



HTML



In this module you will learn

- Introduction to web technology
- Introduction to HTML5
- HTML5 elements
- Semantic Elements
- Table
- List
- Working with Links
- Image Handling

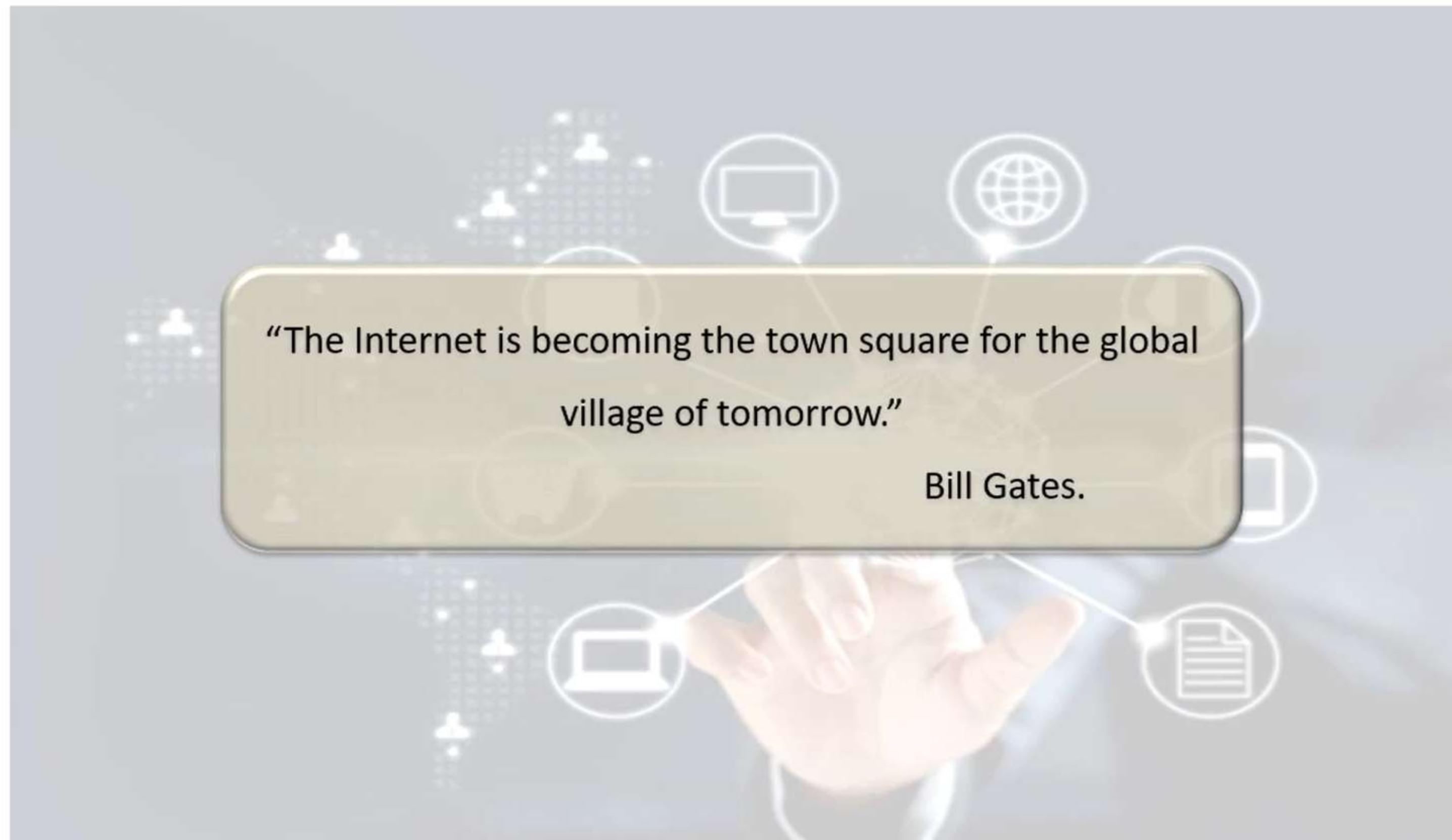


Introduction to Web Technology

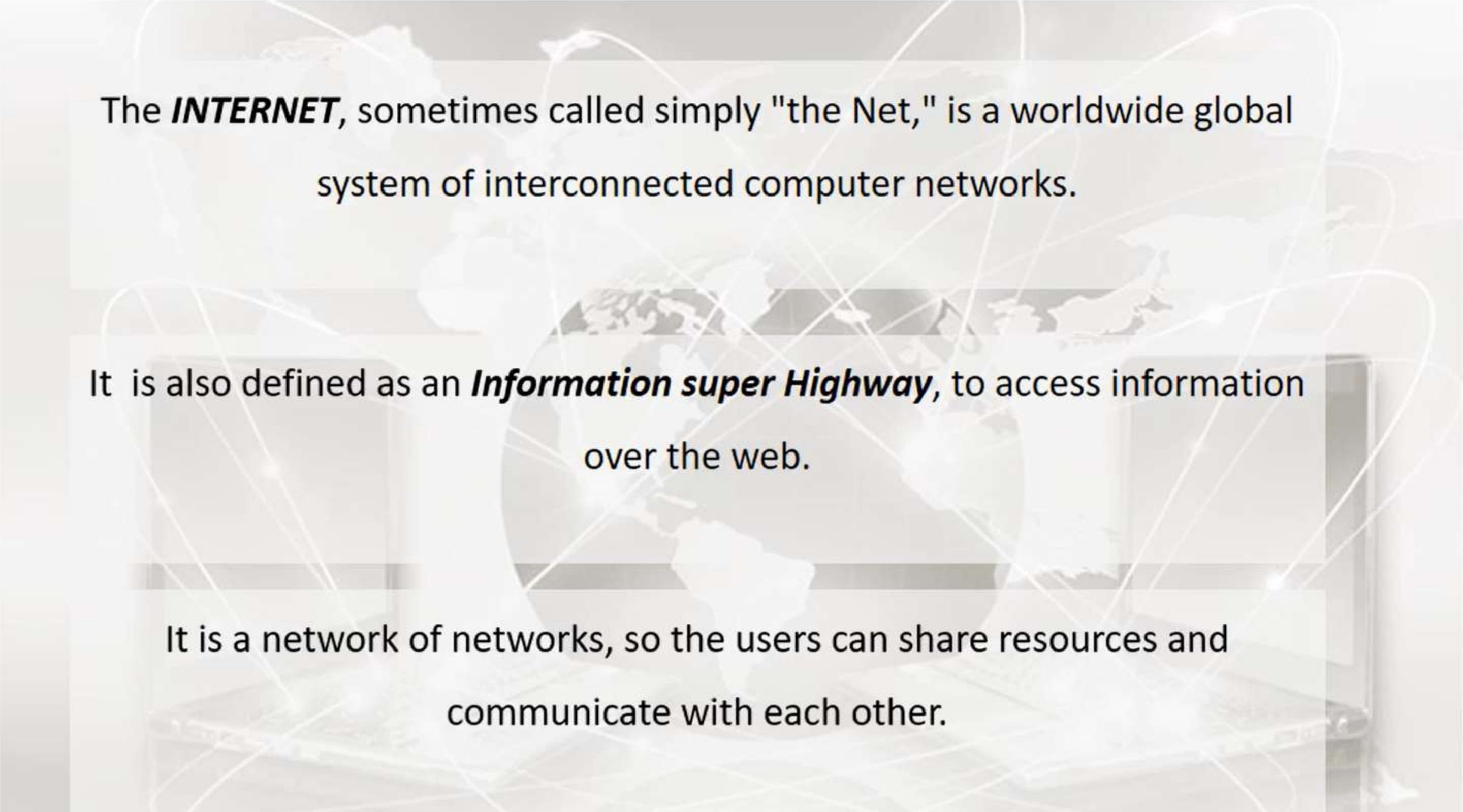


“The Internet is becoming the town square for the global
village of tomorrow.”

Bill Gates.



Web Technology



The ***INTERNET***, sometimes called simply "the Net," is a worldwide global system of interconnected computer networks.

It is also defined as an ***Information super Highway***, to access information over the web.

It is a network of networks, so the users can share resources and communicate with each other.

Terminologies

HTML – It is said to be **Hyper Text Markup Language** for creating web pages and web applications and it called as text formatting language.

Web Pages - A Web Page is a single html document displayed as a single page in a browser and it can be connected to other pages.

Web Sites - A Web site is a collection of several web pages all connected together that may contain text, images, audio and video. and usually at the same internet address. The first page of a website is called home page.

Static Web Page

A static web page Static website is the basic type of website whose content will be static



It is the one that is usually written in plain HTML



Prebuilt content is same every time the page is loaded because it sends exactly the same response for every request

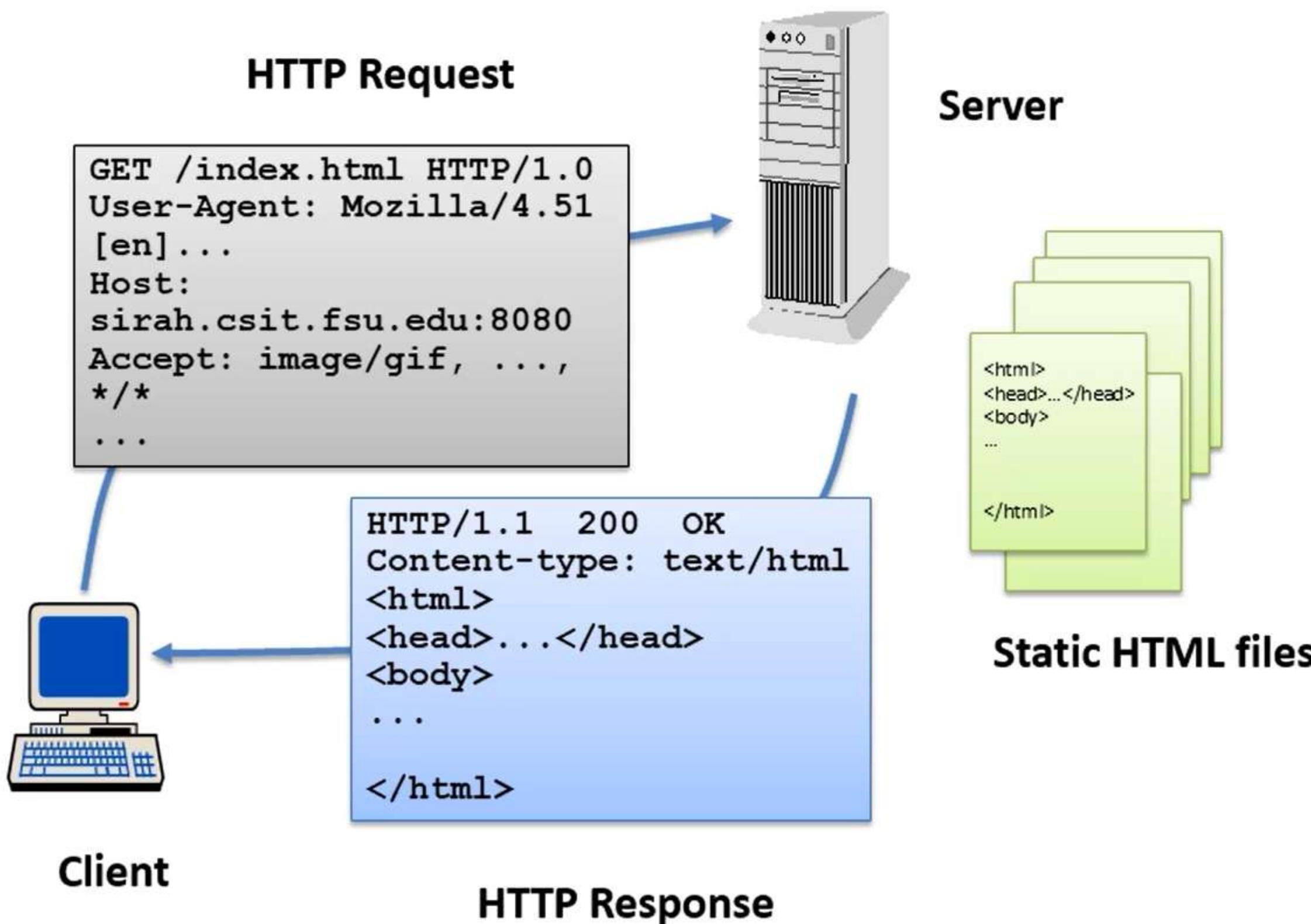


The content is only changes when someone publishes and updates the file (sends it to the web server)



Flexibility is the main advantage of static website

Requesting a Static HTML Document



Dynamic Web Page

- **Client side scripting** generates content at the client computer on the basis of user input. The web browser receives the web page from the server and processes the code within the page to render information to the user.
- In **server side scripting**, the software runs on the server, processes the code and once completed, the pages are sent to the client.

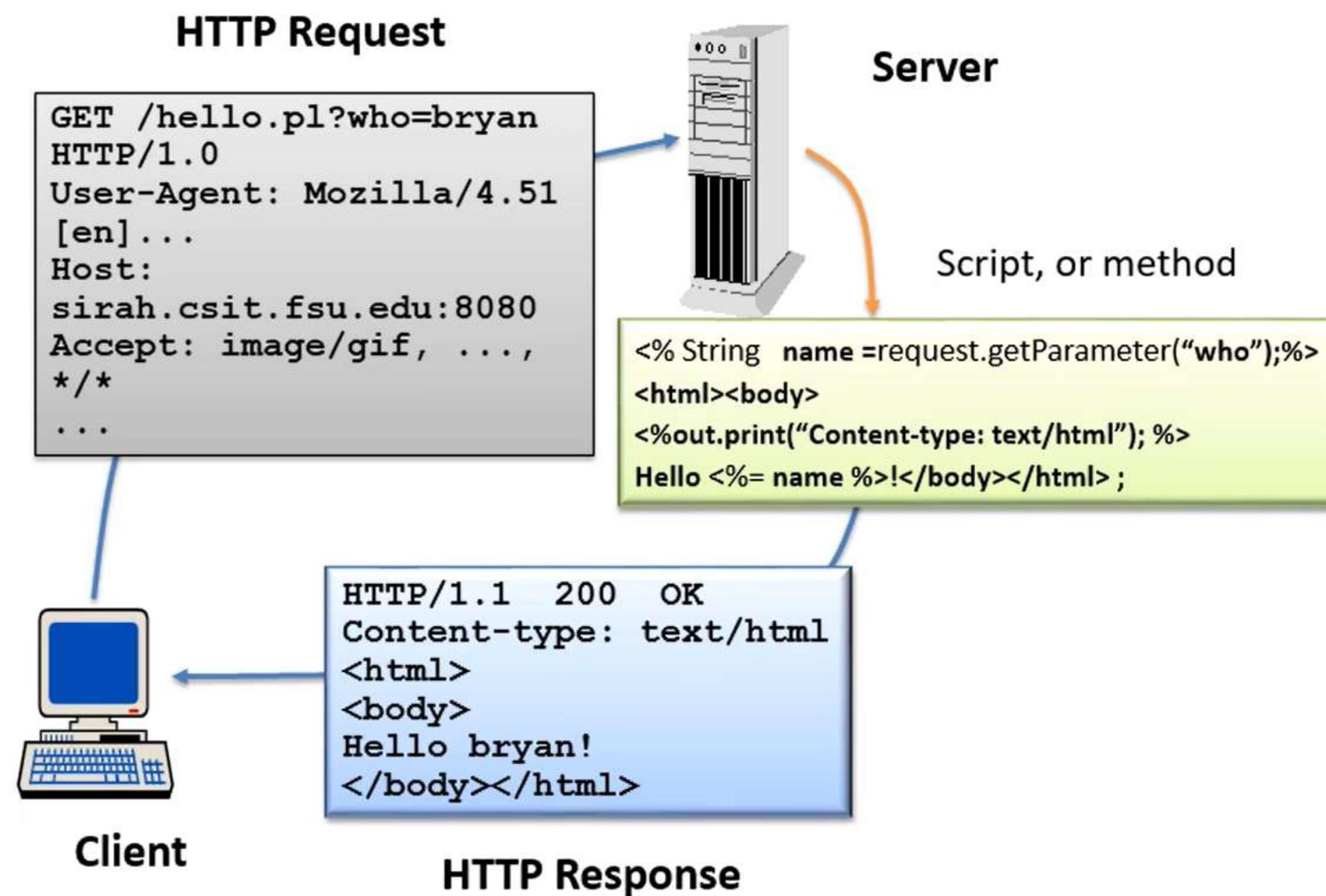
Dynamic website uses client-side scripting or server-side scripting, or both to generate dynamic content.

Dynamic web page is a page whose content changes dynamically. It shows different information at different points of time.

It may generate different HTML for each of the requests.

It accesses content from a database or **Content Management System (CMS)**. When you alter or update the content of the database, the content of the website is also altered or updated.

Dynamic Generation of HTML



Introduction to HTML

HTML stands for **Hyper Text Markup Language**.

Hyper Text - Refers to the way web pages are linked to each other

Markup - Usage of tags to structure the web page

HTML is used for developing web pages for applications. Web pages are text files containing HTML.

Normal text” surrounded by bracketed *tags* that tell browsers how to display web pages

Pages end with “.htm” or “.html”

HTML Editor – A word processor that has been specialized to make the writing of HTML documents more effortless

HTML Structure

HTML is comprised of “**elements**” and “**tags**”. Begins with **<html>** and ends with **</html>**

Elements (tags) are nested one inside another.

HTML describes structure using two main sections: **<head>** and **<body>**



Basic HTML Tags

<html>

<head>

<title>

<body>

<h1> -- <h6>

<p>

<hr>

<!-- -->

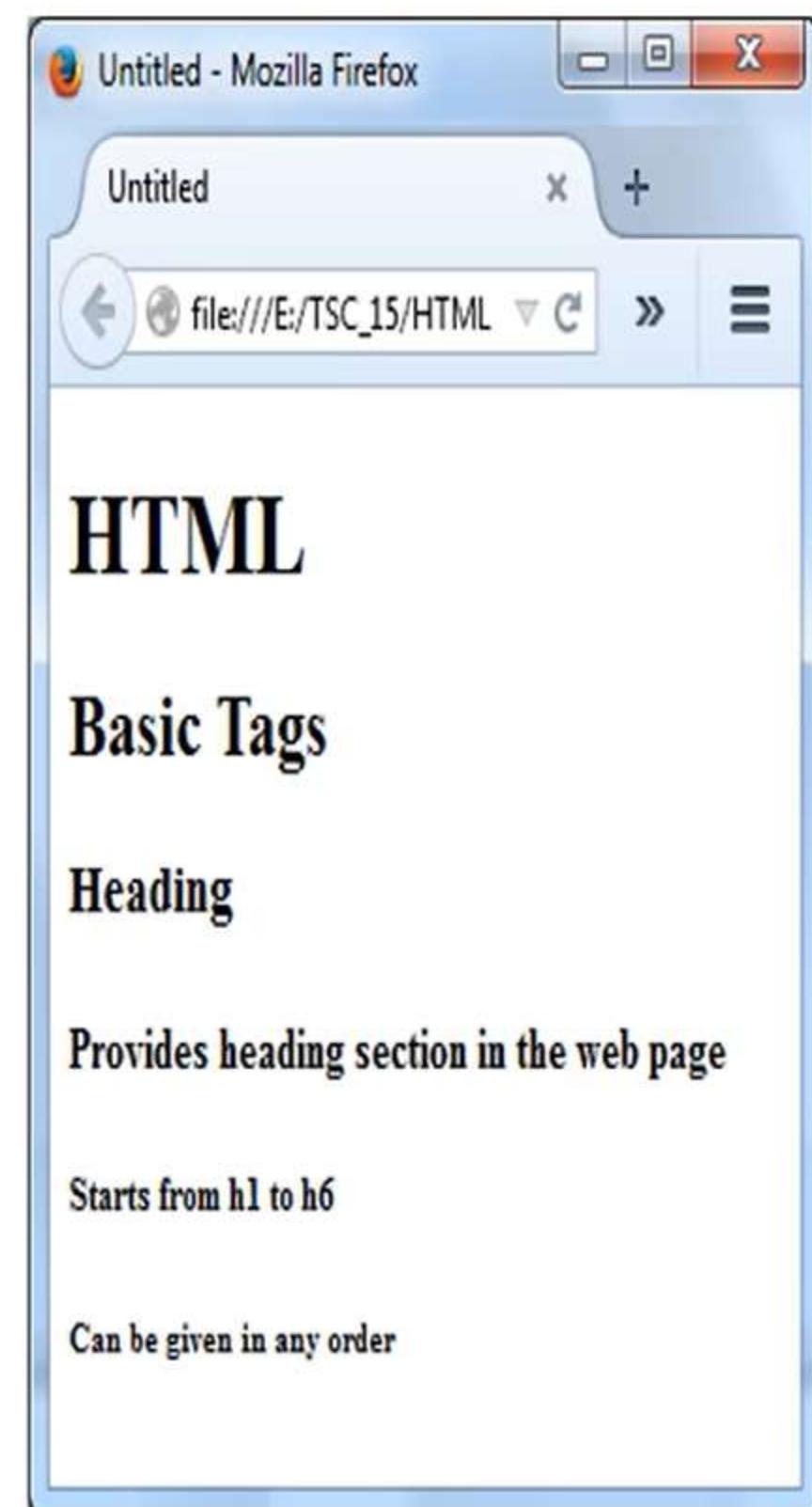
Basic HTML Tags

Heading Tag

Used to provide headings in HTML file

Heading tags start from **<h1>** to **<h6>**

```
<html>
<body>
    <h1>HTML</h1>
    <h2>Basic Tags</h2>
    <h3>Heading</h3>
    <h4>Provides heading section in the web
        page</h4>
    <h5>Starts from h1 to h6</h5>
    <h6>Can be given in any order</h6>
</body>
```



Basic HTML Tags cont ...

Paragraph tag

The **< p >** tag defines a paragraph.

Browsers automatically add **space** before and after each **< p >** element.

Break tag

<body>

< p > < h2 > Learn HTML < /h2 > < /p >

< p > The paragraph tags are used to define a **< br > block of text as a paragraph < /p >**

< /body >



Basic HTML Tags cont ...

Horizontal Rule

The <HR> element causes the browser to display a horizontal line (rule) in your document.

```
<html>
<body>
    <p><h2>Learn HTML</h2> </p>
    <p>The paragraph tags are used to define a
        <br>block of text as a paragraph</p>
    <hr>
    <p>The tag HR gives a horizontal line </p>
</body>
</html>
```



Basic HTML Tags cont ...

<!DOCTYPE>

- Gives instruction to the web browser about the version of HTML used
- Doctype must be the first line of the web page

```
<!DOCTYPE html>
<html>
<body>
    Contents of Web Page
    </body>
</html>
```

Represents
HTML5

Basic HTML Tags cont ...

Comments

- Tag used to add information to the html page, which will not be displayed in the browser
- Tag : <!-- -->

```
<!DOCTYPE html>
<html>
<body>
<!-- body tag contains the details that is to be displayed in the web page -->
    Contents of Web Page
    </body>
</html>
```

Will not be
displayed in
browser

HTML5

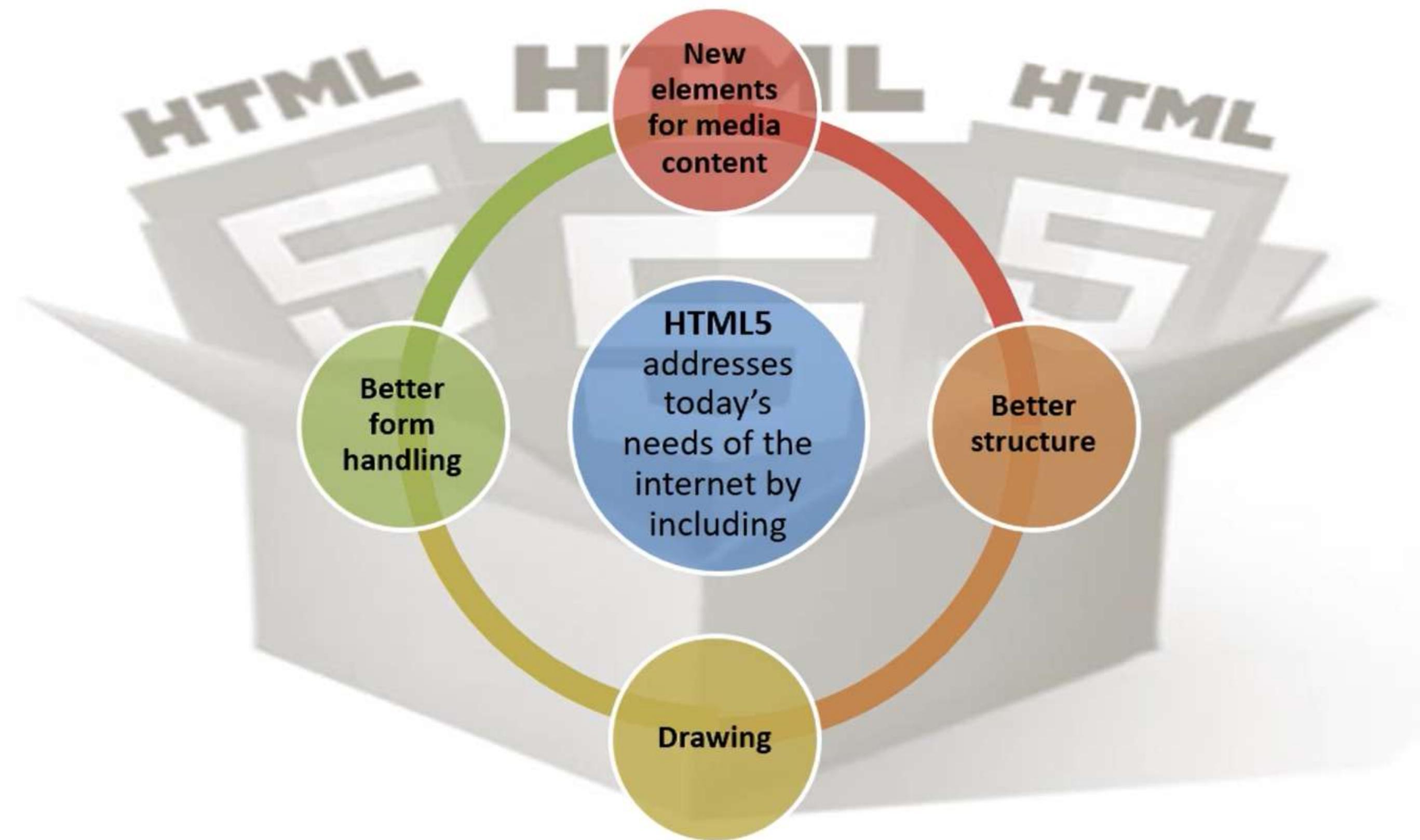


A bit of HTML

A whole sprinkling of JavaScript

A dash of CSS

HTML5 cont ...



HTML5 cont ...

Browser Support

The latest versions of

- Apple Safari
- Google Chrome
- Mozilla Firefox
- Opera

All support many
HTML5 features



The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

New Features

Better support for local offline storage

New form controls like: calendar, date, time, email, url, search, etc.

The canvas element for sketching

The video and audio elements for media playback

- Allows video and audio to be tagged easier
- (like images)... such as
- <video src=...> and <audio src=...>

New content specific elements such as

- article, footer, header, navigation, section, etc.

Forms 2.0 and client-side validation

Native browser support for audio and video

Tables

```

<table border="1">
    <colgroup>
        <col span="2" style="background-color:#81D5D9">
        <col style="background-color: #DDDB5F">
    </colgroup>
    <tr> <caption>Employee Details</caption> </tr>
    <thead>
        <tr>
            <th>ID</th>
            <th>NAME</th> <th>AGE</th> <th>Address</th></tr>
    </thead>
    <tbody>
        <tr> <td>101</td> <td>Johan</td> <td>20</td> <td rowspan="2" style="background-color:#81D5D9">Ganapathy</td>
        </tr>
        <tr> <td>102</td> <td>Mini</td> <td>19</td> </tr>
        <tr> <td>103</td> <td>Ivan</td> <td>23</td> <td>Coimbatore</td> </tr>
    </tbody>
    <tfoot>
        <tr>
            <td colspan="3" style="background-color:#81D5D9">The total Employees</td> <td> 3 </td>
        </tr>
    </tfoot>
</table>

```

Employee Details

ID	NAME	AGE	Address
101	Johan	20	Ganapathy
102	Mini	19	
103	Ivan	23	Coimbatore
The total Employees			3

HTML List Tags



Creates an ordered or unordered list for the contents of the web page

Ordered list

- Tag for defining the list item :
- Attribute : type
 - 1 -- numbered with number
 - A -- numbered with Capital case alphabets
 - a -- numbered with smaller case alphabets
 - I -- numbered with upper case roman letters
 - i -- numbered with lower case roman letters

Example for List Tags

```
<html>
<body>
    basic Tags
    <ol type="1">
        <li>paragraph</li>
        <li>heading</li>
        <li>line break</li>
        <li>horizontal break</li>
    </ol>
    formatting tags
    <ul style="list-style-type: square">
        <li>bold</li>
        <li>strong</li>
        <li>del</li>
        <li>code</li>
    </ul>
</body>
</html>
```

basic Tags

1. paragraph
2. heading
3. line break
4. horizontal break

formatting tags

- bold
- strong
- del
- code

Unordered list :

- list-style-type:disc -bullets (default)
- list-style-type:circle- circles
- list-style-type:square-squares
- list-style-type:none – nothing will appear

Links



Hyperlink, which helps in navigation from one page to another web page or from one part of the page to the other part of the web page

`<a>` - anchor tag

- The hyperlink can be a text or image which is clickable

Attributes

- href: Defines the destination location
- Target :Defines where the targeted page must be opened

```
<body>
<a href="paragraph.html">Click Here</a>
</body>
```

Links – Target Attribute

Target Value	Description
_blank	Opens the linked page in a new window or tab
_self	Opens the linked page in the same frame as it was clicked (this is default)
_parent	Opens the linked page in the parent frame
_top	Opens the linked page in the full body of the window
framename	Opens the linked page in a named frame

Internal links

- Links can also be created inside large documents to simplify navigation.

HTML5 Image Tag

- Defines an image in HTML page
- Attributes
 - src – URL of the image
 - alt -- alternate text that is displayed, if the image is not displayed in the webpage
 - height – height of the image
 - width – width of the image

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

Image map is a map with clickable area
The **<map>** tag is used to define a client-side image-map.



Semantic Tags

A **semantic element** clearly describes its meaning to both the browser and the developer.

semantic elements: <form>, <table>, and <article>

The <div> tag defines a division or a section in an HTML document.

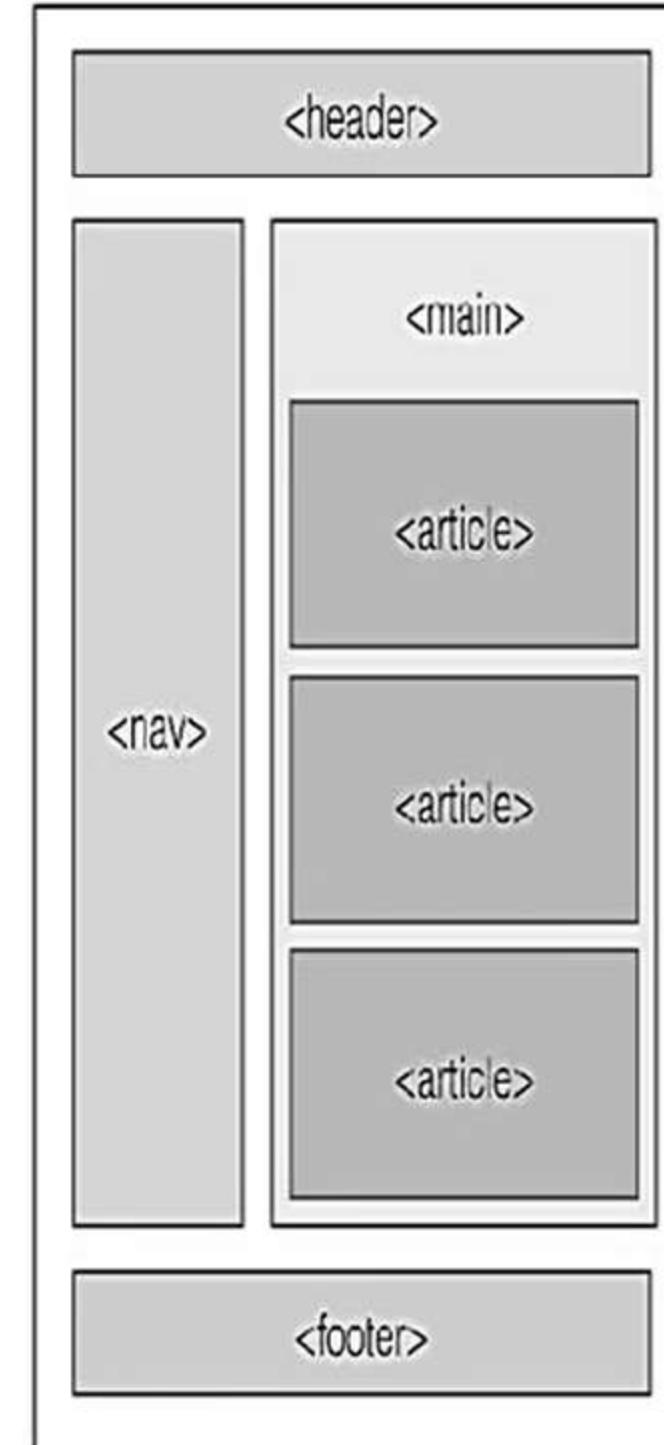
non-semantic elements: <div> and

HTML5 offers a set of new tags that provide the ability to mark up the sections of a document more descriptively than you could in HTML.

HTML 4.01



HTML 5



Why use Semantic tags for a web page?



A semantically meaningful tag is way more powerful than a generic one

The browser can know which area of your site is the header or the footer.

The semantic elements are also used by search engines.

- It is easy to imagine Messrs Google or Yahoo! giving lower weighting to content in footer elements, or extra weight to content in the header.

Site navigable for people with disabilities.

- People with learning difficulties might instruct their browser to always put the articles before the navigation

Useful for code maintenance

- (no more <!-- end: #content -->)

New Semantic Elements

It groups links to other pages or to parts of the current page.

It defines a section that is tangentially related to the content around it.

`<article>`

Specifies a header for a document or section

`<header> <h1>My website rocks !!!</h1> </header>`

`<div id="content">`

`<article>`

`<header> <h2>This is the greatest page ever</h2> </header>`

Content of Article

`</article>`

`<sidebar>`

` Home Page`
`Update Profile `
`Feedback `

`</nav>`

`</nav>`

`</sidebar>`

`<aside> </aside>`

`</div>`

`<footer>`

Copyright 2012 Teknoturf Info Services Pvt., Ltd., </footer>

`</article>`

It is referred to the footer of a web page.

It represents a self-contained composition in a document, page, etc.

Summary

- Introduction to web technology
- Introduction to HTML5
- HTML5 elements
- Semantic Elements
- Table
- List
- Working with Links
- Image Handling





HTML FORMS



In this module you will learn



- Introduction to HTML Forms
- Form-Input Elements
- HTML5 Form elements
- HTML5 Attributes
- Video Tag
- Audio Tag



HTML Forms



HTML forms enable web applications to collect information from users.

Used to interact between a user and a web site or an application.

To build a form, the following HTML elements are used

- <form>
- <label>
- <input>
- <textarea>
- <button>

Form – Elements



Example

HTML forms always start with a **<form>** tag

Form tag has the following attributes

```
<form name="registration" action="registered.html" method="post"  
      autocomplete="on" novalidate="novalidate">  
  </form>
```

Attribute	Description
Action	Specifies the destination on submission
Method	Specifies the HTML method to be used on sending the form Get : the data is sent along the URL Post : the data is sent via the message body
autocomplete	Specifies whether a form should have autocomplete on or off
novalidate	Specifies that the form should not be validated when submitted

Form-Input Elements



Form element contains

- <input>
- <textarea>
- <button>
- <select>
- <option>
- <optgroup>
- <fieldset>
- <label>

PERSONAL INFO

Email Address:

Password:

Gender:

Date of Birth:

PREFERENCES

Favorite Color: Blue Red Green

Interests: News Sports Entertainment Automotive

Input - Attributes



Type attribute is used to specify the type of the **<input>** element

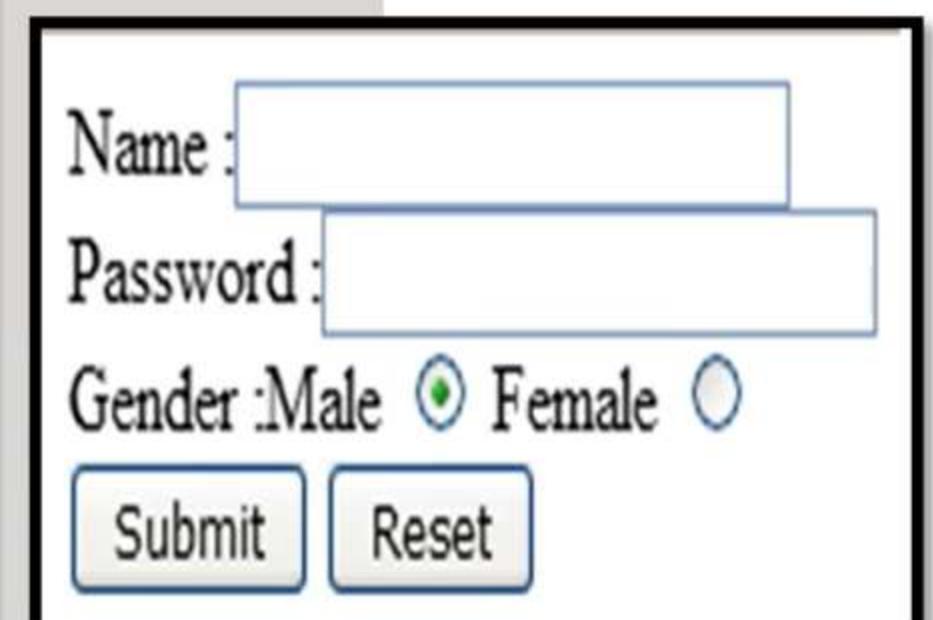
Input types include

- text
- checkbox
- radio
- password
- button
- submit
- reset
- hidden
- file
- Image

The default type is **text**.

Example - 1

```
<html>
<body>
Name      :<input type="text" name="name"/><br>
Password  :<input type="password" name="password"/><br>
Gender    :Male <input type="radio" name="gender" value="male" checked="checked" />
           Female <input type="radio" name="gender" value="female"/>    <br>
<input type="submit" value="Submit"/>
<input type="reset" value="Reset"/>
</body>
</html>
```



Name :

Password :

Gender : Male Female

Example - 2

```
<html>
<body>
<select name="qual">
<option value="UG">UG</option>
<option value="PG">PG</option>
</select>
<input type="button" value="signin"/>
<input type="file" name="img"/>
</body>
</html>
```



UG ▾

UG

PG

signin

Browse...

html_form.jpg

Example - 3

```
<html>
<body>
Certification Java <input type="checkbox" name="cert" />
    Oracle <input type="checkbox" name="cert"/><br>
Address <textarea name="address" cols="20" rows="5"></textarea><br>
<input type="reset" value="Reset"/>
<input type="submit" value="Submit" />
</body>
</html>
```



Certification Java Oracle

Address

Reset Submit

HTML5 Form

HTML5 web forms have introduced new

- Form elements
- Input types
- Attributes
- form validation
- placeholder text



Attributes



Form Attributes:

- autocomplete
- novalidate

Input Attributes:

- autocomplete
- autofocus
- formaction
- formmethod
- formnovalidate
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

Form Attributes

Autocomplete

- Specifies whether a form or an input field should have autocomplete on or off.
- Automatically completes values based on values that a user has entered before

Novalidate

- Boolean attribute.
- Specifies that the form-data (input) should not be validated when submitted.

Form Attributes

FormAction

- Specifies the URL of a file that will process the input control when the form is submitted.
- Overrides the action attribute of the <form> element.

FormMethod

- This attribute defines the HTTP method for sending form-data to the action URL.
- The formmethod attribute overrides the method attribute of the <form> element.

