Unit 1 : Introduction to Html and CSS

Prepared By:

Chirag Patel

Lecturer Computer Engineering

K. D. Polytechnic, Patan

History of Html

- **Hypertext**
- Hypertext is text that links to other information. By clicking on a link in a hypertext document, a user can quickly jump to different content. So hypertext is usually associated with Web pages.
- **Hyperlink**
- A hyperlink is a text or image that you can click on to jump to a new document. When you move the cursor over a hyperlink, whether it is text or an image, the arrow should change to a small hand pointing at the link. When you click it, a new page or place in the current page will open.
- **HTML**
- HTML Stands for Hypertext Markup Language. HTML is the language used to create web pages. "Hypertext" refers to the hyperlinks that an HTML page may contain.
- "Markup language" refers to the way tags are used to define the page layout and elements within the page.
- HTML contains elements (tags) that indicate how a web browser should present the document.

History of Html

- HTML contains elements (tags) that indicate how a web browser should present the document. An HTML document is simply a text file that contains information, which is to be published. The collection of HTML pages makes the World Wide Web.

- HTML pages can be created using simple text editor such as Notepad. The browser interprets this text file (.htm/.html) and renders on the client computer.

- In 1989 HTML was originally developed by Tim Berners-Lee. It is derived from SGML (Standard Generalized Markup Language).

- **HTTP**

- HTTP Stands for Hypertext Transfer Protocol. HTTP is the protocol used to transfer data over the web.

- HTTP uses a server-client model. When you access a website, your browser sends a request to the corresponding web server and it responds with an HTTP status code. If the URL is valid and the connection is granted, the server will send your browser the webpage and related files.

Basic HTML Tags

- **<html>**
- This tag is used to indicate that this is a HTML document. HTML documents should start and end with this tag.
- **<head>**
- This tag is used to indicate the header section of the HTML document, which typically includes the
- <title> and <meta> tags.
- **<title>**
- This indicates the title of this HTML page. The title is what is displayed on the upper left corner of your browser when you view a web page.
- **<body>**
- The <body> tag includes the HTML body of the document. Everything inside the <body> tag is displayed on the browser.

Basic HTML Tags

- **<html>**
- This tag is used to indicate that this is a HTML document. HTML documents should start and end with this tag.
- **<head>**
- This tag is used to indicate the header section of the HTML document, which typically includes the
- <title> and <meta> tags.
- **<title>**
- This indicates the title of this HTML page. The title is what is displayed on the upper left corner of your browser when you view a web page.
- **<body>**
- The <body> tag includes the HTML body of the document. Everything inside the <body> tag is displayed on the browser.
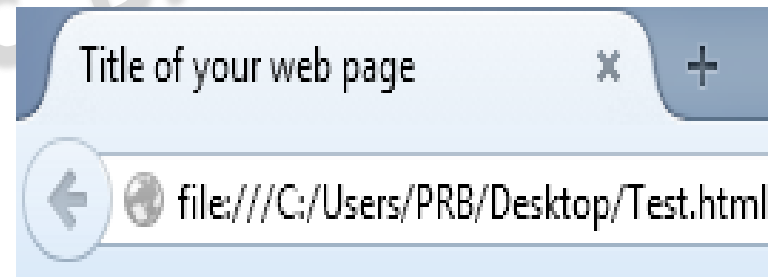
Basic HTML Tags

<html>

<head>

<title>Title of your web page</title>

</head>

<body>HTML web page contents</body>

</html>

Output:

HTML web page contents

Basic HTML Tags

- **<hr>**
- The <hr> tag is used for creating a horizontal line. This is also called Horizontal Rule in HTML.
- The HTML <hr> tag also supports following additional attributes:
  - **align** - Specifies the alignment of the horizontal rule. Value for this attributes is Left, right, center.
  - **size** - Specifies the height of the horizontal rule.
- **width** - Specifies the width of the horizontal rule.
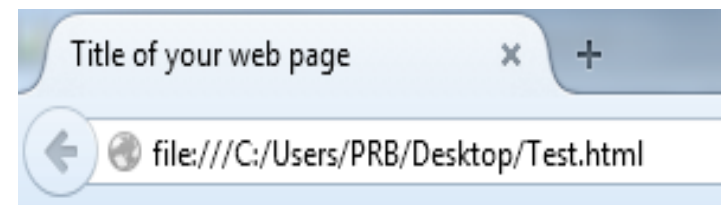
<html>
<head><title>Title of your web page</title></head>
<body>
HTML web page contents
<hr align="left" width="100px">
</body>
</html>

