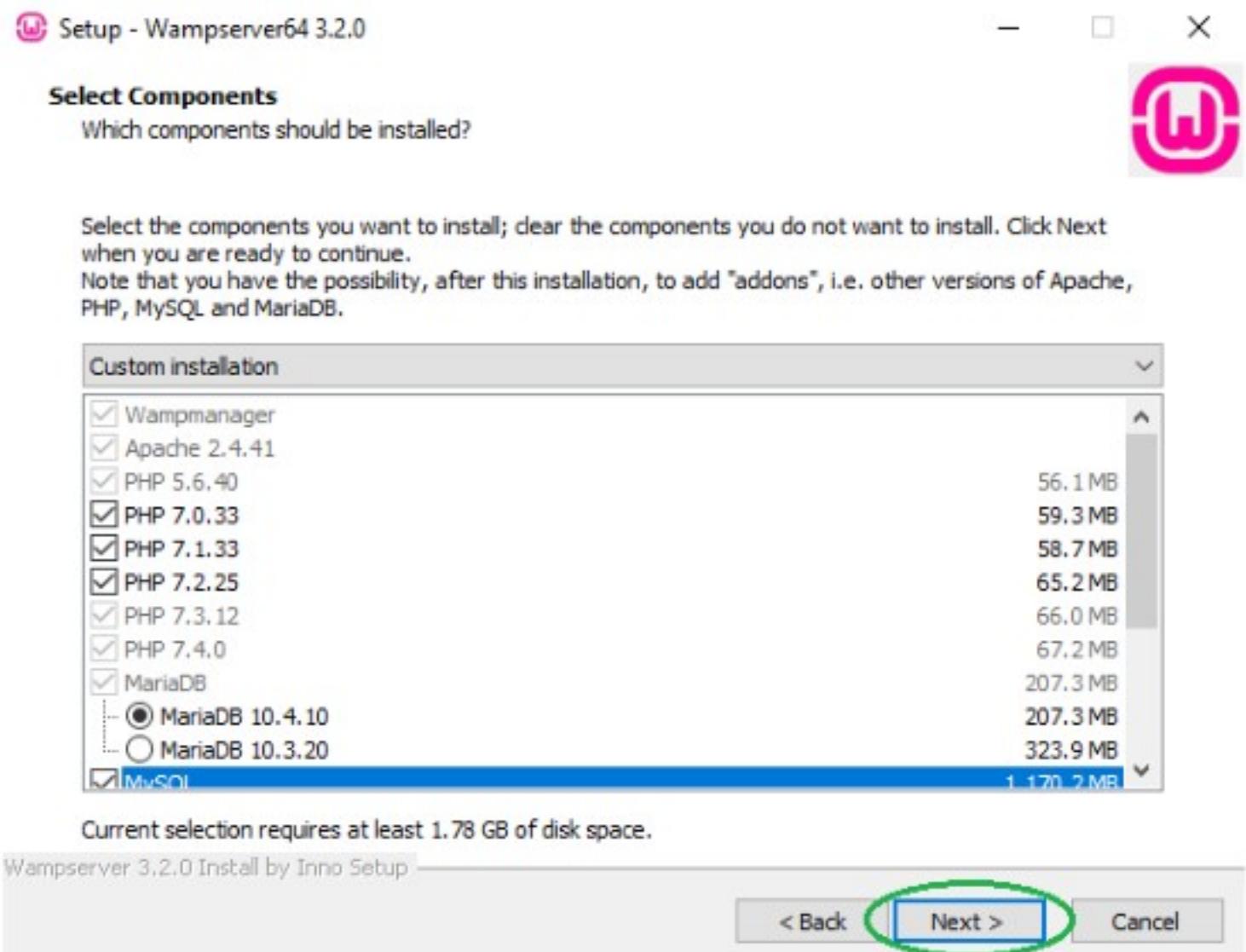
- Step-5: Select Destination Location**

આગળ તમે Destination નું લોકેશન સિલેક્ટ કરો. જો તમે બીજુ ડાઇવ પર WampServer ઇન્સ્ટોલ કરવા માંગતા ન હો, ત્યાં સુધી તમારે કંઈપણ બદલવાની જરૂર નથી. Next પર ક્લિક કરો.



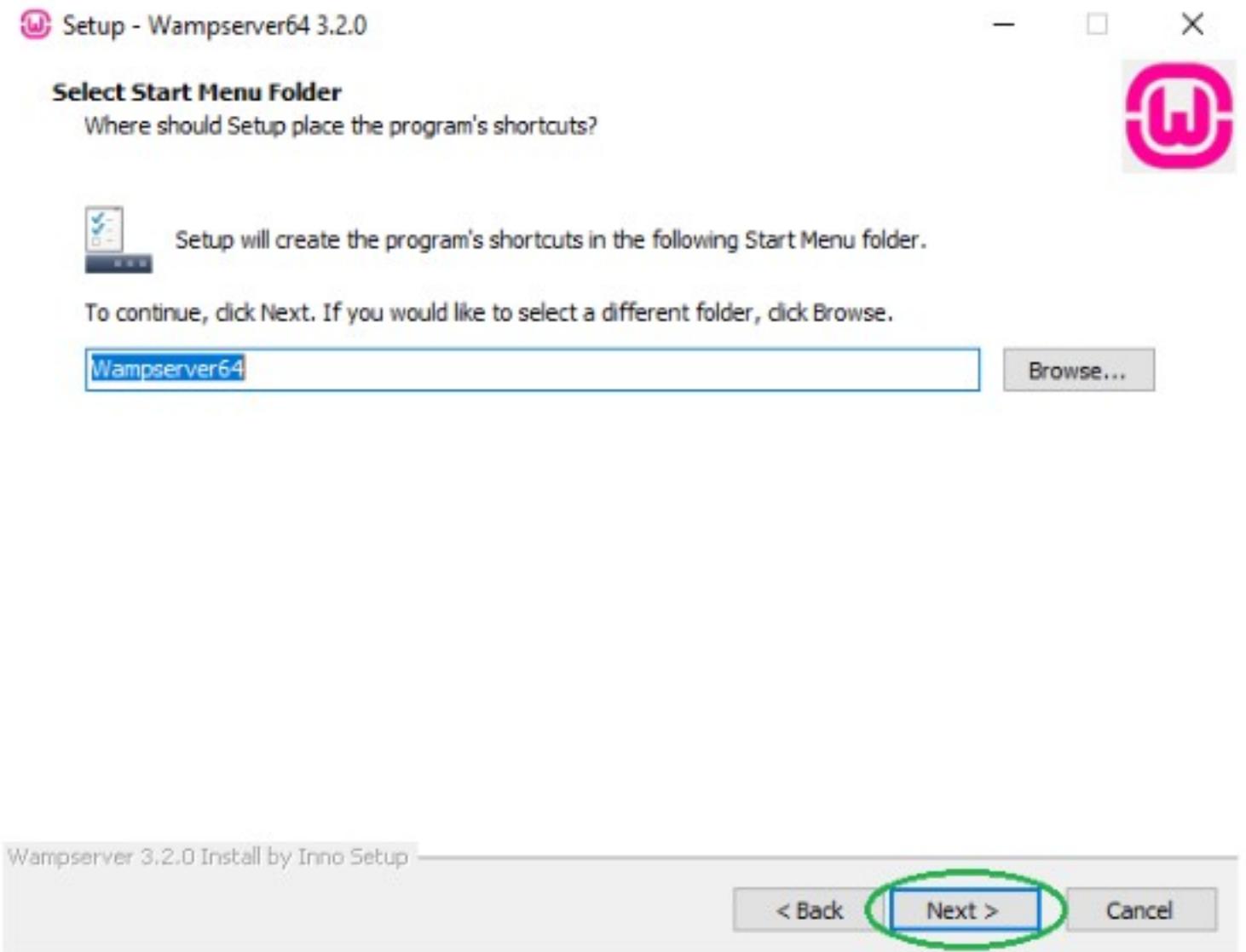
- Step-6: Select Components**

તમે ઇન્સ્ટોલ કરવા માંગો છો તે components પસંદ કરો; તમે ઇન્સ્ટોલ કરવા માંગતા નથી તેવા components clear કરો. જ્યારે તમે ચાલુ રાખવા માટે તૈયાર હો ત્યારે Next પર ક્લિક કરો.



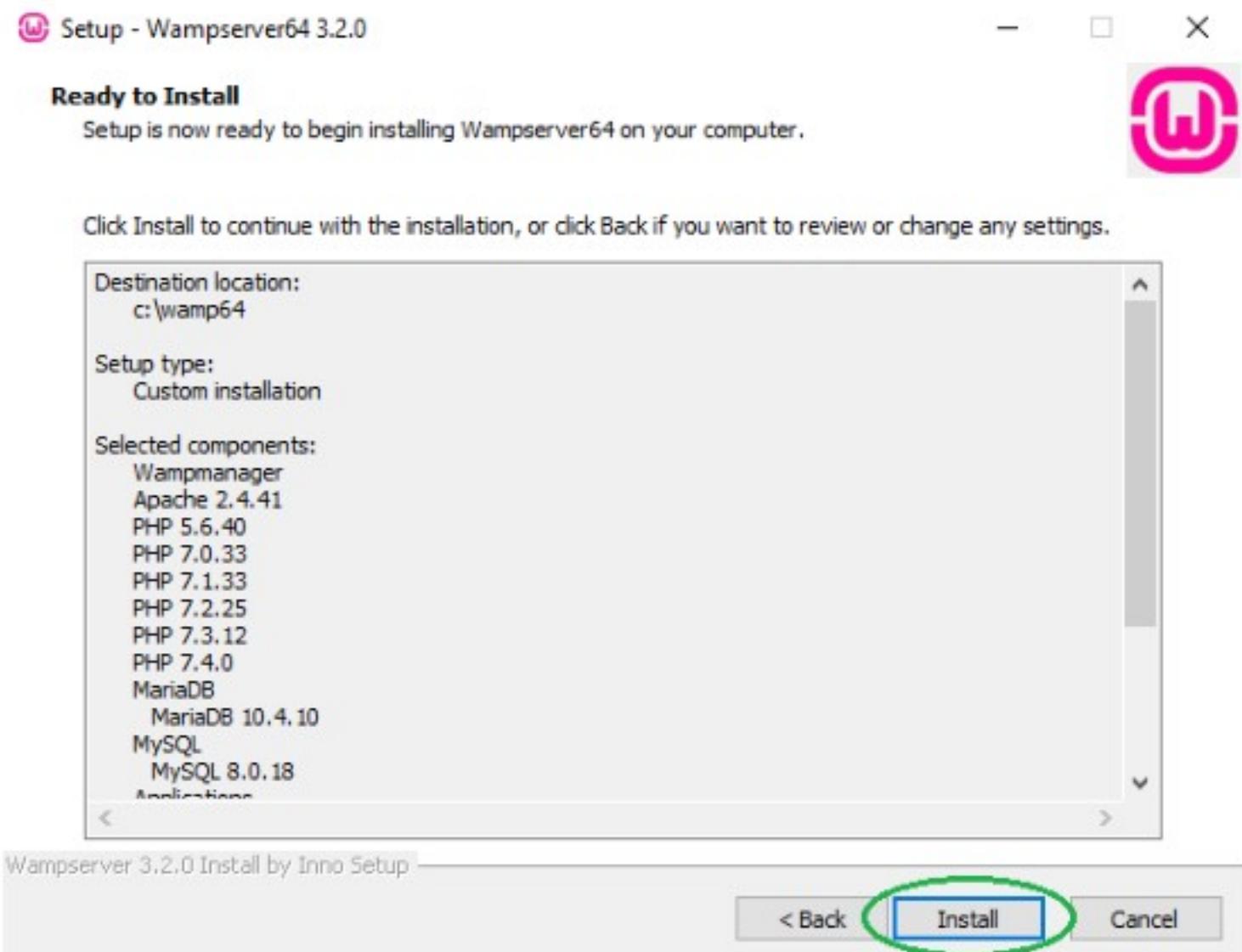
- Step-7: Select Start Menu Folder**

પ્રોગ્રામના શોર્ટકટ્સને સેટઅપમાં ક્યાં મૂકવા જોઈએ? જો તમે કોઈ અલગ ફોલ્ડર પસંદ કરવા માંગતા હો, તો બાઉઝ કરો પછી Next પર ક્લિક કરો.



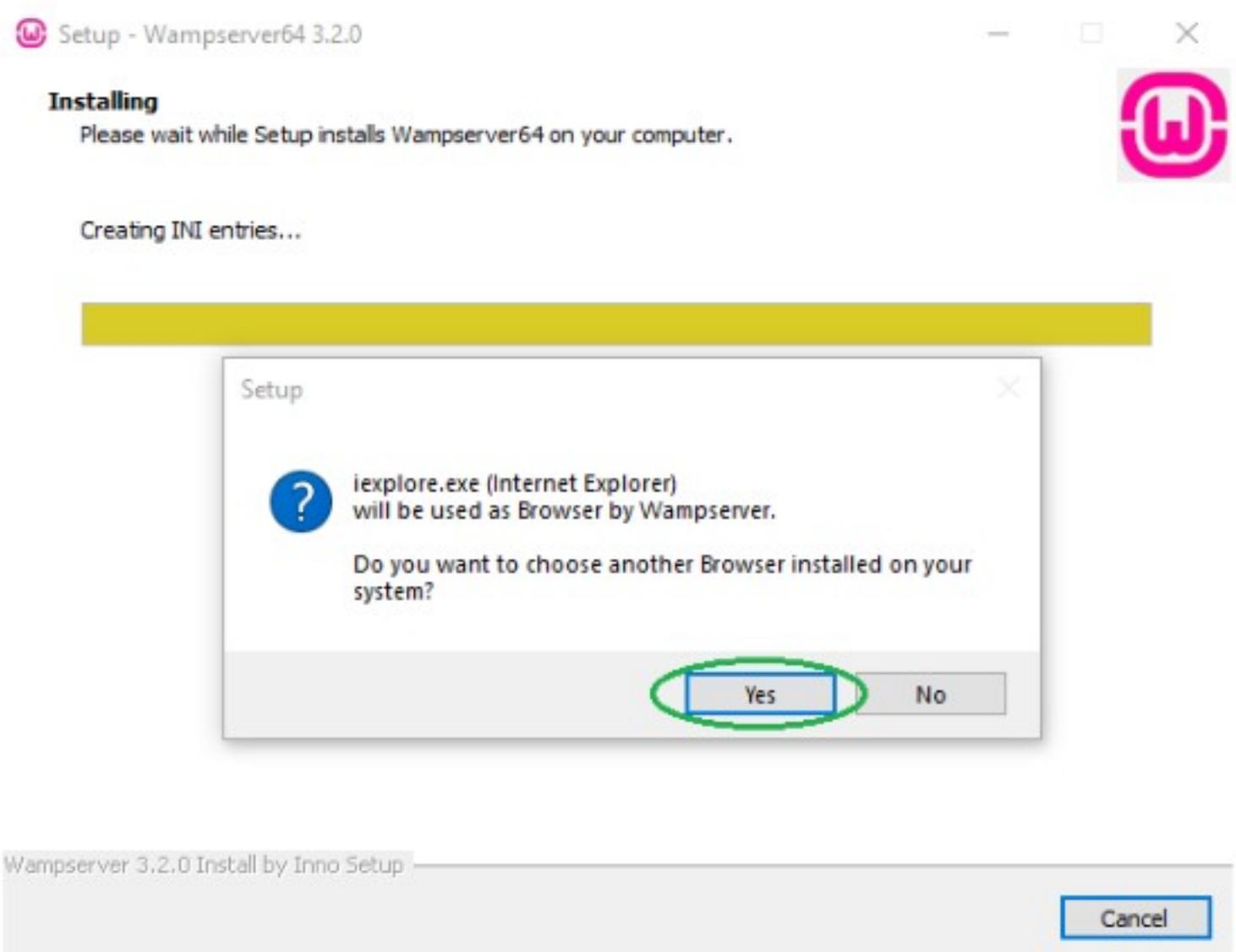
- Step-8: Ready to Install**

આગળ તમે ઇન્સ્ટોલ કરવા માટે Ready To Install સ્ક્રીન જોશો. જો તમે પસંદ કરો છો, તો તમે તમારી સેટઅપ પસંદગીઓને review કરી શકો છો અને તેમાંની કોઈપણ નોંધ સ્ક્રીન પર પાછા ક્લિક કરીને બદલી શકો છો. એકવાર તમે તમારી પસંદગીઓના review કરી લો, પછી ચાલુ રાખવા માટે ઇન્સ્ટોલ ક્લિક કરો.



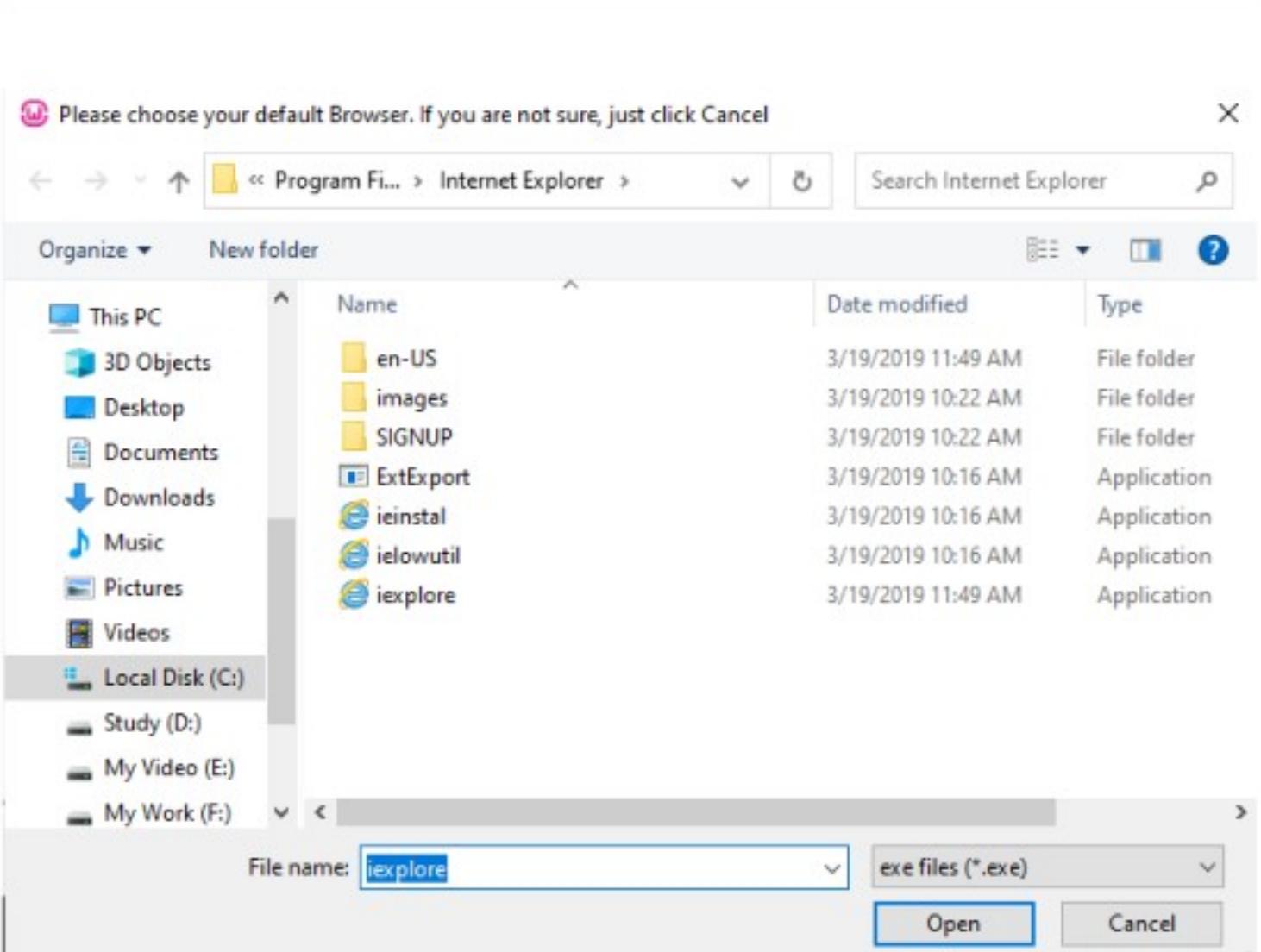
- Step-9: Installing**

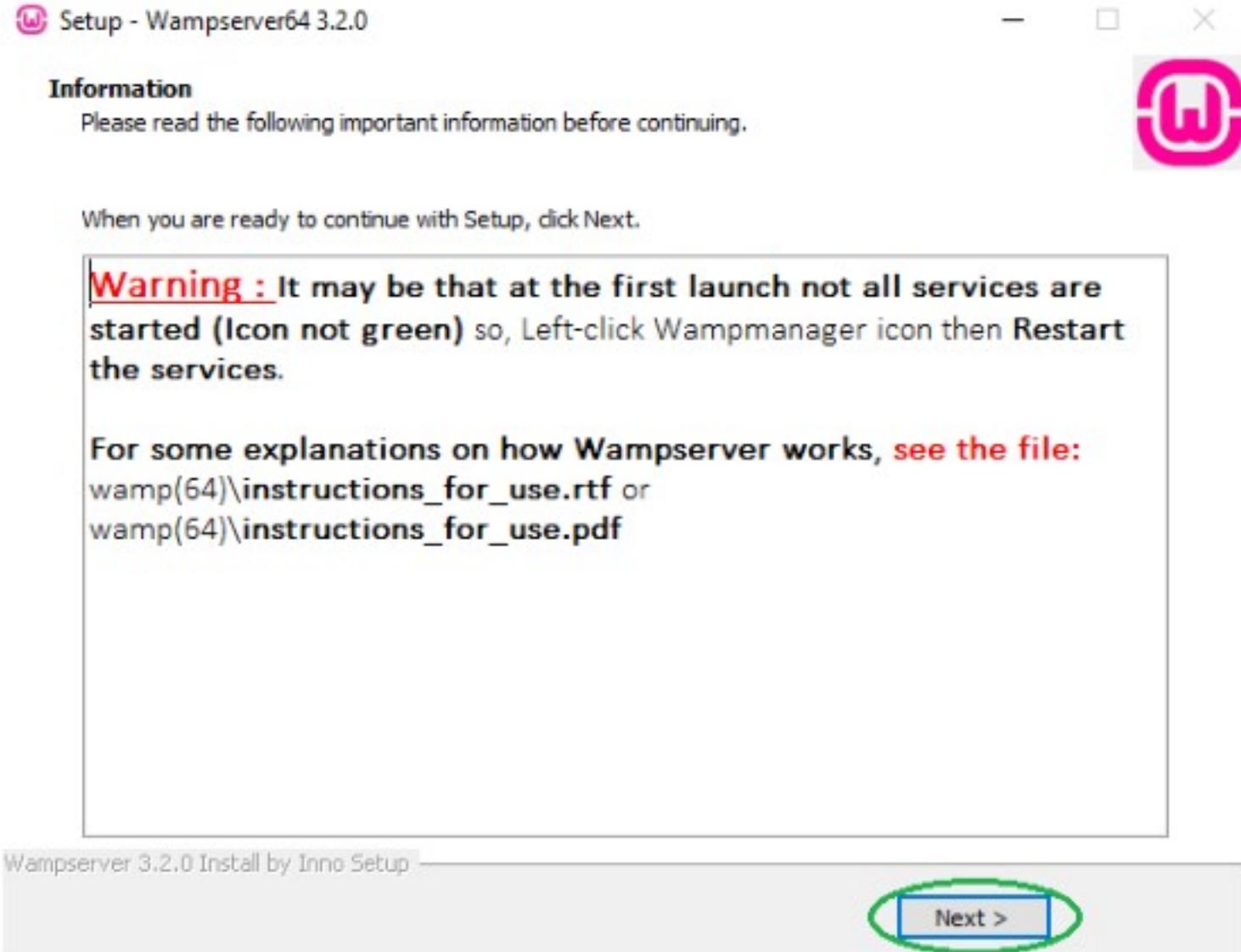
WampServer તમે પસંદ કરેલા location પર ફાઇલો extract કરવાનું ચાલુ કરશો, ચાલુ રાખવા માટે Yes પર ક્લિક કરો.

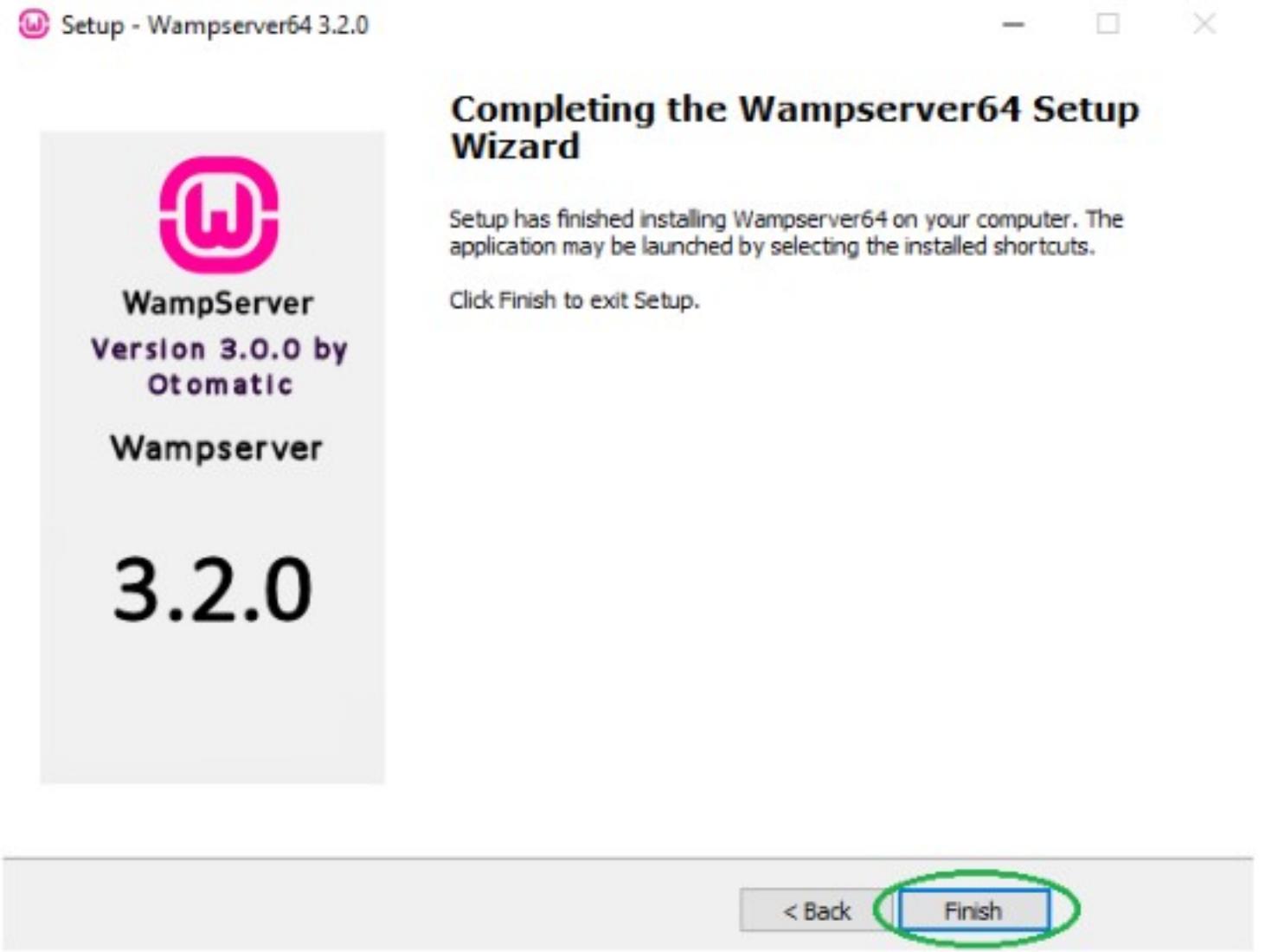


- Step-10: એકવાર ફાઇલો extract કર્યો પછી, તમને તમારું ડિફોલ્ટ બ્રાઉઝર પસંદ કરવાનું કહેવામાં આવશે. લોકલ ફાઇલ બ્રાઉઝર વિડો ખોલ્યા પછી WampServer એ ઇન્ટરનેટ એક્સ્�પ્રોસ્લોરને ડિફોલ્ટ કરે છે. જો તમારું ડિફોલ્ટ બ્રાઉઝર IE નથી, તો અનુરૂપ ફાઇલ .exe file માટે નીચેના લોકેશન પર ધ્યાન આપો:**

- **Opera:** C:\Program Files (x86)\Opera\opera.exe
- **Firefox:** C:\Program Files (x86)\Mozilla Firefox\firefox.exe
- **Safari:** C:\Program Files (x86)\Safari\safari.exe



- **Chrome:**
C:\Users\xxxxx\AppData\Local\Google\Chrome\Application\chrome.exe
 - તમારા ડિફોલ્ટ બ્રાઉઝરની .exe ફાઇલને પસંદ કરો, પછી ચાલુ રાખવા માટે ખોલો ક્લિક કરો.
- **Step-11: Warning**
જો પ્રથમ લોન્ચ થઈ શકે તો બધી સેવાઓ શરૂ કરવામાં આવતી નથી. તેથી, Wampmanager icon પર left-click કરો પછી સેવાઓ ફરીથી ચાલુ કરો. ચાલુ રાખવા માટે Next પર ક્લિક કરો.
- 

The screenshot shows the 'Information' step of the Wampserver 3.2.0 setup. It contains a warning message: 'Warning : It may be that at the first launch not all services are started (Icon not green) so, Left-click Wampmanager icon then Restart the services.' Below this, it provides links to 'instructions_for_use.rtf' and 'instructions_for_use.pdf' for explanations on how Wampserver works.
- **Step-12:** ઇન્સ્ટોલેશનની Complete સ્ક્રીન હવે દેખાશે, પછી ઇન્સ્ટોલેશન complete કરવા માટે Finish ક્લિક કરો.
- 

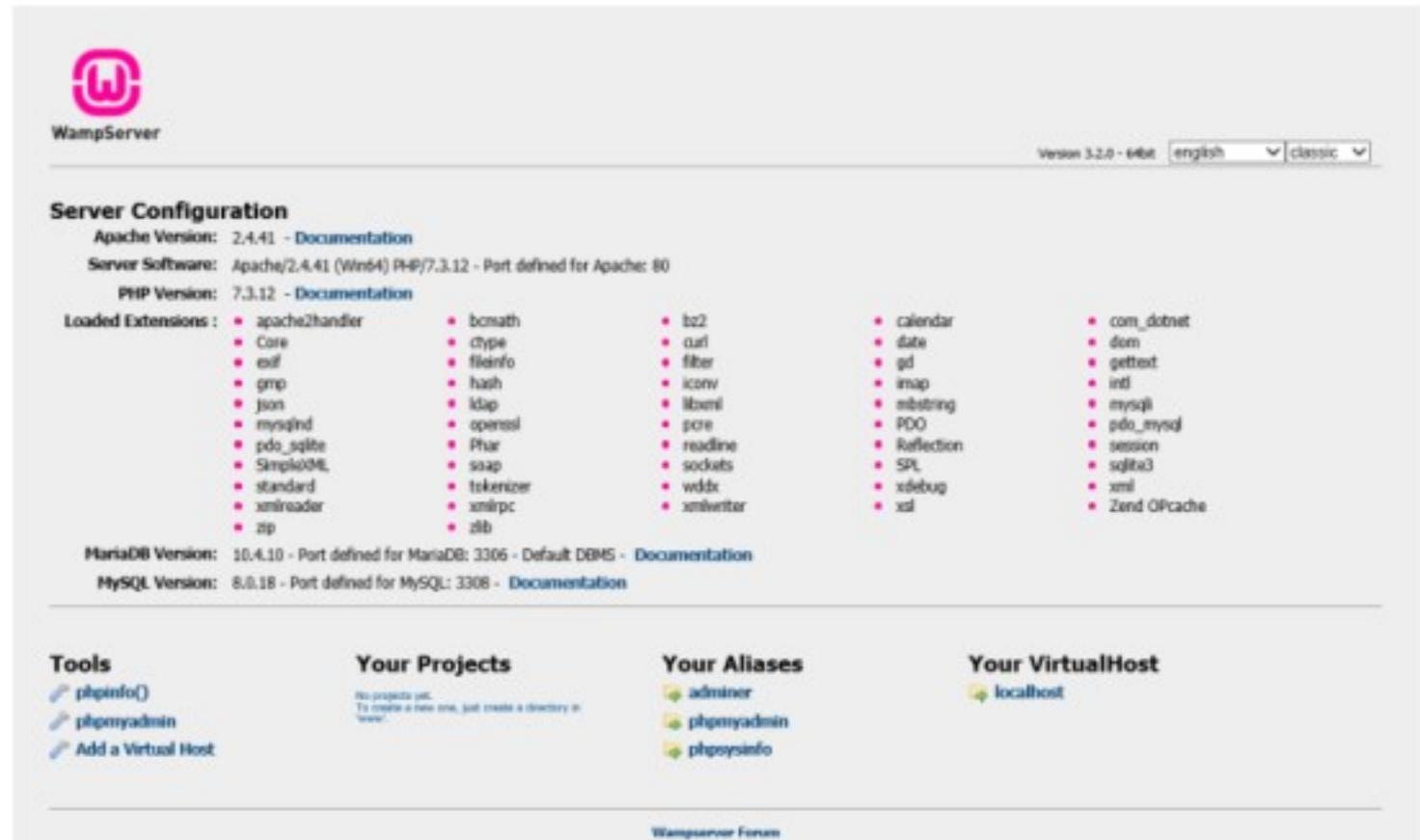
The screenshot shows the 'Completing the Wampserver64 Setup Wizard' step. It displays the WampServer logo, version 3.0.0 by Otomatic, and the text 'Setup has finished installing Wampserver64 on your computer. The application may be launched by selecting the installed shortcuts.' A 'Finish' button is highlighted with a green oval.
- **Step-13: Run Wamp Server**
તમારે તમારા ટાસ્કબારની જમણી બાજુએ iconમાં WampServer જેવું દેખાશે. જો icon લીલો છે, તો પછી બધું બરાબર કાર્ય કરી રહ્યું છે. જો icon નારંગી છે, તો પછી સર્વિસમાંથી કોઈ એક સાથે પ્રોભ્લેમ છે. જો icon લાલ છે, તો પછી
- 

The screenshot shows the Windows taskbar with two icons: a green circular icon with a white 'W' and a smaller orange icon with a white 'W'. Both icons have a green oval highlighting them, indicating they are active or running.

Apache અને MySQL બંને સેવાઓ ચાલી રહી નથી. ચાલુ રાખતા પહેલા તમારે તે issues resolve કરવાની જરૂર રહેશે.

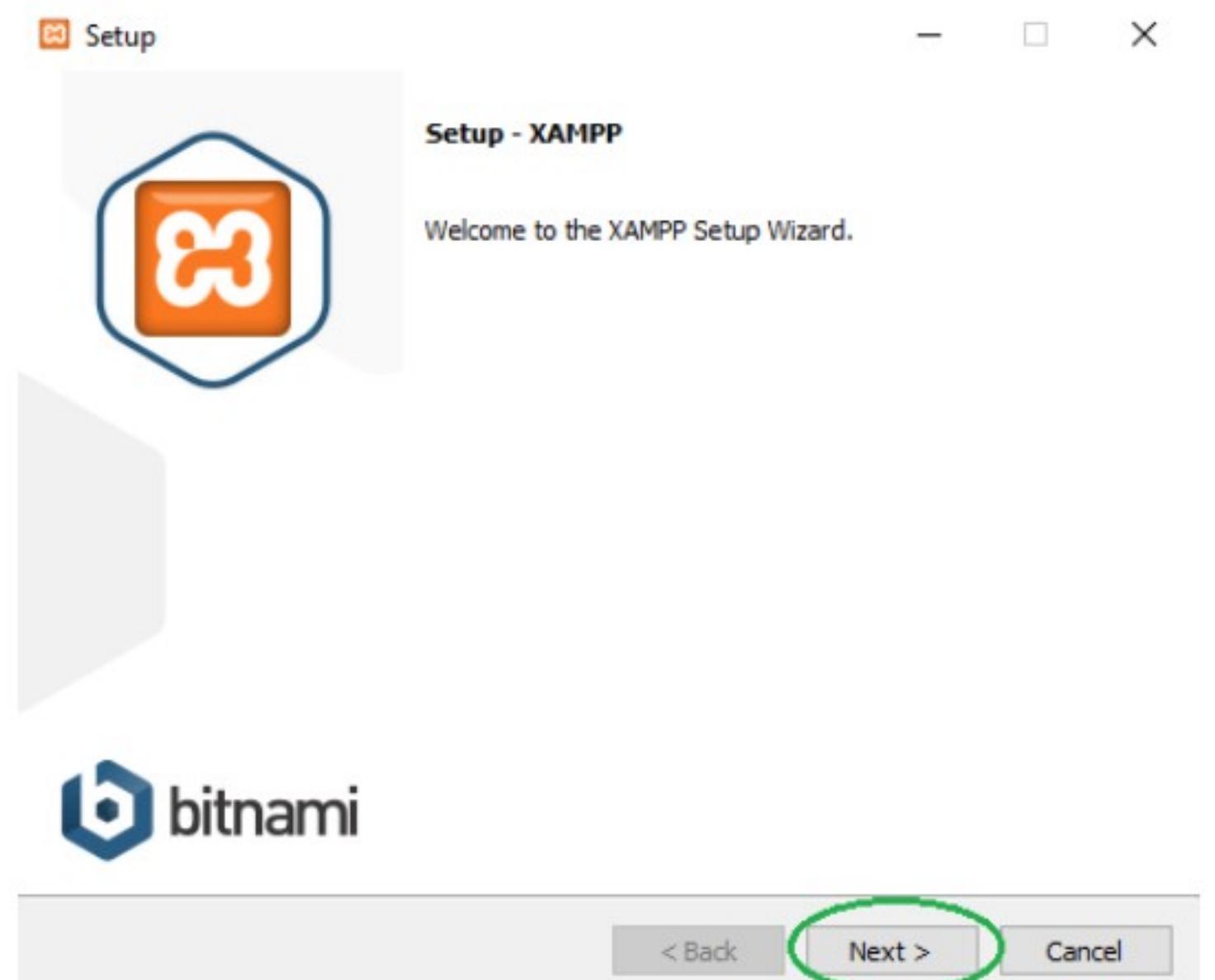
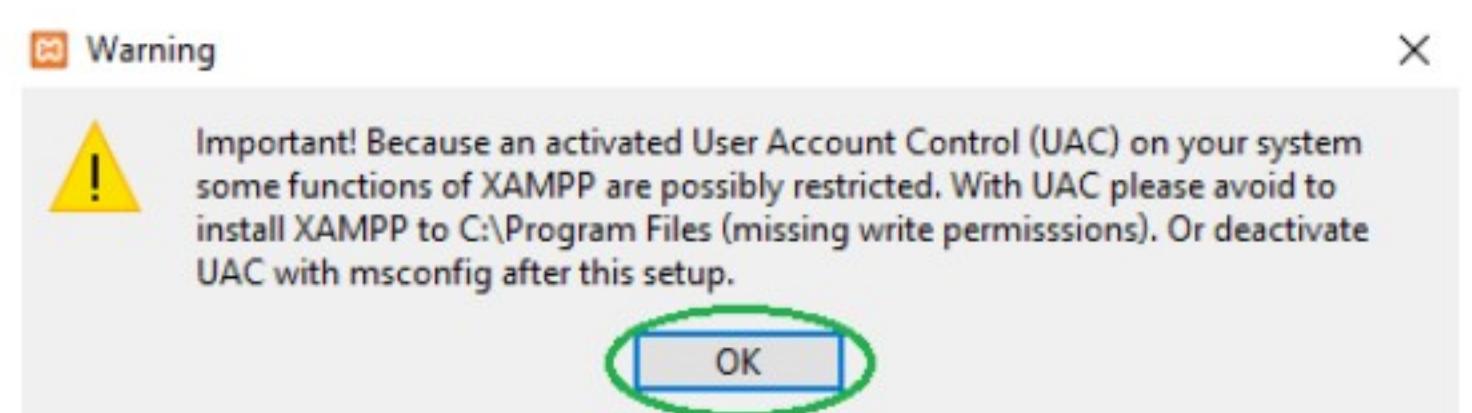


- **Step-14:** એકવાર તમે ઇન્સ્ટોલેશન પ્રક્રિયા complete કરી લો, પછી ચેક કરો કે તમારું ઇન્સ્ટોલેશન તમારા બાઉઝરમાં <http://localhost/> પર જઈને યોગ્ય રીતે કાર્ય કરી રહ્યું છે. તમારું WampServer homepage પર display થવું જોઈએ.



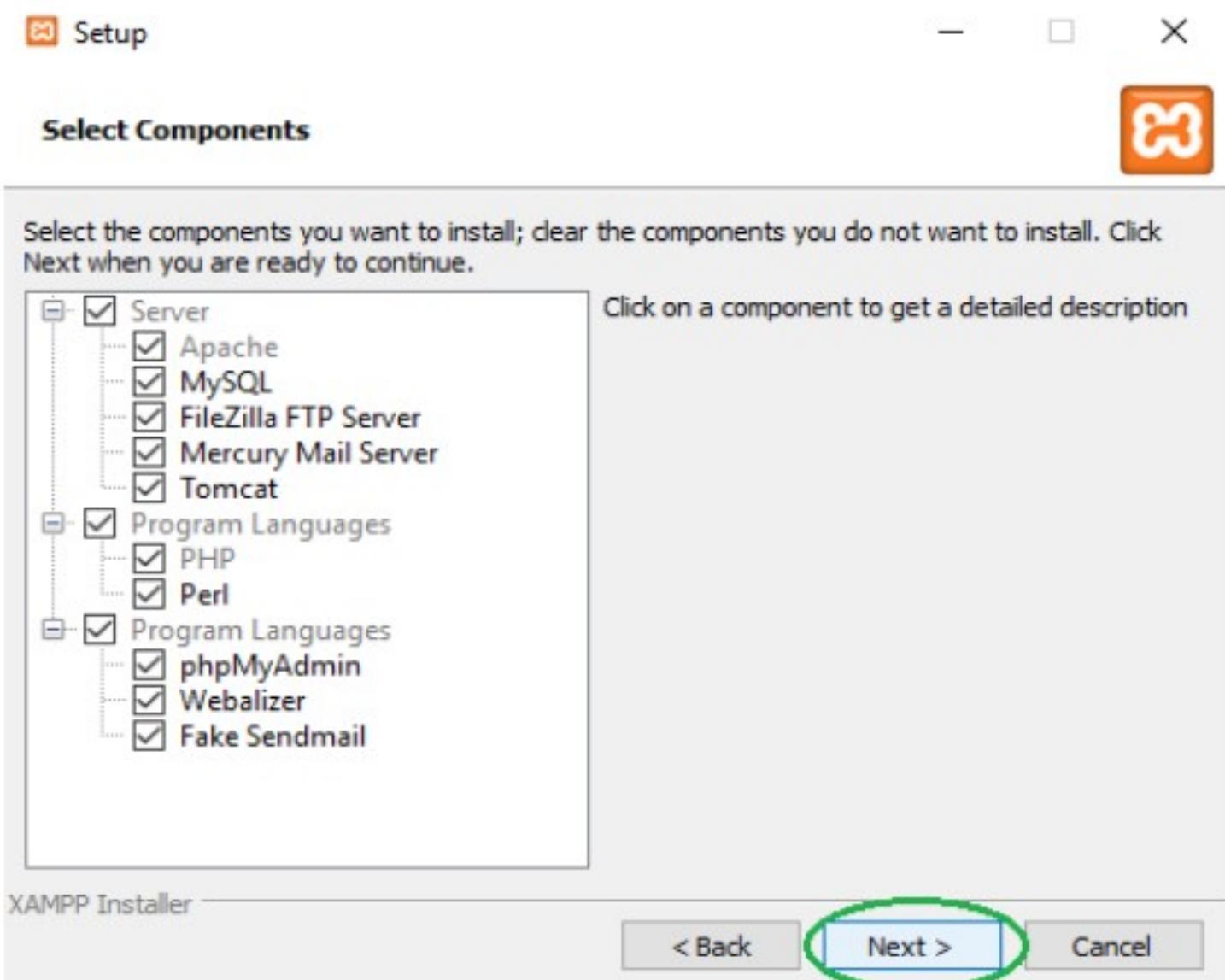
Installation of XAMPP Server

- **Step-1 : Download the XAMPP Server**
- **Step-2: User Access Control Warning**
તમે વિડો pop up, જોશો, તમને સિસ્ટમ પર યુઝર એકાઉન્ટ કંટ્રોલ (યુએસી) એક્ઝિટવ હોવા વિશે warning આપશો. ઇન્સ્ટોલેશન ચાલુ રાખવા માટે OK પર ક્લિક કરો.
- **Step-3: XAMPP Setup Wizard**
હવે તમે actual setup wizard જોશો. ચાલુ રાખવા માટે Next ક્લિક કરો.



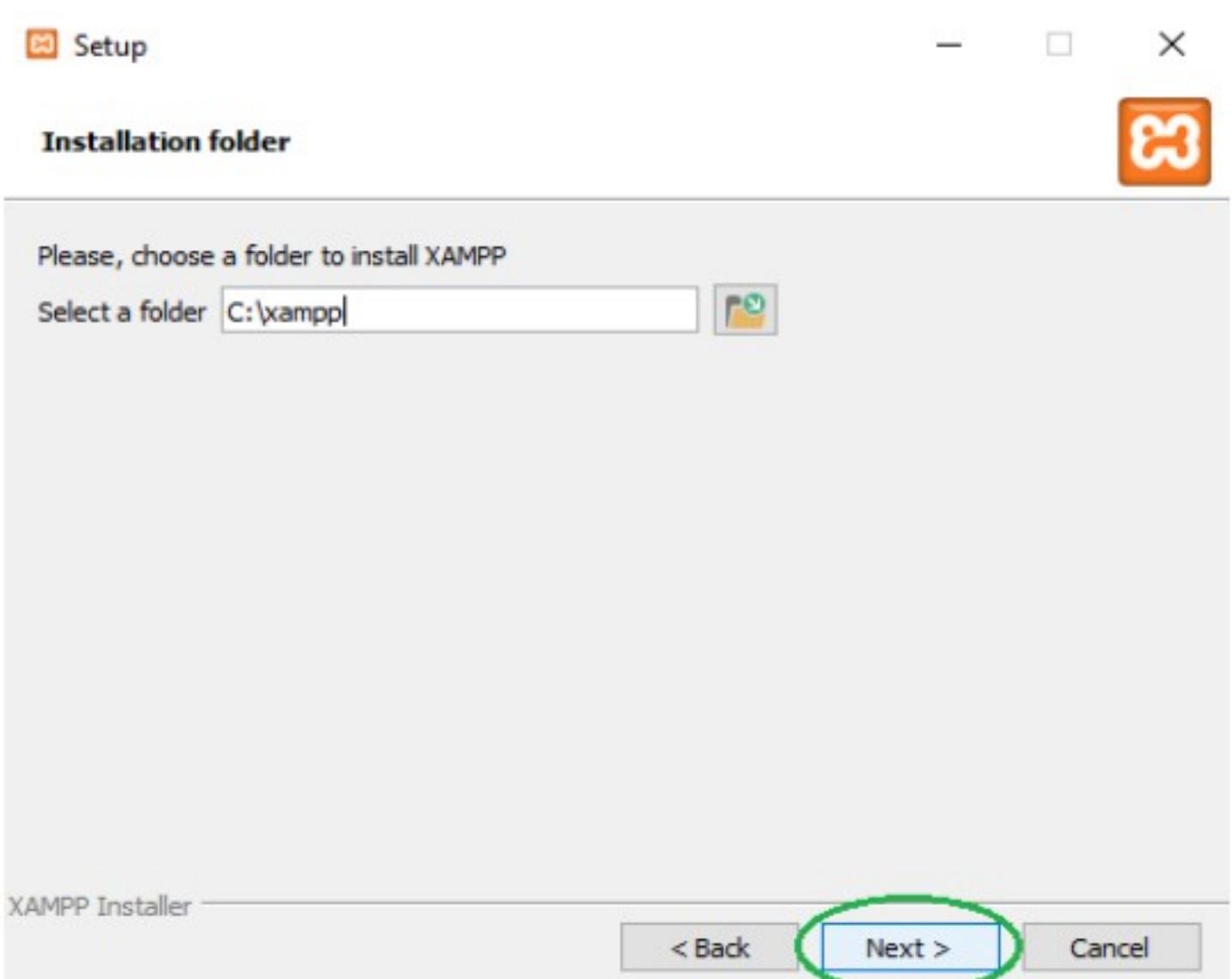
- Step-4: Select Component**

આ dialog box માં તમારે ઇન્સ્ટોલ કરવા માટે components સિલેક્ટ કરવા પડશે. ચાલુ રાખવા માટે Next પર ક્લિક કરો.



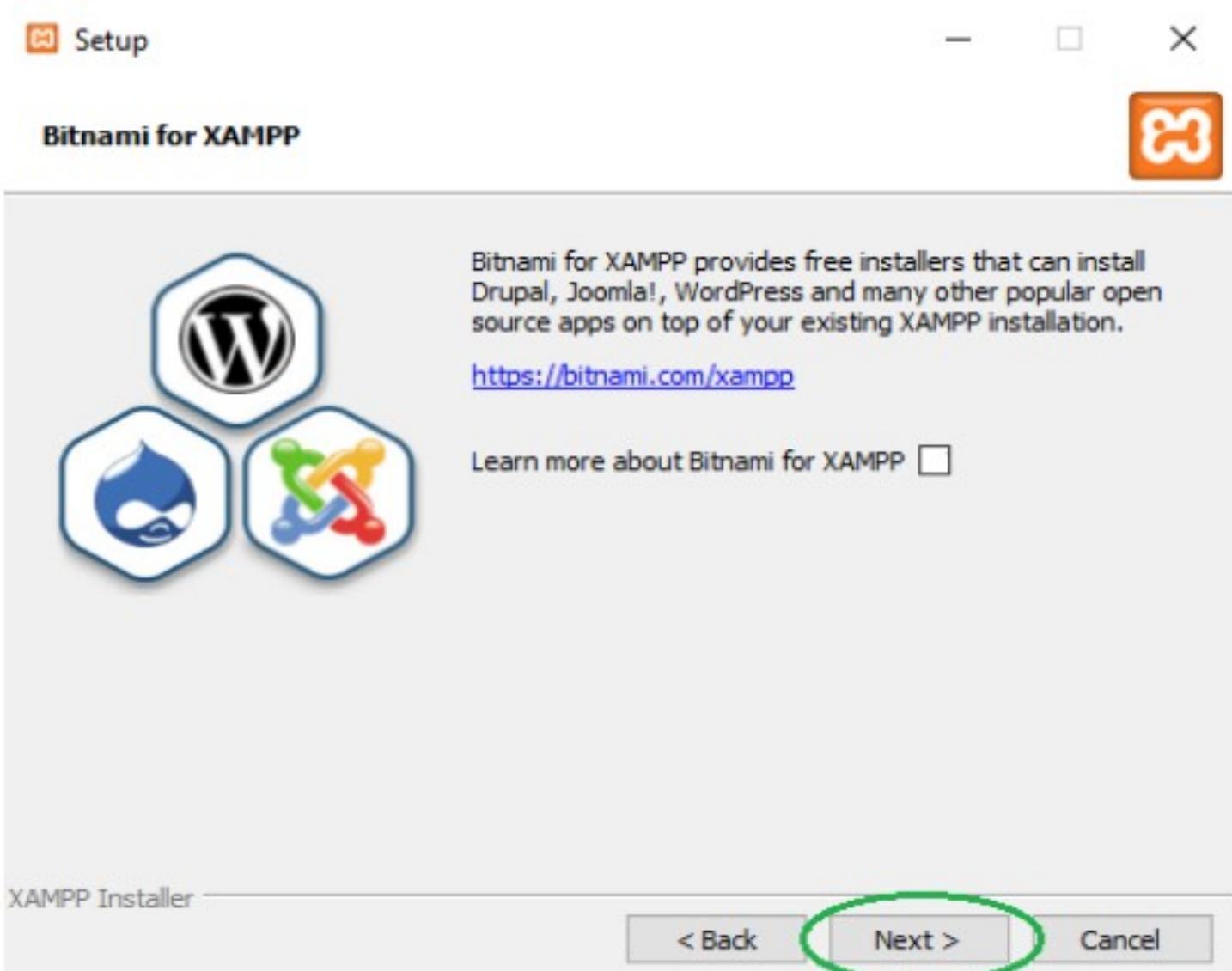
- Step-5: Select Installation Folder**

ઇન્સ્ટોલેશન ફોલ્ડર પસંદ કરો. જો તમે C drive પર XAMPP ઇન્સ્ટોલ કરવા માંગતા નથી, તો પછી આ ફોલ્ડરને બદલો નહીં તો તેને જેમ છે તેમ છોડી દો અને next પર ક્લિક કરો.



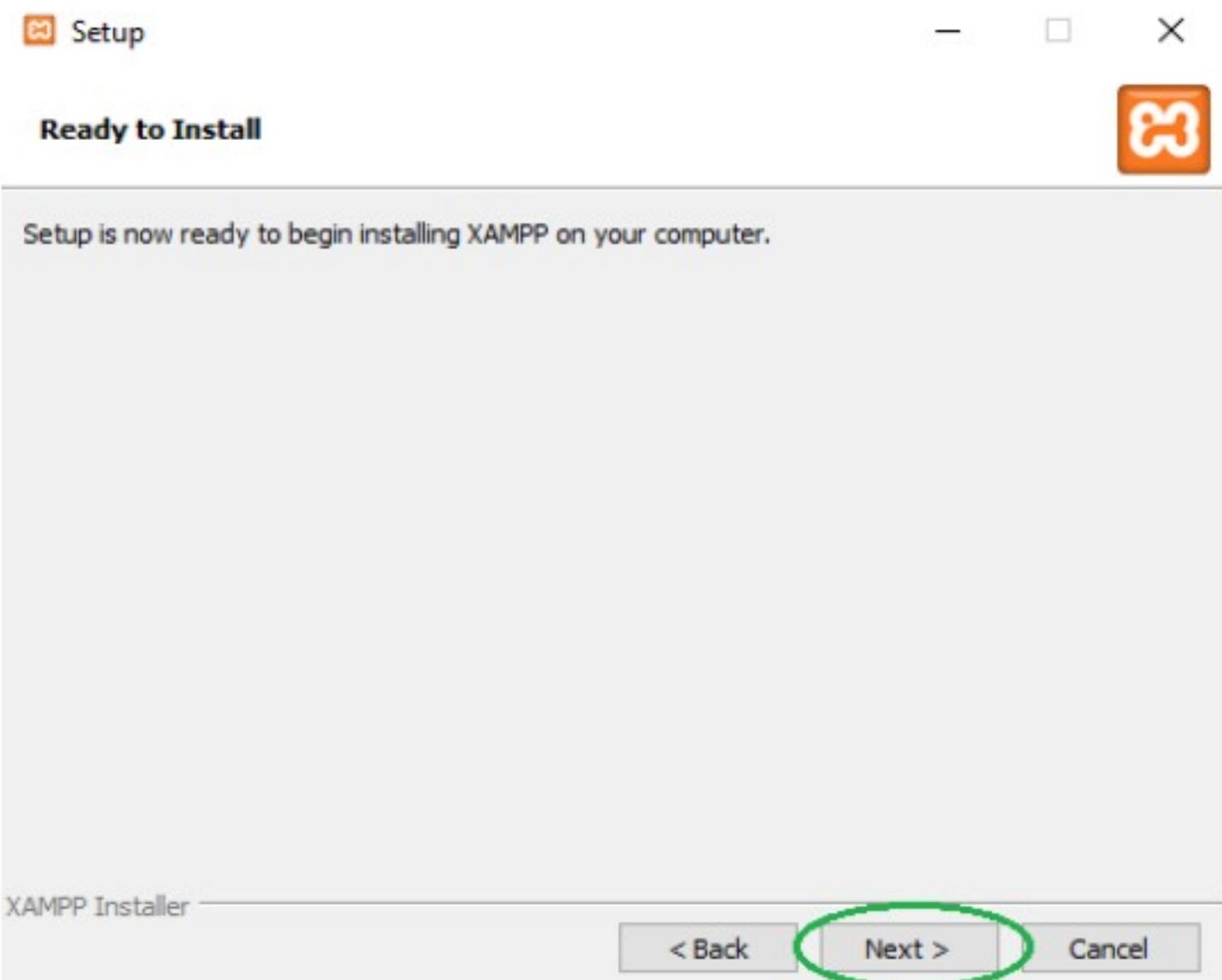
- Step-6: Bitnami for XAMPP**

Bitnami એ WordPress, Drupal, Joomla માટે મફત સેટઅપ આપે છે. પરંતુ આપણે આને હવે સ્ક્રીપ કરી શકીએ છીએ. તેથી, "Learn more about Bitnami" બોક્સને અનયેક કરો. Next પર ક્લિક કરો.