Formnovalidate and Autofocus

FormNoValidate

- The formnovalidate attribute is a boolean attribute.
- It specifies that the <input> element should not be validated when submitted.
- Overrides the novalidate attribute of the <form> element.

Autofocus

- The autofocus attribute is a boolean attribute.
- It specifies that an <input> element should automatically get focus when the page loads.
- Only one form element can have autofocus in a given page.

Required

Required

HTML

- The Boolean required attribute tells the browser to submit the form only if the field in question is filled out correctly.
- If a required field is empty or invalid, the form will fail to submit, and focus will move to the first invalid form element.
- The required attribute can be set on any input type except button, range, color, and hidden, all of which generally have a default value.

Name



List and Datalist

List

- Used to bind the datalist created with the input element.

Datalist

- The `<datalist>` tag specifies a list of pre-defined options for an `<input>` element.
- The `<datalist>` tag is used to provide an "autocomplete" feature on `<input>` elements. Users will see a drop-down list of pre-defined options as they input data.

```
<html>
<body>
Data List Example
<datalist id="names">
    <option value="Ivan"></option>
    <option value="Johan"></option>
    <option value="Teena"></option>
</datalist>
<input list="names" name="name" />
</body>
</html>
```

Data List Example



Placeholder



The placeholder attribute allows a short hint to be displayed inside the form element, telling the user what data should be entered in that field.

The placeholder text disappears when the field gains focus, and reappears on blur if no data was entered.

Mark

Multiple

Multiple

HTML

- If present, a user can select more than one file when the input type is file, and can include several comma-separated email addresses when the input type is email.
- While it was available in previous versions of HTML, it could be applied to select element only.

```
<html>
<body>
Example for Multiple: <br>
Upload File <input type="file" name="img" multiple="multiple"/>
</body>
</html>
```

Example for Multiple:
Upload File Form-Elements.png

Values of Type Attribute

HTML5 gives us input types that provide for more data-specific UI elements and native data validation.

HTML5 has a total of 13 new input types:

Search email

url tel

Datetime date

Month week

Time datetime-local

Number range

color

Values of Type Attribute

search

- The search type is used for search fields
- Search type is only supported in Chrome, Opera, and safari

Search <input type="search"/>

Search X

Values of Type Attribute

url

- The url type is used for input fields that should contain a URL address.
- The value of the url field is automatically validated when the form is submitted.

```
URL <input type="url" />  
<input type="submit" />
```

URL Submit

URL Submit

! Please enter a URL.

Values of Type Attribute



email

HTML

- The email type (`type="email"`) is used for specifying one or more email addresses
- Supports the Boolean multiple attributes, allowing for multiple, comma-separated email addresses

```
Email <input type="email" />  
<input type="submit" />
```

Email Submit

 Please include an '@' in the email address. 'johan' is missing an '@'.

Values of Type Attribute



tel

- (type="tel") is used to accept telephone numbers
- Unlike the url and email types, the tel type doesn't enforce a particular syntax or pattern
- Letters and numbers—indeed, any character other than new lines or carriage returns—are valid

```
contact no : <input type="tel" />  
<input type="submit" value="submit" />
```

contact no : submit

Values of Type Attribute

number

- Restricts the user to input numbers only

Mark <input type="number" />

 Mark

range

- Restricts the user to input a value within the specified range only

Rate(1 to 10) <input type="range" min="1" max="10"/>

 Rate(1 to 10)

Values of Type Attribute

date

DOJ <input type="date"/>

DOJ

October, 2015

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

time

Time Allotted <input type="time"/>

Time Allotted

Values of Type Attribute



month

Repay month & year

Repay month & year

October, 2015

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

week

Summer holidays start from week

Summer holidays start from week

Week 42, 2015

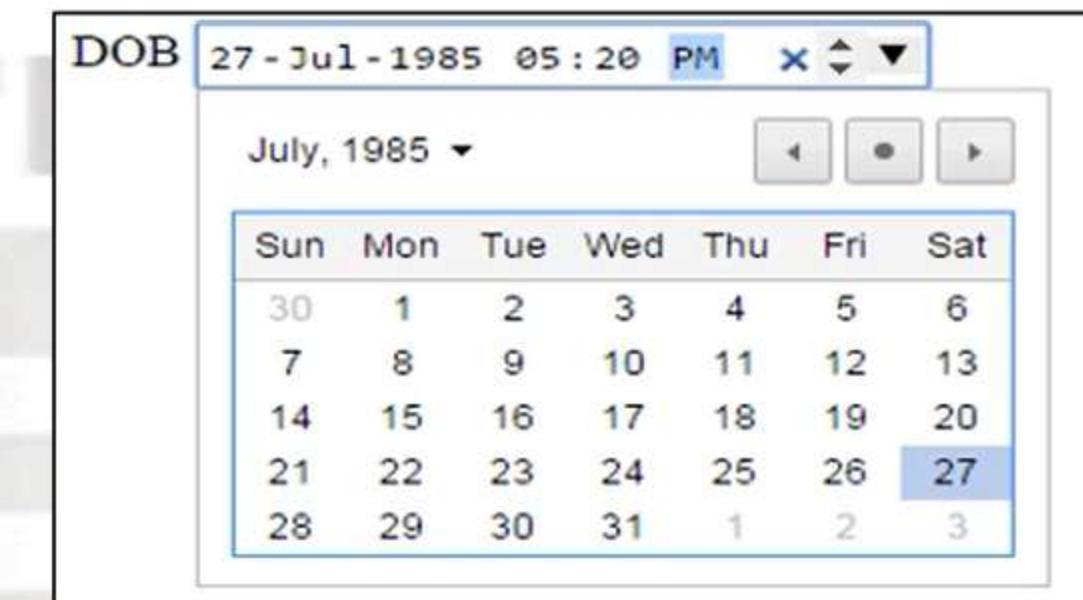
Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
40	27	28	29	30	1	2	3
41	4	5	6	7	8	9	10
42	11	12	13	14	15	16	17
43	18	19	20	21	22	23	24
44	25	26	27	28	29	30	31

Values of Type Attribute

datetime-local

- allows the user to select a date and time

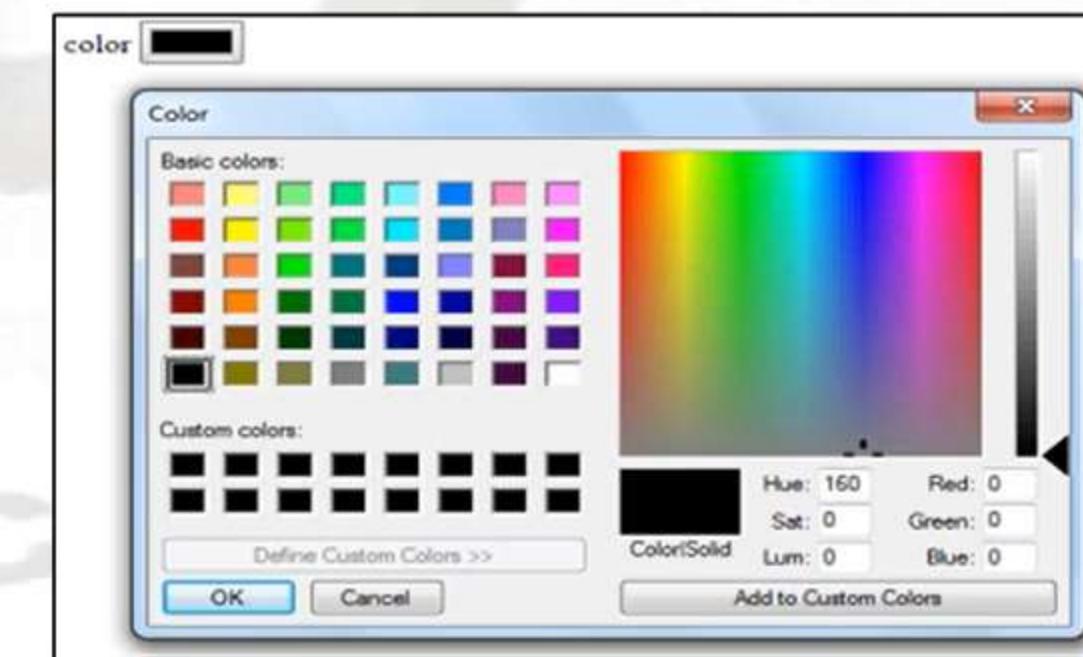
```
DOB <input type="datetime-local"/>
```



color

- allows the user to select a color

```
color <input type="color"/>
```



Restrictions on Various Input Types



Restrictions that can be given on input elements

Attribute	Description
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of characters for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

Restrictions on Various Input Types



```
<table>

    <tr><td>User Id </td><td> <input type="text" pattern="[A-Ba-b0-9_]" /></td></tr>

    <tr><td>Age </td><td> <input type="number" min=18 max=58/></td>

    <tr><td> Payment duration </td><td><input type="number" step="4"
        min="4" max="12"/> </td>

    <tr><td>DOB</td><td><input type="date" min="1965-12-31"
        max="1995-01-31" /></td></tr>

</table>
```

Restrictions on Various Input Types



User Id

Age

Payment duration

DOB

A tooltip message is displayed over the Age input field: **Please match the requested format.**

User Id

Age

Payment duration

DOB

A tooltip message is displayed over the Payment duration input field: **Please enter a valid value. The two nearest valid values are 8 and 12.**

User Id

Age

Payment duration

DOB

A tooltip message is displayed over the Age input field: **Value must be greater than or equal to 18.**

User Id

Age

Payment duration

DOB

A tooltip message is displayed over the DOB input field: **Value must be 31-Dec-1965 or later.**

Audio Tag

<audio>

HTML

- Audio files are played through a plug-in.
- HTML5 defines a new element which specifies a standard way to embed an audio file on a web page: the <audio> element.

```
<audio src="http://songserver/english/batman3/song1.mp3"></audio>
```

Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg:

Browser	MP3	Wav	Ogg
Internet Explorer 9	YES	NO	NO
Firefox 4.0	NO	YES	YES
Google Chrome 6	YES	YES	YES
Apple Safari 5	YES	YES	NO
Opera 10.6	NO	YES	YES

Video Tag

<video>

- Most video files are played through a plug-in (like flash). However, different browsers may have different plug-ins.
- HTML5 defines a new element, which specifies a standard way to embed a video file on a web page: the <video> element.

```
<video src="http://songserver/english/song1.mp3"></video>
```

Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg:

Browser	MP4	WebM	Ogg
Internet Explorer 9	YES	NO	NO
Firefox 4.0	NO	YES	YES
Google Chrome 6	YES	YES	YES
Apple Safari 5	YES	NO	NO
Opera 10.6	NO	YES	YES

Summary

- Introduction to HTML Forms
- Form-Input Elements
- HTML5 Form elements
- HTML5 Attributes
- Video Tag
- Audio Tag





#1 Table Creation

Activate Windows
Go to PC settings to activate Windows.

```
<title>MODULE TEST EVALUATION</title>
<body>
<center>
<h1>Teknoturf School of Computing</h1>
<table border="5" cellpadding="3" cellspacing="3">
<tr>
<th>Module Test Evaluation</th>
</tr>

<tr>
<th>Employee id</th>
<th>C</th>
<th>C++</th>
<th>Java</th>
<th>C#</th>
</tr>

<tr>
<td>Tek123</td>
<td>85</td>
<td>80</td>
<td>95</td>
<td>78</td>
</tr>| ①
```

File Edit Format View Help

```
<td>85</td>
<td>80</td>
<td>95</td>
<td>78</td>
</tr>

<tr>
<td>Tek145</td>
<td>100</td>
<td>85</td>
<td>90</td>
<td>86</td>
</tr>
    [
<tr>
<td>Tek146</td>
<td>98</td>
<td>80</td>
<td>78</td>
<td>89</td>
</tr>

</table>
</center>
</body>
</html>
```

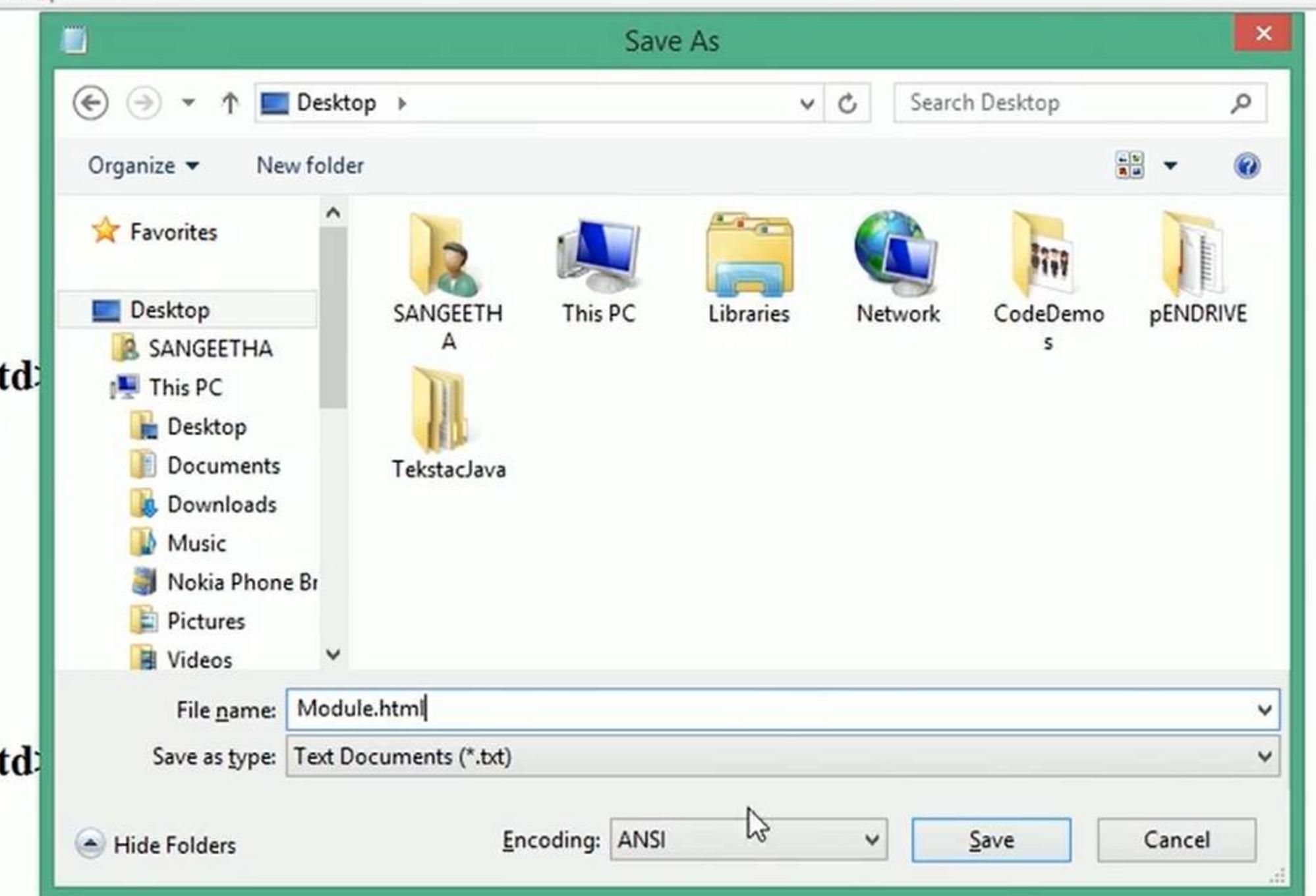
Activate Windows
Go to PC settings to activate Windows.

```
<td>85</td>
<td>80</td>
<td>95</td>
<td>78</td>
</tr>

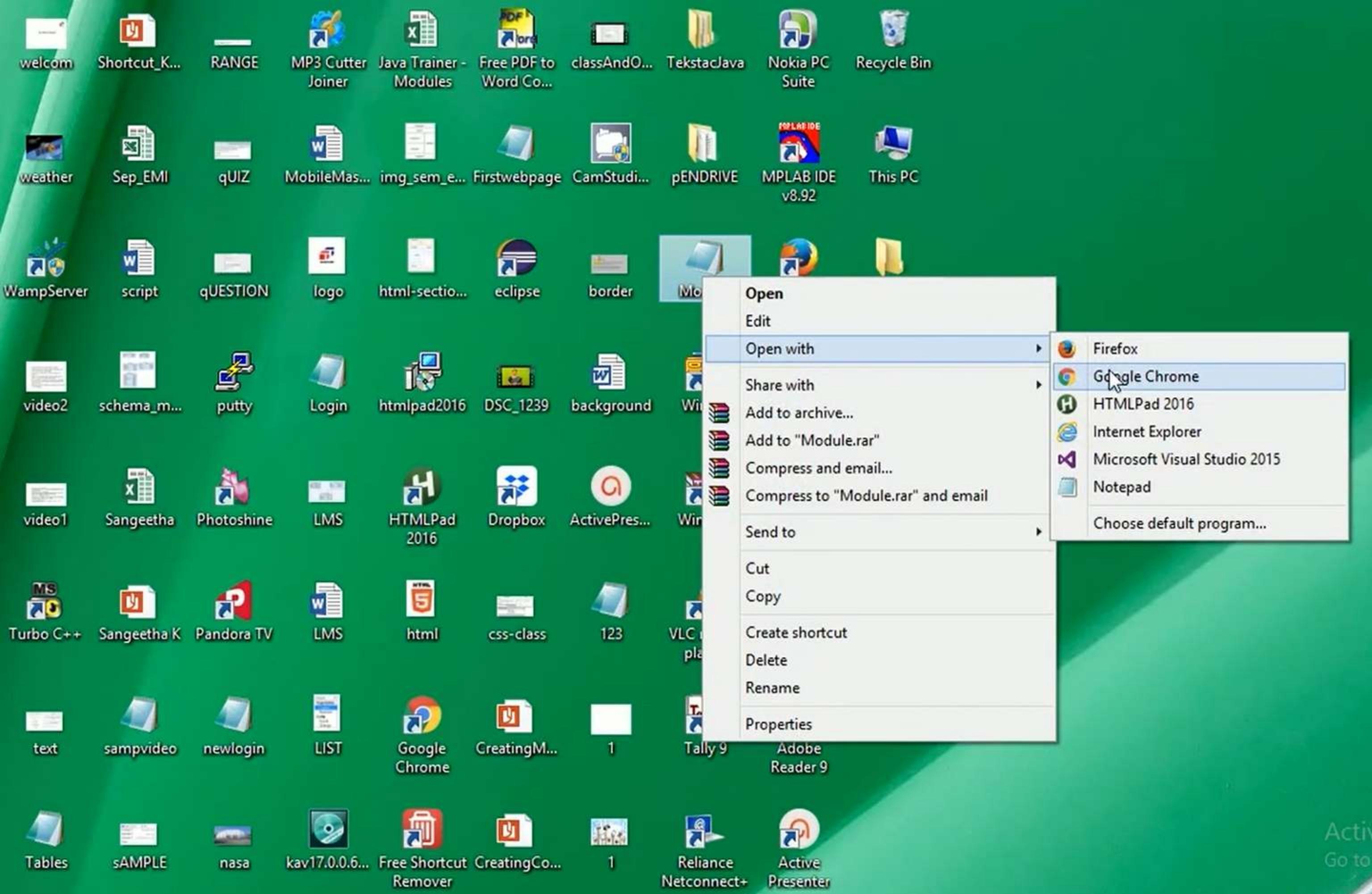
<tr>
<td>Tek145</td>
<td>100</td>
<td>85</td>
<td>90</td>
<td>86</td>
</tr>

<tr>
<td>Tek146</td>
<td>98</td>
<td>80</td>
<td>78</td>
<td>89</td>
</tr>
```

```
</table>
</center>
</body>
</html>
```



Activate Windows
Go to PC settings to activate Windows.



Teknoturf School of Computing

Module Test Evaluation				
Employee id	C	C++	Java	C#
Tek123	85	80	95	78
Tek145	100	85	90	86
Tek146	98	80	78	89



Activate Windows
Go to PC settings to activate Windows.



#3 HTML5 ELEMENTS

Activate Windows
Go to PC settings to activate Windows.

```
<html>
<body>
<center>
</img>
<table border="3" cellpadding="4" bgcolor="grey" height="40%" width="30%">
<tr>
<td><font color="white">Registered E-mail ID</font></td>
</tr>

<tr>
<td><input type="email" size="35"></td>
</tr>

<tr>
<td><font color="white">Username / Registered Mobile Number</font></td>
</tr>

<tr>
<td><input type="text" size="35" placeholder="Enter username/mobile number" required></td>
</tr>

<tr>
<td><font color="white">Password</font></td>
</tr>
<tr>
```

```
<td><input type="email" size="35"></td>
</tr>

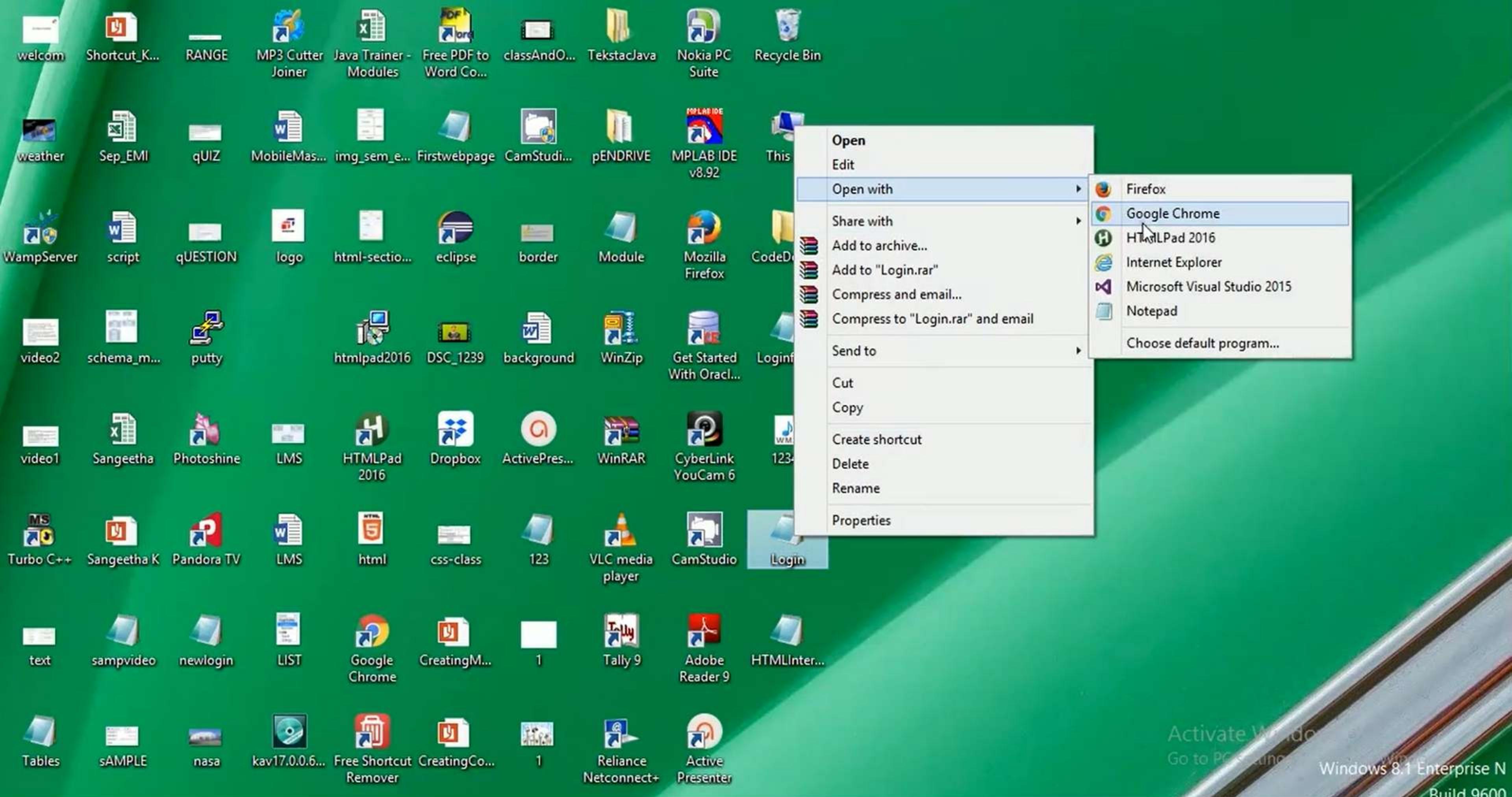
<tr>
<td><font color="white">Username / Registered Mobile Number</font></td>
</tr>

<tr>
<td><input type="text" size="35" placeholder="Enter username/mobile number" required pattern="[7-9]{1}[0-9]{9}"></td>
</tr>

<tr>
<td><font color="white">Password</font></td>
</tr>
<tr>
<td><input type="password" placeholder="Enter your password" required size="35"></td>
</tr>

<tr align="center">
<td><input type="submit" value="Login"> &nbsp;&nbsp;&nbsp;&nbsp;
<input type="reset" value="Reset"></td>
</tr>

</table>
</center>
```





Registered E-mail ID
<input type="text"/>
Username / Registered Mobile Number
<input type="text"/> Enter username/mobile number
Password
<input type="password"/> Enter your password
<input type="button" value="Login"/> <input type="button" value="Reset"/>

Activate Windows
Go to PC settings to activate Windows.

Validation.html

file:///C:/Users/SANGEETHA/Desktop/Validation.html

LOGIN

Registered E-mail ID
sangeetha@gmail.com

Username / Registered Mobile Number
9842697652

Password
.....

Login Reset

Activate Windows
Go to PC settings to activate Windows.



#2 LOGIN FORM

Activate Windows
Go to PC settings to activate Windows.

```
<html>
<body>
<center>
</img>
<table border="5" cellpadding="4" cellspacing="3" bgcolor="grey">
<tr>
<td>Username / Registered Mobile Number</td>
</tr>

<tr>
<td><input type="text"></td>
</tr>

<tr>
<td>Password</td>
</tr>
<tr>
<td><input type="password"></td>
</tr>

<tr>
<td><input type="submit" value="Login"></td>
<td><input type="reset" value="Clear"></td>
</tr>
</table>
```



File Edit Format View Help

```
<center>
<table border="5" cellpadding="4" cellspacing="3">
<tr>
<td>Username / Registered Mobile Number</td>
</tr>

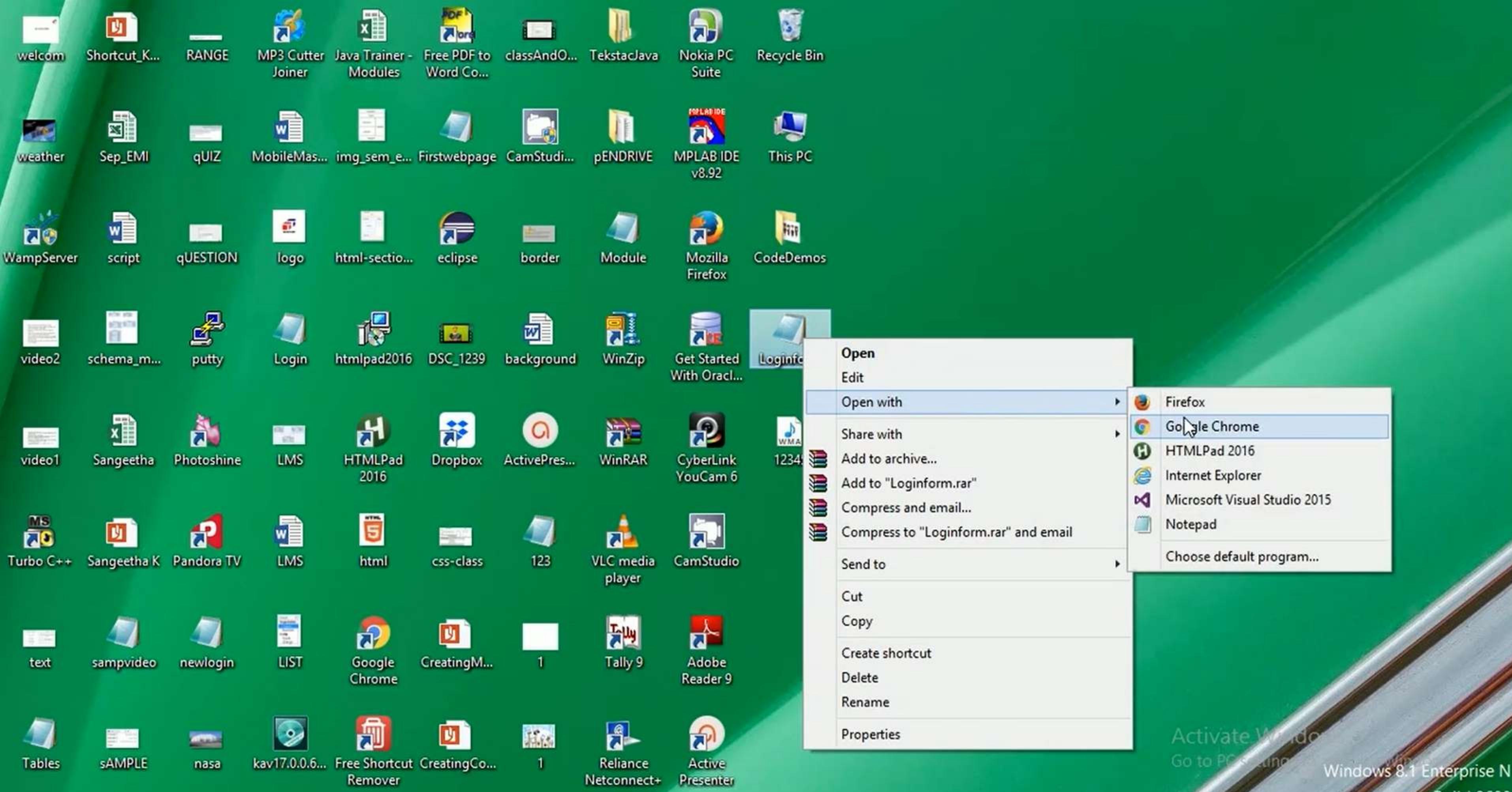
<tr>
<td><input type="text"></td>
</tr>

<tr>
<td>Password</td>
</tr>
<tr>
<td><input type="password"></td>
</tr>

<tr>
<td><input type="submit" value="Login"></td>
<td><input type="reset" value="Clear"></td>
</tr>

</table>
</center>
</body>
</html>
```

Activate Windows
Go to PC settings to activate Windows.





Username / Registered Mobile Number

Password

 T

Login

Clear

Activate Windows
Go to PC settings to activate Windows.

html5 - Forms Code - Code Demo

← → C 🔒 https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232# Press Esc to exit full screen

File List Save **Compile & Run** Evaluate Full screen Description

Run (Ctrl-F11)

CodeDemo.html

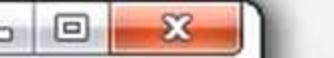
```
1  <!DOCTYPE HTML>
2
3  <html>
4    <head>
5      <script type = "text/javascript">
6
7        function showResult() {
8          x = document.forms["myform"]["newinput"].value;
9          document.forms["myform"]["result"].value = x;
10         }
11       </script>
12     </head>
13
14   <body>
15
16     <form action = "/cgi-bin/html5.cgi" method = "get" name = "myform">
17       Enter a value : <input type = "text" name = "newinput" />
18       <input type = "button" value = "Result" onclick = "showResult();" />
19       <output name = "result"></output>
20     </form>
21
22   </body>
23 </html>
```





i about:blank

Enter a value : Result



i about:blank

Enter a value : welcome

Result

welcome



html5 - Form - Validation - Code Demo

Web Technologies Code Demo X Web Technologies HTML5Form X +

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17074&userid=232#

File List Save Compile & Run Evaluate Full screen Description

html5controls.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8" />
5  <title>New controls example</title>
6  </head>
7  <body>
8  <form action"." oninput="range_control_value.value = range_control.valueAsNumber">
9  <p>
10 Name: <input type="text" name="name_control" autofocus required />
11 <br />
12 Email: <input type="email" name="email_control" required />
13 <br />
14 URL: <input type="url" name="url_control" placeholder="Write the URL of your personal website" />
15 <br />
16 Date: <input type="date" name="date_control" />
17 <br />
18 Time: <input type="time" name="time_control" />
19 <br />
20 Date and time of birth: <input type="datetime" name="datetime_control" />
21 <br />
22 Month: <input type="month" name="month_control" />
23 <br />
24 Week: <input type="week" name="week_control" />
25 <br />
26 Number (min -10, max 10): <input type="number" name="number_control" min="-10" max="10" value="0" />
27 <br />
28 Range (min 0, max 10): <input type="range" name="range_control" min="0" max="10" value="0" /> <output for="range_control" name="range_control_value" />
29 <br />
30 </form>
```

11:52 AM 2/21/2019

Web Technologies Code Demo X Web Technologies HTML5Form X +

← → C 🔒 https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17074&userid=232#

File List Save Compile & Run Evaluate Full screen Description

html5controls.html

```
10 Name: <input type="text" name="name_control" autofocus required />
11 <br />
12 Email: <input type="email" name="email_control" required />
13 <br />
14 URL: <input type="url" name="url_control" placeholder="Write the URL of your personal website" />
15 <br />
16 Date: <input type="date" name="date_control" />
17 <br />
18 Time: <input type="time" name="time_control" />
19 <br />
20 Date and time of birth: <input type="datetime" name="datetime_control" />
21 <br />
22 Month: <input type="month" name="month_control" />
23 <br />
24 Week: <input type="week" name="week_control" />
25 <br />
26 Number (min -10, max 10): <input type="number" name="number_control" min="-10" max="10" value="0" />
27 <br />
28 Range (min 0, max 10): <input type="range" name="range_control" min="0" max="10" value="0" /> <output for="range_control" name="range_control_value" />
29 <br />
30 Telephone: <input type="tel" name="tel_control" />
31 <br />
32 Search term: <input type="search" name="search_control" />
33 <br />
34 Favourite color: <input type="color" name="color_control" />
35 <br />
36 <input type="submit" value="Submit!" />
37 </p>
38 </f-->
```

Windows Taskbar: File Explorer, Google Chrome, Microsoft Edge, File Manager, Task View, Start button.

System tray: Volume, Network, Battery, Date/Time (11:52 AM, 2/21/2019).

i about:blank

Name:

Email:

URL: Write the URL of your person:

Date: mm/dd/yyyy

Time: --:--:--

Date and time of birth:

Month: -----, ----

Week: Week --, ----

Number (min -10, max 10): 0

Range (min 0, max 10): 0

Telephone:

Search term:

Favourite color:

Submit!

e_control_valu

① about:blank

Name: Abin

Email: abin@website.com

URL: http://abin.com

Date: 02/13/2019

Time: 12:59 PM

Date and time of birth: []

Month: -----, ----

Week: Week --, ----

Number (min -10, max 10): 0

Range (min 0, max 10): 0

Telephone: []

Search term: []

Favourite color: []

Submit!

e_control_valu





INTRODUCTION TO CASCADING STYLE SHEETS 3.0



In this module you will learn

- Introduction to CSS3
- CSS Syntax
- CSS Styling
- Text and Fonts properties
- CSS Selectors
- Different color schemes
- CSS Border
- CSS Margin
- CSS Background
- Multi column Layout



Introduction to CSS

CSS is the acronym for 'Cascading Style Sheets'

CSS is an extension to basic HTML that allows to style the web pages

CSS can control many elements of the web pages like colors, fonts, alignment, borders, backgrounds, spacing, margins, and much more

You may create a custom style, set all its properties, give it a unique name and then 'tag' HTML to apply these stylistic properties

CSS3 is backward compatible with its earlier versions

CSS Syntax

The styles for each element, ID, or class used on the HTML page are defined in a CSS document.

Styles are wrapped with curly brackets.

Elements are declared with the element (HTML) tag

`h1 { }`

IDs are declared with a pound sign and the ID name

`#title{ }`

Classes are declared with a period and the class name

`.text{ }`

Styles are written under `<style>` tag which is the child tag of `<head>`

CSS Syntax



What is inside the curly brackets?

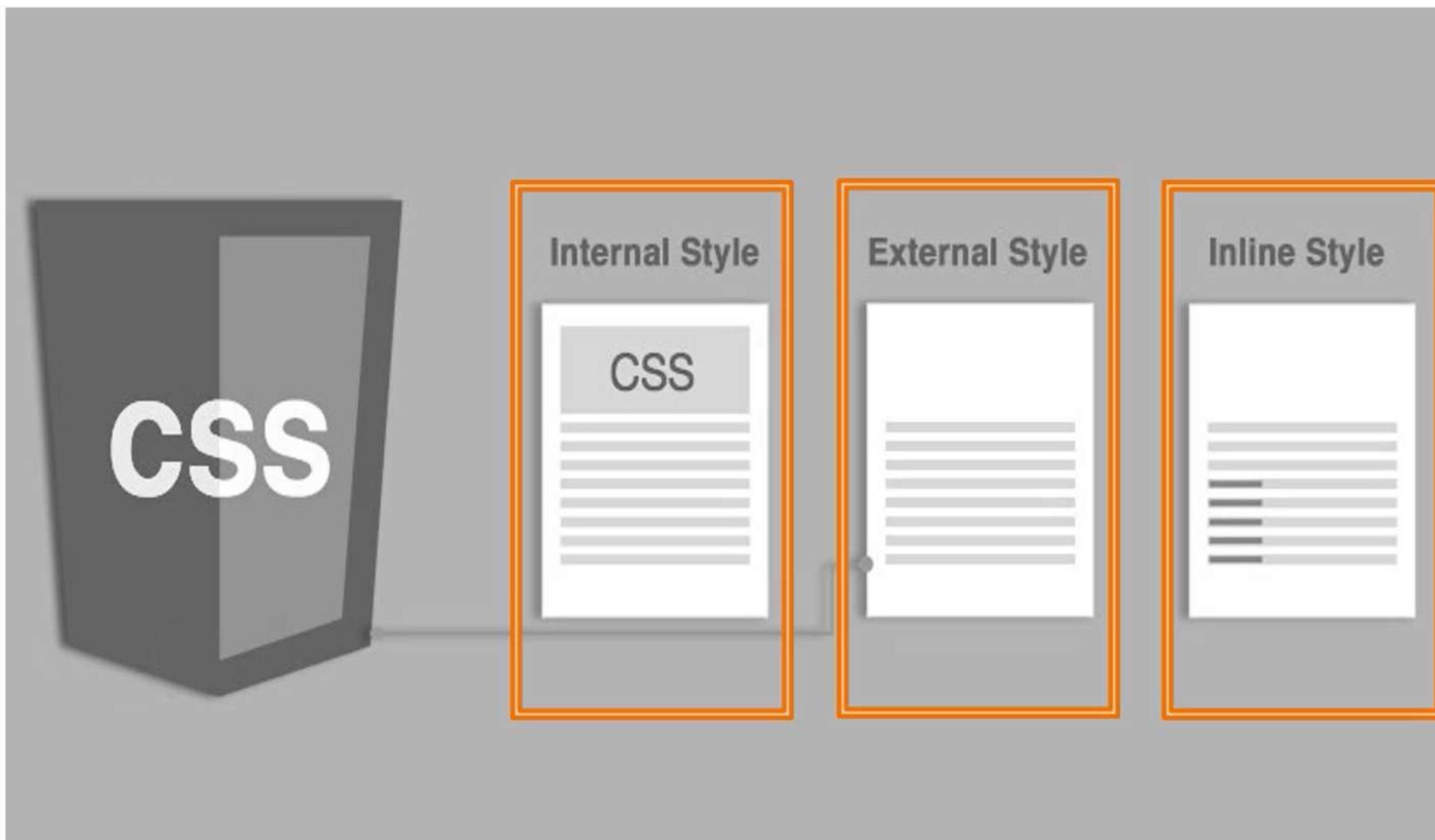
The style **font-size** applies to all `<p>` tag inside the HTML page. The text inside the tag appears with size **12px**.

Example

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
    h1{
        color: #60A216;
    }
</style>
</head>
<body>
    <h1>Welcome to CSS</h1>
    This web page gives an introduction to CSS
</body>
</html>
```



CSS Styling



Inline Style

The Style is declared inline in the html tag.

```
<!DOCTYPE HTML>  
  
<html>  
<head>  
<style type="text/css">  
  h1{  
    color: #60A216;  
  }  
</style>  
</head>  
<h1 style="color: red;">Red Heading </h1>  
<body style="background-color: lightblue;"> A Paragraph </body>  
</html>
```

Internal Style

Inline Style

Red Heading

A Paragraph

Internal Style



The styles are defined within the `<style>` tag present inside the `<head>` tag of the HTML page

```
<head>
<style>
body {
    background-color: linen;
}
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
<body>
    <h1>HTML Page</h1>
    This is a paragraph
</body>
```

External Style

Inline Style

HTML Page

This is a paragraph

External Style

The CSS Styles are written in a separate file with a .css extension



The file can be written in any text editor



It cannot have any html tags



The CSS file is linked to the html page using the <link> tag under the < head> tag



Advantage of having external style is : A single file can be used to change the look of all the web pages of the application

MyCss.css

```
body {  
    background-color: linen;  
}  
h1 {  
    color: maroon;  
    margin-left: 40px;  
}
```

Index.html

```
<head>  
<link rel="stylesheet" type=  
"text/css"  
    href="MyCss.css">  
</head>
```

Text Formatting



```
<head>
    <style type="text/css">
        #heading{
            color: #D62B00;
            text-align: center;
            text-decoration: underline;
            text-shadow: 5px 0px 3px #808080;
            word-spacing: 20px;
        }
        #content{
            text-indent: 20px;
            text-transform: capitalize;
            direction: ltr;
        }
    </style>
</head>
```

CSS3 Text Effects



CSS3 Text Effects

- text-overflow
- word-wrap
- word-break

CSS3 is an extension to CSS. It contains additional features.