Title of your web page   ×   +
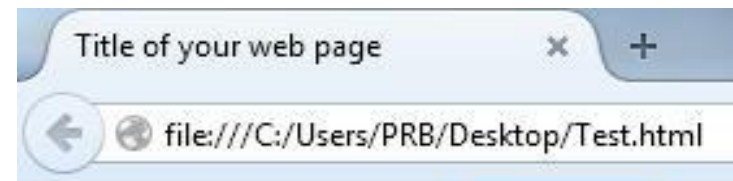
file:///C:/Users/PRB/Desktop/Test.html

HTML web page contents

Basic HTML Tags

- **<img>**
- The <img> tag is used to embed an image on the HTML document. The attributes are as follows:
  - **src**: The file path to the image file.
  - **width**: The width of the image, in pixels.
  - **height**: The height of the image, in pixels.
  - **alt**: This attribute is used to give alternative text that can be displayed in place of image.

```
<html>
<head><title>Title of your web page</title></head>
<body>HTML web page contents
<img src="Penguins.jpg" width="80px" height="80px" alt="text describing the image" />
</body>
</html>
```

Title of your web page   × | +

file:///C:/Users/PRB/Desktop/Test.html

HTML web page contents

Basic HTML Tags

- **<embed>**
- The <embed> tag defines a container for an external application or interactive content (a plug-in).
- height
  - pixels
  - Specifies the height of the embedded content
- src
  - URL
  - Specifies the address of the external file to embed
- type
  - media_type
  - Specifies the media type of the embedded content
- width
  - pixels
  - Specifies the width of the embedded content

<html>
<head><title>Title of your web page</title></head>
<body>
<embed src="Networking Basics.mp4" ></embed>
</body>
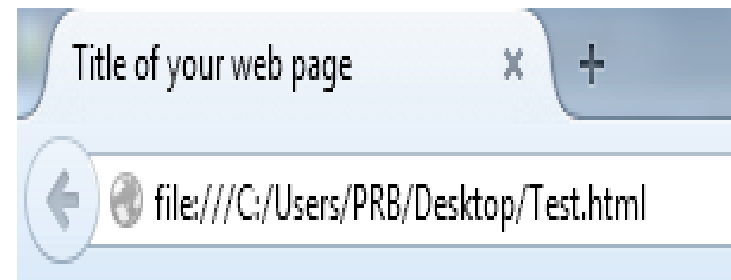</html>
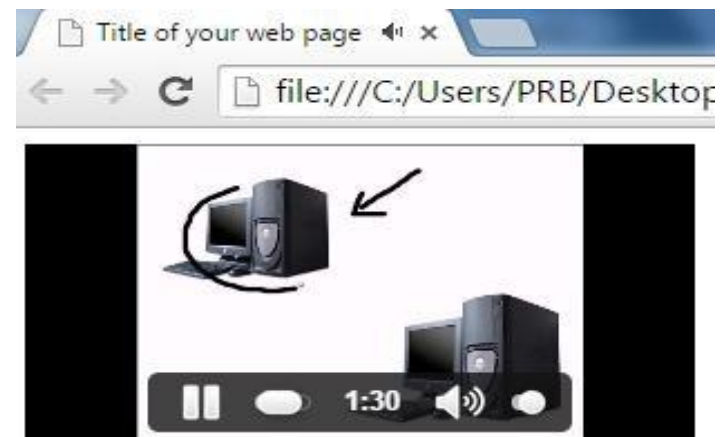
Basic HTML Tags

- **<bgsound>**
- The HTML <bgsound> tag is used to play a soundtrack in the background. This tag is for Internet Explorer only.
- loop
  - number
  - Lets you replay a background soundtrack a certain number of times.
- src
  - URL
  - Specifies the path of the sound file.

