Unit II – UI Design

Copyright Guideline

Topics Covered

HTML5

- HTML 5 Basics
  - What is HTML5
  - Features of HTML5
- Semantic Tags
- New Input Elements and Tags
- Media Tags – Audio and Video Tags
- Graphic Tags – Canvas and SVG
- Geo-location API
- Drag and Drop features
- Web Storage
  - Session and Local storage

CSS3

- Introduction to CSS3
  - What is CSS3?
  - Features of CSS3
- CSS 3 Basics
  - Implementation of
    - Border Radius
    - Box Shadow
    - Image Border
    - Background
    - Advanced Text Effects (shadow)
    - Custom Web Fonts
- CSS3 Transformations
- CSS3 Transitions
- CSS3 Animations

HTML5 Basics

Overview

HTML stands for Hyper Text Markup Language

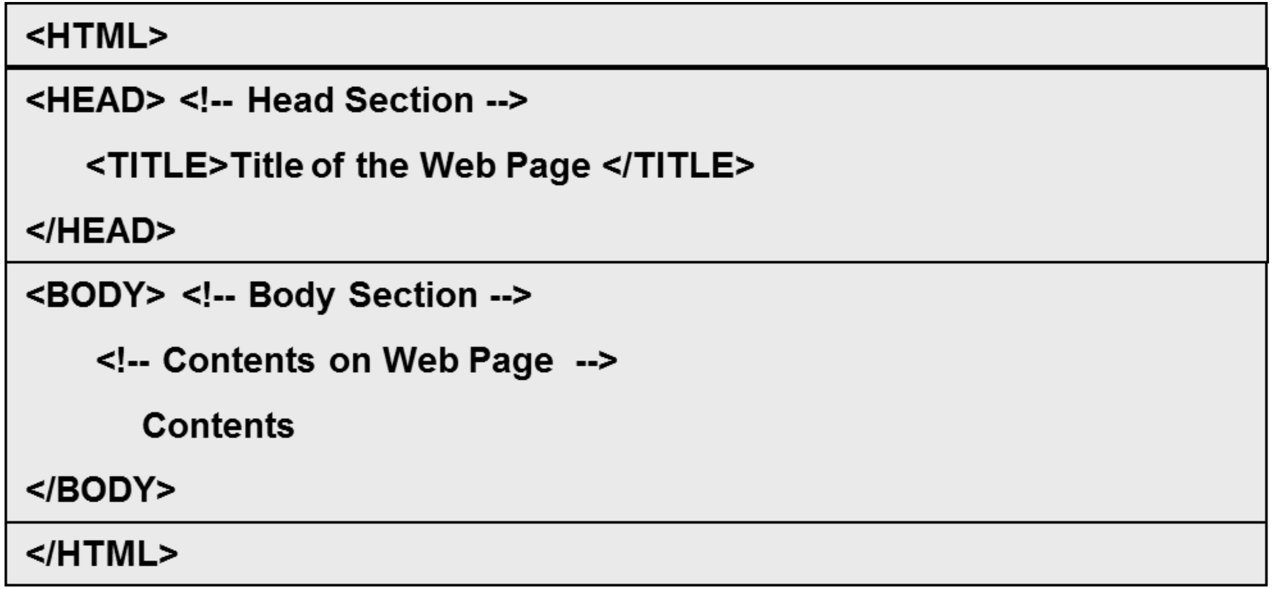
It is used to design and develop Web Pages

Communication on the web happens through HTTP

HTML is

1. A Simple mark up language (NOT a programming language)
2. Browser/Platform Independent (plain ASCII text format)
3. Not Case Sensitive
4. HTML is Interpreted by browser (NOT compiled)
5. A medium for Graphical User Interface (GUI)

Structure of HTML page



```
<HTML>
<HEAD> <!-- Head Section -->
    <TITLE>Title of the Web Page </TITLE>
</HEAD>
<BODY> <!-- Body Section -->
    <!-- Contents on Web Page  -->
        Contents
</BODY>
</HTML>
```

- An HTML file can be created by using a simple text editor viz notepad, text pad, Eclipse IDE editor. HTML file must have an extension .htm or .html.

What is HTML5?

- HTML5 is the latest version(from 2014 onwards) of HTML (Hyper Text Markup Language) prevailing after HTML 4.01, XHTML 1.0 and XHTML 1.1.

- The HTML5 is a standard specification jointly developed by WHATWG and W3C.

- Most of the latest versions of Apple Safari, Google Chrome, Mozilla Firefox and Opera support HTML5 features. Chrome supports highest number of browsers.

- Mobile web browsers pre-installed on iPhones, iPads and Android phones have excellent support for HTML5.

- Earlier versions of HTML required third party plug-ins such as Adobe Flash, Microsoft Silverlight and Google Gears to play video. Whereas, the new standard includes features like video playback to support video and drag-and-drop.

WHATWG and W3C

WHATWG

- WHATWG stands for Web Hyper Text Application Technology Working Group.
- It is a community of people interested in evolving HTML and related technologies.
- This group was founded by individuals from Mozilla Foundation, Opera Software and Apple in 2004

W3C

- W3C stands for World Wide Web Consortium(W3C) is the main International standards organization for the World Wide Web (www or W3)
- W3C involves in education and outreach, develops software and serves as an open forum for discussion about the Web.

WHATWG and W3G

Objective of WHATWG

- Define robust error handling
- Ensure backwards compatibility
- Develop practical new features
- Reduce the need for external plug-in in the browser
- To set new standard for the web

Key objectives of HTML5

Some of the key objectives of the HTML5 spec are:

- To improve documentation of the features
- To bring HTML and JavaScript under the same spec
- To reduce external plugin dependency for media playback
- To reduce the need for JavaScript for common tasks
- To introduce new markups which will simplify page structure and cognition
- To support cross-device compatibility

What’s new in HTML5?

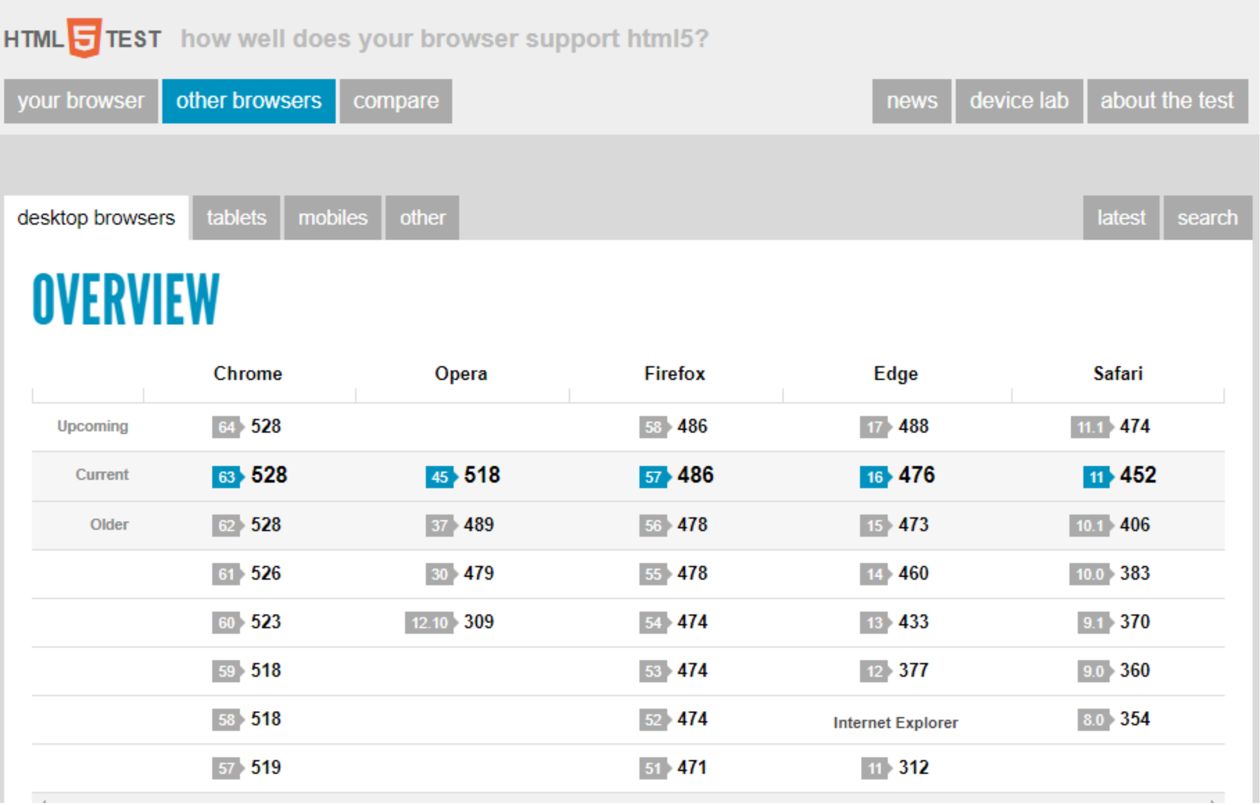
- **New HTML5 Elements:**
  - **Semantic Elements** like <header>, <footer>, <article>, <nav> and <section>
  - **Attributes of Form elements** like number, date, time, calendar and range
  - **Graphic elements** like <svg> and <canvas>
  - **Multimedia elements** like <audio> and <video>

- **Few New HTML5 API’s (Application Programming Interfaces)**
  - Geo - location
  - Drag and Drop
  - Web Storage

Features of HTML5

- New Semantic Elements
- New Input Elements and Form attributes
- Persistent Local Storage
- Canvas
- Audio and Video
- Geo - location
- Drag and Drop and more API's

Browser Support for HTML5



OVERVIEW

	Chrome	Opera	Firefox	Edge	Safari
Upcoming	64 528		58 486	17 488	11.1 474
Current	63 528	45 518	57 486	16 476	11 452
Older	62 528	37 489	56 478	15 473	10.1 406
	61 526	30 479	55 478	14 460	10.0 383
	60 523	12.10 309	54 474	13 433	9.1 370
	59 518		53 474	12 377	9.0 360
	58 518		52 474	Internet Explorer	8.0 354
	57 519		51 471	11 312	

Note: All the data is as per www.html5test.com on Windows 8 OS

Difference between HTML4 and HTML5

- In the HTML5 spec, new tags have been introduced and some old tags and attributes have been removed
- Many of the HTML4 tags will continue to be supported to ensure backward compatibility
- HTML4 used 3 different DOCTYPE declarations to force the browser to operate in the standard mode
- HTML5 proposes just one simple single DOCTYPE