Text Overflow

- It shows how overflowed content that's not displayed in the page can be signaled to the user
- It can be clipped or rendered as an ellipsis

`p {word-wrap: break-word;}`

Word-wrap

- It allows long words to be broken and to be taken to next line

Word Breaking

- It specifies the rules for breaking lines

`p.x1 {word-break: keep-all;}`
`p.x2 {word-break: break-all;}`

Fonts

Sets fonts to the contents of the web page

Properties

- **font-family** : Used to change the face of a font
- **font-style** : Used to make a font italic or oblique
- **font-variant** : Used to create a small-caps effect
- **font-weight** : Used to increase or decrease how bold or light a font appears
- **font-size** : Used to increase or decrease the size of a font
- **font** : Used as shorthand to specify a number of other font properties

CSS Selectors

It selects any element in a page that matches the selector regardless of their position in the document tree.

Element { property : value }

Syntax

h2 { color : green }

Example to apply green color to
all <h2> elements on the page

ID Selector

Selects the id attribute of the HTML element based on its value. ID value will be preceded by '#'

#id_value{ style properties }

Syntax

Example

rollno { background-color: red }

Universal and Class Selector

- **Universal selector** is denoted by (*)
- It applies styles to each element in a page
- Specific css selectors matching elements will override the styles applied by '*'

```
*{ font-family : Arial;}  
body { font-family : Corbel;}
```

Example

Class selector – It selects HTML elements having a specific class attribute. It is denoted by '.' followed by the class name.

```
.magenta  
{  
    color : magenta  
}
```

Example for applying style for all class attributes named *magenta*

Id Versus Class in CSS

IDs and classes function the same way – they both provide the same styling functionality to an HTML element, however

- **IDs are unique.** Each element can only have one ID, and that ID can only be on the page once.
- IDs can be used to style elements that are different from anything else on the page.
- **Classes are not unique;** an element can have multiple classes, and multiple elements can have the same class.
- Classes can be used to style multiple elements on a single page that need have things in common, like font size, color, or style

Adding Id And Class To HTML TAG



The id and class attributes are used along with the html tag

```
<html> <head>
<style>
#intro{
    color: green;
    margin-left: 40px;
}
.bordered{
    width: 300px;
    border: 1px solid #000;
}
</style> </head>
<body>
    <h1 id="intro">HTML Page</h1>
    <p class="bordered"> Example to explain usage of id and class in CSS</p>
</body>
</html>
```

HTML Page

Example to explain usage of id and class in CSS

```
<html>
<head>
<style type="text/css">
.id{
    color:#CC0000 ;
}
</style>
<title>Untitled</title>
</head>
<body>
    <h1 class="id">Welcome to CSS</h1>
    <p>This web page gives an introduction to CSS</p>
    <p class="id">CSS provides styles to the HTML page</p>
</body>
</html>
```

Welcome to CSS

This web page gives an introduction to CSS
CSS provides styles to the HTML page

Apply Styles



Styles mapped to a specific element.

```
<html>
<head>
<style type="text/css">
  p.id{
    color:#CC0000 ;
  }
</style>
<title>Untitled</title>
</head>
<body>
  <h1 class="id">Welcome to CSS</h1>
  <p>This web page gives an introduction to CSS</p>
  <p class="id">CSS provides styles to the HTML page</p>
</body>
</html>
```

Welcome to CSS

This web page gives an introduction to CSS

CSS provides styles to the HTML page

Applying same styles to many elements

```
<html>
<head>
<style type="text/css">
  h1,p {
    color: #FF3300;
  }
</style>
</head>
<body>
  <h1>Welcome to CSS</h1>
  <p>This web page gives an introduction to CSS</p>
  CSS provides styles to the HTML page<br>The second line font will not change
</body>
</html>
```

Welcome to CSS

This web page gives an introduction to CSS

CSS provides styles to the HTML page

The second line font will not change



CSS Background



Gives the HTML page a change in the background

CSS Properties for background

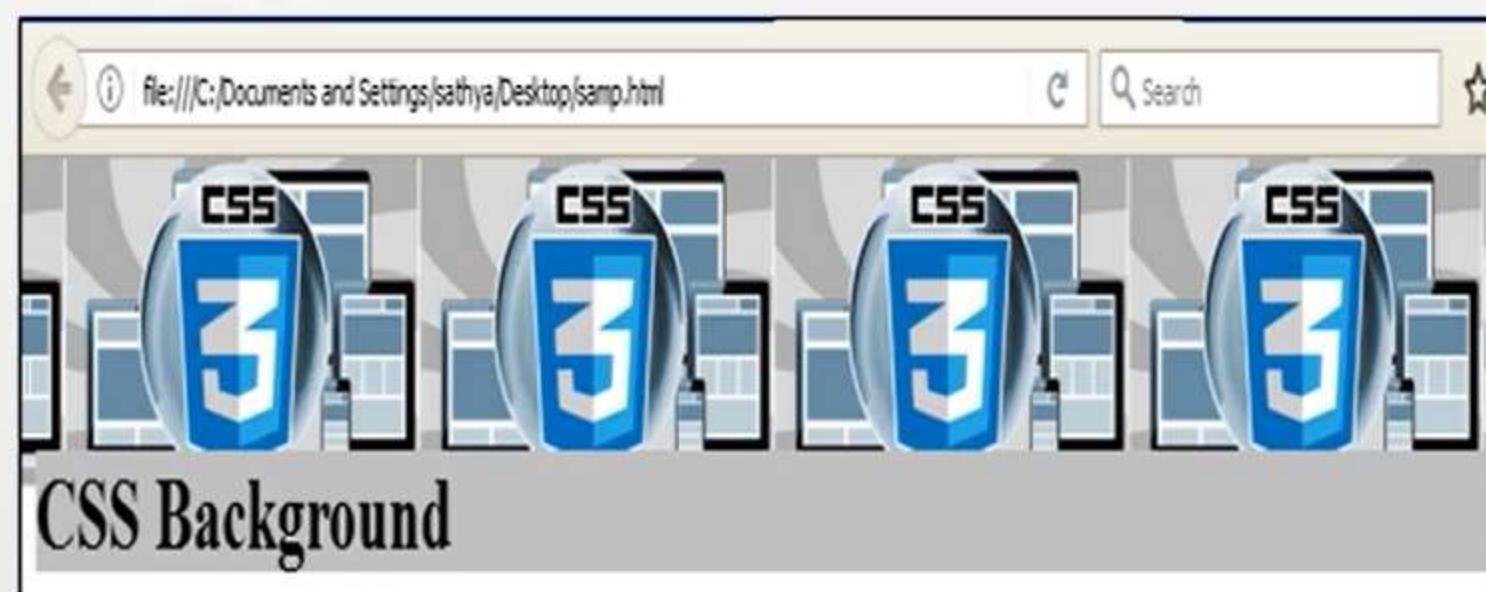
- **background-color** - Sets the background color of the HTML element
- **background-image** - Sets the background of an element with an image
- **background-repeat** -Repeats the background image
- **background-attachment**- Sets whether background image should be scrolled or fixed.
- **background-position**-Sets the starting position of the background image.
- **Background-size** - Specifies the size of background images

Example

```
<html>
<head>
<style type="text/css">
    body{
        background-image: url("css3logo.jpg");
        background-repeat: repeat-x;
        background-position: right top;
        background-attachment: fixed;
        background-size: 200px 100px;
    }
    h1 { background-color:#c0c0c0; }

</style>
</head>
<body>
    <br><br>
    <br><h1>CSS Background</h1>
</body>
</html>
```

Background
properties



CSS3 Background



There are new additions to the Background Category in CSS3

- background-size
 - It is used for creating scalable graphics
- background-origin
 - It offers three locations to position background images
- background-clip
 - It allows background colors to be clipped to the contents of the box

Multiple Backgrounds in CSS3



Multiple Background images can be applied to elements

They are layered on top of each other

The background color must be included as the last background,
since it will appear only as the last layer.

Multiple backgrounds can be specified using background-image
property or using background shorthand property

Multiple Background Using Individual Background Properties



Multiple background images can be specified using **comma** separated list.

Example

```
Background-image: url(flower.jpg), url(marble.jpg)
```

A comma separated list can be used for other background properties like background-position, background-clip and so on.

Multiple backgrounds can be specified using the background shorthand property

Example

```
Background: url(bird.jpg) top left no-repeat,  
url(cat.jpg) center bottom no-repeat
```

Example

```
<body>
    <div id="image2">
        content here
    </div>
</body>
<style>
    body{
        background-image: url(images/flower.png);
    }
    #image2{
        background-image:
        url(images/Dove.png);
        background-repeat: repeat-x;
    }
</style>
```

CSS Color Themes



CSS colors can
be specified
using the
types

- HSL colors
- RGB colors
- RGBA colors
- HSLA colors

RGB Color

It is specified with `rgb(red,green,blue)`

Each color defines the intensity of the color

Each color can contain integer values from 0 to 255 or percentage from 0% to 100%

RGB color is supported by all modern browsers

To get blue color, red and green can be set to 0% and blue to 100%

Example for setting blue color using RGB - `rgb(0%,0%,100%)`

RGBA Color

The color transparency can also be set using RGBA

RGBA color values are supported in IE9+, Firefox 3+, Chrome, Safari, and in Opera 10+

Syntax:

rgba(red, green, blue, alpha);

Example:

```
background-color:  
rgba(255, 0, 0, 1);
```

- Red, Green and Blue color values can be specified between 0 to 255 or between 0% and 100% so that the desired color can be set
- Alpha value can be specified between 0.0 and 1.0 so that the color's opacity/transparency can be set. Here, 0.0 represents 'fully transparent' and 1.0 represents 'fully opaque'

RGBA Color



Converting Percentage to Integers

- To get the Integer equivalent, multiply the percentage value by 255 and then divide by 100%
- For example, `rgba(100%,64.7%,0,1)` can also be written as `rgba(255,165,0,1)`, which ultimately displays yellow color

Converting Integers to Percentages

- To get the Percentage equivalent, divide the integer by 255 and then multiply by 100%
- For example, `rgba(255,0,0,1)` can also be written as `rgba(100%,0%,0%,1%)`, which ultimately displays red color

HSL Model

Colors can be defined in HSL(Hue-Saturation-Lightness) model using hsl() notation

Syntax : **background-color: hsl (hue, saturation, lightness)**

There are six major colors in the HSV color model

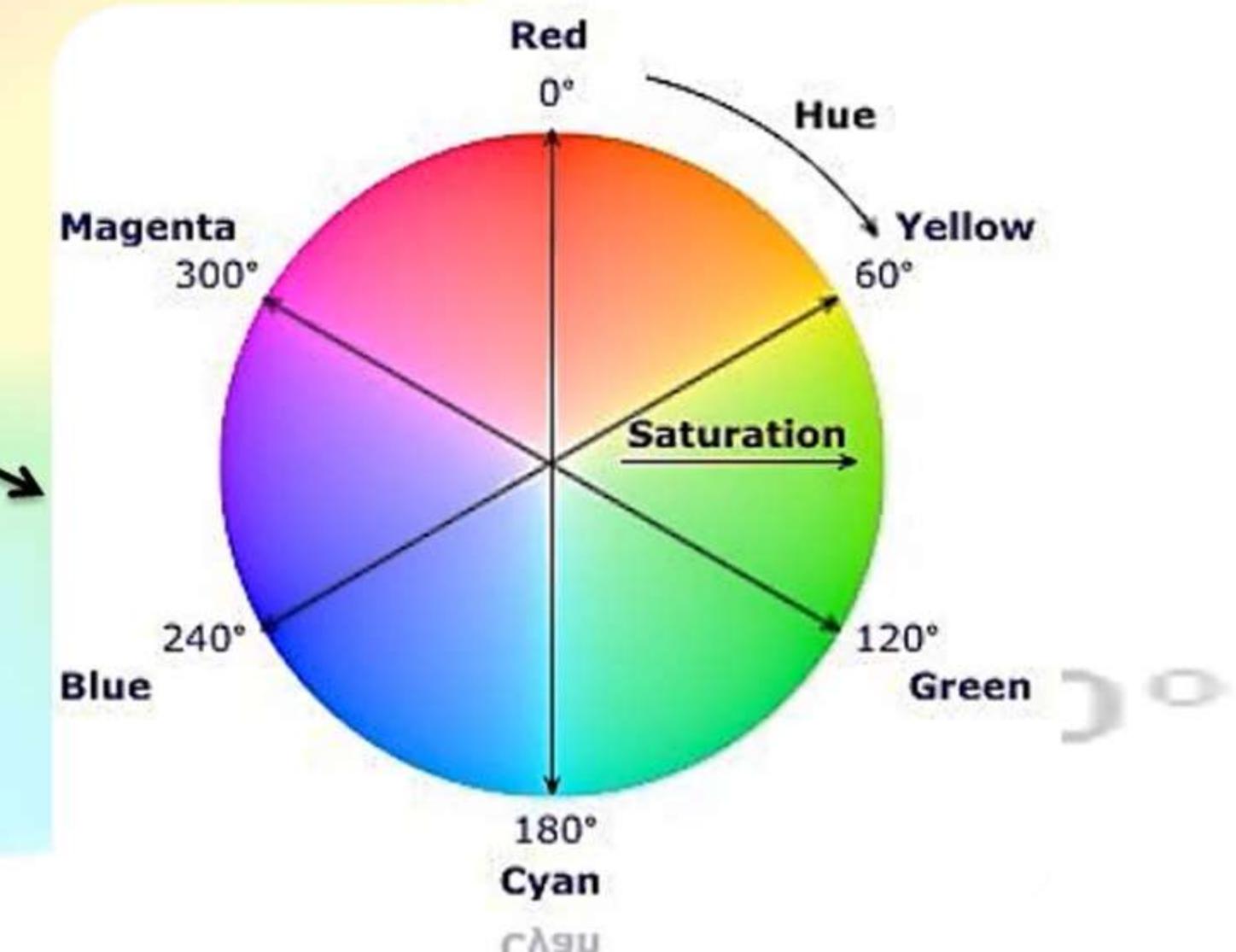
- Yellow, Cyan, Blue, Magenta, Red and Green

These colors are spaced by 60 degrees

Example :

background-color: hsl (120, 100%, 50%)

background-color: hsl (285, 100%, 50%)



HSLA Model



Colors can be defined in HSLA (Hue-Saturation-Lightness-Alpha) Model

HSLA model can be represented using hsla() notation

It is an extension of HSL model with an alpha channel

Example :

```
h1 {
    color: hsla(360,80%,50%,0.5);
}
p {
    background-color: hsla(480,60%,30%,0.3);
}
```

CSS Borders

Defines the style for the borders

Properties

- border-color
- border-style
- border-width
- border-image
- border-radius

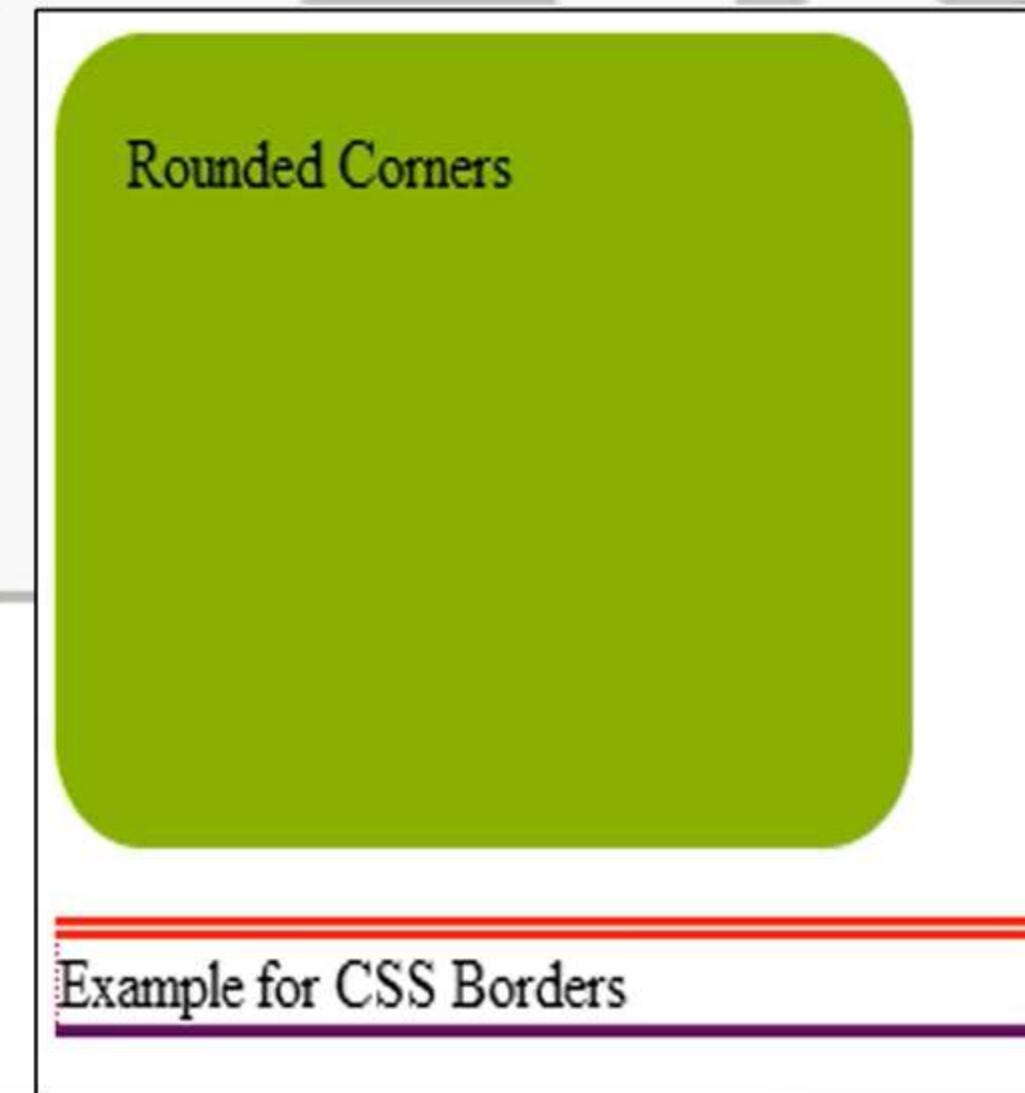
Every property of the border can be separately defined for

- top
- left
- right
- bottom
- positions of the border

Example

```
<html>
<head>
<style type="text/css">
#r1 {
    border-radius: 25px;
    background: #8AC007;
    padding: 20px;
    width: 200px;
    height: 150px;
}
#r2{
    border-bottom-style: solid;
    border-left-style: dotted;
    border-right-style: dashed;
    border-top-style: double;
    border-bottom-color: #660066;
    border-left-color: #FF0066;
    border-right-color: #003300;
    border-top-color: #FF0000;
    border-bottom-width: medium;
    border-left-width: thin;
    border-right-width: medium;
    border-top-width: thick;
}
</style>
</head>
<body>
<p id="r1"> Rounded Corners </p>
<p id="r2"> Example for CSS Borders </p>
</body>
</html>
```

Output:



Rounded Corners

Example for CSS Borders

Applying Shadows In Border



The box-shadow property can have comma separated list of values

Horizontal offset, vertical offset, optional blur distance, optional spread distance of the shadow

Example

```
box-shadow: 10px 10px;  
box-shadow: 10px 10px 5px  
#888;
```

CSS Margins

CSS margin properties are used to generate space around elements.

Margin properties

- margin-top
- margin-right
- margin-bottom
- margin-left

To set different margins for all four sides of a **<div>** tag:

```
div {  
    margin-top: 150px;  
    margin-bottom: 150px;  
    margin-right: 200px;  
    margin-left: 100px;  
}
```

Multi-column Layout

CSS3 has introduced the multi-column layout module for creating multiple column layouts in an easy and efficient way.

Create layouts like we see in magazines and newspapers without using the floating boxes.

Multi- column Layouts

Multi-column Properties:

- column-count
- column-gap
- column-rule-style
- column-rule-width
- column-rule-color
- column-rule
- column-span
- column-width

Multi-column Properties:

The **column-count** property specifies the number of columns that an element should be divided into.

The below example will divide the text in the <p> element into 3 columns:

```
p {  
    -webkit-column-count: 3; /* Chrome, Safari, Opera */  
    -moz-column-count: 3; /* Firefox */  
    column-count: 3;  
}
```

The **column-gap** property specifies the gap between the columns.

```
p {  
    -webkit-column-gap : 40px; /* Chrome, Safari, Opera */  
    -moz-column-gap: 40px; /* Firefox */  
    column-gap: 40px;  
}
```

The column-rule Property

column-rule-style property specifies the style of the rule between columns.

column-rule-width property specifies the width of the rule between columns.

column-rule-color property specifies the color of the rule between columns.

To set the width, style, and color of the rule all-together between columns:

```
p {  
    -webkit-column-rule: 1px solid lightblue;  
    -moz-column-rule: 1px solid lightblue;  
    column-rule: 1px solid lightblue;  
}
```

Example :

Output

```
<html> <head> <style type="text/css">  
p {  
    -webkit-column-count: 3;  
    -moz-column-count: 3;  
    column-count: 3;  
  
    -webkit-column-gap: 40px;  
    -moz-column-gap: 40px;  
    column-gap: 40px;  
  
    -webkit-column-rule: 1px solid lightblue;  
    -moz-column-rule: 1px solid lightblue;  
    column-rule: 1px solid lightblue;  
  
    -webkit-column-span: all;  
    column-span: all;  
}  
</style> </head> <body> <p>  
C-37 was a largely commercial flight as all but three passenger satellites,  
small nanosats, belonged to six other countries.  
The 29-minute launch went off precisely as planned; it took just 11 minutes from  
the release of the primary Cartosat-2 series spacecraft to  
the last launch of a client satellite, ISRO said after the mega-payload launch.  
The PSLV, in the category of launch vehicles that can lift relatively light loads to space,  
now marks 38 successful missions in a row out of a total of 39 flights  
This time, it took to space a total of 1,378 kg, of which  
the primary satellite was 714 kg.The latest Cartosat is the  
fifth in the series of six Cartosat-2 spacecraft, starting from  
Cartosat 2 in 2007 and followed by what were earlier marked A, B, C, D and E. The last one is due.  
</p> </body> </html>
```

C-37 was a largely commercial flight as all but three passenger satellites, small nanosats, belonged to six other countries. The 29-minute launch went off precisely as planned; it took just 11 minutes from the release of the primary Cartosat-2 series spacecraft to the last launch of a client satellite,

ISRO said after the mega-payload launch. The PSLV, in the category of launch vehicles that can lift relatively light loads to space, now marks 38 successful missions in a row out of a total of 39 flights This time, it took to space a total of 1,378 kg, of which the primary satellite was 714 kg.The

latest Cartosat is the fifth in the series of six Cartosat-2 spacecraft, starting from Cartosat 2 in 2007 and followed by what were earlier marked A, B, C, D and E. The last one is due.

Summary

- Introduction to CSS3
- CSS Syntax
- CSS Styling
- Text and Fonts properties
- CSS Selectors
- Different color schemes
- CSS Border
- CSS Margin
- CSS Background
- Multi column Layout



Web Designing - Code Demo

What is WebDesign?

Web design

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

History

1988 - 2001

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators^[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

Show all

X

```
<html>
<body>
<center><h1>What is WebDesign?</h1></center>
```

```
#def1
{
border:10px dashed black;
padding : 5px;
background-color:rgb(150,150,150)|
```

```
<div id="def1">
<h2>Web design </h2>
<p>
```

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

```
</p>
</div>
```

```
<div id="def2">
<h2>History</h2>
```

Activate Windows
Go to PC settings to activate Windows.

What is WebDesign?

Web design

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

History

1988 - 2001

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

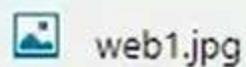
Tools and technologies

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators^[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

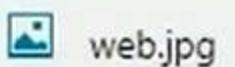
Show all



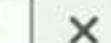
web1.jpg



web1.png



web.jpg



```
<html>
<body>
<center><h1>What is WebDesign?</h1></center>
<style>
#def1
{
border:10px dashed black;
padding : 5px;
background-color:rgb(150,150,150);
} I
```

```
#def2
{
background:url(web1.jpg);
background-color:hsl(290,60%,70%);
```

```
|</style>
<div id="def1">
<h2>Web design </h2>
<p>
```

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their

What is WebDesign?

Web design

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

History

1988 - 2001

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators^[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

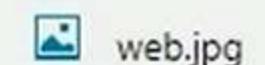
Show all



web1.jpg



web1.png



web.jpg



```
<html>
<body>
<center><h1>What is WebDesign?</h1></center>
<style>
#def1
{
border:10px dashed black;
padding : 5px;
background-color:rgb(150,150,150);
}

#def2
{
background:url(web1.jpg);
background-repeat:no-repeat;
background-color:hsl(290,60%,70%);
}

#def3
{
column-count:3;
column-rule:1px dashed blue|
```



```
</style>
<div id="def1">
<h2>Web design </h2>
```

Activate Windows
Go to PC settings to activate Windows.

What is WebDesign?

Web design

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

History

1988 - 2001



Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators^[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

Show all

<h2>History</h2>

1988 - 2001

<hr>

<p>

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

</p>

</div>

<div id="def3">

<h2>Tools and technologies</h2>

<hr>

<p id="def3">

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.[8]

</p>

</div>

</body>

</html>

What is WebDesign?

Web design

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

History

1988 - 2001



Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web designers use a variety of different

tools depending on what part of the

production process they are involved in.

These tools are updated over time by newer standards and software but the principles behind them remain the

same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design

prototypes. Technologies used to create websites include W3C standards like HTML and CSS,

which can be hand-coded or generated by WYSIWYG editing software. Other tools web

designers might use include mark up validators^[7] and other testing tools for usability and

accessibility to ensure their websites meet web accessibility guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

Show all





TEXT & FONT in CSS3

Activate Windows
Go to PC settings to activate Windows.

Web design

From Wikipedia, the free encyclopedia

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.



History

1988 - 2001

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators^[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.^[8]



Activate Windows

Go to PC settings to activate Windows

Show all

X



web.jpg



```
<html>
<head>
</head>

<body>
 </img>
```

I

```
<h2>Web design</h2>
From Wikipedia, the free encyclopedia
```

```
<hr>
<p id="def1">
Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.
</p>
```

```
<h2>History</h2>
```

1988 - 2001

```
<hr>
```

```
<p id="def2">
```

```
<body>
 </img>

<h2>Web design</h2>
From Wikipedia, the free encyclopedia
<hr>
<p id="def1" style="color:red;">
Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.
</p>

<h2>History</h2>
1988 - 2001
<hr>
<p id="def2">
Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.
</p>
```

Web design

From Wikipedia, the free encyclopedia

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.



History

1988 - 2001

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators^[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

Show all

```
<html>
<head>
<style type="text/css">
p
{
background-color:yellow;
}
</
</head>

<body>
 </img>
```

<h2>Web design</h2>
From Wikipedia, the free encyclopedia

<p id="def1" style="color:red;">
Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.Activate Windows. settings to activate Windows.

</p>

Web design

From Wikipedia, the free encyclopedia

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.



History

1988 - 2001

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators^[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

Show all

```
<html>
<head>
<style type="text/css">
#def1 I
{
background-color:yellow;
}
</style>
</head>

<body>
 </img>
```

Web design

From Wikipedia, the free encyclopedia

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

</p>

```
<html>
<head>
<style type="text/css">
#def1 {
background-color:yellow;
font-style:italic;
font-family:cursive;
}
</style>
</head>

<body>
 </img>
```

Web design

From Wikipedia, the free encyclopedia

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their

Web design

From Wikipedia, the free encyclopedia



Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

History

1988 - 2001

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators^[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

Show all

```
<html>
<head>
<style type="text/css">
#def1
{
background-color:yellow;
font-style:italic;
font-family:cursive;
}   ↵

#def2
{
background-color:grey;
text-decoration:underline;
}

</style>
</head>

<body>
 </img>
```

<h2>Web design</h2>
From Wikipedia, the free encyclopedia
<hr>

Activate Windows
Go to PC settings to activate Windows.

Web design

From Wikipedia, the free encyclopedia



Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

History

1988 - 2001

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web designers use a variety of different tools depending on what part of the production process they are involved in. These tools are updated over time by newer standards and software but the principles behind them remain the same. Web designers use both vector and raster graphics editors to create web-formatted imagery or design prototypes. Technologies used to create websites include W3C standards like HTML and CSS, which can be hand-coded or generated by WYSIWYG editing software. Other tools web designers might use include mark up validators^[7] and other testing tools for usability and accessibility to ensure their websites meet web accessibility guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

Show all

```
<html>
<head>
<style type="text/css">
#def1
{
background-color:yellow;
font-style:italic;
font-family:cursive;
}

#def2
{
background-color:grey;
text-decoration:underline;
}    ↵

#def3
{
text-transform:capitalize;
}

</style>
</head>

<body>
 </img>
```

Activate Windows
Go to PC settings to activate Windows.

Web design

From Wikipedia, the free encyclopedia



Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all.^[1] The term web design is normally used to describe the design process relating to the front-end (client side) design of a website including writing mark up. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and if their role involves creating mark up then they are also expected to be up to date with web accessibility guidelines.

History

1988 - 2001

Although web design has a fairly recent history, it can be linked to other areas such as graphic design. However, web design can also be seen from a technological standpoint. It has become a large part of people's everyday lives. It is hard to imagine the Internet without animated graphics, different styles of typography, background, and music.

Tools and technologies

Web Designers Use A Variety Of Different Tools Depending On What Part Of The Production Process They Are Involved In. These Tools Are Updated Over Time By Newer Standards And Software But The Principles Behind Them Remain The Same. Web Designers Use Both Vector And Raster Graphics Editors To Create Web-Formatted Imagery Or Design Prototypes. Technologies Used To Create Websites Include W3C Standards Like HTML And CSS, Which Can Be Hand-Coded Or Generated By WYSIWYG Editing Software. Other Tools Web Designers Might Use Include Mark Up Validators^[7] And Other Testing Tools For Usability And Accessibility To Ensure Their Websites Meet Web Accessibility Guidelines.^[8]

Activate Windows

Go to PC settings to activate Windows

Show all

Bootstrap CSS

BOOTSTRAP



In this module you will learn

- Introduction to Bootstrap
- Bootstrap Basics
- Bootstrap Components



Introduction To Bootstrap

- It was developed by Mark Otto and Jacob Thornton
- It is a mobile first front-end framework for web development
- Works with HTML, CSS and Javascript
- It is easy to create responsive designs
- It has HTML and CSS based design templates for buttons, tables and other components

Why Use Bootstrap?

- It is easy to get started
- It is supported by all popular browsers
- Great Grid System
- Bulk amount of components
- Base styling for most HTML components
- Bundled with Javascript plugins

Bootstrap Basics

- Bootstrap Environment Set-up
 - Bootstrap can be downloaded from <http://getbootstrap.com/>

Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

[Get started](#)[Download](#)

Bootstrap Basics

- You can get bootstrap in two forms:
 - Precompiled version
 - If the user is not interested in changing the source code, the precompiled version can be used
 - Source code version
 - It uses Less CSS preprocessor, but if the user is more into SASS, an official SASS port for Bootstrap is also available
 - The developers can change and build their own version of bootstrap with all the provided styles

Precompiled Version

- It contains compiled minified version of CSS and Javascript files
- File Structure

```
bootstrap/
+-- css/
| +-- bootstrap.css
| |-- bootstrap.min.css
| |-- bootstrap-theme.css
| |-- bootstrap-theme.min.css
+-- js/
| |-- bootstrap.js
| |-- bootstrap.min.js
+-- fonts/
|-- glyphicons-halflings-regular.eot
|-- glyphicons-halflings-regular.svg
|-- glyphicons-halflings-regular.ttf
|-- glyphicons-halflings-regular.woff
```

Creating Web Page With Bootstrap

Hello.html

```
<!DOCTYPE html>
<html>
<head>
<title>Basic HTML File</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <h1>Hello, world!</h1>
</body>
</html>
```

Hello, world!

Making the HTML File A Bootstrap Template

hellotemplate.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Basic Bootstrap Template</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
<!-- Optional Bootstrap theme -->
<link rel="stylesheet" href="css/bootstrap-theme.min.css">
</head>
<body>
<h1>Hello, world!</h1>
<script src="js/jquery-1.11.3.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Hello, world!

Bootstrap Templates And UI Components

- Bootstrap comes with basic HTML and CSS Templates that includes many UI components like Tables, Buttons etc.
- Bootstrap templates are available as CSS classes
- These CSS classes can be applied to HTML
- By using the class names, the components become reusable
- All the classes can be overridden with custom CSS style and color

Bootstrap Grid System

- It allows up to 12 columns across the page
 - If you do not want to use all 12 column individually, you can group the columns together to create wider columns
 - Grids provide structure to the layout defining the horizontal and vertical guidelines
 - Grid system has various classes

Bootstrap Components

- There are various Bootstrap components
 - Page Header
 - Breadcrumb
 - ButtonGroups
 - Dropdown
 - Nav & Navbars
 - Pagination

Bootstrap Buttons

- Bootstrap provides various button styles
 - Basic
 - Default
 - Primary
 - Success
 - Info
 - Warning
 - Danger
 - Link
- To achieve the button styles, bootstrap has the following classes:
 - .btn
 - .btn-primary
 - .btn-default
 - .btn-info
 - .btn-warning
 - .btn-link

Bootstrap Button Groups

- It allows multiple buttons to be stacked together on a single line.
- Bootstrap provides various classes to use button groups:

Class	Description	Code Sample
.btn-group	This class is used to form a basic button group.	<pre><div class="btn-group"> <button type="button" class="btn btn-default">Button100</button> <button type="button" class="btn btn-default">Button200</button> </div></pre>
.btn-toolbar	This class combines sets of .btn-group into a btn-toolbar for including more complex components	<pre><div class="btn-toolbar" role="toolbar"> <div class="btn-group">...</div> <div class="btn-group">...</div> </div></pre>
.btn-group-vertical	This class make a set of buttons appear vertically stacked	<pre><div class="btn-group-vertical"> ... </div></pre>

Button Group Example

```
<div class="container">
  <h2>Button Group</h2>
  <p>The .btn-group class creates a button group:</p>
  <div class="btn-group">
    <button type="button" class="btn btn-primary">TCL</button>
    <button type="button" class="btn btn-primary">Samsung</button>
    <button type="button" class="btn btn-primary">BlueBerry</button>
  </div>
</div>
```

Button Group

The .btn-group class creates a button group:

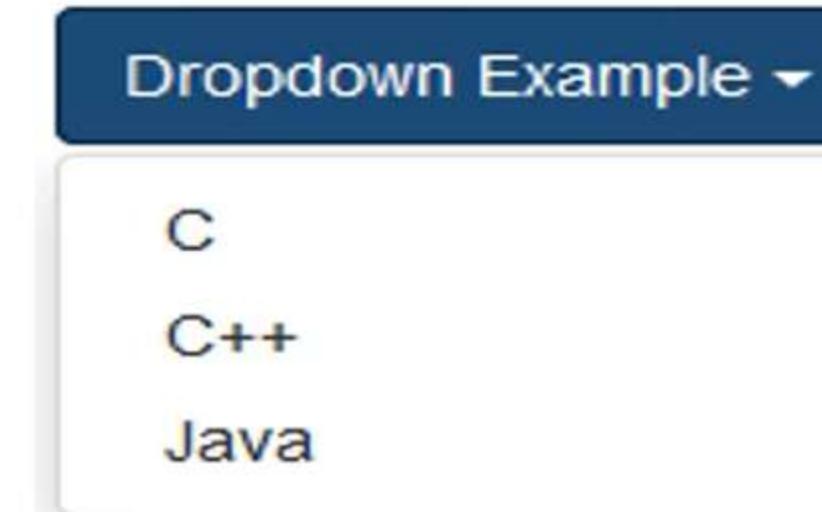
TCL Samsung BlueBerry

Dropdown

- A dropdown menu is a toggleable menu that allows the user to choose one value from a predefined list
- To add a dropdown to a button, wrap the button and dropdown menu in a **.btn-group**
- The `` can also be used to act as an indicator that the button is a dropdown
- The `.dropdown-header` class is used to add headers inside the dropdown menu
- The `.divider` class is used to separate links inside the dropdown menu with a thin horizontal border

Dropdown Example

```
<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button" data-
    toggle="dropdown">Dropdown Example
    <span class="caret"></span></button>
    <ul class="dropdown-menu">
      <li><a href="#">C</a></li>
      <li><a href="#">C++</a></li>
      <li><a href="#">Java</a></li>
    </ul>
</div>
```



Nav

- Bootstrap provides a few different options for styling navigation elements.
- All of them share the same markup and base class, **.nav**.
- Tabular Navigation
- To create a tabbed navigation menu:
 - Start with a basic unordered list with the base class of **.nav**
 - Add class **.nav-tabs**.

Nav Tabs

- Tabs are created with `<ul class="nav nav-tabs">`
- Also mark the current page with `<li class="active">`

```
<ul class="nav nav-tabs">
  <li class="active"><a href="#">Office</a></li>
  <li><a href="#">Centre 1</a></li>
  <li><a href="#">Centre 2</a></li>
  <li><a href="#">Centre 3</a></li>
</ul>
```

Inline List

Home Menu 1 Menu 2 Menu 3

Nav Menu Example

```
<ul class="list-inline">  
  <li><a href="#">Office</a></li>  
  <li><a href="#">Centre 1</a></li>  
  <li><a href="#">Centre 2</a></li>  
  <li><a href="#">Centre3</a></li>  
</ul>
```

Inline List

Home Menu 1 Menu 2 Menu 3

Nav Bar

- A navigation bar is a navigation header that is placed at the top of the page.
- With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.
- A standard navigation bar is created with `<nav class="navbar navbar-default">`.
- Inverted Navigation bars, Navigation bar with drop down is also available

Navbar Example

```
<nav class="navbar navbar-default">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Office</a></li>
      <li><a href="#">Centre 1</a></li>
      <li><a href="#">Centre 2</a></li>
      <li><a href="#">Centre3</a></li>
    </ul>
  </div>
</nav>
```

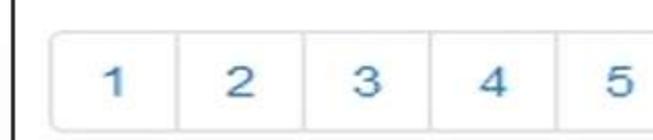


Pagination

- If you have a web site with lots of pages, you may wish to add some sort of pagination to each page.
- Pagination blocks can also be sized to a larger size or a smaller size
- To create a basic pagination, add the `.pagination` class to an ``

element:

```
<ul class="pagination">  
  <li><a href="#">1</a></li>  
  <li><a href="#">2</a></li>  
  <li><a href="#">3</a></li>  
  <li><a href="#">4</a></li>  
  <li><a href="#">5</a></li>  
</ul>
```



Breadcrumbs

- Another form for pagination, is breadcrumbs
- The .breadcrumb class indicates the current page's location within a navigational hierarchy

```
<ul class="breadcrumb">
  <li><a href="#">Trip</a></li>
  <li><a href="#">Local</a></li>
  <li><a href="#">Foreign</a></li>
  <li class="active">Vacation</li>
</ul>
```

Trip / Local / Foreign / Vacation

Page header

- The page header is a nice little feature to add appropriate spacing around the headings on a page.
- To use a page header, wrap your heading in a `<div>` with a class of **.page-header**:

```
<div class="page-header"> <h1>Example  
page header <small>Subtext for  
header</small> </h1> </div> <p>This is a  
sample text. This is a sample text. This is a  
sample text. This is a sample text.</p>
```

Summary

- Introduction to Bootstrap
- Bootstrap Basics
- Bootstrap Components





BOOTSTRAP – PART II



In this module you will learn

- Bootstrap Typography
- Helper classes and Responsive Utilities
- Glyphicons
- Bootstrap List Group
- Bootstrap Tables
- Bootstrap Forms
- Bootstrap Panel
- Jumbotron
- Well
- Alerts



Bootstrap Typography



- Bootstrap typography refers to the styling and formatting of text content like headings, paragraphs, blockquotes, etc. with Bootstrap.