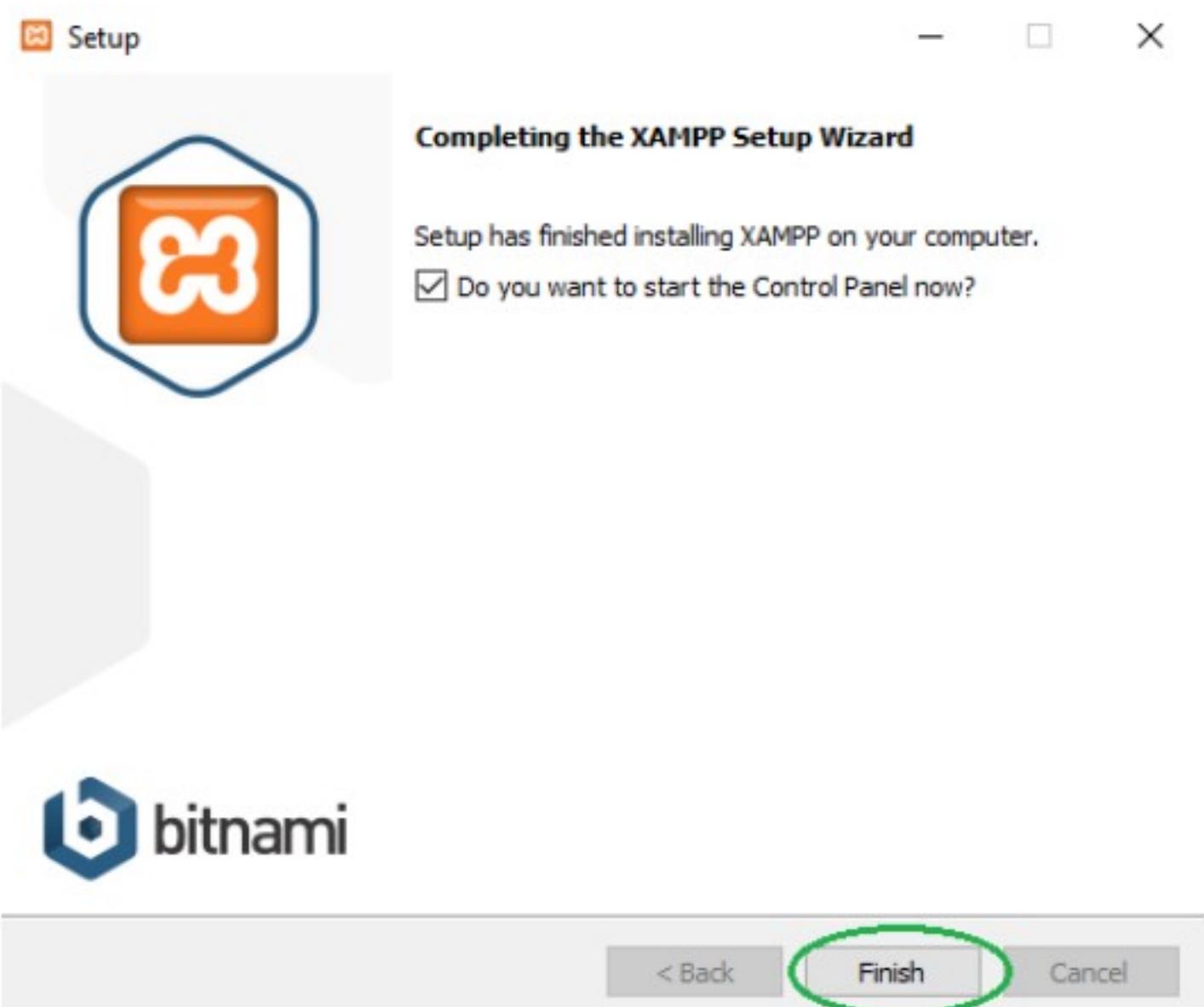
- **Step-7: Ready to Install**

હવે સેટઅપ XAMPP ઇન્સ્ટોલ કરવા માટે ready છે. ઇન્સ્ટોલેશન પ્રક્રિયા શરૂ કરવા Next પર ક્લિક કરો.



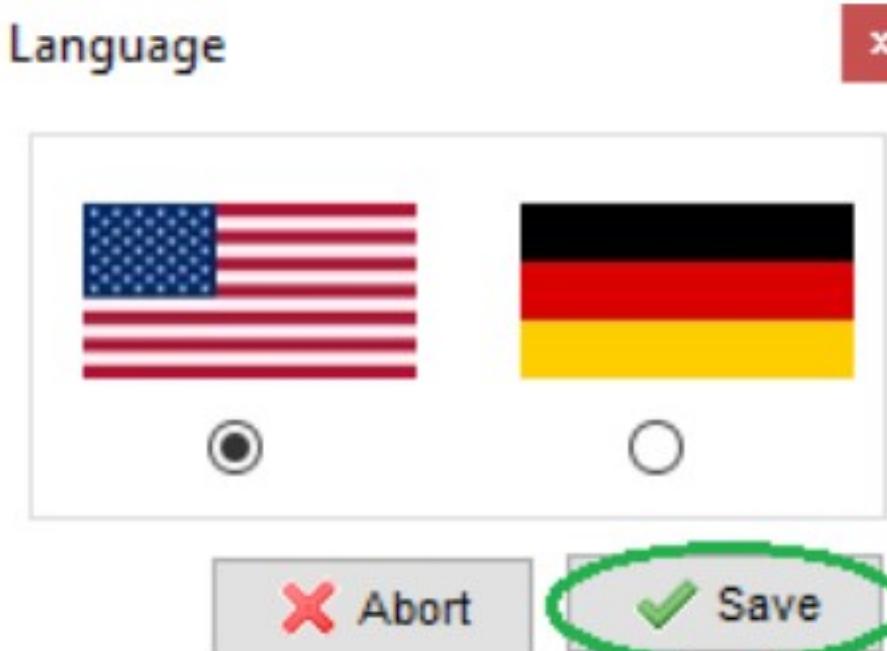
- **Step-8: Installation Complete**

હવે, તમારા કમ્પ્યુટર પર XAMPP ઇન્સ્ટોલ કરવાનું finish થઈ ગયું છે. તેથી Finish પર ક્લિક કરો.



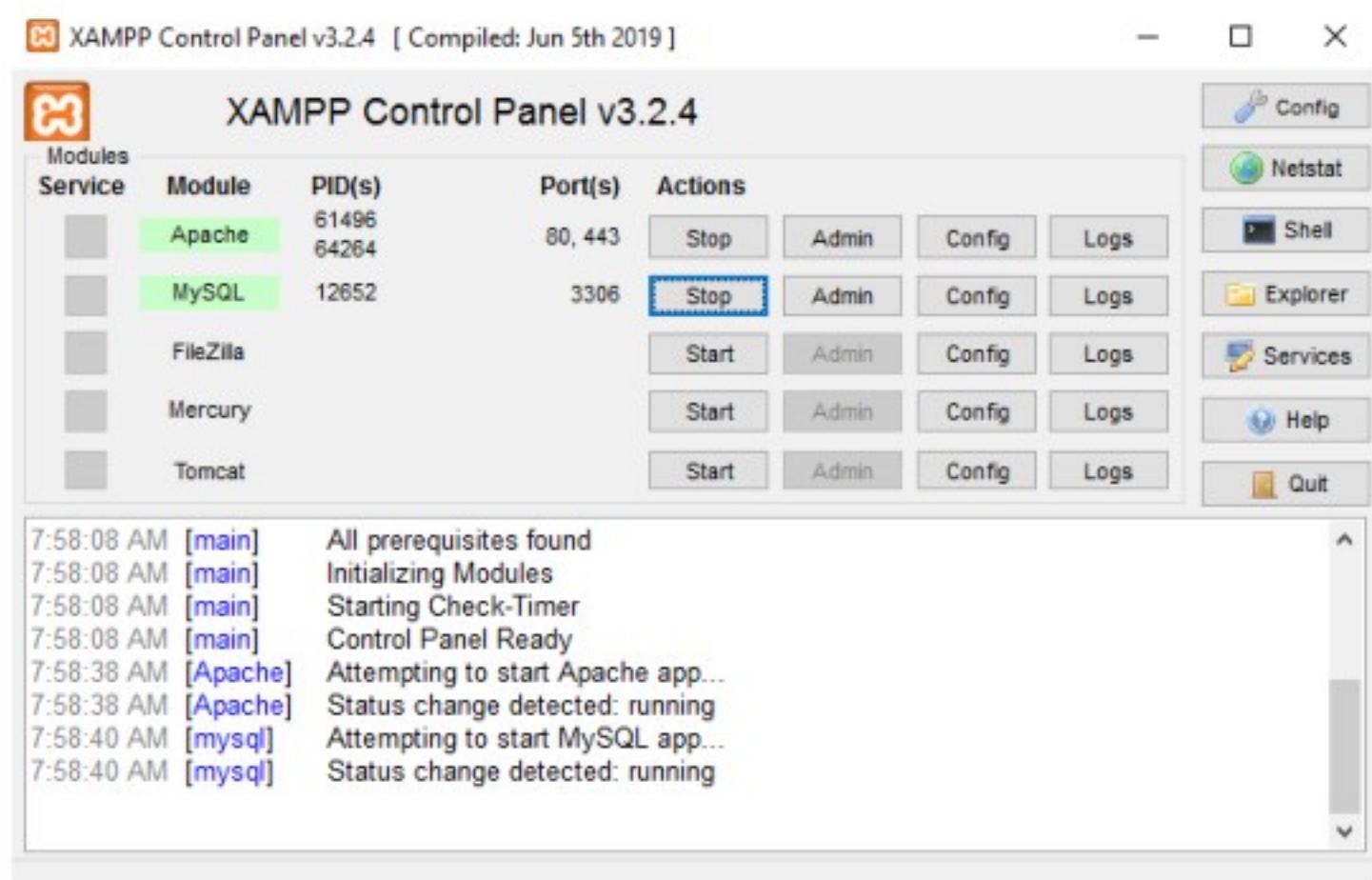
- **Step-9: Select Language**

XAMPP માં તમે જે લેંગ્વેજ વાપરવા માંગો છો તે પસંદ કરો. અંગ્રેજી માટે અમેરિકન flag ની નીચે radio બટન પર ક્લિક કરો અથવા જર્મન માટે જર્મન flag ની નીચે radio બટન પર ક્લિક કરો પછી Save પર ક્લિક કરો.



- **Step-10:** XAMPP શરૂ થાય અને તમે XAMPP કંટ્રોલ પેનલ જોશો.

અને તમે Start બટનને ક્લિક કરવા માટે Apache, MySQL અને બધી સર્વિસ શરૂ કરી શકો છો. જ્યારે સર્વિસ શરૂ થાય છે, ત્યારે તમે એક Stop બટન જોશો. આનો અર્થ એ કે service શરૂ થઈ છે.



PHP Variable and Variable Rules

- માહિતી સ્ટોર કરવા માટે વેરિયિબલ "કન્ટેનર" છે.

Declaring Variables: \$var_name=value;

- Rules for PHP variables:

- વેરિયેબલનું નામ એ \$ sign સાથે શરૂ થાય છે.
 - વેરિયેબલનું નામ એક letter અથવા અંડરસ્કોરથી શરૂ થવું આવશ્યક છે.
 - વેરિયેબલનું નામ કોઈ નંબરથી શરૂ થઈ શકતું નથી.
 - વેરિયેબલ નામમાં ફક્ત alpha-numeric character અને અંડરસ્કોર (A-z, 0-9, અને _) શામેલ હોઈ શકે છે.
 - વેરિયેબલ નામ case-sensitive છે (\$age અને \$AGE બે જુદા જુદા વેરિયેબલ છે).

Example:

```
$txt = "Hello world!"; // વેરિયેબલ $txt એ value Hello world! રાખશે.
```

```
$x = 5; // વેરિયેબલ $x એ value 5 રાખશે.
```

```
$y = 10.5; // વેરિયેબલ $y એ value 10.5 રાખશે.
```

Data Types

- PHP એ ડેટા ટાઇપ ને સપોર્ટ કરે છે.

Scalar data

- Scalar data નો અર્થ એ છે કે તે single વેલ્યુ ને શામેલ કરે છે.

Scalar Data Type	Description	Example
Integer	બધા number	15
Float	Floating - point number	8.23
String	Characters ની series	"Hello, world!"
Boolean	True અથવા false બતાવે છે.	TRUE

Compound data

- Compound data એ એવો ડેટા છે જેમાં એક કરતાં વધારે વેલ્યુ હોય શકે છે.

Compound Data Type	Description
Array	An ordered map (વેલ્યુ પર map કરેલ નામ અથવા numbers શામેલ હોય છે.)
Object	એક ટાઇપ કે જેમાં પ્રોપર્ટીસ અને મેથડ હોય છે.

Special data

- Finally, PHP, બે special ડેટા ટાઇપ ને સપોર્ટ કરે છે, કારણ કે તેમાં scalar અથવા compound ડેટા શામેલ નથી, પરંતુ એક specific meaning ધરાવે છે:

Special Data Type	Description
Resource or database	External resource નો reference શામેલ છે જેમ કે ફાઇલ
Null	વેલ્યુ તરીકે ફક્ત null શામેલ હોઈ શકે છે, જેનો અર્થ છે વેરીએબલ માં સ્પષ્ટ રૂપે કોઈપણ વેલ્યુ શામેલ હોતી નથી

Why PHP is known as loosely typed language?

- PHP, તેના વેલ્યુ ને આધારે, વેરીએબલ ને આપમેળે યોગ્ય ડેટા ટાઇપ ફેરવે છે.
- C, C++, અને JAVA જેવી અન્ય languages માં, પ્રોગ્રામરે તેનો ઉપયોગ કરતા પહેલા વેરીએબલ નું નામ અને ટાઇપ declare કરવું જરૂરી છે.

Changing Types (casting)

gettype

- આ ફંક્શન નો ઉપયોગ વેરિયેબલનાં ટાઇપ ને મેળવવા માટે થાય છે જેનો મતલબ છે કે આ ફંક્શનએ PHP, વેરિયેબલનાં ટાઇપને return કરશે.
- Syntax:** gettype(arg)
- ઉપરોક્ત ફંક્શન arg ના ટાઇપને represent કરતી string આપશે: integer, float, string, array, અને object, NULL અથવા unknown type.

Example:

```
<?php
    $a=10;
    echo gettype($a);
    $b="rahul";
    echo gettype($b);
?>
```

Output:-

integer string

settype

- The settype() ફંક્શન નો ઉપયોગ વેરિયેબલના ટાઇપને સેટ કરવા માટે થાય છે.
- Syntax:** settype(var_name, var_type)
 - var_name:** વેરિયેબલએ convert છે.
 - var_type:** વેરિયેબલના ટાઇપ. Possible વેલ્યુ: boolean, integer, float, string, array, object, null.

Example:

```
$a=65;  
settype($a, "string");
```

casting

- જો તમે temporary ટાઇપ ને બદલવા માંગો છો, એટલે કે, expression ની અંદર ઉપયોગ કરવા માંગો છો, તો તમે type casting નો ઉપયોગ કરી શકો છો.
- Syntax:** (type)\$variable
- એક **type** નો વેરિયેબલ છે કે જ્યાં તમે cast કરવા માંગો છો.

Example:

```
$var1= 10; //integer  
$var2= (string) $var1; // $var1 એ હવે 10 (string) પર set છે.
```

PHP Operators

- Operator નો ઉપયોગ વેરિયેબલ અને વેલ્યુ પર ઓપરેશન કરવા માટે થાય છે.

Arithmetic Operators

- PHP arithmetic operator નો ઉપયોગ common arithmetical operations કરવા માટે numeric values સાથે કરવામાં આવે છે, જેમ કે addition, subtraction, multiplication વગેરે.

Operator	Name	Example	Result
+	Addition	\$x + \$y	\$x અને \$y નો Sum
-	Subtraction	\$x - \$y	\$x અને \$y નો તફાવત
*	Multiplication	\$x * \$y	\$x અને \$y ની Product
/	Division	\$x / \$y	\$x અને \$y નું Quotient
%	Modulus	\$x % \$y	\$x ની રિમાંડર \$y દ્વારા ડીવાઈડ
**	Exponentiation	\$x ** \$y	\$x થી \$y ના પાવર પર વધારવાનું રિઝલ્ટ

Logical Operators

- PHP, logical operator નો ઉપયોગ conditional statements ને combine કરવા માટે થાય છે.

Operator	Name	Example	Result
and, &&	And	\$x અને \$y	જો \$x અને \$y બંને સાચા હોય તો True
or,	Or	\$x અથવા \$y	જો \$x અથવા \$y બંને માથી એક સાચું હોય તો True
!	Not	!\$x	\$x સાચું ના હોય તો True

Assignment Operators

- PHP એ assignment operator ને વેરિયેબલ ની વેચ્યુ લખવા માટે numeric values સાથે ઉપયોગમાં લેવાય છે.
- PHP માં basic assignment operator "=" છે. તેનો અર્થ એ છે કે left operand એ right side ના assignment expression ની વેચ્યુ પર સેટ છે.

Assignment	Same as	Description
$x = y$	$x = y$	Left operand એ right side ની expression ની વેચ્યુ પર સેટ છે.
$x += y$	$x = x + y$	Addition
$x -= y$	$x = x - y$	Subtraction
$x *= y$	$x = x * y$	Multiplication
$x /= y$	$x = x / y$	Division
$x %= y$	$x = x \% y$	Modulus

String Operators

- PHP માં બે operator છે જે સ્પેશિયલી strings માટે ડિઝાઇન કરેલ છે.

Operator	Name	Example	Result
.	Concatenation	$\$txt1 . \$txt2$	$\$txt1$ અને $\$txt2$ નું concatenation
.=	Concatenation assignment	$\$txt1 .= \$txt2$	$\$txt2$ થી $\$txt1$ appends

Increment / Decrement Operators

- PHP, increment operator નો ઉપયોગ વેરિયેબલ ની વેચ્યુ વધારવા માટે થાય છે.
- PHP, decrement operators નો ઉપયોગ વેરિયેબલ ની વેચ્યુ ઘટાડવા માટે થાય છે.

Operator	Name	Description
$++\$x$	Pre-increment	એક દ્વારા $\$x$ increments, પછી $\$x$ returns કરે છે.
$\$x++$	Post-increment	$\$x$ returns કરે છે પછી એક દ્વારા $\$x$ increments થાય.
$--\$x$	Pre-decrement	એક દ્વારા $\$x$ decrements, પછી $\$x$ returns કરે છે.
$\$x--$	Post-decrement	$\$x$ returns કરે છે પછી એક દ્વારા $\$x$ decrements થાય.

Conditional Operator

- Conditional operator તરીકે ઓળખાતું એક વધુ operator છે. તે પહેલા true અથવા false વેચ્યુ માટે expression ને evaluate કરે છે અને પછી તે evaluation ના રિઝલ્ટને આધારે આપેલ statements માંથી એકને execute કરે છે. conditional operator પાસે આ syntax છે:

Operator	Description	Example
$? :$	Conditional Expression	જો Condition true હોય ? તો value X : બાકી value Y

Example:

```
<?php
    $number = 21;
    $result= ($number % 2 == 0) ? "Number is Even" : "Number is Odd";
    echo $result;
?>
```

Output:

Number is Odd

Comparison Operators

- PHP એ comparison operator નો ઉપયોગ બે વેલ્યુ (number અથવા string) ની comparison કરવા માટે કરે છે:

Operator	Name	Example	Result
<code>==</code>	Equal	<code>\$x == \$y</code>	જો \$x અને \$y સરખા હોય તો true returns કરે છે.
<code>== =</code>	Identical	<code>\$x == = \$y</code>	જો \$x અને \$y બરાબર હોય, અને તે સરખી ટાઇપનાં હોય તો true returns કરે છે.
<code>!=</code>	Not equal	<code>\$x != \$y</code>	જો \$x અને \$y સરખા ન હોય તો true returns કરે છે.
<code><></code>	Not equal	<code>\$x <> \$y</code>	જો \$x અને \$y સરખા ન હોય તો true returns કરે છે.
<code>!==</code>	Not identical	<code>\$x !== \$y</code>	જો \$x અને \$y સરખા નથી, અથવા તે સરખી ટાઇપનાં નથી, તો તે true return કરે છે.
<code>></code>	Greater than	<code>\$x > \$y</code>	જો \$x એ \$y કરતાં વધુ હોય તો true return કરે છે.
<code><</code>	Less than	<code>\$x < \$y</code>	જો \$x એ \$y કરતાં ઓછું હોય તો true return કરે છે.
<code>>=</code>	Greater than or equal to	<code>\$x >= \$y</code>	\$x એ \$y કરતાં વધુ અથવા તેના બરાબર હોય તો true return કરે છે.
<code><=</code>	Less than or equal to	<code>\$x <= \$y</code>	\$x એ \$y કરતાં ઓછા અથવા તેના બરાબર છે તો true return કરે છે.

Bitwise Operators

- Bitwise operator એ integer માં યોક્કસ bits ના evaluation અને manipulation ને allow કરે છે.

Operator	Name	Example	Result
<code>&</code>	And	<code>\$x & \$y</code>	\$x અને \$y બંનેમાં સેટ થયેલ bits set છે.
<code> </code>	Or	<code>\$x \$y</code>	Bits જે \$x અથવા \$y માં સેટ છે તે set છે.

<code>^</code>	Xor	<code>\$x ^ \$y</code>	Bits કે જે \$x અથવા \$y માં સેટ છે પરંતુ બંને સેટ કરેલ નથી.
<code>~</code>	Not	<code>~\$x</code>	\$x માં સેટ થયેલ bits સેટ નથી, અને ઉલટું.
<code><<</code>	Shift left	<code>\$x << \$y</code>	Left બાજુ \$x \$y steps નાલ bits ને shift કરો.
<code>>></code>	Shift right	<code>\$x >> \$y</code>	Right બાજુ \$x \$y steps નાલ bits ને shift કરો.

Precedence of PHP Operators

- Operator precedence એ expression માં terms ના grouping ને determine કરે છે.
- આ expression evaluate કેવી રીતે અસર કરે છે.
- ઉદાહરણ તરીકે `x = 7 + 3 * 2;` અહીં x ને 13 assign કરેલ છે, 20 નથી કારણ કે operator * કરતા higher precedence ધરાવે છે તેથી તે પહેલા 3 * 2 સાથે ગુણાકાર થાય છે અને પછી 7 માં ઉમેરે છે.
- Higher precedence ઓપરેટરોનું પેહલા evaluated કરવામાં આવશે.

Category	Operator	Associativity
Unary	<code>! ++ --</code>	Right થી left
Multiplicative	<code>* / %</code>	Left થી right
Additive	<code>+ -</code>	Left થી right
Relational	<code>< <= > >=</code>	Left થી right
Equality	<code>== !=</code>	Left થી right
Logical AND	<code>&&</code>	Left થી right
Logical OR	<code> </code>	Left થી right
Conditional	<code>?:</code>	Right થી left
Assignment	<code>= += -= *= /= %=</code>	Right થી left

PHP Constants

- એક સિમ્પલ વેલ્યુ માટે constant identifier (name) છે. Script દરમિયાન વેલ્યુ બદલી શકાતી નથી.
- Valid constant નું નામ એક letter અથવા underscore થી શરૂ થાય છે (constant name પહેલાં \$ sign નથી).
- Constant બનાવવા માટે, define() ફંક્શનનો ઉપયોગ કરો.
- Syntax:** `define(name, value, case-insensitive)`
 - name:** Constant નું નામ specify કરે છે.
 - value:** Constant ની વેલ્યુ specify કરે છે.
 - case-insensitive:** Specify છે કે constant name case-insensitive હોવું જોઈએ. Default એ false છે.

Example 1:

- નીચેનું ઉદાહરણ **case-sensitive** નામ સાથે constant બનાવે છે.

```
<?php
    define("GREETING", " PHP is a scripting language ");
    echo GREETING;
?>
```

Output:

PHP is a scripting language

Example 2 :

- નીચેનું ઉદાહરણ કેસ-ઇન્સેન્સિટીવી નામ સાથે constant બનાવે છે.

```
<?php
    define("GREETING", " PHP is a scripting language ", true);
    echo greeting;
?>
```

Output:

PHP is a scripting language

constant() function

- constant() ફંક્શન constantની વેલ્યુ ને return કરે છે.
- Syntax:** constant(constant)
 - Constant:** Required. Constant નું name specify કરે છે.

Example:

```
<?php
    define("GREETING","Hello you! How are you today?");
    echo constant("GREETING");
?>
```

Output:

Hello you! How are you today?

PHP Magic constants (predefined constants)

- PHP, તે કોઈપણ script પર મોટી સંઘામાં predefined constants provide કરે છે.
- પાંચ magical constants છે કે જેનો ઉપયોગ તેઓ ક્યાં થાય છે તેના આધારે બદલાશે. ઉદાહરણ તરીકે, __LINE__ ની વેલ્યુ તેની script માં ઉપયોગમાં લેવાતી line પર આધારિત છે. આ special constants એ કેસ-ઇન્સેન્સિટીવી છે અને નીચે પ્રમાણે છે:
- થોડી " magical " PHP, constants નીચે આપેલા છે:

Name	Description
__LINE__	કાઇલના current લાઇન નંબર.
__FILE__	કાઇલનો full path અને filename.

__FUNCTION__	Function નામ. એ કોડ એ ફંક્શન નું નામ return કરે છે. તે declare કરવામાં આવ્યું હતું.
__CLASS__	Class નામ. એ કોડ એ class નું નામ return કરે છે. તે declare કરવામાં આવ્યું હતું.
__METHOD__	Class method નામ. એ કોડ એ મેથડ નું નામ return કરે છે. તે declare કરવામાં આવ્યું હતું.

Flow control statements

- Conditional statements જુદી-જુદી actions પર આધારિત વિવિધ conditions કરવા માટે વપરાય છે.
- PHP માં, અમારી પાસે નીચેના conditional statements છે:
 - if statement** - કોઈ specified condition true હોય તો જે અમૃક code executes થાય છે.
 - if...else statement** - કોઈ condition true હોય તો કોઈ executes થાય છે અને condition false હોય તો બીજો code executes થાય છે.
 - if...elseif....else statement** - જો first condition false છે તો test કરવા માટે નવી condition specify કરે છે.
 - switch statement** - execute કરવા માટે કોઈ ના ઘણા blocks માંથી એક સિલેક્ટ કરે છે.

if Statement

- જો statement નો ઉપયોગ કોઈ કોડ ને execute કરવા માટે થાય છે, તો કોઈ **specified condition true** છે.

Syntax:

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

Example:

```
<?php
    $number = 20;
    if ($number % 2 == 0)
    {
        echo "Number is Even";
    }
    if ($number % 2 != 0)
    {
        echo "Number is Odd";
    }
?>
```

Output:

Number is Even

if...else Statement

- જો કોઈ condition true હોય અને બીજા codeમાં condition false હોય તો કોઈ કોડ execute કરવા માટે if else statement નો ઉપયોગ કરો.

Syntax:

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

Example:

```
<?php
$number = 20;
if ($number % 2 == 0)
{
    echo "Number is Even";
}
else
{
    echo "Number is Odd";
}
?>
```

Output:

Number is Even

if...elseif....else Statement

- જો પહેલી condition ખોટી છે તો નવી condition specify કરવા માટે if....elseif...else statementનો ઉપયોગ થાય છે.

Syntax:

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

```

}
elseif (condition)
{
    code to be executed if condition is true;
}
else
{
    code to be executed if condition is false;
}

```

Example:

```

<?php
$a=10; $b=12; $c=30;
if ($a > $b && $a > $c)
{
    echo "a is max";
}
elseif ($b > $c)
{
    echo "b is max";
}
else
{
    echo "c is max";
}
?>

```

Output:

c is max

switch Statement

- Switch statement વિવિધ actions પર આધારિત વિવિધ conditions perform કરવા માટે વપરાય છે.
- Execute કરવા માટેના કોડના ધણા blocks માંથી એકને સિલેક્ટ કરવા માટે switch statement નો ઉપયોગ થાય છે.

• Syntax:

```

switch (n)
{
    case label1: code to be executed if n=label1; break;
}

```

```

case label2: code to be executed if n=label2; break;
case label3: code to be executed if n=label3; break;
...
default: code to be executed if n is different from all labels;
}

```

- તે કેવી રીતે કાર્ય કરે છે: પહેલા આપણી પાસે એક જ expression n છે (મોટે ભાગે એક વેરિયેબલ), તેનું એકવાર evaluate થાય છે. Expression ની વેચ્યુ પછી structure માં પ્રત્યેક case માટેની વેચ્યુ સાથે compared કરવામાં આવે છે.
- જો કોઈ મેય હોય, તો તે case સાથે સંકળાયેલા કોડનો બ્લોક એકઝેક્યુટ થાય છે.
- કોડને automatically next case માં ચાલતા અટકાવવા **break** ઉપયોગ કરો.
- જો કોઈ મેળ ન પણ તો **default** statement નો ઉપયોગ કરવામાં આવે છે.

Example:

```

<?php
$a=6;
switch($a)
{
    case 1: echo "Monday"; break;
    case 2: echo "Tuesday"; break;
    case 3: echo "Wednesday"; break;
    case 4: echo "Thursday"; break;
    case 5: echo "Friday"; break;
    case 6: echo "Saturday"; break;
    case 7: echo "Sunday"; break;
    default: echo "Wrong Choice";
}
?>

```

Output:

Saturday

Loops

- તમે કોડના same blockને ફરીથી રન કરવા માંગો છો. Script માં લગભગ સરખી code-lines ઓમેરવાને બદલે, આપણે આના જેવા કાર્ય કરવા માટે લૂપ નો ઉપયોગ કરી શકીએ છીએ.
- જ્યારે specify condition true હોય છે ત્યારે Loops એ કોડના block ને execute કરે છે.
- PHP માં, આપણી પાસે નીચેના looping statements છે:
 - **while** - જ્યાં સુધી specify condition true હોય ત્યાં સુધી કોડના block દ્વારા લૂપ ફરે છે.

- **do...while** - જ્યારે કોડના block દ્વારા એક વાર લૂપ ફરે છે અને પછી લૂપને જ્યાં સુધી specify condition true હોય ત્યાં સુધી repeat કરે છે.
- **for** - કોડના બ્લોક દ્વારા specify સંઘામાં લૂપ ફરે છે.
- **foreach** - એરેમાં દરેક એલિમેન્ટ માટે કોડના block દ્વારા લૂપ ફરે છે.

while Loop

- While loop એ કોડનો block executes કરે ત્યાં સુધી specify condition true છે.

- **Syntax:**

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

Example:

```
<?php
$x = 1
while($x<= 5)
{
    echo "The number is: $x <br>";
    $x++;
}
?>
```

Output:

```
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
```

do...while Loop

- do...while loop હંમેશાં એકવાર કોડના block ને execute કરશે જે, તે પછી condition ને ચેક કરશે અને લૂપ ને repeat કરશે જ્યારે specify condition true હશે.

- **Syntax:**

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

Example:

```
<?php
    $x = 1;
    do
    {
        echo "The number is: $x <br>";
        $x++;
    } while ($x <= 5);
?>
```

Output:

The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5

for Loop

- લૂપ માટે PHP, કોઈ ચોક્કસ સંઘામાં કોડના બ્લોકને execute કરે છે.
- જ્યારે તમે અગાઉથી જાણો છો કે script કેટલીવાર રન થવી જોઈએ ત્યારે લૂપનો ઉપયોગ કરવામાં આવે છે.

Syntax:

```
for (init counter; test counter; increment counter)
{
    code to be executed;
}
```

Parameters:

- init counter:** લૂપ counter વેચ્યુને Initialize કરે.
- test counter:** દરેક લૂપ iteration માટે evaluated કરે છે. જો તે TRUE નું evaluates કરે છે, તો લૂપ ચાલુ રહે છે. જો તે FALSE નું evaluates કરે છે, તો લૂપ end થાય છે.
- increment counter:** લૂપ counter ની વેચ્યુ increases થાય છે.

Example:

```
<?php
    for ($x = 0; $x <= 10; $x++)
    {
        echo $x." ";
    }
?>
```

Output:

0 1 2 3 4 5 6 7 8 9 10

foreach Loop

- Fforeach loop ફક્ત એરે પર કામ કરે છે, અને એરે માં દરેક key/value pair દ્વારા લૂપ વપરાય છે.

- **Syntax:**

```
foreach ($array as $value)
```

```
{
```

```
    code to be executed;
```

```
}
```

- દરેક લૂપ iteration માટે, current એરે એલિમેન્ટ નું મૂલ્ય \$value ને assign કરવામાં આવે છે અને array pointer એક દ્વારા move કરવામાં આવે છે, જ્યાં સુધી તે છેલ્લા એરે એલિમેન્ટ સુધી પહોંચે નહીં.

Example:

```
<?php
$colors = array("red", "green", "blue", "yellow");
foreach ($colors as $value)
{
    echo "$value <br>";
}
?>
```

Output:

red
green
blue
yellow

Break and Continue

- Looping structure માંથી બહાર નીકળવા માટેનો standard way એ છે કે main test condition ખોટી હોય.
- સ્પેશિયલ કમાંડ ને break કરે છે અને continue રહે છે while, do-while અને for સહિત તમામ looping constructs માંથી optional side બહાર નીકળવા ની offer કરે છે.
- બ્રેક કમાંડ innermost loop construct માંથી બહાર નીકળી જાય છે.
- Continue command એ લૂપ ના current iteration ને skip કરે છે જેમાં તે શામેલ છે.

Example (Break): નીચેનો કોડ 0 થી 5 ના નંબરને print કરે છે

```
<?php
$i = 0;
while ($i<100)
```

```
{
    if ($i > 5)
        break;
    echo $i." ";
    $i=$i+1;
}
?>
```

Output:

0 1 2 3 4 5 6

Example (Continue): નીચેનો કોડ 5 સિવાય 0 થી 9 નંબરને print કરે છે.

```
<?php
for ($i = 0; $i<10; $i++)
{
    if ($i == 5)
    {
        continue;
    }
    echo $i." ";
}
?>
```

Output:

0 1 2 3 4 6 7 8 9

Difference between Break and Continue

લૂપના executionને terminate કરવા માટે Break નો ઉપયોગ થાય છે.	લૂપના executionને terminate કરવા માટે Continue નો ઉપયોગ થતો નથી.
તે iteration (પુનરાવર્તન)ને break કરે છે.	તે iteration ને skip કરે છે.
જ્યારે આ statementને execute કરવામાં આવે છે, ત્યારે લૂપમાંથી કંટ્રોલ આવશે અને લૂપ પછી તાત્કાલિક statement ને execute કરશે.	જ્યારે આ statementને execute કરવામાં આવે છે, ત્યારે તે લૂપમાંથી બહાર આવશે નહીં પરંતુ લૂપના next iteration પર moves/jumps કરશે.
Loop અને switch case સાથે breakનો ઉપયોગ થાય છે.	Continue ફક્ત લૂપ્સમાં જ વપરાય છે, તેનો ઉપયોગ switch case માં થતો નથી.
Example : <?php for(\$A=0; \$A<=5; \$A++) {	Example : <?php for(\$A=0; \$A<=5; \$A++) {

```
if($A==3)
{
    break;
}
echo " ",$A;
?>
```

Output : 0 1 2

```
if($A==3)
{
    continue;
}
echo " ",$A;
?>
```

Output : 0 1 2 4 5

Arrays

- Array એ **single variable** છે જે એકથી વધુ વેલ્યુ ને એક સા�ે રાખી શકે છે.
- તમે વેલ્યુ ના લિસ્ટ તરીકે array ને લઈ શકો છો.
- Array માં દરેક વેલ્યુ ને એક એલિમેન્ટ કહેવામાં આવે છે, અને દરેક એલિમેન્ટને તેની **પોતાની index** હોય છે, જે તે array માટે unique છે.
- એલિમેન્ટની વેલ્યુને access કરવા માટે તમે તે એલિમેન્ટ ની index નો ઉપયોગ કરો છો તે એલિમેન્ટને create કરવા, read કરવા, write કરવા અથવા delete કરવા ઉપયોગી છે.

Types of Array

- PHP માં, ત્રણ પ્રકારના એરે છે:
 - 1) **Indexed arrays (Numeric arrays)** - Numeric index સાથે array.
 - 2) **Associative arrays** – Name અને keys સાથે array.
 - 3) **Multidimensional arrays (Nested arrays)** - Arrays માં એક અથવા વધુ એરેઝ છે.

1) Indexed arrays (Numeric arrays)

- Numeric array માં દરેક element સાથે numeric key હોય છે જે 0 થી શરૂ થાય છે.

Creating Arrays

- તમે એરે બનાવવા માટે **array()** ફંક્શનનો ઉપયોગ કરી શકો છો.

Syntax: \$array_name=array (value1, value2 ...valueN);
- Indexed arrays બનાવવા માટે ના બે ways છે.
- Index આપમેણે assign થઈ શકે છે (index હંમેશા 0 થી શરૂ થાય છે), આની જેમ:

```
$cars = array("Volvo", "BMW", "Toyota");
```
- Index ને manually assign કરી શકાય છે:

```
$cars[0] = "Volvo";
$cars[1] = "BMW";
$cars[2] = "Toyota";
```

Functions of array:

Length of an Array

- **count()** ફંક્શનનો ઉપયોગ એરે ની length ને (એલિમેન્ટ ની સંખ્યા) return કરવા માટે થાય છે.

Example:

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

Output:

3

Accessing Array Elements

- `print_r()` ફંક્શનનો ઉપયોગ વેરીએબ્લ ની માહિતી ને print કરવા માટે થાય છે.

Example:

```
<?php
    $myarray=array("A","B","C");
    print_r($myarray);
?>
```

Output:

Array ([0] => A [1] => B [2] => C)

- તમે નીચે દર્શાવેલી key અને value નો ઉપયોગ કરીને PHP ની script માં એરેના individual એલિમેન્ટને refer કરી શકો છો:

Example:

```
<?php
    $myarray=array("A","B","C");
    echo $myarray[1];
?>
```

Output:

B

- Numeric Array માં દરેક એલિમેન્ટ દ્વારા iterate (પુનરાવર્તિત) કરવા for, while અથવા do while loop નો ઉપયોગ કરી શકો છો કારણ કે numeric array ની key અને values સતત (consecutive) છે.

Changing Elements

- તમે index નો ઉપયોગ કરીને એલિમેન્ટની વેલ્યુ પણ બદલી શકો છો.

Example:

```
<?php
    $myarray=array("Apache", "MySQL", "PHP");
    $myarray[1]="Oracle";
    for($i=0; $i<3; $i++)
    {
        echo $myarray[$i]."<br>";
    }
?>
```

Output:

Apache

Oracle

PHP

Add Elements to Array

- **array_push()** ફુલશાન એરેના end માં એક અથવા વધુ એલિમેન્ટને insert કરે છે.
- **Syntax:** array_push(array,value1,value2...)

 - **array:** Required. Array ને specify કરે છે.
 - **value1:** Required. Add કરવા વેલ્યુ ને specify કરે છે.
 - **value2:** Optional. Add કરવા વેલ્યુ ને specify કરે છે.

Example:

```
<?php
    $myarray=array("Apache","MySQL", "PHP");
    print_r($myarray);

    echo "<br>";
    $myarray[]="Oracle";
    print_r($myarray);

    echo "<br>";
    array_push($myarray,"Java",".Net");
    print_r($myarray);
?>
```

Output:

```
Array ( [0] => Apache [1] => MySQL [2] => PHP)
Array ( [0] => Apache [1] => MySQL [2] => PHP [3] => Oracle )
Array ( [0] => Apache [1] => MySQL [2] => PHP [3] => Oracle [4] => Java [5] => .Net )
```

Removing Elements from Arrays

- **Unset ()** નો ઉપયોગ PHP માં વેરિએબલને destroy કરવા માટે થાય છે. તે single વેરિએબલ, multiple વેરિએબલ અથવા array માંથી એલિમેન્ટ ને remove કરવા માટે વાપરી શકાય છે.
- **Syntax:** unset (var1, var2....)
 - var1, var2: variable એ unset થશે.

Example:

```
<?php
    $a = "Hello world!";
    echo "The value of variable 'a' before unset: " . $a . "<br>";
    unset($a);
    echo "The value of variable 'a' after unset: " . $a;
?>
```

Output:

The value of variable 'a' before unset: Hello world!

The value of variable 'a' after unset:

- તમે **array_pop()** નો ઉપયોગ કરીને array ના છેલ્લા એલિમેંટ ને દૂર કરી શકો છો. આ તે એલિમેંટ પણ return કરશો:

Example:

```
<?php
    $a=array("red","green","blue");
    array_pop($a);
    print_r($a);
?>
```

Output:

Array ([0] => red [1] => green)

Searching Element

- **array_search()** ફંક્શન વેચ્યુ માટે array ને search કરે છે અને key returns કરે છે.
- **Syntax:** array_search(value,array,strict)
 - **value:** Required. Search માટે વેચ્યુ ને specify કરે છે.
 - **array:** Required. Search માટે એરે ને specify કરે છે.
 - **strict:** Optional. જો આ parameter TRUE પર સેટ કરેલું છે, તો આ ફંક્શન એરે માં સરખા એલિમેંટ માટે search કરશો.

Example:

```
<?php
    $a=array("A"=>"5","B"=>5);
    echo array_search(5,$a);
    echo array_search(5,$a,true);
?>
```

Output:

A

B

- **in_array()** ફંક્શન specific વેચ્યુ માટે array ને search કરે છે.
- **Syntax:** in_array(search,array,type)
 - **search:** Required. search માટે specify કરે છે
 - **array:** Required. search માટે એરે ને specify કરે છે
 - **type:** Optional. જો આ parameter TRUE પર સેટ કરેલું છે, તો in_array () ફંક્શન એરેમાં specific type માટે search કરે છે.

Example:

```
<?php
    $people=array("Peter", "Joe", "Glenn", 23);
    if (in_array("23",$people))
    {
        echo "Match found<br>";
    }
    else
    {
        echo "Match not found<br>";
    }
    if (in_array("23", $people, TRUE))
    {
        echo "Match found<br>";
    }
    else
    {
        echo "Match not found<br>";
    }
?>
```

Output:

Match found
 Match not found

Sorting Array

- **sort()** - Ascending order (ચંડતા કરું)માં એરે ને sort કરે.
- **rsort()** - Descending order (ઉત્તરતા કરું)માં એરે ને sort કરે
- **asort()** - Value મુજબ, ascending order માં associative એરે ને sort કરે.
- **ksort()** - Key મુજબ, ascending order માં associative એરે ને sort કરે.
- **arsort()** - Value મુજબ, descending order માં associative એરે ને sort કરે.
- **krsort()** – key મુજબ, descending order માં associative એરે ને sort કરે.

Example:

```
<?php
    $srtArray=array(2,8,9,5,6,3);
    for ($i=0; $i<count($srtArray); $i++)
    {
```

```

for ($j=0; $j<count($srtArray); $j++)
{
    if ($srtArray[$j] > $srtArray[$i])
    {
        $tmp = $srtArray[$i];
        $srtArray[$i] = $srtArray[$j];
        $srtArray[$j] = $tmp;
    }
}
foreach($srtArray as $item)
{
    echo $item."<br>\n";
}
?>
```

Output:

2 3 5 6 8 9

- ફુંક્શનનો ઉપયોગ કરીને એરે ને sort કરો.

Example:

```
<?php
    $srtArray=array(2,8,9,5,6,3);
    sort($srtArray);
    foreach($srtArray as $item) {
        echo $item."<br>\n";
    }
?>
```

Output:

2 3 5 6 8 9

2) Associative Array

- Associative part નો meaning એ છે કે એરે એ એલિમેન્ટ ની વેચ્યુ ને linear index order ને બદલે key values સાથે store કરે છે.
- જો તમે એરે માં એલિમેન્ટને store કરો છો, તો key સાથે association માં, તમારે પાછળથી key ની વેચ્યુ ને retrieve કરવાની જરૂર પડે છે.
- Key** એ **numeric** અથવા **string** હોઈ શકે છે.
- તમે associative array બનાવવા માટે array() ફુંક્શનનો ઉપયોગ કરી શકો છો.

- **Syntax:** \$array_name=array(key1=>value1, key2=>value2,..... keyN => valueN);

• associative arrayને create કરવાની બે રીતો છે:

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

OR

```
$age["Peter"] = "35";
```

```
$age["Ben"] = "37";
```

```
$age["Joe"] = "43";
```

- તમે તેના key value નો ઉપયોગ કરીને PHP માં એરે ના individual element ને refer કરી શકો છો.

Example:

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

Output:

Ben:37

- Associative array માં તમે **for**, **while** અથવા **do...while loop** નો ઉપયોગ કરી શકતા નથી કારણ કે Associative array ની key value એ અનુક્રમિક નહીં હોય.
- તેથી તમારે foreach loop નો ઉપયોગ કરવો પડશે.

Example:

```
<?php
    $myarray=array("Name"=>"James", "Age"=>25, "Gender"=>"Male");
    foreach($myarray as $item)
    {
        echo $item."<br>";
    }
?>
```

Output:

James

25

Male

3) Multidimensional Array

- કેટલીકવાર તમે એક કરતાં વધુ key સાથે વેલ્યુ ને સ્ટોર કરવા માંગો છો ત્યારે multidimensional array નો ઉપયોગ થાય છે.
- Multidimensional array એ એક એરે છે જેમાં એક અથવા વધુ એરે હોય છે.
- PHP, multidimensional arrays ને સમજે છે જે બે, ત્રણ, ચાર, પાંચ અથવા વધુ લેવલ ના હોય છે.

- પહેલા, નીચેની table પર જુવો:

Volvo	22	18
BMW	15	13
Saab	5	2

Example:

```
<?php
$cars = array (
    array("Volvo",22,18),
    array("BMW",15,13),
    array("Saab",5,2),
);
for ($row = 0; $row < 4; $row++)
{
    echo "<br>";
    for ($col = 0; $col < 3; $col++)
    {
        echo $cars[$row][$col]." ";
    }
}
?>
```

Output:

Volvo 22 18
BMW 15 13
Saab 5 2

User Defined Functions

- Built-in PHP ફંક્શન ઉપરાંત, આપણે આપના પોતાના ફંક્શન બનાવી શકીએ છીએ.
- ફંક્શન એ statements નો એક block છે જેનો ઉપયોગ પ્રોગ્રામમાં વારંવાર કરી શકાય છે.
- કોઈ પેજ લોડ થાય ત્યારે એક ફંક્શન તરત જ execute કરવામાં આવશે નહીં.
- જ્યારે ફંક્શન ને call કરવામાં આવશે ત્યારે execute થશે.

Create a User Defined Function

- User defined function એ "function" word થી શરૂ થાય છે.

Syntax:

```
function functionName()
```

```
{
```

Code to be executed;

}

Example:

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

Output:

Hello world!

Function arguments and returning values from function

- Arguments દ્વારા ફંક્શન માં માહિતી pass કરી શકાય છે. Argument એ વેરીએબલની જેમ જ છે.
- ફંક્શન ના નામ પછી arguments specify થયેલ છે; કૌસની અંદર comma થી separate થાય છે.

Example:

```
<?php
    function addFunction($num1, $num2)
    {
        $sum = $num1 + $num2;
        return $sum;
    }
$result=addFunction(10, 20);
echo "Addition=". $result;
?>
```

Output:

Addition=30

Default values for arguments

- તમે arguments માટે default વેલ્યુ ને specify કરી શકો છો.
- ફંક્શન માંથી argument દ્વારા કરવામાં આવે તો default નો ઉપયોગ થાય છે.
- Default વેલ્યુ એ constant expression હોવી આવશ્યક છે જેમ કે string, integer અથવા NULL.
- તમારી પાસે default વેલ્યુ સાથે multiple arguments હોઈ શકે છે. Default વેલ્યુ જમણી થી ડાખી બાજુ assign કરેલી છે.

Example:

```
<?php
    function addFunction($num1, $num2=5)
```

```

{
    $sum = $num1 + $num2;
    return $sum;
}
$result=addFunction(10,20);
echo "Addition=". $result. "<br>";
$result=addFunction(10);
echo "Addition=". $result;

?>
```

Output:

Addition=30
Addition=15

Call-by-Value (Pass-by-Value)

- PHP मां user-defined फँक्शन माटे default behavior call-by-value છે.
- Call by value નो અર्थ એ છે કે ફँક्शન મां directly વેલ્યુ પસાર કરે છે. Call કરેલ ફँકશન એ લોકલ વેરીએબલ માં વેલ્યુ નો ઉપયોગ કરે છે; તેમાં કોઈપણ changes કરવામાં આવે તે source variable ને અસર કરતું નથી.

Example:

```
<?php
    function swap($num1, $num2)
    {
        $temp = $num1;
        $num1=$num2;
        $num2=$temp;
    }
    $num1=10;
    $num2=20;
    echo "Before Swap."<br>;
    echo "num1=". $num1. "<br>";
    echo "num2=". $num2. "<br>";
    swap($num1,$num2);
    echo "After Swap."<br>;
    echo "num1=". $num1. "<br>";
    echo "num2=". $num2. "<br>";

?>
```

Output:

Before Swap

num1=10

num2=20

After Swap

num1=10

num2=20

Call-by-Reference (Pass-by-Reference)

- Call by reference નો અર્થ એ છે કે જ્યાં actual value store થાય છે ત્યાં વેરીએબલ નું એડ્સ્પેસ pass કરવું.
- Call function એ pass કરેલા એડ્સ્પેસ માં store કરેલી વેલ્યુનો ઉપયોગ કરે છે; તેમાં કોઈપણ **changes** કરવામાં આવે તે source variable ને **affect** કરે છે.

Example:

```
<?php
    function swap(&$num1, &$num2)
    {
        $temp = $num1;
        $num1=$num2;
        $num2=$temp;
    }
    $num1=10;
    $num2=20;
    echo "Before Swap."<br>;
    echo "num1=". $num1.<br>;
    echo "num2=". $num2.<br>;
    swap($num1,$num2);
    echo "After Swap."<br>;
    echo "num1=". $num1.<br>;
    echo "num2=". $num2.<br>;
?>
```

Output:

Before Swap

num1=10

num2=20

After Swap

num1=20

num2=10

Variables Scope

- PHP script मां કોઈપણ જગ્યાએ variables declare કરી શકાય છે.
- Variable નો scope એ scriptનો એક ભાગ છે જ્યાં વેરીએબલ ને referenced/used કરી શકાય છે.
- PHP માં ત્રણ અલગ અલગ variable scopes છે:
 - local
 - global
 - static

Local

- ફંક્શન ની અંદર declare કરાયેલ વેરીએબલ માં local scope હોય છે અને તે ફંક્શન માં જ �access થઈ શકે છે.

Example:

```
<?php
    function myTest()
    {
        $x = 5; // local scope
        echo "Variable x inside function is: $x";
    }
    myTest();
?>
```

Output:

Variable x inside function is: 5

Global

- ફંક્શનની બહાર declare કરાયેલ વેરીએબલ એ global scope ધરાવે છે અને ફંક્શનની બહાર જ તેને access કરી શકાય છે.
- Global keyword નો ઉપયોગ ફંક્શન માંથી global variable ને access કરવા માટે થાય છે.
- આ કરવા માટે, વેરીએબલ (ફંક્શનની અંદર) પહેલા global keyword નો ઉપયોગ કરો: global \$x;

Example:

```
<?php
    $x = 5; // global scope
    function myTest()
    {
    }
    myTest();
    echo "<p>Variable x outside function is: $x</p>";
```

?>

Output:

Variable x outside function is: 5

Static

- સામાન્ય રીતે, જ્યારે કોઈ ફંક્શન completed/executed થાય છે, તેના બધા વેરીએબલ ડિલીટ કરી નાખવામાં આવે છે.
- જો કે, ક્યારેક આપણે વેરીએબલ ની જરૂર પણ છે એટલે આપણે લોકલ વેરીએબલ ને ડિલીટ કરવા માગતા નથી
- આ કરવા માટે, જ્યારે તમે પહેલા વેરીએબલ declare કરો છો ત્યારે **static** કીવર્ડ નો ઉપયોગ કરો:

Example:

```
<?php
    function myTest()
    {
        static $x = 0;
        echo $x;
        $x++;
    }
    myTest();
    echo "<br>";
    myTest();
    echo "<br>";
    myTest();
?>
```

Output:

0
1
2

- ત્યારબાદ, ફંક્શન કોલ થાય ત્યારે દરેક ટાઇમએ વેરીએબલ માં છેલ્લી વખત ફંક્શન કોલ થયેલ હોય તે માહિતી સ્ટોર થયેલી હોય છે.

String Function

chr

- Specify ASCII વેલ્યુ માંથી કોઈ character return કરે છે.
- Syntax: chr(ascii)**

Example:

```
<?php
    echo chr(52) . "<br>"; // Decimal value
    echo chr(052) . "<br>"; // Octal value
    echo chr(0x52) ; // Hex value
?>
```

Output:

```
4
*
R
```

ord

- String ની first character ની ASCII વેલ્યુ આપે છે.
- Syntax: ord(string)**

Example:

```
<?php
    echo ord("A")."<br />";
    echo ord("And");
?>
```

Output:

```
65
65
```

strtolower

- String ને Lowercase latters માં convert કરે છે.
- Syntax: strtolower(string)**

Example:

```
<?php
    echo strtolower("HELLO");
?>
```

Output:

```
hello
```

strtoupper

- String ને uppercase latters માં convert કરે છે.
- Syntax: strtoupper(string)**

Example:

```
<?php
    echo strtoupper ("hello");
?>
```

Output:

HELLO

strlen

- string ની length return કરે છે.
- **Syntax: strlen(string)**

Example:

```
<?php
    echo strlen("hello hi");
?>
```

Output:

8

ltrim

- Whitespace અથવા બીજા predefined character ને સ્ટ્રિંગની ડાય્બી બાજુ એ થી દૂર કરે છે.
- **Syntax: ltrim(string,charlist)**
 - **charlist**:- Optional. સ્ટ્રિંગ માથી જે characters ને દૂર કરવાના હોય તે specify કરે છે.
 - આપેલા બધા અક્ષરો દૂર કરવામાં આવે છે: "\0" – NULL, "\t" – tab, "\n" – new line, "\r" – carriage return, " " – white space.

Example:

```
<?php
    $str="Hello";
    echo ltrim($str,"H")."<br>";
    $str="  Hello";
    echo ltrim($str);
?>
```

Output:

ello

Hello

rtrim

- Whitespace અથવા બીજા predefined character ને સ્ટ્રીંગની જમણી બાજુ એ થી દૂર કરે છે.
- **Syntax: rtrim(string,charlist)**
 - **charlist**:- Optional. સ્ટ્રીંગ માથી જે characters ને દૂર કરવાના હોય તે specify કરે છે.

- આપેલા બધા અક્ષરો દૂર કરવામાં આવે છે: "\0" – NULL, "\t" – tab, "\n" – new line, "\r" – carriage return, " " – white space.

Example:

```
<?php
    $str="Hello";
    echo rtrim($str,"o")."<br>";
    $str="Hello  ";
    echo rtrim($str);
?
>
```

Output:

Hell
HELLO

trim

- Whitespace અથવા બીજી predefined character ને સ્ટ્રિંગની બંને બાજુ એ થી દૂર કરે છે.
- **Syntax: trim(string,charlist)**
 - **charlist**: - Optional. સ્ટ્રિંગ માથી જે characters ને દૂર કરવાના હોય તે specify કરે છે.
 - આપેલા બધા અક્ષરો દૂર કરવામાં આવે છે: "\0" – NULL, "\t" – tab, "\n" – new line, "\r" – carriage return, " " – white space.

Example:

```
<?php
    $str="oHello";
    echo trim($str,"o")."<br>";
    $str="  Hello  ";
    echo trim($str);
?
>
```

Output:

Hell
Hello

substr

- String નો ભાગ રૂપીત કરે છે.
- **Syntax: substr(string,start,length)**
 - **start**: Required. સ્ટ્રિંગ માં ક્યાંથી શરૂ કરવું તે specify કરે છે.
 - A positive number - સ્ટ્રિંગમાં કોઈ specify કરેલ position થી શરૂ કરો.
 - A negative number - સ્ટ્રિંગ ના end થી કોઈ specify કરેલ position પરથી શરૂ કરો.
 - 0 - સ્ટ્રિંગ ના પહેલા character થી શરૂ કરો.

- **length:** Optional. Return થયેલ સ્ટ્રિંગ ની length specify કરે છે.
 - Default એ સ્ટ્રિંગ નો end છે.
 - A positive number – શરૂઆતના parameter થી length return કરે છે.
 - Negative number – સ્ટ્રિંગ ના end માથી length return કરે છે.

Example:

```
<?php
// Positive numbers:
echo substr("Hello world",0,10)."<br>";
echo substr("Hello world",1,8)."<br>";
echo substr("Hello world",0,5)."<br>";
echo substr("Hello world",6,6)."<br>";
echo "<br>";

// Negative numbers:
echo substr("Hello world",0,-1)."<br>";
echo substr("Hello world",-10,-2)."<br>";
echo substr("Hello world",0,-6)."<br>";

?>
```

Output:

```
Hello worl
ello wor
Hello
world
```

```
Hello worl
ello wor
Hello
```

strcmp

- બે સ્ટ્રિંગની સરખામણી કરે છે. (**case-sensitive**)
- strcmp ફંકશન 0 આપશે – જો બે સ્ટ્રિંગ equal હોય ત્યારે, negative – જો સ્ટ્રિંગ 1 એ સ્ટ્રિંગ 2 કરતાં નાની હોય ત્યારે અને positive – જો સ્ટ્રિંગ 1 એ સ્ટ્રિંગ 2 કરતાં મોટી હોય ત્યારે.
- **Syntax: strcmp(string1,string2)**

Example:

```
<?php
echo strcmp("Hello world!","Hello world!")."<br>"; // the two strings are equal
```

```

echo strcmp("Hello world!","Hello")."<br>"; // string1 is greater than string2
echo strcmp("Hello world!","Hello world! Hello!"); // string1 is less than string2
?>
Output:
0
7
-7

```

strcasecmp

- બે સ્ટ્રિંગની સરખામણી કરે છે. (**case-insensitive**)
- strcasecmp ફંક્શન 0 આપશે - જો બે સ્ટ્રિંગ એક હોય ત્યારે, negative - જો સ્ટ્રિંગ 1 એ સ્ટ્રિંગ 2 કરતાં નાની હોય ત્યારે અને positive - જો સ્ટ્રિંગ 1 એ સ્ટ્રિંગ 2 કરતાં મોટી હોય ત્યારે.
- Syntax: strcasecmp(string1,string2)**

Example:

```

<?php
    echo strcasecmp("Hello world!","HELLO WORLD!")."<br>"; // The two strings are equal
    echo strcasecmp("Hello world!","HELLO")."<br>"; // String1 is greater than string2
    echo strcasecmp("Hello world!","HELLO WORLD! HELLO!"); // String1 is less than string2
?>

```

Output:

```

0
7
-7

```

strpos

- બીજી સ્ટ્રિંગ ની અંદરની સ્ટ્રિંગની **first** occurrence ની position return કરે છે. (**case-sensitive**).
- જો સ્ટ્રિંગ ના મળે, તો આ ફંક્શન FALSE આપે છે.
- Syntax:- strpos(string,find,start)**
 - String:** Required. Find કરવા માટે સ્ટ્રિંગ સ્પેચિફિય કરે છે.
 - Find:** Optional. Find કરવું તે સ્પેચિફિય કરે છે.
 - Start:** Optional. Find કરવું તે સ્પેચિફિય કરે છે.

Example:

```

<?php
    echo strpos("hello","e");
    echo strpos("hello","l");
    echo strpos("hellohello","l",5);
?>

```

Output:

1
2
7

strrpos

- બીજુ સ્ટ્રિંગ ની અંદરની સ્ટ્રિંગની **last** occurrence ની position return કરે છે. (**case-sensitive**)
- જો સ્ટ્રિંગ ના મળે, તો આ ફંક્શન FALSE આપે છે.

Syntax:- **strrpos(string,find,start)**

- String:** Required. Find કરવા માટે સ્ટ્રિંગ સ્પેચિફિય કરે છે.
- Find:** Optional. Find ક્યાં કરવું તે સ્પેચિફિય કરે છે.
- Start:** Optional. Find ક્યાંથી કરવું તે સ્પેચિફિય કરે છે.