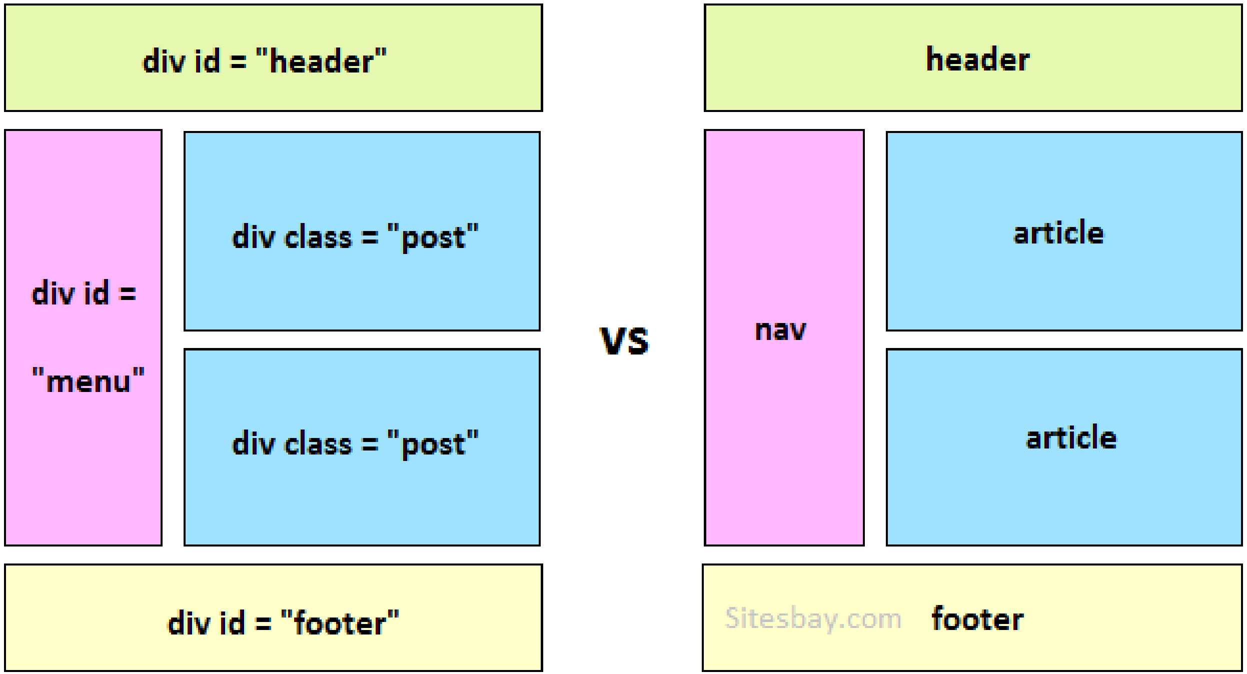
<!DOCTYPE HTML>

One of the important difference between html4 and html5 is the DOCTYPE used

HTML4	HTML5
1. Transitional <!Doctype HTML Public "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtDOCTYPEd"> 2. Strict <! HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" http://www.w3.org/TR/html4/strict.dtd> 3. Frameset <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset/EN" "http://www.w3.org/TR/html4/frameset.dtd">	Only <!DOCTYPE HTML> is used

Difference between HTML4 and HTML5

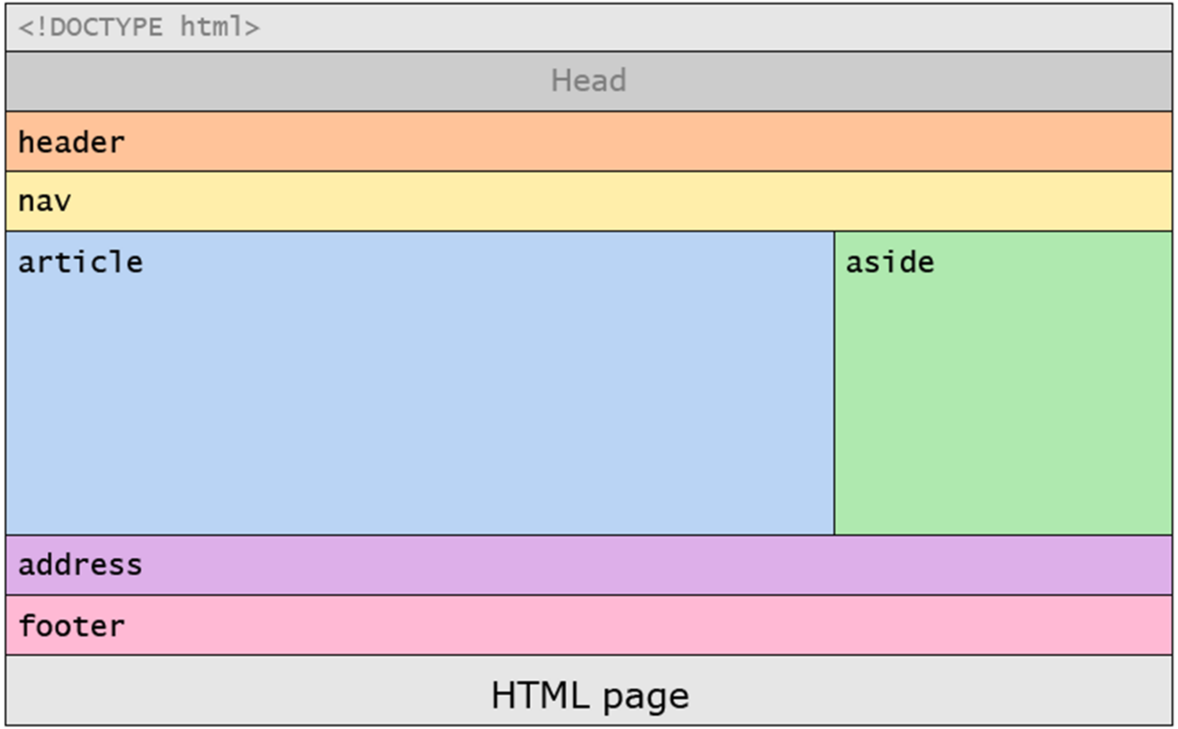
Structure of HTML4 Structure of HTML5



Semantic Tags

Semantic Tags

Describing the structure of a web page in HTML5



<!DOCTYPE html>

Head

header

nav

article

aside

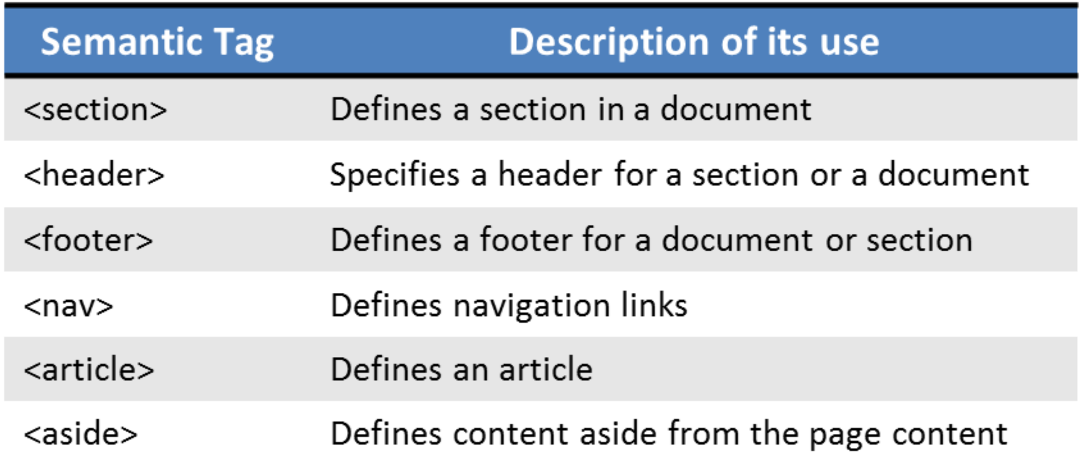
address

footer

HTML page

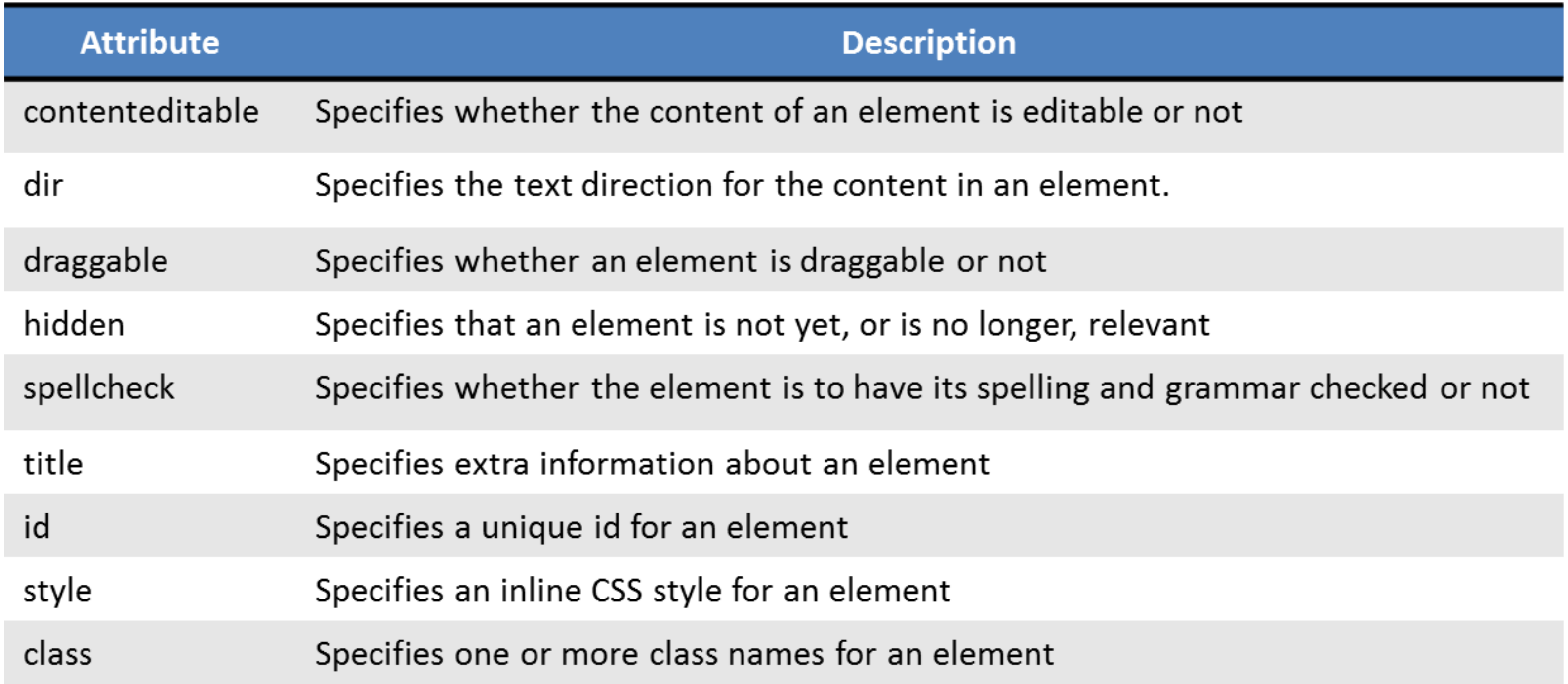
Semantic Tags

- Semantic tags describes its meaning to both browser and the developer.
- They are used to give a structure to the web page.
- Non-semantic elements like <div> or <span> tells nothing about its content.
- Whereas Semantic tags like <form>, <table> and <article> clearly defines its content.