Bootstrap classes for Headings

- Utilize Bootstrap's heading classes `.h1` through `.h6` on other HTML elements, when there is a need to apply the style on the text of the element similar to the headings.



```
<html lang="en">
<head>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7
/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4
/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7
/js/bootstrap.min.js"></script>
</head>
<body>
<div>
    <h1>h1. Bootstrap heading</h1>
    <h2>h2. Bootstrap heading</h2>
    <h3>h3. Bootstrap heading</h3>
    <h4>h4. Bootstrap heading</h4>
    <h5>h5. Bootstrap heading</h5>
    <h6>h6. Bootstrap heading</h6>
    <hr>
    <div class="h1">h1. Bootstrap heading</div>
    <div class="h2">h2. Bootstrap heading</div>
    <div class="h3">h3. Bootstrap heading</div>
    <div class="h4">h4. Bootstrap heading</div>
    <div class="h5">h5. Bootstrap heading</div>
    <div class="h6">h6. Bootstrap heading</div>
</div>
</body>
</html>
```

Bootstrap Typography

Bootstrap with Paragraphs:

- Bootstrap's global default font-size and line-height is applied to the <body> and all the paragraphs
- To make a paragraph stand out, just add Bootstrap class `.lead` to a <p>,

Sample code using `.lead` class :

```
<body>
  <div>
    <p class="lead">This is how a paragraph stands out in Bootstrap.</p>
  </div>
</body>
```

Sample code using `blockquote`:

```
<body>

  <blockquote class = "pull-right">This is a blockquote aligned to the right.
    <small>An example </small>
  </blockquote>

</body>
```

This is a blockquote aligned to the right.
An example —

Bootstrap Typography



Bootstrap typography classes

.small	Indicates smaller text (set to 85% of the size of the parent)
.text-left	Indicates left-aligned text
.text-center	Indicates center-aligned text
.text-right	Indicates right-aligned text
.text-justify	Indicates justified text
.text nowrap	Indicates no wrap text
.text-lowercase	Indicates lowercased text
.text-uppercase	Indicates uppercased text
.text-capitalize	Indicates capitalized text
.initialism	Displays the text inside an <code><abbr></code> element in a slightly smaller font size
.list-unstyled	Removes the default list-style and left margin on list items. It works on both <code></code> and <code></code> tags. This class only applies to immediate children list items
.list-inline	Places all list items on a single line
.dl-horizontal	Lines up the terms (<code><dt></code>) and descriptions (<code><dd></code>) in <code><dl></code> elements side-by-side.
.pre-scrollable	Makes a <code><pre></code> element scrollable

Bootstrap Typography

- **Text contextual classes** - Add meaning through text-colors with the classes below:

Class	Description
.text-muted	Text styled with class "text-muted"
.text-primary	Text styled with class "text-primary"
.text-success	Text styled with class "text-success"
.text-info	Text styled with class "text-info"
.text-warning	Text styled with class "text-warning"
.text-danger	Text styled with class "text-danger"

Bootstrap Typography



- **Bootstrap contextual classes for background** - Add meaning through background-colors with the classes below.:
.bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark and .bg-light.

```
<p class="bg-primary text-white">This text is important.</p>
<p class="bg-success text-white">This text indicates success.</p>
<p class="bg-info text-white">This text represents some information.</p>
<p class="bg-warning text-white">This text represents a warning.</p>
<p class="bg-danger text-white">This text represents danger.</p>
<p class="bg-secondary text-white">Secondary background color.</p>
<p class="bg-dark text-white">Dark grey background color.</p>
<p class="bg-light text-dark">Light grey background color.</p>
```

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary background color.

Dark grey background color.

Light grey background color.

Bootstrap Helper Classes

There are various helper classes in Bootstrap that might come in handy.

Class	Description
.pull-left	Floats an element to the left
.pull-right	Floats an element to the right
.center-block	Sets an element to <code>display:block</code> with <code>margin-right:auto</code> and <code>margin-left:auto</code>
.clearfix	Clears floats
.show	Forces an element to be shown (<code>display:block</code>)
.hidden	Forces an element to be hidden (<code>display:none</code>)
.invisible	Forces an element to be invisible (<code>visibility:hidden</code>). Will take up space on page even though it is invisible
.sr-only	Hides an element to all devices except screen readers
.sr-only-focusable	Combine with <code>.sr-only</code> to show the element again when it is focused (e.g. by a keyboard-only user)
.text-hide	Helps replace an element's text content with a background image
.close	Indicates a close icon
.caret	Indicates dropdown functionality (will reverse automatically in dropdown menus)

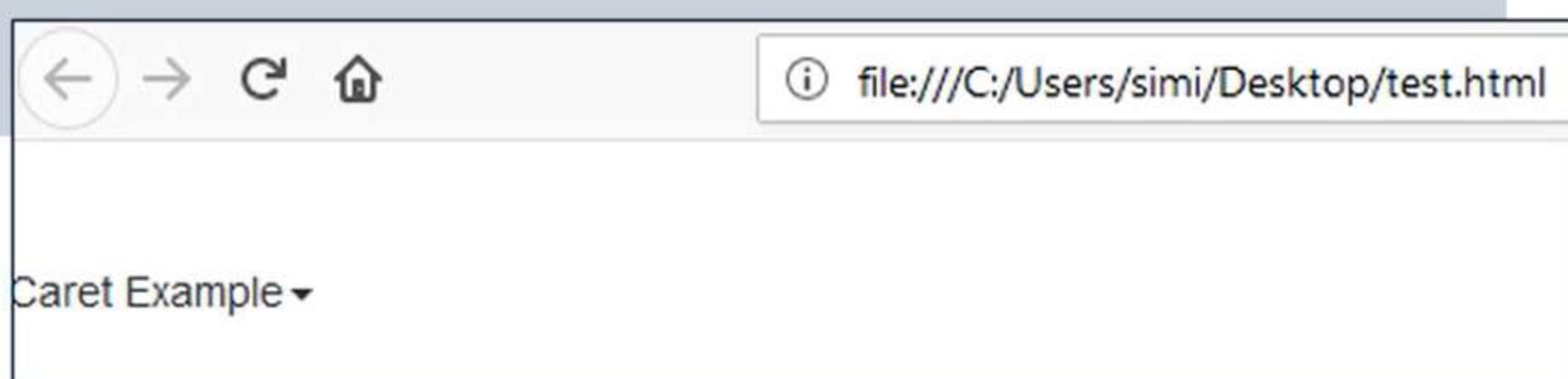
Bootstrap Helper Classes

Sample code for creating a drop down functionality using caret :

```
<!DOCTYPE html>
<html>
<head>
  <title>Try v1.2 Bootstrap Online</title>
  <link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <script src="/scripts/jquery.min.js"></script>
  <script src="/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<p>Caret Example<span class = "caret"></span></p>

</body>
</html>
```



Bootstrap - Responsive utilities

- Use responsive utility classes for showing and hiding content by device via media query for faster mobile-friendly development
- It also includes utility classes for toggling content when printed

Available Classes:

- The .hidden-* -up classes hide the element when the viewport is at the given breakpoint or wider.
- .hidden-* -down classes - hide the element when the viewport is at the given breakpoint or smaller.
- You can combine one .hidden-* -up class with one .hidden-* -down class to show an element only on a given interval of screen sizes.

Bootstrap - Responsive utilities



Classes	Devices
.visible-xs	Extra small (less than 768px) <i>visible</i>
.visible-sm	Small (up to 768 px) <i>visible</i>
.visible-md	Medium (768 px to 991 px) <i>visible</i>
.visible-lg	Larger (992 px and above) <i>visible</i>
.hidden-xs	Extra small (less than 768px) <i>hidden</i>
.hidden-sm	Small (up to 768 px) <i>hidden</i>
.hidden-md	Medium (768 px to 991 px) <i>hidden</i>
.hidden-lg	Larger (992 px and above) <i>hidden</i>

Bootstrap - Responsive utilities



	Extra small devices Portrait phones (<544px)	Small devices Landscape phones (≥544px - <768px)	Medium devices Tablets (≥768px - <992px)	Large devices Desktops (≥992px - <1200px)	Extra large devices Desktops (≥1200px)
.hidden-xs-down	Hidden	Visible	Visible	Visible	Visible
.hidden-sm-down	Hidden	Hidden	Visible	Visible	Visible
.hidden-md-down	Hidden	Hidden	Hidden	Visible	Visible
.hidden-lg-down	Hidden	Hidden	Hidden	Hidden	Visible
.hidden-xl-down	Hidden	Hidden	Hidden	Hidden	Hidden
.hidden-xs-up	Hidden	Hidden	Hidden	Hidden	Hidden
.hidden-sm-up	Visible	Hidden	Hidden	Hidden	Hidden
.hidden-md-up	Visible	Visible	Hidden	Hidden	Hidden
.hidden-lg-up	Visible	Visible	Visible	Hidden	Hidden
.hidden-xl-up	Visible	Visible	Visible	Visible	Hidden

Bootstrap - Responsive utilities

Sample code:

```
<body>

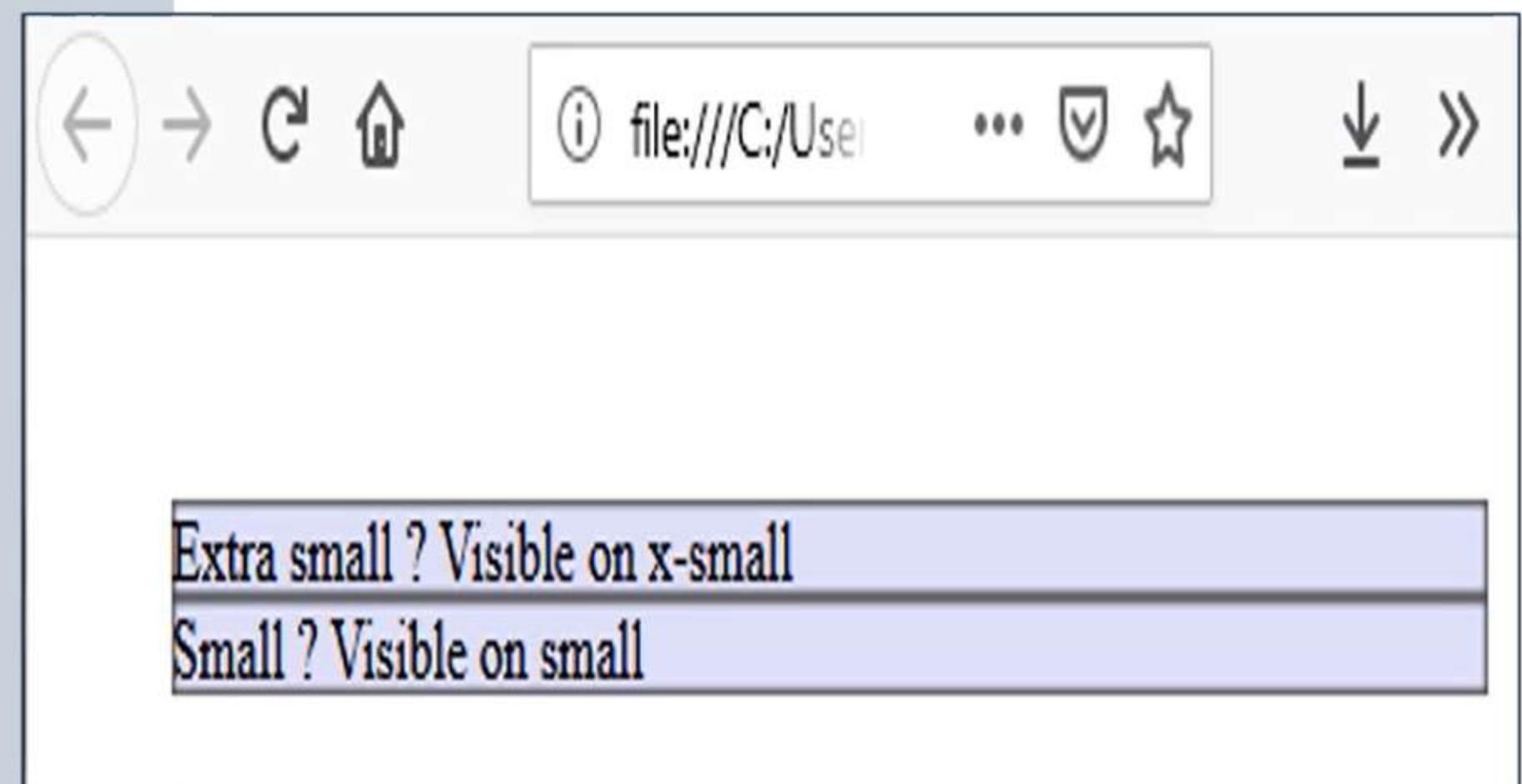
<div class = "container" style = "padding: 40px;">
  <div class = "row visible-on">

    <div class = "col-xs-6 col-sm-3" style = "background-color: #dedef8;
      box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <span class = "hidden-xs">Extra small</span>
      <span class = "visible-xs">✓ Visible on x-small</span>
    </div>

    <div class = "col-xs-6 col-sm-3" style = "background-color: #dedef8;
      box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <span class = "hidden-sm">Small</span>
      <span class = "visible-sm">✓ Visible on small</span>
    </div>

  </div>
</div>

</body>
```



Bootstrap - Tables

- Bootstrap provides a clean layout for building tables

Basic Tables

- A basic Bootstrap table has a light padding and only horizontal dividers
- Add the base class `.table` to create a basic table in Bootstrap

Syntax:

```
<table class="table">
  ...
</table>
```

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Bootstrap - Tables

Optional Table Classes: Along with the `.table` class, there are a few additional classes that you can use to style the table markup.

Striped Table : To get stripes on rows within the `<tbody>`, add the `.table-striped` class

Syntax:

```
<table class="table table-striped">
  ...
</table>
```

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Bordered table - Add `.table-bordered` for borders on all sides of the table and cells

Syntax:

```
<table class="table table-bordered">
  ...
</table>
```

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Bootstrap - Tables

Hover rows - Add `.table-hover` to enable a hover state on table rows within a `<tbody>`

Syntax:

```
<table class="table table-hover">
  ...
</table>
```

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Condensed table - Add `.table-condensed` to make tables more compact by cutting cell padding in half

Syntax:

```
<table class="table table-condensed">
  ...
</table>
```

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

Bootstrap - Tables



Contextual Classes - Contextual classes can be used to color table rows (`<tr>`) or table cells (`<td>`)

Class	Description
<code>.active</code>	Applies the hover color to a particular row or cell
<code>.success</code>	Indicates a successful or positive action
<code>.info</code>	Indicates a neutral informative change or action
<code>.warning</code>	Indicates a warning that might need attention
<code>.danger</code>	Indicates a dangerous or potentially negative action

Bootstrap - Tables

Sample code to create a table using Bootstrap's contextual classes:

```
<!DOCTYPE html> <html lang="en">
<head>
  <link rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
</head>
<body> <div class="container">
<table class="table"
```

Bootstrap - Tables

```

<tbody><tr>
  <td>Default</td> <td>Defaultson</td> <td>def@somemail.com</td> </tr>
<tr class="success">
  <td>Success</td> <td>Dan</td> <td>dan@example.com</td> </tr>
<tr class="danger">
  <td>Danger</td> <td>Ron</td> <td>ron@example.com</td> </tr>
<tr class="info">
  <td>Info</td> <td>July</td> <td>july@example.com</td> </tr>
<tr class="warning">
  <td>Warning</td> <td>Bob</td> <td>bob@example.com</td> </tr>
<tr class="active">
  <td>Active</td> <td>David</td> <td>david@example.com</td> </tr>
</tbody>
</table>
</div>
</body>
</html>

```

Firstname	Lastname	Email
Default	Defaultson	def@somemail.com
Success	Dan	dan@example.com
Danger	Ron	ron@example.com
Info	July	july@example.com
Warning	Bob	bob@example.com
Active	David	david@example.com

Bootstrap Form Layouts

Bootstrap provides various form layouts.

There are three types of form layouts:

- Vertical form (this is default)
- Horizontal form
- Inline form

Standard rules for all the three form layouts:

- Wrap labels and form controls in `<div class="form-group">` (needed for optimum spacing)
- Add class `.form-control` to all textual `<input>`, `<textarea>`, and `<select>` elements

Bootstrap Form Layouts

Vertical or Basic Form: Individual form controls automatically receive some global styling.

```
<div class="container">
<form>
  <div class="form-group">
    <label for="email">Email:</label>
    <input type="email" class="form-control" id="email" placeholder="Enter email" name="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pwd">
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
</div>
```

Vertical Form

Email:

Enter email

Password:

Enter password

Submit

Bootstrap Form Layouts

Bootstrap Inline Form :

- Where all of the elements are inline, left aligned and labels are alongside
- Additional Rule : Add the class `.form-inline` to the `<form>` tag

Example:

```
<div class="container">
  <h3>Inline Form</h3>
  <form class="form-inline" >
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="email" class="form-control" id="email" placeholder="Enter email" name="email">
    </div>
    <div class="form-group">
      <label for="pwd">Password:</label>
      <input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pwd">
    </div>
    <button type="submit" class="btn btn-default">Submit</button>
  </form>
</div>
```

Inline Form

Email:

Password:

Bootstrap Form Layouts

Bootstrap Horizontal Form: To create a form that uses the horizontal layout
Additional rules for a horizontal form:

- Add `class .form-horizontal` to the `<form>` element
- Add class `.control-label` to all `<label>` elements

Example:

```
<h3>Horizontal Form</h3>
<form class="form-horizontal">
<div class="form-group"> <label class="control-label col-sm-2" for="email">Email:</label>
  <div class="col-sm-10">
    <input type="email" class="form-control" id="email" placeholder="Enter email" name="email">
  </div> </div>
<div class="form-group"> <label class="control-label col-sm-2" for="pwd">Password:</label>
  <div class="col-sm-10">
    <input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pwd">
  </div> </div>
<div class="form-group"> <div class="col-sm-offset-2 col-sm-10">
  <button type="submit" class="btn btn-default">Submit</button>
</div> </div>
</form>
```

Horizontal Form

Email:

Password:

Bootstrap List Group

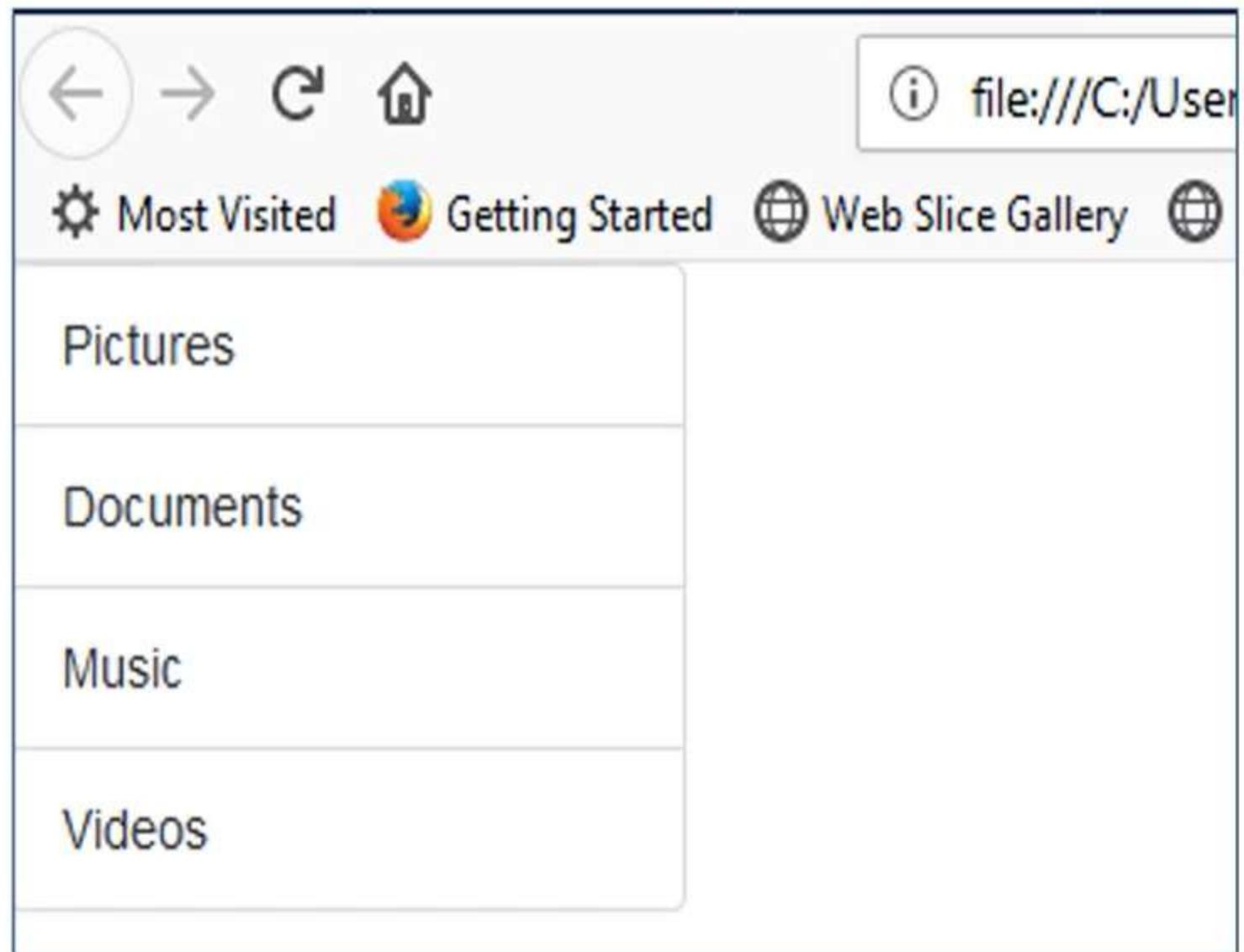
- *List groups are a flexible and powerful component for displaying a series of content*
- *The purpose of list group component is to render complex and customized content in lists*

To get a basic list group:

- *Add the class .list-group to element *
- *Add class .list-group-item to *

Example:

```
<body>
<div>
  <ul class="list-group">
    <li class="list-group-item">Pictures</li>
    <li class="list-group-item">Documents</li>
    <li class="list-group-item">Music</li>
    <li class="list-group-item">Videos</li>
  </ul>
</div>
</body>
```



Bootstrap List Group

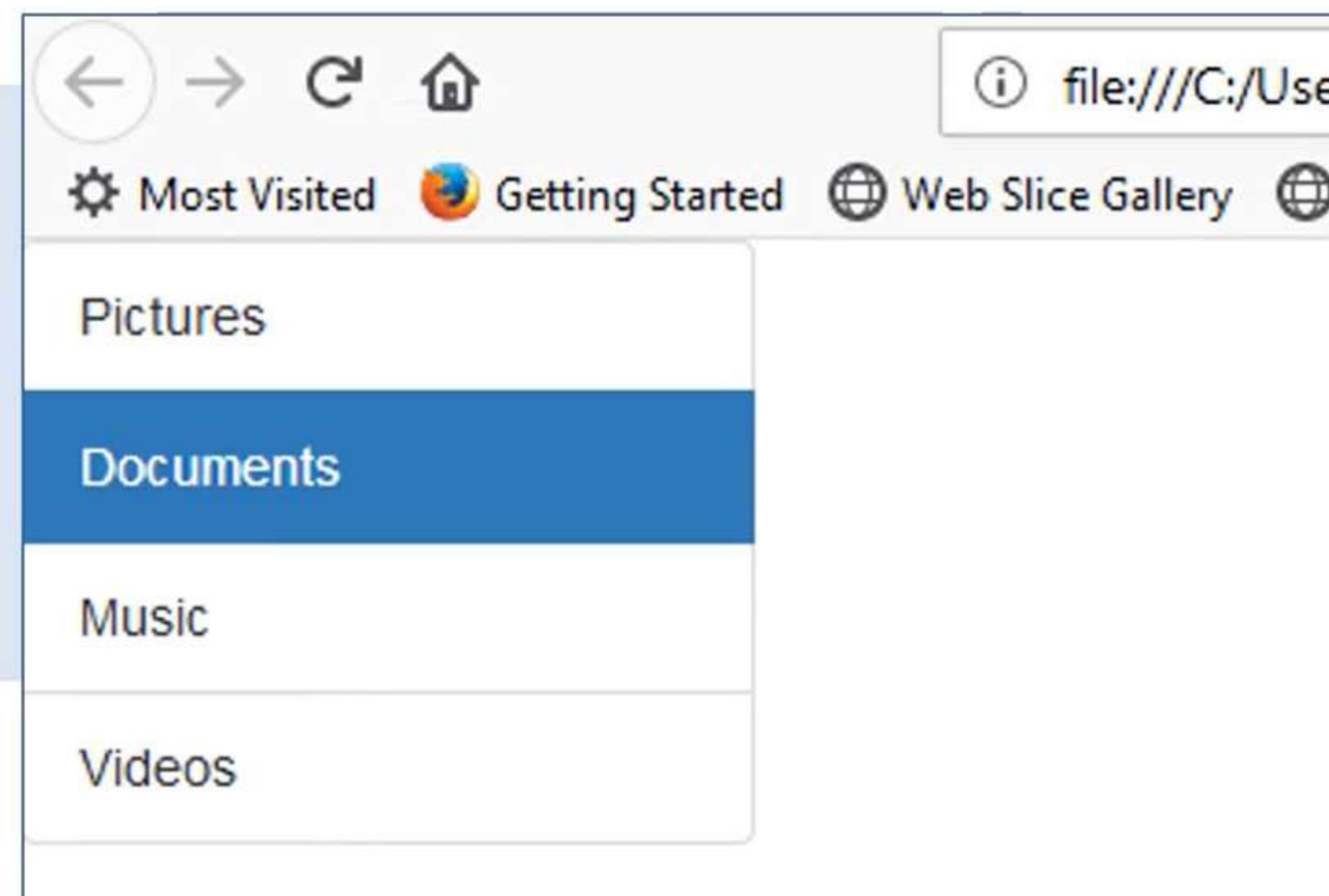


List Group with Active items :

- Add **.active** class to a **.list-group-item** class to indicate the current active selection

Example:

```
<div>
  <ul class="list-group">
    <li class="list-group-item">Pictures</li>
    <li class="list-group-item active">Documents</li>
    <li class="list-group-item">Music</li>
    <li class="list-group-item">Videos</li>
  </ul>
</div>
```



Bootstrap List Group

List Group With Linked Items:

- Use `<div>` instead of `` and `<a>` instead of ``.
- Optionally, add the `.list-group-item-action` class if you want a grey color background on hover

Example:

```
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-action">First item</a>
  <a href="#" class="list-group-item list-group-item-action">Second item</a>
  <a href="#" class="list-group-item list-group-item-action">Third item</a>
</div>
```

List Group Disabled Item:

To disable an item, add the `.disabled` class

```
<div class="list-group">
  <a href="#" class="list-group-item disabled">First item</a>
  <a href="#" class="list-group-item">Second item</a>
  <a href="#" class="list-group-item">Third item</a>
</div>
```

Bootstrap List Group

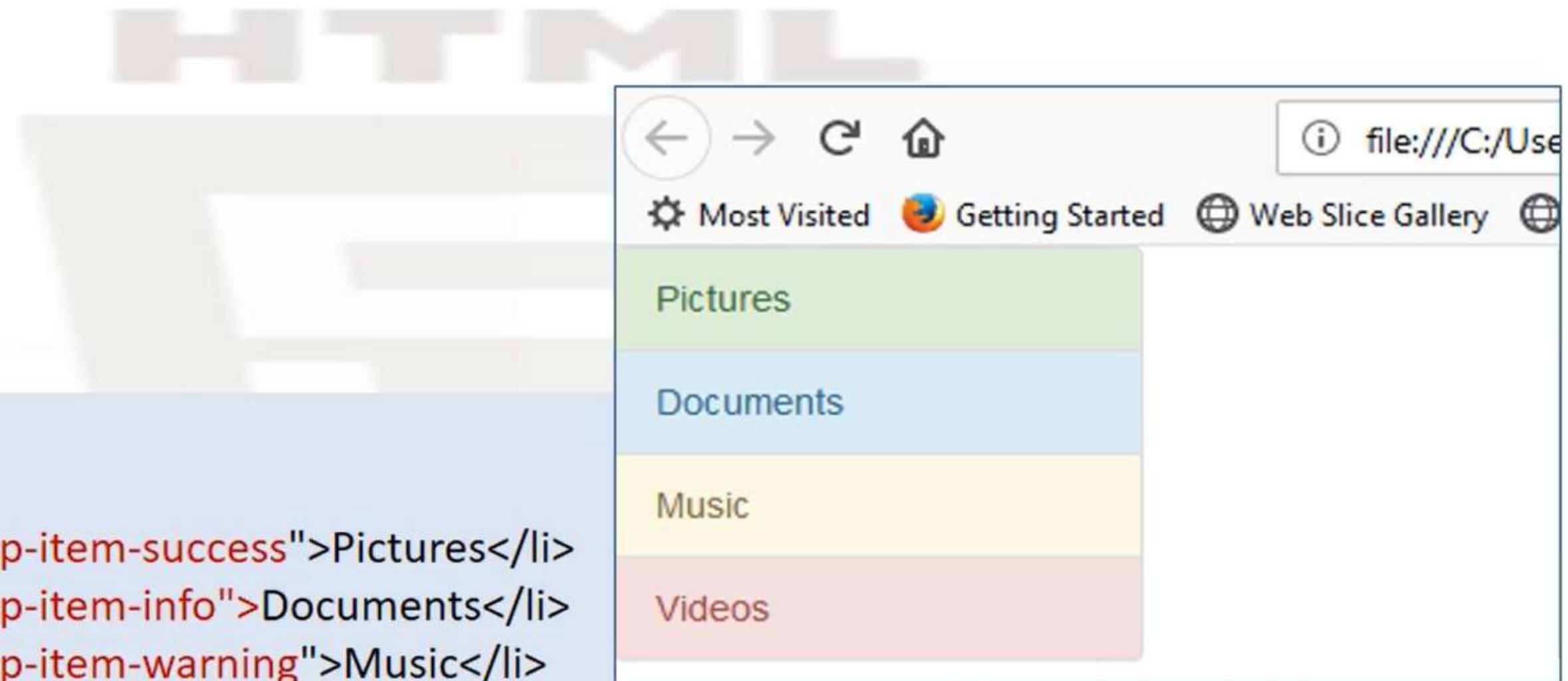
List Group Contextual Classes: Contextual classes can be used to color list items.

The classes for coloring list-items are:

- .list-group-item-success
- .list-group-item-info
- .list-group-item-warning
- .list-group-item-danger

Example:

```
<div>
  <ul class="list-group">
    <li class="list-group-item list-group-item-success">Pictures</li>
    <li class="list-group-item list-group-item-info">Documents</li>
    <li class="list-group-item list-group-item-warning">Music</li>
    <li class="list-group-item list-group-item-danger">Videos</li>
  </ul>
</div>
```



Glyphicons

- Glyphicons are icon fonts that can be used for web projects
- From the Glyphicons Halflings set Bootstrap provides 260 glyphicons
- Glyphicons can be used in text, buttons, toolbars, navigation, forms, etc

A glyphicon can be inserted with the syntax:

```
<span class="glyphicon glyphicon-name"></span>
```

Note: The **name** part in the syntax above must be replaced with the proper name of the **glyphicon**

Example:

```
<div class="container">
  <p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>
  <p>Search icon: <span class="glyphicon glyphicon-search"></span></p>
  <p>Print icon: <span class="glyphicon glyphicon-print"></span></p>
</div>
```

Envelope icon: 

Search icon: 

Print icon: 

Bootstrap Panels

- A panel in bootstrap is a bordered box with some padding around its content
- Panel components are used when you want to put your DOM component in a box
- Panels are created with the `.panel` class,
- Content inside the panel has a `.panel-body` class
- `.panel-default` class is used to style the color of the panel
- `.panel-heading` class adds a heading to the panel or use any `<h1>`- `<h6>` with a `.panel-title` class to add a pre-styled heading
- `.panel-footer` class adds a footer to the panel
- Use contextual classes : `.panel-default`, `.panel-primary`, `.panel-success`, `.panel-info`, `.panel-warning`, or `.panel-danger`

Panel Group:

- To group many panels together, wrap a `<div>` with class `.panel-group` around them.
- The `.panel-group` class clears the bottom-margin of each panel

Bootstrap Panels

Sample code:

```
<div class="panel-group">
  <div class="panel panel-default panel panel-success">
    <div class="panel-heading">I am Panel Header 1</div>
    <div class="panel-body">I am Panel Content 1</div>
  </div>
  <div class="panel panel-default panel-danger">
    <div class="panel-heading">I am Panel Header 2</div>
    <div class="panel-body">I am Panel Content 2</div>
  </div>
</div>
<div class="panel-footer">I am Panel Footer</div>
```

Panel Group

I am Panel Header 1

I am Panel Content 1

I am Panel Header 2

I am Panel Content 2

I am Panel Footer

Bootstrap Jumbotron

- It indicates a big box for calling extra attention to some special content or information
- It is displayed as a grey box with rounded corners. It also enlarges the font sizes of the text inside it
- Inside a jumbotron contain any valid HTML, including other Bootstrap elements/classes
- Use a <div> element with class .jumbotron class to create a jumbotron

Sample code:

```
<div class = "container">
  <div class = "jumbotron">
    <h1>Welcome to the world of Bootstrap!</h1>
    <p>About jumbotron</p>
    <p>
      <a class = "btn btn-primary btn-lg" role = "button">Learn more</a>
    </p>
  </div>
</div>
```

Welcome to the world of Boostrap!

About jumbotron

Learn more

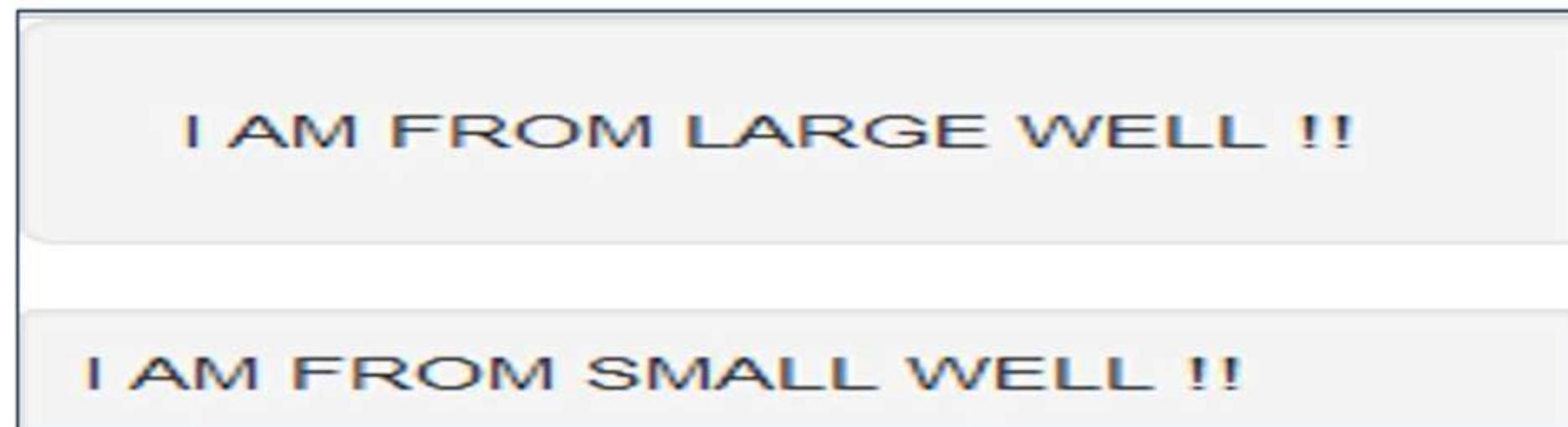
Bootstrap Well



- A well is a container in <div> that causes the content to appear sinking or an inset effect on the page.
- The `.well` class adds a rounded border around an element with a gray background color and some padding
- using the optional classes like `.well-sm` or `.well-lg`

Sample code:

```
<div class = "well well-lg">I AM FROM LARGE WELL !!</div>
<div class = "well well-sm">I AM FROM SMALL WELL !!</div>
```



A screenshot showing two separate well components. The top one is a large well with the text "I AM FROM LARGE WELL !!". The bottom one is a small well with the text "I AM FROM SMALL WELL !!". Both are contained within a light gray rectangular area.

Bootstrap Alerts

- It is an easy way to create predefined alert messages
- Create a basic alert by creating a wrapper `<div>` and adding a class of `.alert` and one of the four contextual classes like : `.alert-success`, `.alert-info`, `.alert-warning`, `.alert-danger`
- Bootstrap 4 adds 4 new alerts in Bootstrap like: `.primary`, `.secondary`, `.dark` and `.light`

```
<div class="alert alert-success">
  <strong>Success!</strong> You should <a href="#" class="alert-link">read this message</a>.
</div>
```

Links in Alerts:

- To add the `.alert-link` class to any links inside the alert box to create "matching colored links"

```
<div class = "alert alert-success">
  <a href = "#" class = "alert-link">Success! Well done its submitted.</a>
</div>

<div class = "alert alert-info">
  <a href = "#" class = "alert-link">Info! take this info.</a>
</div>
```

Success! Well done its submitted.

Info! take this info.

Bootstrap Alerts

- To close the alert message, add a `.alert-dismissible` class to the alert container.
- Add `class="close"` and `data-dismiss="alert"` to a link or a button element

Sample code:

```
<div class="container">
  <h2>Alerts</h2>
  <div class="alert alert-danger alert-dismissible">
    <button type="button" class="close" data-dismiss="alert"

Danger! This alert box could indicate a dangerous or potentially negative action. ×


```

Animated Alerts : Use `.fade` and `.show` classes to add a fading effect when closing the alert message.