Example:

```
<?php
    echo strpos("hello","l");
    echo strpos("hellohello","l", 5);
?>
```

Output:

3
8

stripos()

- બીજુ સ્ટ્રિંગ ની અંદરની સ્ટ્રિંગની **first** occurrence ની position return કરે છે. (**case-insensitive**).
- જો સ્ટ્રિંગ ના મળે, તો આ ફંક્શન FALSE આપે છે.

Syntax:- **stripos(string,find,start)**

- String:** Required. Find કરવા માટે સ્ટ્રિંગ સ્પેચિફિય કરે છે.
- Find:** Optional. Find ક્યાં કરવું તે સ્પેચિફિય કરે છે.
- Start:** Optional. Find ક્યાંથી કરવું તે સ્પેચિફિય કરે છે.

Example:

```
<?php
    echo stripos("hello","L");
    echo stripos("hellohello", "L",4);
?>
```

Output:

2
7

strripos()

- બીજુ સ્ટ્રિંગ ની અંદરની સ્ટ્રિંગની **last** occurrence ની position return કરે છે. (**case-insensitive**)
- જો સ્ટ્રિંગ ના મળે, તો આ ફંક્શન FALSE આપે છે.
- Syntax:- strripos(string,find,start)**
 - String:** Required. Find કરવા માટે સ્ટ્રિંગ specify કરે છે.
 - Find:** Optional. Find ક્યાં કરવું તે specify કરે છે.
 - Start:** Optional. Find ક્યાંથી કરવું તે specify કરે છે.

Example:

```
<?php
echo strripos("hello","L");
echo strripos("hellohello","L",4);
?>
```

Output:

3
8

strstr

- બીજુ સ્ટ્રિંગ ની અંદરની સ્ટ્રિંગની first occurrence ને search કરે છે. (**case-sensitive**)
- આ ફંક્શન બાકીની સ્ટ્રિંગ (matching point થી) return કરે છે, અથવા જો search માટેની string ના મળે તો false return કરે છે.
- Syntax: strstr(string,search,before_search)**
 - String:** Required. Search માટે સ્ટ્રિંગ specify કરે છે. જો આ parameter કોઈ સંખ્યા છે, તો તે નંબરની ASCII value સાથે match થતાં number ના character માટે search કરશે.
 - Search:** Required. Search માટે સ્ટ્રિંગ specify કરે છે. જો આ parameter કોઈ સંખ્યા છે, તો તે નંબરની ASCII વેલ્યુ સાથે match થતાં number ના character માટે search કરશે.
 - before_search:** Optional. ડિફોલ્ટ બુલિયન વેલ્યુ "False" છે. જો "true" પર સેટ કરેલ હોય, તો તે searchના first occurrence પહેલાના સ્ટ્રિંગના ભાગને રિટર્ન કરે છે.

Example:

```
<?php
echo strstr("Hello world!",111);
echo strstr("Hello world!","wo");
echo strstr("Hello world!","world",true);
?>
```

Output:

o world!
world!

Hello

stristr

- બીજું સ્ટ્રિંગ ની અંદરની સ્ટ્રિંગની first occurrence ને search કરે છે. (case-insensitive).
- **Syntax:** stristr(string,search,before_search)
 - **String:** Required. Search માટે સ્ટ્રિંગ specify કરે છે. જો આ parameter કોઈ સંખ્યા છે, તો તે નંબરની ASCII value સાથે match થતાં number ના character માટે search કરશે.
 - **Search:** Required. Search માટે સ્ટ્રિંગ specify કરે છે. જો આ parameter કોઈ સંખ્યા છે, તો તે નંબરની ASCII વેલ્યુ સાથે match થતાં number ના character માટે search કરશે.
 - **before_search:** Optional. ડિફોલ્ટ બુલિયન વેલ્યુ "False" છે. જો "true" પર સેટ કરેલ હોય, તો તે searchના first occurrence પહેલાના સ્ટ્રિંગના ભાગને રિટર્ન કરે છે.

Example:

```
<?php
    echo stristr("Hello world!",87);
    echo stristr("Hello world!","WO");
    echo stristr("Hello world!","WORLD",true);
?>
```

Output:

world!
world!
Hello

str_replace

- સ્ટ્રિંગ માં કેટલાક character ને બદલે છે (case-sensitive).
- **Syntax:- str_replace(find,replace,string,count)**
 - **find:** Required. Find કરવા માટે value specify કરે છે.
 - **replace:** Required. જે value ને find કરી એના બદલે નવી value replace કરે છે.
 - **string:** Required. જે સ્ટ્રિંગ માં search કરવા નું છે તે specify કરે છે.
 - **count:** Optional. એક variable જેના replacement ના count ગણાય છે.

Example:

```
<?php
    echo str_replace("world","Peter","Hello world!");
?>
```

Output:

Hello Peter!

strrev

- Reverse સ્ટ્રિંગ return કરે છે.

- **Syntax:** `strrev(string)`

Example:

```
<?php
    echo strrev("hello");
?>
```

Output:

olleh

Math Function

abs

- Number ની absolute(positive) વેલ્યુ ને return કરે છે.

- **Syntax:** `abs(x)`

Example:

```
<?php
    echo abs(-6.7)."<br>";
    echo abs(-3)."<br>";
?>
```

Output:

6.7

3

ceil

- Smallest integer વેલ્યુ ને return કરે છે જે x કરતા વધારે અથવા બરાબર છે.

- **Syntax:** `ceil(x)`

Example:

```
<?php
    echo ceil(0.60)."<br>";
    echo ceil(5)."<br>";
    echo ceil(-4.9);
?>
```

Output:

1

5

-4

floor

- Largest integer વેલ્યુ ને return કરે છે જે x કરતા ઓછી અથવા બરાબર છે.

- **Syntax:** `floor(x)`

Example:

```
<?php
    echo floor(0.60)."<br>";
    echo floor(5)."<br>";
    echo floor (4.9);
?>
```

Output:

0
5
4

round

- Number ને નજીક ની integer વેલ્યુ માં round કરી આપે છે.
- Syntax: round(x,precision)**
 - precision:** Optional. decimal point પછી digits ની number

Example:

```
<?php
    echo round(5.335,2)."<br>";
    echo round(0.49);
?>
```

Output:

5.34
0

fmod

- Arguments ની division નું remainder આપે કરે છે.
- Syntax: fmod(x,y)**

Example:

```
<?php
    echo fmod(5,2);
?>
```

Output:

1

min

- Specified numbers માથી lowest વેલ્યુ સાથે number આપે છે.
- Syntax: min(x1,x2,x3...) Or min(array_values)**

Example:

```
<?php
    echo min(-2,4,-6,8,10) . "<br>";
    echo min(array(4,6,8,10));
?>
```

Output:

-6
4

max

- Specified numbers માથી highest વેલ્યુ સાથે number આપે છે.
- Syntax: max(x1,x2,x3...) Or max(array_values)**

Example:

```
<?php
    echo max(-2,4,-6,8,10) . "<br>";
    echo max(array(4,6,8,10));
?>
```

Output:

10
10

pow

- x થી y ની power ની વેલ્યુ આપે છે.
- Syntax: pow(x,y)**

Example:

```
<?php
    echo pow(5,2);
?>
```

Output:

25

sqrt

- Number x નું square root આપે છે.
- Syntax: sqrt(x)**

Example:

```
<?php
    echo sqrt(25);
```

?>

Output:

5

rand

- Random integer return કરે છે.
 - **Syntax: rand() or rand(min,max)**
 - **Min:** Optional. Return કરવાના માટેના સૌથી lowest નંબર specify કરે છે. Default 0 છે.
 - **Max:** Optional. Return કરવાના માટેના સૌથી highest નંબર specify કરે છે. Default getrandmax() છે.

Example:

```
<?php
    echo rand()."<br>";
    echo rand(40,500);
?>
```

Output:

18269

417

Date Function

date

- local time/date માટે નું ફોર્મેટ કરે છે અને ફોર્મેટ કરેલી date સ્ટ્રિંગ આપે છે.
- **Syntax: date(format,timestamp)**
 - **format:** Required. Return કરવા માટે date અને time ના ફોર્મેટ ને specify કરે છે.
 - **d** - મહિના નો દિવસ (from 01 to 31)
 - **m** - મહિના ની numeric રજૂઆત (from 01 to 12)
 - **M** - મહિના ની ટુકું textual રજૂઆત (ત્રણ letters)
 - **Y** - વર્ષના ચાર ડિજિટની રજૂઆત
 - **y** - વર્ષના બે ડિજિટની રજૂઆત
 - **timestamp:** Unix time ફોર્મેટમાં સમયનો ઉલ્લેખ કરે છે. ડિફોલ્ટ એ current local time (time()) છે.

Example:

```
<?php
    echo date("d/m/y")."<br>";
    echo date("d M, Y");
?>
```

Output:

11/08/15

11 Aug, 2015

getdate

- એરે return કરે છે જેમાં UNIX timestamp માટે date અને time ની માહિતી શામેલ છે.
- date string ફોર્મેટ કરતી વખતે returning array માં ten elements શામેલ હોય તેને relevant માહિતી ની જીસુર છે: [seconds] – seconds, [minutes] – minutes, [hours] – hours, [mday] - day of the month, [wday] - day of the week, [year] – year, [yday] - day of the year, [weekday] - name of the weekday, [month] - name of the month.
- Syntax: getdate(timestamp)**
 - timestamp:** Optional. UNIX time ફોર્મેટ માં time specify કરો. ડિફોલ્ટ એ current local time (time()) છે.

Example:

```
<?php
    print_r (getdate());
?>
```

Output:

Array ([seconds] => 3 [minutes] => 49 [hours] => 17 [mday] => 11 [wday] => 2 [mon] => 8 [year] => 2015 [yday] => 222 [weekday] => Tuesday [month] => August [0] => 1439308143)

checkdate

- જો check કરેલી date valid હોય તો checkdate() ફંક્શન એ true return કરે છે બાકી false કરશે.
- Syntax: checkdate(month,day,year)**
 - month, day, year:** Required. મહિનો, દિવસ અને વર્ષ specify કરે.
 - date valid છે જો: મહિનો 1 થી 12 વાંચેનો હોય, દિવસ એ particular મહિના માટે માન્ય (allowed) દિવસની અંદર હોય, વર્ષ 1 થી 32767 ની વાંચે હોય.

Example:

```
<?php
    var_dump(checkdate(12,31,-400));
    echo "<br>";
    var_dump(checkdate(2,29,2003));
    echo "<br>";
    var_dump(checkdate(2,29,2004));
?>
```

Output:

```
bool(false)
bool(false)
bool(true)
```

time

- time() ફંક્શન current time ને Unix timestamp (જાન્યુઆરી 1, 1970 00:00:00 GMT ની second ની સંખ્યા) તરીકે return કરે છે.
- **Syntax: time(void)**

Example:

```
<?php
    echo time();
?>
```

Output:

1439309922

mktime

- mktime () ફંક્શન date માટે Unix timestamp return કરે છે.
- આ timestamp માં Unix Epoch (જાન્યુઆરી 1, 1970 00:00:00 GMT) અને specify time વચ્ચે seconds ની સંખ્યા શામેલ છે.
- **Syntax: mktime (hour,minute,second,month,day,year,is_dst)**
 - **hour, minute, second, month, day, year:** Optional. Specifies the hour, minute, second, month, day, year ને specify કરે છે.
 - **is_dst:** Optional. આ parameter ને 1 પર સેટ કરો જો daylight savings time (DST) દરમયાન હોય તો, જો તે ન હોય તો 0, અથવા -1 એ default છે. આ parameter PHP 7.0 માં દૂર કરવામાં આવ્યો છે.

Example:

```
<?php
    echo(date("M-d-Y",mktime(0,0,0,12,36,2001))."<br />");
    echo(date("M-d-Y",mktime(0,0,0,1,1,99))."<br />");
?>
```

Output:

Jan-05-2002
Jan-01-1999

File Function

fopen()

- fopen () ફંક્શન એ file અથવા URL open કરે છે.

- જો fopen() fail થાય છે, તો તે FALSE અને failure પર error આપે છે.
- **Syntax: fopen(filename,mode,include_path,context)**
 - **Filename: Required.** Open માટે ફાઇલ અથવા URL ને specify કરે છે.
 - **Mode: Required.** ફાઇલ / stream માટે તમારે જરૂરી type ના accessને specify કરે છે.
 - "r" (ફક્ત Read. ફાઇલની શરૂઆતમાં શરૂ થાય છે)
 - "r+" (Read/Write. ફાઇલની શરૂઆતમાં શરૂ થાય છે)
 - "w" (ફક્ત Write. ફાઇલના content ને ઓપન કરે છે અને clear કરે છે અથવા નવી ફાઇલ બનાવે છે જો તે exist ના હોય તો)
 - "w+" (Read/Write. ફાઇલના content ને ઓપન કરે છે અને clear કરે છે અથવા નવી ફાઇલ બનાવે છે જો તે exist ના હોય તો)
 - "a" (ફક્ત Write. ઓપન કરે અને ફાઇલના endમાં write કરે છે અથવા નવી ફાઇલ બનાવે છે જો તે exist ના હોય તો)
 - "a+" (Read/Write. ફાઇલના end સુધી write કરીને ફાઇલના content ને સાચવે છે)

Example:

```
<?php
    $file = fopen("test.txt","r");
    $file = fopen("/home/test/test.txt","r");
    $file = fopen("http://www.example.com/","r");
?
```

fwrite()

- fwrite () એ એક open file પર write કરે છે.
- આ ફંક્શન ફાઇલ ના end માં અથવા જ્યારે specify length સુધી પહોંચે ત્યારે જે પણ પહેલા આવે છે તે stop થઈ જશે.
- આ ફંક્શન written bytes ની સંખ્યા, અથવા failure પર FALSE return કરે છે.
- **Syntax: fwrite(file,string,length)**
 - **file:** Required. Write માટે open ફાઇલ ને specify કરે છે.
 - **string:** Required. Open ફાઇલ પર write માટે સ્ટ્રિંગ ને specify કરે છે.
 - **length:** Optional. Write માટે maximum number ના bytes ને specify કરે છે.

Example:

```
<?php
    $file = fopen("test.txt","w");
    echo fwrite($file,"Hello World. Testing!");
    fclose($file);
?
```

Output:

fread()

- fread() ઓપન ફાઇલમાથી read કરે છે.
- **Syntax: fread(file, length)**
 - **file:** Required. Read કરવા માટે ઓપન ફાઇલને specify કરે છે.
 - **length:** Required. Read કરવા માટે મોકસીમમ બાઇટ્સને specify કરે છે.

Example:

```
<?php
    $file = fopen("test.txt", "r");
    fread($file, "10");
    fclose($file);
?>
```

fclose()

- fclose() ફંક્શન open ફાઇલ ને close કરવા માટે ઉપયોગ થાય છે.
- fclose() ને ફાઇલના નામ ની જરૂર છે. (અથવા વેરીએબલ નામ કે જે filename ધરાવે છે).
- **Syntax: fclose(file)**

Example:

```
<?php
    $myfile = fopen("webdictionary.txt", "r");
    // some code to be executed....
    fclose($myfile);
?>
```

Form

- <form> ટેગ નો ઉપયોગ યુઝર ઇનપુટ માટે HTML form ને create કરવા માટે થાય છે.
- HTML forms નો ઉપયોગ યુઝર ઇનપુટ ને collect કરવા માટે થાય છે.
- Form માં યુઝર ઇનપુટ જેવા text fields, checkboxes, radio button, submit buttons અને બીજા પણ હોઈ શકે છે. Forms નો ઉપયોગ સર્વર પર ડેટા pass કરવા માટે થાય છે.
- **Syntax:**
<form parameters> ...form elements... </form>

Attributes of form

Action

- જ્યારે form સબમિટ કરવામાં આવે ત્યારે form-data ને ક્યાં મોકલવાનો છે તે action attribute specify કરે છે.
- **Syntax:** <form action="URL">
- URL માટે ની specify વેલ્યુ:
 - **An absolute URL** – બીજી વેબસાઇટ નું URL મૂકી શકો છો (like action = "http://www.example.com/example.htm")
 - **A relative URL** – વેબસાઇટ ની અંદર ની ફાઇલ ને specify કરે છે. (like action="example.php")

Enctype

- Enctype attribute એ સર્વર પર સબમિટ કરતી વખતે form ડેટા ને કેવી રીતે encoded કરવું તે specify કરે છે.
- **Syntax:** <form enctype="value">
- **Note:** enctype attribute ફક્ત ત્યારે જ ઉપયોગ કરી શકાય છે જો method= "POST" હોય

Method

- Method attribute એ form data કેવી રીતે મોકલવો તે specify કરે છે. (action attribute માં જે પેજ specify કરેલ છે ત્યાં form-data મોકલવો).
- **Syntax:** <form method="get|post">

Value	Description
Get	Default. Name/value જોડીમાં URL પર form-data append કરે છે: URL?name=value&name=value
Post	HTTP post transaction તરીકે ફોર્મ-ડેટા મોકલે છે.

Name

- Name attribute એ ફોર્મ ના નામ ને specify કરે છે.
- Name attribute નો ઉપયોગ JavaScript માં reference એલિમેન્ટમાં અથવા ફોર્મ સબમિટ કર્યું પછી reference તરીકે થાય છે.
- **Syntax:** <form name = "text">

- o **text** – form ના નામ ને specify કરે છે.

Target

- Target attribute એ નામ અથવા કીવર્ડ specify કરે છે કે form સબમિટ કર્યા પછી receive કરેલ response ને ક્યાં ડિસ્પ્લે કરવો.
- **Syntax:** <form target="_blank|_self|_parent|_top|framename">

Value	Description
_blank	Response એ નવી વિન્ડો અથવા નવી ટેબ માં ડિસ્પ્લે થાય છે.
_self	Response એ same frame (આ default છે) માં ડિસ્પ્લે થાય છે.
_parent	Response એ parent frame માં ડિસ્પ્લે થાય છે.
_top	Response એ વિન્ડો ના full body માં ડિસ્પ્લે થાય છે.

HTML Form Controls

Text Box

- <input type="text"> એ text ઇનપુટ માટે એક લાઇન વાળી ઇનપુટ ફીલ્ડ ને define કરે છે.
- text box ના attributes:

Attribute	Description
name	આપેલ ફીલ્ડ ને name assign કરે છે જેથી તમે પછીથી તેનો reference આપી શકો.
maxlength	આ ઇનપુટ ફીલ્ડ માટે character ના maximum number ને specify કરે છે.
readonly	આ ઇનપુટ ફીલ્ડ ફક્ત read કરવા માટે છે તે specify કરે છે. (તેમાં change ના કરી શકીએ).
required	આ ઇનપુટ ફીલ્ડ એ required છે તે specify કરે છે (fill હોવું જરૂરી છે).
size	આ ઇનપુટ ફીલ્ડ ની width (character માં)ને specify કરે છે.
value	ઇનપુટ ફીલ્ડ માટે default વેલ્ચુ ને specify કરે છે.

<input type="text" name="username" size="25" maxlength="10">

Password

- <input type="password"> password ની ફીલ્ડ ને define કરે છે.
- password ના attributes:

Attribute	Description
name	આપેલ ફીલ્ડ ને name assign કરો જેથી તમે પછીથી તમે reference આપી શકો.
maxlength	ઇનપુટ ફીલ્ડ માટે character ના નંબર ને specify કરે છે.
required	આ ઇનપુટ ફીલ્ડ એ જરૂરી છે તે specify કરે છે (fill હોવું જરૂરી છે).
size	આ ઇનપુટ ફીલ્ડ ની width (character માં)ને specify કરે છે.
value	ઇનપુટ ફીલ્ડ માટે default વેલ્ચુ ને specify કરે છે.

<input type="password" name="pass" size="25" maxlength="10">

Text Area

- <**textarea**> ટેગ એ મલિટિપલ લાઇન ઇનપુટ કંટ્રોલ ને define કરે છે.
- textarea એ unlimited character ને hold કરી શકે છે.
- textarea ના અડ્યુટ્યુન્ટ્સ:

Attribute	Description
name	Textarea માટે નામ ને specify કરે છે.
cols	Textarea ની width specify કરે છે.
maxlength	Textarea માં maximum સંખ્યાને specify કરે છે.
placeholder	Short hint specify કરે છે જે text area ની expected વેલ્ચુ ને describe કરે છે.
readonly	Specify કરે છે કે કોઈ textarea ફક્ત read કરવા માટે હોવો જોઈ.
required	Specify કરે છે કે textarea ભરવું આવશ્યક છે.
rows	Textarea માં લાઇનની visible સંખ્યાનો ઉલ્લેખ કરે છે.

```
<textarea cols="25" rows="5" maxlength="50" placeholder="Please Enter Comment...">
</textarea>
```

Radio Button

- <**input type="radio"**> એ radio button define કરે છે.
- Radio buttons એ ચુઝર ને limited સંખ્યામાં ચોઇસ માથી એક સિલેક્ટ કરવા દે છે.
- Radio button ના અડ્યુટ્યુન્ટ્સ:

Attribute	Description
name	આપેલ ફીલ્ડ ને નામ assign કરે છે જેથી પછી તેનો reference આપી શકો.
value	ઇનપુટ ફીલ્ડ માટે default વેલ્ચુને specify કરે છે.
checked	Specify કરે છે radio button ચેક કરવું જોઈએ કે નહીં <ul style="list-style-type: none"> ○ true - radio button એ ચેક છે. ○ false - Default. radio button એ ચેક નથી.

```
<input type="radio" name="gender" value="male" checked> Male
```

```
<input type="radio" name="gender" value="female"> Female
```

Check box

- <**input type="checkbox"**> એ check box ને define કરે છે.
- Check boxes એ સિલેક્ટ ગ્રૂપ માટે મલિટિપલ item ને સિલેક્ટ કરવાનું allow કરે છે.
- Check box ના અડ્યુટ્યુન્ટ્સ:

Attribute	Description
name	આપેલ ફીલ્ડ ને નામ assign કરવામાં આવે છે જેથી તમે પછીથી તેનો reference આપી શકો.
value	ઇનપુટ ફીલ્ડ માટે default વેલ્ચુ specify કરે છે.
checked	Specify કરે છે checkbox ચેક કરવું જોઈએ કે નહીં

- **true** – Check box એ ચેક છે.
- **false** - Default. Check box એ ચેક નથી.

```
<input type="checkbox" name="language" value="java" checked>Java
```

```
<input type="checkbox" name="language" value=".net">.Net
```

```
<input type="checkbox" name="language" value="c++" checked>C++
```

Dropdown List

- Dropdown list નો ઉપયોગ આપેલી ચોઇસ માથી એક ઓષ્ણાન ને સિલેક્ટ કરવા માટે થાય છે.
- <select> એલિમેંટ નો ઉપયોગ drop-down list ને create કરવા માટે થાય છે.
- <option> ટેગ એ <select> એલિમેંટ ની અંદર ના list માં available ઓષ્ણાન ને define કરે છે.
- Dropdown list ના attributes:

Attribute	Description
Name	Drop-down list નું નામ define કરે છે.
Required	Form સબમિટ કરે એની પહેલા યુઝર એ વેબ્યુ ને સિલેક્ટ કરવી જરૂરી છે એવું specify કરે છે.
Value	સિલેક્ટ એલિમેંટ માટે વેબ્યુ ને specify કરે છે.
selected	જ્યારે પેજ લોડ થાય ત્યારે આ ઓષ્ણાન પહેલે થી select કરેલો હોય છે.

```
<select name="language">
  <option>Gujarati</option>
  <option selected>English</option>
  <option>Hindi</option>
</select>
```

List Box

- આપેલા list ની ચોઇસ માથી સિંગલ અથવા મલ્ટિપલ ઓષ્ણાન ને સિલેક્ટ કરવા માટે list box નો ઉપયોગ થાય છે.
- <select> એલિમેંટ નો ઉપયોગ એ list box ને create કરવા માટે થાય છે.
- <option> tags એ <select> એલિમેંટ ની અંદર available ઓષ્ણાન ના list માથી define કરવામાં આવે છે.
- list box ના attributes:

Attribute	Description
Name	List box માટે નામ define કરે છે.
Required	Form સબમિટ થાય એની પહેલા યુઝર એ વેબ્યુ ને સિલેક્ટ કરવી required છે તે define કરે છે.
Value	સિલેક્ટ એલિમેંટ માટે વેબ્યુ ને specify કરે છે.
selected	જ્યારે પેજ લોડ થાય ત્યારે આ ઓષ્ણાન પહેલે થી સિલેક્ટ કરેલો હોય છે.
multiple	Specify કરે છે કે મલ્ટિપલ ઓષ્ણાન એક સાથે સિલેક્ટ કરી શકાય છે.
size	List boxમાં visible ઓષ્ણાન ના number ને define કરે છે.

```
<select name="language" size="3" multiple>
    <option>Gujarati</option>
    <option selected>English</option>
    <option>Hindi</option>
    <option>Telugu</option>
</select>
```

Hidden

- **<input type="hidden">** એ hidden field define કરે છે.
- Hidden field define કરે છે. (યુઝર માટે વીજીબલ નથી).
- Hidden field ધણીવાર default વેલ્યુ ને પણ સ્ટોર કરે છે અથવા તેનું મૂલ્ય JavaScript દ્વારા પણ બદલી શકાય છે.
- Hidden fields ના અડિટોરીમાં નાચાય નથી.

Property	Description
name	Hidden ઈનપુટ ફીલ્ડ ના નામ attribute ની વેલ્યુ ને set અથવા return કરે છે.
value	Hidden ઈનપુટ ફીલ્ડ ની વેલ્યુ attribute ની વેલ્યુ ને set અથવા return કરે છે.

```
<input type="hidden" name="country" value="India">
```

Button

- **<input type="button">** એ બટન ને define કરે છે.
- Clickable બટન એ જ્યારે ક્લિક થાય ત્યારે કેટલીક action કરે છે.
- Button ના અડિટોરીમાં નાચાય નથી.

Property	Description
name	આપેલ ફીલ્ડ ને નામે assign કરો જેથી પછીથી તમે તેનો reference તરીકે ઉપયોગ કરી શકો.
value	બટન પર text label નક્કી કરે છે.

```
<input type="button" name="submit" value="Click me">
```

Reset Button

- **<input type="reset">** reset બટન define કરે છે.
- Form બધી વેલ્યુ ને default વેલ્યુ માં સેટ કરી નાભે છે.

```
<input type="reset">
```

Submit Button

- **<input type="submit">** form handler એ form સબમિટ કરવા માટે નું બટન define કરે છે.
- Form-handler એ ઈનપુટ ડેટા ની process કરવા માટે નું script સાથેનું સર્વર પેજ છે.
- Form handler એ form ના action attribute માં specify કરે છે.

```
<input type="submit" value="Save">
```

Submitting Form values, using \$_GET and \$_POST method

- Browser client એ વેબ સર્વર પર માહિતી મોકલી શકે છે તેની બે મેથડ છે.
 - GET Method
 - POST Method
- GET અને POST બંને એરે બનાવે છે. (e.g. array (key1 => value1, key2 => value2, key3 => value3, ...)).
- Array key/value ની pair બનાવે છે, જ્યાં key એ form controls ના name છે અને વેલ્યુ એ ચુઝર ના ઇનપુટ ડેટા છે.
- GET અને POST ને \$_GET અને \$_POST તરીકે ગણવામાં આવે છે. આ **superglobals**, વેરિએબલ છે કારણ કે તે હંમેશા accessible, scope ને regardless અને કોઈ પણ function, class અથવા કોઈ પણ ફાઇલ વગર તેને access કરી શકાય છે.
- **\$_GET** એ URL ના parameter દ્વારા current script માથી pass થતાં વેરિએબલ નો એરે છે.
- **\$_POST** HTTP POST method દ્વારા current scriptમાં pass કરાયેલ વેરિએબલ નો એરે છે.

GET Method

- GET મેથડ સાથે form માંથી મોકલેલી માહિતી દરેક ને visible હોય છે.
- GET મેથડ થી limited data ને send કરી શકો છો. જેની limitation 2000(approx) character ની છે.
- જોકે વેરિએબલ URL માં display થાય છે તેથી પેજને bookmark કરવું possible છે. આ કેટલાક cases માં ઉપયોગી થઈ શકે છે.
- GET નો ઉપયોગ non sensitive data ને send કરવા માટે થાય છે.
- GET એ બાઈનરી ડેટા ને send કરી શકતો નથી જેમ કે, images અથવા word document
- તે secure નથી પણ fast છે.
- જ્યારે તમે form collection element માં GET મેથડ નો ઉપયોગ કરો છો ત્યારે માહિતી એ query string ના concept નો ઉપયોગ કરીને URL દ્વારા destination file પર મોકલવામાં આવશે.
- Name અને pair ના collection ને query string કર્ણે છે. Name અને value (&) sign થી જુદી પડે છે.
name1=value1&name2=value2&name3=value3
- દરેક query string URL સાથે start થાય છે.
- Query string ના ડેટા ની maximum length limited હોય છે.

Note: passwords અથવા બીજી sensitive માહિતી ને send કરવા માટે GET મેથડ નો ઉપયોગ કરવો નહિ.

Submitting and accessing form value using GET OR Passing variable through URL

Example

```
<html>
  <body>
    <form action="welcome_get.php" method="get">
      Name: <input type="text" name="name"><br>
```

```

E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
</body>
</html>

welcome_get.php
<html>
<body>
<?php
    $name=$_GET['name'];
    $email= $_GET['email'];
    echo "welcome ".$name."<br>";
    echo "Your email address is: ".$email;
?
</body>
</html>

output:
Welcome Peter
Your email address is: Peter94@example.com

```

POST Method

- POST મેથડ સાથેના form માથી મોકલેલી માહિતી બીજા માટે invisible હોય છે (ખધા name/values HTTP requestના body માં embedded કરેલી છે.)
- POST મેથડ માં માહિતી ને send કરવા માટે ની કોઈ limit હોતી નથી.
- જોકે વેરિએબલ URL માં display થાય છે તેથી પેજને bookmark કરવું possible નથી.
- POST મેથડ નો ઉપયોગ ASCII તેમજ બાઇનરી ડેટા ને send કરવા માટે થાય છે.
- POST મેથડ નો ઉપયોગ sensitive data ને send કરવા માટે થાય છે.
- તે વધારે secure છે પણ GET ની compare માં ધીમું છે.
- તે unlimited form વેરિએબલ ને post કરી શકે છે.

Submitting and accessing form value using POST

Example

```

<html>
<body>
<form action="welcome.php" method="post">
    Name: <input type="text" name="name"><br>
    E-mail: <input type="text" name="email"><br>

```

- ```

<input type="submit">
</form>
</body>
</html>

```
- યુઝર જ્યારે ઉપર નું form fill કરે છે અને સબમિટ બટન ને ક્લિક કરે છે ત્યારે form data "welcome.php" નામ ની PHP file પર process કરવા મોકલે છે.
  - Form data એ HTTP POST મેથડ સાથે મોકલવામાં આવે છે.
  - સબમિટ થયેલા ડેટા ને display કરવા માટે બધા વેરિએબલ ને echo કરવા પડે છે.

### welcome.php

```

<html>
<body>
<?php
 $name=$_POST['name'];
 $email= $_POST['email'];
 echo "welcome ".$name."
";
 echo "Your email address is: ".$email;
?

```

</body>

</html>

### Output:

Welcome Peter

Your email address is: Peter94@example.com

## \$\_REQUEST variable

- PHP એ \$\_REQUEST વેરિએબલ એ બંને \$\_GET અને \$\_POST ધરાવે છે.
- GET અને POST બંને સાથે મોકલવામાં આવેલા ડેટા માથી રિઝલ્ટ મેળવવા માટે PHP \$\_REQUEST વેરિએબલ નો ઉપયોગ કરી શકાય છે.

### Example

```

<html>
<body>
<form action = "ProcessRequest.php" method = "POST">
 Name: <input type = "text" name = "name" />
 Age: <input type = "text" name = "age" />
 <input type = "submit" />
</form>
</body>

```

</html>

### ProcessRequest.php

```
<?php
 echo "Welcome ". $_REQUEST['name']. "
";
 echo "You are ". $_REQUEST['age']. " years old.";
?>
```

#### Output:

Welcome Peter

You are 31 years old.

### Difference between GET and POST

|                             | GET                                                                                                                                            | POST                                             |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| Bookmarked                  | Bookmark કરી શકાય છે.                                                                                                                          | Bookmark કરી શકતું નથી.                          |
| Cached                      | Cache કરી શકાય છે.                                                                                                                             | Cache કરી શકતું નથી.                             |
| History                     | બ્રાઉઝર history માં parameters save કરી શકાય છે.                                                                                               | બ્રાઉઝર history માં parameters save કરી શકતા નથી |
| Restrictions on data length | હા, ડેટા મોકલતી વખતે GET મેથડ URL માં ડેટા add કરી આપે છે અને URL ની length limited છે.<br>(Maximum URL ની length approx. 2048 characters છે.) | No restrictions                                  |
| Restrictions on data type   | ફક્ત ASCII characters જ અને allow છે.                                                                                                          | બાઇનરી ડેટા પણ allow હોય છે.                     |
| Security                    | GET એ POST ની સરખામણી માં ઓછું secure છે કારણ કે ડેટા URL ના ભાગ માં send થાય છે જેથી પાસવર્ડ અને બીજી sensitive માહિતી નો ઉપયોગ કરી શકતો નથી. | POST એ GET કરતાં safe છે.                        |
| Visibility                  | ડેટા એ URL માં visible થાય છે.                                                                                                                 | ડેટા એ URL માં display થતો નથી.                  |

### PHP Cookie

- Cookie નો ઉપયોગ ધણીવાર યુઝર ને identify કરવા માટે થાય છે. Cookie એ small text ફાઈલ હોય છે જે સર્વર એ યુઝર ના કમ્પ્યુટર પર embed કરે છે દરેક વખતે તે જ કમ્પ્યુટર બ્રાઉઝર સાથે પેજ ને request મોકલે છે અને તે cookie ને પણ મોકલશે. PHP સાથે તમે cookie વેલ્ચુ બનાવી શકો છો અને retrieve પણ કરી શકો છો.

### Creating Cookie

- PHP मાં તમે setcookies() function નો ઉપયોગ કરી ને cookie ને create કરી શકો છો.
- Syntax:** setcookie(name, value, expire time, path, domain, secure, httponly)
  - Name** – તમે જે name ની cookie બનવા માગો તે name specify કરે છો.
  - Value** – વેલ્યુ જે તમારા દ્વારા બનાવેલ �cookie સાથે associate છે.
  - Expire time** – જ્યારે cookie expire અને delete થાય એ time ને expire time કહેવાય તે optional છે.
  - Path** – આ directories specify કરે છે કે જેના માટે cookie valid છે.
  - Domain** – domain એ ઇન્ટરનેટ ડોમેન ને represent કરે છે જ્યાથી cookie based communication ને allow કરે છે.
  - Secure** – આને specify કરવા માટે 1 ને set કરી શકાય છે કે cookie ને ફક્ત HTTPS નો ઉપયોગ કરીને secure transmission દ્વારા મોકલવી જોઈએ બાકી 0 પર set કરો જેનો અર્થ એ છે કે cookie રેખુલર HTTP દ્વારા મોકલી શકાય છે.

**Note:** setcookie() function એ <html> ટેગ ની પહેલા જ મૂકવામાં આવે છે.

### Example

#### CreateCookies.php

```
<?php
 setcookie('cookiename','cookievalue', time()+60*60*24*5);
 echo "Cookie is set";
?>
```

### Retrieve Cookie/ Accessing Cookies Values

- PHP માં \$\_COOKIE વેરિએબલ નો ઉપયોગ cookie ની વેલ્યુને retrieve કરવા માટે થાય છે.
- નીચે આપેલા ઉદાહરણમાં, આપણે "username" name ની cookie ની વેલ્યુ retrieve કરીને પેજ માં display કરી છે.

### Example

#### RetrieveCookies.php

```
<?php
 $cookiename =$_COOKIE[' cookiename'];
 echo "Cookie named:".$cookiename;
?>
```

- નીચેના ઉદાહરણમાં આપણે કોઈ cookie set કરી છે કે નહીં તે find કરવા માટે isset() function નો ઉપયોગ કરીએ છીએ.

#### RetrieveCookies.php

```
<?php
if(isset($_COOKIE[' cookiename ']))
```

```

echo " Cookie ".$_COOKIE[' cookiename ']. " is not set! " ;
else
echo " Cookie ".$_COOKIE[' cookiename ']. " is set! ";
?>

```

### Removing Cookie/ Delete Cookie

- કોઈ cookie delete કરતી વખતે expiration date એ ભૂતકાળની હોવી જોઈએ.

#### DeleteCookies.php

```

<?php
setcookie(' cookiename ',' cookievalue ',time()-60*60*24*5);
echo "Cookie is deleted";
?>

```

### PHP Session

- PHP session વેરીએબલ નો ઉપયોગ યુઝર session માટેની માહિતી સ્ટોર કરવા અથવા બદલવા માટે થાય છે.
- Session વેરીએબલ વિશે માહિતી ધરાવે છે, અને એક એપ્લિકેશનમાં બધા પેજ માટે available છે.

#### PHP Session Variables

- જ્યારે તમે કોઈ application સાથે work કરો છો, ત્યારે તમે તેને open કરો છો, અમુક changes કરો છો, અને પછી તેને close કરો છો આ બધું session જેવું છે.
- કમ્પ્યુટર જાણો છે તમે કોણ છો. તે જાણો છે તમે ક્યારે application start કરો છો અને ક્યારે end કરો છો. પરંતુ ઇન્ટરનેટ પર એક પ્રોબ્લેમ છે, વેબ સર્વર જાણતું નથી કે તમે કોણ છો અથવા તમે શું કરો છો, કારણ કે HTTP address state ને maintain કરતું નથી.
- Session વેરીએબલ ને જુદા જુદા પેજ પર યુઝર ની માહિતી નો ઉપયોગ કરવા માટે સ્ટોર કરી ને પ્રોબ્લેમ ને solve કરીએ છીએ (e.g. username, favourite color, etc.)
- By default, session વેરીએબલ એ યુઝર બ્રાઉઝર ને close ન કરે ત્યાં સુધી ચાલુ રહે છે.
- તેથી Session વેરીએબલ એક સિંગલ યુઝર ની માહિતી ધરાવે છે અને એક application માં બધા પેજ available હોય છે.
- Session આ UID પર આધારિત દરેક visitor અને સ્ટોર વેરીએબલ માટે unique id (UID) બનાવીને કાર્ય કરે છે. UID કાં તો cookie માં સ્ટોર છે અથવા URL માં propagate છે.

### Starting Session

- તમારા PHP session માં તમે યુઝર ની માહિતી સ્ટોર કરો એની પહેલા session ને start કરવું જરૂરી છે.
- Session એ **session\_start()** function થી start થાય છે.
- session\_start() function એ <html> ટેગ ની પહેલા મૂકવામાં આવે છે.

```

<?php
session_start();
?>

```

- ```
<html>
<body>
</body>
</html>
```
- ઉપર નો કોડ યુઝર ના session ને સર્વર સાથે register કરશે, તમને યુઝર ની માહિતી save કરવાનું start કરવાનું allow કરશે અને યુઝર ના session માટે UID assign કરશે.

Creating session variable

- Session વેરીએબલ ને store અને retrieve માટે \$_SESSION વેરીએબલ નો ઉપયોગ થાય છે.
- Syntax:** \$_SESSION['variableName']=Value;

Example

SessionDemo.php

```
<html>
<head>
<title>Session Demo</title>
</head>
<body>
<form name="formname" method="POST" action="SessionDemoProcess.php">
    User Name:<input type="text" name="username"/><br/>
    Password:<input type="password" name="pass"/><br/>
    <input type="submit" name="submit" value="Save"/>
</form>
</body>
</html>
```

SessionDemoProcess.php

```
<?php
Session_start();
$uname = $_POST['username'];
$pwd = $_POST['pass'];
if($uname=="admin" and $pwd=="123")
{
    $_SESSION['username']=$uname;
    header("location:Welcome.php?un=".$_SESSION['username']);
}
else
{
```

```

        echo "enter correct username and password";
    }
?>
Welcome.php
<?php
    Session_start();
    if(isset($_SESSION['username']))
    {
        echo "Welcome:".$_SESSION['username'];
    }
    else
    {
        header("location:SessionDemo.php");
    }
?>
<html>
<head>
    <title>Session Demo</title>
</head>
<body>
    <a href="Logout.php">Logout</a>
</body>
</html>

```

Destroying a Session

- જો તમે અમુક session ના data ને delete કરવા માગતા હોય તો unset() અથવા session_destroy() function નો ઉપયોગ થાય છે.
- unset() function નો ઉપયોગ specify session વેરીએબલ ને free કરવા માટે થાય છે.
- તમે session_destroy() function ને call કરીને તમે session ને completely destroy કરી શકો છો.
- Note:** session_destroy() તમારું session restart કરશો અને તમે તમારો સ્ટોર session કેટાને ગુમાવશો.

Example

Logout.php

```

<?php
    Session_start();
    session_destroy();
    header("location:SessionDemo.php");

```

?>

Difference between Cookie and Session

COOKIE	SESSION
Cookies એ client-side ફાઇલ છે જેમાં યુઝર ની માહિતી હોય છે.	Sessions એ server-side ફાઇલ છે જેમાં યુઝર ની માહિતી હોય છે.
PHP માં <code>\$_COOKIE</code> એ super global વેરીએબલ છે જેનો ઉપયોગ cookie ને manage કરવા માટે થાય છે.	PHP માં <code>\$_SESSION</code> એ super global વેરીએબલ છે જેનો ઉપયોગ session ને manage કરવા માટે થાય છે.
તમારે cookie ને start કરવાની જરૂર નથી કારણ કે તે તમારા લોકલ મશીન માં સ્ટોર હોય છે.	<code>\$_SESSION</code> નો ઉપયોગ કરતાં પહેલાં તમારે <code>session_start</code> લખવું પડે છે, તે રીતે session શરૂ થશે અને તમે તે પેજ પર <code>\$_SESSION</code> વેરીએબલ ને access કરી શકો છો.
Cookies પાસે long lifespan હોય શકે છે, જે મહિનાઓ અને વર્ષો સુધી ચાલે છે.	Session પાસે limited lifespan હોય છે. જ્યારે બ્રાઉઝર close થાય ત્યારે તે expire થઈ જાય છે.
દરેક બ્રાઉઝર ના default settingના આધારે cookies ની size limited હોય છે.	જો તમે સર્વર પર તેની size ને limited કરો તો session ફક્ત તે size માં limited છે.
જો visitors નું બ્રાઉઝર તેમને allow ન કરે તો cookies ને disable કરી શકાય છે.	Visitor's દ્વારા session disable કરી શકતા નથી કારણ કે તે બ્રાઉઝર માં સ્ટોર કરી શકતા નથી.
Cookies visitor દ્વારા edit કરી શકાય છે (Sensitive ડેટા ને સ્ટોર કરવા માટે cookie નો ઉપયોગ કરવો નહીં)	Session visitor દ્વારા edit કરી શકતા નથી.
Cookie નો end યુઝર એ સેટ કરેલા time પર આધારિત છે.	જ્યારે યુઝર તેના બ્રાઉઝરને close કરે ત્યારે session બંધ થઈ જાય છે.

Database

- Database management system (DBMS) ડેટાબેઝ બનાવવા અને મેનેજ કરવા માટે નો સિસ્ટમ સોફ્ટવેર છે.
- DBMS યુઝર અને પ્રોગ્રામરને ડેટા create કરવા, પુનઃપ્રાપ્ત (retrieve) કરવા, અપડેટ કરવા અને મેનેજ કરવા માટે વ્યવસ્થિત રીતે provide કરે છે.
- અત્યારે, આપણે વિશાળ ડેટાને સ્ટોર અને મેનેજ કરવા માટે relational database management systems (RDBMS) નો ઉપયોગ કરીએ છીએ. આને relational database કહેવામાં આવે છે કારણ કે તમામ ડેટા જુદા જુદા ટેબલમાં સ્ટોર થાય છે અને primary keys અથવા foreign keyનો ઉપયોગ કરીને relations establish થાય છે.
- Relational Database Management System (RDBMS) એ એક સોફ્ટવેર છે જે:
 - Tables, columns અને indexes સાથે ડેટાબેઝ ને implement કરવા માટે તમને enable કરે છે.
 - વિવિધ ટેબલની rows વચ્ચે Referential Integrity ની Guarantees આપે છે.
 - Indexને આપમેળે અપડેટ કરે છે.
 - SQL query ને અર્થધારણ (Interprets) કરે છે અને વિવિધ ટેબલમાંથી ને combine કરે છે.

RDBMS Terminology

- **Database:** ડેટાબેઝ એ related data સાથે ટેબલનું કલેક્શન છે.
- **Table:** ટેબલએ row અને column માં ગોઠવાયેલા ડેટા એલિમેન્ટનું કલેક્શન છે.
- **Column:** Relational ટેબલમાં, columnએ ચોક્કસ પ્રકારની વેલ્યુ છે.
- **Row:** એક row (tuple અથવા રેકૉર્ડ) સંબંધિત ડેટાનું એક group છે.
- **Redundancy:** ડેટાબેઝ માં સરખા ડેટા ની વધારે copies હોય તેને redundancy કહેવાય છે.
- **Primary Key:** ટેબલના રેકૉર્ડને uniquely identify કરવા માટે primary key નો ઉપયોગ કરવામાં આવે છે અને તેમાં duplicate ડેટા અને null વેલ્યુ શામેલ હોતી નથી.
- **Foreign Key:** અન્ય ટેબલની primary key માંથી વેલ્યુ મેળવા માટે foreign key નો ઉપયોગ થાય છે.
- **Composite Key:** Composite key એ ટેબલમાં બે અથવા વધુ column નું combination છે જેનો ઉપયોગ ટેબલની દરેક row ને uniquely identify કરવા માટે કરી શકાય છે.
- **Index:** ડેટાબેઝ index એ ડેટા structure છે જે ટેબલમાં operationની speed ને improve કરે છે.

What is phpMyAdmin?

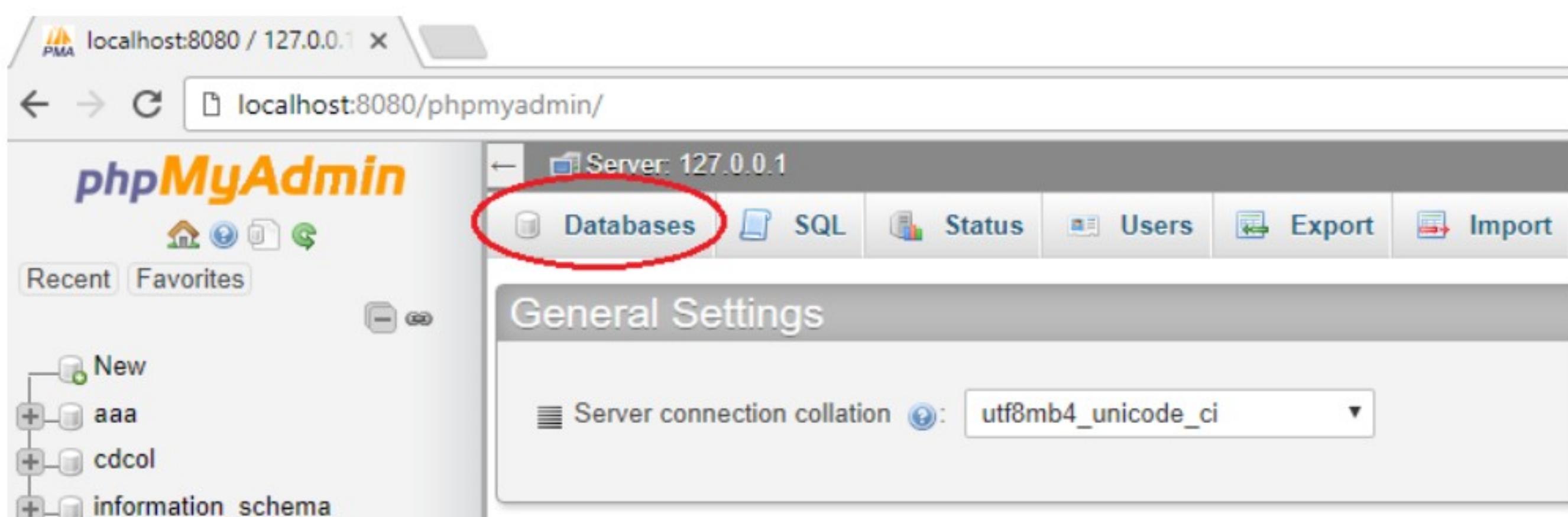
- MySQL ડેટાબેસેસ મેનેજમેન્ટ માટે phpMyAdmin સૌથી લોકપ્રિય એપ્લિકેશન્સમાંની એક છે.
- તે PHP માં લઘેલું free tool છે.
- આ સોફ્ટવેર દ્વારા તમે MySQL ડેટાબેઝ ટેબલને create, alter, drop, delete, import અને export કરી શકો છો.
- તમે MySQL queries ને run શકો છો, optimize, repair અને ટેબલને check, change collation અને અન્ય ડેટાબેઝ મેનેજમેન્ટ ના commands ચલાવી શકો છો.

What is MySQL database? OR Introduction of MySQL Database

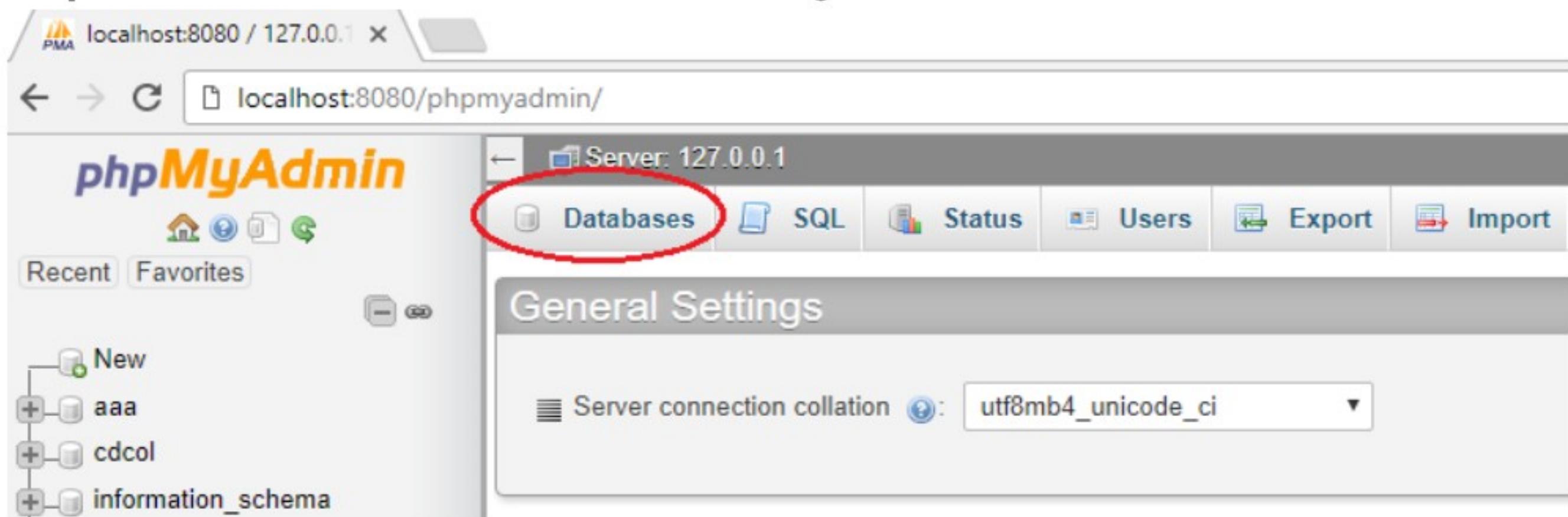
- MySQL એ freely available open source Relational Database Management System (RDBMS) છે જે Structured Query Language (SQL) નો ઉપયોગ કરે છે.
- ડેટાબેઝમાં content add કરવા, access કરવા અને manage કરવા માટે SQL સૌથી લોકપ્રિય લેન્વેજ છે.
- Standard SQL કમાન્ડ્સ, જેમ કે CREATE, DROP, INSERT અને UPDATE નો ઉપયોગ MySQL સાથે કરી શકાય છે.
- MySQL નો ઉપયોગ વિવિધ એપ્લિકેશનો માટે થઈ શકે છે, પરંતુ તે સામાન્ય રીતે વેબ સર્વર પર જોવા મળે છે.
- MySQL નો ઉપયોગ કરતી વેબસાઇટમાં વેબ પેજ આવેલા હોય છે જે ડેટાબેઝમાંથી માહિતીને access કરે છે. આ પેજને "dynamic" પેજ કહેવામા આવે છે, જેનો અર્થ એ છે કે દરેક પેજ નો content એ ડેટાબેઝમાંથી પેજ load થાય તે રીતે જનરેટ થાય છે.
- વેબસાઇટ્સ કે જે ડાયનેમિક વેબ પેજ નો ઉપયોગ કરે છે તેને ઘણીવાર ડેટાબેઝ-આધારિત વેબસાઇટ તરીકે ઓળખવામાં આવે છે.
- MySQL નો ઉપયોગ કરતી ઘણી ડેટાબેઝ-આધારિત વેબસાઇટ્સ ડેટાબેઝમાંથી માહિતીને access કરવા માટે PHP જેવી Web scripting language નો પણ ઉપયોગ કરે છે.
- MySQL commands ને PHP કોડ માં include કરી શકાય છે, જે ડેટા અથવા માહિતીમાંથી વેબ પેજ ને જનરેટ કરવા દે છે. કારણ કે MySQL અને PHP બંને open source (જેનો અર્થ છે કે તેઓ ડાઉનલોડ અને ઉપયોગ માટે free છે). જે PHP/ MySQL ના કોમ્બિનેશન ડેટાબેઝ-આધારિત વેબસાઇટ્સ માટે લોકપ્રિય બની ગયું છે.

Creating Database using phpmyadmin

Step 1: તમારું બ્રાઉઝર ખોલો અને <http://localhost/phpmyadmin> લખો / તમને નીચેની window મળશે,



Step 2: ડેટાબેઝ menu પર ક્લિક કરો જ્યાં mark કર્યું છે.



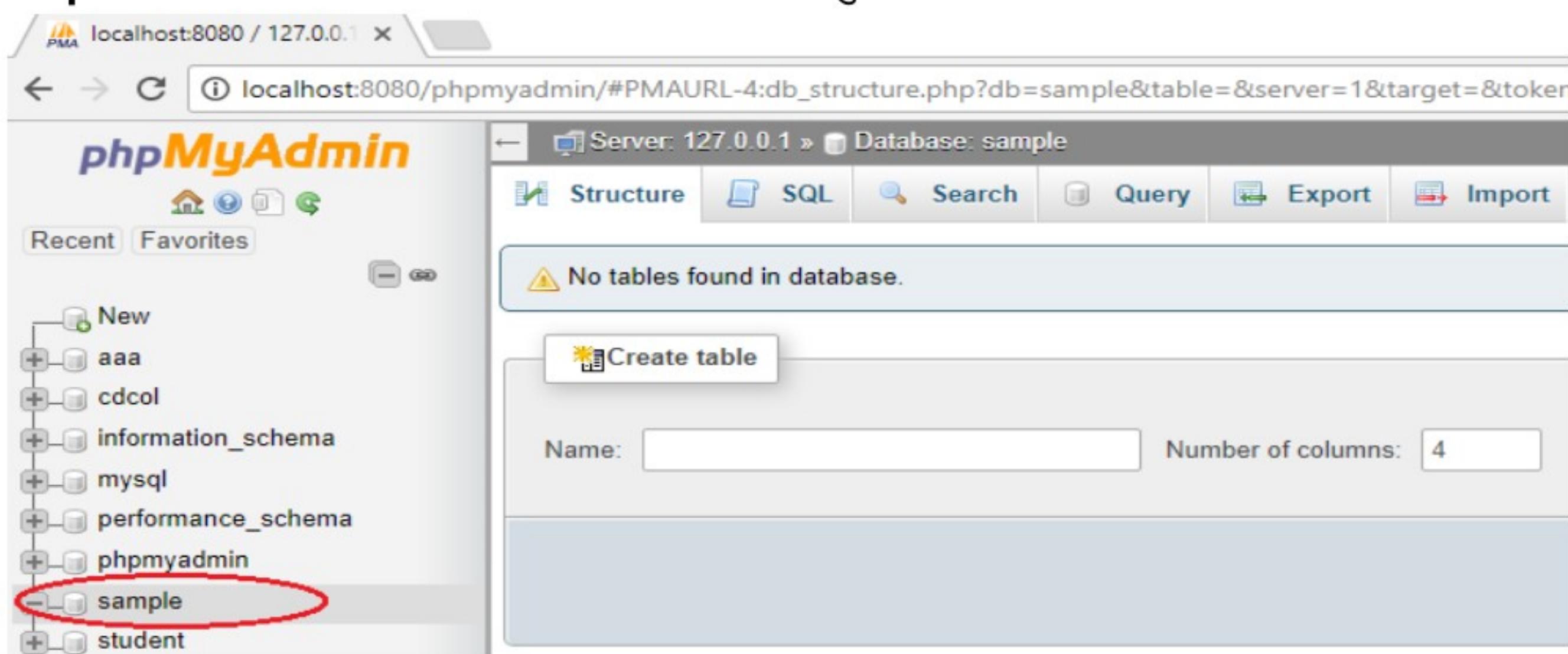
The screenshot shows the phpMyAdmin interface for a MySQL server at 127.0.0.1. The left sidebar lists databases: New, aaa, cdcol, information_schema, and mysql. The main panel displays 'General Settings' with a dropdown for 'Server connection collation' set to 'utf8mb4_unicode_ci'. The 'Databases' tab is highlighted with a red circle.

Step 3: તમારા ડેટાબેઝનું નામ લખો, પછી 'create' બટનને ક્લિક કરો.



The screenshot shows the 'Databases' page in phpMyAdmin. It lists existing databases: aaa, cdcol, information_schema, mysql, and a newly created 'sample'. A 'Create database' button is visible above the list, with the word 'sample' entered into the input field. This input field is highlighted with a red circle.

Step 4: તમારો ડેટાબેઝ બની ગયો છે. તમે તે ડાબીબાજુ પર શોધી શકો છો.



The screenshot shows the 'sample' database selected in the left sidebar. The main panel displays the 'Structure' tab for the 'sample' database, which currently contains no tables. A 'Create table' button is visible. The 'sample' database entry in the sidebar is highlighted with a red circle.