Semantic Tag	Description of its use
<section>	Defines a section in a document
<header>	Specifies a header for a section or a document
<footer>	Defines a footer for a document or section
<nav>	Defines navigation links
<article>	Defines an article
<aside>	Defines content aside from the page content

Some Global Attributes

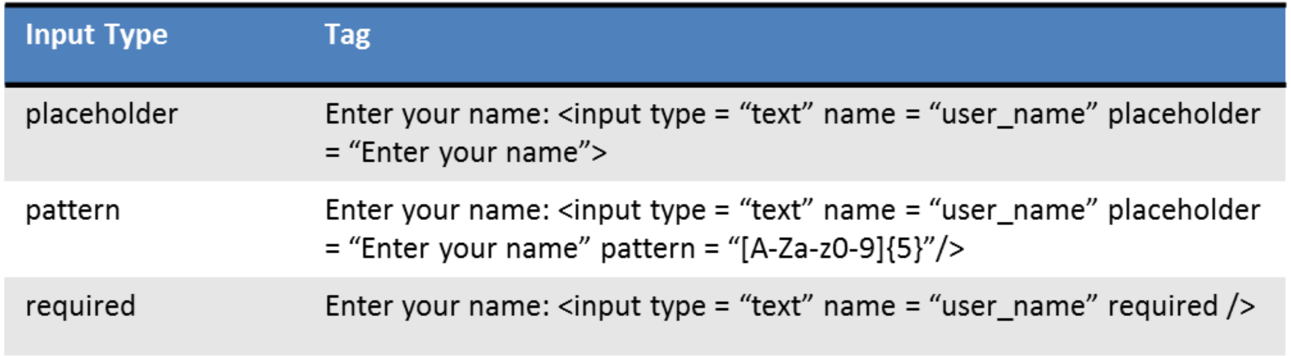


Attribute	Description
contenteditable	Specifies whether the content of an element is editable or not
dir	Specifies the text direction for the content in an element.
draggable	Specifies whether an element is draggable or not
hidden	Specifies that an element is not yet, or is no longer, relevant
spellcheck	Specifies whether the element is to have its spelling and grammar checked or not
title	Specifies extra information about an element
id	Specifies a unique id for an element
style	Specifies an inline CSS style for an element
class	Specifies one or more class names for an element

New Input Elements and Tags

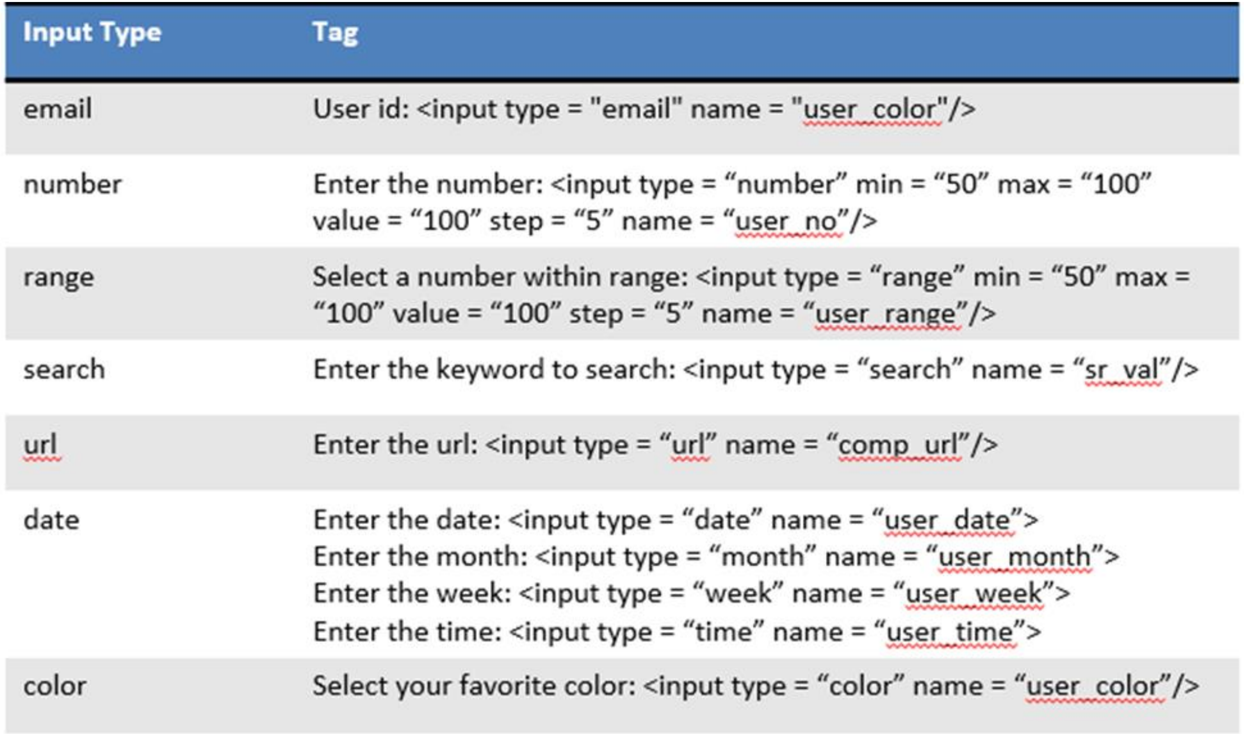
New Input types

- Several input types have been introduced in the HTML5 spec.
- The advantage of these input types is that they have the necessary validation in-built in them without the need of external JavaScript validations.
- Thus even if JavaScript is turned off in the clients machine, the validations will still happen.



Input Type	Tag
placeholder	Enter your name: <input type = “text” name = “user_name” placeholder = “Enter your name”>
pattern	Enter your name: <input type = “text” name = “user_name” placeholder = “Enter your name” pattern = “[A-Za-z0-9]{5}”/>
required	Enter your name: <input type = “text” name = “user_name” required />

New Input Types

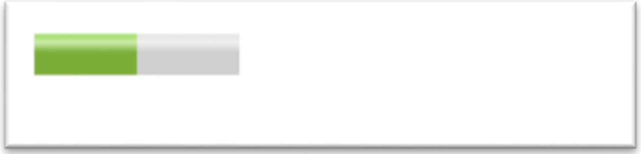


Input Type	Tag
email	User id: <input type = "email" name = "user_color"/>
number	Enter the number: <input type = “number” min = “50” max = “100” value = “100” step = “5” name = “user_no”/>
range	Select a number within range: <input type = “range” min = “50” max = “100” value = “100” step = “5” name = “user_range”/>
search	Enter the keyword to search: <input type = “search” name = “sr_val”/>
url	Enter the url: <input type = “url” name = “comp_url”/>
date	Enter the date: <input type = “date” name = “user_date”> Enter the month: <input type = “month” name = “user_month”> Enter the week: <input type = “week” name = “user_week”> Enter the time: <input type = “time” name = “user_time”>
color	Select your favorite color: <input type = “color” name = “user_color”/>

New Form Elements

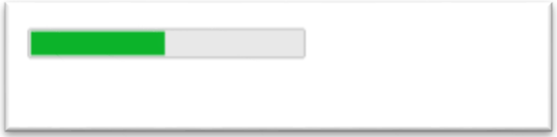
Meter

```
<meter min = “0” max=“100” value=“50”>50 of 100</meter>
```



Progress

```
<progress min = “0” max = “100” value = “50”>50 of 100</progress>
```

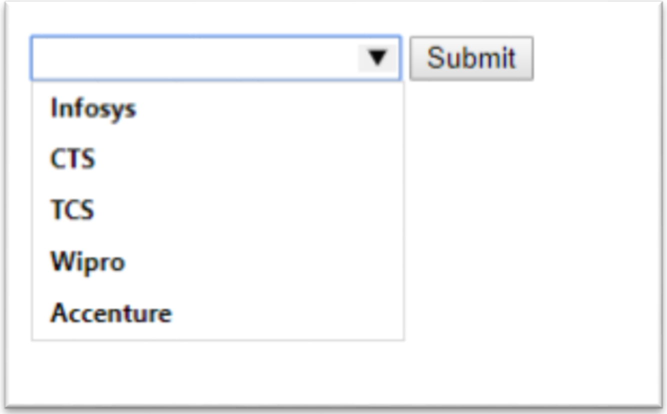


PUBLIC

New Form Elements

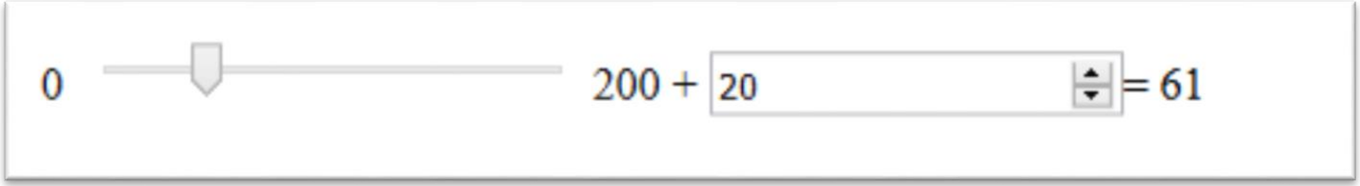
- Some new form elements have been added in the HTML5 spec. They are:
  - Datalist

```
<input type="text" list="Recruiter">
<datalist id="Recruiter">
  <option value="Infosys">
  <option value="CTS">
  <option value="TCS">
  <option value="Wipro">
  <option value="Accenture">
</datalist>
<input type="submit">
```



New Form Elements

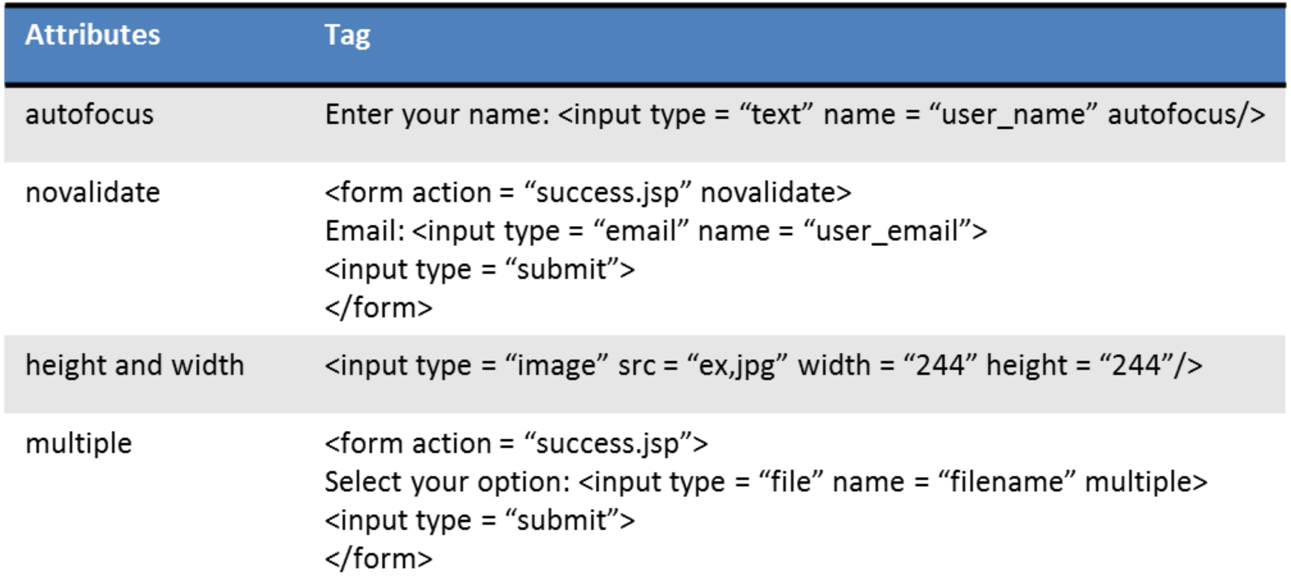
```
<!DOCTYPE html>
<html>
<body>
<form oninput = “x.value = parseInt(a.value) + parseInt(b.value)” > 0
<input type = “range” id = “a” value = “10” min = “1” max = “200”> 200 +
<input type = “number” id = “b” value = “10”> = <output name = “x” for = “a b”></output>
</form>
</body>
</html>
```