Summary

- Bootstrap Typography
- Helper classes and Responsive Utilities
- Glyphicons
- Bootstrap List Group
- Bootstrap Tables
- Bootstrap Forms
- Bootstrap Panel
- Jumbotron
- Well
- Alerts



Bootstrap_Typography - Code Demo

File List

Compile & Run

Evaluate

Full screen

Description

CodeDemo.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4
5   <title>Bootstrap Typography Code Demo </title>
6   <meta charset="utf-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
9     <script src="jquery.min.js"></script>
10    <script src="bootstrap.min.js"></script>
11 </head>
12 <body>
13
14
15 <h1>I'm Heading1 h1. <small>I'm secondary Heading1 h1</small></h1>
16 <h2>I'm Heading2 h2. <small>I'm secondary Heading2 h2</small></h2>
17 <h3>I'm Heading3 h3. <small>I'm secondary Heading3 h3</small></h3>
18 <h4>I'm Heading4 h4. <small>I'm secondary Heading4 h4</small></h4>
19 <h5>I'm Heading5 h5. <small>I'm secondary Heading5 h5</small></h5>
20 <h6>I'm Heading6 h6. <small>I'm secondary Heading1 h6</small></h6>
21
22
23 <small>This content is within tag</small><br>
24 <strong>This content is within tag</strong><br>
25 <em>This content is within tag and is rendered as italics</em><br>
26
27
28 <p class = "text-left">Left aligned text.</p>
29 <p class = "text-center">Center aligned text.</p>
```

I'm Heading1 h1. I'm secondary Heading1 h1

I'm Heading2 h2. I'm secondary Heading2 h2

I'm Heading3 h3. I'm secondary Heading3 h3

I'm Heading4 h4. I'm secondary Heading4 h4

I'm Heading5 h5. I'm secondary Heading5 h5

I'm Heading6 h6. I'm secondary Heading1 h6

This content is within tag

This content is within tag

This content is within tag and is rendered as italics

Left aligned text.

Center aligned text.

Right aligned text.

This content is muted

This content carries a primary class

This content carries a success class

This content carries a info class

This content carries a warning class

This content carries a danger class

WWW

RSS

This is a default blockquote example. This is a default blockquote example.

File List

Save

Compile & Run

Evaluate

Full screen

Description

CodeDemo.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4
5      <title>Bootstrap Typography Code Demo </title>
6      <meta charset="utf-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1">
8      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
9      <script src="jquery.min.js"></script>
10     <script src="bootstrap.min.js"></script>
11 </head>
12 <body>
13
14
15     <h1>I'm Heading1 h1. <small>I'm secondary Heading1 h1</small></h1>
16     <h2>I'm Heading2 h2. <small>I'm secondary Heading2 h2</small></h2>
17     <h3>I'm Heading3 h3. <small>I'm secondary Heading3 h3</small></h3>
18     <h4>I'm Heading4 h4. <small>I'm secondary Heading4 h4</small></h4>
19     <h5>I'm Heading5 h5. <small>I'm secondary Heading5 h5</small></h5>
20     <h6>I'm Heading6 h6. <small>I'm secondary Heading1 h6</small></h6>
21
22
23     <small>This content is within tag</small><br>
24     <strong>This content is within tag</strong><br>
25     <em>This content is within tag and is rendered as italics</em><br>
26
27
28     <p class = "text-left">Left aligned text.</p>
29     <p class = "text-center">Center aligned text.</p>
```

I'm Heading1 h1. I'm secondary Heading1 h1

I'm Heading2 h2. I'm secondary Heading2 h2

I'm Heading3 h3. I'm secondary Heading3 h3

I'm Heading4 h4. I'm secondary Heading4 h4

I'm Heading5 h5. I'm secondary Heading5 h5

I'm Heading6 h6. I'm secondary Heading1 h6

This content is within tag

This content is within tag

This content is within tag and is rendered as italics

Left aligned text.

Center aligned text.

Right aligned text.

This content is muted

This content carries a primary class

This content carries a success class

This content carries a info class

This content carries a warning class

This content carries a danger class

WWW

RSS

This is a default blockquote example. This is a default blockquote example.

File List

Save

Compile & Run

Evaluate

Full screen

Description

CodeDemo.html

```
18 <n4>I'm heading4 n4. <small>I'm secondary heading4 n4</small></n4>
19 <h5>I'm Heading5 h5. <small>I'm secondary Heading5 h5</small></h5>
20 <h6>I'm Heading6 h6. <small>I'm secondary Heading1 h6</small></h6>
21
22
23 <small>This content is within tag</small><br>
24 <strong>This content is within tag</strong><br>
25 <em>This content is within tag and is rendered as italics</em><br>
26
27
28 <p class = "text-left">Left aligned text.</p>
29 <p class = "text-center">Center aligned text.</p>
30 <p class = "text-right">Right aligned text.</p>
31 <p class = "text-muted">This content is muted</p>
32 <p class = "text-primary">This content carries a primary class</p>
33 <p class = "text-success">This content carries a success class</p>
34 <p class = "text-info">This content carries a info class</p>
35 <p class = "text-warning">This content carries a warning class</p>
36 <p class = "text-danger">This content carries a danger class</p>
37
38
39 <abbr title = "World Wide Web">WWW</abbr><br>
40 <abbr title = "Real Simple Syndication" class = "initialism">RSS</abbr>
41
42
43
44
45 <blockquote>
46   <p>This is a default blockquote example. This is a default
47     blockquote example. This is a default blockquote.
```

I'm Heading1 h1. I'm secondary Heading1 h1

I'm Heading2 h2. I'm secondary Heading2 h2

I'm Heading3 h3. I'm secondary Heading3 h3

I'm Heading4 h4. I'm secondary Heading4 h4

I'm Heading5 h5. I'm secondary Heading5 h5

I'm Heading6 h6. I'm secondary Heading1 h6

This content is within tag

This content is within tag

This content is within tag and is rendered as italics

Left aligned text.

Center aligned text.

Right aligned text.

This content is muted

This content carries a primary class

This content carries a success class

This content carries a info class

This content carries a warning class

This content carries a danger class

WWW

RSS

Real Simple Syndication

blockquote example. This is a default blockquote example.

File List

Compile & Run

Evaluate

Full screen

Description

CodeDemo.html

```
35 <p class = "text-warning">This content carries a warning class</p>
36 <p class = "text-danger">This content carries a danger class</p>
37
38
39 <abbr title = "World Wide Web">WWW</abbr><br>
40 <abbr title = "Real Simple Syndication" class = "initialism">RSS</abbr>
41
42
43
44
45 <blockquote>
46     <p>This is a default blockquote example. This is a default
47         blockquote example. This is a default blockquote
48             example.This is a default blockquote example. This is a
49                 default blockquote example.</p>
50 </blockquote>
51
52 <blockquote>
53     This is a blockquote with a source title.
54     <small>Someone famous in <cite title = "Source Title">Source Title</cite></small>
55 </blockquote>
56
57 <blockquote class = "pull-right">This is a blockquote aligned to the right.
58     <small>Someone famous in <cite title = "Source Title">Source Title</cite></small>
59 </blockquote>
60
61
62 <h4>Example of Ordered List</h4>
63 <ol>
```

This content is within tag

This content is within tag

This content is within tag and is rendered as italics

Left aligned text.

Center aligned text.

Right aligned text.

This content is muted

This content carries a primary class

This content carries a success class

This content carries a info class

This content carries a warning class

This content carries a danger class

WWW

RSS

This is a default blockquote example. This is a default blockquote example.



This is a blockquote with a source title.

— Someone famous in Source Title

Example of Ordered List

1. Item 1
2. Item 2
3. Item 3
4. Item 4

This is a blockquote aligned to the right.

Someone famous in Source Title —

File List

Save

Compile & Run

Evaluate

Full screen

Description

CodeDemo.html

```
81 <ul class = "list-unstyled">
82     <li>Item 1</li>
83     <li>Item 2</li>
84     <li>Item 3</li>
85     <li>Item 4</li>
86 </ul>
87
88 <h4>Example of Inline List</h4>
89
90 <ul class = "list-inline">
91     <li>Item 1</li>
92     <li>Item 2</li>
93     <li>Item 3</li>
94     <li>Item 4</li>
95 </ul>
96
97 <h4>Example of Definition List</h4>
98
99 <dl>
100    <dt>Description 1</dt>
101    <dd>Item 1</dd>
102    <dt>Description 2</dt>
103    <dd>Item 2</dd>
104 </dl>
105
106 <h4>Example of Horizontal Definition List</h4>
107
108 <dl class = "dl-horizontal">
109    <dt>Description 1</dt>
110    <dd>Item 1</dd>
```

This is a blockquote with a source title.

— Someone famous in Source Title

Example of Ordered List

1. Item 1
2. Item 2
3. Item 3
4. Item 4

Example of UnOrdered List

- Item 1
- Item 2
- Item 3
- Item 4

Example of Unstyled List

Item 1
Item 2
Item 3
Item 4

Example of Inline List

Item 1 Item 2 Item 3 Item 4

Example of Definition List

Description 1

Item 1

Description 2

Item 2

This is a blockquote aligned to the right.

Someone famous in Source Title —

Example of Horizontal Definition List

BootStrap_Alert - Code Demo

Creating an alert with Bootstrap is easy. All we need to do is create a <div> element with classes alert and one of the following contextual state classes

alert contextual classes

alert-danger

alert-success

alert-info

alert-warning

Error! There is a problem submitting your form

```
<div class="alert alert-danger">
    <strong>Error!</strong> There is a problem submitting your form!
</div>

</div>
</div>
</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js">
</script>
<script type="text/javascript">
$(document).ready(function () {

});
</script>
<script src="bootstrap/js/bootstrap.min.js"></script>
```

Error! There is a problem submitting your form



Creating an alert with Bootstrap is easy. All we need to do is create a <div> element with classes alert and one of the following contextual state classes

alert contextual classes

alert-danger

alert-success

alert-info

alert-warning

Error! There is a problem submitting your form

Closing the alert

Submit

Error! There is a problem submitting your form

```
<div class="alert alert-danger">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Error!</strong> There is a problem submitting your form
</div>

</div>
</div>
</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js">
</script>
<script type="text/javascript">
$(document).ready(function () {

});

</script>
```

Error! There is a problem submitting your form



```
<div class="alert alert-danger collapse">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Error!</strong> There is a problem submitting your form
</div>

</div>
</div>
</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js">
</script>
<script type="text/javascript">
$(document).ready(function () {

});

</script>
```



```
<button id="btnSubmit" class="btn btn-primary">Submit</button>
<br /><br />
<div id="myAlert" class="alert alert-danger collapse">
  <a href="#" class="close" data-dismiss="alert">&times;</a>
  <strong>Error!</strong> There is a problem submitting your form
</div>

</div>
</div>
</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js">
</script>
<script type="text/javascript">
$(document).ready(function () {
  $('#btnSubmit').click(function () {
    $('#myAlert').show('fade|');
```

Submit



Submit



Error! There is a problem submitting your form



Bootstrap_Glyphicon - Code Demo

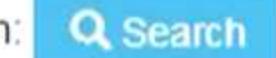
Glyphicon Examples

Envelope icon: 

Envelope icon as a link: [!\[\]\(dbb2a407c8271951d876238f3aeb1347_img.jpg\)](#)

Search icon: 

Search icon on a button:  **Search**

Search icon on a styled button:  **Search**

Print icon: 

Print icon on a styled link button:  **Print**



File List

Save

Compile & Run

Evaluate

Full screen

Description

CodeDemo.html

```
1   <meta name="viewport" content="width=device-width, initial-scale=1">
2
3   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
4   <script src="jquery.min.js"></script>
5   <script src="bootstrap.min.js"></script>
6
7   </head>
8
9   <body>
10    <div class="container">
11      <h2>Glyphicon Examples</h2>
12      <p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>
13      <p>Envelope icon as a link:
14          <a href="#">
15              <span class="glyphicon glyphicon-envelope"></span>
16          </a>
17      </p>
18      <p>Search icon: <span class="glyphicon glyphicon-search"></span></p>
19      <p>Search icon on a button:<br/>
20          <button type="button" class="btn btn-default">
21              <span class="glyphicon glyphicon-search"></span> Search
22          </button>
23      </p>
24      <p>Search icon on a styled button:
25          <button type="button" class="btn btn-info">
26              <span class="glyphicon glyphicon-search"></span> Search
27          </button>
28      </p>
29      <p>Print icon: <span class="glyphicon glyphicon-print"></span></p>
30      <p>Print icon on a styled link button:
31          <a href="#" class="btn btn-success btn-lg">
32              <span class="glyphicon glyphicon-print"></span> Print
33      </p>
```

File List

Save

Compile & Run

Evaluate

Full screen

Description

CodeDemo.html

```
14<div class="container">
15  <h2>Glyphicon Examples</h2>
16  <p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>
17  <p>Envelope icon as a link:
18    <a href="#">
19      <span class="glyphicon glyphicon-envelope"></span>
20    </a>
21  </p>
22  <p>Search icon: <span class="glyphicon glyphicon-search"></span></p>
23  <p>Search icon on a button:
24    <button type="button" class="btn btn-default">
25      <span class="glyphicon glyphicon-search"></span> Search
26    </button>
27  </p>
28  <p>Search icon on a styled button:
29    <button type="button" class="btn btn-info">
30      <span class="glyphicon glyphicon-search"></span> Search
31    </button>
32  </p>
33  <p>Print icon: <span class="glyphicon glyphicon-print"></span></p>
34  <p>Print icon on a styled link button:
35    <a href="#" class="btn btn-success btn-lg">
36      <span class="glyphicon glyphicon-print"></span> Print
37    </a>
38  </p>
39 </div>
40
41</body>
42
```

Bootstrap_Table - Code Demo

ABC company Ltd.,

Details of employees who perform well in projects

Firstname	Lastname	Email
John	Doe	john@mysite.com
Mary	Moe	mary@mysite.com
July	Dooley	july@mysite.com

File List

Save

Compile & Run

Evaluate

Full screen

Description

CodeDemo.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Table Example</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10     </head>
11     <body>
12
13     <div class="container">
14         <div class="jumbotron"><h2>ABC company Ltd.,</h2></div>
15         <p>Details of employees who perform well in projects </p>
16         <table class="table table-bordered table table-striped table-condensed table-hover">
17             <thead>
18                 <tr>
19                     <th>Firstname</th>
20                     <th>Lastname</th>
21                     <th>Email</th>
22                 </tr>
23             </thead>
24             <tbody>
25                 <tr>
26                     <td>John</td>
27                     <td>Doe</td>
28                     <td>john@mysite.com</td>
29                 </tr>
```

Bootstrap_well-Code Demo

Well Size

Small Well

Normal Well

Large Well



Web Technologies Code Demo E X

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

Run (Ctrl-F11)

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Example</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12
13 <div class="container">
14     <h2>Well Size</h2>
15     <div class="well well-sm">Small Well</div>
16     <div class="well">Normal Well</div>
17     <div class="well well-lg">Large Well</div>
18 </div>
19
20 </body>
21 </html>
22
```

Bootstrap_List - Code DemoExternal Video

about:blank

Basic List Group

First item

Second item

Third item

List Group With Badges

New

12

Deleted

5

Warnings

3

List Group With Linked Items

First item

Second item

Third item

Active Item in a List Group



File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Example</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12
13 <div class="container">
14     <h2>Basic List Group</h2>
15     <ul class="list-group">
16         <li class="list-group-item">First item</li>
17         <li class="list-group-item">Second item</li>
18         <li class="list-group-item">Third item</li>
19     </ul>
20 </div>
21
22
23
24 <div class="container">
25     <h2>List Group With Badges</h2>
26     <ul class="list-group">
27         <li class="list-group-item">New <span class="badge">12</span></li>
28         <li class="list-group-item">Deleted <span class="badge">5</span></li>
29         <li class="list-group-item">Warnings <span class="badge">3</span></li>
30     </ul>
```



File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
33
34
35<div class="container">
36    <h2>List Group With Linked Items</h2>
37<div class="list-group">
38    <a href="#" class="list-group-item">First item</a>
39    <a href="#" class="list-group-item">Second item</a>
40    <a href="#" class="list-group-item">Third item</a>
41</div>
42</div>
43
44
45
46
47<div class="container">
48    <h2>Active Item in a List Group</h2>
49<div class="list-group">
50    <a href="#" class="list-group-item active">First item</a>
51    <a href="#" class="list-group-item">Second item</a>
52    <a href="#" class="list-group-item">Third item</a>
53</div>
54</div>
55
56
57
58<div class="container">
59    <h2>List Group With a Disabled Item</h2>
60<div class="list-group">
61    <a href="#" class="list-group-item disabled">First item</a>
62    <a href="#" class="list-group-item">Second item</a>
```





about:blank

Second item

Third item

List Group With Badges

New

12

Deleted

5

Warnings

3

List Group With Linked Items

First item

Second item

Third item

Active Item in a List Group

First item

Second item

Third item



Web Technologies Code Demo | Bootstrap List Groups | Tryit Editor v3.6

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
44
45
46
47<div class="container">
48    <h2>Active Item in a List Group</h2>
49<div class="list-group">
50    <a href="#" class="list-group-item active">First item</a>
51    <a href="#" class="list-group-item">Second item</a>
52    <a href="#" class="list-group-item">Third item</a>
53</div>
54</div>
55
56
57
58<div class="container">
59    <h2>List Group With a Disabled Item</h2>
60<div class="list-group">
61    <a href="#" class="list-group-item disabled">First item</a>
62    <a href="#" class="list-group-item">Second item</a>
63    <a href="#" class="list-group-item">Third item</a>
64</div>
65</div>
66
67
68
69
70
71<div class="container">
```

about:blank

List Group With Linked Items

First item

Second item

Third item

Active Item in a List Group

First item

Second item

Third item

List Group With a Disabled Item

First item

Second item

Third item



File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
69
70
71<div class="container">
72  <h2>List Group With Contextual Classes</h2>
73<ul class="list-group">
74  <li class="list-group-item list-group-item-success">First item</li>
75  <li class="list-group-item list-group-item-info">Second item</li>
76  <li class="list-group-item list-group-item-warning">Third item</li>
77  <li class="list-group-item list-group-item-danger">Fourth item</li>
78</ul>
79
80
81
82 <h2>Linked Items With Contextual Classes</h2>
83 <p>Move the mouse over the linked items to see the hover effect:</p>
84<div class="list-group">
85  <a href="#" class="list-group-item list-group-item-success">First item</a>
86  <a href="#" class="list-group-item list-group-item-info">Second item</a>
87  <a href="#" class="list-group-item list-group-item-warning">Third item</a>
88  <a href="#" class="list-group-item list-group-item-danger">Fourth item</a>
89</div>
90</div>
91
92
93
94
95<div class="container">
96  <h2>List Group With Custom Content</h2>
97<div class="list-group">
98  <a href="#" class="list-group-item active">
```



Web Tech

Bootstrap Example - Google Chrome

about:blank

First item

Second item

Third item

List Group With a Disabled Item

```
56
57
58  <div>
59    <ul>
60      <li>First item</li>
61      <li>Second item</li>
62      <li>Third item</li>
63
64    </ul>
65  </div>
```

First item

Second item

Third item

List Group With Contextual Classes

```
66
67
68
69
70  <div>
71    <ul>
72      <li>First item</li>
73      <li>Second item</li>
74      <li>Third item</li>
75      <li>Fourth item</li>
76
77  </ul>
78</div>
```

First item

Second item

Third item

Fourth item

Linked Items With Contextual Classes

Move the mouse over the linked items to see the hover effect:

File List

CodeDerm

56

57

58 <div>

59

60 First item

61 Second item

62 Third item

63

64

65 </div>

66

67

68

69

70 <div>

71

72 First item

73 Second item

74 Third item

75 Fourth item

76

77

78

79</div>

80

81

82 <div>

83

84 First item

85

10:07 AM

2/21/2019

Web Technologies Code Demo | Bootstrap List Groups | Tryit Editor v3.6

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
86      <a href="#" class="list-group-item list-group-item-info">Second item</a>
87      <a href="#" class="list-group-item list-group-item-warning">Third item</a>
88      <a href="#" class="list-group-item list-group-item-danger">Fourth item</a>
89  </div>
90 </div>
91
92
93
94
95<div class="container">
96  <h2>List Group With Custom Content</h2>
97<div class="list-group">
98  <a href="#" class="list-group-item active">
99    <h4 class="list-group-item-heading">First List Group Item Heading</h4>
100   <p class="list-group-item-text">List Group Item Text</p>
101  </a>
102<a href="#" class="list-group-item">
103  <h4 class="list-group-item-heading">Second List Group Item Heading</h4>
104  <p class="list-group-item-text">List Group Item Text</p>
105 </a>
106<a href="#" class="list-group-item">
107  <h4 class="list-group-item-heading">Third List Group Item Heading</h4>
108  <p class="list-group-item-text">List Group Item Text</p>
109 </a>
110 </div>
111 </div>
112
113
114
115
```

Windows Taskbar: File Explorer, Google Chrome, File Manager, Microsoft Edge, Task View, Start button.

System tray: Volume icon, Network icon, Date/Time (10:08 AM), Date (2/21/2019).



i about:blank

First item

Second item

Third item

Fourth item

Linked Items With Contextual Classes

Move the mouse over the linked items to see the hover effect:

First item

Second item

Third item

Fourth item

List Group With Custom Content

First List Group Item Heading

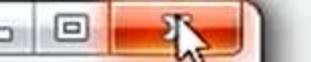
List Group Item Text

Second List Group Item Heading

List Group Item Text

Third List Group Item Heading

Bootstrap_Panel - Code Demo



about:blank

Basic Panel

A Basic Panel



File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Example</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12
13 <div class="container">
14     <h2>Basic Panel</h2>
15     <div class="panel panel-default">
16         <div class="panel-body">A Basic Panel</div>
17     </div>
18 </div>
19
20 <!--|
21
22 <div class="container">
23     <h2>Panel Heading</h2>
24     <div class="panel panel-default">
25         <div class="panel-heading">Panel Heading</div>
26         <div class="panel-body">Panel Content</div>
27     </div>
28 </div>
29
30
```



Basic Panel

A Basic Panel

Panel Heading

Panel Heading

Panel Content

Panel Footer

Panel Heading

Panel Content

Panel Footer

Web Technologies Code Demo E +

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

File List Save Compile & Run Evaluate Full screen Description

★ CodeDemo.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Example</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12 |     |
13 <div class="container">
14     <h2>Basic Panel</h2>
15     <div class="panel panel-default">
16         <div class="panel-body">A Basic Panel</div>
17     </div>
18 </div>
19
20
21
22 <div class="container">
23     <h2>Panel Heading</h2>
24     <div class="panel panel-default">
25         <div class="panel-heading">Panel Heading</div>
26         <div class="panel-body">Panel Content</div>
27     </div>
28 </div>
29
30
```

10:30 AM 2/21/2019

Panel Group

The panel-group class clears the bottom-margin. Try to remove the class and see what happens.

Panel Header

Panel Content

Panel Header



Panel Content

Panel Header

Panel Content

File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
38      <div class="panel-footer">Panel Footer</div>
39    </div>
40  </div>
41
42  -->
43
44<div class="container">
45  <h2>Panel Group</h2>
46  <p>The panel-group class clears the bottom-margin. Try to remove the class and see what happens.</p>
47<div class="panel-group">
48  <div class="panel panel-default">
49    <div class="panel-heading">Panel Header</div>
50    <div class="panel-body">Panel Content</div>
51  </div>
52  <div class="panel panel-default">
53    <div class="panel-heading">Panel Header</div>
54    <div class="panel-body">Panel Content</div>
55  </div>
56  <div class="panel panel-default">
57    <div class="panel-heading">Panel Header</div>
58    <div class="panel-body">Panel Content</div>
59  </div>
60  </div>
61</div>
62
63<!--
64<div class="container">
65  <h2>Panels with Contextual Classes</h2>
66  <div class="panel-group">
67    <div class="panel panel-default">
```



Panels with Contextual Classes

Panel with panel-default class

Panel Content

Panel with panel-primary class

Panel Content

Panel with panel-success class

Panel Content

Panel with panel-info class

Panel Content

Panel with panel-warning class

Panel Content

Panel with panel-danger class

Panel Content



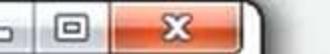
File List Save Compile & Run Evaluate Full screen Description

★ CodeDemo.html

```
62
63<div class="container">
64  <h2>Panels with Contextual Classes</h2>
65<div class="panel-group">
66  <div class="panel panel-default">
67    <div class="panel-heading">Panel with panel-default class</div>
68    <div class="panel-body">Panel Content</div>
69  </div>
70
71  <div class="panel panel-primary">
72    <div class="panel-heading">Panel with panel-primary class</div>
73    <div class="panel-body">Panel Content</div>
74  </div>
75
76  <div class="panel panel-success">
77    <div class="panel-heading">Panel with panel-success class</div>
78    <div class="panel-body">Panel Content</div>
79  </div>
80
81  <div class="panel panel-info">
82    <div class="panel-heading">Panel with panel-info class</div>
83    <div class="panel-body">Panel Content</div>
84  </div>
85
86  <div class="panel panel-warning">
87    <div class="panel-heading">Panel with panel-warning class</div>
88    <div class="panel-body">Panel Content</div>
89  </div>
90
91  <div class="panel panel-danger">
```



Bootstrap_Helper_Class - Code Demo



i about:blank

Close Icon Example



File List | Save | Compile & Run

Compile & Run

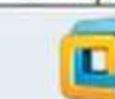
Evaluate

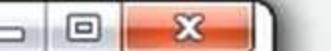
Full screen

Description

CodeDemo.html ▾

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Bootstrap Example</title>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12
13 <p>Close Icon Example
14 <button type = "button" class = "close" aria-hidden = "true">
15   &times; | [ ]<br>
16 </button>
17 </p>
18
19 <!--
20
21 <p>Caret Example<span class = "caret"></span></p>
22
23
24
25 <div class = "pull-left">Quick Float to left</div>
26 <div class = "pull-right">Quick Float to right</div>
27
28
29 <div class = "row">
30   <div class = "center-block" style = "width:200px; background-color:#ccc;">
```



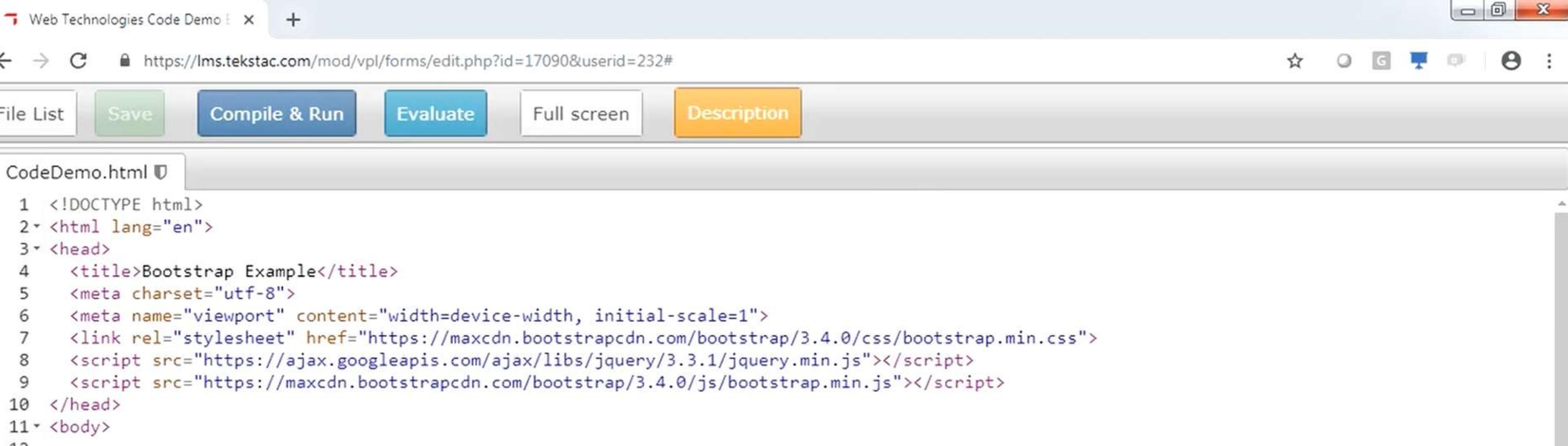


i about:blank

Close Icon Example

Caret Example





```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Example</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12
13 <p>Close Icon Example
14     <button type = "button" class = "close" aria-hidden = "true">
15         &times;
16     </button>
17 </p>
18
19
20
21 <p>Caret Example<span class = "caret"></span></p>
22
23
24 <!-- I
25 <div class = "pull-left">Quick Float to left</div>
26 <div class = "pull-right">Quick Float to right</div>
27
28
29 <div class = "row">
30     <div class = "center-block" style = "width:200px; background-color:#ccc;">
```





i about:blank

Close Icon Example

Caret Example ▾

Quick Float to left

Quick Float to right

File List **Save** Compile & Run Evaluate Full screen Description

★ CodeDemo.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Example</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12
13 <p>Close Icon Example
14     <button type = "button" class = "close" aria-hidden = "true">
15         &times;
16     </button>
17 </p>
18
19
20
21 <p>Caret Example<span class = "caret"></span></p>
22
23
24
25 <div class = "pull-left">Quick Float to left</div>
26 <div class = "pull-right">Quick Float to right</div>
27
28 <!--
29 <div class = "row">
30     <div class = "center-block" style = "width:200px; background-color:#ccc;">
```





about:blank

Close Icon Example

Caret Example ▾

Quick Float to left

This is an example for center-block

Quick Float to right



File List

Save

Compile & Run

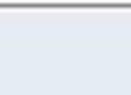
Evaluate

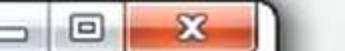
Full screen

Description

★ CodeDemo.html

```
7   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12
13 <p>Close Icon Example
14 <button type = "button" class = "close" aria-hidden = "true">
15     &times;
16 </button>
17 </p>
18
19
20
21 <p>Caret Example<span class = "caret"></span></p>
22
23
24
25 <div class = "pull-left">Quick Float to left</div>
26 <div class = "pull-right">Quick Float to right</div>
27
28
29 <div class = "row">
30     <div class = "center-block" style = "width:200px; background-color:#ccc;">
31         This is an example for center-block
32     </div>
33 </div>
34
35
36
```





i about:blank

Close Icon Example

Caret Example ▾

Quick Float to left

This is an example for center-block

Quick Float to right

This is an example for show class



File List Save Compile & Run Evaluate Full screen Description

★ CodeDemo.html

```
27
28
29 <div class = "row">
30     <div class = "center-block" style = "width:200px; background-color:#ccc;">
31         This is an example for center-block
32     </div>
33 </div>
34
35
36
37 <!--You can force an element to be shown or hidden (including for screen readers) with the use of classes .show and .hidden. -->
38
39 <div class = "row" style = "padding: 91px 100px 19px 50px;">
40
41     <div class = "show" style = "left-margin:10px; width:300px; background-color:#ccc;">
42         This is an example for show class
43     </div>
44
45     <div class = "hidden" style = "width:200px; background-color:#ccc;">
46         This is an example for hide class
47     </div>
48
49 </div>
50
51
52     I
53 <!--You can hide an element to all devices except screen readers with the class .sr-only.-->
54 <!--
55 <div class = "row" style = "padding: 91px 100px 19px 50px;">
56     <form class = "form-inline" role = "form">
```





i about:blank

Close Icon Example

Caret Example ▾

Quick Float to left

This is an example for center-block

Quick Float to right

This is an example for show class

Enter email



Password



Web Technologies Code Demo +

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
47     </div>
48
49 </div>
50
51
52
53 <!--You can hide an element to all devices except screen readers with the class .sr-only.-->
54
55 <div class = "row" style = "padding: 91px 100px 19px 50px;">
56     <form class = "form-inline" role = "form">
57
58     <div class = "form-group">
59         <label class = "sr-only" for = "email">Email address</label>
60         <input type = "email" class = "form-control" placeholder = "Enter email">
61     </div>
62
63     <div class = "form-group">
64         <label class = "sr-only" for = "pass">Password</label>
65         <input type = "password" class = "form-control" placeholder = "Password">
66     </div>
67
68     </form>
69 </div>
70
71 |
72
73
74 </body>
```

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

10:50 AM 2/21/2019

Bootstrap_Responsive - Code Demo



about:blank

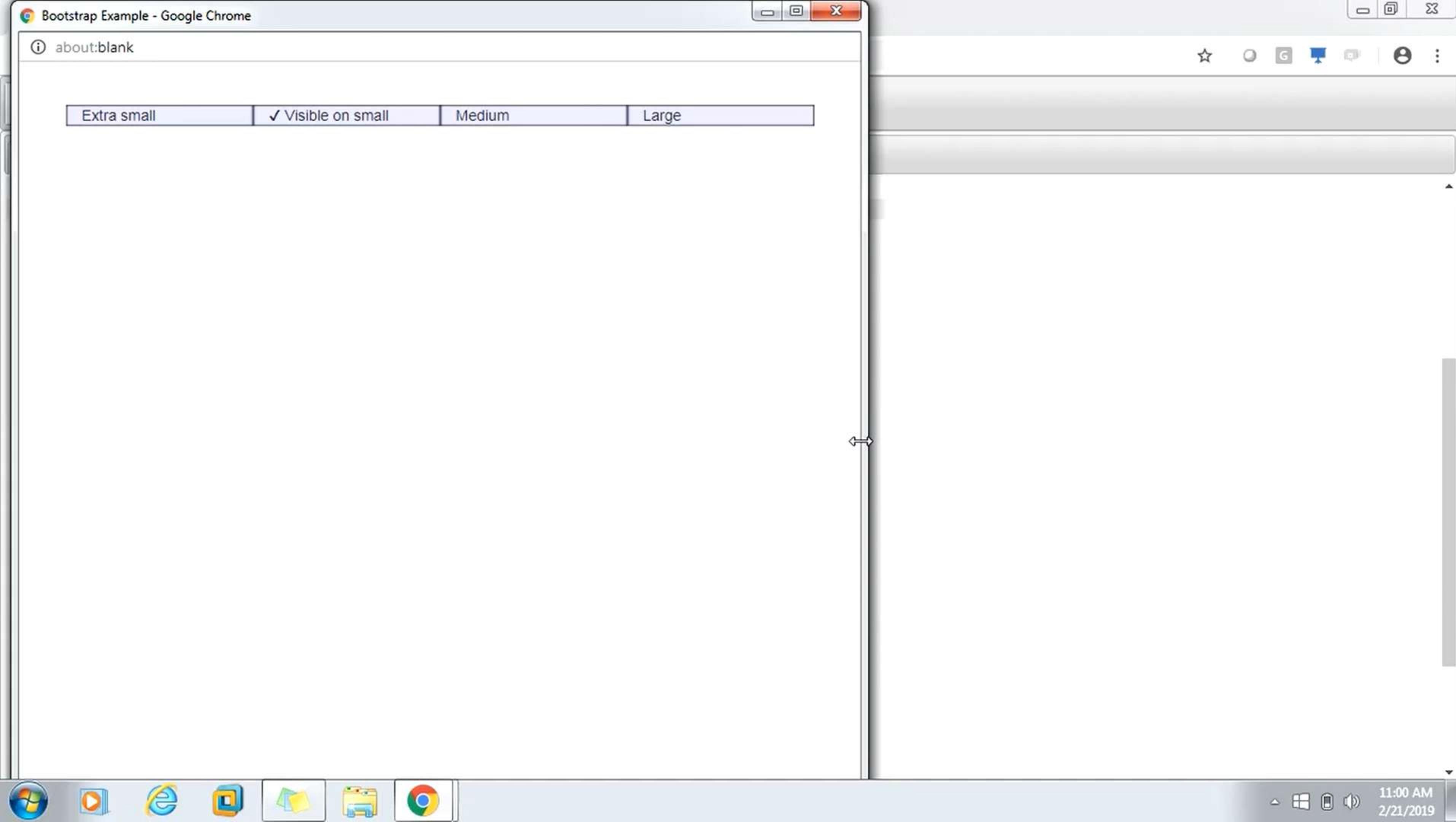
Extra small

Small

Medium

✓ Visible on large





about:blank

✓ Visible on x-small	Small
Medium	Large

17090&userid=232#

Full screen

Description

```
"background-color: #dedef8;
inset -1px 1px 1px #444;">

</span>
on x-small</span>

"background-color: #dedef8;
inset -1px 1px 1px #444;">

>
on small</span>
<-->
>

"background-color: #dedef8;
inset -1px 1px 1px #444;">

n>
on medium</span>

"background-color: #dedef8;
inset -1px 1px 1px #444;">

>
on large</span>
```



Web Technologies Code Demo +

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Bootstrap Example</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
9      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
10 </head>
11 <body>
12
13 <!-- Responsive utilities are currently only available for block and table toggling. -->
14
15 <div class = "container" style = "padding: 40px;">
16     <div class = "row visible-on">
17
18     <div class = "col-xs-6 col-sm-3" style = "background-color: #dedef8;
19         box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
20
21         <span class = "hidden-xs">Extra small</span>
22         <span class = "visible-xs">✓ Visible on x-small</span>
23     </div>
24
25     <div class = "col-xs-6 col-sm-3" style = "background-color: #dedef8;
26         box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
27
28         <span class = "hidden-sm">Small</span>
29         <span class = "visible-sm">✓ Visible on small</span>
30     </div>
```

Windows Taskbar: File Explorer, Google Chrome, Microsoft Edge, File Manager, Task View, Start button, Volume, Network, Battery, Date/Time (10:59 AM, 2/21/2019)

Web Technologies Code Demo +

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
21      <span class = "hidden-xs">Extra small</span>
22      <span class = "visible-xs">✓ Visible on x-small</span>
23  </div>
24
25  <div class = "col-xs-6 col-sm-3" style = "background-color: #dedef8;
26    box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
27
28      <span class = "hidden-sm">Small</span>
29      <span class = "visible-sm">✓ Visible on small</span>
30  </div>
31
32  <div class = "clearfix visible-xs"></div>
33
34  <div class = "col-xs-6 col-sm-3" style = "background-color: #dedef8;
35    box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
36
37      <span class = "hidden-md">Medium</span>
38      <span class = "visible-md">✓ Visible on medium</span>
39  </div>
40
41  <div class = "col-xs-6 col-sm-3" style = "background-color: #dedef8;
42    box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
43
44      <span class = "hidden-lg">Large</span>
45      <span class = "visible-lg">✓ Visible on large</span>
46  </div>
47
48  </div>
49 </div>
50
```

11:00 AM 2/21/2019

Bootstrap_Inline_forms - Code Demo

Inline Forms

 Back to course

0. General 

1. HTML 

2. CSS 

3. Java Script 

4. jQuery 

5. CSS 3 With Bootstrap 

 Pre-Quiz 

 CSS Bootstrap 

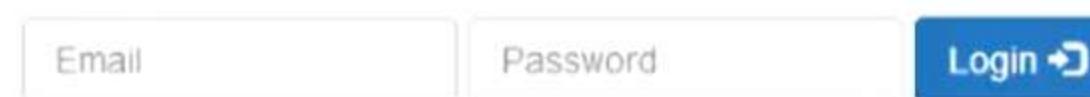
 Inline Forms 

 Bootstrap's Navigati... 

Create a Bootstrap compact form layout in which form controls are placed side-by-side (in-line). The form should contain elegant form fields with Bootstrap.

The snapshot of the web page is given below:

Image 1:



A screenshot of a web page showing an inline form. It consists of two input fields side-by-side: one for 'Email' and one for 'Password'. To the right of these fields is a blue 'Login' button with a white arrow icon.

Image 2:



A screenshot of a web page showing an inline form with validation. The 'Email' input field is highlighted with a blue border. A yellow callout box points to this field with the text 'Please fill out this field.' The 'Password' field is shown with placeholder dots ('.....'). To the right is a blue 'Login' button with a white arrow icon.

Hints:

1. Create a div belongs to the compact class which should contain all required elements like form, input, button and other div tags.
2. Choose appropriate bootstrap class for the form tag to get the form layout mentioned above.
3. Create an email filed with id as "inputEmail" and set the place holder as specified in the given figures.
4. Create a password filed with id as "inputPassword" and set the place holder as specified in the given figure.

[Back to course](#)

0. General >

1. HTML >

2. CSS >

3. Java Script >

4. jQuery >

5. CSS 3 With Bootstrap

Pre-Quiz

CSS Bootstrap

Inline Forms

Bootstrap's Navigati...

Image 2:

The image shows a simple login interface. It consists of two input fields: one for 'Email' and one for a password represented by a series of dots. Below these fields is a blue 'Login' button with a white right-pointing arrow icon. A yellow callout box with an exclamation mark icon is positioned over the 'Email' input field, containing the text 'Please fill out this field.'

Hints:

1. Create a div belongs to the compact class which should contain all required elements likeform, input, button and other div tags.
2. Choose appropriate bootstrap class for the form tag to get the form layout mentioned above.
3. Create an email filed with id as "inputEmail" and set the place holder as specified in the given figures.
4. Create a password filed with id as "**inputPassword**" and set the place holder as specified in the given figure.
5. All the input fields should be required fields and should belongs to the bootstrap form control class
6. The 'Login' button should be a bootstrap primary submit button created using button tag and also set the log-in icon of Bootstrap Glyphicon components using a span tag.

Requested files

[Back to course](#)

0. General >

1. HTML >

2. CSS >

3. Java Script >

4. jQuery >

5. CSS 3 With Bootstrap >

 Pre-Quiz > CSS Bootstrap >

</> Inline Forms >

</> Bootstrap's Navigati... >

[File List](#)[Save](#)[Compile & Run](#)[Evaluate](#)[Full screen](#)[Description](#)inline.html 