  <bgsound src="A.mp3">
- **<blink>**
  - The HTML <blink> tag is used to enclose a text to make it blink. This tag was supported by Netscape.
  - <blink>This text will blink in Netscape Version 5.0</blink>

Basic HTML Tags

- **<font>**

- The <font> tag is used to change the format of the text on the web page.

- Attributes of this tag:
  - **face**: The type of font.
  - **size**: This indicates the size of the text. This can be absolute (0 .. 6).
  - **color**: This indicates the color of the text. Either the color name or the six-character color.

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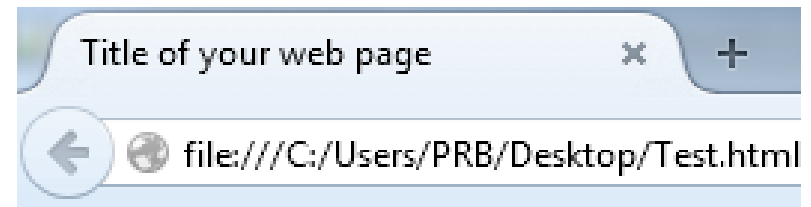
Title of your web page × +

file:///C:/Users/PRB/Desktop/Test.html

HTML web page contents Hello

Basic HTML Tags

- **<center>**
- The <center> tag causes all the text within the tag to be centered. An example is as follows:
- <center>This is centered text.</center>
- **<marquees>**
- An HTML marquee is a scrolling piece or block of text displayed either horizontally across or vertically down your webpage depending on the settings. This is created by using HTML <marquees> tag.
- Attributes of this tag:
  - **Width** - This specifies the width of the marquee.
  - **Height** - This specifies the height of the marquee.
  - **Direction** - This specifies the direction in which marquee should scroll. This can be a value like up, down, left or right.
- <marquee direction="up">This text will scroll from bottom to up</marquee>
- <marquee direction="right">This text will scroll from left to right</marquee>
- **<a>**
- This tag used to create an anchor link and always a preferred way of creating links.
- Attribute of this tag:
  - **href** - Defines URL of the liked page
  - Next Page

Basic HTML Tags

List Tags

- ###

  - The HTML <ul> tag is used for creating an unordered list. Important attribute for this tag:
  - **type** - Specifies the style of the bullet. Value for this attributes is Disc, circle, square.

```
<html>
<head><title>Title of your web page</title></head>
<body>
<ul type="square">
<li>Cricket</li>
<li>Football</li>
<li>Hockey</li>
</ul> </body>
</html>
```

Title of your web page   ×   +

file:///C:/Users/PRB/Desktop/Test.html

- ■ Cricket
- ■ Football
- ■ Hockey

Basic HTML Tags

List Tags

- **<li>**

  - The HTML <li> tag is used for specifying a list item in ordered, unordered, directory, and menu lists.

```
<html>
<head><title>Title of your web page</title></head>
<body>
<ul type="A">
<li>Cricket</li>
<li>Football</li>
<li>Hockey</li>
</ul> </body>
</html>
```

Title of your web page   ×   +

file:///C:/Users/PRB/Desktop/Test.htm
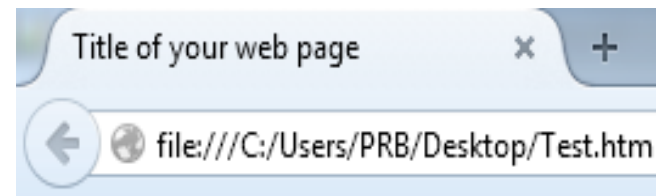
A. Cricket
B. Football
C. Hockey

Basic HTML Tags

List Tags

- ##
  - The HTML <ol> tag is used for creating an ordered list.