0 200 + 20 = 61

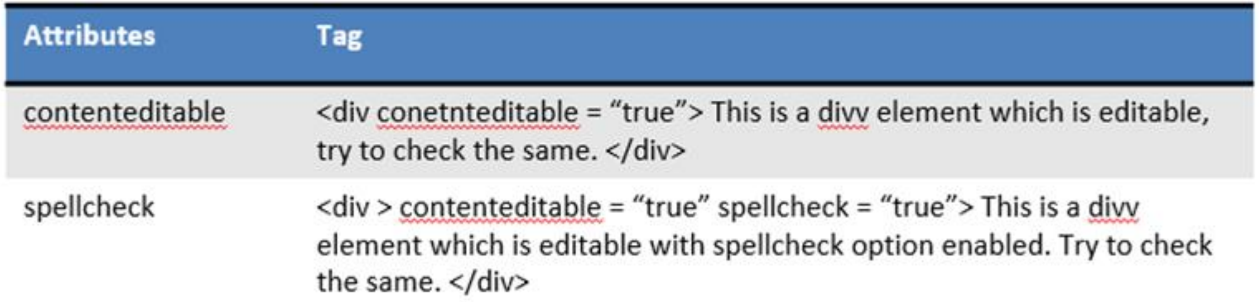
New Form attributes

- HTML5 has introduced a lot of new attributes for form and input type.
- The list of new attributes are:

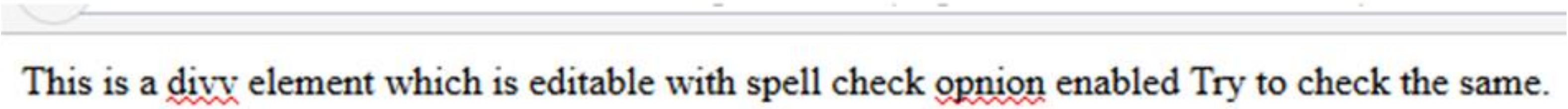


Attributes	Tag
autofocus	Enter your name: <input type = “text” name = “user_name” autofocus/>
novalidate	<form action = “success.jsp” novalidate> Email: <input type = “email” name = “user_email”> <input type = “submit”> </form>
height and width	<input type = “image” src = “ex,jpg” width = “244” height = “244”/>
multiple	<form action = “success.jsp”> Select your option: <input type = “file” name = “filename” multiple> <input type = “submit”> </form>

New Form Attributes



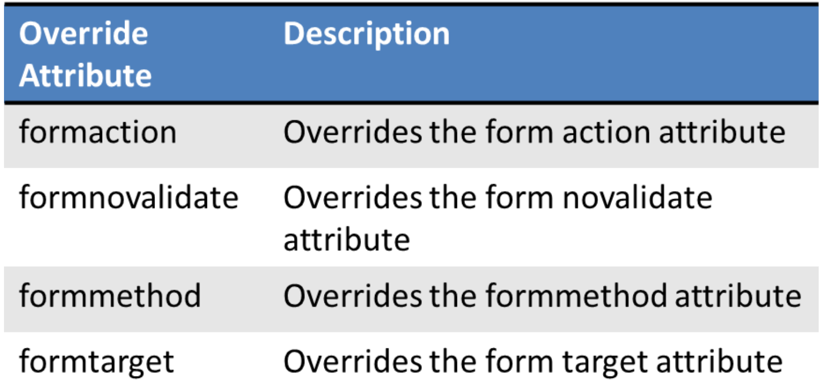
Attributes	Tag
contenteditable	<div conetnteditable = “true”> This is a divv element which is editable, try to check the same. </div>
spellcheck	<div > contenteditable = “true” spellcheck = “true”> This is a divv element which is editable with spellcheck option enabled. Try to check the same. </div>



This is a divy element which is editable with spell check opnion enabled Try to check the same.

Form override attributes

- Form override attributes overrides some attributes of <form> element.
- They are used with input types, submit and image elements.
- Some of form override attributes are:



Override Attribute	Description
formaction	Overrides the form action attribute
formnovalidate	Overrides the form novalidate attribute
formmethod	Overrides the formmethod attribute
formtarget	Overrides the form target attribute

Media Tags

● ● ●

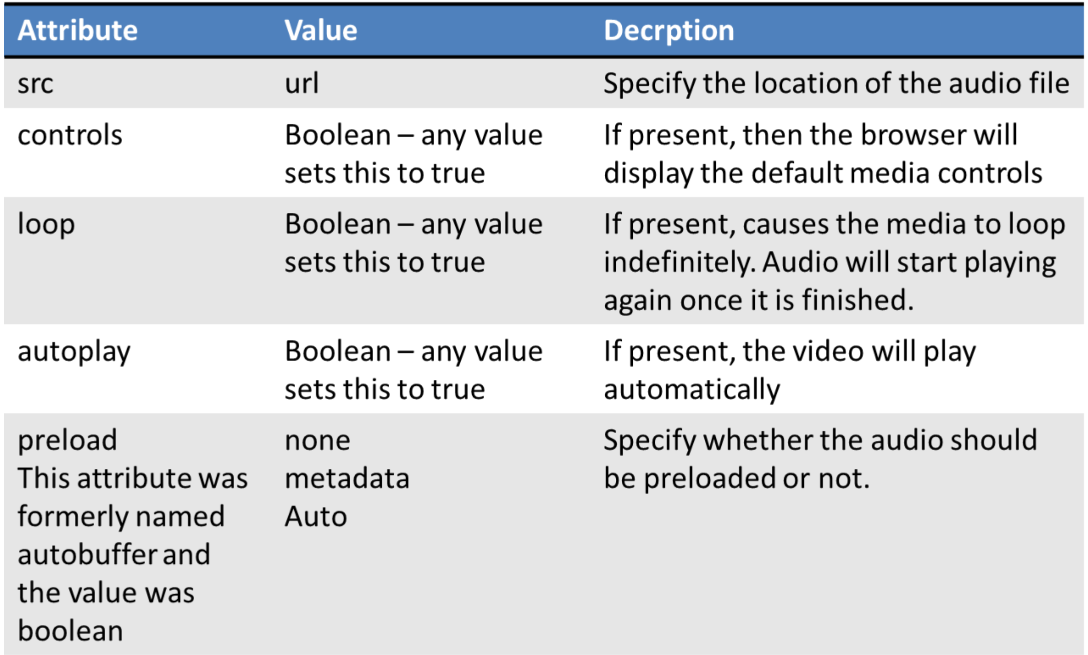
HTML 5 Media Tags

- Earlier versions of HTML required third party plug-ins such as Adobe Flash, Microsoft Silverlight and Google Gears to play video. Whereas, the new standard includes features like video playback to support video
- HTML5 specifies a standard way to embed a media file to a web page without using a extra plug-in.
- The tags used are:
  - Audio
  - Video
- Both have use <src> attribute to specify the source file and controls attribute to add controls like play, pause or stop.

Audio Tag

Media Tags: Audio attributes

- Audio tag supports following attributes:

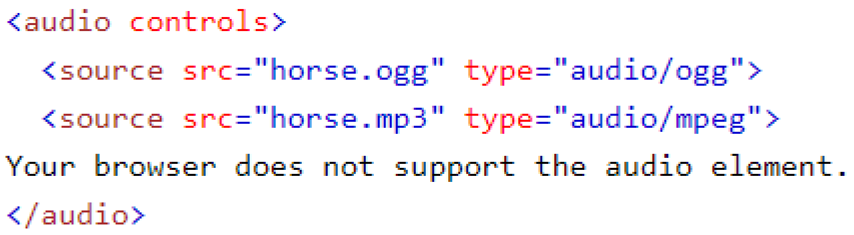


Attribute	Value	Decrption
src	url	Specify the location of the audio file
controls	Boolean – any value sets this to true	If present, then the browser will display the default media controls
loop	Boolean – any value sets this to true	If present, causes the media to loop indefinitely. Audio will start playing again once it is finished.
autoplay	Boolean – any value sets this to true	If present, the video will play automatically
preload This attribute was formerly named autobuffer and the value was boolean	none metadata Auto	Specify whether the audio should be preloaded or not.

- Audio tag does not support following attributes: height, width and poster

Media Tags: Audio attributes

Example:



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

Video Tag

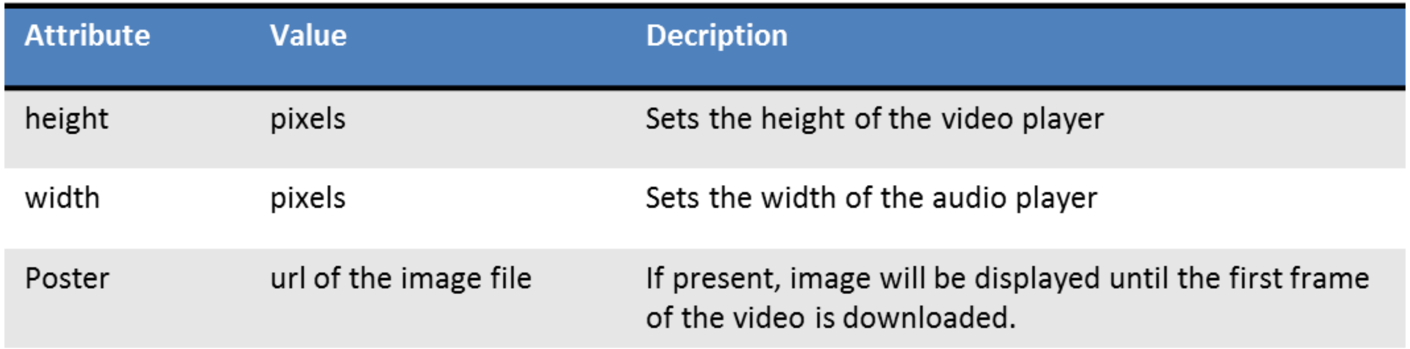
Media Tags: Video attributes

- Some more video attributes are:

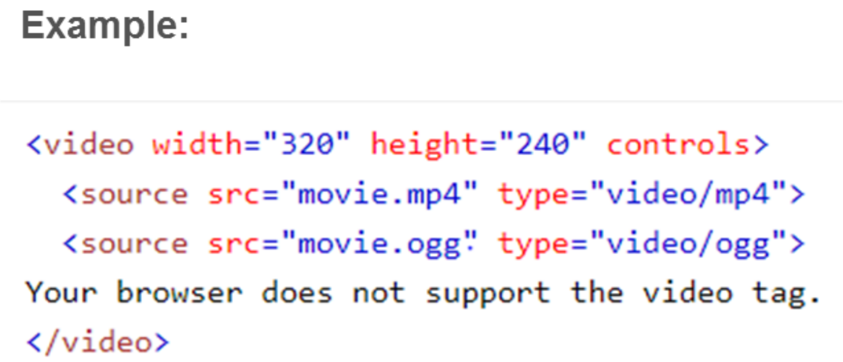
Attribute	Value	Decription
src	url	Specify the location of the video file
controls	Boolean – any value sets this to true	If present, then the browser will display the default media controls
loop	Boolean – any value sets this to true	If present, causes the media to loop indefinitely. Video will start playing again once it is finished.
autoplay	Boolean – any value sets this to true	If present, the video will play automatically
preload This attribute was formerly named autobuffer and the value was boolean	none metadata Auto	Specify whether the video should be preloaded or not.