Creating Database using Console

- કમાંડ લાઇનમાથી નીચેના કમાંડ સાથે MySQL સર્વર પર લોગિન કરો:

mysql -u root -p

- આ case માં, - u flag સાથે user root specify કરે છે, અને પછી - p flag નો ઉપયોગ કરે છે જેથી MySQL પાસવર્ડ માટે prompt આપે છે. લોગિન પૂર્ણ કરવા માટે તમારો current પાસવર્ડ દાખલ કરો.
- તમારે હવે MySQL prompt પર હોવું જોઈએ જે આનાથી ખૂબ સમાન લાગે છે:
mysql>
- Name test સાથે ડેટાબેઝ બનાવવા માટે નીચે આપેલ કમાંડ લખો:

CREATE DATABASE test;

Data types (field types) of MySQL

- MySQL નણ જુદી જુદી ડેટા ટાઇપ છે:
 - Numeric
 - Date and Time
 - String

- Numeric**

Type	Use For	Size
TINYINT	very small integer	Signed range -128 થી 127. Unsigned range 0 થી 255.
SMALLINT	small integer	Signed range -32768 થી 32767. Unsigned range 0 થી 65535.
MEDIUMINT	medium-sized integer	Signed range -8388608 થી 8388607. Unsigned range 0 થી 16777215.
INT	normal-sized integer	Signed range -2147483648 થી 2147483647. Unsigned range 0 થી 4294967295.
BIGIN	large integer	Signed range -9223372036854775808 થી 9223372036854775807. Unsigned range 0 થી 18446744073709551615.

- Date and time**

Type	Description	Example
DATE	YYYY-MM-DD ફોર્મેટમાં એક તારીખ, 1000-01-01 અને 9999-12-31 વાંચે	December 30 th , 1973 એ 1973-12-30 જેમ સ્ટોર કરવામાં આવશે.
DATETIME	YYYY-MM-DD HH માં તારીખ અને સમય combination: YYYY-MM-DD HH:MM:SS	બપોરે ના 3:30 ડિસેમ્બર 30 th , 1973 એ 1973-12-30 15:30:00 જેમ સ્ટોર કરવામાં આવશે.

	ફોર્મેટ, 1000-01-01 00:00:00 અને 9999-12-31 23:59:59 વચ્ચે.	
TIMESTAMP	Midnight 1 ^s જાન્યુઆરી, 1970 અને 2037 માં ક્યારેક્ટ timestamp.	આ અગાઉના DATETIME ફોર્મેટ જેવું લાગે છે, ફક્ત નંબરો વચ્ચે હાઇફન્સ વિના; 3:30 બપોરે 30 મી ડિસેમ્બર, 1973 ના રોજ 19731230153000 (YYYYMMDDHHMMSS) તરીકે સ્ટોર કરવામાં આવશે.
TIME	HH:MM:SS ફોર્મેટ માં સમય store કરો.	02:20:34.
YEAR	એક વર્ષને 2-digit અથવા 4-digit ના ફોર્મેટ માં store કરે છે.	જો લંબાઈ 2 તરીકે specify થાય છે (ઉદાહરણ તરીકે YEAR (2)), વર્ષ 1970 થી 2069 (70 થી 69) વચ્ચે હોઇ શકે છે. જો લંબાઈ 4 તરીકે specify થાય છે, તો વર્ષ 1901 થી 2155 હોઇ શકે છે. Default લંબાઈ 4 છે.

- **String Type**

Type	Description
CHAR(size)	<ul style="list-style-type: none"> Lengthમાં 1 અને 255 characters ની વચ્ચે fixed-length ની string (ઉદાહરણ તરીકે, CHAR (5)). Length define કરવી જરૂરી નથી, પરંતુ default 1 છે.
VARCHAR(size)	<ul style="list-style-type: none"> Lengthમાં 1 અને 255 characters ની વચ્ચે variable-length ની string. ઉદાહરણ તરીકે, VARCHAR(25). VARCHAR field બનાવતી વખતે તમારે length define કરવી જરૂરી છે. જો તમે 255 કરતા વધુ value આપો છો તો તે TEXT ટાઇપ માં convert થશે.
TEXT	<ul style="list-style-type: none"> 65535 characterની maximum length વાળી field. TEXT તરીકે define field પણ વધારે ડેટા ધરાવે છે. Sorting અને comparing કરતી વખતે TEXT એ case insensitive હોય છે.
BLOB	<ul style="list-style-type: none"> BLOB એ text field સમાન છે. 65535 characterની maximum length વાળી field. Sorting અને comparing કરતી વખતે TEXT એ case sensitive હોય છે.
ENUM	<ul style="list-style-type: none"> તે listing કરવા માટે વપરાય છે. ENUM ને define કરતી વખતે, તમે તે item નું list બનાવી રહ્યા છો કે જેનાથી વેલ્ચુ સિલેક્ટ કરવી જરૂરી છે (અથવા તે NULL થઈ શકે છે). ઉદાહરણ તરીકે, જો તમે ઇચ્છતા હો કે તમારું field "A" અથવા "B" અથવા "C" શામેલ હોય, તો તમે તમારા ENUM ને ENUM ('A', 'B', 'C') તરીકે define કરશો તો ફક્ત તે વેલ્ચુ (અથવા NULL) તે field ને populate કરી શકે છે.

Integration of PHP with MySQL

- PHP, માંથી MySQL ના જુદા-જુદા commands ને execute કરવું શક્ય છે. PHP, જુદા-જુદા built-in ફંક્શન આપે છે જે તમે PHP, પેજ થી MySQL commands નો ઉપયોગ કરવાનું allow કરે છે. આમ તમે PHP ને MySQL સાથે integrate કરી શકો છો.
- વિવિધ PHP ફંક્શન છે જે તમને PHP ને MySQL સાથે integrate કરવાની facility આપે છે.

mysql_connect()

- mysql_connect() ફંક્શન MySQL સર્વર પર નવું કનેક્શન open કરે છે.
- MySQL સર્વર થી કનેક્શન represent કરતા ઓફ્ઝિકલ return કરે છે.

Syntax: mysql_connect(host, username, password)

Parameter	Description
host	Host નું name અથવા IP address ને specify કરે છે.
username	MySQL નું username specify કરે છે.
password	MySQL નો password specify કરે છે.

Example

```
<?php
    $conn=mysql_connect("localhost","root","");
    if(!$conn)
    {
        echo " Could not connect";
    }
?>
```

mysql_select_db()

- mysql_select_db() ફંક્શન નો ઉપયોગ કનેક્શન માટે default ડેટાબેસને બદલવા માટે થાય છે.
- success પર TRUE અને failure પર FALSE return કરે છે.

Syntax: mysql_select_db(dbname)

Parameter	Description
dbname	Required. dbname નો ઉપયોગ કરવા માટે default ડેટાબેસ specify કરે છે.

Example

```
<?php
    $conn=mysql_connect("localhost","root","");
    $db=mysql_select_db("student");
    if(!$conn)
    {
        echo " Database error";
    }
?>
```

?>

mysql_query()

- mysql_query() ફંક્શન MySQL ડેટાબેસ પર query ને execute કરે છે.

Syntax: mysql_query(query)

Parameter	Description
query	Required. query સ્ટ્રિંગ ને specify કરે છે.

Example

```
<?php
$conn=mysql_connect("localhost","root","");
$db=mysql_select_db("student");
$qry="select * from studentinfo";
$result=mysql_query($qry);
if(!$result) {
    echo "Database query failed";
}
?>
```

mysql_num_rows()

- mysql_num_rows() ફંક્શન એ રિઝલ્ટ સેટ માં rows ની સંખ્યા આપે છે.

Syntax: mysql_num_rows(result)

Parameter	Description
result	Required. mysql_query() દ્વારા return કરાયેલ રિઝલ્ટ સેટ એ identifier ને specify કરે છે.

Example

```
<?php
$conn=mysql_connect("localhost","root","");
$db=mysql_select_db("student");
$qry="select * from studentinfo";
$result=mysql_query($qry);
$rowcount=mysql_num_rows($result);
echo "Total Row:".$rowcount;
?>
```

mysql_fetch_array()

- mysql_fetch_array() ફંક્શન રિઝલ્ટ rowને associative array, numeric array અથવા બંને રીતે fetch કરે છે.
- સ્ટ્રિંગનો array return કરે છે જે fetch કરાયેલ row થી સંબંધિત છે.

- NULL જો રિઝલ્ટ સેટ માં કોઈ વધુ rows ન હોય તો.

Syntax: mysql_fetch_array(result)

Parameter	Description
result	Required. mysql_query () દ્વારા return કરાયેલ રિઝલ્ટ સેટ એ identifier ને specify કરે છે.

Example

```
<?php
$conn=mysql_connect("localhost","root","");
$db=mysql_select_db("student");
$qry="select StudentID,Name,Email,Gender from studentinfo";
$result=mysql_query($qry);
while($row=mysql_fetch_array($result))
{
    echo $row[0]."<br/>";
    echo $row['Name']."<br/>";
}
?>
```

- **Note:** આ ફંક્શનમાંથી return થતાં fieldname case-sensitive છે.

mysql_fetch_assoc()

- mysql_fetch_assoc() ફંક્શન એ રિઝલ્ટ row ને associative array તરીકે fetch કરે છે.
- Fetch row ને represent કરતી strings એ એક associative array return કરે છે.
- NULL જો result-set માં કોઈ વધુ rows ન હોય તો.

Syntax: mysql_fetch_assoc(result)

Parameter	Description
result	Required. mysql_query() દ્વારા return કરાયેલ રિઝલ્ટ સેટ એ identifier ને specify કરે છે.

Example

```
<?php
$conn=mysql_connect("localhost","root","");
$db=mysql_select_db("student");
$qry="select StudentID,Name,Email,Gender from studentinfo";
$result=mysql_query($qry);
while($row=mysql_fetch_assoc($result))
{
    echo $row['StudentID']."<br/>";
```

```

echo $row['Name']."<br/>";
}
?>

```

- **Note:** આ ફંક્શનમાંથી return થતી fieldname case-sensitive છે.

mysql_close()

- mysql_close() ફંક્શન એ પહેલા open કરેલા ડેટાબેસ કનેક્શનને બંધ કરે છે.

Parameter	Description
connection	Required. Close કરવા માટે MySQL કનેક્શનને specify કરે છે.

```

<?php
$conn=mysql_connect("localhost","root","");
if(!$conn)
{
    echo " Could not connect to database ";
}
mysql_close($conn)
?>

```

mysql_error()

- mysql_error() ફંક્શન, last MySQL operation માંથી error message મેળવવા માટે વપરાય છે.

Parameter	Description
connection	Optional. MySQL ડેટાબેઝ પર કોઈપણ ઓપરેશન કરતા પહેલાં, તમારે જેની સાથે કામ કરવા માંગો છો તે MySQL ડેટાબેસ સાથે કનેક્શન સેટ કરવું આવશ્યક છે. અને આ mysql_connect () ફંક્શન દ્વારા થાય છે.

```

<?php
$conn=mysql_connect("localhost", "root","");
$db=mysql_select_db("student");
if(!$db)
{
    echo "Something else wrong".mysql_error($conn);
    die;
}
?>

```

Developing a sample web based Application

Config.php

```
<?php
    $Conn=mysqli_connect('localhost', 'root','','workingwithsql');
    if(!$Conn)
    {
        echo "Something else wrong".mysqli_error($Conn);
        die;
    }
?>
```

StudentList.php

```
<?php
    include_once 'config.php';
    if(isset($_GET['Mode']))
    {
        if($_GET['Mode']=='delete')
        {
            $StudentID=$_GET['Studentid'];
            $DelSQL="delete from studentdata where StudentID='".$StudentID."'";
            $Result=mysqli_query($Conn,$DelSQL);
            if($Result)
            {
                echo "Record Deleted Successfully";
            }
            else
            {
                echo "Opps!! Something went wrong".mysqli_error($Conn);
            }
        }
    }
?>
<html>
    <head>
        <title>Student Registration</title>
    </head>
    <body>
        <div align="center">
```

```

<a href="StudentAddEdit.php">Add New Student</a>
<table border="1px">
    <tr>
        <th>StudentName</th>
        <th>StudentEmail</th>
        <th>StudentContact</th>
        <th>StudentGender</th>
        <th>Action</th>
    </tr>
    <?php
        $SelectStr="select * from studentdata";
        $Result=mysqli_query($Conn,$SelectStr);
        while($RowData=mysqli_fetch_assoc($Result))
        {
            //print_r($RowData);
            //exit;
        ?>
        <tr>
            <td><?php echo $RowData['StudentName']; ?></td>
            <td><?php echo $RowData['StudentEmail']; ?></td>
            <td><?php echo $RowData['StudentContact']; ?></td>
            <td><?php echo $RowData['StudentGender']; ?></td>
            <td><a href="StudentAddEdit.php?Mode=edit&Studentid=<?php
                echo $RowData['StudentID']; ?>">Update</a> | <a href=
                "StudentList.php?Mode=delete&Studentid=<?php echo
                $RowData['StudentID']; ?>">Delete</a></td>
        </tr>
        <?php
            }
        ?>
        </table>
    </div>
</body>
</html>

```

StudentAddEdit.php

```

<?php
    include_once 'config.php';
    if(isset($_POST['submit']))

```

```
$StudentID=$_POST['StuID'];
$Name=$_POST['StudentName'];
$Email=$_POST['StudentEmail'];
$Contact=$_POST['StudentContact'];
$Gender=$_POST['StudentGender'];
if(empty($Name))
{
    $ErrorMsg="Enter Name";
}
else if(empty($Email))
{
    $ErrorMsg="Enter Email";
}
else if(empty($Contact))
{
    $ErrorMsg="Enter Contact";
}
else if(empty($Gender))
{
    $ErrorMsg="Enter Gender";
}
else
{
    if((isset($StudentID))and !empty($StudentID))
    {
        $UpdateStr="Update studentdata set StudentName = '$Name',
                    StudentEmail ='$Email', StudentContact='$Contact', StudentGender=
                    '$Gender' where StudentID=$StudentID";
        $Result=mysqli_query($Conn,$UpdateStr);
        if($Result)
        {
            echo "Data Updated Seccesfully";
        }
        else
        {
            echo "Error in Data Insertion".mysqli_error($Conn);
        }
    }
}
```

```

        }
        else
        {
            $InsertStr="insert into studentdata (StudentName, StudentEmail,
            StudentContact, StudentGender) values ('$Name','$Email', '$Contact',
            '$Gender')";

            $Result=mysqli_query($Conn,$InsertStr);
            if($Result)
            {
                echo "Data Inserted Seccesfully";
            }
            else
            {
                echo "Error in Data Insertion".mysqli_error($Conn);
            }
        }
    }

?>

<?php
if(isset($_GET['Mode']))
{
    if($_GET['Mode']=='edit')
    {
        $SID=$_GET['Studentid'];
        $SelectStr="select * from studentdata where StudentID=$SID";
        $Result=mysqli_query($Conn,$SelectStr);

        if(mysqli_num_rows($Result)>0)
        {
            $ObjData=mysqli_fetch_assoc($Result);
        }
        else
        {
            echo "Data not found";
        }
    }
}

```

```

}

?>
<html>
<head>
    <title>Student Registration</title>
</head>
<body>
    <form name="StudentDetail" action="" method="POST">
        <div align="center">
            <a href="StudentList.php">Show List of Students</a>
            <fieldset style="width:50%;">
                <legend>Student Registration</legend>
                <table>
                    <tr>
                        <td><label>StudentName</label></td>
                        <td><input type="text" name="StudentName" id="StudentName" value="<?php if(isset($ObjData)) echo $ObjData['StudentName'];?>"><input type="hidden" name="StuID" id="StuID" value="<?php if(isset($ObjData)) echo $ObjData['StudentID'];?>"></td>
                    </tr>
                    <tr>
                        <td><label>StudentEmail</label></td>
                        <td><input type="text" name="StudentEmail" id="StudentEmail" value="<?php if(isset($ObjData)) echo $ObjData['StudentEmail'];?>"></td>
                    </tr>
                    <tr>
                        <td><label>StudentContact</label></td>
                        <td><input type="text" name="StudentContact" id="StudentContact" value="<?php if(isset($ObjData)) echo $ObjData['StudentContact'];?>"></td>
                    </tr>
                    <tr>
                        <td><label>Gender</label></td>
                        <td><select name="StudentGender">
                            <option value="" selected>Select one...</option>

```

```

        <option value="Male" <?php if((isset($ObjData)) and
        ($ObjData['StudentGender']=='Male')) echo "selected";
        ?>>Male</option>
        <option value="Female" <?php if((isset($ObjData)) and
        ($ObjData['StudentGender']=='Female')) echo "selected";
        ?>>Female</option>
        </select>
        </td>
    </tr>
    <tr>
        <td></td>
        <td><?php /*<input type="submit" name="submit" value="<?php
        if(isset($ObjData)) echo "Update"; else echo "Submit";?>
        id="submit"/>*/ ?>
        <input type="submit" name="submit" value="Submit"
        id="submit"/>
        </td>
    </tr>
</table>
</fieldset>
</div>
</form>
</body>
</html>
```

Output:

StudentAddEdit.php

Student Registration	
StudentName	<input type="text"/>
StudentEmail	<input type="text"/>
StudentContact	<input type="text"/>
Gender	<input type="button" value="Select one... ▾"/>
<input type="button" value="Submit"/>	

StudentList.php

<u>Add New Student</u>				
StudentName	StudentEmail	StudentContact	StudentGender	Action
Priya	priya@gmail.com	8888889999	Female	Update Delete
Sumit	sumitpatel@gmail.com	9977775555	Male	Update Delete
Kishan	kishan@yahoo.com	9822227777	Male	Update Delete

Hosting Website Using Filezilla Software

- FTP એ ફાઇલ ટ્રાન્સફર પ્રોટોકોલ માટે વપરાય છે. તે એક એવી સિસ્ટમ છે જે તમને સર્વર પર log in કરવા અને content ને અપલોડ કરવા, ડાઉનલોડ કરવા અથવા content ને મોડિફાય કરવાનું allow કરે છે.
- FileZilla એ Internet પર ફાઇલોને ટ્રાન્સફર કરવા માટે નો પાવરફૂલ અને ફીલ્ડ માં domain name enter કરો (તમે સર્વરના IP address નો ઉપયોગ પણ કરી શકો છો).
- તમારે જે username અને password ટાઇપ કરવાની જરૂર પડે છે તે એ જ છે જેનો ઉપયોગ તમે તમારા cPanel માં log in કરવા માટે કરો છો.

તમારા વેબસર્વર સાથે જોડવા માટે

- પેજની ટોચ પર ફાઇલજિલા Quickconnect બારમાં, હોસ્ટ ફીલ્ડમાં તમારું વેબ સાઇટ URL દાખલ કરો.
- તમારું FTP યુઝર નામ અને પાસવર્ડ દાખલ કરો (તમે પોર્ટ ફીલ્ડ ખાલી છોડી શકો છો).
- Quickconnect પર ક્લિક કરો.



તમારી વેબ સાઇટ ટ્રાન્સફર કરવા માટે

- લોકલ સાઇટ paneમાં, તમારી વેબ સાઇટના લોકેશન પર નેવિગેટ કરો.
- તમારી વેબસાઇટ પર તમે અપલોડ કરવા માંગો છો તે ફાઇલોને પસંદ કરો. તમે Ctrl કીને પકડી રાખીને અને તમને જોઈતી ફાઇલોને ક્લિક કરીને મલિટિપલ ફાઇલો પસંદ કરી શકો છો.
- ફાઇલોને તમે લોકલ સાઇટ paneમાંથી રિમોટ સાઇટ pane પર ટ્રાન્સફર કરવા માંગો છો તે ઘેંચો.

4. ફાઇલજિલા તમારી ફાઇલોને આપમેળે તમારી વેબસાઇટ પર ટ્રાન્સફર કરે છે.
5. જ્યારે તમે તમારી ફાઇલોને અપલોડ કરવાનું પૂરું કરો છો, ત્યારે ફાઇલજિલાને બંધ કરો અને તમારી ફાઇલો યોગ્ય રીતે અપલોડ થઈ છે તેની ખાતરી કરવા માટે તમારા મનપસંદ બ્રાઉઝરમાં તમારી વેબસાઇટને તપાસો.



Advanced Elements in HTML5:

Semantic Page Elements:

- HTML5 એ સંઘાબંધ **sectioning elements** ની રજૂઆત છે જેનો ઉપયોગ તમારા web pages ને mark કરવા માટે કરી શકો છે. આ elements નો ઉપયોગ કરીને તમારા web pages ને વધુ semantic meaning આપી શકો છો.
- HTML5 web pages નાલ different parts ને define કરવા માટે નવા semantic elements આપે છે.
 - address
 - article
 - hgroup
 - header
 - menu
 - nav
 - section
 - footer

1) **<address>** Element

- **<address>** element document નાલ author માટે contact information define કરે છે.
- **<address>** element માં font italic થઈ જાય છે.

Example:

```
<address>
    Written by <a href="mailto:dwpdmaterial@example.com">Vin Diesel</a>.<br>
    Visit us at: WebsiteName.com<br>
    At: Hadala, Rajkot<br>
    India.
</address>
```

Output:

Written by Vin Diesel.
Visit us at: WebsiteName.com
At: Hadala, Rajkot
India.

2) **<article>** Element

- **<article>** element નો use page પરના contain ને define કરે છે જે contains ના જુદા જુદા ભાગ હોઈ છે.
- **<article>** element માટે potential sources જેવા કે Blog post, Newspaper article, comment or Forum post.

Example:

```
<article>
    <h1>Blog Post Title</h1>
```

```
<p>Your Blog Post Content....</p>
</article>
```

Output:

Blog Post Title

Your Blog Post Content....

3) <header> Element

- <header> element નો ઉપયોગ દокумент અથવા section ની header ને mark up કરવા માટે થાય છે.
- <header> element માં heading element (<h1> to <h6>) હોય છે.

Example:

```
<article>
<header>
  <h1>About Form Designing using Canvas and CSS</h1>
  <p>Some additional information about this topic here.</p>
</header>
<p>The section content....</p>
</article>
```

Output:

About Form Designing using Canvas and CSS

Some additional information about this topic here.

The section content....

4) <hgroup> Element

- <hgroup> element નો ઉપયોગ બે અથવા વધારે heading elements ને group કરવા માટે થાય છે.
- <hgroup> elementનો ઉપયોગ <h1> થી <h6> elementની groupને set કરવા માટે થાય છે.
- <hgroup> કોઈ particular formatting માટે header group provide કરે છે.

Example:

```
<hgroup>
  <h1>Welcome to DWSA Material Page</h1>
  <h2>Learn HTML5 Elements from Here..</h2>
</hgroup>
<p>The rest of the content...</p>
```

Output:

Welcome to DWSA Material Page

Learn HTML5 Elements from Here..

The rest of the content...

5) <menu> Element

- <menu> element નો use commands ના list/menu ને define કરવા માટે થાય છે.
- <menu> tag નો ઉપયોગ content menu અને toolbar માટે થાય છે અને controls અને commands માંથી listing થાય છે.

Syntax: <menu type= "list | context | toolbar">

- <menu> tag એ નીચેના attribute ને support કરે છે.
 - **list:** commands ની જેમ list ને present કરે list ની જેમ command present કરે.
 - **context:** જ્યારે user right-click કરે છે ત્યારે element pop-up menu જેવું કાર્ય કરે છે.
 - **toolbar:** Commands toolbar તરીકે present કરે છે.
- <menuitem> tag command/menu item ને define કરે છે કે જે popup menu માંથી user invoke કરી શકે છે.

Example:

```
<div style="background:yellow; border:1px solid #cccccc; padding: 10px; width:25%;"  

contextmenu="mymenu">  

<p>Right-click inside this box to see the context menu!  

<menu type="context" id="mymenu">  

<menuitem label="Refresh" onclick="window.location.reload();"  

icon="ico_reload.png"></menuitem>  

<menu label="Share on...">  

<menuitem label="Facebook" icon="ico_facebook.png"  

onclick="window.open('http://www.facebook.com');">  

</menuitem></menu>  

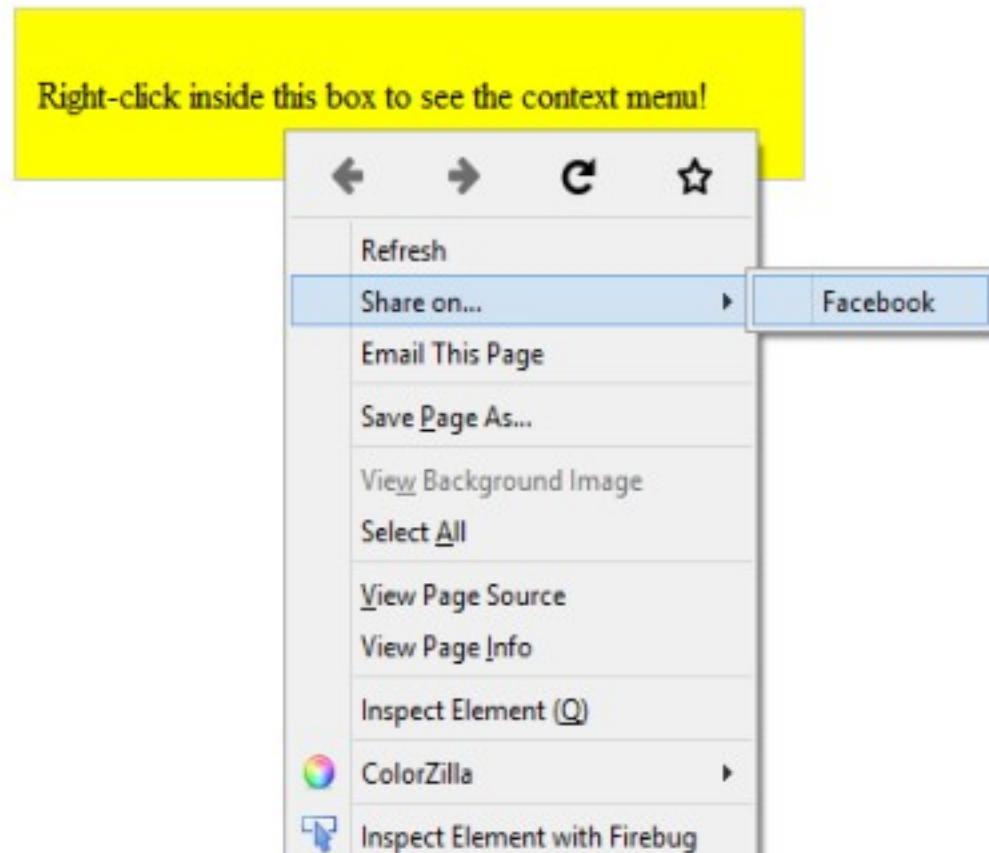
<menuitem label="Email This Page"  

onclick="window.location='mailto:?body='+window.location.href;"> </menuitem>  

</menu>  

</div>
```

Output:



6) <nav> Element

- <nav> element નો ઉપયોગ તે પેજ પર ના area ને define કરવા માટે થાય છે જેમાં બીજા page અથવા same document ની links શામેલ હોય છે.
- <nav> element નો ઉપયોગ primary navigation links, માટે કરવો જોઈ. જેમ કે main site navigation.

Example:

```
<nav>
  <ul>
    <li><a href="home.html">Home</a></li>
    <li><a href="about.html">About Us</a></li>
    <li><a href="services.html">Services</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

Output:

- [Home](#)
- [About Us](#)
- [Services](#)
- [Contact](#)

7) <section> Element

- <section> નો ઉપયોગ block ના related content ને markup કરવા માટે થાય છે.
- <section> element એ document માં section define કરે છે જેવા કે chapters, headers, footers અથવા બીજા document ના section define કરે છે.

Example:

```
<section>
  <header>
    <h2>Chapter 1: Form Designing</h2>
  </header>

  <p>
    The section text go here...
    The description of this chapter....
  </p>
  <footer>By Peter Pan</footer>
</section>
```

Output:

Chapter 1: Form Designing

The section text go here... The description of this chapter....

By Peter Pan

8) <footer> Element

- <footer> tag કોઈ document અથવા section માટે footer define કરે છે.
- <footer> element સામાન્ય રીતે authorship information, copyright information, contact information, sitemap, back to top links, related documents સમાવેશ કરે છે.

Example:

```
<footer>
  <p>Posted by: Peter Pan</p>
  <p>Contact information: <a
    href="mailto:peterpan@example.com">peterpan@example.com</a>
  </p>
  Copyright Peter Pan 2016
</footer>
```

Output:

Posted by: Peter Pan
Contact information: peterpan@example.com
Copyright Peter Pan 2016

Inline Semantic Elements:

- Semantic element (જે entire page પર ધણા structure ને add કરે છે) તેમજ નવા અથવા modify tags ને page ની અંદર નવા element તરીકે ઉમેરવામાં આવે છે.
- web page ના વિવિધ ભાગો ને define કરવા માટે HTML5 એ new inline semantic elements આપે છે.
 - command
 - details
 - dfn
 - figcaption
 - figure
 - summary/details
 - time
 - wbr

1) <command> Element

- <command> element એ command ને represent કરે છે. જેનો user use કરી શકે છે.

- Command એ context menu અથવા toolbar નો ભાગ હોઈ શકે છે, જે menu element નો use કરીને, અથવા keyboard shortcut ને define કરવા માટે, બીજા કોઈપણ page માં મૂકી શકાય છે.
- <command> tag attributes ના number ને accept કરે છે.
 - icon:** Image નું address indicate કરે છે જે આ command માટે icon તરીકે serve થાય છે.
 - checked:** તે બતાવે છે કે current command check થઈ ગયેલ છે.
 - disabled:** જો આ attribute present હોઈ તો command disabled છે.
 - label:** text ને indenties કરે છે. જે user ને display કરે છે.
 - onclick:** જ્યારે user એ command ને click કરે ત્યારે execute કરવા માટે javascript code ની single line ને indicates કરે છે.
 - title:** command શું કરે છે. તેના વિશે user ને tip display કરે છે.
 - type:** Element ના પ્રકાર અથવા behavior ને determine કરે છે. (આ command,checkbox, અથવા radio હોઈ શકે છે.)
- <command> tag નો ઉપયોગ menu system ને page પર add કરવા માટે <menu> tag ના context માં થાય છે. અત્યારે કોઈ major browser <command> tag ને support કરતું નથી.

Example:

```
<menu>
    <command label="one" onclick="alert('one')">
    <command label="two" onclick="alert('two')">
    <command label="three" onclick="alert('three')">
</menu>
```

2) <details> Element

- <details> tag additional details ને specify કરે છે. કે જે user ની પર hide અને view કરી શકે છે.
- <details> element માં summary element અને બીજા HTML tag હોઈ શકે છે. કોઈ પણ sort content ને <details> tag ની અંદર મૂકી શકાઈ છે.

Example:

```
<details>
    <summary>See More Detail Here</summary>
    <p>It's all about form designing</p>
    <p>In this chapter, we learn different HMTL elements and CSS properties.</p>
</details>
<p><b>Note :</b> The details tag is currently only supported in Opera, Chrome, and in Safari 6.</p>
```

Output:

Before:

► See More Detail Here

Note: The details tag is currently only supported in Opera, Chrome, and in Safari 6.

After Click:

▼ See More Detail Here

It's all about form designing

In this chapter, we learn different HMTL elements and CSS properties.

Note: The details tag is currently only supported in Opera, Chrome, and in Safari 6.

3) <summary> Element

- <summary> tag <details> element માટે visible heading define કરે છે.
- Heading પર click કરવાથી detail ને hide અને view કરી શકો છો.

Example:

```
<details>
  <summary>See More Detail Here</summary>
  <p>It's all about form designing</p>
  <p>In this chapter, we learn different HMTL elements and CSS properties.</p>
</details>
<p><b>Note :</b> The details tag is currently only supported in Opera, Chrome, and in Safari 6.</p>
```

Output:

► See More Detail Here

Note: The details tag is currently only supported in Opera, Chrome, and in Safari 6.

4) <dfn> Element

- <dfn> tag HTML માં term ને instance defining કરી represent કરે છે.
- મોટા ભાગના browser <dfn> tag ને italics તરીકે format કરે છે. પરંતુ આ CSS દ્વારા બદલી શકાઈ છે.

Example:

```
<p><dfn>HTML</dfn> is the standard markup language for creating web pages.</p>
```

Output:

HTML is the standard markup language for creating web pages.

5) <figure> Element

- <figure> એ semantic element છે. જે optional caption સાથે એક અથવા વધુ images ને describe કરે છે. <figure> tag directly કોઈ પણ images display કરતું નથી, તે કોઈ પણ પ્રકારની images રાખવા માટે container તરીકે ઓળખાતું હોઈ છે.
- <figure> tag એ self-contained content ને specify કરે છે. ઉદાહરણ તરીકે, illustration, diagrams, photos, code listings વગેરે.
- <figcaption> tag નો ઉપયોગ કરીને optional caption સાથે image ને supply કરી શકાય છે.

Example:

```
<p>The Sun Image</p>
<figure></figure>
```

Output:

The Sun Image



6) <figcaption> Element

- <figcaption> tag એ એક figure માટે caption provide કરે છે. અને એ જુદી <figure> tag અંદર આવેલું હોવું જોઈએ.
- Document માં photo ને mark up કરવા માટે <figure> element નો ઉપયોગ કરવામાં આવે છે. <figure> element માં caption પણ આવેલા છે.

Example:

```
<p>The Sun Image</p>
<figure>

<figcaption>Fig.1 - The Sun.</figcaption>
</figure>
```

Output:

The Sun Image



Fig.1 - The Sun.

7) <time> Element

- <time> tag એ human-readable કરી શકે તેવી date/time define કરે છે. આ field માં date, time અથવા date હશે.
- આ element નો ઉપયોગ machine-readable કરી શકે તેવી રીતે date અને time ને encode કરવા માટે થઇ શકે છે જેથી user agents એ birthday reminders અથવા scheduled events ને users નાલ calender માં add કરી શકે.

Example:

```
<p>Our Lecture start at <time>10:20</time> every morning.</p>
<p>I have a date on <time datetime="2008-02-14 20:00">this month</time>.</p>
```

<p>Note: The time element does not render as anything special in any of the major browsers.

</p>

Output:

Our Lecture start at 10:20 every morning.

I have a date on this month.

Note: The time element does not render as anything special in any of the major browsers.

8) <wbr> Element

- Word Break માટે સૂચનો ઉમેરવા માટે <wbr> (Word Break Opportunity) element નો ઉપયોગ થાય છે.
- જ્યારે word ધણો લાંબો હોય, ત્યારે તમે word break કરવા માટે <wbr> element નો ઉપયોગ કરી શકો છો.

Example:

<p>The <wbr> tag is used to mark a space in a long word where it would be <wbr> appropriate to break the word if necessary.</p>

<p>Note: No current browsers support this tag yet.</p>

Output:

The <wbr> tag is used to mark a space in a long word where it would be appropriate to break the word if necessary.

Note: No current browsers support this tag yet.

Media Elements:

- Media element HTML5 નું સૌથી સારું feature છે . HTML 4 અને તેના પહેલાં, જો તમે એનિમેશન બતાવવા અથવા વિડિઓ ચલાવવા માંગતા હો, તો તમારે Adobe Flash Player જેવા browser plug-in નો ઉપયોગ કરવાની જરૂર પડતી હતી.
- હવે HTML 5માં audio અને video ચલાવવા <audio> અને <video> tag નો ઉપયોગ કરવામાં આવે છે .તેમાં કોઈ પણ plug-in ની જરૂર પડતી નથી.
- તે <CANVAS> અને <svg> tag દ્વારા vector-based graphics માટે પણ સપોર્ટ કરે છે.
- HTML5 ના new media element નીચે આપેલા છે.
 - Canvas
 - Embed
 - Source
 - svg

1) <canvas> Element

- <CANVAS> element તમને JavaScript નો ઉપયોગ કરીને ગ્રાફિક્સ દોરવાની એક easy અને powerful રીત આપે છે .તેનો ઉપયોગ graph draw કરવા માટે, fly કરવા માટે, photosના compositions માટે અથવા simple animation માટે કરી શકીએ છીએ.
- <CANVAS> ટેગ ગ્રાફિક્સ માટે ફક્ત એક કન્ટેનર છે, ગ્રાફિક્સ દોરવા માટે સ્ક્રિપ્ટનો ઉપયોગ કરવો આવશ્યક છે.
- નીચે <CANVAS> નું ઉદાહરણ આપેલું છે જે બટનને ક્લિક કરવામાં આવે ત્યારે લંબચોરસ માં લાલ રંગ પૂરે છે.

Example:

```
<canvas id="myCanvas" width="190px" height="140px" style="border:1px solid #d3d3d3;">
```

Your browser does not support the HTML5 canvas tag.

```
</canvas>
```

```
<br>
```

```
<button type="button" onclick="draw()">click me to see a drawing</button>
```

```
<script>
```

```
  function draw()
```

```
{
```

```
    var fillcolor = document.getElementById("myCanvas");
```

```
    var redfill = fillcolor.getContext("2d");
```

```
    redfill.fillStyle = "#FF0000";
```

```
    redfill.fillRect(20, 20, 150, 100);
```

```
}
```

```
</script>
```

Output:



click me to see a drawing

2) <embed> Element

- <embed> tag નો ઉપયોગ webpage માં external application ને integrate કરવા માટે થાય છે. આ element નો ઉપયોગ સામાન્ય રીતે Flash application ને embed કરવા થાય છે.

Example:

```
<embed src="helloworld.swf">
```

- ઉપર ના example માં helloworld.swf add કરવા માં આવે છે અને જ્યારે page load થાય ત્યારે automatically play થાય છે.

3) <source> Element

- <source> tag નો ઉપયોગ media elements માટે multiple media resources ને specify કરે છે જેમ કે <video> અને <audio>.
- <source> tag તમને alternative video/audio filesને specify કરવાનું allow કરે છે .

Example:

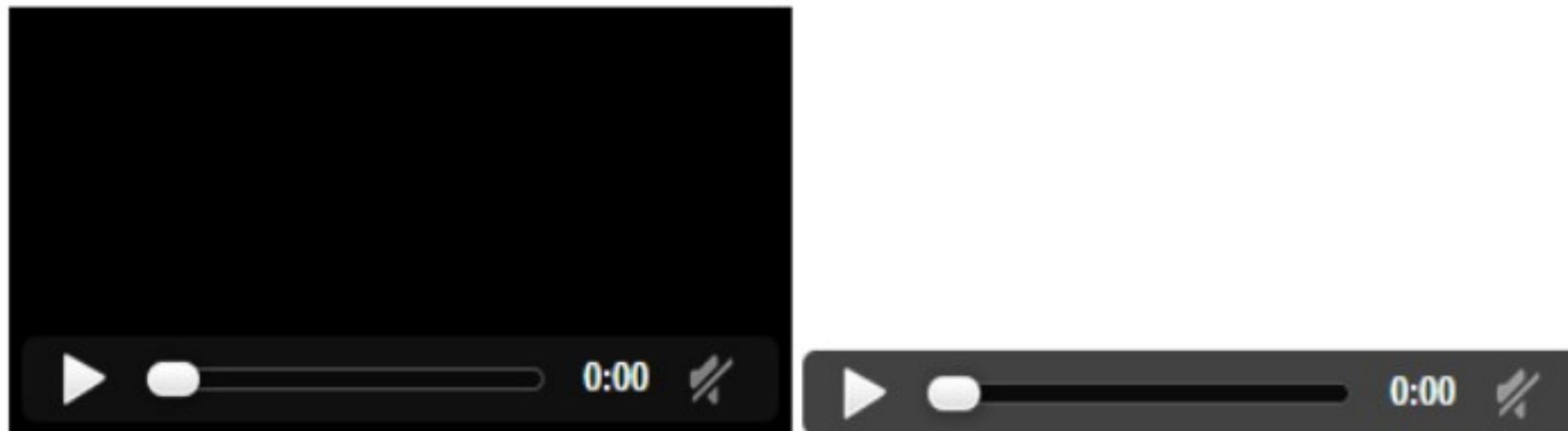
```
<video controls>
```

```
<source src="unit1material.webm" type="video/webm">
<source src="unit1material.mp4" type="video/mp4">
```

Your browser does not support the audio element.
</video>

```
<audio controls>
  <source src="unit1material.ogg" type="audio/ogg">
  <source src="unit1material.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

Output:



4) <svg> Element

- SVG નું પૂરું નામ Scalable Vector Graphics છે . Svg નો ઉપયોગ વેબ માટે vector-based graphics ને define કરવા માટે થાય છે.
- Svg XML format માં ગ્રાફિક્સને define કરે છે . Svg ગ્રાફિક્સ zoom કરેલું હોય અથવા તેનું કદ બદલ્યું છે તો તે કોઈપણ quality ગુમાવતું નથી.
- દરેક element અને દરેક attribute માં SVG ફાઇલો એનિમેટેડ હોઈ શકે છે .SVG એ W3C recommendation છે.
- SVG મોટાભાગે vector type diagrams માટે Pie charts, Two-dimensional graphs, X, Y coordinate system વગેરે માટે ઉપયોગી છે.
- SVGમાં કેટલાક predefined shape elements છે જેનો ઉપયોગ આપણા દ્વારા થઈ શકે છે.
 - Rectangle <rect>
 - Circle <circle>
 - Ellipse <ellipse>
 - Line <line>
 - Polyline <polyline>
 - Polygon <polygon>
 - Path <path>

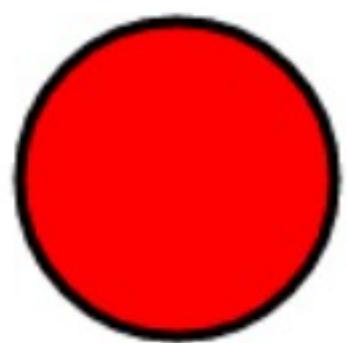
Example:

```
<h2>HTML5 SVG Rectangle</h2>
<svg height="100px" width="100px" xmlns="http://www.w3.org/2000/svg">
```

```
<circle cx="50px" cy="50px" r="40px" stroke="black" stroke-width="3" fill="red"/>
Sorry, your browser does not support inline SVG.
</svg>
```

Output:

HTML5 SVG Circle



Event List

- અમુક condition પૂરી થાય ત્યારે એક event start થાય છે. for example, page load, mouse click, key press અને form submission માટે events નીચેની ડોષકમાં આપેલી છે.

Attribute	Description	Supported HTML Tags	Syntax	Example
Onabort	જ્યારે user અથવા <input type="image"> element load ના થાય ત્યારે Onabort event occur થાય છે.	 and <input type="image">	<element onabort="myScript">	
Onafterprint	જ્યારે page printing કરવાનું start કરે અથવા print dialogue box close થાય ત્યારે Onafterprint attributes fire થાય છે.	<body>	<element onafterprint="script">	<body onafterprint="myFunction()"> </body><script>function myFunction(){ alert("This document is now being printed"); }</script>
Onplay	જ્યારે audio/video start કરવામાં આવે ત્યારે Onplay event occur થાય છે.	<audio>, <video>	<element onplay="myScript">	<video controls onplay="myFunction()"></video>
Onpause	જ્યારે audio/video ને user દ્વારા અથવા programmatically દ્વારા pause થાય ત્યારે Onpause event occur થાય છે.	<audio>, <video>	<element onpause="myScript">	<video width="320" height="240" controls onpause="myFunction()"></video>

Onprogress	જ્યારે browser specify audio/video ને download કરી રહ્યું હોય ત્યારે progress event થાય છે.	<audio>, <video>	<audio video onprogress= "myScript">	<pre><audio id="myAudio" controls> <source src="audio.ogg" type="audio/ogg"> <source src="audio.mp3" type="audio/mpeg"> Your browser does not support the audio element. </audio> <script> var aud = document.getElementById("m yAudio"); aud.onprogress = function() { alert("Downloading audio"); }; </script></pre>
Onscroll	Onscroll event ત્યારે થાય છે જ્યારે કોઈ element ના scrollbar ને scroll કરવામાં આવે.	<address>, <blockquote>, <body>, <caption> <center>, <dd>, <dir>, <div>, <dl>, <dt>, <fieldset>, <form>, <h1> to <h6>, <html>, , <menu>, <object>, , <p>, <pre>, <select>, <tbody>, <textarea>, , <tfoot>, <thead>	<element onscroll= "myScript">	<pre><div onscroll="myFunction()"></d iv></pre>
Onunload	Page ને Onunload કરવામાં આવે તે પછી Onunload થાય છે. (અથવા browser window બંધ(close) કરવામાં આવી છે.) જ્યારે user page પરથી દુર જાય છે ત્યારે (link પર click કરીને submit કરીને Onunload થાય છે. a form,	<body>	<element onunload= "myScript">	<pre><body onunload="myFuncio n()"></body></pre>

	closing the browser window etc.)			
Onvolumechange	video/audio નું volume change કરવામાં આવે ત્યારે Onvolumechange event occur થાય છે.	<audio> and <video>	<element onvolumechange="myScript">	<video controls onvolumechange="myFunction()"> <source src="video.mp4" type="video/mp4"> <source src="video.ogg" type="video/ogg"> Your browser does not support HTML5 video. </video>
Onwaiting	જ્યારે video બંધ થાય ત્યારે Onwaiting event occur થાય છે. કારણકે તેને next frame માં buffer ની જરૂર પડે છે. આ event નો use <audio> element પર પણ થઈ શકે છે, પરંતુ તે મોટાભાગે videos માટે ઉપયોગમાં લેવાય છે.	<audio>, <video>	<element onwaiting="myScript">	<video controls onwaiting="myFunction()"> <source src="video.mp4" type="video/mp4"> <source src="video.ogg" type="video/ogg"> Your browser does not support HTML5 video. </video>

Working with Canvas

- <canvas> element તમને javascript નો ઉપયોગ કરી ને તમને graphics દોરવાનું એક easy અને powerful રીત આપે છે.
- તે graphs પર, on the fly, photo compositions અથવા simple animations માટે ઉપયોગ કરી શકાય છે.
- <canvas> tag graphics માટે only એક container છે, તમારે actually graphics ને draw કરવા માટે script નો ઉપયોગ કરવો આવશ્યક છે.
- paths, boxes, circles, text, graphics images ને દોરવા માટે canvas મા ઘણી બધી method છે.

Setting up the canvas

- Canvas element સાથે web page બનવા canvas tag નો ઉપયોગ થાય છે. આપણે width, height અને id parameter provide કરીએ છીએ.
- canvas tag ની અંદર તમે ઈછો તો short message સાથે કોઈ પણ HTML code મૂકી શકો છો. જો browser canvas tag ને support નહિ કરે તો આ code દેખાશે.
- નીચે નું ઉદાહરણ એક simple <canvas> tag નું છે જે એક લંબચોરસ દોરે છે.

Example:

```
<canvas id="myCanvas" width="190px" height="140px" style="border:1px solid #d3d3d3;">
<p>Your browser does not support the HTML5 canvas tag.</p>
</canvas>
```

Controlling Fill and Stroke Styles

- Canvas માં only main બે function છે. Fill અને Stroke.
- stroke થી ફક્ત line draw થાય છે તેથી જો તમે લંબચોરસ ને stroke આપો તો એ તમને લંબચોરસ ની outline draw કરીને આપશો. પરંતુ તેમાં color fill નહિ થાય. fill એ shape માં fill કરીને આપશો. તેથી fill થયેલ લંબચોરસ એ લંબચોરસ ના interior ને બતાવશો.
- ત્યાં ત્રણા primary type ની style છે જેનો ઉપયોગ fills અને strokes માં કરી શકાય છે. color, gradient, pattern.

1) Colors

- canvas tag માં fill અથવા stroke color ને specify કરવા માટે, css અને HTML માં same color અને tools નો ઉપયોગ કરી શકીએ છીએ.
 - **Six-digit hex values:** તે સામાન્ય રીતે css માં વપરાય છે. તે red, green, blue ના બે digit નો ઉપયોગ કરે છે. ઉદાહરણ તરીકે, #FFF000 એ yellow.
 - **Three- digit hex values:** આ value ને three – digit number તરીકે ઉપયોગ કરી શકીએ છીએ. આ scheme માં red #F00, અને yellow #FF0.
 - **Color names:** તમે ધણીવાર color names જેમ કે "red" અથવા "yellow" નો ઉપયોગ કરી શકો છો. common color names usually work કરશો, પરંતુ બધા browser color names ના same list ને support કરતા નથી. જેમ કે "papaya whip" supported કરતું નથી.
 - **RGB and RGBA values:** interger(0-255) અથવા percentages(0%-100%) નો ઉપયોગ કરીને color બનાવવા માટે તમે rgb() function નો ઉપયોગ કરી શકો છો. Red rgb(255,0,0) અને yellow rgb(100%,100%,0%) છે. જો તમે alpha include કરવા માગો છો, તો fourth parameter add કરો, જેની 0 થી 1 value છે. Transparent red rgba(255,0,0,0.5) છે.
 - **HSL and HSLA (Hue, Saturation, Lightness and alpha):** નવા hsl અને hsla color formats canvas element દ્વારા supported છે. તે features જુદા જુદા browser દ્વારા બદલાય છે.

2) Gradients

- Gradients એ color ની pattern છે જે solid color ને બદલે shapes ના fill અથવા stroke માટે ઉપયોગ થાય છે. Gradients એ color ની pattern છે જે ધીમે ધીમે એક color થી બીજા માં બદલાતી જાય છે.
- Gradients ના બે પ્રકાર છે. **Linear** અને **Radial**
 - **A Linear Gradient** એ linear pattern સાથે horizontally, vertically અથવા diagonally color ને બદલે છે.
 - **A Radial Gradient** એ circular pattern સાથે color change કરે છે, inside અને outside માંથી color ને બદલે છે.

Syntax:

Radial Gradient: context.createRadialGradient(x0,y0,r0,x1,y1,r1);

Linear Gradient: context.createLinearGradient(x0,y0,x1,y1);

- The addColorStop() method એ gradient object માં color અને positions specify કરે છે.
- The addColorStop() method નો ઉપયોગ createLinearGradient() અથવા createRadialGradient() સાથે મળી ને થાય છે.

Syntax:

```
gradient.addColorStop(stop,color); //Stop value 0.0 અને 1.0 ની વચ્ચે હોય છે જે gradientમાં start
અને end વચ્ચે ની position આપે છે.
```

Example:

Radial Gradient

```
<canvas id="RGradient" width="200px"
height="150px"></canvas>
<script>
  var drawing =
    document.getElementById("RGradient");
  var context = drawing.getContext("2d");

  //build a linear gradient
  context.rect(0, 0, 200, 200);

  // create radial gradient
  var RadialGradient=
    context.createRadialGradient(100, 75, 0,
    100, 75, 75);

  // light blue
  RadialGradient.addColorStop(0,
  '#8ED6FF');
  // dark blue
  RadialGradient.addColorStop(1,
  '#004CB3');
  context.fillStyle = RadialGradient;
  context.fill();
</script>
```

Linear Gradient

```
<canvas id="LGradient" width="500px"
height="150px"></canvas>
<script>
  var drawing =
    document.getElementById("LGradient ");
  var context = drawing.getContext("2d");

  //build a linear gradient
  context.rect(0, 0, 200, 200);

  // add linear gradient
  var LinearGradient =
    context.createLinearGradient(0, 0, 100,
    100);

  // light blue
  LinearGradient.addColorStop(0,
  '#8ED6FF');
  // dark blue
  LinearGradient.addColorStop(1,
  '#004CB3');
  context.fillStyle = LinearGradient;
  context.fill();
</script>
```

Output:



Output:



3) Patterns

- Fill અથવા stroke તરીકે ઉપયોગ માં લેવાએલા image ને define કરવા માટે pattern નો ઉપયોગ કરવા માં આવે છે. HTML5 canvas સાથે એક pattern create થાય છે, આપણે canvas context ની createPattern() method નો ઉપયોગ કરી શકીએ છીએ જે pattern નો object return કરે છે. Pattern object ની fill style property set કરો અને પછી fill() નો ઉપયોગ કરી ને shape fill કરો.
- createpattern() method ને image object અને repeat option ની જરૂર છે. જે **repeat , repeat-x,repeat-y** અથવા **no-repeat** હોય શકે છે.