Attribute	Value	Description
reversed	reversed	Specifies the order of the list (descending).
start	number	Specifies the initial number to start the list.
type	A, a, I, i, 1	Specifies the style of the list.

```
<html>
<head><title>Title of your web page</title></head>
<body>
<ol type="1" reversed start=5>
<li>Cricket</li>
<li>Football</li>
<li>Hockey</li>
</ol> </body>
</html>
```

Title of your web page

file:///C:/Users/PRB/Desktop/Test.htr

5. Cricket
4. Football
3. Hockey

Basic HTML Tags

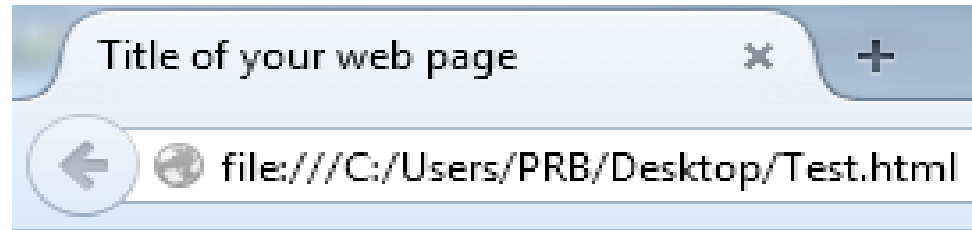- **<table>**

- The <table> tag specifies the table. This is very often used with the <tr> and the <td> tags. The following attributes are commonly used to define the properties of this table:
  - **width**: This specifies the width of the table.
  - **border**: This specifies table border. The number indicates the thickness of the border.
  - **cellspacing**: The amount spacing between the cell wall and the cell border.
  - **cellpadding**: The amount padding between cells and the each cell wall in a table.
  - **bgcolor**: This specifies the background color. The color value as color name or color code.

Basic HTML Tags

- **<tr>**
  - The <tr> tag specifies the presence of a row. The following attributes are commonly used to define the properties of this row:
    - **bgcolor**: This specifies the background color for this row.
    - **height**: This specifies the height of the row.
    - **rowspan**: This specifies the number of rows this particular row occupies.

- **<td>**
  - The <td> tag specifies the presence of a column. Columns are specified within each row. The following attributes are commonly used to define the properties of this column:
    - **valign**: This specifies how the text is vertically aligned inside the column. Common values include "top", "middle", and "bottom."
    - **width**: This specifies the width of the column.
    - **colspan**: This specifies the number of columns this particular column occupies.

Basic HTML Tags

```html
<html>
<head><title>Title of your web
    page</title></head>
<body>
<table border=1>
<tr>
    <td>Element 1</td>
    <td>Element 2</td>
    <td>Element 3</td></tr>
<tr>
    <td colspan=2>Element 4</td>
    <td>Element 5</td></tr>
<tr>
    <td>Element 1</td>
    <td>Element 2</td>
    <td rowspan=2>Element 3</td>
</tr>
<tr><td>Element 1</td>
<td>Element 2</td>
</tr>
</table>
 </body>
</html>
```

Title of your web page ✕ +

← 🌐 file:///C:/Users/PRB/Desktop/Test.html

Element 1	Element 2	Element 3
Element 4		Element 5
Element 1	Element 2	Element 3
Element 1	Element 2	

Basic HTML Tags

- **<div>**

- The HTML <div> tag is used for defining a section of your document. With the div tag, you can group large sections of HTML elements together and format them with CSS.

```
<html>
<head><title>Title of your web page</title></head>
<body>
<div style="color:Red">
<h3>This is a heading in a div element</h3>
<p>This is some text in a div element.</p>
</div> </body>
</html>
```

Title of your web page   ×   +

file:///C:/Users/PRB/Desktop/Test.html
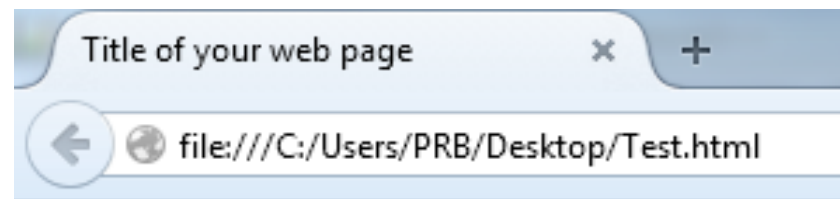
This is a heading in a div element

This is some text in a div element.