- The preload attribute is an hint to the browser of your preference, even in the absence of the autoplay attribute. Ignored if autoplay is present.

Media Tags: Video attributes



Attribute	Value	Decription
height	pixels	Sets the height of the video player
width	pixels	Sets the width of the audio player
Poster	url of the image file	If present, image will be displayed until the first frame of the video is downloaded.



Example:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

Media Tags: Audio/Video for DOM

- HTML5 defines DOM methods, properties, and events for the **<video>** element.
- This allows you to load, play, and pause videos and set duration and volume.
- We can programmatically control the video or audio which is embedded in the html file using JavaScript.
- Various event handling media elements exist.

Example 1:

```
var playVideo= document.getElementsByTagName(‘audio')[0];
playVideo. play();
```

Example 2:

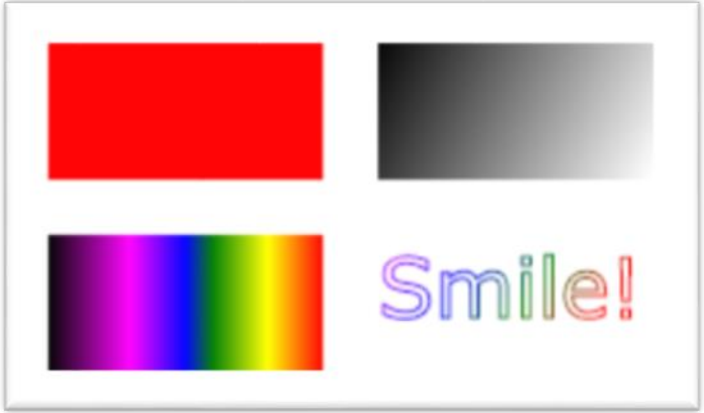
```
var playVideo= document.getElementsByTagName(‘video')[0];
playVideo. play();
```

Graphic Tags

Canvas API

Canvas API

- HTML5 **<canvas>** tag is used to draw graphics on the fly using scripting languages.
- Graphics below has a red rectangle, gradient rectangle, multicolored rectangle and a multicolor text created using **<canvas>**.

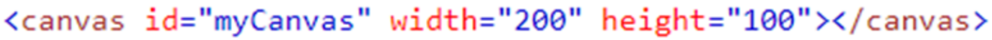


- **<canvas>** element is only a container for graphics. Scripts are used to draw the graphics.
- It has methods for drawing paths, boxes, circles, text and adding images

Canvas API

- <canvas> can used to draw:
  - Colorful text with or without animation.
  - Graphical data representation with an imagery of graphs and charts.
  - Gaming applications
  - Interactive by responding to user actions like key/mouse/button clicks, finger movement.
  - Simple animations like bouncing a ball to complex animation.

Example



```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

- <canvas> element must have id attribute to referred by JavaScript.
- width and height to define the size of canvas

Canvas API

Draw on the Canvas using JavaScript

- **Step 1:**
  - Find the **<canvas element>** using HTML DOM method **getElementById()**

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

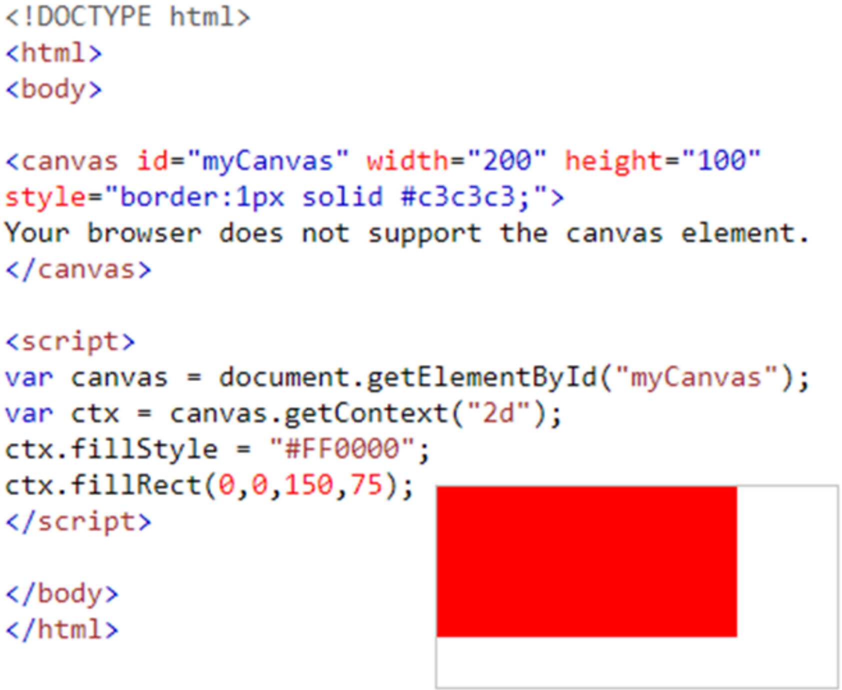
- **Step 2:**
  - Create a drawing object for canvas using **getContext()** with properties and methods for drawing

```
var ctx = canvas.getContext("2d");
```

- **Step 3:**
  - Draw on the canvas using the object created in Step 2

```
ctx.fillStyle = "#FF0000";
ctx.fillRect(0,0,150,75);
```

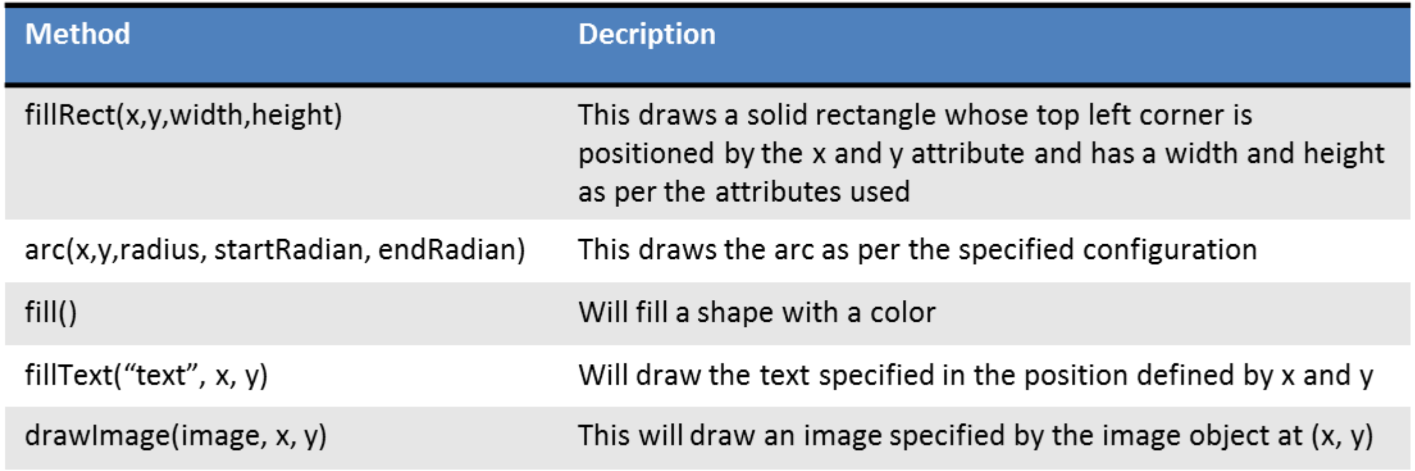
Example:



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

<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #c3c3c3;">
Your browser does not support the canvas element.
</canvas>

<script>
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
ctx.fillStyle = "#FF0000";
ctx.fillRect(0,0,150,75);
</script>

</body>
</html>
```

Canvas API

- Some of the simple methods available in canvas include



Method	Decription
fillRect(x,y,width,height)	This draws a solid rectangle whose top left corner is positioned by the x and y attribute and has a width and height as per the attributes used
arc(x,y,radius, startRadian, endRadian)	This draws the arc as per the specified configuration
fill()	Will fill a shape with a color
fillText("text", x, y)	Will draw the text specified in the position defined by x and y
drawImage(image, x, y)	This will draw an image specified by the image object at (x, y)

Scalable Vector Graphics (SVG) API

Scalable Vector Graphics (SVG)

- SVG refers to Scalable Vector Graphics.
- SVG is used to draw vector based graphics on the web page.
- Unlike canvas, SVG contents are interactive as all the elements in <svg> are available as objects.
- All browsers support the use of SVG.
- SVG can be associated with event handlers too.
- It is a language for defining 2D-graphical applications in XML. The XML is then rendered by an SVG viewer.
- SVG is useful for diagrams like Pie charts, 2D graphs in an X-Y coordinate system etc.
- Most of the web browsers can display SVG just like how they will display the formats PNG, GIF and JPG(With an exception some browsers need to install Adobe SVG Viewer to view SVG in them.)
- HTML5 allows embedding SVG directly using **<svg>…</svg>** tag.

Scalable Vector Graphics (SVG)

- 2 ways to create SVG:

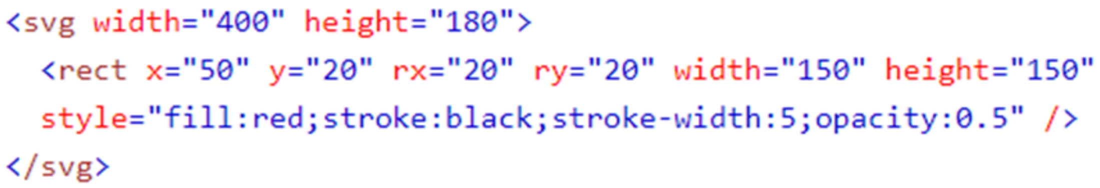
  1) Create the svg tags in the HTML page directly.

     (Useful for small primitive types like square, rectangle etc)

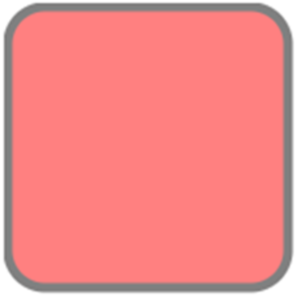
  2) Use tools like inkscape and create svg files and include it in HTML pages.