```
4
5  <!DOCTYPE html>
6  <html lang="en">
7  <head>
8    <meta charset="utf-8">
9    <meta http-equiv="X-UA-Compatible" content="IE=edge">
10   <meta name="viewport" content="width=device-width, initial-scale=1">
11   <title>Example of Bootstrap 3 Inline Form Layout</title>
12
13  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
14  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
15  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
16
17
18  <style type="text/css">
19    .compact{
20      margin: 20px;
21    }
22  </style>
23
24  </head>
25  <body>
26
27  <!-- WRITE YOUR CODE HERE -->
28
```

[Back to course](#)

0. General >

1. HTML >

2. CSS >

3. Java Script >

4. jQuery >

5. CSS 3 With Bootstrap

Pre-Quiz

CSS Bootstrap

</> Inline Forms

</> Bootstrap's Navigati...

Web Technology

[Dashboard](#) / Courses / Web Technology_2 / CSS 3 With Bootstrap / Inline Forms[--> Previous Activity](#)[Next Activity -->](#)[Description](#)[Submission](#)[Edit](#)[Grading view](#)[File List](#)[Save](#)[Compile & Run](#)[Evaluate](#)[Full screen](#)[Description](#)★ inline.html 

```
22  </body>
23
24  </head>
25  <body>
26
27  <!-- WRITE YOUR CODE HERE -->
28
29  <div class="compact">
30    <form class="form-inline" action="#">
31      <input type="email" class="form-control" id="inputEmail" placeholder="Email" required>
32
33      <input type="password" class="form-control" id="inputPassword" placeholder="Password" required>
34
35      <button id="login" type="submit" class="btn btn-primary">Login <span class="glyphicon glyphicon-log-in"></span></button>
36    </form>
37    <br>
38  </div>
39
40
```

JAVA SCRIPT



In this module you will learn

- Introduction to scripting Language
- Javascript - Introduction
- Execution of Javascript
- Scripts in head and body of HTML
- Functions in Javascript
- Internal and External Javascript
- Variables, Datatypes, Operators
- Programming Constructs in JavaScript
- Built-in methods in Javascript
- Javascript Statements, Block, Comments



Introduction to Scripting Language



Computer language with a series of commands within a file.

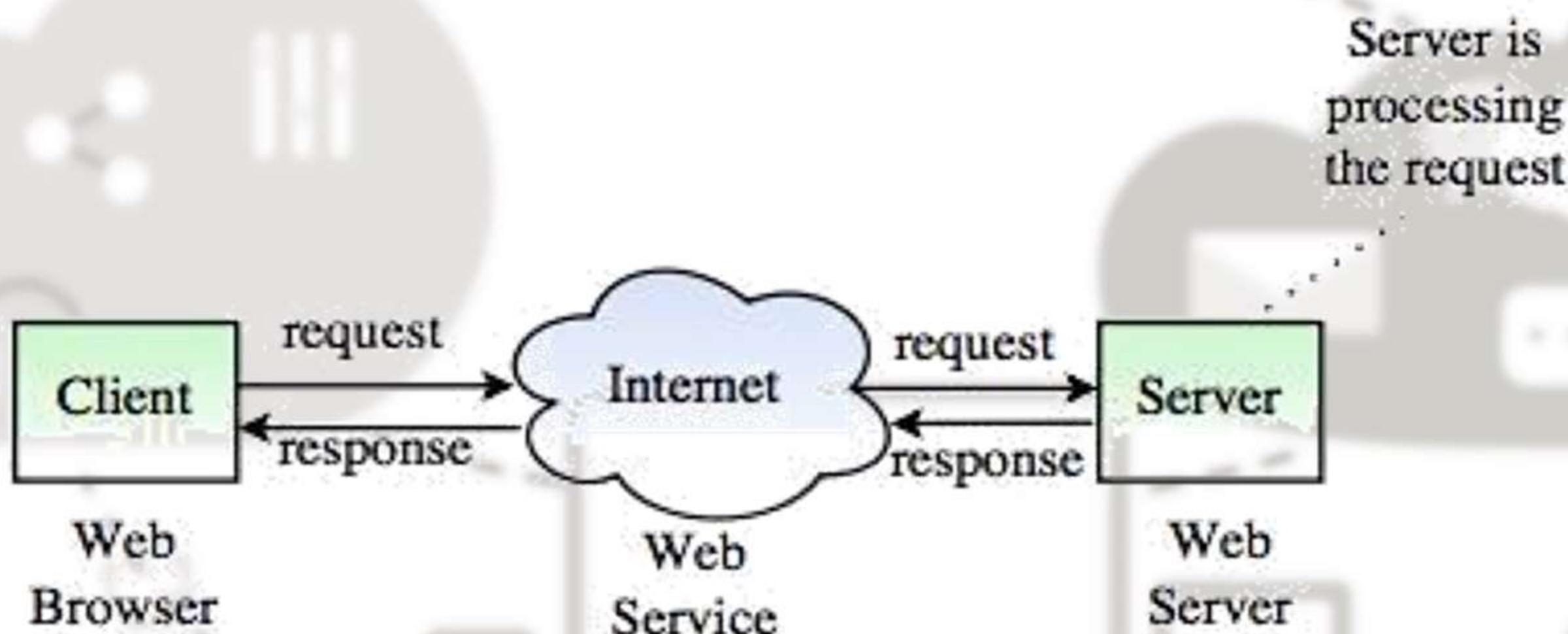
Capable of being executed without being compiled.

Script provides changes to the webpage.

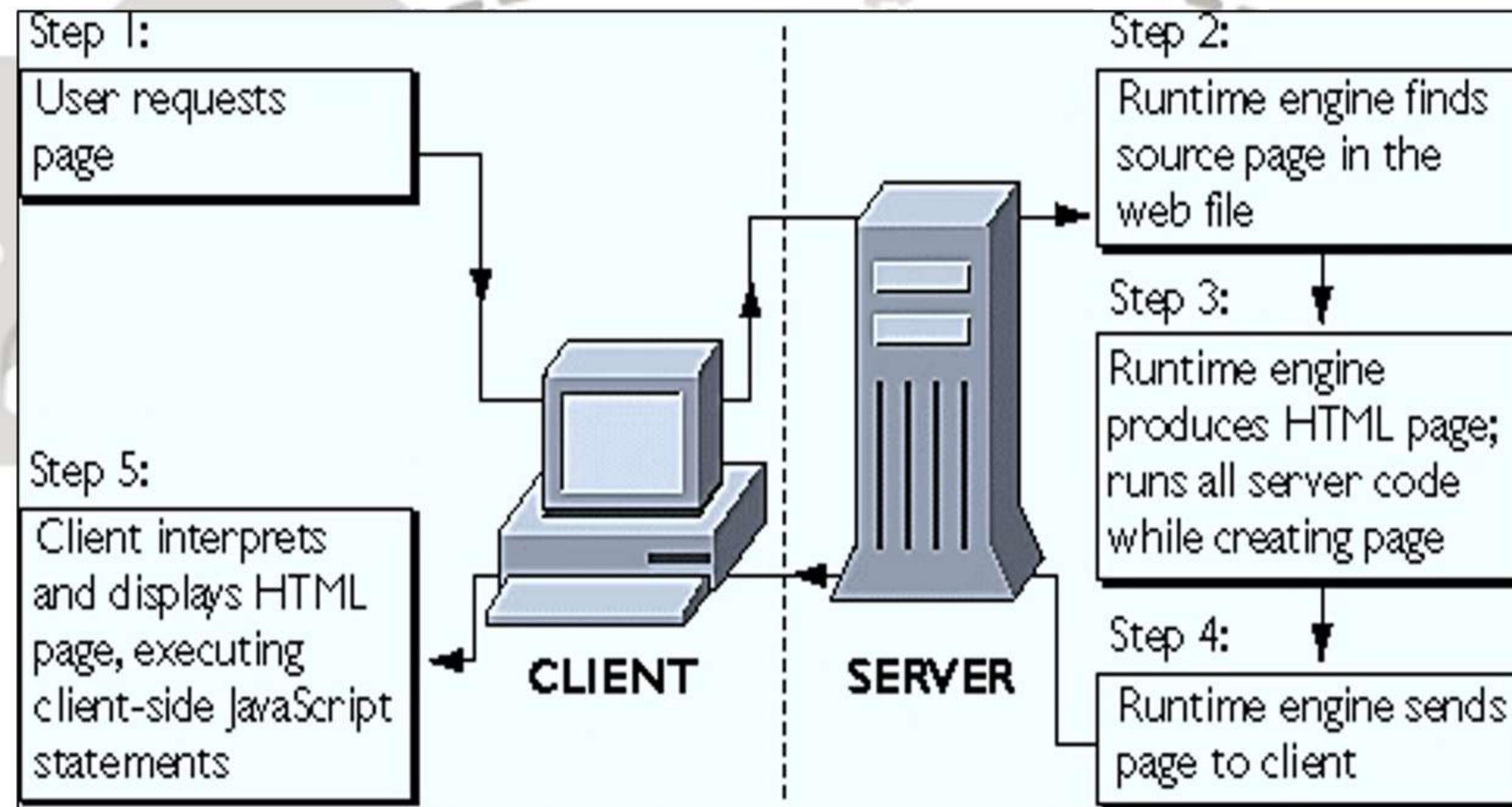
Two types of Scripting

- Server side scripting
- Client side scripting

Server Side Scripting



Client Side Scripting



JavaScript Introduction

- JavaScript can be used in **client and server side**.
- Traditionally on **client side**
 - processing user input
 - validations
- Latest Implementation
 - **Client-side/Single page application**
- On **Server-Side** using **Node JS**
 - It can create standalone application, Network application, Web Application
 - **REST services**, service layers
- From traditional validation module to
 - Javascript libraries like **JQuery**, **DOJO**, **React** and so on
 - Javascript frameworks **like Angular**, **Ember** and so on

Advantages of JavaScript

Improves transaction response time

Reduces server load

Less server interaction – Can validate user input before sending the page off to the server.

Provides immediate feedback

Embedding JavaScript in HTML



JavaScript is included in an HTML file with the help of `<script>` tag

There are two methods to embed javascript in to **HTML code**.

Internal Script

External Script

```
<script type="text/javascript">
    document.write("<h1>Hello World!</h1>");
</script>
```

```
<head>
<script type="text/Javascript">
    // Javascript Code
</script>
</head>
```

Inline JavaScript

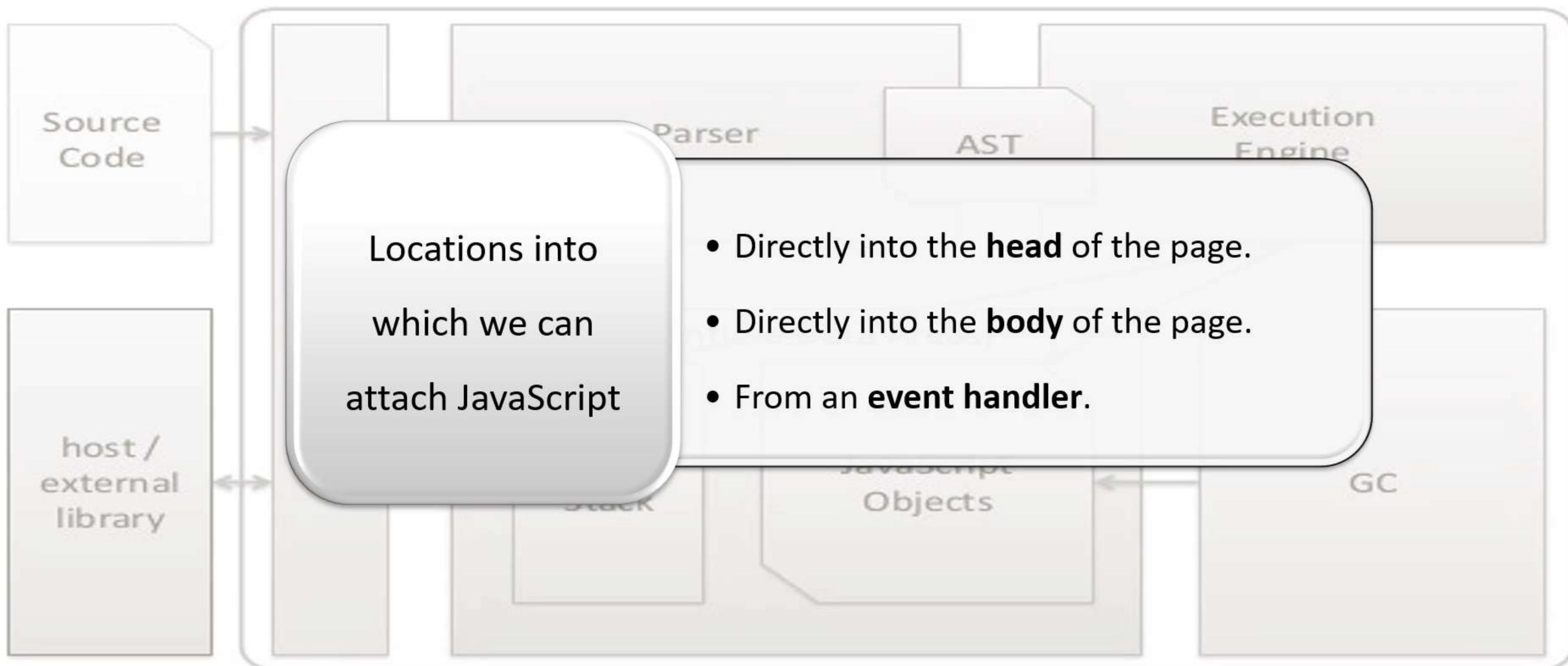
```
<body>
<script type="text/Javascript">
    // JavaScript Code
</script>
</body >
```

External JavaScript

- **JavaScript can be put in a separate .js file**
 - Includes the external java script file in a HTML file using the code

```
<script src="myJavaScriptFile.js">
</script>
```
 - An **external .js** file can be used in multiple HTML pages wherever necessary
 - The **external .js** file **cannot** itself **contain a <script> tag**
- **JavaScript can be put in an HTML form object, such as a button**
 - This JavaScript will be executed when the form object is used

Execution of JavaScript



Data Types



Javascript Datatypes

Primitive Type

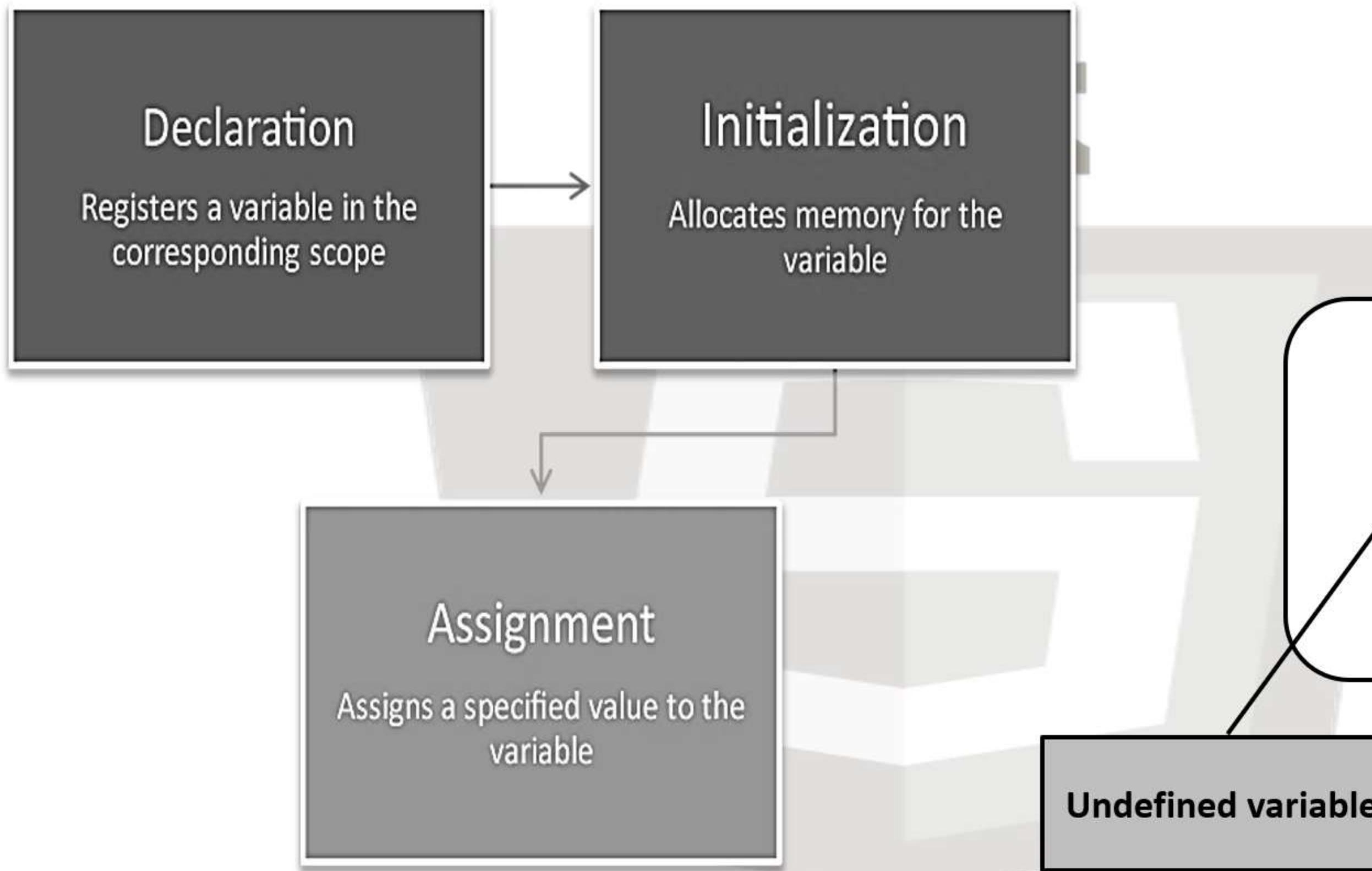
1. String
2. Number
3. Boolean
4. Undefined
5. Null

Reference Type

1. Array
2. Object
3. Function
4. Date
5. Regex

```
var age = 16; // Number  
var name = "John"; // String  
var ids = [101, 102, 103]; // Array  
var x = {id:101, name:"John", salary:40000}; //Object
```

JavaScript Variables

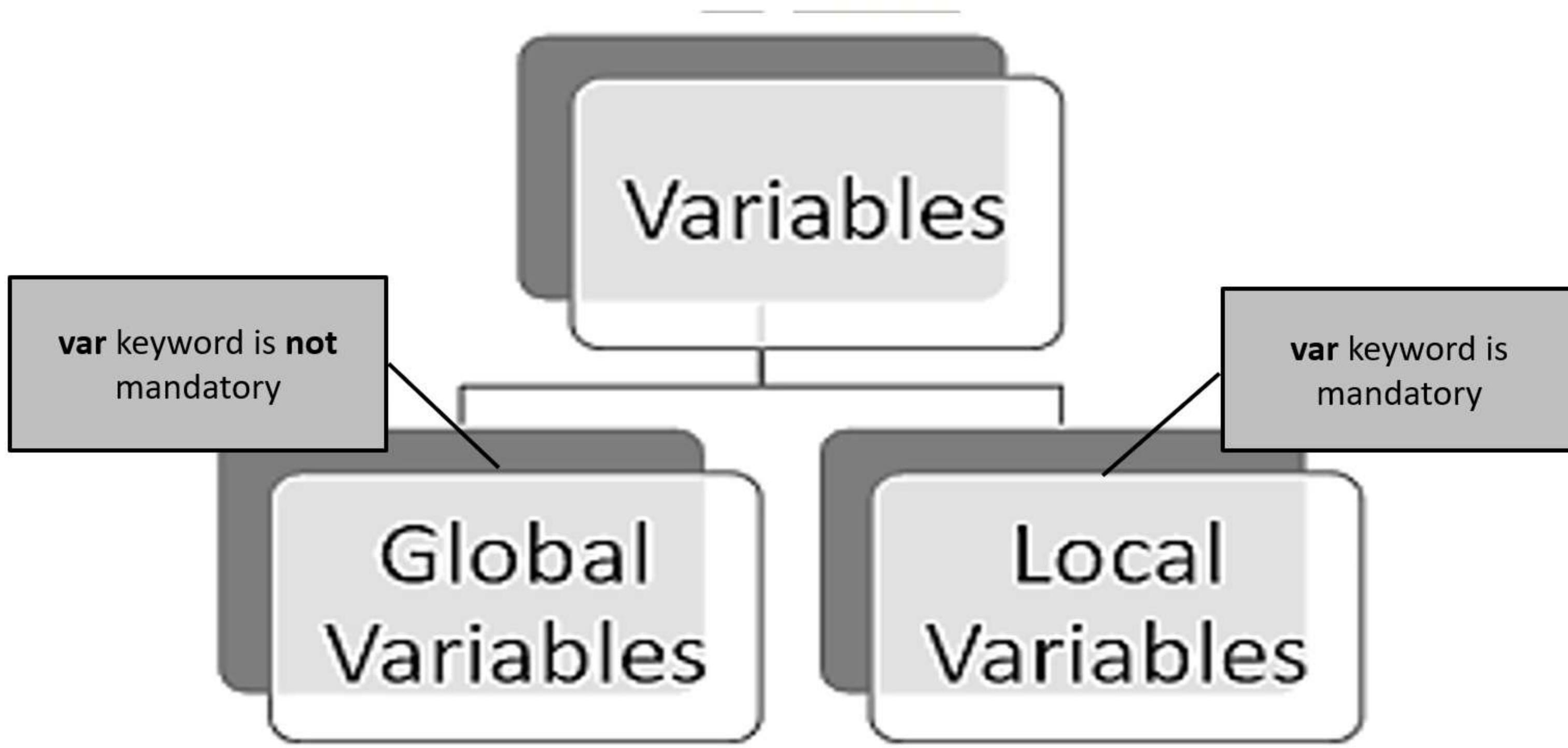


Example

```
var name="Johan";  
var id , salary;  
var age = 18;  
age="eighteen";
```

Undefined variable

Scope of variables



Scope of variables

```
<body>
<p id="demo"></p>
<script>
    myCar();
    document.getElementById("demo").innerHTML = "My Car Name: " + carName;
    function myCar()
    {
        carName = "BMW";
    }
</script>
</body>
```

var a = 20; → Global Variable

function checkVariable(){ → Local Variable

var a = 23;

}

Operators

Arithmetic Operators

Operator	Description	Examples
+	Addition	<code>var z = 5 + 2;</code>
-	Subtraction	<code>var z = 5 - 2;</code>
*	Multiplication	<code>var z = 5 * 2;</code>
/	Division	<code>var z = 5 / 2;</code>
%	Modulus	<code>var z = 5 % 2;</code>
++	Increment	<code>var x = 5; z = x++;</code>
--	Decrement	<code>var x = 5; z = x--;</code>

Assignment Operators

Operator	Description	Examples
=	$x = y$	<code>x = y</code>
+=	$x += y$	<code>x = x + y</code>
-=	$x -= y$	<code>x = x - y</code>
*=	$x *= y$	<code>x = x * y</code>
/=	$x /= y$	<code>x = x / y</code>
%=	$x %= y$	<code>x = x % y</code>
=	$x = y$	<code>x = y</code>

Operators contd..

Relational and Logical Operators

Operator	Description	Comparing	Returns
		$x > 8$	false
		$x < 8$	true
		$x \geq 8$	true
		$x \leq 8$	false
		$x == 8$	true
		$x != 8$	false

Relational and Logical Operators

Operator	Description	Comparing	Returns
>	greater than	$x > 8$	false
<	less than	$x < 8$	true
\geq	greater than or equal to	$x \geq 8$	true
\leq	less than or equal to	$x \leq 8$	false
		$x == 8$	true
		$x != 8$	false

Conditional Operators

Operator	Description	Example
$&&$	and	$? : $ (Conditional)
$ $	or	If Condition is true? Then value X : Otherwise value Y
!	not	$var x=10 , y=20;$ $var z=(x<y)?x:y;$

$!(x == y)$ true

Operators contd..

typeof Operator

- The typeof operator is a unary operator
- It returns String

Example

```
var data=10;  
  
var result = typeof data;  
  
document.write(result);
```

Type	Return data
Number	"number"
String	"string"
Boolean	"boolean"
Object	"object"

Comments in JavaScript

Comments are used to provide additional information about the JavaScript code, and make it more readable.

These statements will not get executed by the browser.

Single line comment

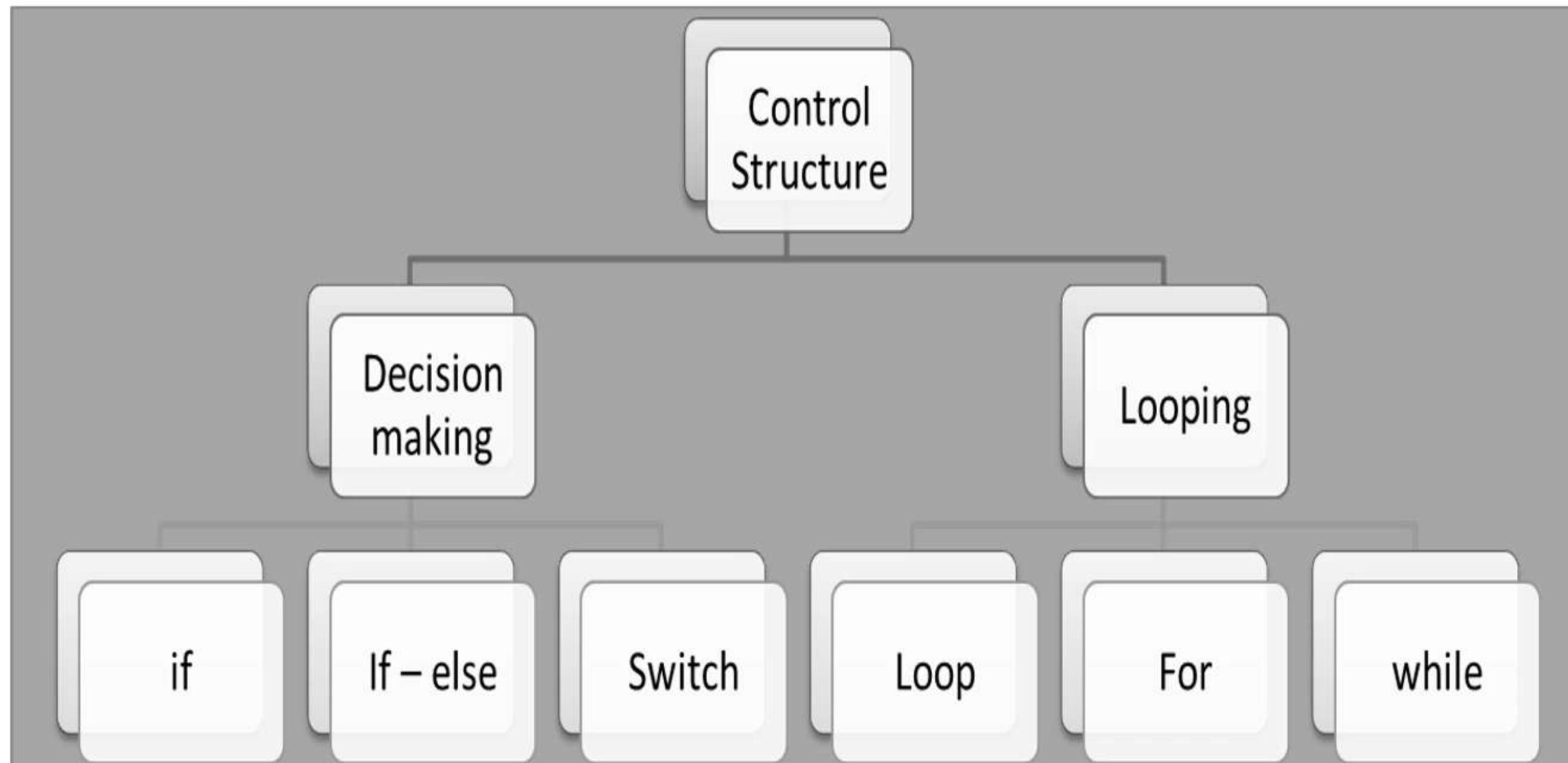
- **// Statements**

Multi line comment

- **/* Statements */**

Programming Constructs in JavaScript

Decides the flow of execution of the JavaScript program.



if Statement

Syntax

```
if (condition)
{
    //Statement(s) to be executed
    if the condition is true
}
```

Example

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

    var num=10;

    if (num % 2 ==0)

    {
        document.write("num is Even");
    }

</script>
```

If..else statement

Syntax

```
if (condition)
{
    //Statement(s) to be executed if the
    condition is true
}
else
{
    //Statement(s) to be executed if the
    condition is false
}
```

Example

```
<script type="text/javascript">
    var num=10;
    if (num % 2 ==0)
    {
        document.write("num is Even");
    }
    else
    {
        document.write("num is Odd");
    }
</script>
```

If..else if..else statement

Syntax

```
if (condition1)
{
    //Statement(s) to be executed if
    the condition1 is true
}
else if(condition2)
{
    //Statement(s) to be executed if
    the condition2 is false
}
.....
else
{
    //statement to be executed
}
```

Example

```
<script type="text/javascript">
    var num=10;
    if (num > 0)
    {
        document.write("num is Positive ");
    }
    else if(num < 0)
    {
        document.write("num is Negative
    ");
    }
    else
    {
        document.write("num is equal to
zero ");
    }
</script>
```

Switch statement

Syntax

```
switch(expression)
{
    case n:
        code block
        break;
    case n:
        code block
        break;
    default:
        code block
}
```

Example

```
<script type="text/javascript">
var exp = 0;
switch(exp)
{
    case 1 :
        document.write("the value is positive");
        break;
    case -1 :
        document.write("the value is negative");
        break;
    default :
        document.write("the value is zero");
}
</script>
```

for Loop

Syntax

```
<script type="text/javascript">
    var initvalue;
    for(initvalue=startvalue;initvalue condition;inc/decvalue)
    {
        //Statements to be executed
    }
</script>
```

Example

```
<script type="text / javascript">
    var sum=0;
    for(var i=1 ; i<=5 ; i++)
    {
        sum=sum+i;
    }
    document.write("The sum is : "+sum);
</script>
```

while Loop

```
while (expression)
{
    Statement(s) to be executed
    if expression is true
}
```

Syntax

Example

```
<script type="text/javascript">
    var sum=0;
    var i=1;
    while(i<=5)
    {
        sum=sum+i;
        i++;
    }
    document.write("the sum is : "+sum);
</script>
```

do – while Loop

```
do
{
    Statement(s)
        to be executed;
} while (expression);
```

Syntax

E X A M

Example

```
<script type="text/javascript">
    var sum=0;
    var i=1;
    do
    {
        sum=sum+i;
        i++;
    }while(i<=5);
    document.write("the sum is :" +sum);
</script>
```

for .. in loop

```
for (variablename in object)
{
    statement or block to execute
}
```

Syntax

Example

```
<script type="text/javascript">
    var ids = [101,102,103];
    var data;
    for(data in ids)
    {
        document.write(ids[data]+ " ");
    }
</script>
```

Built-in Functions



Array Methods

String Methods

Boolean Methods

Math Methods

Number Methods

RegExp Methods

Date Methods

Date Static Methods

String HTML wrappers

Built-in Functions

Function	Description	Function	Description
isNaN	Determines whether value is a legal number or not.	parseInt	Converts string value to integer.
isFinite	To find whether a number is a finite legal number.	parseFloat	Converts string value to floating point number.
eval	Executes JavaScript source code.	escape	Encodes the string value into world wide acceptable format.
number	Converts object to the corresponding number value.	encodeURI	To encode URI.
string	Converts object to the corresponding string value.	decodeURI	To decode URI.
		encodeURIComponent	To encode URI component.
		decodeURIComponent	To decode URI component.

Built-in Functions

```
document.write(isNaN(0));
```

```
var obj2=new Boolean(0);  
document.write(String(obj2));
```

```
document.write(escape("this is  
javascript escape function!!"));
```

```
document.write(encodeURI("http://www.t  
echstrikers.com/test.php?id=23&str=this is  
test"));
```

```
isFinite("5678");  
isFinite("isFinite");  
isFinite("5678-34");
```

```
var obj1=new String("7893");  
document.write(parseInt(obj1));
```

Example

```
function functionname(parameter-list)
{
    statements
}
```

Syntax

Example

```
<script type="text/javascript">
function display()
{
    document.write("Welcome to
        JavaScript");
}
display();
</script>
```

Function
call

JavaScript Functions



Functions with Parameters

```
<script type="text/javascript">
function add(no1 , no2)
{
    var sum = no1 + no2;
    document.write(sum);
}
add(10,20);
</script>
```

Function with return data

```
<script type="text/javascript">
function add(no1 , no2)
{
    var sum = no1 + no2;
    return sum;
}
var sum =add(10,20);
document.write(sum);
</script>
```

Summary

- Introduction to scripting Language
- Javascript - Introduction
- Execution of Javascript
- Scripts in head and body of HTML
- Functions in Javascript
- Internal and External Javascript
- Variables, Datatypes, Operators
- Programming Constructs in JavaScript
- Built in methods in Javascript
- Javascript Statements, Block, Comments



JAVA SCRIPT



In this module you will learn

- Events
- Javascript event handling
- Java script validation
- Working with Form Object (Form elements properties, methods and events)



Event Handling



- Event – an action that is fired (initiated) within a webpage.
- JavaScript is Single Thread.
- It is so useful in creating interactive web sites.
- JavaScript uses asynchronous callback.
- Simplest way to run .js code in response to an event is to use an event handler (function)

Event Handling

Example :



```
<form>
  <input type="button" name="test" value="Click me"
         onclick="inform()">
</form>
```

```
<script>
function inform()
{
  alert("You have activated me by clicking the grey
        button!")
}
</script>
```

When the user
clicks the
button,
“inform()” will
be called.

JavaScript Validation



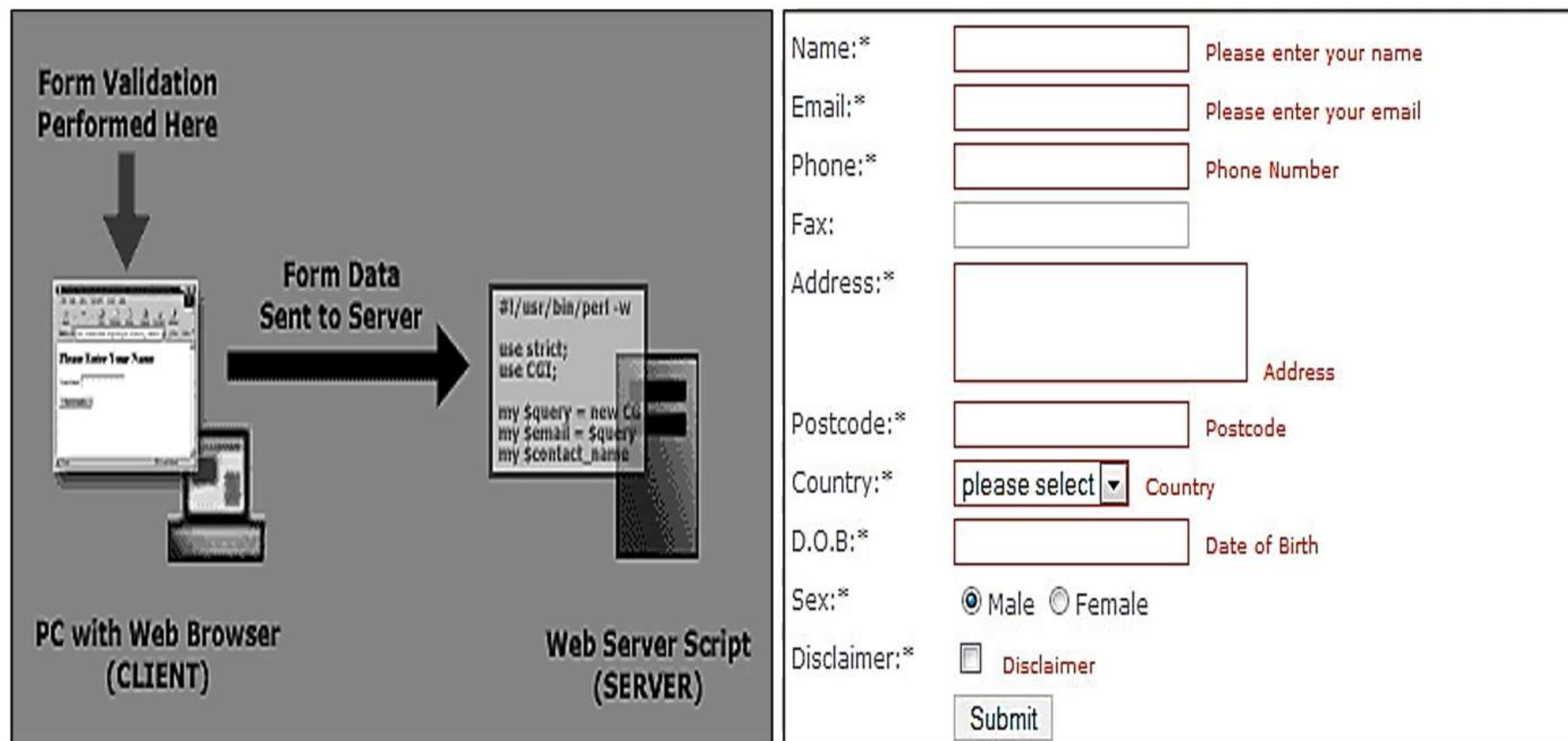
JavaScript data validation happens before form is submitted.

Server-side application validation happens after the form is submitted to the application server.

Form validation performs the following functions :

- Basic Validation
- Data Format Validation

JavaScript Validation



JavaScript Validation



```
<form name="register" action="#" method="post">
First Name <input type="text" name="fname" > <br/>
Last Name <input type="text" name="lname" > <br/>
<input type="button" value="Register" onclick="validateData()">
<br/>
</form>
```

```
var fname=document.register.fname.value;
var lname=document.register.lname.value;

if(fname==null || fname== "" || lname==null ||
lname.trim()=="")
{
    document.getElementById("msg").innerHTML += "Enter value
for name <br />";
}
```

JavaScript Validation



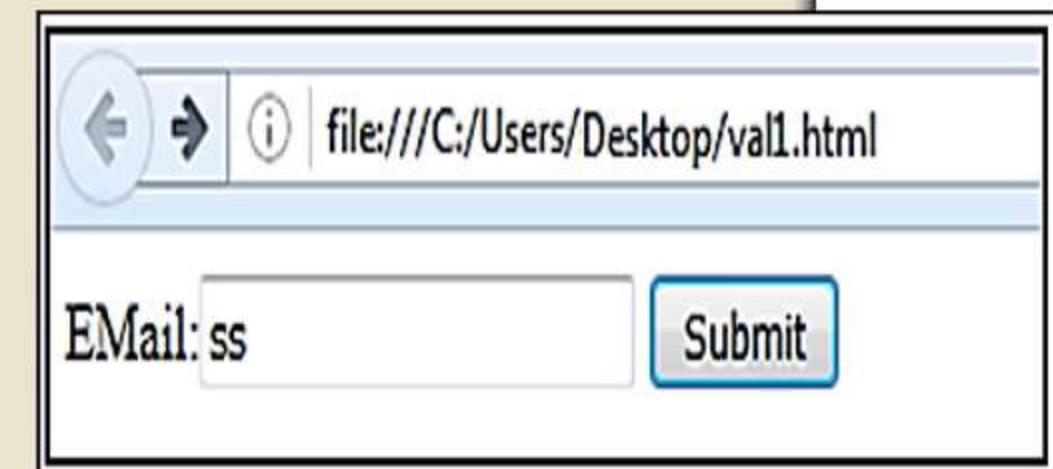
The data entered in a form can be validated for its format.

Our code must include appropriate logic to test the correctness of data.