Basic HTML Tags

- **ID**
 - The **id** attribute of an HTML tag can be used to uniquely identify any element within an HTML page.
  - If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.
    - <p id="html">This para explains what is HTML</p>
    - <p id="css">This para explains what is Cascading Style Sheet</p>

Basic HTML Tags

- ## class
  - The class attribute specifies one or more class names for an element.
  - The class attribute is mostly used to point to a class in a style sheet.

Title of your web page   ✕   +

file:///C:/Users/PRB/Desktop/Test.html
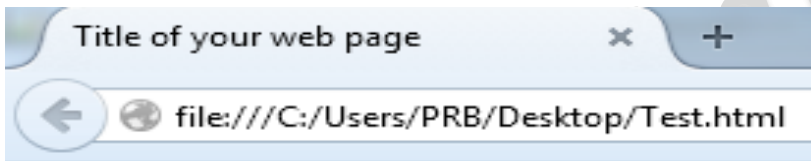
Header 1

A paragraph.

Note that this is an important paragraph.

Hello World

```
<html>
<head><title>Title of your web page</title>
<style>
h1.intro { color: blue; } p.important {  color:
green; }
</style>
</head>
<body>
<h1 class="intro">Header 1</h1>
<p>A paragraph.</p>
<p class="important">Note that this is an
important paragraph. </p>
<h3 class="important">Hello World</h3>
</body></html>
```

Basic HTML Tags

- ## Special Character or tags
  - Special characters and symbols not found on the keyboard can be inserted using special character entities. These characters are recognized in HTML as they begin with an ampersand (&) and end with a semi-colon. Following are some special character:

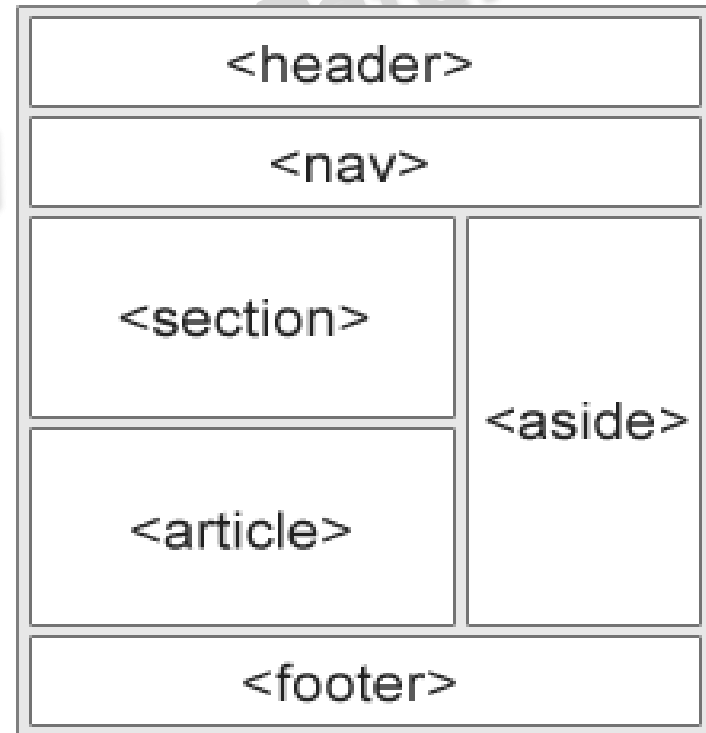| Sr. No.	Result	Description	Entity Name
1		non-breaking space	
2	<	less than	<
3	>	greater than	>
4	&	ampersand	&
5	¢	cent	¢
6	£	pound	£
7	¥	yen	¥
8	€	euro	€
9	©	copyright	©
10	®	registered trademark	®

Difference between HTML 4.0 & 5.0

HTML 4	HTML 5
HTML 4 uses common structures like headers, footers.	HTML5 uses new structures such as drag, drop and much more.
HTML 4 cannot embed video or audio directly. It makes use of flash player for it.	HTML 5 can contain embedded video and audio without using flash player.
In HTML 4, local storage is not possible and tags that can handle only one dimension are present.	In HTML 5, new tags and new features like local storage are enhanced.
HTML 4 cannot handle inaccurate syntax.	HTML 5 is capable of handling inaccurate syntax.
Vector Graphics is possible with the help of technologies such as VML, Silverlight, Flash etc.	Vector graphics is integral part of HTML5 e.g. SVG and canvas.
Browser cache can be used as temporary storage.	Application Cache, Web SQL database and Web storage is available as client side storage.
Works with all old browsers.	Most of modern browser have started supporting HTML5 specification e.g. Firefox, Mozilla, Opera, Chrome, Safari etc.