     (useful for complicated shapes)

Example:



```
<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
</svg>
```



Geo-location API

]

Geo-location API

- The geographical position of the user can be identified by the HTML5 Geo-location API
- The global navigator object (i.e) the geo-location object can be created as follows:

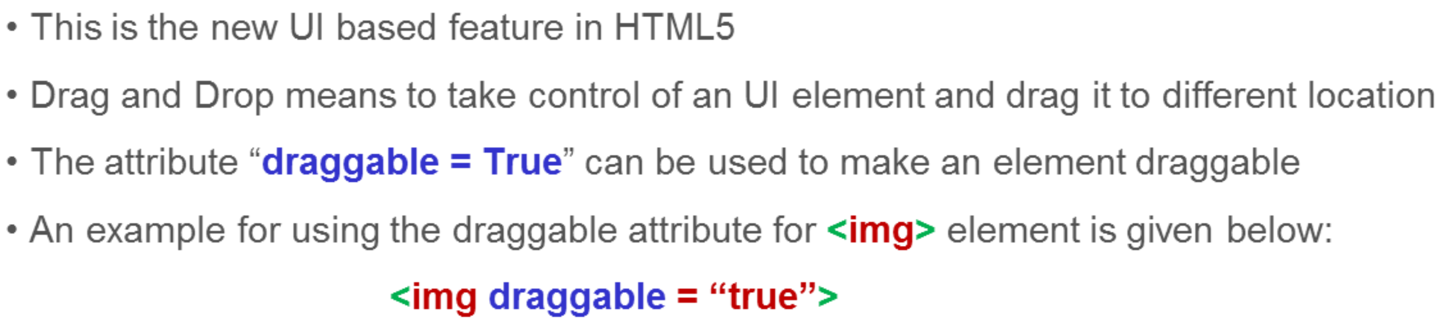
```
var geoobj=navigator.geolocation;
```

- The geolocation object allows devices to retrieve information about the geographical location of the device.
- The geolocation object supports the following methods:
  - getCurrentPosition()
  - watchPosition()
  - clearWatch()

Demo

Drag and Drop features

Drag and Drop



- This is the new UI based feature in HTML5
- Drag and Drop means to take control of an UI element and drag it to different location
- The attribute “**draggable = True**” can be used to make an element draggable
- An example for using the draggable attribute for **<img>** element is given below:

<img draggable = “true”>

Demo

Web Storage

Web Storage

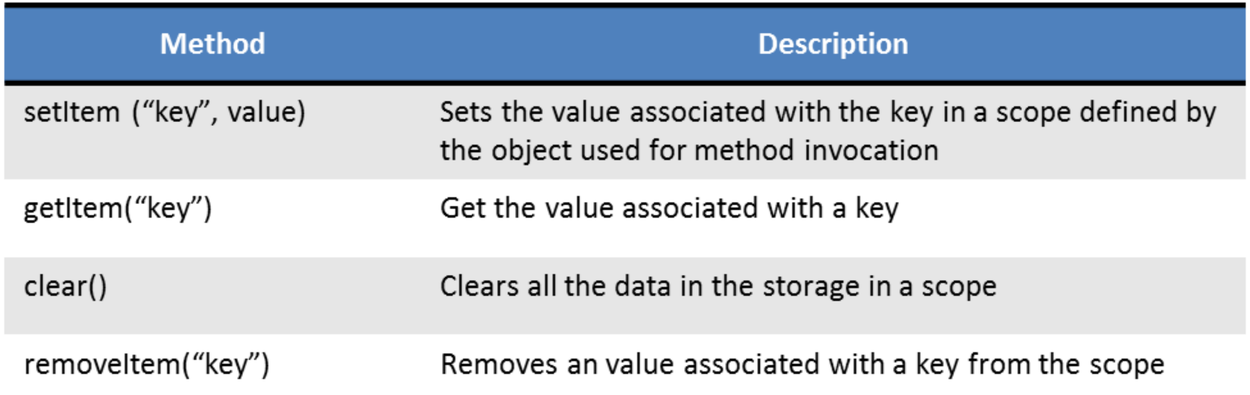
- Web applications store data locally within the user's browser.
- Prior to HTML5, Cookies were seen as a popular solution to store data/information at client side. But it had issues like size restrictions, increased request-response cycles and security considerations.
- Hence, web storage can be used to store data locally within the client's browser.
- Web storage is more secure and permits large amount of data storage locally, without affecting website performance.
- HTML5 spec provides a elegant solution for this problem by giving two types of storage:
  - Session storage – stores data for one session (when browser tab is closed data is lost)
  - Local storage – stores data with no expiration date

How does storage work?

- Each domain is given a part of the memory in clients file system accessible by the browser.
- This will ensure that one domain does not override/access the data of another domain
- The storage is done in the form of key, value pairs
- The key is always a String
- When we want to store some data on the client side for the duration of a session, we can use the Session Storage object.
- When we want to store some data beyond a session, then we can use Local Storage Object

Web Storage Methods

- Some of the key methods in Storage are:



Method	Description
setItem (“key”, value)	Sets the value associated with the key in a scope defined by the object used for method invocation
getItem(“key”)	Get the value associated with a key
clear()	Clears all the data in the storage in a scope
removeItem(“key”)	Removes an value associated with a key from the scope

Demo

Cascading Style Sheet (CSS3)

Introduction to CSS3

What is CSS3?

Style sheet markup language used to **separate document content** and **presentation**.

This is the newest version of CSS providing **new styling elements**.

Unlike the previous versions CSS3 is divided into **Modules**

CSS3 is **backward compatible**.

History

CSS1

- Officially recommended and published by W3C in **Dec 1996**.
- Provided capabilities like Font, Background, Margin, Border etc.

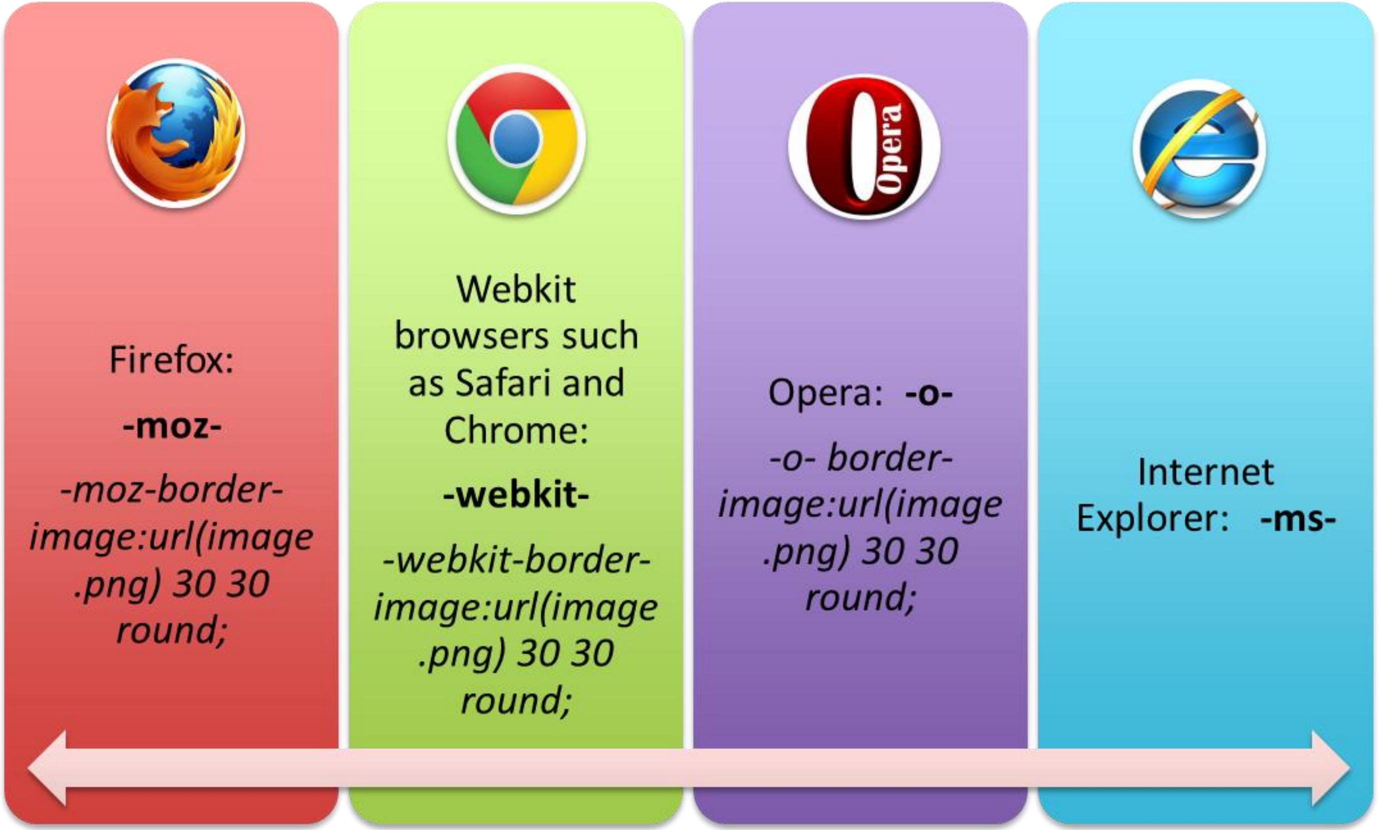
CSS2

- Published in **1998**, supported capabilities like new font properties, positioning of elements etc.
- CSS2.1 fixed errors and removed poorly supported features of CSS2.

CSS3

- Work had started during the publication of CSS2 recommendation.
- Draft was published in June **1999**.
- CSS3 has numerous modules, each of which have different stability and are in different status.
- As of NOV 2011 over **50 modules** have been published.

Vendor Prefixes



Firefox:

-moz-

-moz-border-image:url(image.png) 30 30 round;

Webkit browsers such as Safari and Chrome:

-webkit-

-webkit-border-image:url(image.png) 30 30 round;

Opera: **-o-**

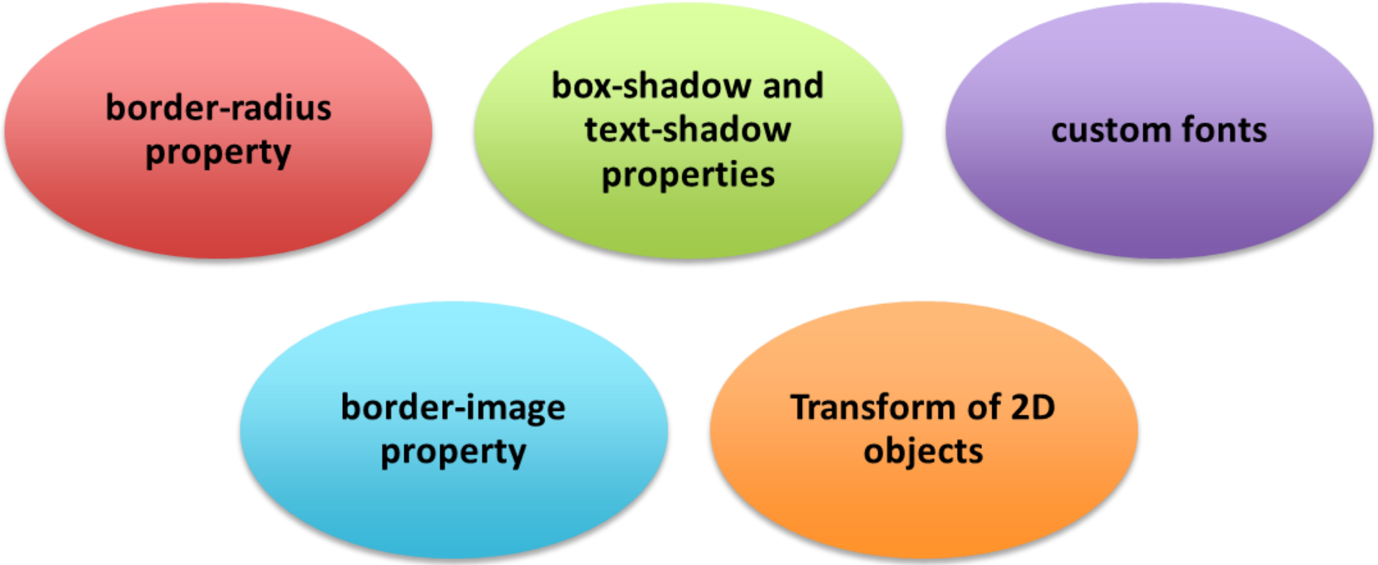
-o- border-image:url(image.png) 30 30 round;