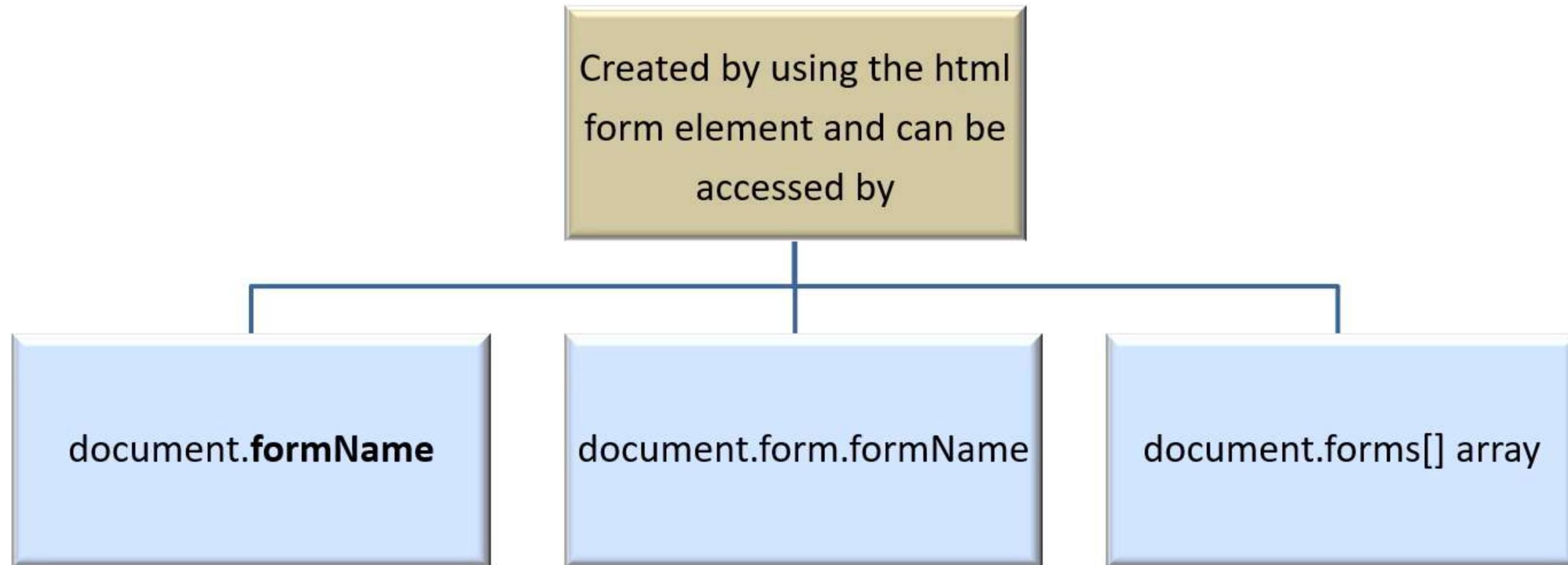
```
<body>
<form name="myForm" action="welcomePage.html" method="post" >
Email:<input type="text" name="email" id="email" onBlur="return validateEmail()">
<input type="submit" value="Submit" onSubmit="return validateEmail()">
</form>
```

JavaScript Validation

```
<script type="text/javascript">
    function validateEmail()
    {
        var emailID = document.myForm.EMail.value;
        atpos = emailID.indexOf("@");
        dotpos = emailID.lastIndexOf(".");
        if(atpos < 1 || ( dotpos - atpos < 2 ))
        {
            alert("Please enter correct email ID");
            document.myForm.EMail.value="";
            document.getElementById("email").focus();
            return false;
        }
        return true ;
    }
</script>
</body>
```



Form Object



Form Object

Property	Description
action	Presents the action attribute.
autocomplete	Presents the autocomplete attribute (on / off)
encoding	Presents the enctype attribute.
length	Presents the number of elements on a form.
method	Presents the forms method attribute.
name	Presents the name attribute of the form.
noValidate	Presents if form data needs to be validated or not (true / false)
target	Presents the target attribute of the form. It represents the name of the frame or window to which the form submission response is sent by the server.

Form Object

```
<form id="myForm" action="homepage.html">
<table>
<tr><td>User name </td><td><input type="text" name="uname"></td></tr>
<tr><td>Password</td><td> <input type="password" name="pwd"></td></tr>
<tr><td colspan="2"><input type="button" value="Submit" onClick = "changeAction()" >
</td> </tr>
</table> </form>
<div id="msg" ></div>
<script>
    function changeAction()
    {
        document.getElementById("myForm").action = "form_action.asp";
        document.getElementById("myForm").autocomplete = "off";
        document.getElementById("msg").innerHTML = "The value of the action
            attribute was changed";
    }
</script>
```

Form Object - Methods

```
<body>
<form name="register" action="registerUser.html">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="button" onclick="myFunction()" value="Submit form">
</form>
<script>
function myFunction()
{
  document.register.submit();
}
</script>
</body>
```

Form accessed as
`document.formName`
and form submission
invoked using `submit()`
method

Form Event Handler

```
<body>

<form name="register" onresetonsubmit="return
displayValues()">
    First name: <input type="text" name="fname" onblur = "alert(this.value)"><br>
    Last name: <input type="text" name="lname"><br>
    <input type="submit" >
    <input type="reset" >
</form>

<script>

function display()
{
    document.register.fname.value="Pearson";
    document.register.lname.value="David";
    return false;
}


```

Form Events onset,
onreset and onblur
used

Accessing a textbox
value from a form

Form Event Handler

Executed when onsubmit event occurs

```
function displayValues()
{
    var fname=document.register.fname.value;
    var lname=document.register.lname.value;
    alert("First name is "+fname+" Last name is
"+lname);
    return false;
}
</script>
</body>
```

Text Object

Text object represents a single-line text input field in a HTML form object.

```
<input type="text" name="firstname">
```

Properties

- defaultValue
- form
- name
- type
- value

Methods

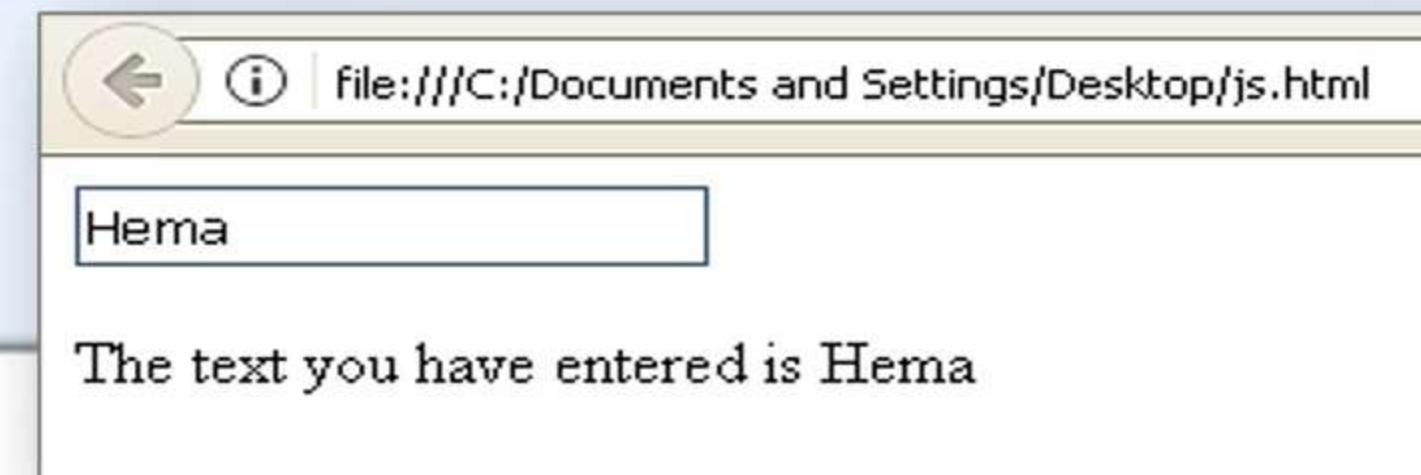
- blur()
- focus()
- select()

Events

- onBlur
- onChange
- onFocus
- onSelect

Text Object

```
<html>
<body>
<form>
<input type="text" onblur="display(this)"/>
<p id="demo"></p>
<script type="text/javascript">
function display(str)
{
    document.getElementById("demo").innerHTML = "The text you have entered is "
                                                +str.value;
}
</script>
</body> </html>
```



Button Object

```
<input type="button" name="myButton" value="Press This" onClick="clickFunction">
```

The input type could be "button" or "submit" or "reset".

Properties

- disabled
- form
- name
- type
- value

Methods

- blur() - Takes the focus away from the radio button.
- click() - This function acts as if the user clicked the button.
- focus() - Gives the focus to the checkbox.

Events

- onBlur
- onClick
- onFocus

Checkbox Object

```
<INPUT TYPE="checkbox" NAME="Name1" VALUE="1" CHECKED onClick="clickFunction">
```

The option "CHECKED" sets the button so it is selected when it is initially displayed.

Properties

- checked
- defaultChecked
- form
- name
- type
- value

Checkbox Object

Methods

- blur()
- click()
- focus()

Events

- onBlur
- onClick
- onFocus

Radio Object

Represents an HTML <input> element with type="radio"

Properties

- checked
- defaultChecked
- form
- name
- type
- value

Methods

- blur()
- click()
- focus()

Radio Object

Events

- onBlur
- onClick
- onFocus

```
<body>
  <form name="form1">
    <p><input type=radio name="seats" value="sleeper">Sleeper</p>
    <p><input type=radio name="seats" value="semisleeper">Semi Sleeper</p>
    <p><input type=radio name="seats" value="normal">Normal</p>
    <p><input type=button value="Show Selected Seat"
      onClick="getSelectedSeat(this.form.seats)"></p>
    <p><div id="msg"></div></p>
  </form>
```

Select Object

Represents HTML <select> element

Properties

- form
- length
- name
- options
- selectedIndex
- type

Methods

- blur()
- focus()
- add()
- remove()

Select Object

Events

- onBlur
- onChange
- onFocus

```
<body>
<form>
<select id="technology"
onchange="selectMethodsDemo()">
    <option>HTML</option>
    <option>CSS3</option>
    <option>Javascript</option>
    <option>JQuery</option>
</select>
</form>
```

Summary

- Events
- Javascript event handling
- Java script validation
- Working with Form Object (Form elements properties, methods and events)



JAVA SCRIPT



In this module you will learn

- String, Date, Array, RegExp in Javascript
- Implicit Objects of Javascript
- Document Object Model (Window, Frame, Navigator Objects)
- Working With Document Object (Its Properties and methods)
- Cookie handling
- Working with Regular Expressions



Implicit Objects in JavaScript



Object-based Language

Has State and Behaviour

Template based

Array, String, Number, Boolean, Date, Math are
few implicit objects in JavaScript.

String

Array of characters

```
var name= "Teknoturf";  
var name= new  
String("Teknoturf");
```

Property

```
name.length → 9
```

Special Characters

```
name="John \"David\"";
```

Code	Outputs
\'	single quote
\"	double quote
\\\	backslash

String Methods

Method	Description
charAt()	Returns the character at the specified index (position)
concat()	Joins two or more strings, and returns a copy of the joined strings
indexOf()	Returns the position of the first occurrence of a specified value in a string
lastIndexOf()	Returns the position of the last occurrence of a specified value in a string
localeCompare()	Compares two strings in the current locale
replace()	Searches within a string for a value and returns a new string with the replaced value
search()	Searches a string for a value and returns the position of the match

URL String Encoding and Decoding

The encodeURI() function is used to encode a URI.

This function encodes special characters, except: , / ? : @ & = + \$ #

```
<p>Click the button to encode a URI.</p>
<button onclick="testEncode()">Try it</button>
<p id="uridemo"></p>

<script>
function testEncode ()
{
    var uri = "teknoturf infoservices?name=Tinå&location=USA";
    var res = encodeURI(uri);
    document.getElementById("uridemo").innerHTML = res;
}
</script>
```

*teknoturf%20infoservices?name=Tin%C3%A5
&location=USA*

URL String Encoding and Decoding



The decodeURI() function is used to decode a URI.

```
<p>Click the button to decode a URI after encoding it.</p>
<button onclick="testDecode()">Try it</button>
<p id="uridemo"></p>
<script>
function testDecode() {
    var uri = "teknoturf infoservices?name=Tin%C3%A5&location=USA";
    var encode = encodeURI(uri);
    var decode = decodeURI(encode);
    var res = "Encoded URI: " + encode + "<br>" + "Decoded URI: " + decode;
    document.getElementById("uridemo").innerHTML = res;
}
</script>
```

Encoded URI: teknoturf%20infoservices?name=Tin%C3%A5&location=USA
Decoded URI: teknoturf infoservices?name=Tinå&location=USA

Math Object



Properties	Description	Example
PI	Returns PI (approx. 3.14)	<code>document.write(Math.PI);</code> //returns 3.141592653589793
SQRT1_2	Returns the square root of 1/2 (approx. 0.707)	<code>document.write(Math.SQRT1_2);</code> //returns 0.7071067811865476
SQRT2	Returns the square root of 2 (approx. 1.414)	<code>document.write(Math.SQRT2);</code> //returns 1.4142135623730951

Math Functions



Function	Description	Example
abs(x)	Returns the absolute value of x	Math.abs(-7.25); // 7.25
ceil(x)	Returns x, rounded upwards to the nearest integer	Math.ceil(1.4) //2
floor(x)	Returns x, rounded downwards to the nearest integer	Math.floor(1.6); //1
max(x,y,z,...,n)	Returns the number with the highest value	Math.max(5, 10); //10
min(x,y,z,...,n)	Returns the number with the lowest value	Math.min(5, 10); //5
pow(x,y)	Returns the value of x to the power of y	Math.pow(4, 3); //64

Date Object

Helps to work with dates in JavaScript

Can be created in many ways

- new Date() Tue Jan 24 2017 11:59:10 GMT+0530 (India Standard Time)
- new Date(milliseconds) Tue Jan 24 2017 11:59:00 GMT+0530 (India Standard Time)
- new Date(dateString) Tue Jan 24 2017 12:02:10 GMT+0530 (India Standard Time)
- new Date(year, month, day, hours, minutes, seconds, milliseconds)

Examples

```
<script type="text/javascript">
    var date1 = new Date();
    var date2 = new Date("January 24 , 2017 11:59:00");
    var date3 = new Date(2017,0,24,12,02,10,15,20);
    document.write(date1+"<br />"+date2+"<br />"+date3);
</script>
```

Date Object

Date Formats – can be used when constructing date object

ISO Format - (YYYY-MM-DD) -- DD and MM are optional

Long Date - (MMM DD YYYY) -- year, month, and day can be in any order

Short Date - (MM/DD/YYYY) -- Either "/" or "-" can be used as a separator

Method	Description
getDate()	Get the day as a number (1-31)
getDay()	Get the weekday as a number (0-6)
getFullYear()	Get the four digit year (yyyy)
getHours()	Get the hour (0-23)
getMilliseconds()	Get the milliseconds (0-999)
getMinutes()	Get the minutes (0-59)
getMonth()	Get the month (0-11)

Date Function

Print current Date

```
<script type="text/javascript">
var currDate= new Date()
var year=currDate.getFullYear()
var month=currDate.getMonth()+1
var day=currDate.getDate()

document.write("Today's date is: ")
document.write(year+"/"+month+"/"+day)
</script>
```

Print the day of the week - when date is given

```
birthday = new Date(1998,2,14)
weekDay = birthday.getDay()

alert(weekDay) //alerts 6
```

Arrays

- Array is a data structure consisting of similar data items sharing a common name
- JavaScript arrays are “dynamic” entities where the size can be dynamically changed
- At each individual location is an element, which is accessed by its position or index.
- The first element in every array is at 0th position.
- In general, the nth element of an array c is referred to as `c[n-1]`

Arrays - Example

```
<script type="text/javascript">
  var n1=new Array(5);
  var n2=new Array();
  for(var i=0;i<n1.length;i++)
    n1[i]=i;
  for(var i=0;i<n1.length;i++)
    n2[i]=1;
  displayArray(n1);
  displayArray(n2);
</script>
```

Operator new creates an array that can hold 5 elements, under the name n1

Operator new creates an empty array under the name n2

```
function displayArray(array)
{
  for(var i=0;i<array.length;i++)
    document.writeln(array[i]+" ");
}
</script>
```

Arrays

Method	Description
concat()	Joins two or more arrays, and returns a copy of the joined arrays
indexOf()	Searches the array for an element and returns its position
join()	Joins all elements of an array into a string
lastIndexOf()	Searches the array for an element, starting at the end, and returns its position
pop()	Removes the last element of an array, and returns that element
push()	Adds new elements to the end of an array, and returns the new length
reverse()	Reverses the order of the elements in an array
shift()	Removes the first element of an array, and returns that element

Arrays

```
<script type="text/javascript">
var names=["John","Pinky","George"];
document.write("<b>Concatenated Names : </b>"+names.join(" ")+<br />");
document.write("<b>Index of \"Pinky\" : </b>"+names.indexOf("Pinky")+<br />");
document.write("<b>Pop last element : </b>"+names.pop()+<br />");
document.write("<b>Push \"Tom\" element at last </b>"+names.push("Tom")+"<br />");
document.write("<b>Reverse Elements : </b>"+names.reverse()+"<br />");
document.write("<b>Shift elements from 1st : </b>"+names.shift()+"<br />");
document.write("<b>Unshift elements to 1st : </b>"+names.unshift("Tom")+"<br />");
document.write("<b>Slice element from 0 to 2 : </b>"+names.slice(0,2)+"<br />");
document.write("<b>Sort Array : </b>"+names.sort()+"<br />");
document.write("<b>Splice element at pos 1 :
</b>"+names.splice(1,0,"Prem","Caesar"));
document.write(names+"<br />");
document.write("<b>Delete an element at position 0 : </b>"+names.splice(0,1)+"<br />");
document.write("<b>Name List : </b>"+names);
</script>
```

Boolean Object

Syntax

```
var x = new Boolean(expression);
```

Function	Description
toSource	Returns a string which represents the source code of a boolean object.
toString	Returns a string representing the specified boolean object.
valueof	Returns the primitive value of a boolean object.

Initial Boolean
value : false

`new Boolean (false);`
`new Boolean ();`
`new Boolean ("");`
`new Boolean (0);`
`new Boolean (null);`

*Boolean
Methods*

Boolean Object

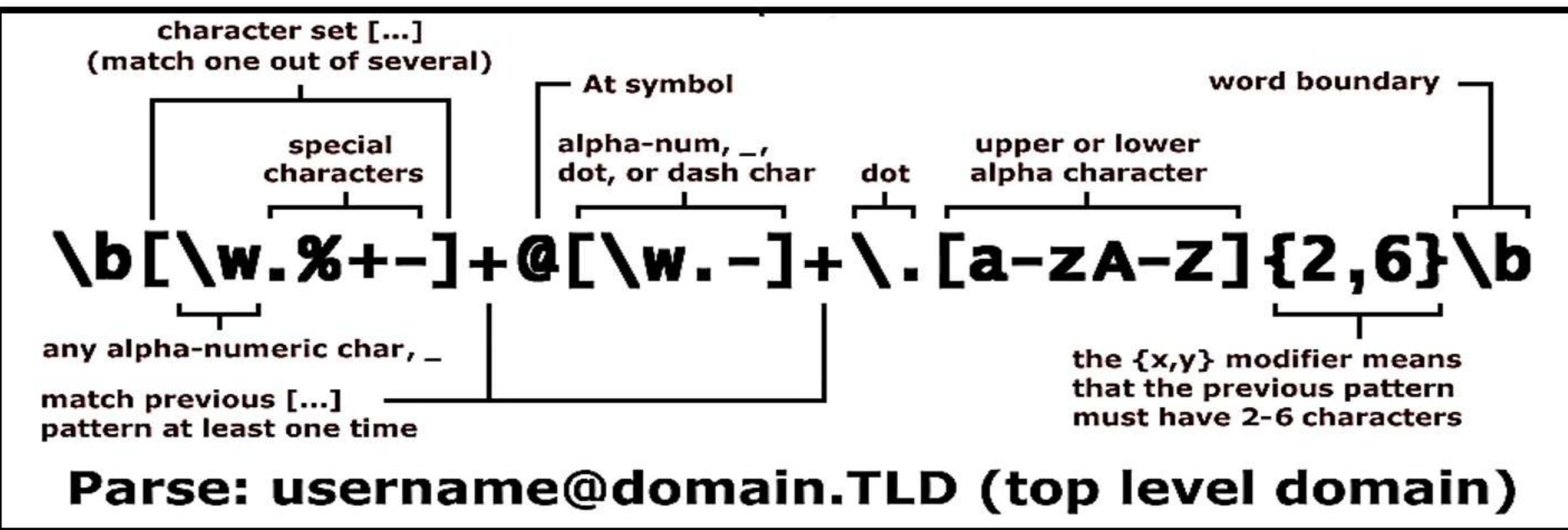
Example

```
<html>
  <head>
    <title>JavaScript toString() Method</title>
  </head>
  <body>
    <script type="text/javascript">
      var flag = new Boolean(false);
      document.write( "flag.toString is : " + flag.toString()+"<br>" );
      document.write( "flag.valueOf is : " + flag.valueOf()+"<br>" );
      document.write( "flag source is : " + flag.toSource());
    </script>
  </body>
</html>
```

flag.toString is : false
flag.valueOf is : false
flag source is : (new Boolean(false))

RegExp Object

A regular expression is an object that describes a pattern of characters.



RegExp Object

A regular expression could be defined with the RegExp() constructor, as follows:

```
var pat = new RegExp(pattern, modifiers); OR var pat = /pattern/modifiers;
```

pattern : A string that specifies the pattern of the regular expression or another regular expression.

modifiers : Specifies global, case-insensitive and multiline matches

A modifier can be : “g” – Finds all global matches

“i” – Does case-insensitive matches

“m” – Does multiline matches

RegExp Object

In Regular expressions, brackets have a special meaning.

They are used to find a range of characters.

Expression	Description
[abc]	Finds any one character between the brackets.
[^abc]	Finds any one character NOT between the brackets.
[0-9]	It matches any decimal digit from 0 through 9.
[^0-9]	It matches any decimal digit not from 0 through 9.
[a-z]	It matches any character from lowercase a through lowercase z.
[A-Z]	It matches any character from uppercase A through uppercase Z.
[a-Z]	It matches any character from lowercase a through uppercase Z.
(x/y)	Finds any of the alternatives specified . x or y

RegExp Object Methods

Here is a list of the properties associated with RegExp and their description.

Method	Description
exec()	Returns the first match. If no match returns null Syntax: RegExpObject.exec(string);
test()	Returns true or false Syntax: RegExpObject.test(string);
toString()	Syntax: RegExpObject.toString ();

RegExp Object

```
<body>
<p> To do a global case-insensitive search for the characters "a","i" and
"s" in the string
<h3 style="color:red" >"I saw Susie sitting in a shoeshine shop" </h3>
</p>
<p id="demo"></p>
<script>
  var str = "I saw Susie sitting in a shoeshine shop";
  var patt1 = /[ais]/gi; //or var patt1 = new RegExp(/ais/,"gi");
  var result = str.match(patt1);
  document.getElementById("demo").innerHTML = result;
</script>
</body>
```

JavaScript Document Object Model

-DOM



- Every web page displayed inside a browser window can be considered as an object.
- JavaScript arranges objects in a Document Object Model or DOM.
- The DOM defines the logical structure of objects and the way in which an object is accessed and manipulated.

JAVASCRIPT DOCUMENT OBJECT MODEL -DOM

Object	JavaScript Object Name
A frame within the browser window	frame
The history list combining the Web pages the user has already visited in the current session	history
The Web browser being run by the user	navigator
The URL of the current Web page	location
The Web page currently shown in the browser window	window
A hyperlink on the current Web page	link
A target or anchor on the current Web page	anchor
A form on the current Web page	form

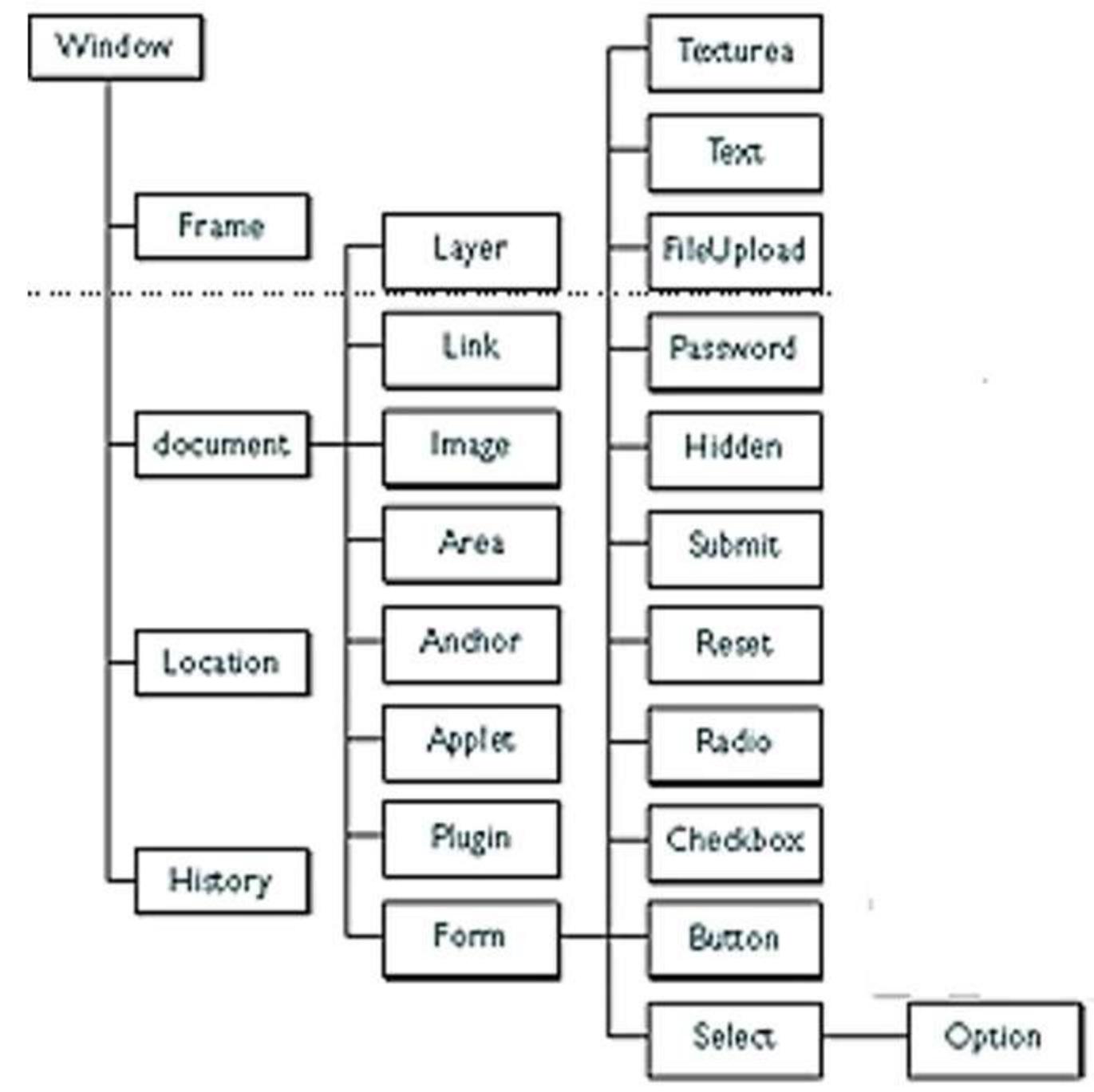
DOM Hierarchy

JavaScript uses Document Object Model(DOM) to navigate the HTML document in a hierarchy

The document object model can be thought of as a hierarchy moving from the most general object to the most specific.

Example:

To access the text field element :
`document.form.text`



DOM Properties



There are several ways of working with properties.

- the value of a property can be changed
- property's value can be stored in a variable
- you can test whether the property equals a specified value using an If...then expression

Some properties are **read - only**, which means you can read the property value, but cannot modify it.

The syntax for changing the value of a property is : **object.property = expression**

Window Object

Window object is the top level object in DOM

Window Object Properties:

- Window.innerHeight – represents the inner height of the browser window
- Window.innerWidth – represents the inner width of the browser window
- **Note : It works for IE, Chrome, Firefox, Opera and Safari**

Window Object Methods:

- Window.open() – opens a new window
- Window.close() – closes the current window
- Window.moveTo() – moves the current window
- Window.resizeTo() – resizes the current window

Window Object Example

```
<!DOCTYPE html>
<html>
<body>
<button onclick="openWindow()">Open gmail.com in a new window </button>
<button onclick="closeWindow()">Close the window mail.com)</button>
<script>
var myWindow;
function openWindow()
{
    myWindow = window.open("http://www.gmail.com", "_blank", "width=500,
height=500");
}
function closeWindow()
{
    myWindow.close();
}
</script>
</body>
</html>
```

_blank : Loads the URL in a new window

_self : URL replaces the current page

Window Object Example

```
<!DOCTYPE html>
<html><body>
<script>
var newWindow = window.open("", "msgWindow", "width=300,height=100,
                           menubar=yes, status=yes,resizable=yes");

newWindow.document.write("<html><head><title>New Window
</title></head><body><h1>Hello World</h1></body></html>");

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



Window Object Example

```
<!DOCTYPE html>
<html><body>
<script>
var newWindow = window.open("", "msgWindow", "width=300,height=100,
                           menubar=yes, status=yes,resizable=yes");

newWindow.document.write("<html><head><title>New Window
</title></head><body><h1>Hello World</h1></body></html>");

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



Frame Object

It is the representation of HTML frame which belongs to a HTML frameset.

Is a property of the window object.

Properties

- frames - An array of frames in a window or frame set
- name - Name of the frame defined using the name attr
- Length - Length of the frames array.
- parent - Parent of the current frame.
- self - Current frame.

Frame Object

Methods

- blur()
- focus()
- setInterval()
- clearInterval()
- setTimeout(expression, milliseconds)
 - Makes a timeout value and returns a timeout ID. After the number of milliseconds, the expression is evaluated.
- clearTimeout(timeout)

Events

- onBlur
- onFocus

Navigator Object

It is an object that has information about the browser

Properties

- **appName** : The name of the browser ex: Mozilla Firefox
- **appVersion** : The version of the browser which may include a compatibility value and operating system name.
- **cookieEnabled** : Boolean value depending on whether cookies are enabled in the browser.
- **mimeTypes** : An array of MIME type descriptive strings that are supported by the browser. Internet Explorer supports the mimeTypes collection, but it is always empty.
- **userAgent** : Describes the browser associated user agent header

Navigator Object

```
<body>
<button onclick="getBrowserName()"> Click Here to know the Navigator
    Properties</button>
<p id="navigatorProperties"></p>

<script>
function getBrowserName() {
    var name = "Name of the browser is " + navigator.appName;
    name = name+<br />Version info " + navigator.appVersion;
    name = name+<br />Cookies enabled " + navigator.cookieEnabled;
    name = name+<br />User-agent header sent: " + navigator.userAgent;

    document.getElementById(" navigatorProperties ").innerHTML = name;
}
sss
</script>
```

Navigator Object - Methods

`javaEnabled()`

Returns a boolean indicating whether javascript is enabled or not in the browser

`taintEnabled()`

Returns a boolean indicating whether the tainting is enabled or not in the browser. This method is removed in JavaScript version 1.2. Tainting is a security protection mechanism for data.

Document Object

Becomes a document object

Each page has only one document object

In HTML, DOM is a node.

Document object is the root node

Provides properties and methods

Document object refers the html document's <body> tag

Document Object - Properties

Property	Description
cookie	Returns the value of the cookie
domain	Returns domain name of the document server
bgColor	Sets the background color Ex : <code>document.bgColor="#FFFFFF"</code>
fgColor	Sets the text color attribute in the <code><body></code> tag
title	Returns the title of the page
forms	Returns an array containing an entry for each form in the document

 **Note:** The document is a part of the Window object and can be accessed as `window.document`

Document Object - Properties

```
<body>
    <h2>Learning Objects in JavaScript</h2>
    <script type="text/javascript">
        document.title="JavaScript Object
Example";
        document.bgColor="grey";
        document.fgColor="white";
    </script>
</body>
```



Document Object - Methods

Method	Description
document.getElementById()	Finding an element by element id
document.getElementsByTagName()	Finding elements by tag name
document.getElementsByClassName()	Finding elements by class name
document.forms[]	Finding the form element with id passed as argument

Inside the document we have a form object

```
<form name="userlogin">  
User name : <input type="text" id="userId" name="userName">  
</form>
```

getElementById()

getElementById() is a function that helps to access or set the document elements directly

To set the text between container tags like <p>,<div>,,<td> etc.

Syntax: `document.getElementById("elementId").innerText="value";`

To get the text between container tags like <p>,<div>,,<td> etc.

Syntax: `document.getElementById("elementId").value;`

“**elementId**” represents the value of the “**id**” attribute of the form element like

getElementById()

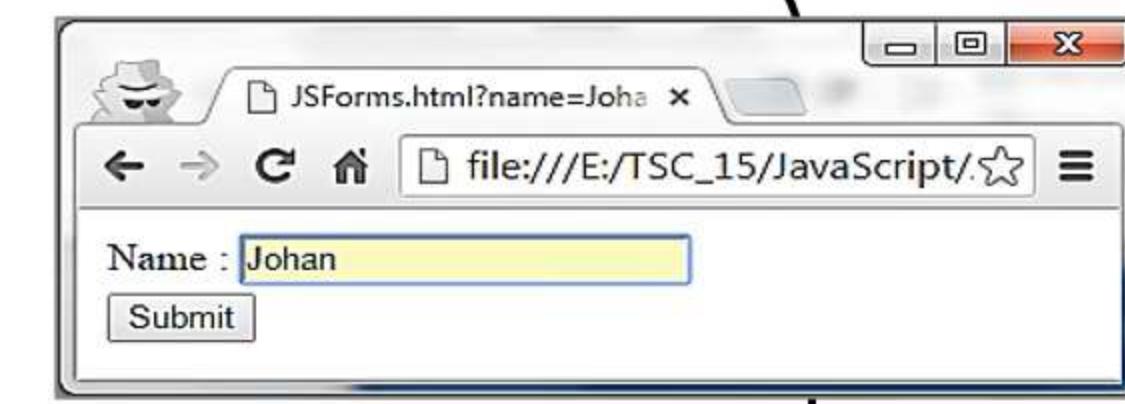
According to the DOM, we can access the value inside the textbox using JavaScript in several ways like:

```
document.getElementById("userId").value;  
      or  
document.userlogin.userId.value;  
      or  
document.forms["userlogin"]["userid"].value
```

Note:
id should be unique

Access Form Element By Field name

```
<head>
  <script type="text/javascript">
    function getData(){
      var name = document.myForm.name.value;
      alert(name);
    }
  </script>
</head>
<body>
  <form name="myForm" onSubmit="getData()">
    <table>
      <tr><td>Name :</td><td><input type="text" name="name"/></td></tr>
      <tr><td colspan="2"><input type="submit" value="Submit"/></td></tr>
    </table>
  </form>
</body>
```



getElementsByClassName()

This method is used to access the element using its class name.

Syntax: var variablename =
document.getElementsByClassName("classname")

Example: var values =
document.getElementsByClassName("msgCheckbox");
//This gets all the elements with the class name "msgChecBox"

Individual elements can be accessed using index.

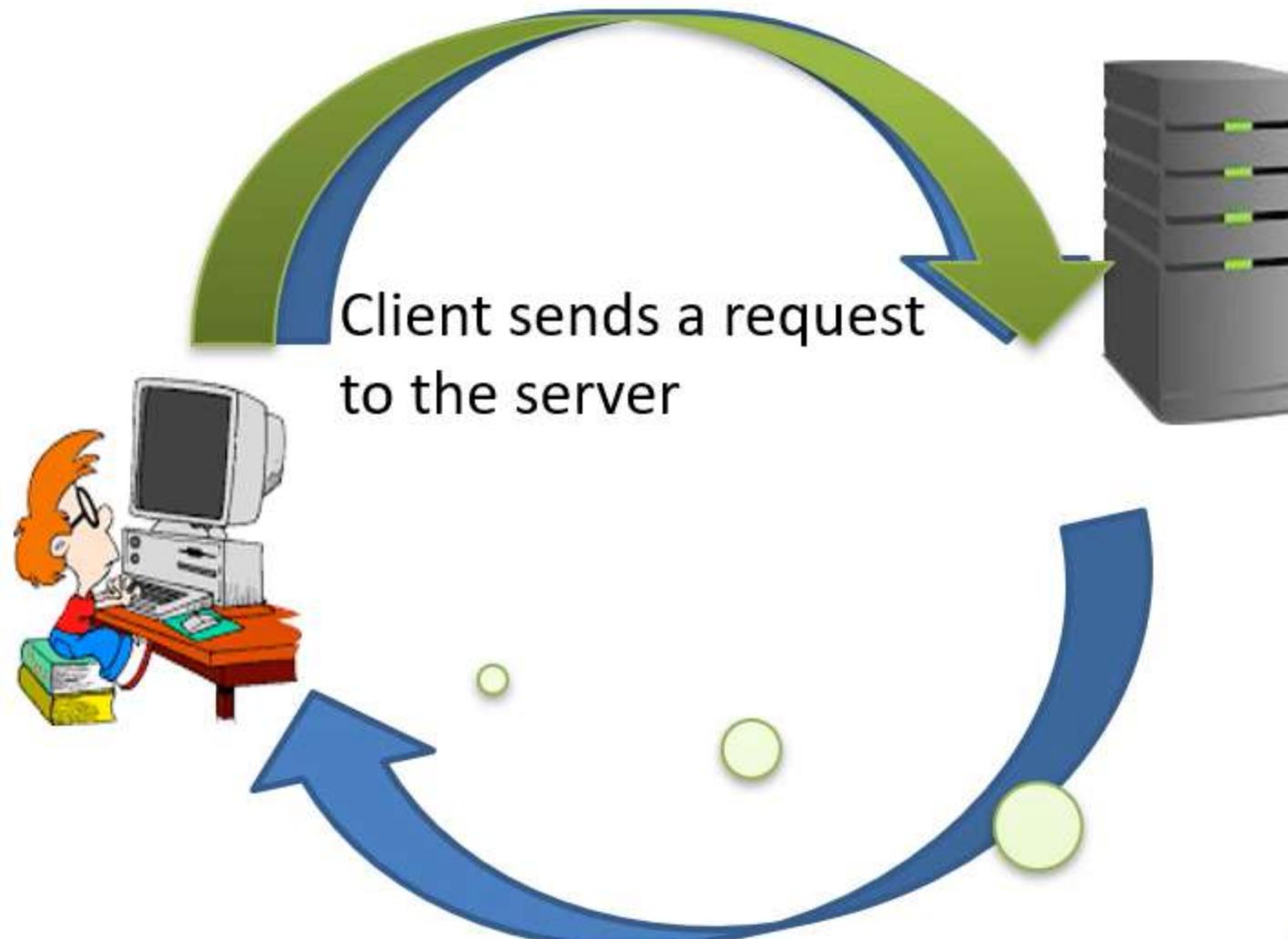
options[0] .value - represents the first element in the collection

getElementsByClassName()

```
<!DOCTYPE html>
<html>
<script>
function check()
{
    var values=document.getElementsByClassName("msgCheckbox");
    document.getElementById("div1").innerHTML="You have selected : ";
    for(var i=0 ; i < values.length ; i++)
    {
        if(values[i].checked)
            document.getElementById("div1").innerHTML += values[i].value+",";
    }
}
</script>
```

Cookies

Client sends a second request to the server



In the client machine, a text file called cookie, will hold the information about the user. Along with the request, the cookie information will be passed to the server. So the server can easily understand it is from the same client.

Now, how does the server know that the request is from the same client?

Read the cookie value

To read the cookie value :

```
var cookievalue=document.cookie;
```

To delete the cookie, set the expires parameter to a passed date as :

- `document.cookie = "emailid=; expires=Mon, 27 Feb 2017 00:00:00 UTC";`

Working with Regular Expressions



Regular expressions are patterns used to match character combinations in strings.

In JavaScript, regular expressions are also objects. These patterns are used with the exec and test methods of RegExp.

Can be used with match, replace, search and split methods of string.