Syntax:

```
context.createPattern(image,"repeat|repeat-x|repeat-y|no-repeat");
```

Example:

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

<p><b>Image that you want to use:</b></p>

<p><b>Canvas:</b></p>
<canvas id="Pattern" width="250px" height="175px"></canvas>
<br>
<button onclick="draw('repeat')>Repeat</button>
<button onclick="draw('repeat-x')>Repeat-x</button>
<button onclick="draw('repeat-y')>Repeat-y</button>
<button onclick="draw('no-repeat')>No-repeat</button>

<script>
function draw(direction) {
    var drawing = document.getElementById("Pattern");
    var context = drawing.getContext("2d");
    context.clearRect(0, 0, drawing.width, drawing.height);
    var img = document.getElementById("imgPattern");
    var patt = context.createPattern(img, direction);
    context.rect(0, 0, 225, 155);
    context.fillStyle = patt;
    context.fill();
}
```

```

        }
</script>
</body>
</html>
```

Output:

Image that you want to use:



Canvas:



Drawing Essential Shapes

- કેટલાક primitive shapes સીધા graphics context પર directly draw કરી શકીએ છીએ. સૌથી common shapes rectangle અને text છે. જેની પાસે shadow પણ હોય શકે છે.

Rectangles

- HTML5 canvas નો ઉપયોગ કરી ને એક rectangle બનાવવા માટે આપણે rect() method નો ઉપયોગ કરી શકીએ છીએ.
- તમે અલગ અલગ પ્રકાર ના rectangle draw કરી શકો છો.
 - fillRect(x, y, width, height):** "filled" rectangle draw કરે છે.
 - strokeRect(x, y, width, height):** rectangle draw કરે છે (no fill).
 - clearRect(x, y, width, height):** આપેલ rectangle ની અંદર specify કરેલા pixels ને clear કરે છે.

JavaScript Syntax:

```
context.rect(x,y,width,height);
```

Example:

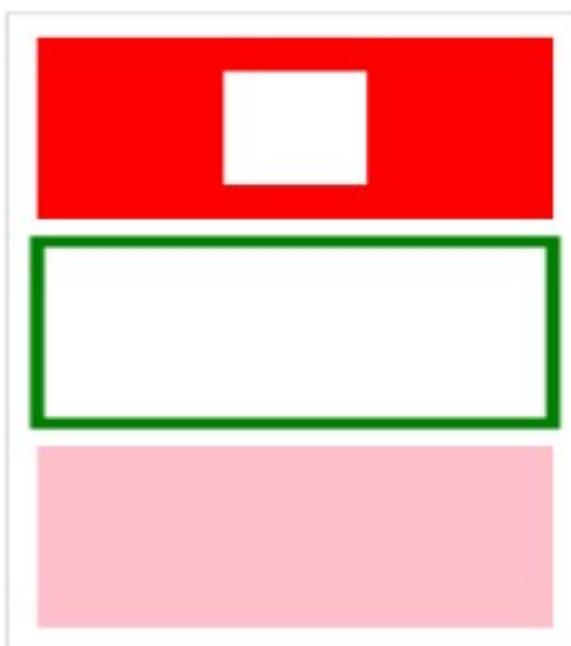
```
<canvas id="myCanvas" width="200px" height="280px" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>
<script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');
    context.fillStyle = "red";
    context.strokeStyle = "green";
    context.lineWidth = "5";
```

```

context.fillRect(10, 10, 180, 80);      //Using fillRect() Method
context.clearRect(75, 25, 50, 50);    //Using clearRect() Method
context.strokeRect(10, 100, 180, 80); //Using strokeRect() Method
context.rect(10, 190, 180, 80);       //Using rect() Method
context.fillStyle = 'pink';
context.fill();
</script>

```

Output:



Draw Text

- HTML5 canvas નો ઉપયોગ કરી ને text draw કરવા માટે font property અને **filltext() method** અથવા **stroketext() method** નો ઉપયોગ કરી શકીએ છીએ.
- filltext() method canvas પર fill કરેલ text draw કરે છે અને stroke text() method canvas પર text (with no fill) draw કરે છે. text નો default color black છે. આ method string x અને y position હશે.

JavaScript Syntax:

```
context.fillText(text,x,y,maxWidth);  or  context.strokeText(text,x,y,maxWidth);
```

Example:

```

<canvas id="myCanvas" width="578px" height="200px"></canvas>
<script>
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  context.font = '40pt Calibri';
  context.fillStyle = 'blue';           // Using fillText() method
  context.fillText('Hello World!', 150, 100);
  context.strokeStyle = 'blue';        // Using strokeText() method
  context.strokeText('Hello World!', 150, 150);
</script>

```

Output:

Hello World!
Hello World!

Enhancing shapes with shadows

- HTML5 canvas સાથે shadow add કરવા, **shadowColor**, **shadowBlur**, **shadowOffsetX** અને **shadowOffsetY** properties નો ઉપયોગ કરી શકીએ છીએ.
- shadowoffsetX** અને **shadowoffsetY** સેટ કરે છે કે shape માંથી shadow કેટલી દૂર સુધી ખેચી શકાય છે.
- shadowBlur property એ shadow માટે blur level set અથવા return કરે છે.
- shadowColor property shadow માટે color ને set અથવા return કરે છે.

Example:

```
<canvas id="shadow" width="200px" height="150px"></canvas>
<script>
var drawing = document.getElementById("shadow");
var context = drawing.getContext("2d");
context.fillStyle = "white";      //clear background
context.fillRect(0,0, 150, 150);
context.fillStyle = "red";        // draw font in red
context.font = "20pt sans-serif";
context.shadowOffsetX = 3;        //add shadows
context.shadowOffsetY = 3;
context.shadowColor = "black";
context.shadowBlur = 15;
context.fillText("DWSL Tutorial", 5, 100);
</script>
```

Output:

DWSL Tutorial

Drawing more complex shapes

- Path mechanism નો ઉપયોગ કરી ને તમે વધારે complex shapes create કરી શકો છો. Path એ graphics context દ્વારા play કરવા માં આવતી commands ની series છે.
- HTML5 canvas સાથે path create કરવા માટે આપણે multiple subpaths ને connect કરી શકીએ છીએ. દરેક નવા subpath નો ending point એ નવો context point બને છે.
- આપણે **lineTo()**, **arcTo()**, **quadraticCurveTo()** અને **bezierCurveTo()** methods નો ઉપયોગ દરેક subpath બનવા ઉપયોગ કરી શકીએ છીએ.

1) Line-drawing options

- Canvas પર line ને draw કરવા માટે ની જરૂરી method નીચે આપેલી છે.
 - beginPath()**: આ method current path ને reset કરે છે.
 - moveTo(x,y)**: આ method આપેલા point થી new subpath create કરે છે.
 - closePath()**: આ method current point થી starting point સુધી path create કરે છે.

- **fill()**: આ method current fill style સાથે subpaths ને fill કરે છે.
- **stroke()**: આ method current stroke style સાથે subpaths ને stroke કરે છે.
- **lineTo(x, y)**: આ method new point add કરે છે. અને તે point થી canvas માં last specific point સુધી line create કરે છે. (આ method line draw કરતાં નથી)
- ત્યાં ઘણી property છે જે stylelines ને allow કરે છે.
 - **linewidth**: આ property current line ની width ને pixels માં return કરે અથવા set કરે છે.
 - **line cap**: આ property line માટે styles ની end caps ને set કરે અથવા return કરે છે. line caps ની style એ butt ,round અને square છે.
 - **lineJoin**: આ property બતાવે છે કે તમારા path માં corners કેવી રીતે બતાવી છે. default form "miter" (જે shape corners બનાવે છે) છે. તમે "round" પણ પસંદ કરી શકો છો. (જે rounded corners આપે છે.) અને "level" (જે square corner બંધ કરે છે.) પણ પસંદ કરી શકે છે.
 - **miterLimit**: આ property maximum miter length ને set અથવા return કરે છે મિટર length એ outer corner અને inner corner વચ્ચે નું distance છે, જ્યાં બે line મળે છે.

Example:

```
<canvas id="LineDraw" width="250px" height="150px"></canvas>
<script>
    var drawing = document.getElementById("LineDraw");
    var context = drawing.getContext("2d");
    //change line width and color
    context.strokeStyle="green";
    context.lineWidth=10;

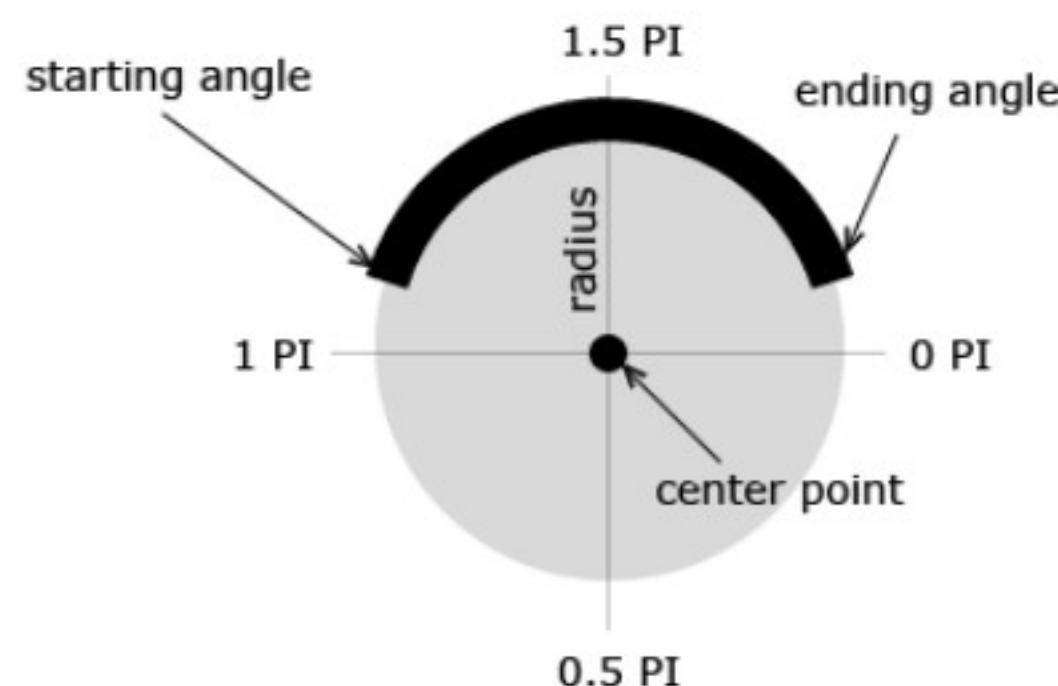
    context.lineJoin="round";
    context.beginPath();
    context.moveTo(40,40);
    context.lineTo(50,10);
    context.lineTo(60,40);
    context.stroke();
    context.closePath();
    //line caps
    context.lineCap="round";
    context.strokeStyle="blue";
    context.beginPath();
    context.moveTo(10,45);
    context.lineTo(90,45);
    context.stroke();
    context.closePath();
</script>
```

Output:



2) Making arcs and circles

- arc() method એ arc/curve (used to create circles, or parts of circles) create કરે છે.



- arc center point, radius, starting angle, ending angle અને drawing ની direction (clock wise અથવા anticlock wise) define કરે છે. Arcs ને linewidth, strokestyle અને linecap properties ની style આપી શકે છે.

JavaScript Syntax:

```
context.arc(x,y,r,StartingAngle,EndingAngle,Counterclockwise);
```

Example:

```
<canvas id="DrawArc" width="250px" height="300px"></canvas>
<script>
```

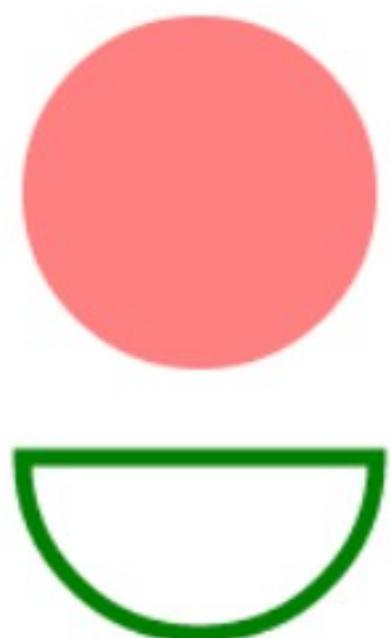
```
  var drawing = document.getElementById("DrawArc");
  var context = drawing.getContext("2d");
  context.strokeStyle="green";
  context.fillStyle="rgba(255,0,0,0.5)";
  context.lineWidth="5";
```

```
//half-circlestroked
context.beginPath();
context.arc(75,125,50,0,Math.PI,false);
context.closePath();
context.stroke();
```

```
//fullcirclefilled
context.beginPath();
context.arc(75,50,50,0,Math.PI*2,true);
context.closePath();
context.fill();
```

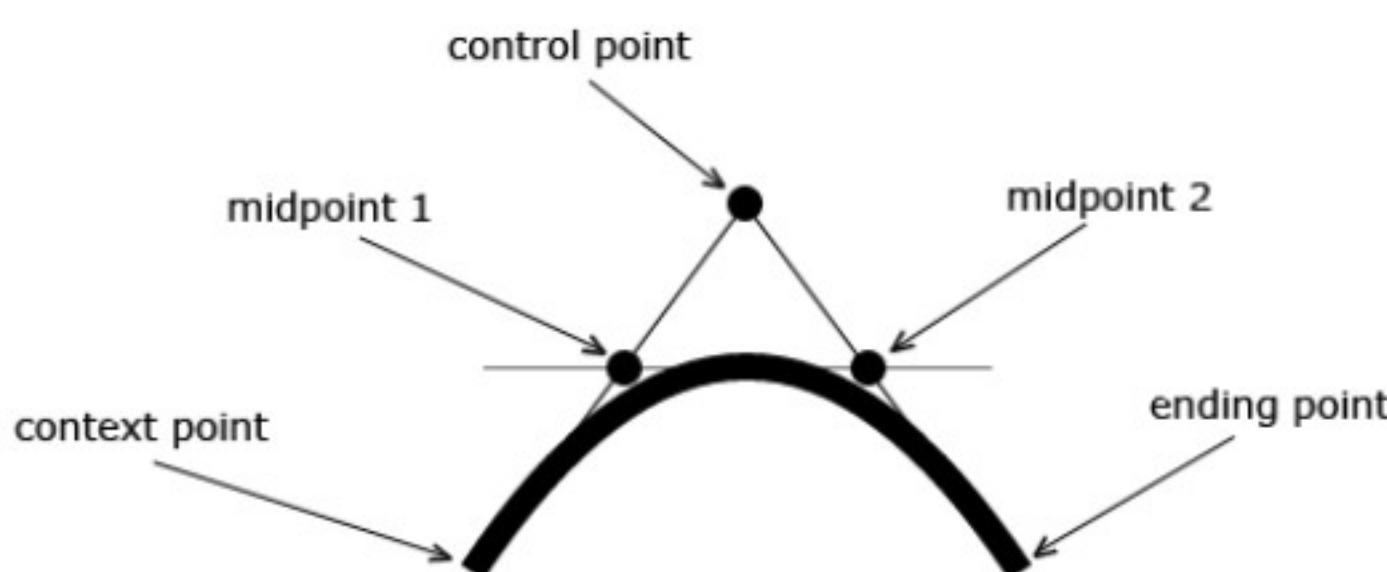
</script>

Output:



3) Drawing Quadratic Curves

- **quadraticCurveTo()** method એ specify કરેલ �control point દ્વારા current path થી point ને add કરે છે જે quadratic Bezier curve represent કરે છે.
- એક quadratic Bezier curve ને બે point ની requirement છે.
- પહેલા point એ control point છે. જેનો quadratic Bezier calculation માં ઉપયોગ થાય છે અને બીજો point curve ની ending point માટે ઉપયોગ થાય છે.
- Control point બે imaginary tangential lines બનાવીને તમારા quadratic curve ને define કરે છે. જે context point અને ending point સાથે જોડાયેલ છે. context point એ **moveTo()** દ્વારા define કરવામાં આવે છે.



JavaScript Syntax:

```
context.quadraticCurveTo(cpx,cpy,x,y); //cpx and cpy are the x and y-coordinate of the Bézier control point.
```

Example:

```
<canvas id="QuadraticCurve" width="250px" height="150px"></canvas>
<script>
var drawing = document.getElementById("QuadraticCurve");
var context = drawing.getContext("2d");
context.beginPath();
context.moveTo(20, 20);
context.quadraticCurveTo(20, 200, 200, 20);
context.lineWidth = 10;
```

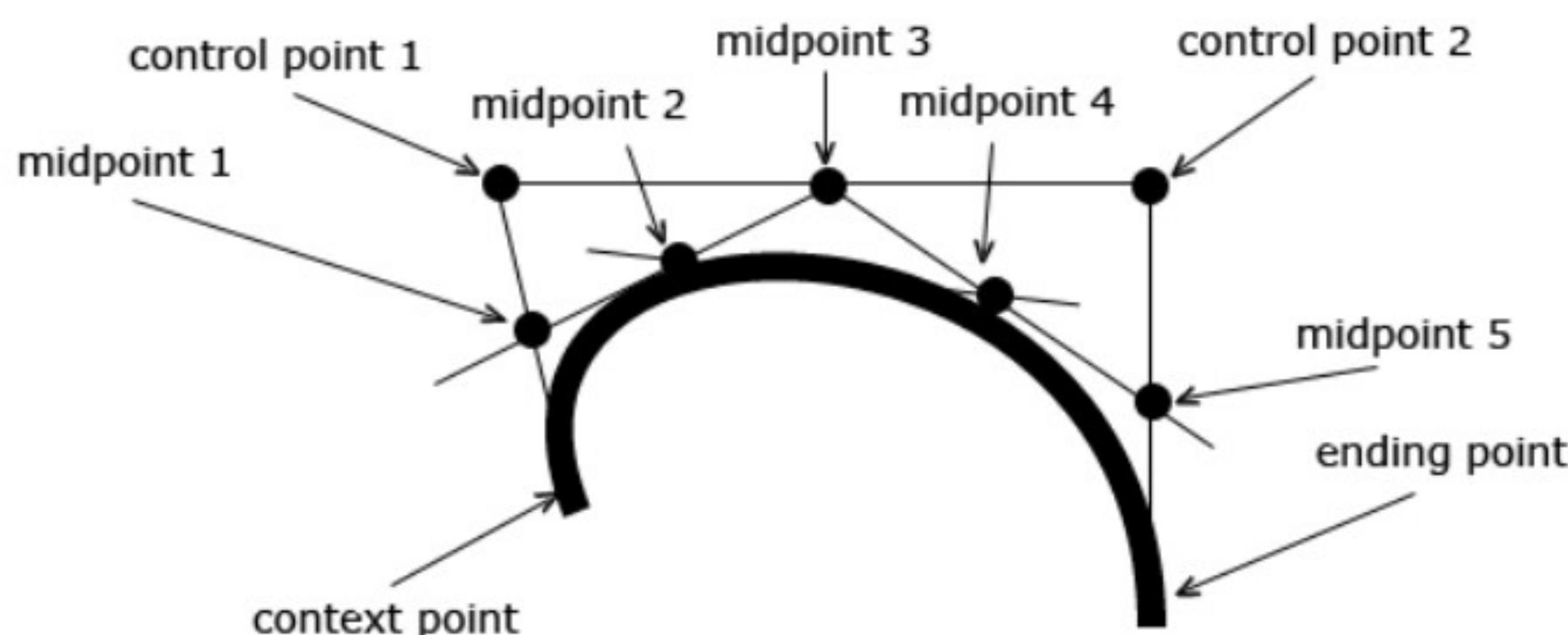
```
// line color
context.strokeStyle = 'black';
context.stroke();
</script>
```

Output:



4) Drawing a Bezier Curve

- **BezierCurveTo()** method એ specify control points દ્વારા current path ની points ને add કરે છે. અને એક cubic Bezier curve represent કરે છે.
- એક cubic bezier curve ને ત્રણ point ની જરૂર છે.
- પહેલા બે point એ control points છે. જેનો ઉપયોગ cubic Bezier calcultaion માં થાય છે અને છેલ્લો point curve ending point માટે ઉપયોગ થાય છે.
- જો કોઈ path exist નથી તો starting point ને define કરવા માટે **beginpath()** અને **moveTo()** method નો ઉપયોગ કરવામાં આવે છે.



JavaScript Syntax:

```
context.bezierCurveTo(cp1x,cp1y,cp2x,cp2y,x,y); // cp1x and cp1y are the x and y-coordinate of
the first Bézier control point. cp2x and cp2y are the x and y-coordinate of the second Bézier
control point.
```

Example:

```
<canvas id="BezierCurve" width="300px" height="150px"></canvas>
<script>
var drawing = document.getElementById("BezierCurve");
var context = drawing.getContext("2d");
context.beginPath();
context.moveTo(20, 20);
```

```

context.bezierCurveTo(20, 100, 200, 100, 200, 20);
context.lineWidth = 10;
// line color
context.strokeStyle = 'black';
context.stroke();
</script>

```

Output:



Formatting with CSS:

- HTML સામાન્ય રીતે information ને present કરવા માટે design કરવા માં આવી હતી. પરંતુ પાછળ થી web popular બન્યું. અને site ના content માં presentation આપવામાં આવે છે. જેથી CSS આવ્યું.

What is CSS? Or Define CSS.

- CSS નું પૂરું નામ Cascading Style Sheet થાય છે.
- HTML એલિમેન્ટ ને કેવી રીતે display કરવા તે CSS define કરે છે.
- CSS ની મદદ થી same content માં ધણી વાર ફેરફાર કર્યા વગર ઉયયોગ કરી શકીએ છીએ. જેનાથી time save થાય છે.
- Style Sheetએ પેજ ના content માં ચેંજ કર્યા વગર presentation ને specify કરવા સક્ષમ છે.

Uses of CSS

- CSSએ time ને save કરે છે કારણ કે કોઈ પણ element માટે આપણે content setting detail ને એક વાર specify કરીએ છીએ અને જ્યારે element બને ત્યારે તે CSS automatically apply થાય છે.
- CSS ની file size નાની હોય છે એના લીધે website ને load થવા માટે ઓછો time લાગે છે.
- CSSની મદદ થી web design flexible બનાવી શકો છો.
- CSS ને maintain કરવી easy છે કારણ કે તમે element ની CSS ને એક જગ્યા એ change કરો તો બધી જગ્યા એ તે element ની CSS change થઈ જાય છે.

Syntax of CSS

- CSS ની syntax ના ત્રણ ભાગ છે : Selector, Property અને Value.

Syntax:

```
selector {property: value;}
```

- Selector એ HTML element/tag છે .
- Property એ attribute છે જેને આપણે change કરી શકીએ છીએ અને દરેક attribute ને value હોય છે.
- Property અને value ને "," થી અલગ કરી શકાય છે.
- CSS નું declaration semicolon થી પૂરું થાય છે અને declaration groups curly braces થી પૂરા થાય છે.
- નીચેના ઉદાહરણ દ્વારા, બધા <p> tag લીલા કલર સાથે middle માં ગોઠવાયેલા હશે.

Example:

```
p {  
    color: green;  
    text-align: center; }
```

Types of CSS

- CSS ના ત્રણા type છે :
 - Inline style
 - Embedded/Internal style
 - External/Linked style sheet

1. Inline Style

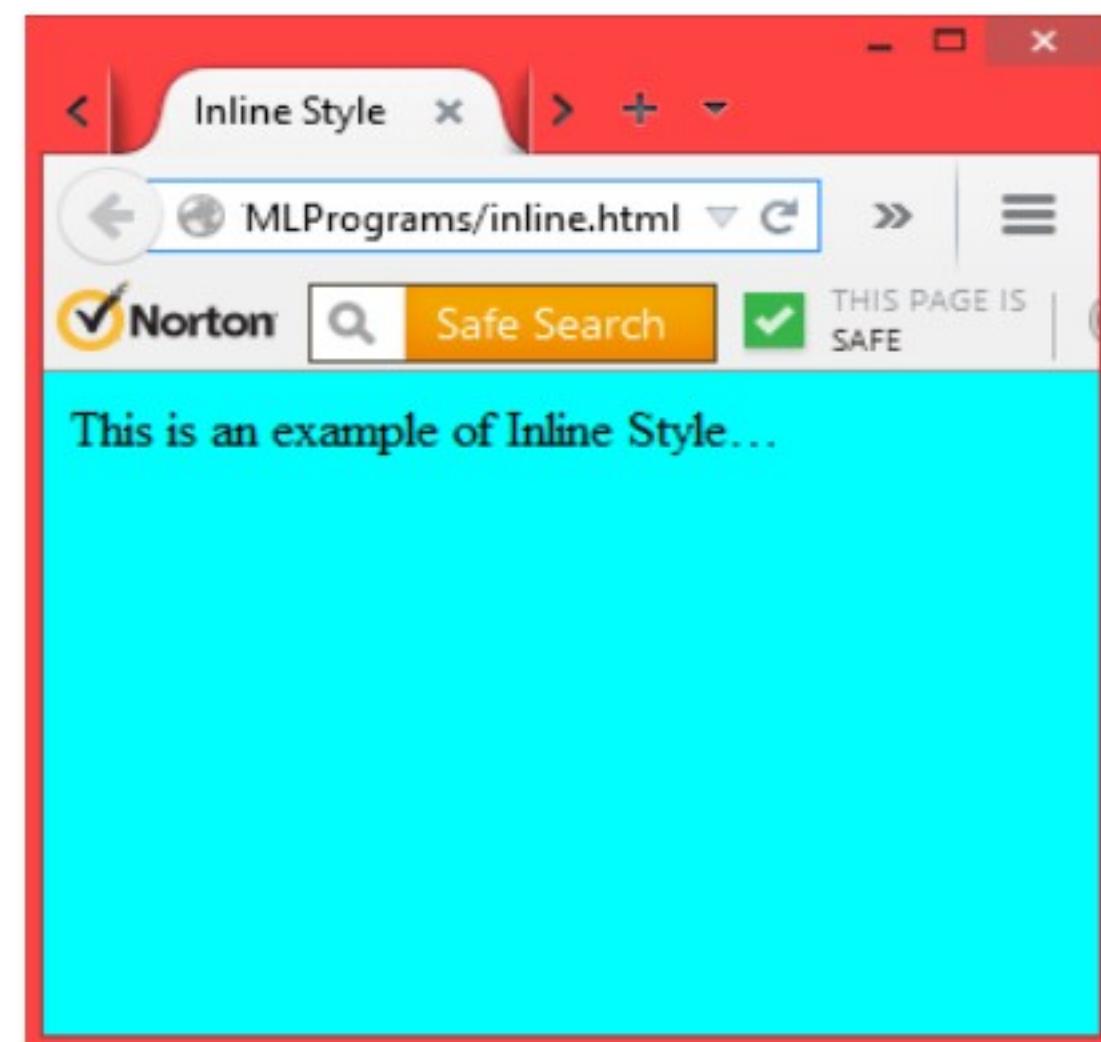
- Inline style એ એવી style છે જે ડોક્યુમેટ ના સીધા tag માં લખાય છે. Inline style ફક્ત tag ને જ એફેક્ટ કરે છે જેનો તે ઉપયોગ કરે છે.

Syntax:

```
<tag style="property1: value; property2: value;"> </tag>
```

Example: Using Inline Stylesheet

```
<html>  
<head>  
    <title>Inline Style</title>  
</head>  
<body bgcolor="#ffffff">  
    <p style="color:blue; font-family:arial;">  
        This is an example of Inline Style...  
    </p>  
</body>  
</html>
```



2. Embedded/Internal Style

- Embedded /Internal Style નો ઉપયોગ કરતી વખતે <head> tag ની અંદર <style> tag add કરવા માં આવે છે.
- Embedded style ફક્ત તે જ પેજ પર ના tag ને અસર છે જેમાં તેઓ embed કરેલા છે.

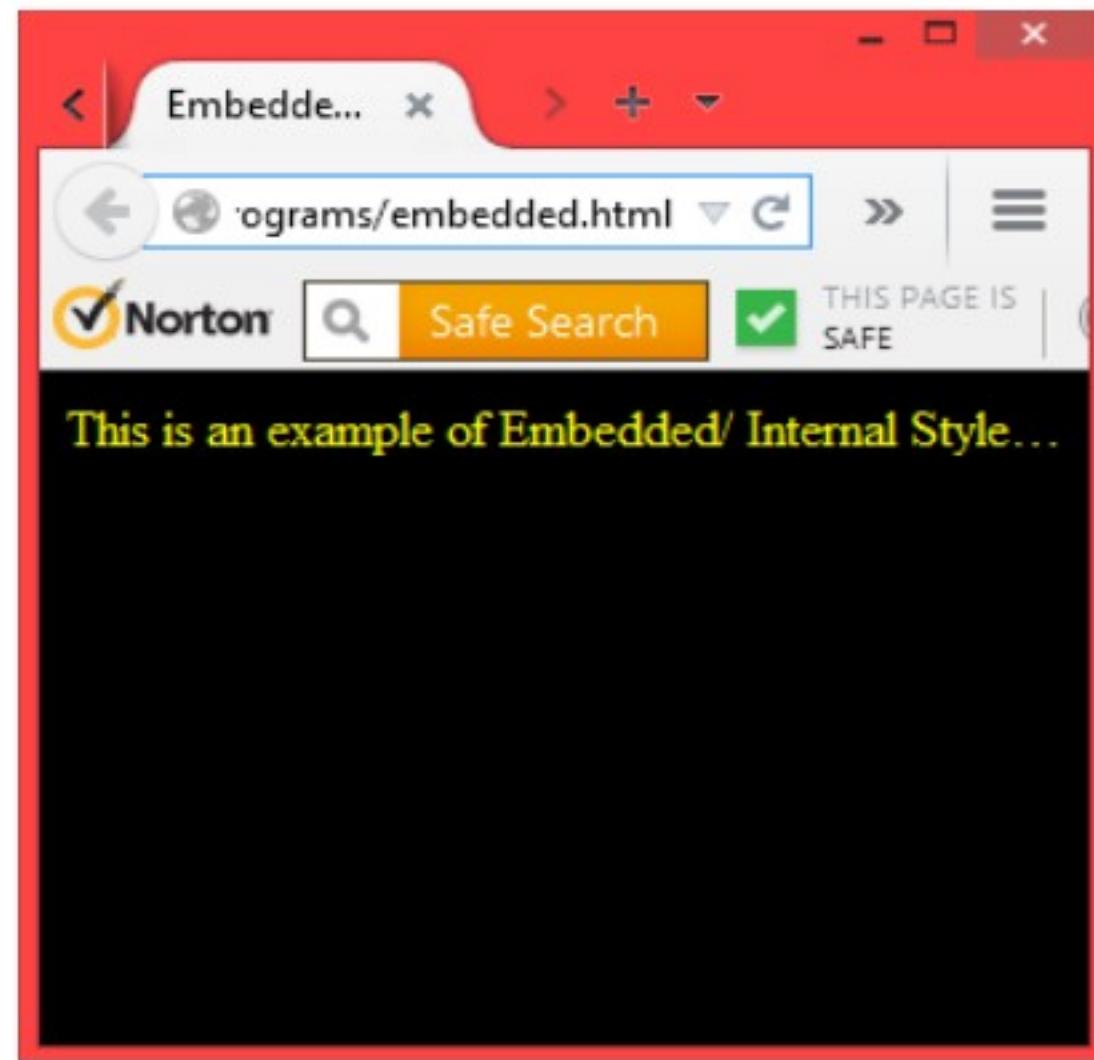
Syntax:

```
<style type="text/css">  
  
    Selector { property: value; }  
    Selector { property: value; }
```

.....
 </style>

Example: Embedding Stylesheet

```
<html>
<head>
    <title>Embedded Style</title>
    <style>
        p{color: yellow;}
        body{background-
color:black;}
    </style>
</head>
<body>
    <p>
        This is an example of Embedded/
Internal Style...
    </p>
</body>
</html>
```



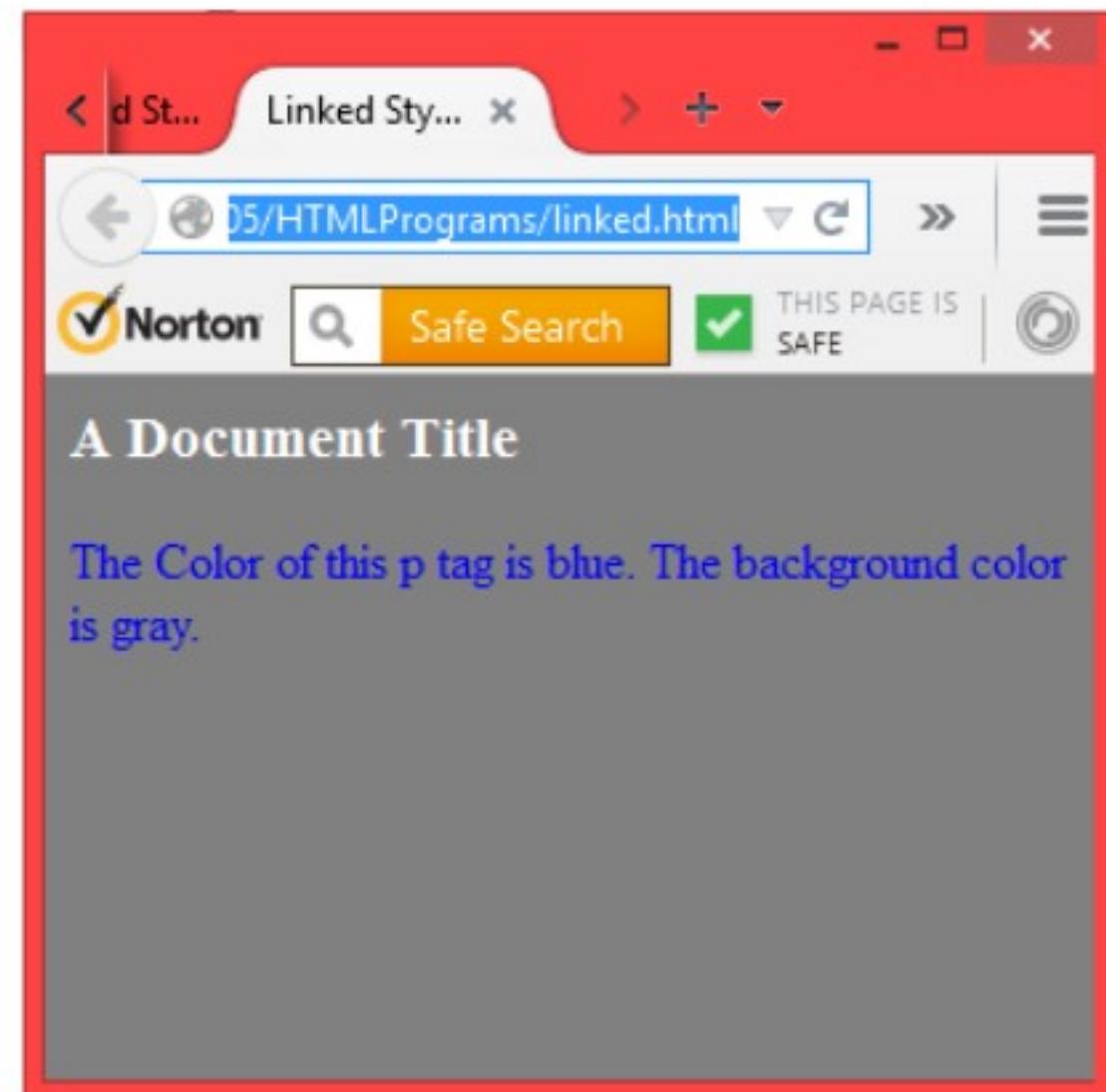
3. External/Linked Style Sheet

- External style sheet જુદા document માં લખવા માં આવે છે અને તે document ને attach કરવા માં આવે છે.
- External CSS file માં ખાલી CSS code જ હોય છે જેને ".CSS" extension થી save કરવા માં આવે છે.
- આ file ને HTML document માં <style> tag ની અંદર <link> tag નો ઉપયોગ કરી ને લખવામાં આવે છે.

```
body {
    background-color:gray;
}
p {
    color:blue;
}
h3 {
    color:white;
}
```

Example: Using Linked Stylesheet

```
<html>
<head>
    <title>Linked Style Sheets</title>
    <link rel="stylesheet"
        href="style.css" type="text/css"/>
</head>
<body>
    <h3>A Document Title</h3>
    <p>
        The Color of this p tag is blue. The
        background color is gray.
    </p>
</body>
</html>
```



CSS Margin

- CSS margin properties એલિમેન્ટ ની આસપાસની space ને ડિફાઇન કરે છે. તે Element ની આસપાસના areaને clear કરે છે. (બોર્ડર ની બહાર)
 - Margin ને background color હોતો નથી તે completely transparent હોય છે .
 - Margin માટે possible value:

Value	Description
Auto	Browser એ margin ની ગણતરી કરે છે.
Length	Margin px, pt, cm માં specify થાય છે .Default વેલ્યુ 0px.
% %	Element ની પહોળાઈ percentage માં specify થાય છે.
Inherit	Parent element માથી margin inherit થઈ સકે છે .

Example: Applying margins for paragraph tag.

```
p{  
    margin-top: 100px; margin-bottom: 100px; margin-right: 150px; margin-left: 150px;  
}
```

- Margin property પાસે એક થી ચાર value હોય શકે છે.

```
margin: 25px 50px 75px 100px; // top margin 25px, right margin 50px, bottom margin 75px  
      // left margin 100px
```

```
margin: 25px 50px 75px; // top margin 25px, right 50px, left margins 50px, bottom  
margin 75px
```

```
margin: 25px 50px;           // top અને bottom margins 25px, right અને left margins  

                            50px  

margin: 25px;                // બધી side ના margins 25px
```

CSS Padding

- CSS padding properties element ની border અને element ના content વાચે ની જગ્યા define કરે છે.
- Padding element ના content ની આસપાસ ની જગ્યા ને clear કરે છે. (બોર્ડર ની અંદર)
- Padding ને background color થી ફેર પડે છે.
- Padding માટે possible value:

Value	Description
Length	Padding px, pt, cm માં specify થાય છે.
%	Element ના content ને percentage માં specify કરાય છે.

Example: Applying padding for paragraph tag.

```
p {  

    padding-top: 100px; padding-bottom: 100px; padding-right: 150px; padding-left: 150px;  

}
```

- Padding property પાસે એક થી ચાર value હોય શકે છે.

```
padding: 25px 50px 75px 100px; // top padding 25px, right padding 50px, bottom padding  

                                75px, left padding is 100px
```

```
padding: 25px 50px 75px;      // top padding 25px, right અને left paddings 50px,  

                                bottom padding 75px
```

```
padding: 25px 50px;          // top અને bottom paddings 25px, right અને left paddings  

                                50px
```

```
padding: 25px;                // બધી side ના paddings 25px
```

CSS Text

All CSS Text Properties

Property	Description	Example
color	તેનો ઉપયોગ text ના color ને set કરવા માટે થાય છે. CSS સાથે color દ્વારા specify થાય છે: HEX value : "#ff00ff" RGB value : "rgb(255,0,0)" Color name : "black"	body { color: black; } h1 { color: #ff00ff; }
text-align	તેનો ઉપયોગ text ના horizontal alignment માટે થાય છે.	p.main

	<p>Text ને centered, અથવા aligned to the left or right, અથવા justified કરી શકાય છે.</p>	<pre>{ text-align: justify; } h1 { text-align: right; }</pre>
text-decoration	<p>તેનો ઉપયોગ text ના decoration ને remove કરવા માટે થાય છે. આ property પાસે નીચેની value છે.</p> <ul style="list-style-type: none"> (1) none: normal text ને define કરે છે. આ default છે. (2) underline: text ની નીચે line define કરે છે. (3) overline: text ની ઊપર line define કરે છે. (4) line-through: text પર line define કરે છે. 	<pre>a { text-decoration: none; } h1 { text-decoration: underline; }</pre>
text-transform	<p>તેનો ઉપયોગ text ની capitalization ને control કરવા માટે થાય છે. આ property ની value નીચે આપેલી છે:</p> <ul style="list-style-type: none"> (1) none: No capitalization. (2) capitalize: એક word નો first character capital હશે. (3) uppercase: બધા character uppercase માં થઈ જશે. (4) lowercase: બધા character lowercase માં થઈ જશે. 	<pre>p.uppercase { text-transform: uppercase; } p.lowercase { text-transform: lowercase; } p.capitalize { text-transform: capitalize; }</pre>
text-indent	<p>તેનો ઉપયોગ text-block માં first line નું indent ને specify કરવા માટે થાય છે.</p>	<pre>p { text-indent: 50px; }</pre>
text-shadow	<p>આ property text નો shadow add કરે છે તેની value નીચે આપેલી છે.</p> <ul style="list-style-type: none"> (1) h-shadow(Required): તે horizontal shadow ની position છે. (2) v-shadow(Required): તે vertical shadow ની position છે. (3) blur-radius(Optional): તે blur radius છે. Default value 0 છે. (4) color(Optional): તે shadow નો color છે. (5) none: Default value. No shadow. 	<pre>h1 { text-shadow: 2px 2px 8px #FF0000; }</pre>
letter-spacing	<p>આ property એ text માં character વિચેની space ને increases અથવા decreases કરવા માટે થાય છે.</p>	<pre>h1 { letter-spacing: 2px; } h2 { letter-spacing: -3px; }</pre>

line-height	આ property line ની height ને specify કરે છે.	p.small { line-height: 70%; }
--------------------	--	-------------------------------------

CSS Font

All CSS Text Properties

Property	Description	Example
font	એક declaration માં બધી font property set કરે છે.	p { font: italic bold 12px/30px Georgia, serif; }
font-family	Text માટે font family specify કરે છે. ત્યાં એ type ની font family name છે. (1) family-name like "times", "courier", "arial", etc. (2) generic-family like "serif", "sans-serif", etc.	p.serif { font-family: "Times New Roman", Times, serif; }
font-size	Text ની font size ને specify કરે છે. Font size માટે જુદી જુદી property value: medium, xx-small, x-small, small, large, x-large, xx-large, smaller, larger, length, initial and inherit	h1 { font-size:150%; } div { font-size:larger; }
font-style	Text ની font style ને specify કરે છે. Font style માટે જુદી જુદી property value: (1) normal : normal text બતાવે છે. (2) italic : italic text બતાવે છે. (3) oblique : આ text એ italic ને similar છે પરંતુ તે less supported છે.	p.normal { font-style: normal; }
font-variant	Small caps font માં text display થવો જોઈએ કે નહીં તે specify કરે છે. તેમાં ચાર property ની value છે: (1) normal : normal font display કરે છે. (2) small-caps : small-caps font display કરે છે. (3) initial : આ property default value set કરે છે. (4) inherit : Parent element માથી આ property inherits થાય છે.	p.normal { font-variant: normal; } p.small { font-variant: small-caps; }
font-weight	The font-weight property એ character કેટલો thick અને thin છે તે display કરે છે. આ property પાસે નિયેની value છે: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, initial and inherit.	p.normal { font-weight: normal; } p.thick { font-weight: bold; }

Making use of ids and classes with Example or CSS Selector

ID Selector

- યોક્કસ element શોધવા માટે" id "selector HTML tagના id attribute ઉપયોગ કરે છે.
- ID unique હોય છે .દરેક elementમાં ફક્ત એક જ ID હોઈ શકે છે અને દરેક pageમાં તે id સાથે ફક્ત એક element હોઈ શકે છે.
- Id element ની આગળ #(hash) લખવા માં આવે છે.

Class Selector

- કોઈ specific class સાથે element શોધવા માટે" class "selector નો ઉપયોગ થાય છે .Class selector html class attributes નો ઉપયોગ કરે છે.
- Class unique નથી હોતા .આપણે એક જ class ને multiple element પર ઉપયોગ કરી શકીએ છીએ અને આપણે same element માં multiple class નો ઉપયોગ કરી શકીએ છીએ.
- Class element ની આગળ .(period) લખવા માં આવે છે.

Example:

```

<html>
<head>
<style>
    #intro{
        color: yellow;
    }
    #intro1{
        color: hotpink;
    }
    .main{
        background-color: #013375;
        color: white;
        width:25%;
    }
</style>
</head>
<body>
    <div class="main">
        <p id="intro">Welcome to DWSL material webpage.</p>
        <p id="intro1">Learn the use of selector ID and Class.</p>
        <p>Thanking you.<p>
    </div>
</body>

```

</html>

Output:

Welcome to DWSL material webpage.

Learn the use of selector ID and Class.

Thanking you.

Using floating positioning and absolute positioning

- CSS Positioning properties નો ઉપયોગ element ને position આપવા માટે થાય છે.
- Elements પર top, bottom, right અને left properties નો ઉપયોગ કરી ને position આપી શકાય છે.

Using Floating positioning

- Float એ CSS ની position properties છે. Float css નો ઉપયોગ કરી ને element ને left અને right બાજુ ખસેડી શકાય છે. Float ધણી વાર image સાથે પણ વપરાય છે. અને layout સાથે work કરતી વખતે પણ float CSS વપરાય છે.
- Float સાથે ચાર જુદી જુદી value છે: left, right, none અને inherit.
- Clear property specify કરે છે કે બીજા floating elements કઈ side allow નથી.
- તેની પાસે પાંચ વેચ્યુ છે : left, right, both, none અને inherit.

Example:

Floating Elements next to each other

```
.thumbnail{
  float: left;
  width: 110px;
  height: 90px;
}
```

Turning off Float- Using Clear Property

```
.text_line{
  clear:both;
}
```

Using absolute positioning

- Absolute property નો ઉપયોગ webpage પર બે div ને overlap કરવા માટે થાય છે. Absolute position આપવા માટે parent element ને relative position અથવા static position આપવી પડે છે.
- Absolute position વાળા element બીજા element ને overlap કરી શકે છે.

Example:

```
h2 {
  position: absolute;
  left: 100px;
  top: 150px;
}
```

Overlapping Elements

- જ્યારે normal flow માં element ગોઠવામાં આવે ત્યારે તે બીજા element ને overlap કરી શકે છે. Z-index property specify કરે છે કે ક્યાં element ને આગળ, પાછળ કે પાછળ ના ભાગમાં મૂકવું જોઈએ.
- આ element પાસે positive અને negative stack order હોય છે.

Example:

```
img {
    position: absolute;
    left: 0px;
    top: 0px;
    z-index: -1;
}
```

New CSS3 selection tools:

- કેટલાક CSS3 ના મહત્વના elements: Selectors, Box Model, Backgrounds અને Borders, Image values અને Replaced content, Text effects, 2D/3D Transformations અને Animations.
- Traditional CSS એ tag type, id type અને class type ને allow કરે છે.
- ઉદાહરણ: fancy or #fancy.
- CSS3 નવી capability સાથે જુદા જુદા નવા selector ને support કરે છે.

Attribute selection

- CSS માં, html elements એ attributes અથવા attribute value ની style ને specify કરે છે. [attribute] selector નો ઉપયોગ specify attribute સાથે element ને select કરવા માટે છે.
 - [attr] : [attribute] selector નો ઉપયોગ specify attribute સાથે element ને select કરવા માટે થાય છે.
 - [attr=value] : [attribute=value] selector નો ઉપયોગ specify attribute અને value સાથે element ને select કરવા માટે થાય છે.
 - [attr^=value] : [attribute^=value] selector નો ઉપયોગ specify word ધરાવતી attribute ની value સાથે element ને select કરી શકીએ છીએ.
 - [attr|=value] : [attribute|=value] selector નો ઉપયોગ specify value થી start થતાં, specify attribute વાળા element ને select કરવા માટે થાય છે.
 - [attr\$=value] : [attribute\$=value] selector નો ઉપયોગ એવા element ને select કરવા માટે થાય છે કે જે attribute ની value specify value સાથે begin થાય છે.
 - [attr*=value] : [attribute*=value] selector નો ઉપયોગ એવા element ને select કરવા માટે થાય છે કે જેમની attribute value માં specify value હોય છે.

:nth-child() Selector

- :nth child ફોર્મુલા મુજબ તમે source order ના આધારે એક અથવા વધારે elements પસંદ કરી શકો છો.
- :nth-child(n) selector nth child ના દરેક element ને મેચ કરે છે. N એ નંબર, કીવર્ડ અથવા ફોર્મુલા હોય શકે છે.

Syntax:

```
:nth-child(number) {
    css declarations;
}
```

Example:

```
p:nth-child(2) {
    background: #ff0000;
}
```

- ઉપર ના code નો use કરીને આપણે દરેક <p> element માટે એક background color set કરી શકીએ છીએ. જે તેના parent ના બીજા child છે.
 tr:nth-child(2n+1) **OR** tr:nth-child(odd)
- HTML ટેબલ ની odd row ને represent કરે છે.
 tr:nth-child(2n) **OR** tr:nth-child(even)
- HTML ટેબલ ની even row ને represent કરે છે.

New pseudo-classes

- **A pseudo-class** એ element ના special state ને define કરવા માટે વપરાય છે.

ઉદાહરણ તરીકે:

- જ્યારે કોઈ element તેની ઉપર mouse over કરે ત્યારે style apply થાય છે.
- Visited અને unvisited links પર differently style apply કરો.

Syntax

```
selector:pseudo-class {
    property:value;
}
```

All CSS Pseudo Classes

Selector	Description	Example
:link	:link selector નો ઉપયોગ �unvisited link ને select કરવા માટે થાય છે.	a:link { background-color: yellow; }
:visited	:visited selector નો ઉપયોગ visited links ને select કરવા માટે થાય છે.	a:visited { background-color: yellow; }
:active	:active selector નો ઉપયોગ active link ને select અને કરવા માટે થાય છે. જ્યારે તમે click કરો છો ત્યારે link active બને છે.	a:active { background-color: yellow; }
:hover	:hover selector નો ઉપયોગ element ને select કરવા માટે થાય છે જ્યારે તમે mouse over કરો ત્યારે.	a:hover { background-color: yellow; }

:focus	:focus selector નો ઉપયોગ element ને select કરી તેના પર focus કરવા માટે થાય છે. Keyboard events અથવા બીજા user inputs એ element પર focus selector ને allow કરે છે.	input:focus { background-color: yellow; }
::first-letter	::first-letter selector નો ઉપયોગ specify selector નાલ first letter માં style add કરવા માટે થાય છે.	p::first-letter { font-size: 200%; color: #8A2BE2; }
::first-line	::first-line selector નો ઉપયોગ specify selector નાલ first line માં style add કરવા માટે થાય છે.	p::first-line { background-color: yellow; }
:first-child	:first-child selector નો ઉપયોગ specify selector ને select કરવા માટે થાય છે, ફક્ત જો તેના parent નું first child હોય.	p:first-child { background-color: yellow; }
::before	::before selector એ selected element ના content ની પહેલા content insert કરે છે.	p::before { content: "Read this:"; }
::after	::after selector એ selected element ના content ની પછી content insert કરે છે.	p::after { content: " - Remember this"; }
:lang (language)	:lang() selector નો ઉપયોગ specify value સાથે lang attribute વાળા element select કરવા માટે થાય છે.	p:lang(fr) { background: yellow; }

@font-face

- વેબ ફોન્ટ્સએ વેબ ડિઝાઇનર્સને ફોન્ટ્સનો ઉપયોગ કરવાની મંજૂરી આપે છે જે યુઝરના કમ્પ્યુટર પર ઇન્સ્ટોલ કરેલી નથી.
- જ્યારે તમે ઉપયોગમાં લેવા માંગતા હોય તે font હોય, તો તમારા વેબ સર્વર પર ફોન્ટ ફાઇલ શામેલ કરો અને જ્યારે જરૂર પડે ત્યારે તે આપમેળે ડાઉનલોડ થઈ જશે.
- તમારા "પોતાના" ફોન્ટ્સ CSS3 @ font-face ruleની અંદર define કરવામાં આવે છે.
- @font-face CSS at-rule authorsને તેમના વેબ page પર text display કરવા માટે online font ને specify કરે છે.

Syntax:

```
@font-face
{
    font-properties
}
```

CSS3 Font Descriptors

- નીચેના table ખધા font descriptors નું list કરે છે. જે @font-face rule ની અંદર define કરી શકાય છે.

Descriptor	Values	Description
font-family	Name	Required. Font નું name define કરે છે.
src	URL	Required. Remote font file location માટે નો URL, અથવા users ની computer પરના font નું નામ local સ્વરૂપમાં ("Font Name").
font-stretch	normal, condensed, ultra-condensed, extra-condensed, semi-condensed, expanded semi-expanded, extra-expanded, ultra-expanded	Optional. કેવી રીતે font ને stretched કરવો જોઈએ તે define કરે છે. Default value "normal" છે.
font-style	Normal, italic, oblique	Optional. કેવી રીતે font ને style આપવી જોઈએ તે define કરે છે. Default value "normal" છે.
font-weight	Normal, bold, 100, 200, 300, 400, 500, 600, 700, 800, 900	Optional. Font ની boldness ને define કરે છે. Default value "normal" છે.
unicode-range	unicode-range	Optional. Unicode characters ની range ને define કરે છે. Default value "U+0-10FFFF" છે.

Example:

```

<html>
<head>
<style>
    @font-face {
        font-family: MyWebFont;
        src: url(Raleway-Regular.ttf);
    }
    div {
        font-family: MyWebFont;
        font-size: 20px;
    }
</style>
</head>
<body>
    <div>The @font-face CSS rule allows authors to specify online fonts to display text on their web pages.
    </div>
</body>
</html>

```

Output:

The @font-face CSS rule allows authors to specify online fonts to display text on their web pages.

Column Support

- CSS ના અમુક rule સાથે, તમે **print-inspired layout** બનાવી શકો છો જેમાં web ની flexibility હોય.
- CSS3 multi-column layout એ text ની multiple column ની definition ને allow કરે છે. જેમ કે, newspaper
- CSS feature એ browser માં તમારો content કેવી રીતે print કરવામાં આવે છે તેના પર full control આપે છે અને તે properties નીચે આપેલી છે:
 - **column-count**: આ property નો ઉપયોગ કરીને, તમે element પર apply કરવા માંગો છો તે column ના number ને specify કરી શકો છો.
 - **column-width**: Single column ની width ને specify કરી શકો છો.
 - **columns** : આ property column width અને column count માટે ની shorthand property છે.
 - **column-gap**: Column વચ્ચે ના gap ની width ને specify કરે છે.
 - **column-rule-width**: border માટે ના rule નો part છે અને અહીં તમે તે border ની width ને specify કરી શકો છો.
 - **column-rule-style**: border ની જેમ style ને specify કરી શકો છો.
 - **column-rule-color**: color ના rule ને specify કરી શકો છો.
 - **column-span**: આ property ની value થી તમે કેટલી column નું span કરવા માંગો તે specify કરે છે.

Example:

In Body Tag:

```
<p class="FiveColumn">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularized in the 1960s with the release of Letterset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. </p>
```

In CSS:

```
.FiveColumn{
  -moz-column-count:5; // -moz prefix is used for Mozilla Firefox and -webkit
                      prefix is used for chrome, safari and opera browser.
  -moz-column-gap:40px;
  -webkit-column-count:5;
  -webkit-column-gap:40px;
  -moz-column-rule-color:#bbb;
  -moz-column-rule-style:solid;
  -moz-column-rule-width:2px;
  -webkit-column-rule-color:#bbb;
  -webkit-column-rule-style:solid;
}
```

```
-webkit-column-rule-width:2px;
text-align:justify;
```

Output:

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever

since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not

only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with

the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus

PageMaker including versions
Lorem Ipsum.

text-stroke Property

- **text-stroke** એ એક experimental property છે જે Adobe Illustrator અથવા બીજી vector drawing applicationsમાં મળતા જેવું જ �text decoration આપે છે.
- The **text-stroke** property એ બે બીજી property માટે shorthand property છે:
 - **text-stroke-width**, જે unit value લે છે (1px, 0.125em) અને stroke effect ની thickness ને describe કરે છે.
 - **text-stroke-color**, જે color value લે છે (hex, rgb/rgba).

Example:

In Body:

```
<p id="paragraph_decoration"> The text-stroke property provides text decoration</p>
```

In CSS:

```
#paragraph_decoration {
    -webkit-text-stroke: 2px brown;      // Only support in opera 15+, safari and chrome.
    font-size:20px;
}
```

text-shadow Property

- The **text-shadow property** એ text માં shadow ઓમેરે છે.

Syntax:

```
text-shadow: h-shadow v-shadow blur-radius color| none| initial| inherit;
```

- **h-shadow**: horizontal shadow. (Required)
- **v-shadow**: vertical shadow. (Required)
- **blur-radius**: blur radius. Default value 0. (Optional)
- **color**: color ની shadow. (Optional)
- **none**: Default value. shadow નહોં
- **initial**: આ property default value ને સેટ કરે.
- **Inherit**: parent element માંથી આ property ને inherits કરે.

Example:

In Body Tag:

```
<p id="textshadow">Apply Text Shadow to Paragraph Header</p>
```

In CSS:

```
#textshadow{
    text-shadow: 2px 2px 10px #FF0000;
    font-size:30px;
```

}

Flexible Box Layout Model

- **The Flexbox Layout module** નો aim container માં items ની વચ્ચે space distribute અને align કરવા માટે નો efficient way provide કરે છે. ભલે તેની size dynamic અથવા unknown હોય.
- **A flex container** એ available free space ને fill કરવા માટે expand કરે છે અથવા overflow રોકવા માટે તે સંકોચાણ (shrink) જાય છે.
- **Flexbox layout** એ application ના components અને small scale layout માટે સૌથી યોગ્ય છે જ્યારે grid layout એ large scale layout માટે બનાવાયેલ છે.
- Flexible-boxes ની **properties**:
 - **display**: Flex container ને define કરે છે; આપેલી value ને આધારે inline અથવા block કરવા. તે તેના બધા direct children માટે flex content enables છે.
 - **flex**: Flex property એ જ �container ની અંદર ની flexible item ની relative માં item ની length નો ઉલ્લેખ કરે છે. Flex property એ flex-grow, flex-shrink અને flex-basis માટે ની shorthand properties છે.
 - **flex-basis**: Flex-basis property એ flexible item ની initial length ને specify કરે છે.
 - **flex-direction**: Flex-direction property એ flexible items ની direction ને specify કરે છે.
 - **flex-flow**: Flex-flow property એ flex-direction અને flex-wrap ની shorthand property છે. Flex direction property એ flexible item ની direction ને specify કરે છે. Flexible wrap property specify કરે છે કે flexible item ને wrap કરવી જોઈએ કે નહીં.
 - **flex-grow**: Flex-grow property specify કરે છે કે same container ની અંદર બાકીની flexible item ની relative માં item કેટલી વધશે.
 - **flex-shrink**: Flex-shrink property specify કરે છે કે same container ની અંદર બાકીની flexible item ની relative માં item કેટલી સંકોચાશે.
 - **flex-wrap**: Flexible wrap property specify કરે છે કે flexible item ને wrap કરવી જોઈએ કે નહીં.
 - **justify-content**: આ main axis સાથે alignment ને define કરે છે. જ્યારે line પર ની બધી flex item inflexible હોય પરંતુ તેમના maximum size સુધી પહોંચી જાય ત્યારે બાકીની વધારા ની flex space ના distribution માં તે help કરે છે.

Example:

In Body Tag:

```
<p><strong>Simple Flexbox Example</strong></p>
<div class="flex">
    <div>Green</div>
    <div>White</div>
    <div>Red</div>
</div>
```

In CSS:

```
.flex
{
    width: 350px;
```

```

height: 200px;
border: 1px solid #555;
font: 14px Arial;
color:black;
font-weight:bold;
text-align:center;

/* flexbox setup */
display: -webkit-flex;
-webkit-flex-direction: row;

display: flex;
flex-direction: row;
}

.flex > div
{
    -webkit-flex: 1 1 auto;
    flex: 1 1 auto;
    width: 30px; /* To make the transition work nicely. */
    -webkit-transition: width 0.7s ease-out;
    transition: width 0.7s ease-out;
}

/* colors */
.flex > div:nth-child(1){ background : #009246; }
.flex > div:nth-child(2){ background : #F1F2F1; }
.flex > div:nth-child(3){ background : #CE2B37; }

.flex > div:hover
{
    width: 200px;
}

```

Output:

Simple Flexbox Example

Green	White	Red

- ઉપર નું example બતાવે છે કે કોઈ element પર "flexibility" કેવી રીતે લાગુ કરવી અને flexible state માં element ના behave ને કેવી રીતે બદલવું.

New Visual Elements:

Opacity (Transparency)

- Opacity property element માટે opacity level set કરે છે. CSS3 property માટે opacity એ transparency છે.
- Opacity property 0.0 થી 1.0 વિશે value લઈ શકે છે.
- Lower value element ને વધારે transparent બનાવે છે.
- Opacity level એ transparency level define કરે છે જ્યાં 1 value એટલે transparent નથી, 0.5 એટલે જોઈ શકાઈ એટલું transparent અને 0 value એટલે completely transparent.

Syntax:

```
opacity: number|initial|inherit;
```

Example:

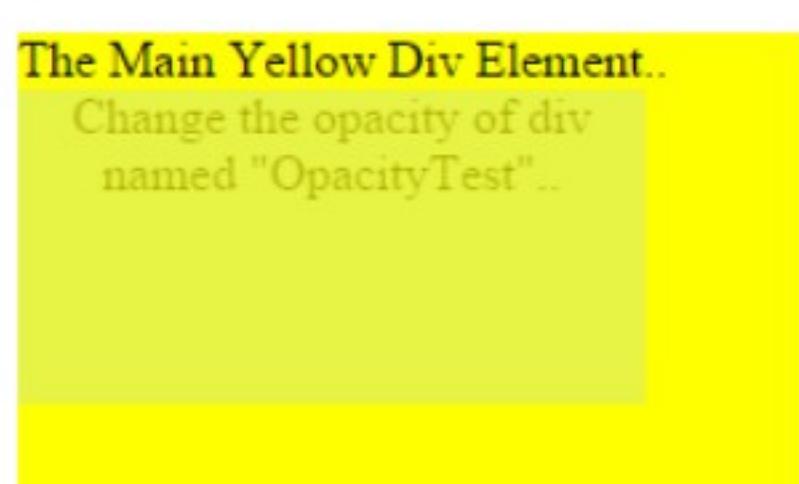
In Body Tag:

```
<div style="background:yellow; height:150px; width:250px">
  The Main Yellow Div Element
  <div id="OpacityTest">
    Change the opacity of div named "OpacityTest"
  </div>
</div>
```

In CSS:

```
#OpacityTest {
  background-color:lightblue;
  width:200px;
  height:100px;
  opacity:0.3;
  filter: alpha(opacity=30); /* For IE8 and earlier */
  text-align:center;
}
```

Output:



Border-radius (Rounded Corners)

- Border radius property ચારુ borders ની radius property ને set કરવા માટે use થાય છે. આ property element ની border ને round કરવા માટે allow કરે છે.

Syntax:

```
border-radius: 1-4 length|% / 1-4 length|%|initial|inherit;
```

Example:

In Body Tag:

```
<div id="BorderRadiusDemo">this is the demo example of the css property border-radius. </div>
```

In CSS:

#BorderRadiusDemo {	/* border-radius:10px;
border: 2px solid #000000;	is equivalent to:
padding: 10px 40px;	border-top-left-radius: 10px;
background: #ababab;	border-top-right-radius: 10px;
width: 300px;	border-bottom-right-radius: 10px;
border-radius: 10px;	border-bottom-left-radius: 10px; */
}	

Output:

This is the demo example of the css property border-radius.

Gradients

- Gradients** એ **color** ની **sequence** છે. simple gradients નો flow foreground થી background તરફ હોય છે. પરંતુ gradients contain માં બીજા ઘણા color હોય છે. CSS3 gradients તમને બે અથવા બે કરતા વધારે smooth transitions ને display કરે છે અથવા વધારે specified color દર્શાવે છે.
- CSS3 gradients નો ઉપયોગ કરીને તમે download time અને bandwidth નો ઉપયોગ ઘટાડી શકો છો અને elements જ્યારે zoom કરેલા હોય ત્યારે સારા ડેખાય છે. કારણકે browser દ્વારા gradients generate થાય છે.
- CSS એ બે પ્રકારના gradients define કરે છે.
 - Linear Gradient
 - Radial Gradient

1) Linear Gradients

- Linear Gradient નો flow straight line માં એક color માંથી બીજા color માં હોય છે.
- Linear gradient બનાવવા માટે તમારે બે color define કરવા પડે છે તે color ની વચ્ચે smooth transitions કરવા માટે starting point અને direction પણ set કરી શકાય છે. Gradient effect ની સાથે.

Syntax

```
background: linear-gradient(direction, color-stop1, color-stop2, ...);
```

Example:

In Body Tag:

```
<h4>Linear Gradient - Left to Right</h4>
<div id="LinearGradient"></div>
```

In CSS:

```
#LinearGradient {
    height: 200px;
    width: 200px;
    /* Internet Explorer 9 and earlier versions do not support gradients. */
    background: -webkit-linear-gradient(left, red, yellow) /* For Safari 5.1 to 6.0 */;
    background: -o-linear-gradient(right, red, yellow); /* For Opera 11.1 to 12.0 */;
    background: -moz-linear-gradient(right, red, yellow); /* For Firefox 3.6 to 15 */;
    background: linear-gradient(to right, red, yellow); /* Standard syntax (must be last) */
}
```

Output:

Linear Gradient - Left to Right



2) Radial Gradients

- **Radial Gradients** એ એક specify point થી concentrated હોય છે અને બીજી color visible છે. By default તેનો shape ellipse, Size એ farthest-corner, position center છે. Radial gradients ને create કરવા માટે બે color ને define કરવા પડે છે.

Syntax:

```
background: radial-gradient(shape size at position, start-color, ..., last-color);
```

Example:

In Body Tag:

```
<h4>Radial Gradient - Circle Shape</h4>
<div id="RadialGradient"></div>
```

In CSS:

```
#RadialGradient {
    height: 150px;
    width: 200px;
    /* Ellipse shape is by default. If you don't specify circle in this below example. */
    background: -webkit-radial-gradient(circle, pink, yellow, green);
```

```

background: -o-radial-gradient(circle, pink, yellow, green);
background: -moz-radial-gradient(circle, pink, yellow, green);
background: radial-gradient(circle, pink, yellow, green);
}
  
```

Output:

Radial Gradient - Circle Shape



Image Borders

- CSS3 નો નવો border feature **border-image property** છે. આ feature ની મદદ થી તમે border નો image તરીકે ઉપયોગ કરી શકો છો.
- Border એ **border-image-source**, **border-image-slice**, **border-image-width**, **border-image-outset** અને **border-image-repeat** properties એ set કરવા માટેની shorthand property છે.

Syntax:

```
border-image: source slice width outset repeat|initial|inherit;
```

Example:

In Body Tag:

```

<p>The below is the original Image:</p>

<p>The below two images are the example of border-image:</p>
<div id="round">Lorem ipsum dolor sit amet.</div><br/>
<div id="stretch">Lorem ipsum dolor sit amet.</div>
  
```

In CSS:

```

div {
  border: 15px solid transparent;
  width: 100px;
  padding: 10px 20px;
}

#round {
  -webkit-border-image: url(border.png) 30 30 round;      /* Safari 3.1-5 */
  -o-border-image: url(border.png) 30 30 round;          /* Opera 11-12.1 */
  border-image: url(border.png) 30 30 round;
}

#stretch {
  -webkit-border-image: url(border.png) 30 30 stretch;    /* Safari 3.1-5 */
  
```

```

-o-border-image: url(border.png) 30 30 stretch; /* Opera 11-12.1 */
border-image: url(border.png) 30 30 stretch;
}

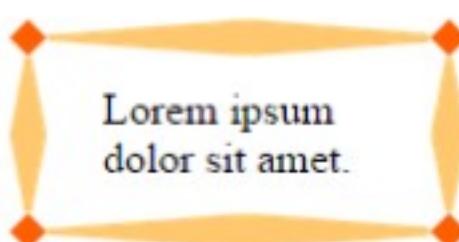
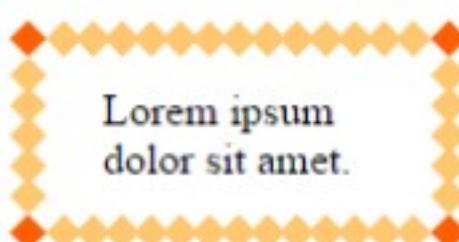
```

Output:

The below is the original Image:



The below two images are the example of border-image:



Reflections

- **Image reflection** એ CSS નો નવો feature છે જે તમને તમારી web image નીચે reflection add કરી ને આપે છે.
-webkit-box-reflect CSS property તમને એક element ના content ને એક specify direction માં reflect કરવા દે છે.

Syntax

```

-webkit-box-reflect: <direction> <offset> <mask-box-image>;

```

- **<direction>**: આ reflection ની position define કરીને આપે છે.
 - **Above** - Reflection એ border box ની ઉપર દેખાઈ છે.
 - **Below** - Reflection એ border box ની નીચે દેખાઈ છે.
 - **Left** - Reflection એ border box ની ડાબ્યો બાજુ દેખાઈ છે.
 - **Right** - Reflection એ border box ની જમણી બાજુ દેખાઈ છે.
- **<offset>**: box ની border થી reflection નું distance નક્કી કરે છે. Default value 0 છે.
- **<mask-box-image>**: box image ના reflection ને overlay કરવા માટે વપરાય છે. જો omit કરવામાં આવે તો reflection માં કોઈ mask નથી.

Example:

In Body Tag:

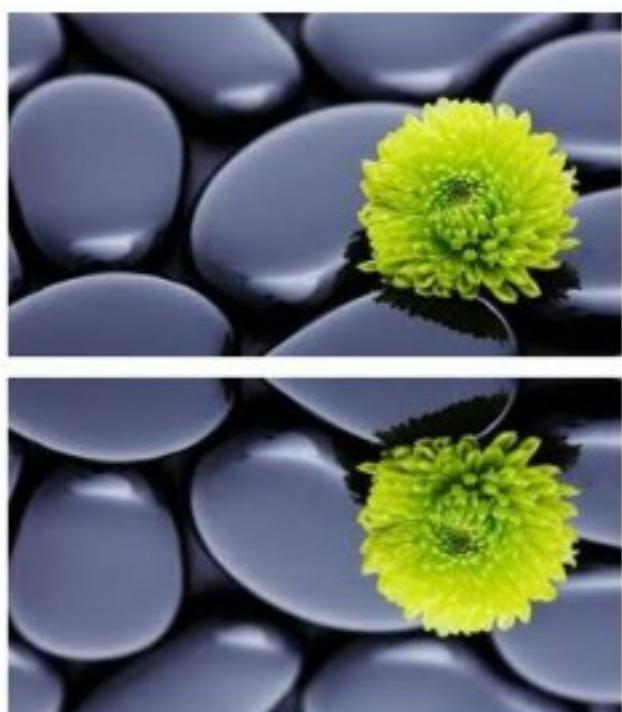
```

```

In CSS:

```
img {
    -webkit-box-reflect: below 10px;
}
```

Output:



Transformations

- Transformation એ એક effect છે જે element નો shape, size અને position માં change કરવા દે છે.
- The transform property એ element માં 2D અથવા 3D transformation apply કરે છે. આ property element ને rotate, scale, move, skewX, skewY અને skew કરવાનું allow કરે છે.
 - જ્યારે તમે કોઈ element માં transform apply કરો છો ત્યારે તમારે transformationની type ને describe કરવા માટે તમારે નીચેના એક અથવા વધારે parameter apply કરવા ની જરૂર છે:
 - **translate()**: આ method એ current position માથી element ને move કરે છે. (X-axis અને Y-axis માટે આપવામાં આવેલા parameter અનુસાર).
 - **rotate()**: આ method આપેલી degree અનુસાર elementને clockwise અથવા counter-clockwise ફેરવે છે. Negative value એ element ને counter-clockwise ફેરવે છે.
 - **scale()**: આ method element ની size ને increases અથવા decreases કરે છે.
 - **skew()**: આ method આપેલા angles દ્વારા X અને Y axis સાથે elements ને skew કરે છે.
 - **skewX()**: આ method આપેલા angles દ્વારા x – axis ની સાથે element ને skewX() કરે છે.
 - **skewY()**: આ method આપેલા angles દ્વારા y – axis ની સાથે element ને skewY() કરે છે
 - **matrix()**: આ method બધી 2D transform method ને એક માં combine કરે છે. આ method 6 parameter દે છે. જે elements ને rotate, scale, move (translate), અને skew કરવા નું allow કરે છે.

Syntax

transform: none | transform-functions | initial | inherit;

Example:

In Body Tag:

```
<div id="box1">Original Box</div>
<div id="box2">Translate</div>
<div id="box3">Rotate</div>
<div id="box4">Scale</div>
<div id="box5">Skew</div>
<div id="box6">Matrix</div>
```

In CSS:

```
#box1, #box2, #box3, #box4, #box5, #box6{
    width: 75px;
```

```

height: 40px;
border: 3px solid black;
background-color: yellow;
}

#box2 {
  transform: translate(100px,15px);
}

#box3 {
  transform: rotate(80deg);
}

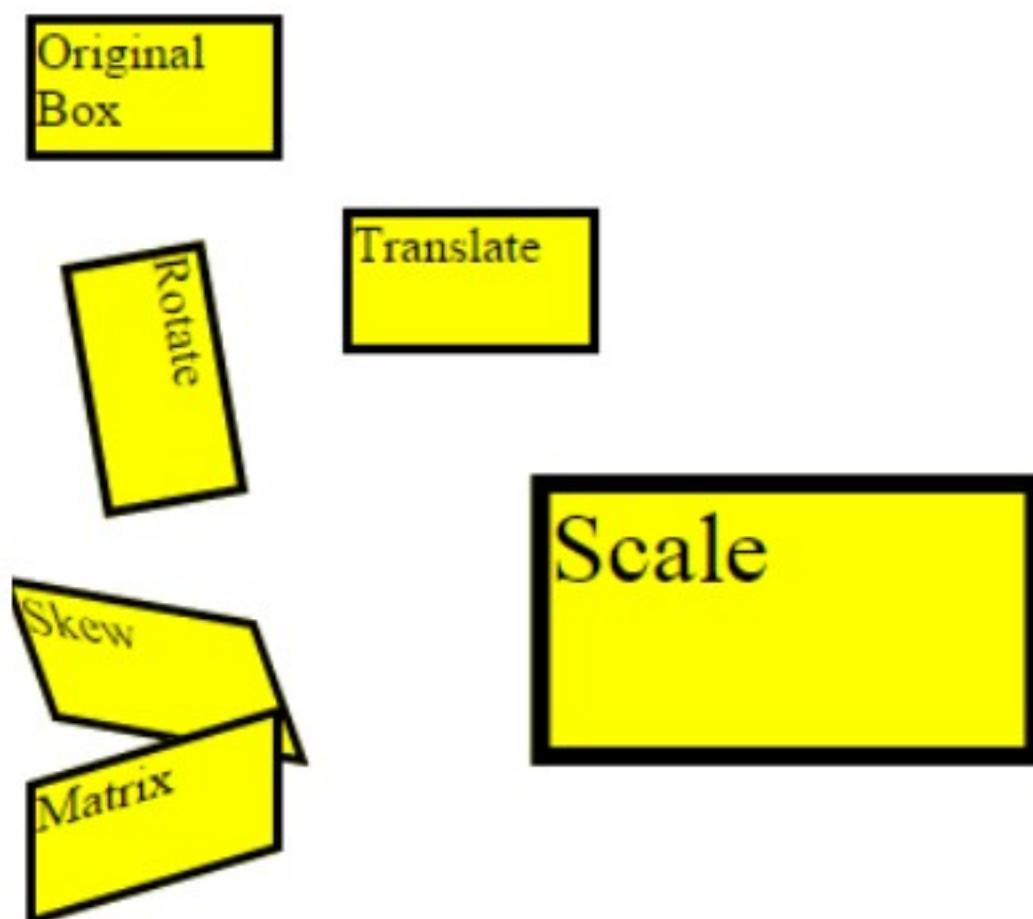
#box4 {
  transform: scale(2) translate(100px,15px);
}

#box5 {
  transform: skew(20deg,10deg);
}

#box6 {
  transform: matrix(1, -0.3, 0, 1, 0, 0);
}

```

Output:



Transition Animation

- **Animation** એ element ને ધીમે ધીમે એક style થી બીજામાં change કરી શકે છે.
- CSS3 animation નો ઉપયોગ કરવા માટે, તમારે પહેલા કેટલાક keyframes ને specify કરવા જરૂરી છે.
- **Keyframes** યોક્કસ time એ element ની style ને hold કરે છે. CSS3 JavaScript અને Flash નો ઉપયોગ કર્યા વગર મોટાભાગ ના HTML elementના animation ને allow કરે છે.
- **CSS3 transitions** એ property ની value ને smoothly (એક value થી બીજામાં) change કરે છે.
- **The transition attribute** એ ધણાલી parameter લે છે:

- **Animation property:** એ tag દ્વારા animation ની type define થાય છે. Default value બધી છે, પરંતુ other type, color, length, width, percentage, opacity અને number સહિતના બધા કામ કરે તેવી exaptation છે.
- **Duration:** secondsમાં animation ની length.
- **Timing function:** જો તમે ઈથ્થો છો કે animation constant speed એ થાય તો linear use કરો. જો તમારે વધારે natural motion જોઈએ છે જે ધીમે ધીમે animation ના અંત માં speed slow થઈ જાય તો નીચેના માથી એક નો ઉપયોગ કરો: **ease, ease-in, ease-out, ease-in-out.**
- **Delay:** જો તમે બીજી વાર value include કરો છો તો આને delay consider કરવામાં આવશે. આ પછી animation શરૂ થશે નહીં.

Example:

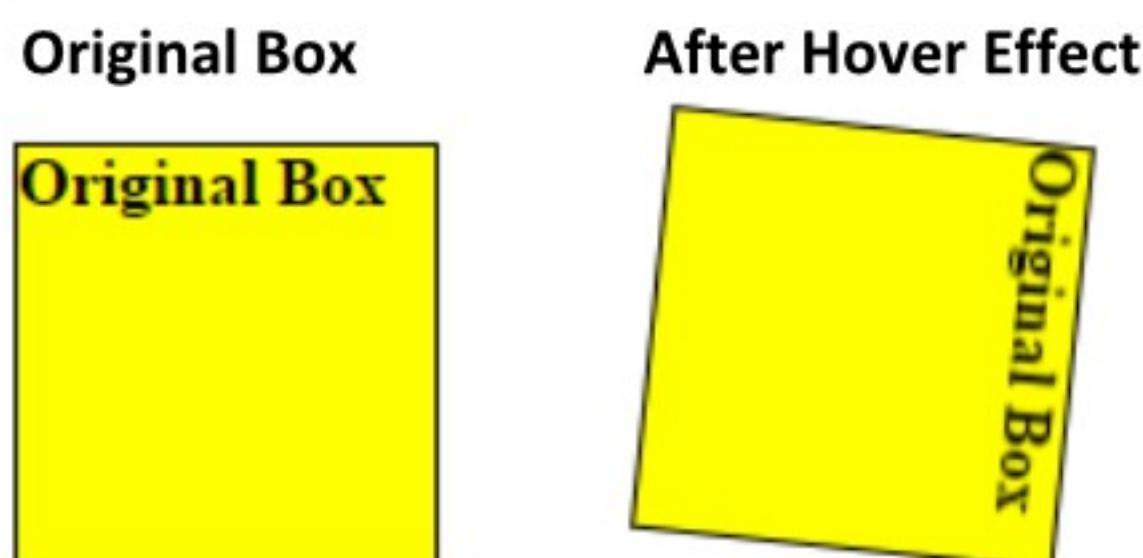
In Body Tag:

```
<div id="box"><b>Original Box</b></div>
```

In CSS:

```
#box {  
    transition: all 3s ease-in;  
    height: 100px;  
    width: 100px;  
    background-color: yellow;  
    border: 1px solid black;    }  
  
#box:hover {  
    transform: rotate(360deg);    }
```

Output:



Media Query - Responsive Design/Web page

- **CSS media queries** તમને HTML page પર બતાવે છે તે device પર આધાર રાખીને તમારા HTML page માં વિવિધ CSS style apply કરવા માટે સક્ષમ છે.
- Media queries તમને browser window ની width, device screen width, deviceની type aspect ratio અને pixel ratio ના આધારે જુદી જુદી style apply કરવા માટે સક્ષમ છે.
- **@media rule** નો ઉપયોગ કરીને, website માં screen, print, mobile phone, tablet વગેરે માટે એક અલગ layout હોય શકે છે.

CSS Syntax:

```
@media not|only mediatype and (media feature) {  
    CSS-Code;  
}
```

// You can also have different stylesheets for different media:

```
<link rel="stylesheet" media="mediatype and|not|only (media feature)" href="medial.css">
```

- **Only: only value** means એવી media query ફક્ત યોક્કસ media type સાથે match થાય છે.
- **Not: not value** means એવી media query યોક્કસ media type સિવાય ના તમામ સાથે match થાય છે.
- Media query syntaxમાં define [mediatype] token માટે, તમે નીચેની value માથી એકનો use કરી શકો છો.

Value	Description
all	Used for all media type devices.
aural	Deprecated. Used for speech and sound synthesizers.
braille	Deprecated. Used for braille tactile feedback devices.
embossed	Deprecated. Used for paged braille printers.
handheld	Deprecated. Used for small or handheld devices.
print	Used for printers.
projection	Deprecated. Used for projected presentations, like slides.
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud.
tty	Deprecated. Used for media using a fixed-pitch character grid, like teletypes and terminals.
tv	Deprecated. Used for television-type devices.

- Media query syntax માં define [media feature] token માટે, તમે નીચેની value માથી એકનો use કરી શકો છો.

Value	Description
aspect-ratio	Specifies the ratio between the width and the height of the display area.
device-aspect-ratio	Specifies the ratio between the width and the height of the device.
device-height	Specifies the height of the device, such as a computer screen.
device-width	Specifies the width of the device, such as a computer screen.
height	Specifies the height of the display area, such as a browser window.
max-aspect-ratio	Specifies the minimum ratio between the width and the height of the display area.
max-device-aspect-ratio	Specifies the minimum ratio between the width and the height of the device.
max-device-height	Specifies the maximum height of the device, such as a computer screen.
max-device-width	Specifies the maximum width of the device, such as a computer screen.

max-height	Specifies the maximum height of the display area, such as a browser window.
max-resolution	Specifies the maximum resolution of the device, using dpi or dpcm.
max-width	Specifies the maximum width of the display area, such as a browser window.
min-aspect-ratio	Specifies the minimum ratio between the width and the height of the display area.
min-device-aspect-ratio	Specifies the minimum ratio between the width and the height of the device.
min-device-width	Specifies the minimum width of the device, such as a computer screen.
min-device-height	Specifies the minimum height of the device, such as a computer screen.
min-height	Specifies the minimum height of the display area, such as a browser window.
min-resolution	Specifies the minimum resolution of the device, using dpi or dpcm.
min-width	Specifies the minimum width of the display area, such as a browser window.
orientation	Specifies the whether the display is in landscape mode or portrait mode.
resolution	Specifies the resolution of the device, using dpi or dpcm.
width	Specifies the width of the display area, such as a browser window.

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>

  @media only screen and (max-width:500px) {
    /* CSS rules for browser widths equal to or less than 500px */
    body {
      background-color: #0000ff;
      color:white;
    }
  }

  @media only screen and (min-width: 501px) and (max-width: 1000px) {
    /* CSS rules for browser widths from 501px to 1000 px */
    body {
      background-color: #ff0000;
    }
  }
}

```

```

        color:yellow;
    }

}

@media only screen and (min-width: 1001px) {
    /* CSS rules for browser widths from 1001px and up */
    body {
        background-color: #ffffff;
        color:brown;
    }
}

</style>
</head>
<body>
    <h2>The @media Rule</h2>
    <p><b>See it yourself !</b> resize the browser window width/height, and you will see the effect of media query.</p>
</body>
</html>
```

- ઉપર ના example માં **First media query** બધા browserની width ને 500 pixels ઓછી અથવા equal બનાવે છે. તેનો અર્થ એ છે કે media query ના body માં list બધા CSS rules (inside { ... }) ફક્ત ત્યારે apply કરવામાં આવશે જો browser window ની width 500 px અથવા તેનાથી ઓછી હોય.
- Second media query** window ની width ને 501px થી 1000 px સુધી match થાય છે. Media query ના body માં CSS rules ત્યારે જ �apply થશે જો browser window ની પહોળાં આ interval માં આવે.
- Third media query** 1001 px અને તેથી વધુ browser window ની width સાથે match થાય છે. Media query ના body માં CSS rule ત્યારે જ �apply થશે જો browser window ની width 1001px અથવા તેથી વધુ હશે.

JavaScript concept or what is JavaScript?

- **JavaScript** એ web ની programming language છે. તે object-oriented language છે જે interactive web pages બનાવવાનું allow કરે છે.
- તે સામાન્ય રીતે web browser ના ભાગ રૂપે ઉપયોગમાં લેવાય છે. જેનું implementation client-side scripts ને user સાથે interact કરવાનું allow કરે છે, browser ને control કરે છે, asynchronously રીતે communicate કરે છે, અને display કરેલા document content માં ફેરફાર (alter) કરે છે.
- તે server-side network programming માં runtime એ પણ use થાય છે.
- Javascript ECMA Script language standard એ implementation કરે છે. ECMA-262 એ official javascript standard છે.
- Javascript એ:
 - Javascript એ lightweight, interpreted programming language છે.
 - Network-centric application માટે design create કરે છે.
 - Java સાથે complementary અને integrate છે.
 - HTML સાથે complementary અને integrate છે.
 - Open અને cross-platform છે.
- તે **case sensitive scripting language** છે.

Origin of JavaScript

- 1995 માં Netscape અને sun microsystem દ્વારા javascript બહાર પાડવામાં આવી હતી. જોકે javascript એ java જેવી same નથી.
- Javascript અને java બંને concept અને design માં completely જુદી-જુદી languages છે.
- **Brendan Eich** 90 ના દાયકા માં Netscape માટે javascript develope કરી હતી. અને તે 1994 માં Netscape 2 ના early version સાથે include કરવામાં આવ્યું હતું અને 1997 માં ECMA એ standard બનાવ્યું હતું.
- **Development** દરમિયાન javascript ને Mocha અને પછી Livescript કહેવામાં આવતું હતું.

Why we use and learn JavaScript?

- Javascript એ 3 language માં ની એક છે. જે બધા web developer ને શીખવી આવશ્યક છે.
 - 1) Web page ના **content** ને define કરવા માટે **HTML**.
 - 2) Web page ના **layout** ને specify કરવા માટે **CSS**.
 - 3) Web page ના **behavior** ને program કરવા માટે **javascript**.

Advantages of JavaScript

- **Speed:** Client-side હોવાને કારણે, JavaScript ખુબં fast છે. કારણેકે server નો contact કરવાને બદલે કોઈ પણ code function થી તરત જ રૂન કરી શકાય છે. અને answer નો wait કરવો પડતો નથી.
- **Less server interaction:** તમે server પર page ને મોકલતા પહેલા �user input ને validate કરી શકે છો. આ �server traffic ને save કરે છો, જેનો meaning તમારા server પર ઓછો load છે.

- **Simplicity:** Javascript એ શીખવા અને implement કરવા માટે પ્રમાણ માં સરળ છે.
- **Versatility:** Javascript બીજી languages સા�ે સારી રીતે work કરી શકે છે અને વિવિધ પ્રકારની application માં તેનો ઉપયોગ કરી શકાય છે. javascript અને અન્ય language માં લખેલી script ની અંદર પણ javascript નો ઉપયોગ કરી શકાય છે.
- **Server Load:** Client-side હોવાથી website server પર ઉપયોગ ઓછો થાય છે.
- **Immediate feedback to the visitors:** તેઓ કોઈ page ને reload કરતા ભૂલી ગયા છે કે નહિ તે જોવા માટે page ફરીથી load કરવા માટે રાહ જોવી પડતી નથી.
- **Richer interfaces:** તમે તમારા site ના visitors ને સારો interface આપવા માટે drag-and-drop component અને sliders જેવી વસ્તુઓ સામેલ કરવા માટે javascript નો ઉપયોગ કરી શકો છો.
- **Easy Debugging and Testing:** Javascript એ line by line થી test કરવામાં આવે છે. તેથી error શોધવામાં, changes કરવામાં અને overhead વગર test કરવું અને delay કરવામાં compiling કરવું સરળ છે.

JavaScript Syntax

- Javascript માં javascript statements સામેલ હોય છે. જે web page માં <script>....</script> HTML tags માં મુકવામાં આવે છે.
- તમે તમારી script ને <script> tag મૂકી શકો છો જે તમારી javascript ને web page ની અંદર ગમે ત્યાં <head> tags માં મુકવામાં આવે છે.

Syntax:

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

- Script tag ના બે important attribute:
 - **language:** આ attribute તમે કઈ scripting language નો ઉપયોગ કરી રહા છો. તે specify કરે છે. સામાન્ય રીતે તેની value javascript હશે.
 - **type:** તેની value "text/javascript" set હોવી જોઈએ.

Example:

```
<html>
    <head>
        <title>Javascript Program</title>
        <script language="javascript" type="text/javascript">
            document.write("Hello World!");
        </script>
    </head>
</html>
```

Output:

Hello World!

JavaScript Variables

- Javascript variable એ data value ને store કરવા માટેનું container છે. તમે આ container માં data મૂકી શકો છો અને પછી container ને નામ આપી ને એના data દ્વારા refer થાય છે.
- તમે javascript program માં variable નો ઉપયોગ કરો તે પેહલા, તમારે તેને declare કરવું જરૂરી છે. **var** keyword સાથે variable નીચે મુજબ છે:

```
<script type="text/javascript">
    var money;
    var name;
</script>
```

- Var keyword નો ઉપયોગ ફક્ત declaration અથવા initialization માટે કરો
- Javascript **untyped language** છે. આનો અર્થ એ છે કે javascript variable કોઈ પણ data type ની value રાખી શકે છે.

JavaScript Variable Scope

- Variable નો scope એ તમારા program નો region define કરે છે.
 - **Global Variables:** Global scope નો javascript code માં તમે દરેક જગ્યાએ define કરી શકો છો.
 - **Local variables:** તે function માં જ �visible હશે જ્યાં તે define કરવામાં આવે છે. function parameters હમેશા તે function માટે local હોય છે.
- Global variable, effectively રીતે global variable નો hide કરો છો. જે ઉદાહરણ નીચે મુજબ છે.

Example

```
<html>
    <head>
        <title>Javascript Program</title>
        <script type="text/javascript">
            var demo = "global";           // Declare a global variable
            function checkscope( )
            {
                var demo = "local";     // Declare a local variable
                document.write(demo);
            }
        </script></head>
        <body>
            <button onclick="checkscope();">Click Me</button>
        </body>
    </html>
```

Output:

local

JavaScript Identifiers

- બધા javascript variable પાસે unique નામ હોય છે. આ unique નામ ને identifiers કહેવામાં આવે છે. javascript identifiers case-sensitive છે.
- variable (unique identifiers) ના નામ બનાવવા માટેના સામાન્ય નિયમ નીચે આપેલા છે.
 - નામ માં letters, digits, underscores અને dollar signs સામેલ હોઈ શકે છે.
 - નામ letter થી શરૂ થવું જરૂરી છે.
 - નામ \$ અને _ થી પણ start થઇ શકે છે.
 - નામ case sensitive ('a' અને 'A' જુદા variable છે.)
 - Reserved words (જેમ કે javascript keywords) નામ તરીકે ઉપયોગ કરી શકતા નથી.

JavaScript Reserved Words

- JavaScript ના reserved words નીચે આપેલા છે. તે JavaScript variables, functions, methods, loop labels અથવા કોઈ object name માં ઉપયોગ કરી શકતા નથી.

abstract	else	instanceof	switch
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof
continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	import	static	with
double	in	super	

Table 2.1: JavaScript reserved keywords

JavaScript Data Types

- Data type એ data ની type નું classification છે. જે variable અથવા object ને computer programming માં hold કરે છે.
- Javascript એ ત્રણી **primitive data type** સાથે work કરે છે: **Number, String, Boolean**
 - Numbers:** Number એ javascript માં 64-bit નું floating દશાવે છે. javascript એ integer અને floating-point numbers વચ્ચે નથી. javascript માં integer અને floating point numbers વચ્ચે કોઈ પણ તફાવત નથી. ઉદાહરણ માટે:

1

3.14

.0001

6.02e23

- **Strings:** Javascript ના programs માં single અથવા double quotes ની વચ્ચે string આવેલી હોઈ શકે છે. ઉદાહરણ માટે:
 - 'name = "MyData" '.
 - 'testing'
 - "3. 14"
 - "Wouldn't you prefer O'Reilly's book?"
- **Boolean:** Boolean type માં બે possible value હોઈ શકે છે, જેમ કે true અને false.
- Javascript એ બે **trivial datatype** define કરે છે: **null** અને **undefined**.
- **Null:** JavaScript keyword null એ special value છે જે "no value" બતાવે શકે છે.
- **Undefined:** JavaScript માં, variable પાસે value હોતી નથી તેની value undefined હોય શકે છે. Typeof એ undefined શકે છે.
- Primitive data types માં javascript એ **composite data type** ને support કરે છે. જેમ કે **object** અને **array**
 - **Object:** Javascript object ને braces સાથે લખવામાં આવે શકે છે. object ની property એ **name:value** pairs માં હોઈ શકે છે. જે (,) દ્વારા છુટા પડે શકે છે. example માં આવેલ object three properties ધરાવે શકે છે. firstName,lastName અને age.

```
var person = { firstName:"Peter", lastName:"Pan", age:25 };
```

- **Array:** Javascript માં array એ square brackets સાથે લખવામાં આવે શકે છે. તે (,) થી છુટું પડે શકે છે, નીચેના example માં cars નામનો array છે. તે ત્રણે cars ના નામ contain કરે શકે છે.

```
var cars = ["Audi", "Skoda", "Maruti"];
```

JavaScript Operators

- Operator એ symbol છે. Specific mathematical અથવા logical operation ને perform કરવા માટે operator નો ઉપયોગ થાય.
- Javascript language ને support કરતા operators ના types નીચે પ્રમાણે શકે છે.
 - Arithmetic Operators
 - Comparison Operators
 - Logical Operators
 - Bitwise Operators
 - Assignment Operators
 - Conditional (or ternary) Operators
 - String Operators

- `typeof` Operators

Arithmetic Operators

- Variable અને value ની વચ્ચે ગણતરી કરવા માટે arithmetic operator નો ઉપયોગ થાય.
- Arithmetic operator એ javascript language ને support કરે છે.
- Variable A માં 10 અને variable B માં 20 તેથી,

operator	Description	Example
+	બે operand ને add કરે.	A + B એ 30 આપશે.
-	First operand માંથી second operand ને subtract કરે.	A - B એ -10 આપશે
*	બંને operand ને multiply કરે.	A * B એ 200 આપશે.
/	Denominator દ્વારા numerator divide કરે છે.	B / A એ 2 આપશે.
%	Integer division પછી modulus operator અને remainder return કરે.	B % A એ 0 આપશે.
++	Increment operator એ integer value માં 1 નો વધારો કરે.	A++ એ 11 આપશે.
--	Decrement operator એ integer value માં 1 નો ઘટાડો કરે.	A-- એ 9 આપશે.

Assignment Operators

- Javascript variable ને value assign કરવા માટે assignment operator નો ઉપયોગ થાય છે.
- Javascript language એ assignment operators ને support કરે છે.

Operator	Description	Example
=	Simple assignment operator કે જે right side નાથી operands માં value ને assign કરે છે.	C = A + B એ c માથી value A + B ને value assign કરશે.
+=	Add AND assignment operator કે જે right operand થી left operand માં add કરે છે. એ result left operand ને assign કરે છે.	C += A એ C = C + A ને સમાન છે.
-=	subtract AND assignment operator કે જે left operand ને right operand માં બાદ કરે છે. એ result left operand ને assign કરે છે.	C -= A એ C = C - A ને સમાન છે.
*=	Multiply AND assignment operator કે જે right operand ને left operand સાથે multiply કરે છે. અને left operand ને assign કરે છે.	C *= A એ C = C * A ને સમાન છે.
/=	Divide AND assignment operator કે જે right operand ને left operand સાથે divide કરે છે. અને	C /= A એ C = C / A ને સમાન છે.

	left operand ने assign करे छे.	
%=	Modulus AND assignment operator के जे बे operands नो use करीने modulus ने ले छे. अने result left operand ने assign करे छे.	C %= A ए C = C % A ने समान छे.

Conditional Operator

- जे अमुक condition ना base पर variable नी value assign करे छे.
- आ operator true अने false value माटे पहेला expression ने evaluate करे छे अने पछी एक थी बे आपेला statement ने execute करे छे. (statement evaluation ना result पर depend करे छे)

Operator	Description	Example
? :	Conditional expression	जो condition true हशे तो value X : otherwise value Y दर्शावे.

Comparison Operators

- Comparison operators नो उपयोग variable अथवा value वर्चे equality अथवा difference नक्की करवा माटे logical statements मां थाय छे. नीय javascript language द्वारा comparison operators दर्शाविल छे.
- Variable A मां 10 अने Variable B मां 20:

Operator	Description	Example
==	Check करे छे के बे operand नी value equal छे के नहि, जो yes होय तो condition true थई जाय छे.	(A == B) true नथी.
!=	Check करे छे के बे operand नी value equal छे के नहि, जो value equal नथी तो condition true थई जाय छे.	(A != B) true छे.
>	Check करे छे जो left operands नी value right operands नी value करता greater than होय तो condition true थई जाय छे.	(A > B) true नथी.
<	Check करे छे जो left operands नी value right operand नी value करता less than होय तो condition true थई जाय.	(A < B) true छे.
>=	Check करे छे जो left operands नी value greater than or equal छे, जो yes तो condition true थई जाय छे.	(A >= B) true नथी.
<=	Check करे छे जो left operands नी value less than or equal छे, जो yes तो condition true थई जाय छे.	(A <= B) true छे.

Logical Operators

- Logical operators નો ઉપયોગ variable અથવા values વચ્ચે ના logic ને determine કરવા માટે થાય છે. ત્યાં javascript language દ્વારા logical operators નીચે મુજબ છે.
- Variable A માં 10 અને Variable B માં 20:

Operator	Description	Example
&&	Logical ને "AND" operator પણ કહેવાય છે. જો બંને operands zero ન હોય તો પછી condition true થઈ જાય છે.	(A && B) true નથી.
	Logical ને "OR" operator પણ કહેવાય છે. જો એ operands માંથી કોઈ પણ zero ન હોય તો condition true થઈ જાય છે.	(A B) true છે.
!	Logical ને "NOT" operator પણ કહેવાય છે. user તેના operand ની logical state ને reverse કરી શકે છે. જો કોઈ પણ condition true હોય તો logical Not operator false બનાવશે.	!(A && B) false છે.

Bitwise Operators

- Bit operators 32 bits number પર કામ કરે છે operation માં કોઈ પણ numeric operand ને 32 bit number માં convert કરવામાં આવે છે. તેનું result પાછુ javascript number માં convert થાય છે.
- ત્યાં javascript language દ્વારા bitwise operator support કરે છે.
- Variable A માં 2 અને Variable B માં 3:

Operator	Description	Example
&	Bitwise AND operator તરીકે ઓળખાય છે. તે તેના integer arguments ના દરેક bit પર Boolean AND operation કરે છે.	(A & B) એ 2 છે.
	Bitwise OR operator તરીકે ઓળખાય છે. Boolean OR તેના integer arguments ના દરેક bit પર operation કરે છે.	(A B) એ 3 છે.
^	Bitwise XOR operator ને call કરે. તે તેના integer argument ના દરેક bit પર Boolean exclusive OR operation કરે છે. Exclusive OR means એ છે કે operand એક સાચું છે અથવા operand બે સાચું છે, પરંતુ બંને નહિ.	(A ^ B) એ 1 છે.
~	Bitwise NOT operator તરીકે ઓળખાય છે. તે એક unary operator છે અને તે operator માં તમામ Bits ને reverse કરી ને work કરે છે.	(~ B) એ -4 છે.
<<	Bitwise shift left operator કહેવામાં આવે છે. તે (A << 1) એ 4 છે.	

	બધા bits ને move કરે છે. A એ બીજી operand માં specify place દ્વારા left side છે. નવા bits zero થી fill કરેલી છે. એક position દ્વારા બાકીની value ને shift કરવું અને તે 2 દ્વારા multiply કરવા જોવું છે, બે positions ને equivalent છે. 4, વગેરે દ્વારા multiply થાય.	
>>	Sign operator સાથે bitwise shift Right કહેવામાં આવે છે. તે બીજી operand માં specifie કરેલ place ની સંખ્યા દ્વારા તેના bits ને તેના first operand માં right side ખસેડે છે. result ની sign ને સાચવવા માટે left side સાચવેલા bits original operand ના sign bit પર depend છે. જો first operand negative હોય તો result high bits માં મૂકવામાં આવે છે. value ને divide કરવું એ એક place ને equivalent છે. 2 (remanider ને અવગણવું) દ્વારા division કરવું right side બે places ને integer ને division થી 4 અને તેથી આગળ છે.	(A >> 1) એ 1 છે.
>>>	Zero operator સાથે Bitwise shift right કહેવામાં આવે છે. આ operator ફક્ત >> operator ની જેમ જ છે. સીવાય કે left side એ shift bits હંમેશાં zero હોય છે.	(A >>> 1) એ 1 છે.

String Operators

- **+ Operator** નો ઉપયોગ concatenate strings ને (add) કરવા માટે પણ થઈ શકે છે. એક સાથે બે અથવા વધુ string variables ઉમેરવા માટે + operator નો ઉપયોગ થાય જે નીચે ઉદાહરણ માં બતાવવામાં આવેલા છે.

```
Data1= "What a Very";
```

```
Data2= "nice day";
```

```
Ans = txt1 + txt2;
```

- Ans નું output :

What a very nice day

- બે number ઉમેરવાથી રકમ return થશે, પરંતુ number અને string add કરીને એક string return થશે.

```
y="5" + 5;
```

```
z="Hello" + 5;
```

- X, Y અને Z નું result:

55

Hello5

Hello5

The typeof Operator

- typeof એ unary operator છે. જે તેના single operand પહેલા મુકવામાં આવે છે, જે operand કોઈ પણ પ્રકારનો હોઈ શકે છે. તેની value string છે જે operand ની data type ને indicate કરે છે.
- typeof operator "number", "string" અથવા "Boolean" ને evaluates કરે છે, તેનો operand કોઈ number, string અથવા boolean value છે. તો evaluation ના base પર true અથવા false આપે છે. અહીં typeof operator ની return value નું list આપેલું છે.

Type	String Returned by typeof
Number	"number"
String	"string"
Boolean	"boolean"
Object	"object"
Function	"function"
Undefined	"undefined"
Null	"object"

JavaScript Literals

- **Literals** એ javascript માં value ને represent કરે છે.
- તેની value fix હોય છે. અને તે variable નથી.
- નીચે literals ના types ને સમજાવેલ છે.

Integers Literals

- **Integers literals** ને decimal (base 10), hexadecimal (base 16) અથવા octal (base 8) format માં બતાવી શકાય છે. decimal integer literals એ 0 (zero) વિના �digit ના sequence નો સમાવેશ થાય છે.
- Integers literals નું ઉદાહરણ:

```
123, -20, 12345 // Decimal (base 10)
7b, -14, 3039 // Hexadecimal (base 16)
173, -24, 30071 // Octal (base 8)
```

Floating Point Literals

- Floating point ના literal નીચેના ભાગો ધરાવે છે: decimal integer, decimal point ("."), fraction, exponent અને suffix type. Floating point literal માં ઓછામાં ઓછો એક digit હોવો જોઈએ અને decimal point અથવા "e" (અથવા "E") હોવું જરૂરી છે.
- Floating point literals નું ઉદાહરણ:

 - 3.1415
 - 3.1E12
 - .1e12
 - 2E-12

Boolean Literals

- Boolean literals ની બે values છે : **true** અને **false**

Object literals

- Object literal માં zero અથવા વધારે name અને value થી જોડાયેલી pair હોય છે. જે comma(,) થી જુદી પડે છે.
- આ પાર �curly braces ની અંદર હોય છે.
- JavaScript માં object literal નીચે પ્રમાણે declare કરવામાં આવે છે.

1. Properties વગર object literal:

```
var userObject = {}
```

2. અમૃત properties સાથે object literal:

```
var student = { FirstName:"Peter", LastName:"Parker", Age:24 };
```

String Literals

- String literal એ zero અથવા વધુ character સાથે double (") અથવા single (') quotes માં હોય છે. એક string માં સરખી type નો quotes ન હોવો જોઈએ.
- String literals નું ઉદાહરણ નીચે મુજબ નું છે.

"Hello! How are you?"

'Good Morning'

"1234"

"I am Peter \n My Hobby is Cricket"

Special Characters

- તમે javascript string literals માં નીચેના special characters નો ઉપયોગ કરી શકો છો.

\b indicates a backspace.

\f indicates a form feed.

\n indicates a new line character.

\r indicates a carriage return.

\t indicates a tab character.

Escaping Characters

- તમે backslash દવારા તેમને પહેલેથી strings ની અંદર quote insert કરી શકો છો. આ quotes થી escaping તરીકે જાણીતું છે.
- નીચે escaping character literals નું ઉદાહરણ આપેલું છે:

```
var Message="He read \"the book of JavaScript\" by Peter James.";
document.write (Message);
```

- આ result છે.

He read "the book of JavaScript" by Peter James.

JavaScript Array

- Array નો ઉપયોગ single variable માં multiple value ને store કરવા થાય છે.
- Array એ single નામ સાથે ધણી value ને hold કરી શકે છે. અને તે value ને index number દવારા access કરી શકાઈ છે.
- Array એ single પ્રકાર નો object છે. **typeof Operator** એ javascript માટે array માટે object return કરે છે. Array અને object વચ્ચે main તફાવત array એ **numbered indexes** ઉપયોગ કરે છે અને object એ **named indexes** ઉપયોગ કરે છે.

Creating an Array

Syntax:

```
var array-name = [item1, item2, ...];
```

OR

```
var array-name = new Array(item1, item2, ...); //Creating an Array object
```

- આ array parameter એ string અથવા interger નું list છે. જ્યારે તમે array constructor સાથે single numeric parameter ને specify કરો, ત્યારે તમારે array ની initial length ને specify કરવી પડે છે. array એ maximum 4,294,967,295 સુધી ની length ને allow કરે છે.

Example:

```
var Games = [ "Chess", "Cricket", "Football" ]; //Create an array by simply assigning values
                                                assigning values
```

- તમે તેના index number નો ઉપયોગ કરી ને array element ને access કરી શકો છો.
- Games[0] is the first element (Chess)
- Games[1] is the second element (Cricket)

Games[2] is the third element (Football)

Creating a multi-level (multidimensional) structure

Example:

```
var matrix = [[1, 2, 3],[4, 5, 6],[7, 8, 9]];
```

- Multi-level structure માંથી value ને access કરી શકો છો.

```
matrix[1][1]); //it will gives a value "5"
```

Array Properties:

- એહી array properties નું list દર્શાવેલ છે.

Property	Description
Constructor	Array function ના reference ને return કરે અને object ને create કરે છે.
index	string ની zero-based index ને represent કરે છે.
length	Array માં element ના number ને set અથવા return કરે છે.
prototype	Array object થી property અને method add કરે છે.

Array Methods

- એહી array methods નું list દર્શાવે છે.

Method	Description	Syntax
concat()	બે અથવા બધારે array join કરે છે અને joined array ની copy return કરે છે.	array.concat(value1, value2, ..., valueN);
indexOf()	Element માટે array search કરે છે અને તેની position return કરે છે.	array.indexOf(searchElement[, fromIndex]);
join()	String માંથી બધા elements ને join કરે છે.	array.join(separator);
lastIndexOf()	Element માટે starting થી end સુધી array search કરે છે અને position return કરે છે.	array.lastIndexOf(searchElement[, fromIndex]);
pop()	Array ના last element ને remove કરે અને તે element ને return કરે છે.	array.pop();
push()	Array નો last element માં નવો element add કરે છે અને new length return કરે છે.	array.push(element1, ..., elementN);
reverse()	Array માં element ને reverse order માં arrange કરે છે.	array.reverse();
shift()	Array ના first element ને remove કરે અને element ને return કરે છે.	array.shift();
slice()	Array ના part ને select કરે અને new	array.slice(begin [,end]);

	array ने return करे छे.	
sort()	Array ना element ने sort करे छे.	array.sort(compareFunction);
splice()	Array ना element Adds/remove करे छे.	array.splice(index, howMany, [element1][, ..., elementN]);
toString()	Array ने string मां convert करे अने result ने return करे छे.	array.toString();
unshift()	Array नी शरूआत मां नवा element ने add करे अने नवी length ने return करे छे.	array.unshift(element1, ..., elementN);
valueOf()	Array नी primitive value ने return करे अने array ए string तरीके return करे छे.	array.valueOf();

Example:

```

<html>
<head>
    <title>The Array Method Example</title>
    <script type="text/javascript">
        var Games=["Cricket", "Chess", "Football", "TableTennis","WWE"];
        document.write("<b>Pop Method Output :</b>" + Games.pop() + "<br>");
        document.write("<b>Final Result after using Pop method:</b>" + Games + "<br><br>");

        document.write("<b>Push Method Output:</b>" + Games.push("WWE") + "<br>");
        document.write("<b>Final Result after using Push method:</b>" + Games + "<br><br>");
        document.write("<b>Shift Method Output:</b>" + Games.shift() + "<br>");
        document.write("<b>Final Result after using Shift method:</b>" + Games + "<br><br>");
        document.write("<b>Unshift Method Output:</b>" + Games.unshift("KhoKho") + "<br>");
        document.write("<b>Final Result after Unshift method:</b>" + Games + "<br><br>");

    </script>
</head>
</html>

```

Output:

Pop Method Output: WWE

Final Result after using Pop method: Cricket,Chess,Football,TableTennis

Push Method Output: 5

Final Result after using Push method: Cricket,Chess,Football,TableTennis,WWE

Shift Method Output: Cricket

Final Result after using Shift method: Chess,Football,TableTennis,WWE

Unshift Method Output:5

Final Result after Unshift method: KhoKho,Chess,Football,TableTennis,WWE

Javascript Functions

- Javascript function એ particular task ને perform કરવા માટે code ના block ને design કરે છે.
- Eliminates ને ધણીવાર સરખા code ને વારંવાર લખવા ની જરૂર પડે છે. તે programmer ને modular code લખવામાં મદદ કરે છે.
- તમે મોટા program ને નાના નાના function માં devide કરી શકો છો.
- જેવી કે કોઈ પણ advance programming language, function નો ઉપયોગ કરીને modular code write કરવા માટે javascript એ જરૂરી એવા બધા feature ને support કરે છે.

Function Definition:

- Function નો ઉપયોગ કર્યું પહેલા આપણે function ને define કરવું પડે છે. Function keyword દ્વારા તેને define કરવામાં આવે છે. function name unique હોઈ છે. Parameter નું list અને statement block ને curly braces વડે પુરા કરવામાં આવે છે.

Syntax:

```
<script type="text/javascript">
    function functionname(parameter-list)
    {
        Block of code or statements
    }
</script>
```

Example:

```
<script type="text/javascript">
    function PrintMessage()
    {
        alert("Hello! How are you?");
    }
</script>
```

Calling a function:

- જ્યારે function ને call કરવું હોઈ ત્યારે function name થી call કરાવી શકો છો.

```
<script type="text/javascript">
    PrintMessage();
</script>
```

Function parameters:

- Function માં, જ્યારે function calling થાય ત્યારે different parameter pass કરવામાં આવે છે. Function ની અંદર captured થયેલાં parameter pass કરવામાં આવે છે.

- તને (,) comma દ્વારા દર્શાવામાં આવે છે.

Example:

```
<script type="text/javascript">
    function MyIntro(Name, Hobby)
    {
        alert("My Name is " + Name + ". My Hobby is " + Hobby);
    }
</script>
```

- હવે નીચે પ્રમાણે આપણે function call કરશું:

```
<script type="text/javascript">
    MyIntro('Peter Parker', 'Cricket' );
</script>
```

The return statement:

- JavaScript ફંક્શનમાં return statement એ optional હોઈ શકે છે. જો તમે ફંક્શનમાંથી value return કરવા માંગતા હોય તો આ statement એ function માં છેલ્લું હોવું જોઈએ.
- ઉદાહરણ તરીકે, જો તમે કોઈ ફંક્શનમાં બે number pass કરવા માંગો છો અને પછી તમે ફંક્શનથી તમારા program ને વારંવાર call કરવા માગો છો જે નીચે બતાવેલ છે:

Example:

```
<script type="text/javascript">
    function Multiplication(Number1, Number2)
    {
        var Total;
        Total=Number1*Number2;
        return Total;
    }
</script>
```

- હવે આપણે નીચે પ્રમાણે આ ફંક્શનને call કરી શકીએ છીએ. તે એક popup box display કરશે જે દર્શાવે છે કે બે કુમારીનો ગુણાકાર છે.

```
<script type="text/javascript">
    var result;
    result=Multiplication(25, 25);
    alert(result);
</script>
```

JavaScript Control Statements

- JavaScript ની અંદરના Control structure એ program ના flow ને code અથવા function ના unit ની અંદર change કરવાનું allow કરે છે.

- JavaScript conditional statement ને support કરે છે.તેનો ઉપયોગ જુદી જુદી action ના based પર જુદી જુદી condition perform કરવા થાય છે.
- javascript ના નીચે પ્રમાણે ના conditional statement છે.
 - if:** જો condition True હોય તો જુદી code નો block executed થાય છે.
 - if...else:** જો condition True હોય તો જુદી if વાળો ભાગ executed થાય અને Flase હોય તો else વાળો ભાગ executed થાય છે.
 - if...else if....:** જો પેહલી condition Flase હોય તો નવી condition test થાય.
 - switch :** ધણા બધા alternative code નો block executed થાય.

The if statement

- જો condition True હોય તો javascript નો code executed થાય છે.

Syntax:

```
if (condition or expression)
{
    // Block of code to be executed if the condition is true
}
```

- નીચે આપેલા ઉદાહરણમાં નીચે બતાવેલ message દર્શાવો જો 18 વર્ષથી ઉંમર વધારે હોય.

Example:

```
<script type="text/javascript">
    var age=20;
    if (age>18)
    {
        document.write("Great you are eligible to apply for driving license");
    }
</script>
```

Output:

Great you are eligible to apply for driving license

The if...else Statement

- જો condition flase હોય ત્યારે code નો block execute થાય છે.

Syntax:

```
if(condition or expression)
{
    Block of code to be executed if the condition is true
}
else
{
    Block of code to be executed if the condition is false
}
```

}

- નીચે આપેલા ઉદાહરણમાં જો કોઈ ની ઉંમર 18 કરતા મોટી હોય તો message બતાવે નહીં તો તે else વાળા ભાગમાં લખાયેલ message display કરશે.

Example:

```
<script type="text/javascript">
  var age=15;
  if(age < 18)
  {
    document.write("Great you are eligible to apply for driving license");
  }
  else
  {
    document.write("Sorry you are not eligible to apply for driving license");
  }
</script>
```

Output:

Sorry you are not eligible to apply for driving license

The if...else if... Statement

- જો condition false હોય ત્યારે નવી condition specify કરે છે.

Syntax:

```
if(condition1 or expression1)
{
  Block of code to be executed if condition1 is true
}
else if (condition2 or expression2)
{
  Block of code to be executed if the condition1 is false and condition2 is true
}
else
{
  Block of code to be executed if the condition1 is false and condition2 is false
}
```

Example:

```
<script type="text/javascript">
  var book="history";
  if(book=="history")
  {
```

```

document.write("<b>You Select History Book</b>");
}
else if(book=="maths")
{
    document.write("<b>You Select Maths Book</b>");
}
else if(book=="economics")
{
    document.write("<b>You Select Economics Book</b>");
}
else
{
    document.write("<b>You Select nothing</b>");
}
</script>

```

Output:

You Select History Book

The Switch Statement

- તેનો ઉપયોગ select કરેલ ધરણા code ના block ને executed કરે છે.
- Switch expression એ એકવાર run થાય છે. Expression value એ દરેક case ની value સાથે compared થાય છે. જો તે match થાય તો તેને associated code નો block executed થાય છે.
- જ્યારે javascript code માં break keyword આવે, તો break એ switch block ની બહાર નીકળી જાય છે. તે execution ને stop કરે છે.

Syntax:

```

switch(expression)
{
    case condition 1: statement(s)
        break;
    case condition 2: statement(s)
        break;
    .....
    .....
    case condition n: statement(s)
        break;
    default: statement(s)
}

```

Example:

```
<script type="text/javascript">
    var grade=2;
    document.write("You got ");
    switch (grade)
    {
        case 1: document.write("Distinction.<br />");
        break;
        case 2: document.write("First Class.<br />");
        break;
        case 3: document.write("Second Class.<br />");
        break;
        case 4: document.write("Pass Class.<br />");
        break;
        default: document.write("Fail.<br />");
    }
</script>
```

Output:

You got First Class.

Looping Structure

- જ્યારે તમે program લખો, ત્યારે અમુક વાર એવી situation આવે છે કે અમુક action ને વારંવાર perform કરવી પડે છે, જેથી loop statement ની જરૂર પડે છે. તે coding ની line માં ઘટાડો કરે છે.
- JavaScript એ બધી જ �necessary loops ને support કરે છે: **for, for...in, while and do...while.**

The for loop

- For loop એ looping નો મહત્વ નો ભાગ છે. તેના ત્રણ important part છે.
 - The loop initialization** જ્યાં આપને counter ને starting value થી initialize કરી શકીએ છીએ. loop ની શરૂઆત થાય તે પહેલા initialization statement execute થાય છે.
 - The test statement** આપેલી condition સાચી અથવા ખોટી હોય તો test કરશે. જો આપેલી condition સાચી હશે તો code ની અંદર નો ભાગ execute થશે અને જો ખોટી હશે તો loop ની બહાર આવી જશે.
 - The iteration statement** જ્યારે તમારે counter ને increase અથવા decrease કરવું હોય.
- આ ત્રણેય part single line નાલ semicolon દ્વારા saperated થાય છે.

Syntax:

```
for(initialization; test condition; iteration statement)
{
```

Block of code to be executed if test condition is true

}

Example:

```
<script type="text/javascript">
    var count;
    document.write("Print the numbers from 1 to 5" + "<br />");
    for(count=1; count<=5; count++)
    {
        document.write("Current Count: " + count );
        document.write("<br />");
    }
    document.write("Finally Print!");
</script>
```

Output:

Print the numbers from 1 to 5

Current Count: 1

Current Count: 2

Current Count: 3

Current Count: 4

Current Count: 5

Finally Print!

The while loop

- While loop નો ઉપયોગ એ છે કે જ્યાં સુધી code અથવા statement ની expression વારેવાર true ની થાય ત્યાં સુધી execute કરે છે.
- એકવાર expression false થશે પછી loop ત્યાંથી exit થઇ જશે.

Syntax:

```
while(expression)
{
    Block of code to be executed
}
```

Example:

```
<script type="text/javascript">
    var count=1;
    document.write("Print the numbers from 1 to 4" + "<br />");
    while(count<5)
    {
        document.write("Current Count: " + count);
        document.write("<br />");
        count++;
    }
```

```

}
document.write("Finally Print!");
</script>

```

Output:

Print the numbers from 1 to 4

Current Count: 1

Current Count: 2

Current Count: 3

Current Count: 4

Finally Print!

The do...while loop

- **The do...while loop** એ while loop જેવું જ વર્ક કરે છે. આં લોપ માં condition loop નાં end માં check થાય છે. આનો અર્થ એ છુ કે condition false હશે તો પણ લોપ એક વાર તો execute થશે જ.

Syntax:

```

do
{
    Block of code to be executed;
}
while(expression);

```

Example:

```

<script type="text/javascript">
    var count=1;
    document.write("Starting Loop" + "<br />");
    do
    {
        document.write("Current Count: " + count + "<br />");
        count++;
    }while(count < 2);
    document.write("Loop stopped!");
</script>

```

Output:

Starting Loop

Current Count: 1

Loop stopped!

The for...in Loop

- આ loop javascript દ્વારા જ સપ્પોર્ટ છે. તેણે **for...in loop** કહેવામાં આવે છે. આ loop નો ઉપયોગ loop ના object ની properties નો ઉપયોગ કરવા માટે થાય છે.
- દરેક interation માં એક property માથી object એ variablename તરીકે assign થાય છે અને આ loop ચાલુ રહે ત્યાં સુધી object ની બધી propertie exhausted થાય છે.

Syntax:

```
for (variablename in object)
{
    Statement(s) or block of code to execute.
}
```

Example:

```
<script type="text/javascript">
    var message="";
    var person={FirstName:"Peter", LastName:"Parker", Age:20};
    var addvalue;
    document.write("<b>Information of an Employee:</b><br/> ");
    for(addvalue in person)
    {
        document.write(person[addvalue]);
        document.write("<br />");
    }
    document.write("Thanks for the Details!");
</script>
```

Output:

Information of an Employee:

```
Peter
Parker
20
Thanks for the Details!
```

JavaScript Loop Control

- JavaScript તમને loops અને switch statement ને handle કરવા માટે ના controls આપે છે. એવી situation હોય જ્યારે તમને end સુધી પહોંચા વગર loop માંથી બહાર આવવાની જરૂર પડે ત્યારે તમે code block નો part skip કરીને next iteration start કરી શકો છો.
- આવી બધી situation ને handle કરવા માટે, javascript break અને continue statement provide કરે છે આ statement નો ઉપયોગ loop માંથી તરત જ બહાર નીકળવા માટે અથવા તેને respectively કોઈ પણ loop નું next iteration start કરે છે.

The break Statement

- **Break statement** નો ઉપયોગ loop માંથી બહાર આવવા માટે થાય છે. Break statement એ code નું execution stop કરશે અને / અથવા switch statement ની અંદર નો case testing કરશે અને loop પછી નું execution continue રાખશે.

Example:

```
<html>
    <head>
        <title>The Break Statement</title>
        <script type="text/javascript">
            function breakloop()
            {
                var count = 0;
                document.write("Entering the loop<br/>");
                while(count < 20)
                {
                    if (count==5)
                    {
                        break;          // breaks out of loop completely
                    }
                    count=count+1;
                    document.write(count + "<br />");
                }
                document.write("Exiting the loop!<br/>");
            }
        </script>
    </head>
    <body>
        <p>Click the button to see the result using Break Statement.</p>
        <button onclick="breakloop();">Try it</button>
    </body>
</html>
```

- Button પર click કરવામાં આવે ત્યારે ઉપરોક્ત code નીચેનું result આપશે.

Output:

Entering the loop

1

2

3

The Continue Statement

- Continue statement નો ઉપયોગ interpreter ને immediately loop ના આગલા iteration ને start કરવા અને બાકી code ને skip કરવા માટે થાય છે.
- જ્યારે એક statement ચાલુ હોય છે ત્યારે program નો flow loop ની expression ને check કરશે જો condition સાચી હશે તો next iteration start કરશે બાકી loop ની બહાર આવી જશે.

Example:

```
<html>
  <head>
    <title>The Continue Statement</title>
    <script type="text/javascript">
      function continueloop()
      {
        var count = 0;
        document.write("Entering the loop<br/>");
        while(count < 10)
        {
          count=count+1;
          if (count==5)
          {
            continue; // skip rest of the loop body
          }
          document.write(count + "<br />");
        }
        document.write("Exiting the loop!<br/>");
      }
    </script>
  </head>
  <body>
    <p>Click the button to see the result using Continue Statement.</p>
    <button onclick="continueloop();">Try it</button>
  </body>
</html>
```

- Button પર click કરવામાં આવે ત્યારે ઉપરોક્ત code નીચેનું result આપશે.