Internet Explorer: **-ms-**

Popular Modules

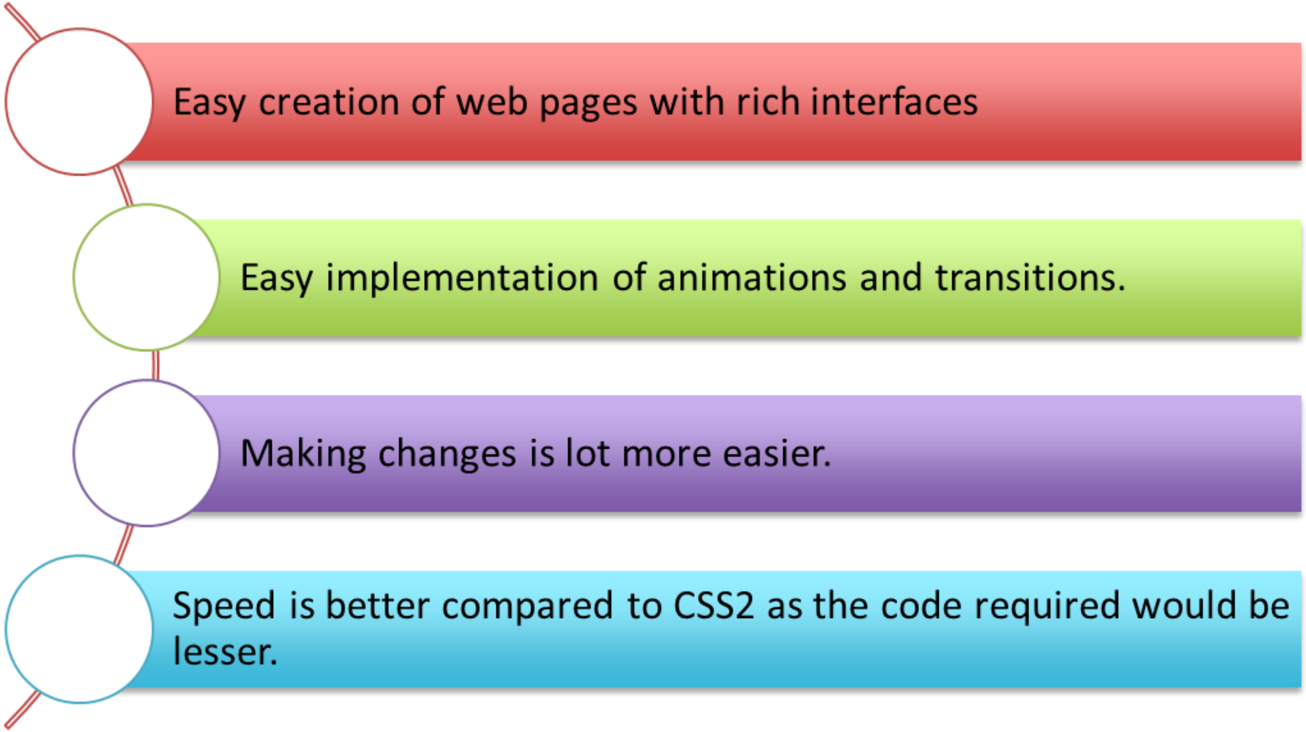
The Box Module	• Describes the characteristics of the flow and various kinds of boxes.
Lists Module	• Contains the list styling and Positioning.
Hyperlink Presentation	• Describes the presentation of hyperlink anchors and effects of hyperlink activation.
Backgrounds and Borders	• Describes borders consisting of images, boxes with multiple backgrounds, shadows etc.
Text Effects	• Describes the new properties added to the text layout.
Multi column layout	• Describes multi-column layout in CSS.
Fonts Module	• Describes how font properties are specified and how font resources are loaded dynamically.
2D & 3D transform	• Describes how CSS elements be transformed in 2D and 3D space.
Transitions	• Describes how property changes in CSS values occur smoothly and in a specified duration.
Animation	• Allows us to modify CSS property values over time.

Features in CSS3



- border-radius property
- box-shadow and text-shadow properties
- custom fonts
- border-image property
- Transform of 2D objects

Benefits of CSS3



- Easy creation of web pages with rich interfaces
- Easy implementation of animations and transitions.
- Making changes is lot more easier.
- Speed is better compared to CSS2 as the code required would be lesser.

CSS3 Basics

CSS3 Border Radius

Border–radius property is used to change the radius of each corner of the border.

Radius of individual corners can also be changed separately.

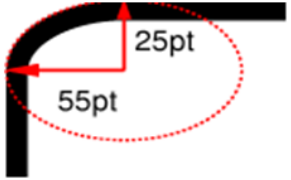
```
Div{
        Border: 3px solid #FF0000;
        Width: 500px;
        Height: 100px;
        Border-radius: 30px;
        }
```

CSS3 Border Radius

The curvature of a corner can be changed by providing the horizontal and vertical radii.

eg: Border-radius-topleft: 55pt 25pt;

```
Div{

     Border:2px  solid #FF0000

    Width:500px;  Height:  100px;

    -moz-Border-top-left-radius:  4em;

    -moz-Border-top-right-radius:1em;

    -moz-Border-bottom-right-radius:4em;

     -moz-Border-bottom-left-radius:1em;

       }
```



Border-radius:4em 1em;

CSS3 Box Shadow

Shadows can be applied to a box elements using **box-shadow property**.

The box-shadow property is supported by IE9+, Firefox 4, Chrome, Opera, and Safari 5.1.1.

Box-shadow: x-axis y-axis blur spread color inset;

```
div {
        width:300px;
        height:100px;
        background:red;
        -moz-box-shadow:10px 10px 2px
                              0px #998877;
         }
```

CSS3 Box Shadow

- Using inset the shadow can be made available inside the border.

```
div
    {
      width:300px;
      height:100px;
      border: 5px solid yellow;
      background: red;
      box-shadow: inset 10px 10px 2px #998877;
    }
```

CSS3 Box Shadow

- Using Spread property the shadow can be spread along a larger area.
- Eg:

```
div
    {
     width:300px;
     height:100px;
     background:red;
     box-shadow:10px  10px 2px 10px  #998877;
    }
```

CSS3 Image Border

- Values for border-image-repeat property

repeat : The image is made to repeat to fill the area.

stretch : stretches the image to fill the area.

round : The image is made to repeat to fill the area, if it doesn’t fill the area in whole numbers the image is rescaled to fit accordingly.(repeat and round are the same in chrome)

scale : The image is made to repeat to fill the area. If it doesn’t fill the area in whole numbers, extra space is provided around the tiles. (currently not supported by browsers)

CSS3 Background

- **CSS3 allows us to have images as background for not only webpages but also to other elements**.

- Properties used:
  - Background-size specifies the background image size.
  - Background-origin property specifies where the background-image needs to be placed. [values: padding-box, content-box, border-box]
  - Background-repeat property specifies the how the image is to be repeated. [values: repeat-x/y, repeat, space, round, no-repeat]
  - Background-position property specifies the initial position within the background area.[values: center/left/right/bottom or percentage or len]

CSS3 Background

- **CSS3 supports multiple backgrounds separated by comma.**

```
div
      {
     border:15px solid black;
     font-size:30px;
     height: 100px;
     width:800px;
     padding:35px;
     background-image: url('windows.jpg'), url(‘img1.gif);
     background-repeat:repeat-y, repeat-x;
     background-position: left, right;
     }
```

CSS3 Text Effects

- **Text Shadow**

  CSS3 allows us to add shadow to each letter of text using text- shadow property.

- **Syntax:** ‘horizontal distance‘ ‘vertical distance’ ‘blur’ ‘color’

- Ex:

```
H2{
text-shadow: 3px 3px 5px #0000FF;
}
```

CSS3 Text Effects

- **CSS3 also supports usage of multiple shadows for text.**

- h2{color: black;

  text-shadow: 0.2em 0.3em 0.1em #FF0000,

  0.8em 0.2em 0.1em #00FF00 , 0.3em -0.2em 0.1em #0000FF}

- Glow effects can be produced by keeping horizontal and vertical distances as zero and giving a high value to blur.

- Eg: h2{color: black;

  text-shadow: 0em 0em 8em #FF0000}

CSS3 Word Wrapping

- If a long word is present and cannot be accommodated inside a box width, the word extends beyond the boundary of the box, to avoid this the word can be split and be accommodated inside the box using 'word-wrap' property.
- Eg: p{

```
        width: 10px;
        border: 2px solid #FF0000;
        word-wrap: break-word;
        }
```

- A line can also be broken by using text-wrap property, but presently none of the browsers support this property.

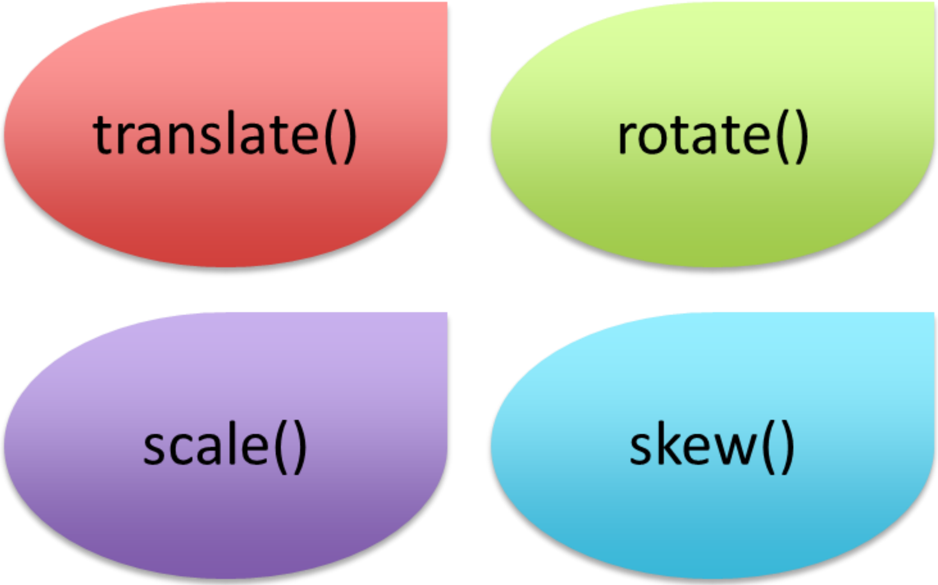
CSS3 Web Fonts

- Previous versions of CSS allowed only the usage of fonts which were already available in the user’s computer, CSS3 allows the user to use fonts of his/her own choice.
- The path of **True Type Font (.ttf)** or **Embedded Open Type (.eot)** font file has to be given with the @font-face rule available in CSS3.
- With the @font-face rule a name for the font has to be provided.
- Font type can be changed to another when font-style, font-weight or font-stretch change.