Quantifiers

Expression	Description
+	Represents “One or more”, same as {1,}.
*	Represents “Zero or more”, same as {0,}.
?	Represents “Zero or one”, same as {0,1}.
\$	Represents value at the end
^	Represents value at the beginning. But if it is inside [^] – Not in the given range

Quantifiers

Expression	Description
k+	It matches any string containing at least one occurrence of 'k'.
k*	It matches any string containing zero or more k's.
k?	It matches any string containing zero or one k's.
k{N}	It matches any string containing a sequence of N k's
k{2,3}	It matches any string containing a sequence of two or three k's
k{2,}	It matches any string containing a sequence of at least two k's.
k\$	It matches any string with k at the end of it.
^k	It matches any string with k at the beginning of it.
?=k	Matches any string that is followed by a specific string k
?!k	Matches any string that is not followed by a specific string k

Quantifiers



Example

```
<body>
  <p> To do a global search for an "e" followed by zero or more "t" in the string
  <h3 style="color:red" >"The secret of getting ahead is getting started."</p>
  <p id="demo" style="color:green"></p> </h3>
<script>
  var str = "The secret of getting ahead is getting started.";
  var patt1 = /et*/g;
  var result = str.match(patt1);
  document.getElementById("demo").innerHTML = result;
</script>
</body>
```

Meta characters

Expression	Description
.	Finds a single character
\s	Finds a whitespace character (space, tab, newline)
\S	Finds a non-whitespace character
\d	Finds a digit (0-9)
\D	Finds a non-digit
\w	Finds a word character (a-z, A-Z, 0-9, _)
\W	Finds a non-word character
\b	Finds a match at the beginning/end of a word
\B	Finds a match not at the beginning/end of a word
\0	Finds a NULL character

Meta Characters - Example

```
var str="5678941982384";
var patt1 = /[1-5]/g;
var result = str.match(patt1);
document.writeln("<br />[1-5]/g in '"+str+"' is "+result);

//output : 5,4,1,2,3,4
-----
var str="Hello7815";
var patt1 = /\d/g;
var result = str.match(patt1);
document.writeln("<br /> "+result);

//output : 7,8,1,5
-----
var str="People\nwho\nlive\nin\nglass\nhouses\nshouldn't\nthrow\nstones";
var patt1 = /es$/gm;
var result = str.match(patt1);
document.writeln(result);

//output : es,es
```

RegExp Modifiers

Using “i” modifier to do a case-insensitive search for characters in the string

```
<body>
<p>Example to do a case-insensitive search for "Asia" in the string " Indians are the
Italians of Asia and vice versa." .</p>
<p id="demo" style="color:red"></p>
<script>
var str = "Indians are the Italians of Asia and vice versa.";
var patt1 = /asia/i ;    var result = str.match(patt1);
if(result!=null)
document.getElementById("demo").innerHTML = "Asia string found in
"""+str+"""";
else
document.getElementById("demo").innerHTML = "Asia string not found in
"""+str+"""";
</script></body>
```

We can also create RegExp object as: var patt1=new RegExp("/Asia/", "i")

RegExp using String methods

In JavaScript, regular expressions are often used with following **string methods**: `match()` , `search()`, `replace()`

The `match ()`This method is used to retrieve the matches when matching a string against a regular expression. **Syntax:** `string.match (param/regexp)`

The `search()` method uses an expression to search for a match, and returns the position of the first occurrence of match. **Syntax:** `string.search(regexp);`

The `replace()` method returns a modified string where the pattern is replaced.

Syntax: `string.replace(regexp/substr, newSubStr/function[, RegExp flags]);`

Summary

- String, Date, Array, RegExp in Javascript
- Implicit Objects of Javascript
- Document Object Model (Window, Frame, Navigator Objects)
- Working With Document Object (Its Properties and methods, Cookie handling)
- Work with Regular Expressions





Form Validation Using JavaScript

Activate Windows
Go to PC settings to activate Windows.

Login

 UserID Password

TEKNOTURF Info Services
'Visahala' No.51 Ram Nagar, Coimbatore - 641009



Activate Windows
Go to PC settings to activate Windows.

File Edit Format View Help

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
table
{
border-collapse:collapse;
border:1px solid blue;
}
</style>
</head>
<script language="javascript">
function check()
{
var user=document.getElementById("User").value;
var password=docuemnt.getElementById("pwd").value;

if(user==""||password=="")
{
alert("Fields should not be empty");
}
else
|
<body><br><br><br>
<form>
```

Activate Windows
Go to PC settings to activate Windows.

```
</head>
<script language="javascript">
function check()
{
var user=document.getElementById("User").value;
var password=docuemnt.getElementById("pwd").value;

if(user==""||password=="")
{
alert("Fields should not be empty");
}
else if(user=="Tekno" && password=="tek428")
{
alert("Login Successfully");
}
else
{
alert("Invalid Crediantials");
}
return false;
}
</script>
<body><br><br><br>
<form>
<center>
<table cellpadding=10>
```

Press Esc to exit full screen



Replace a String using RegExp

Activate Windows
Go to PC settings to activate Windows.

```
<html>
<body>
<h2>Replace a Word</h2><br>
<p id="para1">Welcome to Google! Please visit Google Play Store </p>
<p id="para2" style="color:red"></p>
</body>
<script>
var str=document.getElementById("para1").innerHTML;
var text=str.replace(/google/gi,"Yahoo");
document.getElementById("para2").innerHTML=text;
</script>
</html>
```

Replace a Word

Welcome to Google! Please visit Google Play Store

Welcome to Yahoo! Please visit Yahoo Play Store

Activate Windows
Go to PC settings to activate Windows.

JavaScript Validation - Code Demo

i about:blank

Name	<input type="text"/>
EMail	<input type="text"/>
Zip Code	<input type="text"/>
Country	[choose yours] ▾
	<input type="button" value="Submit"/>

File List

Save

Compile & Run

Evaluate

Full screen

Description

CodeDemo.html

```
i 1<html>
2  <head>
3    <title>Form Validation</title>
4    <script type = "text/javascript">
5      // Form validation code will come here.
6    function validate() {
7
8      if( document.myForm.Name.value == "" ) {
9        alert( "Please provide your name!" );
10       document.myForm.Name.focus() ;
11       return false;
12     }
13     if( document.myForm.EMail.value == "" ) {
14       alert( "Please provide your Email!" );
15       document.myForm.EMail.focus() ;
16       return false;
17     }
18     else
19     {
20       validateEmail();
21     }
22
23     if( document.myForm.Zip.value == "" || isNaN( document.myForm.Zip.value ) ||
24       document.myForm.Zip.value.length != 5 ) {
25
26       alert( "Please provide a zip in the format #####." );
27       document.myForm.Zip.focus() ;
28       return false;
29     }
30     if( document.myForm.Country.value == "-1" ) {
```



File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
26         alert( "Please provide a zip in the format #####." );
27         document.myForm.Zip.focus() ;
28         return false;
29     }
30     if( document.myForm.Country.value == "-1" ) {
31         alert( "Please provide your country!" );
32         return false;
33     }
34     return( true );
35 }
36
37
38 function validateEmail() {
39     var emailID = document.myForm.EMail.value;
40     atpos = emailID.indexOf("@");
41     dotpos = emailID.lastIndexOf(".");
42
43     if (atpos < 1 || ( dotpos - atpos < 2 )) {
44         alert("Please enter correct email ID")
45         document.myForm.EMail.focus() ;
46         return false;
47     }
48     return( true );
49 }
50
51
52
53     </script>
54 </head>
```



File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
46         return false;
47     }
48     return( true );
49 }
50
51
52
53     </script>
54 </head>
55
56<body>
57<form action = "." name = "myForm" onsubmit = "return(validate());">
58<table cellspacing = "2" cellpadding = "2" border = "1">
59
60<tr>
61     <td align = "right">Name</td>
62     <td><input type = "text" name = "Name" /></td>
63 </tr>
64
65<tr>
66     <td align = "right">EMail</td>
67     <td><input type = "text" name = "EMail" /></td>
68 </tr>
69
70<tr>
71     <td align = "right">Zip Code</td>
72     <td><input type = "text" name = "Zip" /></td>
73 </tr>
74
75<tr>
```



File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
63             </tr>
64
65             <tr>
66                 <td align = "right">EMail</td>
67                 <td><input type = "text" name = "EMail" /></td>
68             </tr>
69
70             <tr>
71                 <td align = "right">Zip Code</td>
72                 <td><input type = "text" name = "Zip" /></td>
73             </tr>
74
75             <tr>
76                 <td align = "right">Country</td>
77                 <td>
78                     <select name = "Country">
79                         <option value = "-1" selected>[choose yours]</option>
80                         <option value = "1">USA</option>
81                         <option value = "2">UK</option>
82                         <option value = "3">INDIA</option>
83                     </select>
84                 </td>
85             </tr>
86
87             <tr>
88                 <td align = "right"></td>
89                 <td><input type = "submit" value = "Submit" /></td>
90             </tr>
91
92         </table>
```



JavaScript Objects - Code Demo

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

<h2>JavaScript Object Properties</h2>

<p>You can delete object properties.</p>

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

<script>
var person = {
  firstname:"John",
  lastname:"Doe",
  age:50,
  eyecolor:"blue"
};

delete person.age;
document.getElementById("demo").innerHTML =
person.firstname + " is " + person.age + " years old.";
</script>

</body>
</html>
```

JavaScript Object Properties

You can delete object properties.

John is undefined years old.

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

<h2>JavaScript Object Properties</h2>

<p>You can delete object properties.</p>

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

<script>
var person = {
    firstname:"John",
    lastname:"Doe",
    age:50,
    eyecolor:"blue"
};

<!--delete person.age;-->
document.getElementById("demo").innerHTML =
person.firstname + " is " + person.age + " years old.";
</script>

</body>
</html>
```

JavaScript Object Properties

You can delete object properties.

John is 50 years old.

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

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

<script>
var person = {
  firstName: "John",
  lastName : "Doe",
  id      : 5566,
};
person.name = function() {
  return this.firstName + " " + this.lastName;
};

document.getElementById("demo").innerHTML =
"My father is " + person.name();
</script>

</body>
</html>
```

My father is John Doe

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

<h2>JavaScript Getters and Setters</h2>

<p>Getters and setters allow you to get and set properties via methods.</p>
<p>This example uses a lang property to get the value of the language
property.</p>

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

<script>
// Create an object:
var person = {
  firstName: "John",
  lastName : "Doe",
  language : "en",
  get lang() {
    return this.language;
  }
};
// Display data from the object using a getter:
document.getElementById("demo").innerHTML = person.lang;
</script>

</body>
```

JavaScript Getters and Setters

Getters and setters allow you to get and set properties via methods.

This example uses a lang property to get the value of the language property.
en

```
<p id="demo"></p>

<script>
// Constructor function for Person objects
function Person(firstName,lastName,age,eyeColor) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
    this.eyeColor = eyeColor;
    this.changeName = function (name) {
        this.lastName = name;
    }
}
// Create a Person object
var myMother = new Person("Sally","Rally",48,"green");

// Change last name
myMother.changeName("Doe");

// Display last name
document.getElementById("demo").innerHTML =
"My mother's last name is " + myMother.lastName;
</script>

</body>
</html>
```

JavaScript Object Constructors

My mother's last name is Doe

```
<p id="demo"></p>

<script>
// Constructor function for Person objects
function Person(firstName,lastName,age,eyeColor) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
    this.eyeColor = eyeColor;
    this.changeName = function (name) {
        this.lastName = name;
    }
}
// Create a Person object
var myMother = new Person("Sally","Rally",48,"green");

// Change last name
//myMother.changeName("Doe");

// Display last name
document.getElementById("demo").innerHTML =
"My mother's last name is " + myMother.lastName;
</script>

</body>
</html>
```

JavaScript Object Constructors

My mother's last name is Rally

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

<h2>JavaScript Objects</h2>

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

<script>
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
  this.nationality = "English";
}

var myFather = new Person("John", "Doe", 50, "blue");
var myMother = new Person("Sally", "Rally", 48, "green");

document.getElementById("demo").innerHTML =
"The nationality of my father is " + myFather.nationality + ". The
nationality of my mother is " + myMother.nationality;
</script>

</body>
</html>
```

JavaScript Objects

The nationality of my father is English. The nationality of my mother is English

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

<h2>JavaScript Objects</h2>

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

<script>
function Person(first, last, age, eye) {
    this.firstName = first;
    this.lastName = last;
    this.age = age;
    this.eyeColor = eye;
    this.nationality = "Fench";
}

var myFather = new Person("John", "Doe", 50, "blue");
var myMother = new Person("Sally", "Rally", 48, "green");

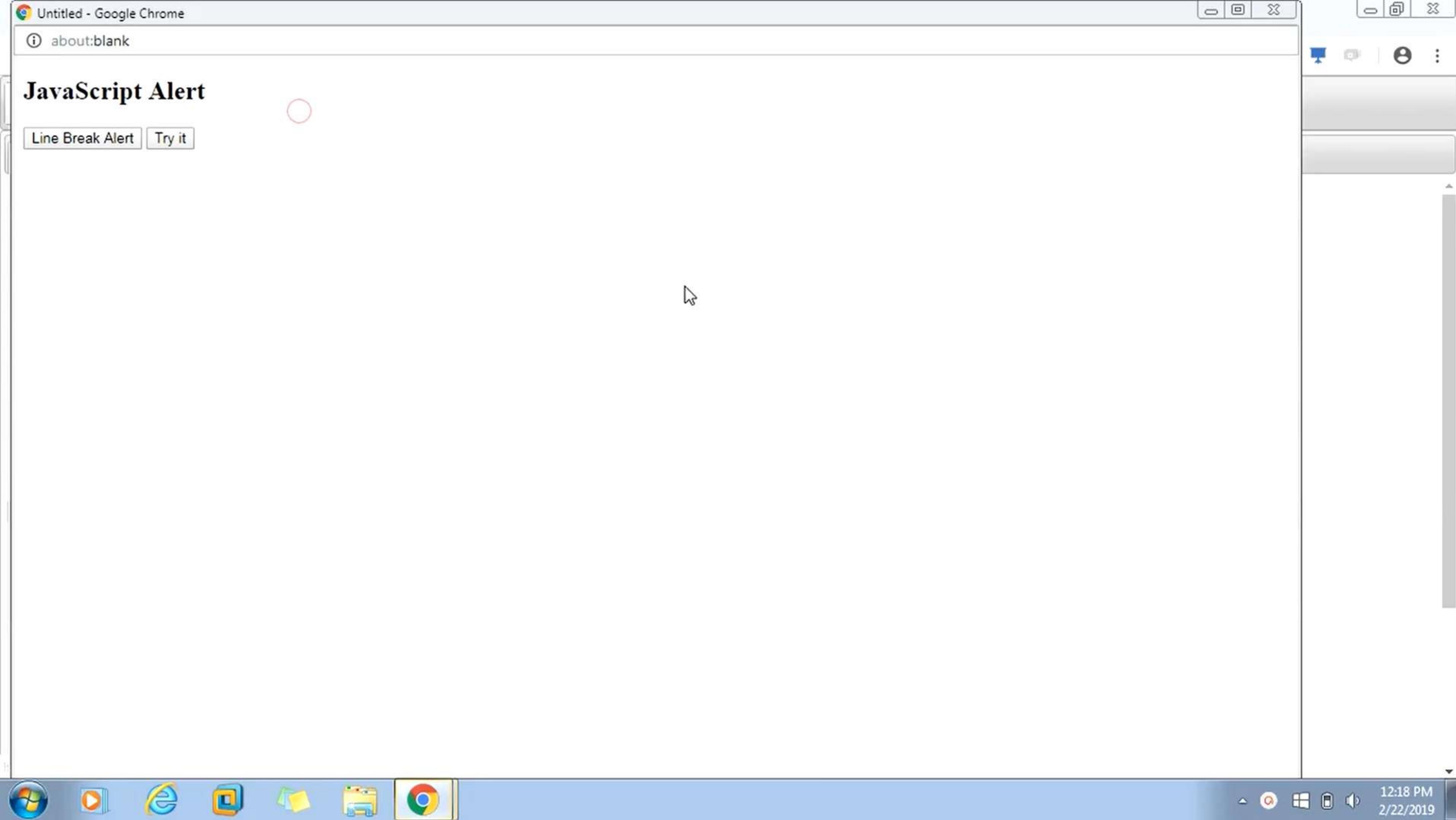
document.getElementById("demo").innerHTML =
"The nationality of my father is " + myFather.nationality + ". The
nationality of my mother is " + myMother.nationality;
</script>

</body>
</html>
```

JavaScript Objects

The nationality of my father is Fench. The nationality of my mother is Fench

JavaScript Alert - Code Demo



JavaScript Alert

[Line Break Alert](#) [Try it](#)

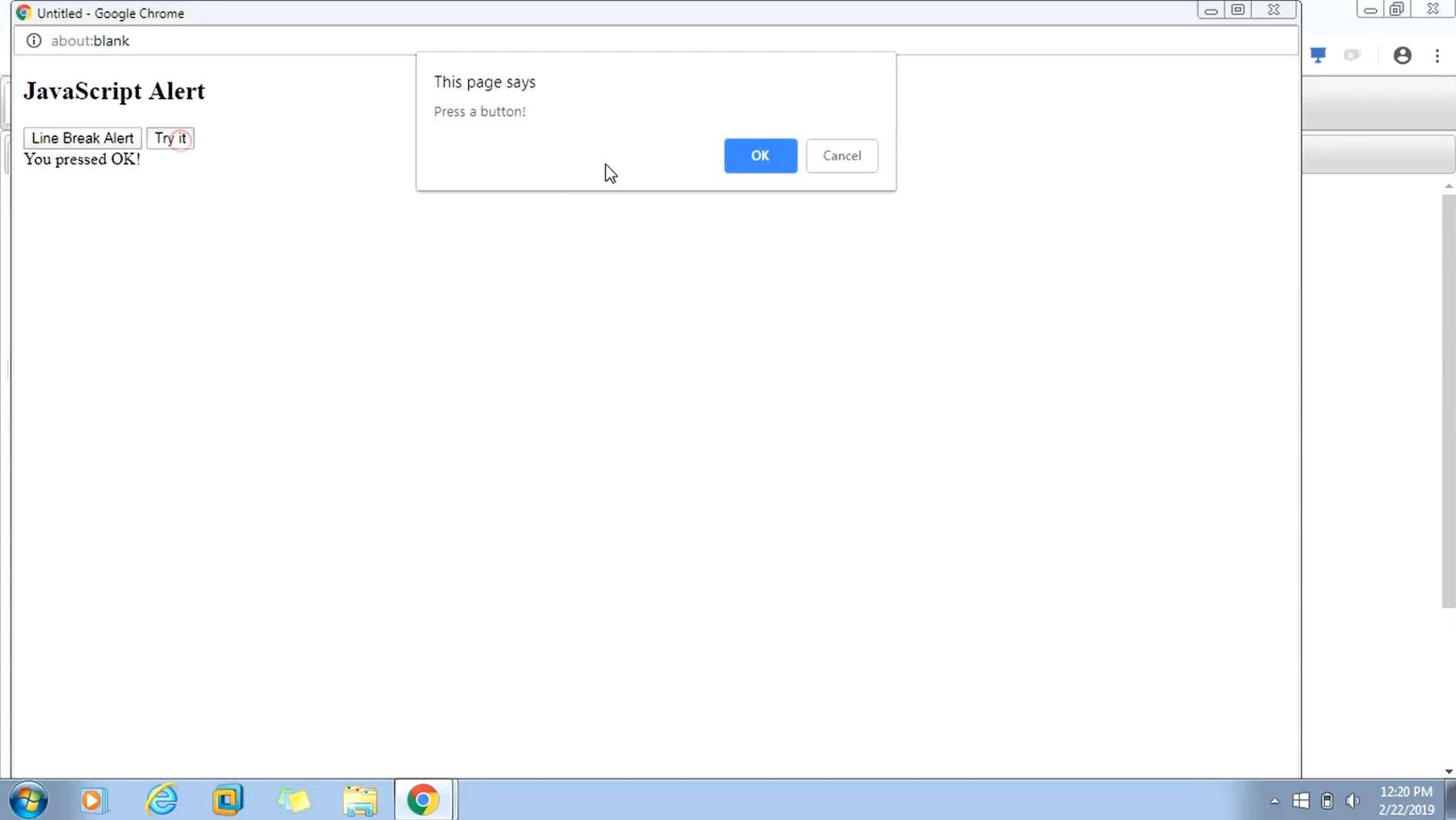


File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h2>JavaScript Alert</h2>
6
7  <button onclick="alert('Hello\nHow are you?')">Line Break Alert</button>
8
9  <button onclick="myFunction1()">Try it</button>
10
11 <script>
12 function myFunction1() {
13     alert("I am an alert box!");
14 }
15
16 function myFunction2() {
17     var txt;
18     if (confirm("Press a button!")) {
19         txt = "You pressed OK!";
20     } else {
21         txt = "You pressed Cancel!";
22     }
23     document.getElementById("demo").innerHTML = txt;
24 }
25
26 function myFunction3() {
27     var txt;
28     var person = prompt("Please enter your name:", "Harry Potter");
29     if (person == null || person == "") {
30         txt = "User cancelled the prompt.";
```





JavaScript Alert

[Line Break Alert](#)[Try it](#)

You pressed OK!

This page says

Press a button!

OK

Cancel

Web Technologies Code Demo +

https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
4
5  <h2>JavaScript Alert</h2>
6
7  <button onclick="alert('Hello\nHow are you?')">Line Break Alert</button>
8
9  <button onclick="myFunction2()">Try it</button>
10 <div id="demo"></div>
11<script>
12function myFunction1() {
13    alert("I am an alert box!");
14}
15
16function myFunction2() {
17    var txt;
18    if (confirm("Press a button!")) {
19        txt = "You pressed OK!";
20    } else {
21        txt = "You pressed Cancel!";
22    }
23    document.getElementById("demo").innerHTML = txt;
24}
25
26function myFunction3() {
27    var txt;
28    var person = prompt("Please enter your name:", "Harry Potter");
29    if (person == null || person == "") {
30        txt = "User cancelled the prompt.";
31    } else {
32        txt = "Hello " + person + "! How are you today?";
33    }
}
```

12:20 PM
2/22/2019

Web Technologies Code Demo +

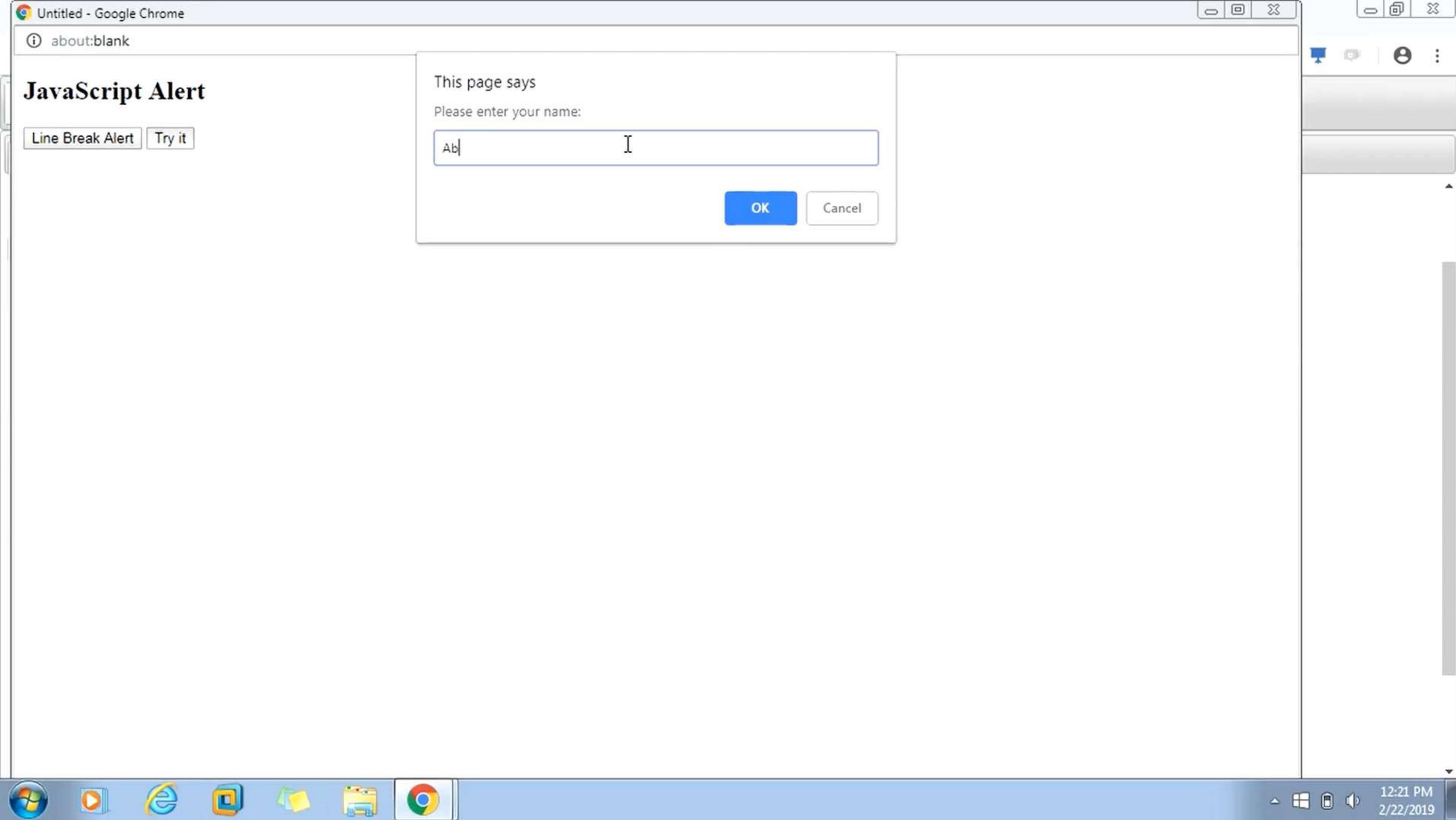
https://lms.tekstac.com/mod/vpl/forms/edit.php?id=17090&userid=232#

File List Save Compile & Run Evaluate Full screen Description

CodeDemo.html

```
12+ function myFunction1() {
13    alert("I am an alert box!");
14 }
15
16+ function myFunction2() {
17    var txt;
18+   if (confirm("Press a button!")) {
19        txt = "You pressed OK!";
20+    } else {
21        txt = "You pressed Cancel!";
22    }
23    document.getElementById("demo").innerHTML = txt;
24 }
25
26+ function myFunction3() {
27    var txt;
28+   var person = prompt("Please enter your name:", "Harry Potter");
29+   if (person == null || person == "") {
30        txt = "User cancelled the prompt.";
31+    } else {
32        txt = "Hello " + person + "! How are you today?";
33    }
34    document.getElementById("demo").innerHTML = txt;
35 }
36
37 </script>
38
39 </body>
40 </html>
41
```

12:21 PM
2/22/2019



JavaScript Alert

[Line Break Alert](#) [Try it](#)

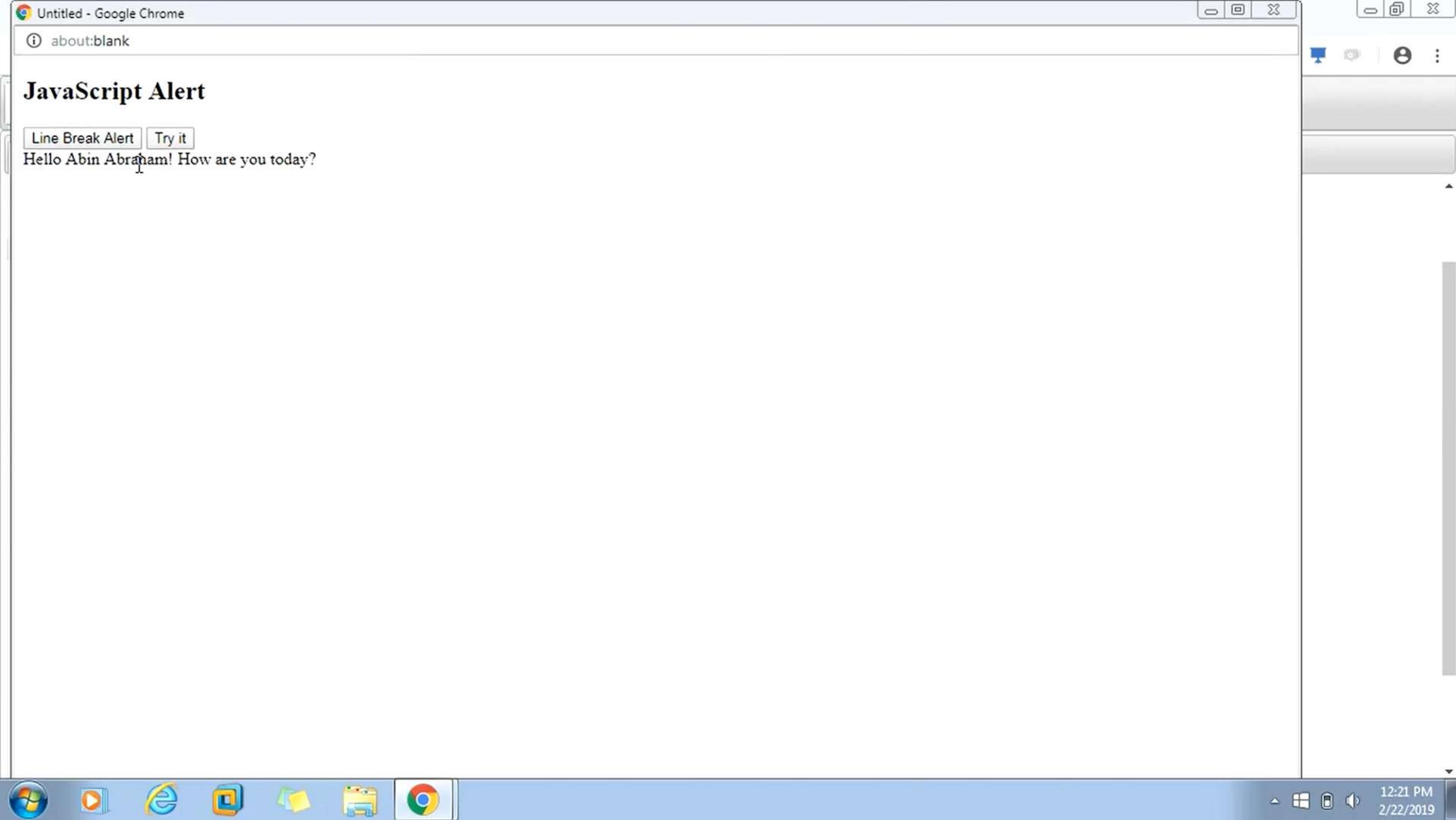
This page says

Please enter your name:

OK

Cancel





JavaScript Alert

[Line Break Alert](#)

[Try it](#)

Hello Abin Abraham! How are you today?

JavaScript BOM - Code Demo

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

<h2>JavaScript</h2>

<h3>JavaScript Window - The Browser Object Model
</h3>

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

<script>
document.getElementById("demo").innerHTML =
"Output : " + window.location.protocol;
</script>

</body>
</html>

<!--
window.location.href returns the href (URL) of the current page
window.location.hostname returns the domain name of the web host
window.location.pathname returns the path and filename of the current page
window.location.protocol returns the web protocol used (http: or https:)
window.location.assign loads a new document
-->
```

JavaScript

JavaScript Window - The Browser Object Model

Output : https:

The `window.screen` object can be written without the `window` prefix.

```
<script>
document.getElementById("demo").innerHTML =
"Output : " + screen.width;
</script>

</body>
</html>

<!--
window.location.href returns the href (URL) of the current page
window.location.hostname returns the domain name of the web host
window.location.pathname returns the path and filename of the current page
window.location.protocol returns the web protocol used (http: or https:)
window.location.assign loads a new document
```

The window.screen object can be written without the window prefix.

Properties:

```
screen.width
screen.height
screen.availWidth
screen.availHeight
screen.colorDepth
screen.pixelDepth
```

window.innerHeight - the inner height of the browser window (in pixels)

JavaScript

JavaScript Window - The Browser Object Model

Output : 1366

```
<script>
document.getElementById("demo").innerHTML =
"Output : " + screen.availWidth;
</script>

</body>
</html>

<!--
window.location.href returns the href (URL) of the current page
window.location.hostname returns the domain name of the web host
window.location.pathname returns the path and filename of the current page
window.location.protocol returns the web protocol used (http: or https:)
window.location.assign loads a new document
```

The window.screen object can be written without the window prefix.

Properties:

```
screen.width
screen.height
screen.availWidth
screen.availHeight
screen.colorDepth
screen.pixelDepth
```

window.innerHeight - the inner height of the browser window (in pixels)

JavaScript

JavaScript Window - The Browser Object Model

Output : 1366

```
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
"Output : " + navigator.platform;
</script>

</body>
</html>

<!--
window.location.href returns the href (URL) of the current page
window.location.hostname returns the domain name of the web host
window.location.pathname returns the path and filename of the current page
window.location.protocol returns the web protocol used (http: or https:)
window.location.assign loads a new document
```

The window.screen object can be written without the window prefix.

Properties:

```
screen.width
screen.height
screen.availWidth
screen.availHeight
screen.colorDepth
```

JavaScript

JavaScript Window - The Browser Object Model

Output : Win32

JavaScript HTML DOM - Code Demo

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

<h2>My First Page</h2>
<input type="text" name="tb">
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!!";
//document.getElementsByTagName("input").tb.value="From Script"
//document.getElementsByName(name)
//document.getElementsByClassName(name)
//document.write(Date())
</script>

</body>
</html>
```

My First Page

Hello World!!

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

<h2>My First Page</h2>
<input type="text" name="tb">
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello Globe";
//document.getElementsByTagName("input").tb.value="From Script"
//document.getElementsByName(name)
//document.getElementsByClassName(name)
//document.write(Date())
</script>

</body>
</html>
```

My First Page

Hello Globe

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

<h2>My First Page</h2>
<input type="text" name="tb">
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello Globe";
document.getElementsByTagName("input").tb.value="From Script"
//document.getElementsByTagName(name)
//document.getElementsByClassName(name)
//document.write(Date())
</script>

</body>
</html>
```

My First Page

From Script

Hello Globe

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

<h2>My First Page</h2>
<input type="text" name="tb">
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello Globe";
document.getElementsByName("input").tb.value="From Script"
//document.getElementsByTagName(name)
//document.getElementsByClassName(name)
document.write(Date())
</script>

</body>
</html>
```

My First Page

From Script

Hello Globe

Fri Feb 22 2019 11:43:08 GMT+0530 (India Standard Time)

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

<h2>Finding HTML Elements by Query Selector</h2>

<p>Hello World!</p>

<p class="intro">The DOM is very useful.</p>
<p class="intro">This example demonstrates the <b>querySelectorAll</b> method.
</p>

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

<script>
var x = document.querySelectorAll("p.intro");
document.getElementById("demo").innerHTML =
'The first paragraph (index 0) with class="intro": ' + x[0].innerHTML;
</script>

</body>
</html>
```

Finding HTML Elements by Query Selector

Hello World!

The DOM is very useful.

This example demonstrates the `querySelectorAll` method.

The first paragraph (index 0) with class="intro": The DOM is very useful.

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

<h2>Finding HTML Elements by Query Selector</h2>

<p>Hello World!</p>

<p class="intro">The DOM is very useful.</p>
<p class="intro">This example demonstrates the <b>querySelectorAll</b> method.
</p>

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

<script>
var x = document.querySelectorAll("p.intro");
document.getElementById("demo").innerHTML =
'The first paragraph (index 0) with class="intro": ' + x[1].innerHTML;
</script>

</body>
</html>
```

Finding HTML Elements by Query Selector

Hello World!

The DOM is very useful.

This example demonstrates the `querySelectorAll` method.

The first paragraph (index 0) with `class="intro"`: This example demonstrates the `querySelectorAll` method.

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

<h2>Finding HTML Elements Using document.forms</h2>

<form id="frm1" action="/action_page.php">
  First name: <input type="text" name="fname" value="Donald"><br>
  Last name: <input type="text" name="lname" value="Duck"><br><br>
  <input type="submit" value="Submit">
</form>

<p>Click "Try it" to display the value of each element in the form.</p>

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

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

<script>
function myFunction() {
  var x = document.forms["frm1"];
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
  document.getElementById("demo").innerHTML = text;
}

1
```

Finding HTML Elements Using document.forms

First name: Donald
Last name: Duck

Submit

Click "Try it" to display the value of each element in the form.

Try it

Donald
Duck
Submit

```
<form id="frm1" action="/action_page.php">
  First name: <input type="text" name="fname" value="Donald"><br>
  Last name: <input type="text" name="lname" value="Duck"><br><br>
  <input type="submit" value="Submit">
</form>

<p>Click "Try it" to display the value of each element in the form.</p>

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

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

<script>
function myFunction() {
  var x = document.forms["frm1"];
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>

</body>
</html>
```

Finding HTML Elements Using document.forms

First name:

Last name:

Click "Try it" to display the value of each element in the form.

Donald
Duck
Submit

```
<form id="frm1" action="/action_page.php">
  First name: <input type="text" name="fname" value="Donald"><br>
  Last name: <input type="text" name="lname" value="Duck"><br><br>
  <input type="submit" value="Submit">
</form>

<p>Click "Try it" to display the value of each element in the form.</p>

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

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

<script>
function myFunction() {
  var x = document.forms["frm1"];
  var text = "";
  var i;
  for (i = 0; i < x.length ;i++) {
    text += x.elements[i].value + "<br>";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>

</body>
</html>
```

Submitted Form Data

Your input was received as:

fname=Abin&lname=Abraham

The server has processed your input and returned this answer.

Note: This tutorial will not teach you how servers are processing input.
Processing input is explained in our [PHP tutorial](#).

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

<p id="p1">Hello World!</p>
<p id="p2">Hello World!</p>

<script>
document.getElementById("p2").style.color = "blue";
document.getElementById("p2").style.fontFamily = "Arial";
document.getElementById("p2").style.fontSize = "larger";
</script>

<p>The paragraph above was changed by a script.</p>

</body>
</html>
```

Hello World!

Hello World!

The paragraph above was changed by a script.

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

<p id="p1">Hello World!</p>
<p id="p2">Hello World!</p>

<script>
document.getElementById("p2").style.color = "red";
document.getElementById("p2").style.fontFamily = "Arial";
document.getElementById("p2").style.fontSize = "larger";
</script>

<p>The paragraph above was changed by a script.</p>

</body>
</html>
```

Hello World!

Hello World!

The paragraph above was changed by a script.

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

<h1 id="id1">My Heading 1</h1>

<button type="button"
onclick="document.getElementById('id1').style.color = 'red'">
Click Me!</button>

</body>
</html>
```

My Heading 1

Click Me!



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

<h1 id="id1">My Heading 1</h1>

<button type="button"
        onclick="document.getElementById('id1').style.color = 'red'">
    Click Me!</button>

</body>
</html>
```

My Heading 1

Click Me!

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  var x = document.getElementById("fname");
  x.value = x.value.toUpperCase();
}
</script>
</head>
<body>
```

Enter your name: <input type="text" id="fname" onchange="myFunction()">

//onclick
//onload
//onunload
//onmouseover
//onmouseout
//onmousedown
//onmouseup
//onfocus

<p>When you leave the input field, a function is triggered which transforms
the input text to upper case.</p>

</body>
</html>

Enter your name:

When you leave the input field, a function is triggered which transforms the input text to upper case.

```
<body>

<h2>JavaScript addEventListener()</h2>

<p>This example uses the addEventListener() method to add many events on the same button.</p>

<button id="myBtn">Try it</button>

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

<script>
var x = document.getElementById("myBtn");
x.addEventListener("mouseover", myFunction);
x.addEventListener("click", mySecondFunction); I
x.addEventListener("mouseout", myThirdFunction);

function myFunction() {
  document.getElementById("demo").innerHTML += "Moused over!<br>";
}

function mySecondFunction() {
  document.getElementById("demo").innerHTML += "Clicked!<br>";
}

function myThirdFunction() {
  document.getElementById("demo").innerHTML += "Moused out!<br>";
}
```

JavaScript addEventListener()

This example uses the addEventListener() method to add many events on the same button.

Try it

```
<body>

<h2>JavaScript addEventListener()</h2>

<p>This example uses the addEventListener() method to add many events on the same button.</p>

<button id="myBtn">Try it</button>

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

<script>
var x = document.getElementById("myBtn");
x.addEventListener("mouseover", myFunction);
x.addEventListener("click", mySecondFunction);
x.addEventListener("mouseout", myThirdFunction);

function myFunction() {
  document.getElementById("demo").innerHTML += "Moused over!<br>";
}

function mySecondFunction() {
  document.getElementById("demo").innerHTML += "Clicked!<br>";
}

function myThirdFunction() {
  document.getElementById("demo").innerHTML += "Moused out!<br>";
}
```

JavaScript addEventListener()

This example uses the addEventListener() method to add many events on the same button.



Moused over!