Output:

Entering the loop

1

2

```

3
4
6
7
8
9
10
Exiting the loop!

```

Using Labels to Control the Flow

- Label એ simply identifier છે જે colon દ્વારા follow થાય છે. જે statement અથવા code ના block પર લાગુ પડે છે. Label ને સમજવા માટે આપણે break અને continue સાથે ના જુદા જુદા ઉદાહરણો જોઈશું.
- Label નો ઉપયોગ flow ને control માં રાખવા માટે break અને continue સાથે કરી શકીએ છીએ.

Example:

```

<html>
    <head>
        <title>Using Labels to Control the Flow</title>
        <script type="text/javascript">
            function Labeldemo()
            {
                document.write("Entering the loop!<br />");
                outerloop: // This is the label name
                for(var i=0;i<3; i++)
                {
                    document.write("Outerloop: " + i + "<br />");
                    innerloop:
                    for(var j=0;j<5;j++)
                    {
                        if(j > 2)
                        {
                            break innerloop; // Quit the
                            innermost loop
                        }
                        document.write("Innerloop: " + j + " <br />");
                    }
                }
                document.write("Exiting the loop!<br />");
            }
        </script>
    </head>
    <body>
        <h1>Using Labels to Control the Flow</h1>
    </body>
</html>

```

```
        }
    </script>
</head>
<body>
    <p>Click the button to see the result using Labels.</p>
    <button onclick="Labeldemo();">Try it</button>
</body>
</html>
```

- Button પર click કરવામાં આવે ત્યારે ઉપરોક્ત code નીચેનું result આપશે.

Output:

Entering the loop!

Outerloop: 0

Innerloop: 0

Innerloop: 1

Innerloop: 2

Outerloop: 1

Innerloop: 0

Innerloop: 1

Innerloop: 2

Outerloop: 2

Innerloop: 0

Innerloop: 1

Innerloop: 2

Exiting the loop!

JavaScript Document Object Model:

What is the DOM?

- DOM નું પૂરું નામ **Document Object Model** છે.
- W3C એ document object model ને HTML અને XML documents ની script ને access અને manipulation કરવા માટે standard application programming interface (API) તરીકે establish કર્યું છે.

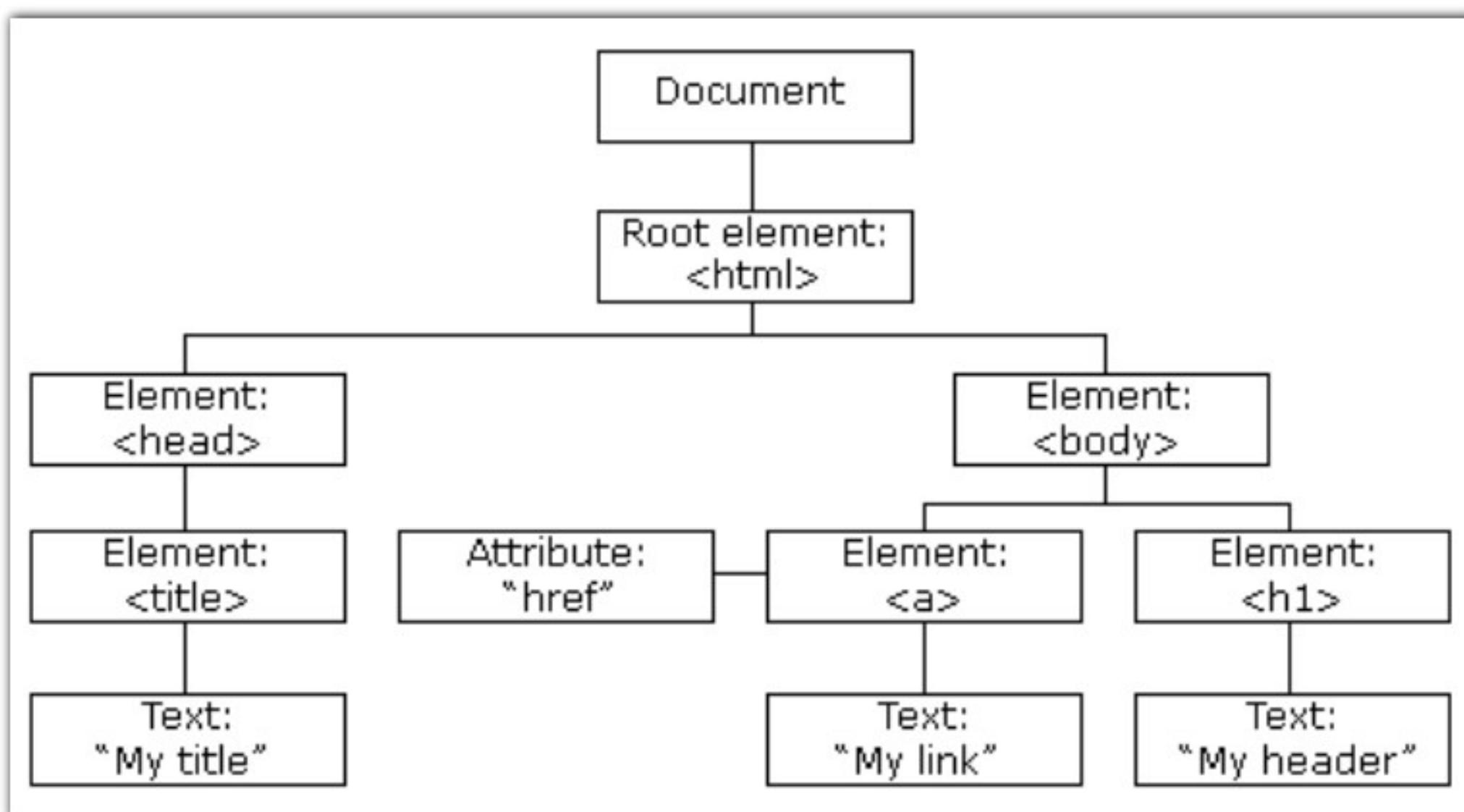


Fig 3.1 Document Object Model Tree Structure

- જ્યારે web page load થાય ત્યારે browser page નું document object model બનાવે છે જે document નો logical view આપે છે જ્યાં objects જુદા-જુદા ભાગોને રજુ કરે છે જેવા કે **Windows documents, elements, attributes, text, events, stylesheets, style rules etc.**
- આ DOM objects કોઈ document ના natural organization ને reflect કરવા માટે tree structure માં ગોઠવવામાં આવે છે.
- દરેક web page પાસે tree ના root પર document ના node હોય છે. Head અને body node document ના child node છે. (Fig 3.1)
- Object model સાથે, dynamic HTML બનાવવા માટે JavaScript નો ઉપયોગ કરી શકો છો.
 - JavaScript બધા HTML elements અને attributes અને page માં CSS style ને manage કરે છે.

Introducing object in Model

- દરેક વેબ પેજ browser window ની અંદર રહે છે. જેને object તરીકે considere કરવામાં આવે છે. ડોક્યુમેન્ટ object HTML ડોક્યુમેન્ટ ને represent કરે છે. જે window માં display થાય છે.
- Document object માં વિવિધ properties છે. જે બીજા objects ને refer કરે છે. જે document content ને access અને modification માટે allow કરે છે. objects એ hierarchy માં organize થાય છે. જે નીચેની આફ્રિત 3.2 માં દર્શાવેલ છે.

- **Window object:** તે hierarchy માં top પર છે.
- **Document object:** દરેક HTML ડોક્યુમેન્ટ જે window માંથી load થાય છે. તે ડોક્યુમેન્ટ object બને છે. ડોક્યુમેન્ટ એ page content દર્શાવે છે.
- **Form object:** <form>...</form> tags માં આવેલ દરેક form object ને set કરે છે.

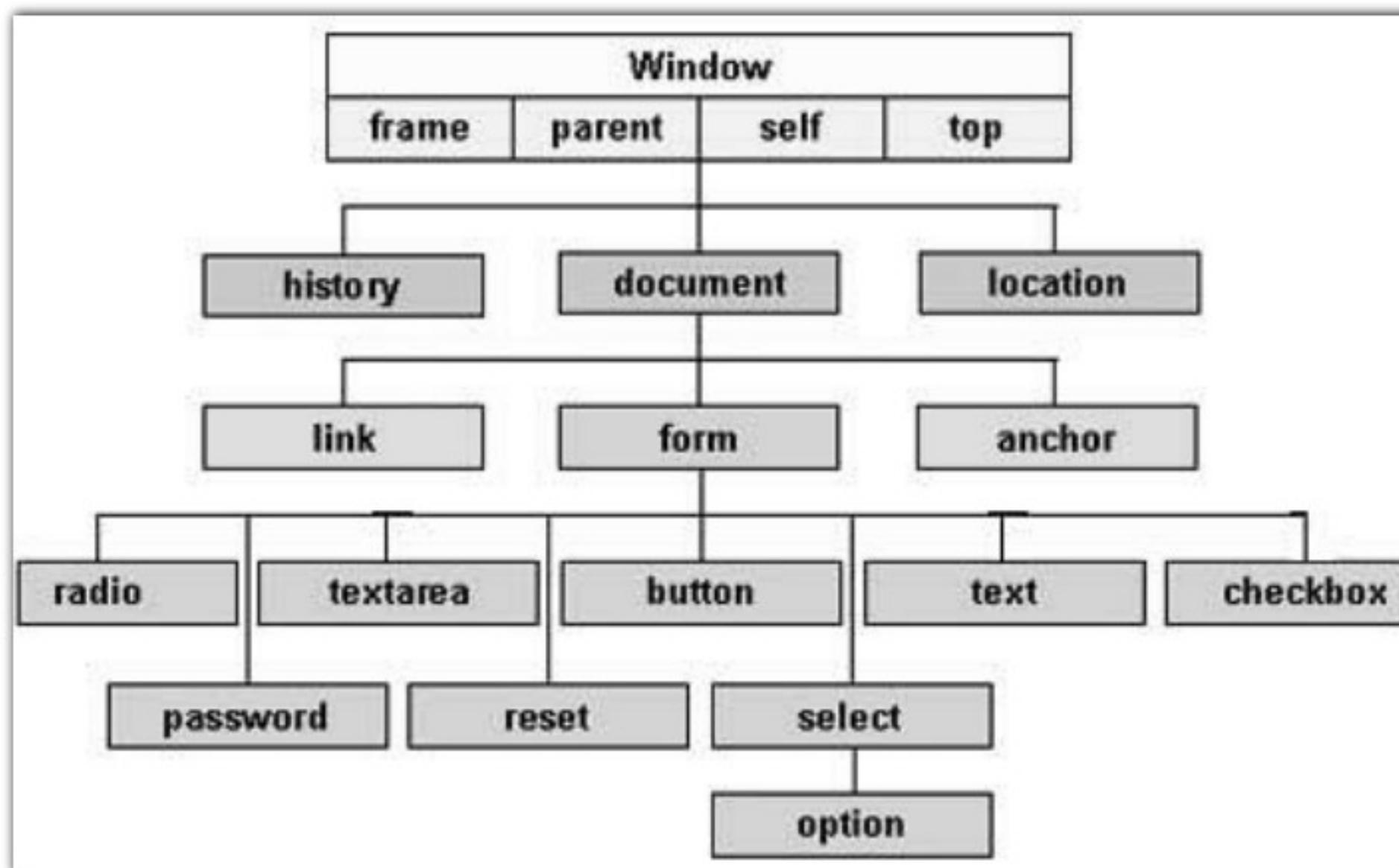


Figure 3.2 hierarchy of important objects

- **Form control elements:** Form object માં તે object માટે define કરેલ બધા element આવેલ છે. જેમ કે text fields, buttons, radio buttons, and checkboxes.

The Document Object

- જ્યારે કોઈ HTML ડોક્યુમેન્ટ વેબ browser માંથી load થાય છે, તે ડોક્યુમેન્ટ object બને છે.
- Document Object એ HTML Document નું root node છે. Elements nodes, text nodes, attribute nodes અને comment nodes છે.
- Document Object એ બધા node objects ને access કરવા માટે methods અને properties provide કરે છે. Document Object એ બધા browser ને support કરે છે.
- નીચે આપેલી properties અને methods એ HTML document નો ઉપયોગ કરી શકે છે:
document.body, **document.close()**, **document.forms**, **document.open()**, **document.title**,
document.getElementById(), **document.getElementsByTagName()**,
document.getElementsByName(), **document.writeln()** અને **document.write()**.

Writing to Documents

- **Write()** method થી document પર HTML Expression અથવા JavaScript code લખી શકાય છે.
- HTML document full load થાય પછી **document.write()** નો ઉપયોગ કરવાથી તમામ existing HTML delete થઈ જશે.

- `document.writeln()` method એ `write()` જેવી જ છે, ફક્ત દરેક statement પછી નવી line માં character add કરે છે.

Example:

```
<script>
    document.write("Hello Friends....");
    document.write("Hi, I'm Peter Parker");
</script>
```

Dynamic Documents

- HTML DOM એ JavaScript ને HTML element ના content બદલવા માટે allow કરે છે.
- HTML element ના content change કરવા માટેની Syntax:
`document.getElementById(id).innerHTML = new HTML`
- HTML elements ની property ને modify કરવા માટે inner HTML property નો ઉપયોગ કરી શકાય છે.
- HTML attribute ની value ને change કરવા માટે નીચેની syntax નો ઉપયોગ કરી શકીએ છીએ.

Syntax:

`document.getElementById(id).attribute = new value`

Example:

```
<html>
<head>
    <title>Dynamic Document using DOM</title>
<script>
    function showdata()
    {
        document.getElementById("p1").innerHTML =
        document.getElementById("data").value;
    }
</script>
</head>
<body>
    <p id="p1" style="font-weight:bold;">Change Me by Entering Data</p>
    <input type="text" id="data" />
    <button onclick="showdata()">Click Me</button>
</body>
</html>
```

- ઉપર ના Code મા જ્યારે આપણે બટન પર ક્લિક કરીશું, ત્યારે આપણે enter કરેલા textbox ના content સાથે <p> element ના content બદલશે.

Browser Object

- Browser Object Model (BOM) JavaScript ને Browser થી "talk to" માટે allow કરે છે. Browser Object Model (BOM) માટે કોઈ official standard નથી.
- Browser આપણે object ની hierarchy provide કરે છે. જેનો ઉપયોગ page, time, screen, element વગેરે ની વિવિધ માહિતી ને control કરવા અને access કરવા માટે ઉપયોગ કરી શકીએ છીએ.
- BOM Objects Browser ને control કરવા ની મજુરી આપે છે. ઉદાહરણ તરીકે current URL ને change કરવા, frame access કરવા, XMLHttpRequest સાથે server પર background ને request કરવા. Function જેવા કે alert, confirm, prompt જેવા BOM થી સંબંધિત છે. તે browser દ્વારા provide કરવામાં આવે છે.
- નીચેની details માં કેટલાક JavaScript browser option છે.
 - History Object:** Browser ની history માટે store થયેલા URL ને check કરવા અને move કરવા માટે વપરાય છે.
 - Location Object:** Current URL વિશેની માહિતી store કરે છે. આ object નવા page ને load કરવું અથવા current page ને ફરીથી load કરવા જેવા task કરવા માટેના object provide કરે છે.
 - Navigator Object:** Current Browser વિશેની information store કરે છે. ઉદાહરણ તરીકે, તમે browser નો type નક્કી કરી શકો છો. Browser ની cookie enable છે કે નહિ તે check કરી શકો છો.
 - Screen Object:** Page ની height, width અને color, depth, page ને display કરવા માટે વપરાયેલા device ની physical characteris ને specify કરે છે.
 - Window Object:** Browserની window ના access provide કરે છે. જેથી message box, display કરવા જેવા task perform કરી શકાય. frame ધરાવતા page સામે કામ કરતી વખતે browser સમગ્ર HTML ડોક્યુમેન્ટ માટે એક window બનાવે છે અને દરેક frame માટે બીજી window બનાવે છે.

Note: આ object નું ઉદાહરણ નીચે આપેલે navigator object અથવા window object જેવું જ છે. (Refer Page No.19).

Navigator Object

- Navigator object માં browser વિશેની માહિતી હોય છે.
- તે તમારા user ના browser અને તે browser પર શું enable કરે છે તેના આધારે તમારી JavaScript ને customize કરવા માટે ઉપયોગી છે.
- બધા browser જુદા છે અને JavaScript ને અલગ રીતે handle કરે છે. ઉદાહરણ તરીકે, user પાસે cookie enable નહીં હોય પરંતુ JavaScript enable છે.

Properties :

- appCodeName :** Browser નું code name return કરે છે.
- appName :** Browser નું name return કરે છે.

- **appVersion** : Browser ની version ની માહિતી આપે છે.
- **cookieEnabled** : Browser માં cookie enable છે કે નહિ તે determine કરે છે.
- **platform** : જે platform પર browser compile થાય છે તે return કરે છે.
- **userAgent** : Browser થી server દ્વારા મોકલવામાં આવેલા user agent header ને return કરે છે.

Methods :

- **javaEnabled()** : Browser માં java enable છે કે નહિ તે specify કરે છે.
- **taintEnabled()** : Browser માં data tainting enable છે કે નહિ તે specify કરે છે.

Example:

```
<script type="text/javascript">
    document.write(navigator.appCodeName + "<br>");
    document.write(navigator.appName + "<br>");
    document.write(navigator.appVersion + "<br>");
    document.write(navigator.cookieEnabled + "<br>");
    document.write(navigator.platform + "<br>");
</script>
```

Output:

Mozilla

Netscape

5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/41.0.2272.118 Safari/537.36

true

Win32

The String Objects

- String literal એ zero અથવા વધુ character એક અથવા double quotation marks માં હોય છે. String literals પાસે primary (primitive) data type ની string હોય છે.
- New operator નો ઉપયોગ કરીને string object બનવા માં આવે છે અને તેની પાસે datatype નો object હોય છે.
- Escape character(\) નો ઉપયોગ string માં અન્ય special character ને insert કરવા માટે પણ થાય છે. special character કે જે backslash sign સાથે text string માં add કરી શકાય છે: single quote('), double quote("), backslash(\\"), new line(\n), tab(\t), backspace(\b) વગેરે.

Syntax:

```
var val = new String(string);
```

Properties

- **Constructor:** Function return કરે છે. જે string object ની prototype create કરે છે.

- **length:** String ની length return કરે છે.
- **Prototype:** Object માં methods અને properties add કરવા માટે તમને allow કરે છે.

Methods

- **length:** String માં character નાં number return કરે છે.
- **indexOf():** String માં specify કરેલ value ની first occurrence ની position return કરે છે.
- **lastIndexOf():** String માં specify કરેલ value ની last occurrence ની position return કરે છે.
- **match():** Regular expression માં જે string match થતી હોય એ string return કરે છે.
- **substr():** Character નાં number દ્વારા start position થી string નો part return કરે છે.
- **substring():** એ specify position વચ્ચે ની string નો part આપે કરે છે.
- **toLowerCase():** String ને lowercase letter માં convert કરે છે.
- **toUpperCase():** String ને uppercase letter માં convert કરે છે.
- **trim():** String ના બંને છેડે એથી whitespace remove કરે છે.
- **valueOf():** String object ની primitive value return કરે છે.
- **split():** String ને substring નાં array માં split કરે છે.

Example:

```
<script>
    function myFunction() {
        var str = "      Have a nice day      ";
        alert(str.trim());
    }
</script>
```

ઉપરની script માં stringના બંને બાજુએ થી whitespace દૂર કરે છે.

Understanding built-in object (JavaScript Native Objects)

- બધી programming languages માં built-in object છે. જે languages ની functionality create કરે છે.
- **Built-in objects** એ language નું foundation છે. જેમાં તમે custom code લાખો છો. જે તમારા imagination ના આધારે custom functionality ને create કરે છે.
- Javascript માં built-in અથવા native objects જુદા જુદા છે. આ objects તમારા program માં ગમે ત્યાં accessible છે અને કોઈ પણ operating system ના કોઈ પણ browser માં તે work કરશે.
- નીચે બધા મહત્વપૂર્ણ javascript Native objects દરેકે છે.
 - JavaScript Number Object
 - JavaScript Boolean Object
 - JavaScript String Object
 - JavaScript Array Object
 - JavaScript Date Object

- JavaScript Math Object

User defined object

- JavaScript object આધારિત programming language છે. Objects attribute થી બનેલ છે. જો કોઈ attribute માં function આવેલા હોય, તો તેને object ની method consider કરવામાં આવે છે અને attribute ને property consider કરવામાં આવે છે.
- બધા user-defined object અને built-in objects એ મૂળ objects માંથી બનેલા છે.

The new Operator:

- New operator નો ઉપયોગ object ને create કરવા માટે થાય છે.

Example:

```
var employee = new Object();
```

The Object() Constructor:

- Constructor એ એક function છે. જે object create કરે છે અને initialize કરે છે. Javascript એ object બનાવવા માટે એક object() નામનું special constructor function આપે છે. Object() constructor ની return value એ variable ને સોપેલ છે.

The with Keyword:

- Object ની properties અને methods ના reference માટે with keyword નો ઉપયોગ થાય છે.
- Block duration માટે જે object ને argument તરીકે specify કરવામાં આવે તે default object નો બનેલો છે. Properties અને methods માટે object ને તમે object ના name વગર ઉપયોગ કરી શકો છો.

Syntax:

```
with (object)
{
    properties used without the object name and dot
}
```

Example:

```
<html>
<head>
<title>User-defined objects</title>
<script type="text/javascript">
    // Define a function which will work as a method
    function addPrice(amount){
        with(this){
            price = amount;
    }
}
```

```

        }
    }

function book(title, author){
    this.title = title; // Assign properties to the object
    this.author = author;
    this.price = 0;
    this.addPrice = addPrice; // Assign that method as property.
}

</script>
</head>
<body>
<script type="text/javascript">

    var book = new book("DWSL", "Peter Parker"); // Create the object
    book.addPrice(100);
    document.write("Book title is : " + book.title + "<br>");
    document.write("Book author is : " + book.author + "<br>");
    document.write("Book price is : " + book.price + "<br>");

</script>
</body>
</html>

```

Output:

- ઉપરોક્ત script એ user define function સાથે કોઈ, object કેવી રીતે બનાવો તે define કરે છે. અહીં, this keyword નો ઉપયોગ object નો ઉલ્લેખ કરવા માટે થાય છે. જે function માં pass થાય છે.

Form Object

- Javascript form object એ ડોક્યુમેન્ટ object ની એક property છે. Form object નો ઉપયોગ user data ને પ્રોસેસિંગ માટે input તરીકે લેવા માટે થાય છે. Javascript માં submit method ને call કરી ને અથવા form ના submit button પર click કરીને form submit કરી શકાય છે.
- દરેક HTML form element માટે form object બનવા આવે છે. Form object એ HTML<form> element ને represent કરે છે.
- Form ને તમે **document.form.formName** અથવા **document.formName** દ્વારા access કરી શકો છો. જ્યાં formName એ form નું name છે.
- તમે getElementById() દ્વારા <form>element ને access કરી શકો છો. ઉદાહરણ તરીકે,
`var x = document.getElementById("myForm");`
- તમે document.createElement() method દ્વારા <form>element ને બનાવી શકો છો.

- ઉદાહરણ, `var x = document.createElement("FORM");`

Form Object Properties:

- action:** Action attribute નો ઉપયોગ form field ને get અથવા set કરવા માટે થાય છે.
- elements:** એક form માં આ બધા object ને સમાવતો array return કરે છે.
- encoding:** Enctype attribute નો ઉપયોગ form field ને get અથવા set કરવા માટે થાય છે.
- length:** તેનો ઉપયોગ form માં present સંખ્યા ને check કરવા માટે થાય છે.
- method:** આ attribute નો ઉપયોગ GET અને POST ને get અથવા set કરવા માટે થાય છે.
- name:** તેનો ઉપયોગ form ના name ને get અથવા set કરવા માટે થાય છે.
- target:** તેનો ઉપયોગ form ના attribute ના "target" ને get અથવા set કરવા માટે થાય છે.

Form Object Methods:

- submit():** Javascript નો ઉપયોગ કરીને form ને dynamically submit કરવા માટે થાય છે.
- reset():** Form ની value ને dynamically reset કરવા માટે વપરાય છે.

Working with Form Elements and Their Properties

Button Object

- સામાન્ય રીતે form types માં button નો ઉપયોગ થાય છે. Button object એ HTML <button> element ને represent કરે છે.
- Javascript માં button object ને access કરવાની syntax નીચે મુજબ છે.
`document.formname.buttonname` અથવા `document.getElementById("buttonid")`;

Methods:

- click():** Dynamically રીતે button ને click કરવા માટે તેનો ઉપયોગ થાય છે.
- blur():** Dynamically રીતે button ને blur કરવા માટે તેનો ઉપયોગ થાય છે.
- focus():** Dynamically રીતે button ને focus કરવા માટે તેનો ઉપયોગ થાય છે.

Properties:

- Name:** Button ના name get કરવા માટે તેનો ઉપયોગ થાય છે.
- Type:** Form ની type get કરવા માટે તેનો ઉપયોગ થાય છે.
- Value:** Button ની value ને get અથવા set કરવા માટે તેનો ઉપયોગ થાય છે.
- Disabled:** Button disabled છે કે નહીં તે set કરે અથવા return કરે છે.
- onMouseOver,onClick,onBlur, onFocus એ button objects ની basic events છે.

Text Object

- Javascript text object એ form object ની property છે. જે form માં text field ની જગ્યાએ મુકવામાં આવે છે.
- Example code: `<input type="text" name="Namehere">`.

- Javascript text object ને access કરવા માટેની syntax:

document.formname.textName અથવા **document.getElementById("textid");**

Methods:

- **Blur:** Object માંથી focus remove કરે છે.
- **focus:** Object ને focus આપે છે.
- **select:** Object ના input area ને select કરે છે.

Properties:

- **name:** Text filed ના name ને get કરે છે.
- **type:** Form type ના type ને get કરે છે.
- **value:** Text field ની value ને get અથવા set કરે છે.
- **readOnly:** જો text filed માં readonly set કરેલું હોય તો user કોઈ પણ value ને enter કરી શકતું નથી.
- Text object ની basic event **onMouseOver, onMouseDown, onMouseUp, onClick, onBlur, onFocus** વગેરે.

Text Area Objects

- Javascript text object એ form object ની property છે. જે form માં text field ની જગ્યાએ મુકવામાં આવે છે.
- Javascript text object ને access કરવા માટેની syntax:

document.getElementById("textareaid");

Methods:

- **blur:** From object માંથી focus ને removes કરવા માટે થાય છે.
- **focus:** Object ને focus આપે છે.
- **select:** Object ના input area ને select કરે છે.

Properties:

- **name:** Text area ના name ને represent કરે છે.
- **type:** Textarea object ના type ને represent કરે છે.
- **value:** Textarea object ની current value ને represent કરે છે.
- Text objects ની basic event **onMouseOver, onMouseDown, onMouseUp, onClick, onBlur, onFocus** વગેરે.

Hidden Objects

- Hidden object HTML<input> element ના type="hidden" સાથે represent કરે છે અને form submit કરવામાં આવે ત્યારે data pass કરવા માટે ઉપયોગ થાય છે.

- Javascript मां text object ने access करवा माटेनी syntax:

```
document.getElementById("hiddenfieldName");
```

Properties:

- **name:** Name attribute ने reflect करे छे.
- **type:** Type attribute ने reflect करे छे.
- **value:** Hidden object नी current value ने reflect करे छे.

Check Box Objects

- Check Box Object HTML <input> element ने type="checkbox" represent करे छे अने ते on अने off value ने set करीने उपयोग करे छे.
- Javascript मां text objects ने access करवा माटेनी syntax:

```
document.getElementById("myCheckID");
```

Properties:

- **checked:** Checkbox नी current state ने indicate करे छे.
- **name:** Name attribute ने indicate करे छे.
- **type:** Type attribute ने indicate करे छे.
- **value:** Value attribute ने indicate करे छे.

Methods:

- **blur:** Checkbox मांथी focus remove थए जाय छे.
- **click:** Checkbox पर mouse click ने indicate करे छे.
- **focus:** Checkbox पर focus set करे छे.
- Checkbox objects नी basic events **onClick, onBlur, onFocus** वरीे.

Radio Box Objects

- Radio Box object HTML <input> element नी type="radio" साथे represent करे छे अने value ने on अथवा off करवा माटे तेनो उपयोग थाय छे.
- Javascript मां text object ने access करवा माटेनी syntax:

```
document.getElementById("myRadioID");
```

Properties:

- **checked:** Boolean property जो radio button नी current state ने reflect करे छे, जो ते check करेल होय तो "true" अने जो ना होय तो "false".
- **name:** ते name attribute ने indicate करे छे.
- **type:** ते type attribute indicate करे छे.
- **value:** ते value attribute ने indicate करे छे.

Methods:

- **blur:** Radio Button मांथी focus remove करे छे.

- **click:** Radio Button पर mouse-click indicate करे छे.
- **focus:** Radio Button पर focus आपे छे.
- Radio Button object नी Basic events **onMouseOver, onMouseDown, onMouseUp, onClick, onBlur, onFocus** वर्गे.

Selecting Objects

- Select object HTML<select> element ने represent करे छे. Javascript मां text object ने access करवा माटेनी Syntax:

```
document.getElementById("mySelectID");
```

Properties:

- **length:** Select object मां length ने get करवा माटे उपयोग थाय छे.
- **name:** Select box मां name ने get करवा माटे उपयोग थाय छे.
- **text:** Select box मां present text value ने return करे छे.
- **Disable:** Drop-down list disable छे के नहि ते set करे छे अथवा return करे छे.

Methods:

- **blur():** Dynamically रीते Radio button ने blur घनाववा माटे उपयोग थाय छे.
- **focus():** Dynamically रीते radio button पर focus मेगववा माटे उपयोग थाय छे.
- Radio Button object नी Basic events **onMouseOver, onMouseDown, onMouseUp, onChange, onBlur, onFocus** वर्गे.

Form Validation

- Form validation ए form ने check करवानी process छे. ते process पूरी थय जाय ते पहेला form ने correctly fill करवामां आवे छे.
- उदाहरण तरीके, जो तमारा form मां user नुँ email address type करवा माटे नुँ box होय अने तमे बीजा form मां work करता पहेला तमारुँ email address fill थयुँ छे के नहि ते चेक करी शको छे.
- Validating forms मां बे main method छे.
 - **Server side (using CGI scripts, ASP)**
 - **Client side (usually done using javascript)**

Process:

- **Server side form validation** नो उपयोग server मां थाय छे, त्यार पछी client जस्ती data ने enter करी ने submit button press करे छे.
- जो client द्वारा enter करवामां आवेला डेटा खोटो अथवा missing होय तो server ने बधो ज डेटा client पर पाइछो मोकलवो पडशे अने ते form नी साची माहिती ने फरीथी enter करी ने submit करवा माटे नी request करवी पडशे.

- આ process ધણી લાંબી અને time consuming છે.
- તેના માટે javascript, વેબ સર્વર પર મોકલતા પહેલા client ના computer પર form ના data ને check કરવાની facility આપે છે. જેને form validation કહેવાય છે. તે બે type ના function perform કરે છે.
 - Basic Validation:** આ method form ની અંદર data enter થાયો છે કે નહિ તે check કરે છે. તેનો meaning એવો થશે કે દરેક form ની અંદર ડેટા હોવો જોઈએ.
 - Data Format Validation:** આ method form ની અંદર જે ડેટા enter થાય છે તેની value correct છે કે નહિ તે check કરશો. ઉદાહરણ તરીકે, જ્યારે કોઈ યુઝર ફોર્મ નંબર ના field માં character enter કરશે ત્યારે error આવશે કારણકે ફોન નંબર ની અંદર ખાલી number ની value ને જ અનુભવ કરવામાં આવ્યું છે.

Testing Data:

- Testing data** એ એવી process છે જે entered કરેલો ડેટા એ clean, correct અને useful છે તે check કરે છે.
- Testing data માટે ના task:
 - બધી જ required fields એ યુઝર દ્વારા enter થયેલી છે કે નહિ.
 - નંબર ફોર્મટ અથવા રેંજ (ઉદાહરણ તરીકે, Zip code એ 5-digit નો હોવો આવશ્યક છે).
 - Entry એ specific word સાથે થાય છે.
 - Enter કરેલી બે entries થાય છે કે નહીં. (ઉદાહરણ તરીકે, password ની double entry).
 - Specific pattern માટે (ઉદાહરણ તરીકે, Email address).

Trapping Empty Fields

- Form validation** નું પહેલું step એ null value ને ચેક કરવાનું છે. નીચે આવેલા ફૂકશાન નો ઉપયોગ એ user એ આપેલ field માં કઇપણ પણ add કર્યું છે કે નહિ તે જોવા માટે ઉપયોગ થાય છે. ખાલી fields એ બે પ્રકારની value ને બતાવે છે. Zero અથવા Null value.

Example:

```
<script type="text/javascript">
    function validateForm()
    {
        var Name = document.getElementById("txtname").value;
        // If the length of the element's string is 0 then display alert message
        if(Name == null || Name == "" || Name.length == 0)
        {
            alert("Please Enter your Name.");
            return false;
        }
    }

```

- ```

 }
 </script>
• ઉપરના code માં જો form field (txtname) empty હોય તો આ ફંક્શન submit થવાથી form ને
અટકાવવા માટે એક alerts message આપે છે. False value ને return કરે છે.

```

### Finding Invalid Values (Regular expressions)

- **Regular expressions** માં performing pattern મેચ કરવા માટે પાવરફૂલ ટૂલ છે. Regular expression નો ઉપયોગ બધા જ પ્રકારના text search અને text replace ઓપરેશન ને performed કરવા માટે ઉપયોગ થાય છે.
- Regular expression નો ઉપયોગ કરીને તમે user દ્વારા add કરેલા data ને easily validate કરી શકો છો અને user ને તે error ને related error આપી શકો છો.
- Javascript માં regular expression ને બે રીતે implement કરી શકાય છે.

#### 1. Using literal syntax.

```
var RegularExpression = /pattern/
```

#### 2. When you need to dynamically construct the regular expression, via the RegExp() constructor.

```
var RegularExpression = new RegExp("pattern");
```

#### Example:

```

<html>
<head>
 <title>Email Validation</title>
 <script>
 function ValidateEmail(inputText)
 {
 var mailformat = /^[\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$]/;
 //Regular Expression for email validation
 if(inputText.value == "")
 {
 alert("Please Enter an Email");
 document.FormValidate.email.focus();
 return false;
 }
 else if(inputText.value.match(mailformat))
 {
 alert("Great!! you entered a right email");
 document.FormValidate.email.focus();
 }
 }
 </script>
</head>
<body>
 <form name="FormValidate">
 <input type="text" name="email" placeholder="Enter Email Address" />
 <input type="button" value="Validate" onclick="ValidateEmail(this.form.email.value)" />
 </form>
</body>

```

```

 return true;
 }
 else
 {
 alert("You have entered an invalid email address!");
 document.FormValidate.email.focus();
 return false;
 }
}
</script>
</head>
<body>
 <form name="FormValidate" action="#">
 <input type='text' name='email' />
 <input type="submit" name="submit" value="Submit"
 onclick="ValidateEmail(document.FormValidate.email);"/>
 </form>
</body>
</html>

```

### Intercepting the Submit Button

- JavaScript मां form submit थाय ते पहेला form items ने validate करवा माटे JavaScript ॲटिन लघवू सरण छे.
- Name अने action साथे form टेग ने define करीने, तमारा javascript code ने call करवा माटे onsubmit event उमेरवी जस्ती छे. फूकशन नी अंदर, False return करे जो validation fails थाय तो.
- नीये आपेला उदाहरण, ज्यारे user submit बटन ने क्लिक करे छे त्यारे onsubmit event javascript ने “**ValidateDate()**” call करे छे.
- जो **validation fails** थाय छे तो function call थाय छे अने form submit थतु नथी.

#### Example:

```

<html>
 <head>
 <script language="javascript">
 function ValidateData(txtname)
 {
 if ((txtname.length==0) || (txtname.value==""))
 {

```

```

 alert("Please Enter you Name");
 txtname.focus();
 return false;
 }
 return true;
}
</script>
</head>
<body>
<form name="myform" onsubmit="ValidateData(this.txtname); " action="#">
<p>Enter Name:

<input type="text" name="txtname">
<input type="submit" name="submit">
</form>
</body>
</html>

```

### Validating Non-text Form Objects

- બધા Non-text content નો meaning એ છે કે યુઝર પાસે રજૂ કરાયેલ text એ alternative છે. જે સમાન હેતુ માટે કામ કરે છે. Non-text content ના ઉદાહરણ નીચે મુજબ છે: Images, specific sensory experience, Word Art, video/audio files વગેરે.

#### Example:

```

<script language="javascript">
 function CheckUpload()
 {
 var FileName = document.getElementById('FileUpload');
 var FileNameValue = FileName.value;
 var ext = FileNameValue.substring(FileNameValue.lastIndexOf('.') + 1);
 if(ext == "gif" || ext == "GIF" || ext == "JPEG" || ext == "jpeg" || ext == "jpg"
 || ext == "JPG")
 {
 alert("Great!! your file is validate for uploading");
 return true;
 }
 else
 {

```

```

 alert("Only Upload GIF or JPEG images... ");
 FileName.focus();
 return false;
 }
}

</script>

```

- ઉપરની script નો ઉપયોગ file ને upload કરવા માટે થાય છે. તેનું extension .jpg or .gif.

### Windows Object

- Browser માં open window એ window object ને represent કરે છે.
- Browser દ્વારા window નાલ object automatically create થાય છે.
- જો document માં frames (<iframe> tags) હોય તો HTML document માટે browser એક window object create કરે છે. અને દરેક frame માટે window object બનાવે છે.

#### Properties:

- **closed:** એક boolean value return કરે છે તે indicat કરે છે. કઈ window બંધ થઈ છે કે નહિ.
- **history:** Window માટે history object return કરે છે.
- **status:** Window status bar માં text set અને return કરે છે.
- **location:** Window માટે location object return કરે છે.
- **history:** Window માટે history object return કરે છે.
- **name:** Window નું name set અને return કરે છે.
- **parent:** Current window ની parent window return કરે છે.

#### Methods:

- **alert():** Message અને ok button સાથે alert box display કરે છે.
- **blur():** Current window માંથી focus ને remove કરે છે.
- **confirm():** Dialog box ને message, ok અને cancel button સાથે display કરે છે.
- **open():** New browser window ને open કરે છે.
- **close():** Current window ને close કરે છે.
- **prompt():** Dialog box display કરે છે. જે input માટે visitor ને પૂછે છે.

### Dialog Boxes

- Javascript a basic ત્રણા type ના dialog box ને support કરે છે. આ dialog box alert ને display કરી શકે છે અથવા કોઈ પણ input માંથી confirmation મેળવે છે અથવા user પાસેથી input મેળવે છે.

#### 1. alert() Dialog Box

- **alert()** method specify message અને ok button સાથે alert box display કરે છે.

- જો તમે sure હોય કે user દ્વારા માહિતી આવે છે, તો alert box નો વારંવાર ઉપયોગ થાય છે.

### Example:

```
<script>
 function GetAlert()
 {
 alert("Great! You doing a good job by entering data");
 }
</script>
```

### 2. confirm() Dialog Box

- confirm()** method ok અને cancel button સાથે, કોઈ specify message સાથે એક dialog box display કરે છે.
- જો user કેટલું verify અથવા accept કરવા માંગતા હો તો confirm box નો વારંવાર ઉપયોગ થાય છે. જો user પર "Ok" પર click કરે તો method true return કરે છે બાકી false આપશે.

### Example:

```
<script>
 function GetConfirm()
 {
 var value = confirm("Do you want to continue?");
 if(value == true)
 {
 alert("Great! You will redirect to our home page.");
 return true;
 }
 Else
 {
 alert("Oppps!!! You do not want to continue.");
 return false;
 }
 }
</script>
```

### 3. prompt() Dialog Box

- prompt()** method એક dialog box display કરે છે. જે visitor ને input માટે પૂછે છે.
- prompt box નો વારંવાર ઉપયોગ થાય છે. જો user page માં enter કરતા પહેલા value ને input કરવા માંગો છો.

- જો user "Ok" પર ક્લિક કરે છે તો prompt() method ઇન્પુટ વેચ્યુ આપે છે. જો user "Cancel" પર ક્લિક કરે છે, તો method null return કરે છે.

### Example:

```
<script>
 function myFunction()
 {
 var person = prompt("Please Enter Your Name", "Peter Parker");
 if (person != null)
 {
 alert("Hi " + person + "! How are you today?");
 }
 }
</script>
```

### Status Bar Messages

- status property browser ના bottom પર status bar માં text set કરે છે અથવા પહેલા text ને set કરી આપે છે.
- Javascript એ window.status નો ઉપયોગ કરીને status bar માં messages દર્શાવી શકે છે. ઉદાહરણ તરીકે, જ્યારે પણ user hyperlink પર hover કરે, ત્યારે તમે javascript status bar message display કરી શકો છો.

### Example:

```
<a href="JavascriptDemo/index.php" onMouseover="window.status='Status Bar
Message goes here'; return true;" onMouseout="window.status=''; return true;">Hover over
me!
```

- આ status property IE, Firefox, Chrome, Safari અથવા Opera 15 અને newer ના default configuration માં work નથી કરતી.

### Window Manipulations

- javascript નો ઉપયોગ કરીને આપણે window માં જુદી જુદી રીતે ફેરફાર કરી શકીએ છીએ. નવી window open કરી શકીએ, close કરી શકીએ. તેના page ને ફરીથી load કરી શકીએ, નવી window ના attribute બદલી શકીએ, thumbnail image popups, browser windows ને resize કરવું window content ને scroll કરવું.
- javascript તમને browser window ના ક્યાં part માં દેખાય તે નક્કી કરવા command નો ઉપયોગ કરવાનું allow કરે છે. આ window.open command ના ત્રીજા ભાગની મદદથી કરવામાં આવે છે. આ તે છે. જ્યાં તમે window ના feature decide કરી શકાય છો:

```
win = window.open("URL Here", "The_First_Windows");
```

- `win.close();`
- તમારી નવી window માં તમે ધણું બધું include કરી શકો છો.

```
// move the screen position of the window to the specified x and y offset
window.moveTo(iX, iY)

// move the screen position of the window by the specified x and y offset
window.moveBy(iX, iY)

// scroll the window to the specified x and y offset
window.scrollTo(iX, iY)

// scroll the window by the specified x and y offset
window.scrollBy(iX, iY)

// set the size of the window to the specified width and length
window.resizeTo(iWidth, iHeight)

// change the current size of the window by the specified x and y offset
window.resizeBy(iX, iY)
```

### Examples of Manipulating Windows:

- નીચેના કોડમાં 100 pixels width અને 100 pixels height વાળી એક window ખુલશે. જ્યારે તમે 'Resize Window' કરો અથવા 'Move Window' બટન પર ક્લિક કરો છો, ત્યારે તે આપેલા parameter ની value મુજબ open window ની size બદલશે અથવા move કરશે.

```
<html>
<head>
<script>
 var NewWindow;
 function openwindow()
 {
 NewWindow = window.open("", 'width=100,height=100');
 NewWindow.focus();
 }
 function resizeWindow()
 {
 NewWindow.resizeBy(100, 100);
 NewWindow.focus();
 }
 function moveWindow()
 {
```

```

 NewWindow.moveTo(300, 200);
 NewWindow.focus();
 }
</script>
</head>
<body>
 <div align="center">
 <button onclick="openwindow()">Create Window</button>
 <button onclick="resizeWindow()">Resize Window</button>
 <button onclick="moveWindow()">Move Window</button>
 </div>
</body>
</html>

```

### Dates Objects

- Date એ javascript માં predefind object છે. Date object નો ઉપયોગ dates અને times સાથે work કરવા માટે થાય છે. Local time અને UTC(universal or, GMT) time નો ઉપયોગ year, month, day, hour, minute, second અને millisecond ને get અને set કરવા માટે allow કરે છે.
- Date objects એ **new Date()** સાથે create થાય છે. javascript માં ચાર રીતે date ને declare કરી શકાય છે.

#### Syntax:

```

var NewDate = new Date();
var NewDate = new Date(milliseconds);
var NewDate = new Date(dateString);
var NewDate = new Date(year, month, date[, hour, minute, second, millisecond])
 //Parameters in the square brackets is optional.

```

#### Date Properties:

- અહીં દરેક properties નું list અને તેનું description આપેલું છે.  
**constructor:** specify function એ object નું prototype create કરે છે.  
**prototype:** prototype property એ object માં property અને method ને add કરવા માટે allow કરે છે.

#### Date Methods:

- Date object ની મહત્વ ની method નું list નીચે મુજબ છે.
  - **getDate():** Month (from 1-31) ની date returns કરે છે.
  - **getDay():** Week (from 0-6) ના day returns કરે છે.

- **getFullYear()**: Year (from digits) return કરે છે.
- **getHour()**: Hour (from 0-23) return કરે છે.
- **getMilliseconds()**: Millisecond (from 0-999) return કરે છે.
- **getMinutes()**: Minutes (from 0-59) return કરે છે.
- **getMonth()**: Month (from 0-11) return કરે છે.
- **getSecond()**: Seconds (from 0-59) return કરે છે.
- **getTime()**: 1 January, 1970 થી miliseconds return કરે છે.
- **getUTCDate()**: Universal time (from 1-131) ને આધારે month ના day return કરે છે.
- **getUTCFullYear()**: Universal time (four digits) ને આધારે year return કરે છે.

### Using and manipulating dates

- Date object એ date સાથે work કરે છે. તે date માં year, month, day, hour, minute, second અને milliseconds હોય છે.

#### Example:

```
<script>
```

**//Display current date and time.**

```
var DateManipulate = new Date();
document.write("Current Date and Time is:" + DateManipulate + "
");
```

**//Using new Date(date string), creates a new date object from the specified date and time.**

```
var DateManipulate = new Date("March 26, 2014 12:30:25");
document.write("Create a new date object from the specified date:" + DateManipulate +
"
");
```

**//Using new Date(number), creates a new date object as zero time plus the number.**

```
var DateManipulate = new Date(86400000689);
document.write("Create a new date object as zero time plus the number:" +
DateManipulate + "
");
```

**//Using new Date(7 numbers), creates a new date object with the specified date and time.**

```
var DateManipulate = new Date(2010,5,15,5,40,30);
document.write("Create a new date object from the specified date:" + DateManipulate +
"
");
```

**//Return the name of the weekday**

```
var d = new Date();
var weekday = new Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday");
document.write(weekday[d.getDay()] + "</br>");
</script>
```

**Output:**

Current Date and Time is:Mon Apr 06 2015 13:48:41 GMT+0530 (India Standard Time)

Create a new date object from the specified date:Wed Mar 26 2014 12:30:25 GMT+0530 (India Standard Time)

Create a new date object as zero time plus the number:Wed Sep 27 1972 05:30:00 GMT+0530 (India Standard Time)

Create a new date object from the specified date:Tue Jun 15 2010 05:40:30 GMT+0530 (India Standard Time)

Monday

### Displaying the date and time

- Javascript ની **Date()** method એ આજની date અને time આપે છે અને call માટે કોઈ object ની જરૂર નથી.

**Example:**

```
<script type="text/javascript">
 var displayDate = Date();
 document.write("Date and Time: " + displayDate);
</script>
```

**Output:**

Date and Time: Mon Apr 06 2015 12:43:15 GMT+0530 (India Standard Time)

### TimeZones

- **gettimezoneoffset()** method એ UTC time અને local time વચ્ચે ના સમય નો તફાવત minute માં આપે છે.
- ઉદાહરણ તરીકે, જો તમારો time zone GMT+10 છે, તો -600 return કરશે. Daylight saving નો time આ value ને constant બનતા અટકાવે છે.
- **Universal Coordinated Time (UTC)** એ world time standard દ્વારા નક્કી કરવામાં આવેલો સમય છે. UTC time **GMT (Greenwich Mean Time)** time જેટલો જ છે.

**Example:**

```
<script type="text/javascript">
 var date = new Date();
 var minutes = date.getTimezoneOffset();
 if (minutes < 0)
 document.write(minutes / 60 + " hours after UTC" + "</br>");
 else
 document.write(minutes / 60 + " hours before UTC" + "</br>");
</script>
```

**Output:**

-5.5 hours after UTC

### Extracting the Date

- Javascript date **getDate()** method એ local time ને અનુસાર specify date માટે month ની day return કરે છે. getDate() દ્વારા return થયેલ value 1 અને 31 integer ની વાંચે હોય છે.

**Example:**

```
<script type="text/javascript">
 var displayDay = new Date();
 document.write("Current day using getDate() : " + displayDay.getDate() + "</br>");
 var displayDay = new Date("December 25, 1995 23:15:00");
 document.write("Current Day using getDate() : " + displayDay.getDate() + "</br>");
</script>
```

**Output:**

Current day using getDate() : 6

Current Day using getDate() : 25

### Extracting the Hrs.

- Javascript date **getDate()** method local time ને અનુસાર specify date માટે month ની hour return કરે છે.

**Example:**

```
<script type="text/javascript">
 var displayHour = new Date();
 document.write("Get current hour using getHours() : " + displayHour.getHours() +
 "</br>");
 var displayHour = new Date("December 25, 1995 23:15:00");
```

```
document.write("Get current hour using getHours : " +
displayHour.getHours() +
"</br>");

</script>
```

### **Output:**

Get current hour using getHours() : 12  
Get current hour using getHours getHours() : 23

## **The Math Object and its constants**

- Math object એ નંબર પર mathematical task કરવા માટે allow કરે છે. Math object તમને mathematical constant અને functions માટે properties અને methods આપે છે.
- બીજી global object એ Math constructor નથી, Math ની બધી properties અને method static છે અને તેને create કર્યા વગર object તરીકે Math નો ઉપયોગ કરી શકાય છે.

### **Syntax:**

- Math ની properties અને methods ને call કરવા ની simple syntax:

```
var pi_val = Math.PI;
var sine_val = Math.sin(30);
```

## **Math Object Properties (Mathematical Constants)**

- Javascript 8 mathematical constant આપે છે. જે Math object સાથે access કરી શકાય છે.
  - **E:** Euler નાલ number return કરે છે.
  - **LN2:** 2 નો natural logarithm return કરે છે.
  - **LN10:** 1 નું natural logarithm return કરે છે.
  - **LOG2E:** E ની base-2 logarithm return કરે છે.
  - **LOG10E:** E નાલ base-10 logarithm return કરે છે.
  - **P1:** P1 return કરે છે.
  - **SQRT1\_2:**  $\frac{1}{2}$  નું square root return કરે છે.
  - **SQRT2:** 2 નું square root return કરે છે.

## **Math Object Methods**

- **abs(x):** x નું absolute value return કરે છે.
- **ceil(x):** નાળુક ની integer સુધી ઉપરની value ને rounded કરીને આપે છે.
- **cos(x):** x નું cosine return કરે છે (x radians)
- **floor(x):** નાળુક નાલ integer સુધી નીચેની value ને rounded કરીને આપે છે.
- **log(x):** x નું natural logarithm (base E) return કરે છે.
- **max(x,y,z,...,n):** highest value સાથેનો નંબર return કરે છે.

- **min(x,y,z,...,n):** lowest value સાથેનો નંબર return કરે છે.
- **pow(x,y):** y ની power થી x ની value return કરે છે.
- **random():** 0 અને 1 ની વચ્ચે random નંબર return કરે છે.
- **round(x):** x ને નજીક ના integer સુધી પહોંચાડે છે.
- **sin(x):** x ની sine value return કરે છે. (x એ radians માં છે)
- **sqrt(x):** x નું square root return કરે છે.
- **tan(x):** ખૂણા ના tangent ને return કરે છે.

### Example:

```
<script>
 document.write("Value of Math.PI:" + Math.PI + "</br>");
 document.write("Value of Math.ceil(2.5):" + Math.ceil(2.5) + "</br>");
 document.write("Value of Math.floor(4.7):" + Math.floor(4.7) + "</br>");
 document.write("Value of Math.random()*10:" + Math.floor(Math.random()*10) + "</br>");
 document.write("Value of Math.max(10,10.5,24):" + Math.max(10,10.5,24) + "</br>");
 document.write("Value of Math.pow(5, 2):" + Math.pow(5,2) + "</br>");
 document.write("Value of Math.sqrt(81):" + Math.sqrt(81) + "</br>");
</script>
```

### Output:

Value of Math.PI:3.141592653589793

Value of Math.ceil(2.5):3

Value of Math.floor(4.7):4

Value of Math.random()\*10:9

Value of Math.max(10,10.5,24):24

Value of Math.pow(5, 2):25

Value of Math.sqrt(81):9

### JQuery concept or what is JQuery?

- **jQuery** એ fast, small અને વધારે feature વાળી javaScript library John Resig દ્વારા 2006 માં બનાવામાં હતી. તેનો હેતુ: **write less, do more** હતો.
- **jQuery** એ HTML document ને traversing, event handling, animating અને Ajax interactions ને simplify કરે છે.
- javascript નાલ complicated things ને jQuery simplify કરે છે. જેમ કે, Ajax call અને DOM manipulation.
- jQuery library માં નીચેના features હોય છે.
  - HTML/DOM manipulation
  - CSS manipulation
  - Event handling
  - Effects અને animations
  - AJAX Support
  - Lightweight
  - Cross Browser Support
- JQuery ધ્યાન ભધી રીતે ઉમેરી શકાય છે. તમે આ રીતે કરી શકો છો. (1). jQuery.com માંથી jQuery library ને download કરો અથવા (2). CDN માંથી jQuery ને include કરી શકો છો. જેમ કે Google.
- તેને તમે HTML<script> tag સાથે add/reference કરી શકો છો. (<script> tag એ <head> tag section ની અંદર હોય છે.)

### JQuery Syntax:

- jQuery ની Syntax એ selecting HTML elements પર અમૃક action perform કરવા માટે બનાવામાં આવી છે.

```
$(selector).action()
 ➤ A $ sign to define/access jQuery
 ➤ A (selector) to "query (or find)" HTML elements
 ➤ A jQuery action() to be performed on the element(s)
```

### Examples:

```
$(this).hide() - hides the current element.
$("p").hide() - hides all <p> elements.
$(".test").hide() - hides all elements with class="test".
$("#test").hide() - hides the element with id="test".
```

- ઉદાહરણ માની બધી જ jQuery મેથ્સ, document ready event ની અંદર છે.

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

```
 // jQuery methods go here...});
```

- ડોક્યુમેન્ટ સંપૂર્ણ load થાય ત્યાર બાદ **\$(document).ready()** method એ function ને execute કરવા માટે allow કરે છે.

### jQuery Events

- Event એ વિવિધ પ્રકાર ના visitor ની action છે જે webpage/web application ને respond આપી શકે છે. જ્યારે કોઈ moment થાય ત્યારે ચોક્કસ event ને represent કરે છે.
- જુદી જુદી event ના ઉદાહરણ નીચે મુજબ છે.
  - A mouse click
  - Selecting a radio button
  - A web page loading
  - Moving a mouse over an elements
  - Submitting an HTML form
  - A keystroke on your keyboard
- નીચે કેટલીક સામાન્ય રીતે વપરાયેલી DOM events છે.

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	Keypress	submit	load
dblclick	Keydown	change	resize
hover	Keyup	focus	scroll
mouseenter		blur	unload
mouseleave			

### Mouse Events

- Mouse events એવી events છે જે mouse ની action સાથે હોય છે. જેમ કે, click, double-click, mouse down, hover વગેરે હોય શકે છે.
- Mouse events નો ઉપયોગ બે જુદી જુદી રીતે કરી શકાય છે.
  - Click handler ને HTML elements સાથે bind કરવાનો અને,
  - jQuery code ની અંદર click events ને call કરવાનો છે.

#### 1) click()

- click() method એ event handler function ને HTML element સાથે જોડે છે. જ્યારે user HTML elements પર ક્લિક કરે છે ત્યારે ફુંક્શન execute થાય છે.

### Syntax:

**Selected elements माटे click event ने trigger करवा माटे :** \$(selector).click()

**Click event मां function ने attach करवा माटे :** \$(selector).click(function) // function is optional

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
 $(document).ready(function(){
 $("p").click(function(){
 $(this).hide();
 });
 });
</script>
```

#### HTML Code:

<p>Click Me</p>

- ઉપરના ઉદાહરણમાં, <p> element पર ક્લિક કરતી વખતે click() method triggered થાય છે.

## 2) dblclick()

- dblclick event ત્યારે આવે છે જ્યારે element પર double-click કરવામાં આવે છે. dblclick() method dblclick events ને trigger કરે છે અથવા જ્યારે dblclick event occurs થાય ત્યારે run કરવા માટે ફુંક્શન ને attached કરે છે.

### Syntax:

**Selected elements माटे click event ने trigger कરવा माटे:** \$(selector).dblclick()

**Click event मां function ने attach કરવा माटे:** \$(selector).dblclick(function) // function is optional

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
 $(document).ready(function(){
 $("p").dblclick(function(){
 $(this).hide();
 });
 });
</script>
```

### HTML Code:

```
<p>Double-Click on Me</p>
```

### 3) hover()

- hover() method જ્યારે mouse pointer એ select કરેલા HTML element માં enter થાય ત્યારે અને તેને leave કરે છે ત્યારે fire થાય છે.
- hover() methodમાં બે ફુંક્શન છે અને તે mouseenter() અને mouseleave() method નું combination છે. માઉસ જ્યારે HTML elements માં enter થાય છે ત્યારે પહેલું ફુંક્શન executed થાય છે અને માઉસ જ્યારે HTML elements leave કરે છે ત્યારે બીજું ફુંક્શન executed થાય છે.

### Syntax:

```
$(selector).hover(inFunction, outFunction)
```

- **inFunction** એ required છે જ્યારે mouseenter event occurs થાય ત્યારે run કરવા માટે આ ફુંક્શન specify કરવામાં આવે છે.
- **outFunction** એ optional છે જ્યારે mouseleave event occurs થાય ત્યારે run કરવા માટે આ ફુંક્શન specify કરવામાં આવે છે.