CSS3 Transforms

CSS3 2D Transform

- Using 2D transform property to change shape, size and position of an element.
- Transformation requires use of browser prefixes.
- 2D transform methods:



translate()

rotate()

scale()

skew()

Translate Method

- Used to place an element at a specific position.
- X-axis and Y-axis are the parameters required.
- Y-axis is an optional parameter, If not given will take a value of zero.
- Syntax: transform: translate(x-axis , y-axis);

Rotate Method

- Used to rotate an element by an angle specified.

Syntax: “transform: rotate(x deg/rad);”

- **Ex**: #div2

```
{
    transform:rotate(30deg);
    -ms-transform:rotate(30deg);         /* IE */
     -moz-transform:rotate(30deg);       /* Firefox */
     -webkit-transform:rotate(30deg);    /* Safari, chrome*/
     -o-transform:rotate(30deg);         /* Opera */
}
```

Scale Method

- Used to increase or decrease the size of an element.
- Requires x-axis and y-axis.

```
syntax:   transform: scale(2,4);
```

- Also can be used for providing value for only x-axis or y-axis using scaleX() and scaleY().

Skew Method

- Used to turn an element by a certain angle.
- The element can be turned along x-axis and y-axis.
- Syntax:

```
transform: skew(x-axis deg/rad, y-axis deg/rad);
```

3D Transform Properties

- Transform-origin : Allows us to change the position of transformed elements.
- Syntax:

```
transform-origin: x-axis   y-axis   z-axis;
```

- X and Y axis can be specified as left, right, center or length or percentage. Z axis has to be specified in length.
- This property is supported by all the browsers.

3D Transforms

- rotateX() and rotateY() are the methods used to rotate an element in 3 dimension.
- The property is supported by only Webkit browsers chrome and safari.

CSS3 Transitions

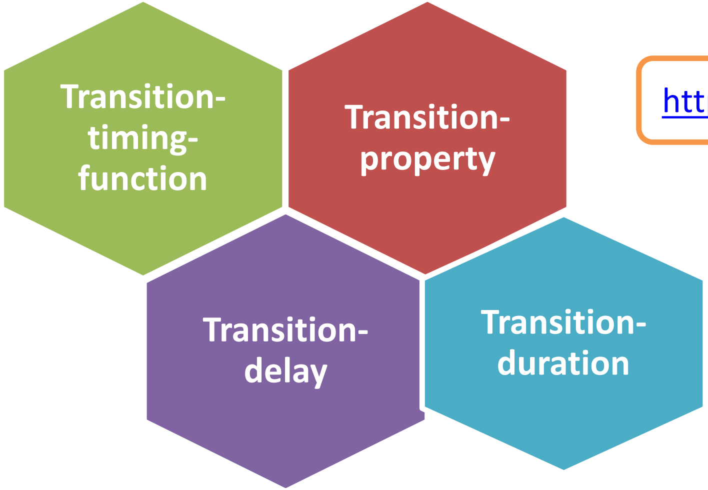
CSS3 Transition

EFFECTS to Change the Style

IE Brower - Incompatible

- C Choose the property on which transition needs to be applied.
- S Specify the duration for the effect. Default value is 0.
- S Specify the transform for the element and also when.

CSS3 Transition Properties



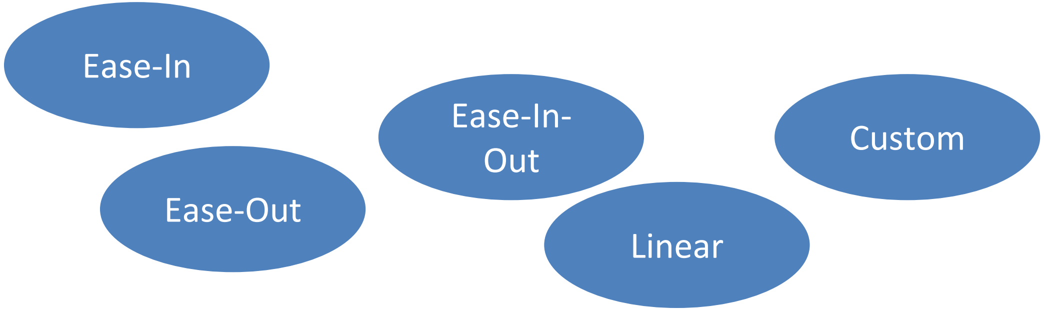
Transition-timing-function

Transition-property

Transition-delay

Transition-duration

http://matthewlein.com/ceaser/



Ease-In

Ease-Out

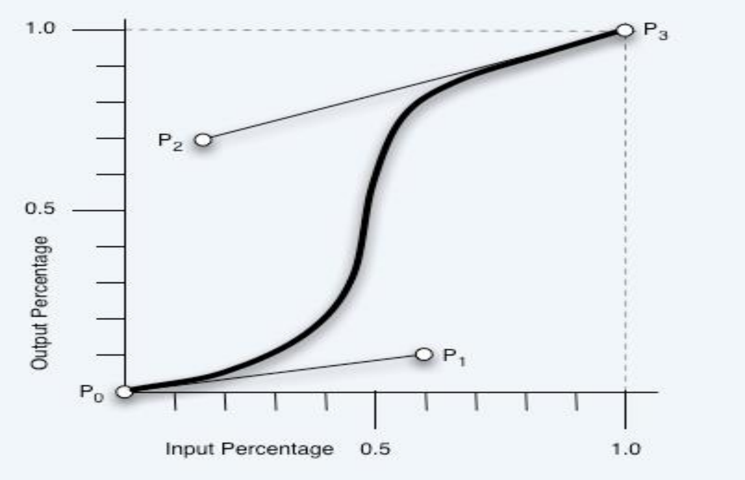
Ease-In-Out

Linear

Custom

CSS3 Transition Properties

cubic-bezier (p0,p1,p2,p3) : transition-timing-function can also be specified using cubic-bezier function, we have to specify 4 values using cubic-bezier curve.



CSS3 Transition Properties

- The other transition-timing function values can also be provided using cubic-bezier function

  eg: linear = cubic-bezier(0.0, 0.0, 1.0, 1.0)

  ease = cubic-bezier(0.25, 0.1, 0.25, 1.0)

- P0-P3 values can be got after setting the speed curve using tools available on net.

  eg: http://cssglue.com/cubic

CSS3 Transition Properties

Transition: All the four transition properties can be included in this single property.

Syntax:

‘transition-property’,‘transition-duration’,‘transition-timing-function’ ‘transition-delay’

Eg: Transition: length 10s ease-in 3s;

CSS3 Transitions – Browser Compatibility

Browser Compatibility					
Property	IE	Firefox	chrome	Safari	Opera
Transition-property	☒	☑	☑	☑	☑
Transition-duration	☒	☑	☑	☑	☑
transition-timing-function	☒	☑	☑	☑	☑
transition-delay	☒	☑	☑	☑	☑
transition	☒	☑	☑	☑	☑

CSS3 Animation

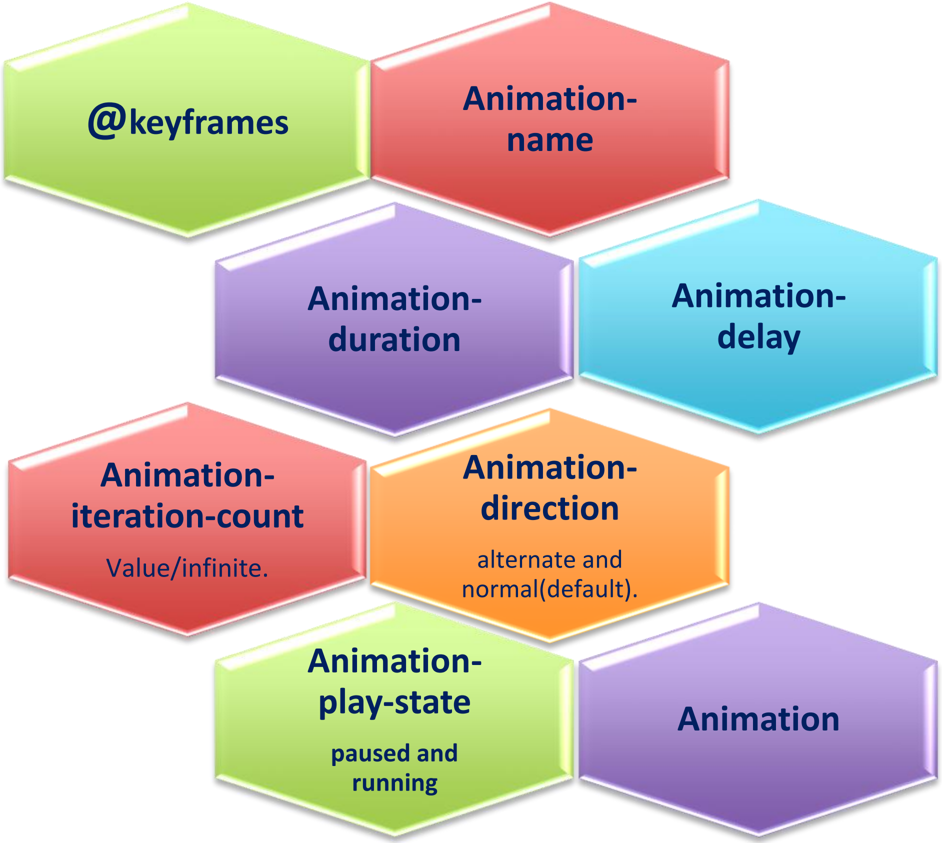
Animation

Animation without Javascript

IE and Opera are Incompatible

@keyframes

Properties used for Animation



@keyframes

Animation-name

Animation-duration

Animation-delay

Animation-iteration-count

Value/infinite.

Animation-direction

alternate and normal(default).

Animation-play-state

paused and running

Animation

Properties used for Animation

- **Animation:** This is used as a short hand property to provide the values of all the properties in one.

  syntax: animation: name duration timing-function delay

  iteration-count direction

Properties used for Animation

Property	IE	Firefox	Chrome	Safari	Opera
Animation-name	☒	☑	☑	☑	☒
Animation-duration	☒	☑	☑	☑	☒
Animation-timing-function	☒	☑	☑	☑	☒
Animation-delay	☒	☑	☑	☑	☒
Animation-iteration-count	☒	☑	☑	☑	☒
Animation-direction	☒	☑	☑	☑	☒
Animation-play-state	☒	☑	☑	☑	☒

Summary

- Introduction to HTML5 for Beginners
- What's new in HTML5?
- Advantage of HTML5 over HTML4
- Use HTML elements to add semantics to content of web page
- Designing Graphics using Canvas API
- Geo-location API
- Drag and Drop features
- Web Storage
- Apply new features of CSS3 in web pages
- Apply border-radius, box-shadow, text-shadow and border image effects
- Apply 2D transformations, transitions and animations

References

- https://www.w3schools.com/html/default.asp
- http://www.css3.info/
- https://www.w3schools.com/w3css/default.asp
- https://www.tutorialspoint.com/html5/index.htm
- https://www.tutorialspoint.com/css/css3_tutorial.htm
- https://html5demos.com/
- http://html5test.com/
- http://diveinto.html5doctor.com/
- Mark Piligrim, HTML 5.- Up and Running, O’Reilly Publications and Google Press
- Alexis Goldstein, Louis Lazaris, and Estelle Weyl, HTML5 and CSS3 in The Real World, SitePoint Pty. Ltd, 2011.

Thank You

● ● ●

Infosys *be more*