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("#MouseHover").hover(function(){
 $("div").css("background-color", "pink");
 }, function(){
 $("div").css("background-color", "lightblue");
 });
});
</script>
```

#### HTML Code:

```
<div id="MouseHover">Hover the mouse pointer to see the effect.</div>
```

- ઉપર નો code <div> elements નો background color બદલી નાખે છે જ્યારે માઉસ નું pointer તેના પર hover થાય છે.

### Keyboard Events

- **jQuery keyboard events** કોઈ પણ HTML element સાથે હોઈ શકે છે. પરંતુ તે ફક્ત જે element ને focus કરે છે તે event ને fire કરશે.

- મોટા ભાગ ની કીબોર્ડ event form elements (HTML input elements) સાથે જોડાયેલ હોય છે. જેમ કે basic કીબોર્ડ events **keypress, keydown, keyup, keyrelease** વગેરે.

### 1) **keypress()**

- કીબોર્ડ માં keypress કરવામાં આવે પછી **jquery keypress event** trigger થાય છે. જ્યારે keypress event occur થાય ત્યારે તેમાં attach કરેલ ફંક્શન run થાય છે.
- Keypress event એ keydown event જેવી જ છે. જ્યારે બટન press કરવામાં આવે ત્યારે event occur થાય છે.
- જ્યારે modifier keys (**ALT, CTRL, SHIFT, and ESC**) press કરવામાં આવે છે. ત્યારે keypress event fire થશે નહિં.

#### Syntax:

**Selected elements માટે click event ને trigger કરવા: \$(selector).keypress()**

**Click event માં function ને attach કરવા માટે: \$(selector).keypress(function) // function is optional**

#### Example:

##### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
 i = 0;
 $(document).ready(function(){
 $("#txtName").keypress(function(){
 $("label").text(i += 1);
 });
 });
</script>
```

##### HTML Code:

Enter your name: <input type="text" id="txtName">

<p>Total Keypresses are: <label>0</label></p>

- ઉપરોક્ત code એ <input> field માં keypress થાય એ નંબર ને count કરે છે. જે user દ્વારા enter કરવામાં આવે છે.

### 2) **keydown()**

- જ્યારે કીબોર્ડ ની key press કરવામાં આવે ત્યારે keydown event trigger થાય છે. **keydown()** method એ **keydown** event ને trigger કરે છે અથવા જ્યારે keydown event થાય ત્યારે તેમાં attach કરેલ ફંક્શન run થાય છે.

- જ્યારે modifier keys (**ALT, CTRL, SHIFT, and ESC**) press કરવામાં આવે છે ત્યારે keydown event fire થશે.

### Syntax:

**Selected elements માટે click event ને trigger કરવા:** \$(selector).keydown()

**Click event માં function ને attach કરવા માટે:** \$(selector).keydown(function) //function is optional

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("#txtName").keydown(function(){
 $("#txtName").css("background-color", "pink");
 });
 $("#txtName").keyup(function(){
 $("#txtName").css("background-color", "lightblue");
 });
});
</script>
```

#### HTML Code:

<p>Enter your Name and see the effect of keydown and keyup event.</p>

Enter your name: <input type="text" id="txtName">

- જ્યારે કીબોર્ડ ની keypress કરવામાં આવે અથવા release કરવામાં આવે ત્યારે ઉપર ના કોડ માં <input> field નો background color set કરે છે.

### 3) keyup()

- જ્યારે keypress અને પછી release કરવામાં આવે ત્યારે keyup event fire થાય છે. જ્યાં સુધી keyrelease નહિ થાય ત્યાં સુધી આ �event fire થશે નહીં. **keyup()** method keyup ની event ને trigger કરે છે અથવા જ્યારે keyup event થાય ત્યારે તેમાં attach કરેલ ફંક્શન run થાય છે.
- keyup અને keypress / keydown વચ્ચે એ તફાવત છે કે જ્યારે key નીચે hold કરેલી હોય ત્યારે આ event fire નહિ થાય પરંતુ બીજી keyboard event key નીચે hold કરેલી હશે ત્યારે fire થશે.

### Syntax:

**Selected elements માટે click event ને trigger કરવા:** \$(selector).keyup()

**Click event માં function ને attach કરવા માટે:** \$(selector).keyup(function) // function  
is optional

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>

$(document).ready(function(){

 $("#txtName").keydown(function(){

 $("#txtName").css("background-color", "pink");

 });

 $("#txtName").keyup(function(){

 $("#txtName").css("background-color", "lightblue");

 });

});

</script>
```

#### HTML Code:

<p>Enter your Name and see the effect of keydown and keyup event.</p>

Enter your name: <input type="text" id="txtName">

- **Note:** jQuery માં કોઈ keyrelease event નથી પરંતુ આપણે તેને keyup ની event તરીકે લઇ શકીએ છીએ.

## Form Events

- jQuery form elements ની action ને handle કરવા માટે ધણી form events આપે છે. Form events એ standard JavaScript events છે પરંતુ jQuery દ્વારા add કરેલ ધણી જુદી જુદી events support છે. જેમ કે submit, change, select, blur વગેરે.

### 1) submit()

- Submit event ત્યારે આવે છે જ્યારે કોઈ form એ form submission element (button, input (type button અથવા image) અને enter (અમૃક caseમાં) નો ઉપયોગ કરે. આ event નો ઉપયોગ <form> elements પર જ થઈ શકે છે.
- submit() મેથડ એ form ને સબમિટ કરો ત્યારે event ને trigger કરે છે અથવા submit event થાય તેમાં attach કરેલ ફંક્શન run થાય છે.

#### Syntax:

**Selected elements માટે click event ને trigger કરવા:** \$(selector).submit()

**Attach a function to the submit event:** \$(selector).submit(function) // function is optional

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("#Registration").submit(function(){
 alert("Great!! Form submitted Successfully");
 });
});
</script>
```

#### HTML Code:

```
<form id="Registration" action="#">
<label>FirstName: </label> <input type="text" name="FirstName" id= "FirstName" />
 <input type="submit" value="Submit">
</form>
```

### 2) ready()

- જ્યારે DOM load થાય ત્યારે ready event occur થાય છે. કારણ કે ડોક્યુમેન્ટ ready થાય પછી આ event occur થાય છે.
- Ready event નો હેતુ એ છે કે તે ડોક્યુમેન્ટ ને load કર્યા પછી શક્ય તેટલું વહેલું થવું જોઈએ.
- Ready મેથડ સ્પષ્ટ કરે છે જ્યારે ready event થાય ત્યારે શું થાય છે. <Body onload = ""> સાથે ready() મેથડ નો ઉપયોગ એક સાથે થવો જોઈએ નહીં.
- html document load થયા પછી ready event occurs થાય છે જ્યારે બધા content load થઈ જાય પછી onload event occur થાય છે onload event DOM માં standard event છે જ્યારે ready event jquery માટે special છે.

#### Syntax:

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

**Ready () methodનો ઉપયોગ current document પર જ થઈ શકે છે, તેથી કોઈ selector આવશ્યક નથી:** \$(function) // function is required

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("#Registration").submit(function(){
```

```

 alert("Great!! Form submitted Successfully");
 });
});

</script>

```

### HTML Code:

```

<form id="Registration" action="#">
<label>FirstName: </label> <input type="text" name="FirstName" id="FirstName"/>
 <input type="submit" value="Submit">
</form>

```

## Document/ Windows Events

### 1) load()

- જ્યારે parent element ની નીચેના બધા sub element અને બીજી element load થાય છે ત્યારે load() event fire થાય છે
- Load event ત્યારે થાય છે જ્યારે images, iframes, frames, scripts અથવા window ડોક્યુમેટ જેવા specify element સંપૂર્ણ પણે (completely) load થાય.
- Load () method jQuery version 1.8 માં disapprove કરવામાં આવી હતી.

### Syntax:

```
$(selector).load(function) /*function is optional */
```

### Example:

#### jQuery code:

```

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>

$(document).ready(function(){
 $("#LoadImage").load(function(){
 alert("Image loaded successfully.");
 });
});
</script>

```

### HTML Code:

```



```

### 2) resize()

- જ્યારે browserની window resize થાય ત્યારે resize event occur થાય છે. આ event ને properly work કરવા માટે "window" element નું attached હોવું આવસ્યક છે.

- resize() method એ resize event ને trigger કરે છે અથવા જ્યારે resize event થાય ત્યારે તેમાં attached કરેલ ફંક્શન run થાય છે.

### Syntax:

**Selected elements માટે click event ને trigger કરવા: \$(selector).resize()**

**Click event માં function ને attach કરવા માટે: \$(selector).resize(function) // function is optional**

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js">
</script>
<script>
$(document).ready(function(){
 $(window).resize(function(){
 alert("You resized me!");
 });
});
</script>
```

#### HTML Code:

```
<p>Try to resize your browser window.</p>
```

### 3) scroll()

- Scroll event ત્યારે થાય છે જ્યારે યુઝર specify element માં scroll કરે છે.
- Scroll event બધા scrollable elements અને window object માટે work કરે છે.

### Syntax:

**Selected elements માટે click event ને trigger કરવા: \$(selector).scroll()**

**Click event માં function ને attach કરવા માટે: \$(selector).scroll(function) //function is optional**

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"> </script>
<script>
$(document).ready(function(){
 $("#intro").scroll(function(){
 alert("Great!! You scrolled me!");
 });
});
```

```
</script>
```

### HTML Code:

```
<p>Try the scroll in the div</p>
<div id="intro" style="overflow:scroll; height:100px; width: 200px;"> Lorem ipsum
dolor sit amet, consectetur adipiscing elit. Mauris dapibus fermentum leo, ut
consectetur libero sagittis eu. Sed vel interdum diam. Curabitur quis magna
dolor.
</div>
```

### 4) unload()

- જ્યારે યુઝર navigates પેજ થી દુર જાય છે ત્યારે unload event occur થાય છે.
- unload event triggered થાય છે જ્યારે પેજ leave કરવા માટેની link પર ક્લિક કરવામાં આવે છે addressbar માં એક નવું URL type કરવામાં આવે છે તે forward અથવા back button નો ઉપયોગ થાય છે, browser window બંધ થાય છે પેજ ફરીથી load થાય છે
- unload() method જ્યારે unload event થાય છે ત્યારે શું થાય છે તે specify કરે છે.
- unload() method jquery version 1.8 માં disapprove કરવામાં આવી છે.

### Syntax:

```
$(selector).unload(function) // function is optional
```

### Example:

#### jQuery code:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $(window).unload(function(){
 alert("Please Wait for the page load.");
 });
});
</script>
```

### HTML Code:

```
Click Here to go Google.com
```

### bind() Method

- jQuery દ્વારા events સાથે deal કરવાના સૌથી મહત્વ નો factor bind() અને unbind() મેથડ છે.
- Event triggers અને handlers ને તમારા DOM element માં add કરવા માટે bind() method ખૂબજ ઉપયોગી છે અને custom elements ને પણ bind કરી શકે છે

- unbind() method એ event handlers ને select કરેલા element માંથી દુર કરે છે. આ method બધા અથવા select કરેલા event handler ને remove કરે છે અથવા event occur થાય ત્યારે run થતા specify ફંક્શન ને બંધ કરે છે
- Possible event values blur, focus, load, resize, scroll, unload, click વગેરે.
- bind() method select કરેલા elements માટે એક અથવા વધુ event handlers ને જોડે (bind) છે અને જ્યારે event occur થાય ત્યારે run કરવા માટે ફંક્શન specify કરે છે.

### Syntax:

```
$(selector).bind(event, data, function, map)
```

- **event** જરૂરી parameter છે જે event થાય ત્યારે run કરવા માટે ફંક્શન ને specify કરે છે.
- **data** એ optional parameter છે જે ફંક્શન ની સાથે pass કરવા વધારાના data ને specify કરે છે.
- **function** જરૂરી parameter છે જે event થાય ત્યારે run કરવા માટે ફંક્શન ને specify કરે છે.
- **map** event map({event : function ,event : function ,.....}) ને specific કરે છે એક અથવા વધારે events આવેલી હોય છે અને જ્યારે event occur થાય ત્યારે ફંક્શન run થાય છે.

### Example:

#### jQuery code:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
 $(document).ready(function(){
 $("p").bind("mouseout mouseover", function(){
 $("p").toggle(".intro");
 });
 });
</script>
<style>
.intro {
 font-size: 150%;
 color: blue;
}
</style>
```

#### HTML Code:

```
<p class="intro">Hover the mouse pointer over this paragraph.</p>
```

### Event Helper Methods

- jQuery event એ helper ફંક્શન નો એક set પણ આપે છે જેનો ઉપયોગ ઉપર mention કરેલા કોઈ પણ event type ને બંધ કરવા માટે અથવા eventને trigger કરવા માટે ઉપયોગ માં લેવાય છે.
- **Trigger Methods:**  
નીચે આપેલા ઉદાહરણ માં બધા paragraph પર blur event ને trigger કરે છે.  

```
$("p").blur();
```
- **Binding Methods:**  
નીચે આપેલા ઉદાહરણ માં <div> પર ક્લિક event ને bind કરશે.  

```
$("div").click(function () {
 // do something here
});
```
- અહીં JQuery અને અન્ય દ્વારા આપવામાં આવેલી બાકી ની મેથડ નું list છે જે પહેલા જ આ chapter માં ચર્ચા થઈ ચૂકી છે.
  - **blur():** દરેક match થતી element ની blur event ને triggers કરે છે.
  - **blur(fn):** દરેક match થતી element ની blur event માં ફંક્શન ને bind કરે છે.
  - **change():** દરેક match event ની changed event ને triggers કરે છે.
  - **change(fn):** દરેક match થતી element ને change event માં ફંક્શન ને bind કરે છે
  - **error():** દરેક match થતી element ની blur error ને triggers કરે છે.
  - **error(fn):** દરેક match થતી element ની error event માં ફંક્શન ને bind કરે છે.
  - **focus():** દરેક match થતી element ની focus event ને triggers કરે છે.
  - **focus(fn):** દરેક match થતી element ને focus event માં ફંક્શન ને bind કરે છે.
  - **mousedown(fn):** દરેક match થતી element ની mousedown event માં ફંક્શન ને bind કરે છે.
  - **mouseenter(fn):** દરેક match થતી element ની mouseenter event માં ફંક્શન ને bind કરે છે.
  - **mouseleave(fn):** દરેક match થતી element ની mouseleave event માં ફંક્શન ને bind કરે છે.
  - **mousemove(fn):** દરેક match થતી element ની mousemove event માં ફંક્શન ને bind કરે છે.
  - **mouseout(fn):** દરેક match થતી element ની mouseout event માં ફંક્શન ને bind કરે છે.
  - **mouseover(fn):** દરેક match થતી element ની mouseover event માં ફંક્શન ને bind કરે છે.
  - **mouseup(fn):** દરેક match થતી element ની mouseup event માં ફંક્શન ને bind કરે છે.
  - **select():** દરેક match થતી element ની select કરેલી event ને trigger કરે છે.
  - **select(fn):** દરેક match થતી element ની select event માં ફંક્શન ને bind કરે છે.

**Note:** bind() અને helper મેથડ ના ઉદાહરણ ઉપર સમજાવેલા છે. (Refer Page No. 11-12)

### What is Ajax?

- **AJAX** એટલે **Asynchronous JavaScript and XML**.
- **AJAX** scenes ની પાઇળના server સાથે ડેટા ને exchange કરીને વેબપેજ ને **asynchronously** રીતે update કરે છે.
- આનો અર્થ એ છે કે સંપૂર્ણ પેજ ફરીથી load કર્યા વિના વેબપેજ ના ભાગ ને update કરવું શક્ય છે.
- વેબ પર ધાર્યી વેબ એપ્લિકેશન run થાય છે જે Ajax ટેકનોલોજી નો ઉપયોગ કરી રહ્યા છે જેમ કે Gmail, Facebook, Twitter, Google Map, YouTube વગેરે.

### What is the purpose of AJAX? (Ajax Technologies)

- Ajax એ JavaScript, DOM, XML, HTML અને CSS જેવી inter-related ટેકનોલોજી નો ઉપયોગ કરે છે.
- Ajax contentને display કરવા અને style ના presentation માટે **CSS** અને **HTML** નો ઉપયોગ કરે છે.
- User અને application વચ્ચે dynamic interaction મેળવવા માટે JavaScript નો ઉપયોગ કરીને **DOM** ને જોડવામાં આવે છે.
- **XML** સામાન્ય રીતે server માંથી જે ડેટા આવે છે તેને મેળવા માટેના ફોર્મટ તરીકે ઉપયોગમાં લેવાય છે, ટુકમાં, તેનો ઉપયોગ server માંથી ડેટા ને લઈ જવા માટે થાય છે.
- JavaScript અને **XMLHttpRequest** objects એક સાથે server અને browser વચ્ચે asynchronous રીતે ડેટા exchange થાય ત્યારે સપૂર્ણ પેજ ફરીથી load થાય તેને avoid કરે છે.

### What is XMLHttpRequest?

- **XMLHTTP** ને **Microsoft Internet Explorer 5** માં **Microsoft ActiveX control** તરીકે display કરાયું હતું ત્યાર બાદ આ object **Asynchronous JavaScript** અને **XML (AJAX)** અને **Simple Object Access Protocol (SOAP)** પર આધારિત વેબ application ના બીજા browsing platforms દ્વારા implement કરવામાં આવે છે.
- **XMLHTTP** object browser ને વેબ server પર http requests તે બદલે **asynchronous** રીતે **send** અને **receive** કરવા માટે enable કરે છે જે xml સાથે response આપે છે.
- Client-side script સાથે response ને બદલી શકાય છે અથવા Extensible Stylesheet Language Transformations (XSLT) સાથે transform કરી શકાય છે.

### The XMLHttpRequest Object

- **XMLHttpRequest object** નો ઉપયોગ server સાથે ડેટા ને exchange કરવા માટે થાય છે. આનો અર્થ એ છે કે સંપૂર્ણ પેજ ને ફરીથી load કર્યા વિના વેબપેજ ના parts ને update કરવું શક્ય છે.
- બધા modern browser માં XMLHttpRequest object (IE5 અને IE6 use an ActiveXObject) support કરે છે.

### Send a Request to a Server

- Server ને request મોકલવા માટે, આપણે XMLHttpRequest object ને open() અને send() methods નો ઉપયોગ કરીએ છીએ.

### Syntax:

- **open(method, url, async):** Request અને URL ની type specify કરે છે અને request ને asynchronously handle કરવી જોઈએ કે નહીં તે specify કરે છે.
  - **method:** get અથવા post એ request ની type છે.
  - **url:** server પર ફાઇલ નું location દર્શાવે છે.
  - **async:** true (asynchronously) અથવા false (synchronously).
  - **send(string):** request server પર મોકલે છે જ્યાં string નો ઉપયોગ ફક્ત post request માટે થાય છે.

### Server Response

- Server પાસેથી response મેળવવા માટે XMLHttpRequest object ની responseText અથવા responseXML property નો ઉપયોગ કરો.
  - **responseText:** જો server તરફ થી response મળે એ xml ના હોય, તો response text property નો ઉપયોગ કરો.
  - **responseXML:** જો server તરફથી response મળે એ xml હોય અને તમે તેને xmlobj તરીકે parse કરવા માંગો છો, તો responseXML propertyનો ઉપયોગ કરો.

### What is callback function?

- Callback function એ એવું function છે જે બીજા function માં parameter તરીકે pass થાય છે.
- જો તમારી પાસે તમારી વેબસાઇટ પર એક કરતાં વધુ AJAX task છે, તો તમારે XMLHttpRequest object બનાવવા માટે એક standard ફંક્શન બનાવવું જોઈએ અને દરેક AJAX કાર્ય માટે આ functionને call કરવું જોઈએ.

### Example:

```
function AjaxDemo()
{
 LoadAjax("MyInfo.txt", function() //Callback Function
 {
 if (xmlhttp.readyState==4 && xmlhttp.status==200)
 {
 document.getElementById("AjaxData").innerHTML=xmlhttp.responseText;
 }
 });
}
```

### The onreadystatechange event

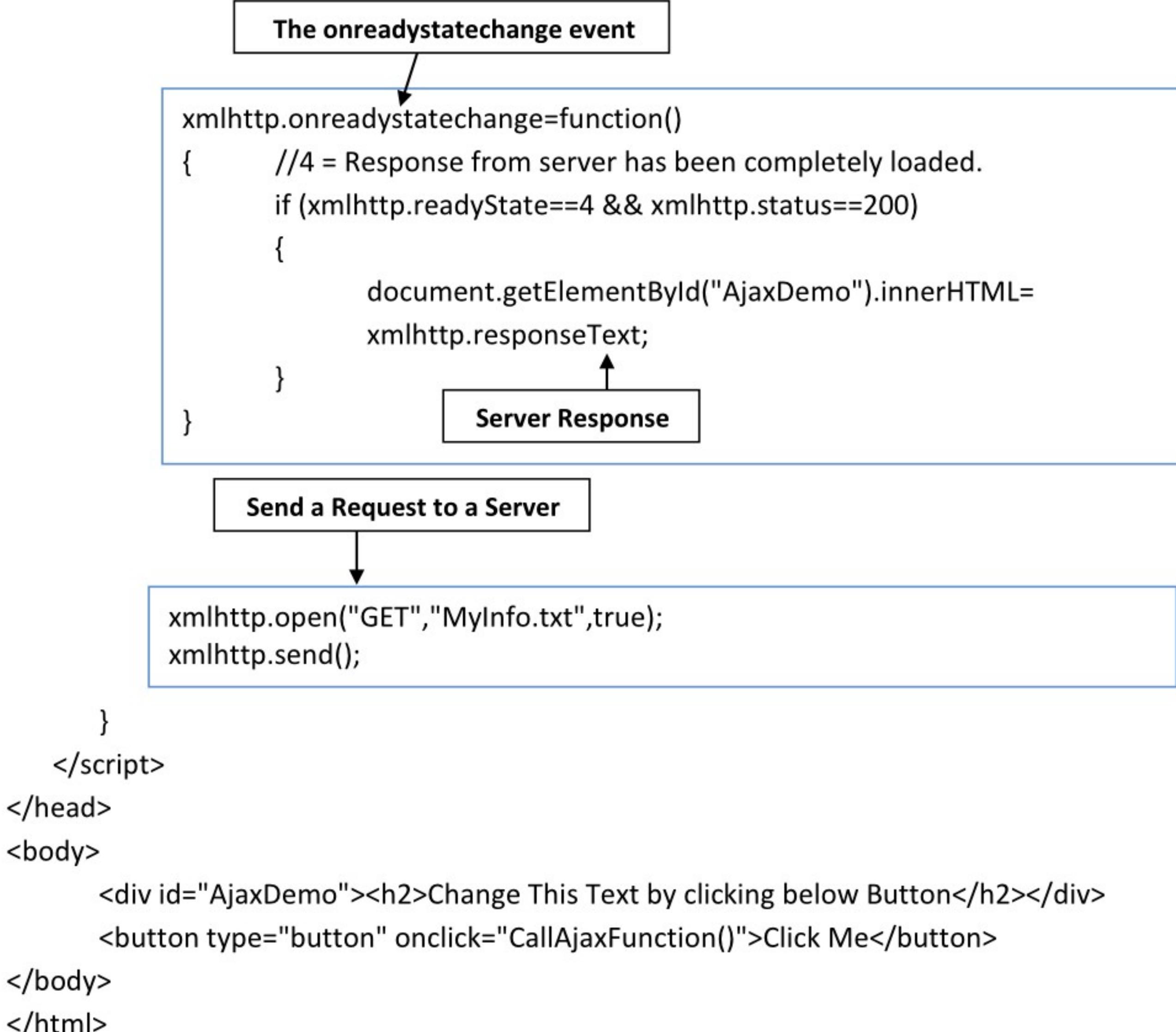
- જ્યારે server ને request મોકલવામાં આવે છે, ત્યારે response પર આધારિત કેટલીક actions ને perform કરવામાં આવે છે જેમ કે **onreadystatechange** event દરેક સમયે **readyState** માં થતા changes ને trigger કરે છે.
- readyState** property XMLHttpRequest નાલ status ને hold કરે છે XMLHttpRequest object ની ત્રણ મહત્વપૂર્ણ properties.
  - onreadystatechange**: જ્યારે ready state એ property ની value ને change કરે છે ત્યારે જે ફુંક્શન ને automatically call કરવા નું હોઈ તેનું નામ અહીં લખવા માં આવે છે.
  - readyState**: XMLHttpRequest નાલ status ને hold રાખે છે અને તે 0 થી 4 માં change થાય છે:
    - 0: request not initialized
    - 1: server connection established
    - 2: request received
    - 3: processing request
    - 4: request finished and response is ready
- status**: Request નાલ response નું status દર્શાવે છે. Ex : 200 = ok અને 404 = page not found

Example:

```

<html>
<head>
<script>
 function CallAjaxFunction()
 {
 Create an XMLHttpRequest Object
 var xmlhttp;
 if (window.XMLHttpRequest)
 {
 // code for IE7+, Firefox, Chrome, Opera, Safari
 xmlhttp=new XMLHttpRequest();
 }
 else
 {
 // code for IE6, IE5
 xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
 }
 }

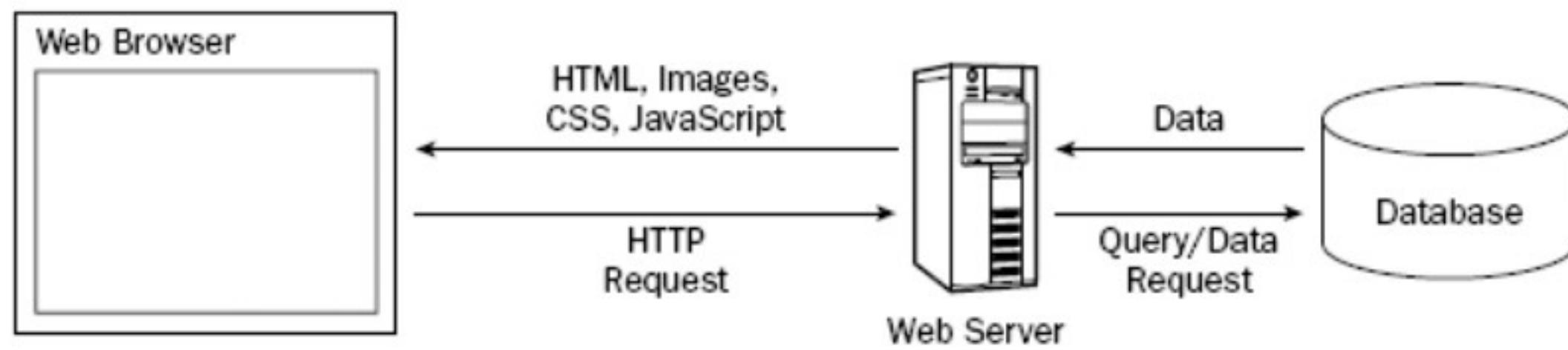
```



### Traditional Web Application (Synchronous)

- **Traditional web applications** માં વેબ સર્વર html વેબપેજ ને display કરે છે કેટલાક વેબપેજ static છે જ્યારે બીજા કેટલાક logic દ્વારા dynamically રીતે generate થાય છે ખાસ કરીને એપ્લિકેશન માં dynamic data હોય, ત્યારે સર્વર આ data ને HTML markup માં convert કરે છે અને તેને HTML પેજ તરીકે display કરવા માટે browser પર send કરે છે.
- **Traditional web application model** આના જેવું જ work કરે છે મોટા ભાગના યુઝરની action એ વેબ સર્વર પર **HTTP request** ને **trigger** કરે છે. સર્વર કેટલીક process કરીને data મેળવે છે અને પછી client ને **HTML page** પાઠું મોકલે છે આ request synchronously કરવામાં આવે છે.
- **Synchronous request** એ operation પૂરું થાય નહિ ત્યાં સુધી client ને block રાખે છે એટલે કે browser responsive નથી તે સમયે user બીજુ operation perform કરી શકતું નથી.

Traditional Web Application Model



Ajax Web Application Model

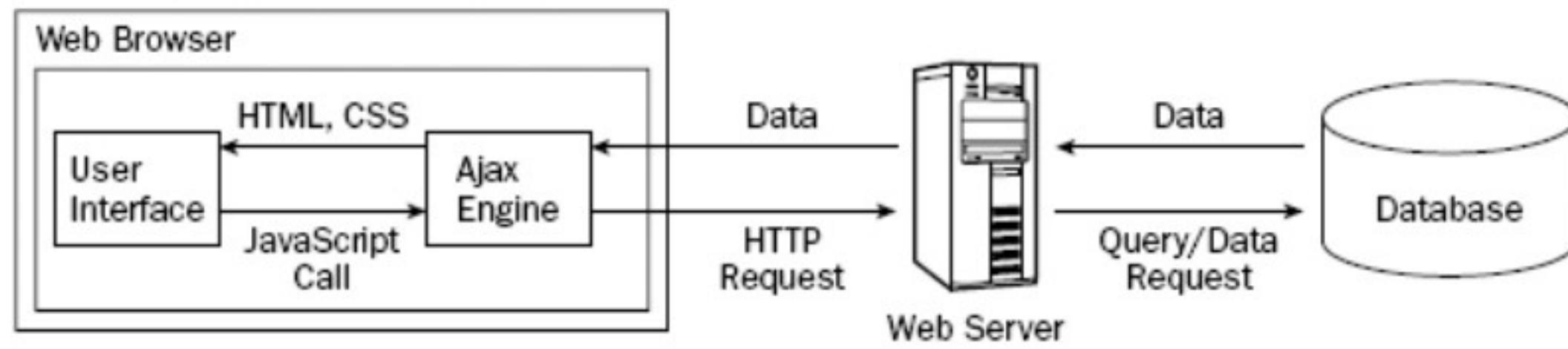


Fig 5.1 the traditional model for web applications (top) compared to the Ajax model (bottom).

### Ajax Web page Application (Asynchronous)

- Ajax model मां Ajax engine बधी request process करे છે request asynchronously રીતે કરવામાં આવી છે, જેનો અર્થ છે કે code નું execution ચાલુ થાય તેની પહેલા response માટે રાહ જોતો નથી.
- Ajax web application માં, સર્વર એ ડેટાને HTML markup માં convert કરવાની જરૂર નથી. Server એ direct ક્લાયન્ટ બાજુ ડેટા મોકલી શકે છે. Client side codeએ browser ની અંદર ડેટાને પ્રોસેસ કરશે અને dynamic રીતે HTML ભાગને update કરશે અને તે મુજબ display કરશે (આફ્ટર 5.1 જુઓ)
- **Asynchronous** request clientને block કરતી નથી એટલે કે browser એ responsive છે તે સમયે ચુઝર બીજુ operation perform કરી શકે છે.

### Use of HTML and XML in Ajax

- **Ajax** એ **CSS** અને **HTML** નો ઉપયોગ content અને style ને display કરવા માટે કરે છે. વધારે ઉપયોગ presentation માટે થાય છે.
- **XML** નો ઉપયોગ સામાન્ય રીતે સર્વર માંથી data ને receive કરવા માટે format તરીકે થાય છે, (જોકે કોઈપણ format, plain (સાદા) text સાથે) ટુકમાં તેનો ઉપયોગ સર્વર થી ડેટા ને લઇ જવા માટે થાય છે.
- Ajax માં સર્વર ને request મોકલતી વખતે, પાછો આપેલ ડેટા plain text/html અથવા XML document માં હોય શકે છે. કેટલાક special instructions, ajax તે સારી રીતે formed XML text file ને retrieve કરે છે અને XML object તરીકે તેને return કરે છે. Standard DOM methodsનો ઉપયોગ કરીને આ XML data ને easily જુદો કરવામાં આવશે.
- **Note:** **XML, HTML** અને **Ajax** ના ઉદાહરણ માટે (refer page નો 16)

### Dynamic Table

- ડેટાબેસ ટેબલ પર Ajax નો ઉપયોગ કરીને dynamic ટેબલ એ row ને automatic generate કરે છે.

- Sort ટેબલ ને implement કરી sort ટેબલ ની ability ને column દ્વારા allow કરવી એ common task છે અને તેનું solution એ છે કે આપણે સર્વર ને request મોકલી નવા પેજ માં sort ટેબલ include કરવી શકાય છે.

### Dynamic Table Features

- Column header પર ક્લિક કરીને અથવા sort ફંક્શન ને JavaScript દ્વારા call કરીને sort table columns બનાવાય છે.
- Large ટેબલ ને સરળ જોવા માટે scrollbar ઉમેરવામાં આવે છે.
- ઝડપી sorting speed આપે છે.
- Columns ને Show/hide કરી શકાય છે.
- ડેટા અથવા columns ને આપમેળે શોધી કાઢે છે અને તે મુજબ ગોઠવે છે.
- ટેબલ ના appearance માં interfere કરતું નથી.
- ટેબલ start કરતી વખતે shorting અથવા column hide કરીને individually રૂપે enabled કરેલું છે કે નહીં તે set કરે છે.

### Example:

#### Index.php

```

<html>
<head>
<script>

 function showUsers() {
 if (window.XMLHttpRequest) {
 xmlhttp = new XMLHttpRequest();
 } else {
 xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
 }
 xmlhttp.onreadystatechange = function() {
 if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
 document.getElementById("txtHint").innerHTML = xmlhttp.responseText;
 }
 }
 xmlhttp.open("GET","getusers.php",true);
 xmlhttp.send();
 }
</script>
</head>
<body>
 <div align="center">
 <button onclick="showUsers();">Fetch User Data</button>

 </div>
</body>

```

```

<div id="txtHint">Person info will be listed here...</div></div>
</body>
</html>
getusers.php
<html>
<head>
<style>
 table {
 width: 100%;
 border-collapse: collapse;
 }
 table, td, th {
 border: 1px solid black;
 padding: 5px;
 }
 th {text-align: left;}
</style>
</head>
<body>
<?php
 $con = mysqli_connect('localhost','root','','ajaxbind');
 if (!$con) {
 die('Could not connect: ' . mysqli_error($con));
 }
 mysqli_select_db($con,"ajaxbind");
 $sql="SELECT * FROM users";
 $result = mysqli_query($con,$sql);
 echo "<table><tr>
 <th>Firstname</th>
 <th>Address</th>
 <th>Age</th> </tr>";
 while($row = mysqli_fetch_array($result)) {
 echo "<tr>";
 echo "<td>" . $row['username']. "</td>";
 echo "<td>" . $row['address']. "</td>";
 echo "<td>" . $row['age']. "</td>";
 echo "</tr>";
 }
 echo "</table>";

```

```

mysqli_close($con);

?>
</body>
</html>

```

### Output:

Fetch User Data

Firstname	Address	Age
peter@gmail.com	USA	29
James@gmail.com	China	25
JOhnMedina@gmail.com	Spain	24

- ઉપરોક્ત કોડ માં જ્યારે આપણે "fetch user data button" પર ક્લિક કરીએ છીએ ત્યારે ઉપર નું ટેબલ script દ્વારા dynamically બને છે અને તે Ajax નો ઉપયોગ કરીને display કરે છે.

### Object Literals

- Object literals zero અથવા વધારે expression pairs નું લીસ્ટ છે જે **curly braces ({} )** ની અંદર હોય છે જ્યાં દરેક expression pairs એ property નું નામ અને તેને related valueને represent કરે છે.
- દરેક **number** ને **comma** સાથે **delimit** કરવામાં આવે છે દરેક numberની અંદર નામ અને value ને **colon(:)** દ્વારા જુદા કરવામાં આવે છે. નીચે object literal નું ઉદાહરણ બનાવેલ છે.

```

var user={

 "Address":"43, Street No.5 ",
 "City":"HongKong";
 "Zipcode":1110001
};

```

- આ objectની value ને access કરવા તમે "dot notation" નો ઉપયોગ કરી શકો છો એટલે કે, objectનું નામ, dot અને property નું નામ.

```
alert(user.Address);
```

### Array in Objects

- Objects અને Arrays નું combination એક complex, multidimensional object બનાવે છે. નીચેના ઉદાહરણમાં જેમાં object બનાવવામાં આવ્યું છે અને તેના member તરીકે array સાથે initialized કરેલું છે:

```

var person={

 "Name": "Peter Parker",
 "IsActive": true,
 "Contact": [
 {
 "Location": "Home",
 "Number": "999999999"
 }
]
};

```

```
{
 "Location": "Office",
 "Number": "123456789"
}
};
```

- ઉપરોક્ત કોડમાં, person object બનાવવામાં આવે છે અને બે string members અને array literal સાથે initialized થાય છે. Contact એ array member છે જે object ના members અને તેમની values ધરાવે છે.
- આ object ની values ને access કરવા માટે:

```
alert("Hello!" + person.Name + " we called at number" + person.Contact[0].Number);
```

### Array Literals

- Array literals માં દરેક expression array ના elements ને represent કરે છે. Array elements comma દ્વારા પુરા થાય છે અને square brackets ([ ]) માં આવેલા હોય છે. નીચેનું ઉદાહરણ ચાર string elements સાથે array literal define કરે છે.

```
var games=[

 "Cricket",

 "Hockey",

 "Football",

 "Tennis"

];
```

- Value ને access કરવા માટે, તમે numeric index નો ઉપયોગ કરો છો.

```
alert(games[0]); //Answer is Cricket.
```

### Objects in Array

- નીચે આપેલા ઉદાહરણ માં array literals બે object સાથે value ને define કરે છે.

```
var person=[

 {key1:'value1'},

 {key2:'value2'}];
```

- આ array ની value ને access કરવા માટે:

```
alert(person[0].key1);
```

### JSON Introduction or What is JSON?

- JSON એટલે Javascript Object Notation JSON formate મૂળરૂપે Douglas Crockford દ્વારા specify કરવામાં આવ્યું હતું.
- JSON એ lightweight data interchange format છે જે read અને write કરવામાં સરળ છે અને machine ને પાર્સ (parse) અને જનરેટ કરવા માટે પણ સરળ છે.

- તે **JavaScript scripting language** થી extended કરવામાં આવ્યું છે. JSON official internet media type **JSON application/json** છે.
- JSON ની file નું extension .json આપવામાં આવે છે.
- JSON એ independent language છે અને easily ઉપયોગ કરી શકાય તેવી XML ની alternative language છે.

### JSON Syntax

- JSON syntax એ JavaScript object notation syntax નો subset છે.
  - Data** એ name/value ની pair માં represent (રજૂ) થાય છે અને name/value ની pairs ',' (comma) દ્વારા અલગ કરવામાં આવે છે.
  - JSON ની **value: number** (integer અથવા floating point), **string** (double quotes માં), **Boolean** (true અથવા false), **array** (square brackets માં), **object** (curly braces માં), **null**, અને **Whitespace**.
  - Curly braces એ object એ array ને hold કરે છે અને દરેક નામ ':' (colon) દ્વારા follow કરવામાં આવે છે.
  - Square brackets એ array ને hold કરે છે.
- JSON એ બે data structure ને follow કરે છે.
  - Name/value ની pair નું collection:** આ data structure એ વિવિધ programming language દ્વારા supported છે.
  - Value નું ordered list:** તેમાં array, list, vector અથવા sequence વગેરે include હોય છે.
- નીચે આપેલ JSON ઉદાહરણ માં array ના 3 family menubars record સાથે family object define કરે છે.

```
{"family": [{
 "FirstName": "John",
 "LastName": "Smith"
,
 {
 "FirstName": "Pater",
 "LastName": "Parker"
 }
}]}
```

### JSON Advantages

- JSON એ **light weight component** છે. તેને execution માટે વધારે સમય લાગતો નથી.
- Smaller message ની size** javascript દ્વારા easily વપરાશમાં લેવાય છે.
- Modern language માં ઉપયોગમાં લેવાતા **data structure** ને support કરે છે.
- JSON એ **data oriented** છે. object oriented system પર JSON ને વધુ easily map કરી શકાય છે.

- JSON એ XML કરતા ધણું સરળ **format** છે.
- JSON માં ગમે ત્યાથી value retrieve કરી શકીએ છીએ તે બધી language માં support કરે છે.
- તે null value ને represent કરવા માટે સરળ છે.

### JSON Disadvantages

- Human ને JSON સમજવું **difficult** છે. કારણાંકે તેમાં ધણા braces અને comma વપરાય છે.
- JSON એ **large data** માટે યોગ્ય નથી.
- JSON માં text images વગેરે જેવી **multimedia formate** ને **handle** કરવાના **features** નથી.
- Applications વચ્ચે data exchange કરવા માટે compact data format છે.

### Ajax Applications

#### Login Form Example

- પહેલા **phpmyadmin** માં "ajaxbind" નામનો database create કરો અને users નામનું table create કરો.

users		
ColumnName	Datatype	Constrains
userid	int	Primary Key
username	varchar(100)	
password	Varchar(50)	

#### Index.php

```

<html>
<head>
 <title>Login</title>
 <script src="jquery-1.10.2.min.js" language="javascript"></script>
 <script>
 function loginuser()
 {
 var username=document.getElementById('uname').value;
 var password=document.getElementById('pass').value;
 if (window.XMLHttpRequest)
 xmlhttp=new XMLHttpRequest();
 else
 xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
 xmlhttp.onreadystatechange=function() {
 if(xmlhttp.readyState == 1)
 {
 document.getElementById("content").innerHTML=
 }
 }
 }
 </script>
</head>
<body>
 <form>
 <input type="text" id="uname" name="username" placeholder="Enter Username" />
 <input type="password" id="pass" name="password" placeholder="Enter Password" />
 <input type="button" value="Login" onclick="loginuser()" />
 </form>
 <div id="content"></div>
</body>

```

```

else if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
{
 if(xmlhttp.responseText=="success")
 {
 window.location.assign("data.php");
 }
 else
 {
 document.getElementById("content").innerHTML=xmlhttp.responseText;
 }
}
xmlhttp.open("GET","select.php?username="+username+"&password="+password,true);
xmlhttp.send();
}
</script>
</head>
<body>
</div>
<div align="center">
<fieldset style="width:25%;>
 <legend>Login</legend>
 <table border="0" align="center">
 <tr><td><label>Username:</label></td>
 <td><input type="text" value="" name="uname" id="uname" /></td>
 </tr>
 <tr>
 <td><label>Password:</label></td>
 <td><input type="password" value="" name="pass" id="pass" /></td>
 </tr>
 <tr>
 <td></td>
 <td><button onclick="loginuser();>Login</button></td>
 </tr>
 <tr><td></td>
 <td><div id="content"></div></td>
 </tr>
</table></fieldset></div>
</body>

```

```
</html>
select.php
<?php
 session_start();
 $con=mysqli_connect("localhost","root","","ajaxbind");
 if(!$con)
 {
 echo "Error in connection"; }
 mysqli_select_db($con,"ajaxbind");
 $un=$_GET['username'];
 $pas=intval($_GET['password']);
 $query="select *from users where username='$un' AND password='$pas'";
 $result=mysqli_query($con,$query);
 $count=mysqli_num_rows($result);
 $row=mysqli_fetch_assoc($result);
 if($count>0)
 {
 $_SESSION['uname']=$row['username'];
 if(isset($_SESSION['uname']))
 {
 echo "success";
 }
 }
 else
 {
 echo "Invalid Username or Password!";
 }
?>
```

### **data.php**

```
<html>
<head><title>Logout</title>
<body>
 <p><?php session_start(); echo $_SESSION['uname']; ?>Log Out
</body>
</html>
```

### **logout.php**

```
<?php
 session_start();
 session_destroy();
 header('Location:index.php');
?>
```

### Preloaded Data Example

#### Index.php

```

<html>
<head>
<title>Load Data</title>
<script src="jquery-1.10.2.min.js" language="javascript"></script>
<script>
$(document).ready(function(){
 if (window.XMLHttpRequest)
 xmlhttp=new XMLHttpRequest();
 else
 xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
 xmlhttp.onreadystatechange=function() {
 if(xmlhttp.readyState == 1)
 {
 document.getElementById("content").innerHTML=
 }
 else if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
 document.getElementById("content").innerHTML=xmlhttp.responseText;
 }
 xmlhttp.open("GET","test.html",true);
 xmlhttp.send();
});
</script>
</head>
<body>
 <div id="content">
 </div>
</body>
</html>

```

#### test.html

```

<html>
<head>
 <title>Original Data</title>
</head>
<body>
 <p align="justify">

```

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. </p>

```

</body>
</html>
```

### Example of feedback from using validation

#### Index.php

```

<html>
<head>
 <title>Ajax-HTML Registration Form</title>
 <link rel="stylesheet" type="text/css" href="css/style.css">
 <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.min.js"></script>
 <script type='text/javascript' src="js/validation.js"></script>
</head>
<body>
<div id='container'>
 <form name="contactForm" id='reg_form' method="post" action='email.php'>
 <h2>Feedback Form Validation using Ajax</h2>
 <p> Your Name: <div id='name_error' class='error'>Please enter your name.</div>
 <div><input type='text' name='name' id='name'> </div></p>
 <p> Your E-mail ID: <div id='email_error' class='error'>Please enter your valid E-mailID.</div>
 <div> <input type='text' name='email' id='email'><div></p>
 <p> Your Message:<div id='message_error' class='error'>Please enter your message.</div> <div>
 <textarea name='message' id='message'></textarea></div> </p>
 <div id='mail_success' class='success'>Mail Send successfully.</div>
 <div id='mail_fail' class='error'> Sorry, Mail not Sent.</div>
 <p id='submit'><input type='submit' id='send_message' value='Submit Form'></p>
 </form></div>
</body>
</html>
```

#### validate.js

```

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

```

$('#send_message').click(function(e){
 e.preventDefault();
 var error = false;
 var name = $('#name').val();
 var email = $('#email').val();
 var message = $('#message').val();

 // Form field validation
 if(name.length == 0){
 var error = true;
 $('#name_error').fadeIn(500);
 }else{
 $('#name_error').fadeOut(500);
 }
 if(email.length == 0 || email.indexOf('@') == '-1'){
 var error = true;
 $('#email_error').fadeIn(500);
 }else{
 $('#email_error').fadeOut(500);
 }
 if(message.length == 0){
 var error = true;
 $('#message_error').fadeIn(500);
 }else{
 $('#message_error').fadeOut(500);
 }
 if(error == false){
 $('#send_message').attr({'disabled' : 'true', 'value' : 'Sending...' });
 $.post("email.php", $("#reg_form").serialize(),function(result){
 if(result == 'Success'){
 $('#submit').remove();
 $('#mail_success').fadeIn(500);
 }else{
 $('#mail_fail').fadeIn(500);
 $('#send_message').removeAttr('disabled').attr('value', 'Submit');
 }
 });
 }
});

```

});

### email.php

```
<?php
 $subject = $_REQUEST['subject'] . ' Ajax HTML Contact Form : Demo'; // Subject of your email
 $to = $_REQUEST['email']; //Recipient's E-mail
 $headers = 'MIME-Version: 1.0' . "\r\n";
 $headers .= "From: " . $_REQUEST['email'] . "\r\n"; // Sender's E-mail
 $headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
 $message .= 'Name: ' . $_REQUEST['name'] . "
";
 $message .= $_REQUEST['message'];
 if (@mail($to, $subject, $message, $headers))
 {
 echo 'Success';
 }
 else
 {
 echo 'failed';
 }
?>
```

### Live Search Example (Use of HTML and Xml in Ajax)

#### Index.php

```
<html>
<head>
<title>Live Search using XML,PHP and AJAX</title>
<script>
 function showResult(str)
 {
 if (str.length==0)
 {
 document.getElementById("livesearch").innerHTML="";
 return;
 }
 if (window.XMLHttpRequest)
 {
 xmlhttp=new XMLHttpRequest(); // code for IE7+, Firefox, Chrome, Opera, Safari
 }
 else
```

```

 {
 xmlhttp=new ActiveXObject("Microsoft.XMLHTTP"); // code for IE6, IE5
 }
 xmlhttp.onreadystatechange=function()
 {
 if (xmlhttp.readyState==4 && xmlhttp.status==200)
 {
 document.getElementById("livesearch").innerHTML=xmlhttp.responseText;
 }
 }
 xmlhttp.open("GET","Search.php?q="+str,true);
 xmlhttp.send();
}
</script>
</head>
<body>
<form>
 <label>Search by Keyword:</label>

 <input type="text" size="30px" onkeyup="showResult(this.value)">
 <div id="livesearch">
</form>
</body>
</html>

```

### Search.php

```

<?php
 $data=$_GET["q"];
 $keyword=strtoupper($data);
 $xml=simplexml_load_file("Data.xml") or die("Error: Cannot create object");
 $length=count($xml);
 for($i=0;$i<$length;$i++)
 {
 $title=(string)$xml->item[$i]->title; // Title Post
 $link=(string)$xml->item[$i]->link; // Url Link
 $strpos = stripos($title, $keyword);
 if($strpos !== FALSE) {
 echo "<a href=\"".$link;
 echo "">";
 echo $title;
 }
 }
}

```

```

 echo "
";
}
}

?>

```

### Data.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<channel>
 <item>
 <title>Online Website Design Tutorials</title>
 <link>http://www.google.com</link>
 </item>
 <item>
 <title>Create Flexible box model using CSS3</title>
 <link>http://www.cssdemo.com</link>
 </item>
 <item>
 <title>How To Search data using XML,PHP and Ajax</title>
 <link>http://www.ajaxdemo.com</link>
 </item>
 <item>
 <title>How to delete record in PHP using PHP-MySQL</title>
 <link>http://www.phpdemo.com</link>
 </item>
</channel>

```

### Output:

Search by Keyword:

[Online Website Design Tutorials](#)

### Dependable Dropdown using Ajax-Country, State and City Examples.

- सौ प्रथम Phpmyadmin में "ajaxbindcountry" नामको database create करो और नीचे उल्लिखित कोड को बनावें:

loc_country		
ColumnName	Datatype	Constrains
CountryID	int	Primary Key
CountryName	varchar(100)	

loc_state		
ColumnName	Datatype	Constrains
StateID	int	Primary Key
CountryID	int	Foreign Key
StateName	varchar(100)	

loc_city		
ColumnName	Datatype	Constrains
CityID	int	Primary Key
CountryID	int	Foreign Key
StateID	int	Foreign Key
StateName	varchar(100)	

### connection.php

```
<?php
 $Conn=mysqli_connect("localhost", "root","","ajaxbindcountry") or die('Cannot connect to the
database because: '.mysqli_error($Conn));
 if(!mysqli_select_db($Conn,"ajaxbindcountry"))
 {
 die('Can not connect to the Database'.mysqli_error($Conn));
 }
?>
```

### data.php

```
<?php
include_once 'connection.php';
if(isset($_GET['CountryID']))
{
 $id=$_GET['CountryID'];
 $str1="select * from loc_state where CountryID=$id";
 $StateIDResult=mysqli_query($Conn,$str1);
 $DropDownControl='<option value="">Select Any...</option>';
 if(isset($StateIDResult) and $StateIDResult!=null)
 {
 if(mysqli_num_rows($StateIDResult)>0)
 {
 while($StateRow=mysqli_fetch_assoc($StateIDResult))
 {
 $DropDownControl.= "<option
value='".$StateRow['StateID']."'>".$StateRow['StateName']. "</option>";
 }
 echo $DropDownControl;
 }
 }
}
```

```

}

if(isset($_GET['StateID']))
{
 $id=$_GET['StateID'];
 $str1="select * from loc_city where StateID=$id";
 $CityIDResult=mysqli_query($Conn,$str1);
 $DropDownControl='<option value="">Select Any...</option>';
 if(isset($CityIDResult) and $CityIDResult!=null)
 {
 if(mysqli_num_rows($CityIDResult)>0)
 {
 while($CityRow=mysqli_fetch_assoc($CityIDResult))
 {
 $DropDownControl.= "<option
value='".$CityRow['CityID']."'".">".$CityRow['CityName']. "</option>";
 }
 echo $DropDownControl;
 }
 }
?>

index.php
<?php
 include_once 'connection.php';
?>
<html>
<head>
<title>Dependable Dropdown using Ajax-Country, State and City</title>
<script>
function SelectStateByCountryID(CountryID)
{
 if (window.XMLHttpRequest)
 xmlhttp=new XMLHttpRequest();
 else
 xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
 xmlhttp.onreadystatechange=function()
 {
 if (xmlhttp.readyState == 4 && xmlhttp.status == 200)

```

```

document.getElementById("ddlStateID").innerHTML =
 xmlhttp.responseText;
}

xmlhttp.open("GET","data.php?CountryID="+CountryID,true);
xmlhttp.send();
}

function SelectCityByStateID(StateID)
{
 if (window.XMLHttpRequest)
 xmlhttp=new XMLHttpRequest();
 else
 xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
 xmlhttp.onreadystatechange=function()
 {
 if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
 document.getElementById("ddlCityID").innerHTML =
 xmlhttp.responseText;
 }
 xmlhttp.open("GET","data.php?StateID="+StateID,true);
 xmlhttp.send();
}

</script>
</head>
<body>
<form>

 <label>Country</label>
 <select name='ddlCountryID' id='ddlCountryID'
 onchange="SelectStateByCountryID(this.value);">
 <option value="">Select Any...</option>
 <?php
 $str="select * from loc_country";
 $CountryIDResult=mysqli_query($Conn,$str);
 if(isset($CountryIDResult) and $CountryIDResult!=null)
 {
 if(mysqli_num_rows($CountryIDResult)>0)
 {
 while($CountryRow=mysqli_fetch_assoc($CountryIDResult))
 {
 <option value=$CountryRow[CountryID]>$CountryRow[CountryName]</option>
 }
 }
 }
 </?php>

```

```

echo "<option
value='".$CountryRow['CountryID']."'>".$CountryRow['CountryName']."</option>";
}
}

?>
</select>
<label>State</label>
<select name='ddlStateID' id='ddlStateID' onchange="SelectCityByStateID(this.value);">
<option value="">Select Any...</option>
</select>
<label>City</label>
<select name='ddlCityID' id='ddlCityID'>
<option value="">Select Any...</option>
</select>
</form>
</body>
</html>

```

### JQuery in Ajax

- jQuery AJAX functionality माटे धण्डी methods provide करे छे.
- jQuery AJAX methods साथै, तमे **HTTP Get** अने **HTTP Post** बंगेनो उपयोग करीने remote server मांथी text, HTML, XML अथवा JSON ने request करी शको छो अने तमे **external data** ने सीधा ज तमारी web page ना select करेला HTML element मां load करी शको छो.
- jQuery ए \$ajax() नामनी primary method आपे छे, जेनो उपयोग **AJAX (asynchronous HTTP)**ने **request** करवा माटे थाय छे. बधी jQuery ए AJAX methodsनो उपयोग करे छे. आ method मोटाभागे request माटे वपराय छे ज्यां बीजु methods नो उपयोग करी शकातो नथी.

#### Syntax:

```
$.ajax({name:value, name:value, ... })
```

#### Example:

```

("button").click(function(){
 $.ajax({url: "MyData.txt", success: function(result){
 $("#ShowMyData").html(result);
 }});
});
```

- उपरोक्त उदाहरण AJAX request नो उपयोग करीने <div> element ना text बदले छे.

- Ajax સાથે કામ કરવા માટે સૌથી મહત્વપૂર્ણ jQuery shorthand methods **load()**, **\$post()**, and **\$get()** છે.

### JQuery-AJAX load() Method

- Load () method** સર્વરમાંથી data load કરે છે અને પાછું ડેટા select કરેલ એલેમેન્ટમાં મૂકે છે.

**Syntax:**

```
$(selector).load(URL,data,callback);
```

➤ **Required URL parameter** તમે જો URL ને load કરવા માંગો છો તે specifies કરે છે.

➤ **optional data parameter** request સાથે મોકલવા માટે query string key/value pairs નો set specifies કરે છે.

➤ **optional callback parameter** load() method complete થયા પછી ફંક્શનનું નામ છે. callback function માં વિવિધ parameters હોઈ શકે છે: **responseTxt** - જો call success થાય છે, resulting content શામેલ હોય, તો **statusTxt** - call નું status અને **xhr** - XMLHttpRequest object શામેલ હોય છે.

**Example:**

```
 $("button").click(function(){
 $("#MyData").load("MyIntro.txt", function(responseTxt, statusTxt, xhr){
 if(statusTxt == "success")
 alert("External content loaded successfully!");
 if(statusTxt == "error")
 alert("Error: " + xhr.status + ": " + xhr.statusText);
 });
});
```

### JQuery-AJAX get() and post() Methods

- HTTP **get()** અને **post()** methods નો ઉપયોગ HTTP GET અથવા POST request સાથે server થી dataને request મોકલવા માટે થાય છે.

- Client અને server વચ્ચેની request-response માટે બે સામાન્ય રીતે ઉપયોગમાં લેવાતી methods છે:  
**GET અને POST**

➤ **GET:** Specified resource માંથી dataને requests કરે છે અને તે મૂળસૂપે server માંથી કેટલાક data મેળવવા (retrieve કરવા) માટે ઉપયોગમાં લેવાય છે.

➤ **POST:** Specified resource પર process કરવા માટે ડેટા submit કરે છે અને તેનો ઉપયોગ server થી કેટલાક ડેટા મેળવવા માટે પણ થઈ શકે છે. જો કે, POST method data ને caches કરે નહીં અને ઘણીવાર request સાથે ડેટા મોકલવા માટે વપરાય છે.

### JQuery \$.get() Method

- \$.get () method** HTTP GET request નો ઉપયોગ કરીને server થી data ને request કરે છે.

### Syntax:

```
$.get(url, [data], [callback], [type]);
```

### Example:

#### Index.html

```
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("button").click(function(){
 $.get("Data.php", function(data, status){
 alert("Data: " + data + "\nStatus: " + status);
 });
 });
});
</script>
</head>
<body>
 <button>Click Here to see the result</button>
</body>
</html>
```

#### Data.php

```
<?php
echo " The $.get() method requests data from the server with an HTTP GET
request";
?>
```

- નીચેના ઉદાહરણાં server પર ફાઇલમાંથી ડેટા retrieve કરવા માટે **\$.get() method** નો ઉપયોગ કરે છે.
- **\$.get()** નો first parameter તે URL છે જેને ("Data.php") માં request કરીએ છીએ. second parameter callback ફૂકશાન છે.
- પહેલો callback parameter requested કરેલા પેજના content ને ધરાવે છે, અને બીજા callback parameter request નું status ધરાવે છે.

### jQuery **\$.post()** Method

- **\$.post() method** HTTP POST request નો ઉપયોગ કરીને server થી ડેટાને request કરે છે.

### Syntax:

```
$.post(url, [data], [callback], [type]);
```

### Example:

#### Index.html

```
$(document).ready(function(){
 $("button").click(function(){
```

```
$.post("Data.php",
{
 name: "Peter Parker",
 city: "London"
}, function(data,status){
 alert("Data: " + data);
});
});
});
```

### Data.php

```
<?php
$Name=$_POST['name'];
$City=$_POST['city'];
echo "My Name is ".$Name.". ";
echo "I'm from ".$City.". ";
?>
```

- \$.post () નો first parameter એ URL છે જેની ("Data.php") ને request કરીએ છીએ. પછી request (name અને city) સાથે મોકલવા માટે કેટલાક data pass કરીએ છીએ. Third parameter callback function છે.