New Semantic Elements in HTML5

- Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.
- HTML5 offers new semantic elements to define different parts of a web page:
  - <article>
  - <aside>
  - <footer>
  - <header>
  - <nav>
  - <section>
  - <main>
  - <summary>
  - <time>
  - <figure>

<header>
<nav>

<section>	<aside>
<article>	

<footer>

New Semantic Elements in HTML5

- **<section>**
    - The <section> element defines a section in a document.
    - A section is a grouping of content, typically with a heading.
    - A Web site's home page could be split into sections for introduction, content, and contact information.
- **<article>**
    - The <article> element specifies independent, self-contained content.
    - It should be possible to read it independently from the rest of the web site.
    - Examples of where an <article> element can be used:
        - Forum post
        - Blog post
        - Newspaper article
- **<header>**
    - The <header> element specifies a header for a document or section.
    - The <header> element should be used as a container for introductory content.
    - You can have several <header> elements in one document.

New Semantic Elements in HTML5

- **<footer>**
    - The <footer> element specifies footer for a document or section.
    - Footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.
    - You can have several <footer> elements in one document.

- **<nav>**
    - The <nav> element defines a set of navigation links.
    - The <nav> element is intended for large blocks of navigation links. However, not all links in a document should be inside a <nav> element.

- **<aside>**
    - The <aside> element defines some content aside from the content it is placed in (like a sidebar).
    - The aside content should be related to the surrounding content.

Example of New Semantic Elements in HTML5

```html
<header>
<h1>This is page heading</h1>
</header>
<nav>
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">About Us</a></li>
<li><a href="#">Contact Us</a></li>
</ul>
</nav>
<article>
<h1>This is article heading</h1>
<p>Hello world! Hello world! Hello world!</p>
</article>
<aside>
<figure><img src="Penguins.jpg"
    height="100px" width="100px" /></figure>
</aside>
<section>
<h1>This is a section heading</h1>
<p>Hello world! Hello world! Hello world! </p>
</section>
<footer><hr />Copyright (C) 2013. All rights
    reserved. </footer>
```


Title of your web page

file:///C:/Users/PRB/Desktop/Test.html

This is page heading

- Home
- About Us
- Contact Us

This is article heading

Hello world! Hello world! Hello world!

This is a section heading

Hello world! Hello world! Hello world!

Copyright (C) 2013. All rights reserved.

HTML5 Form Input Types

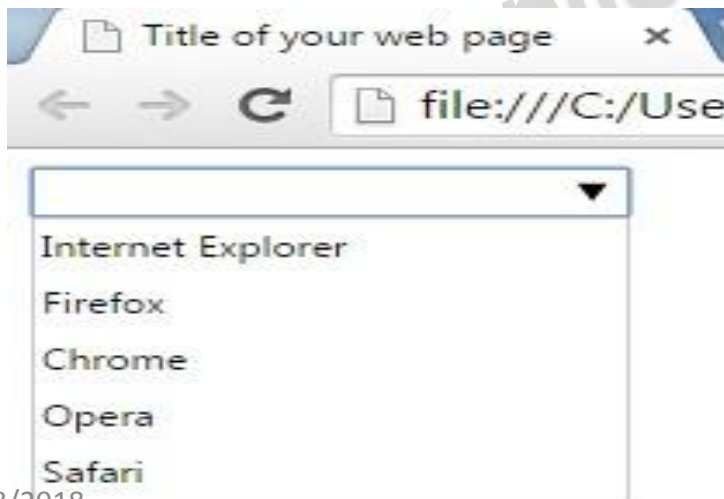| Input Type	Description	Example
number	It is used for input fields that should contain a numeric value.	Quantity (between 1 and 5): <input type="number" name="quantity" min="1" max="5">
Date	It is used for input fields that should contain a date.	Birthday: <input type="date" name="bday">
color	It is used for input fields that should contain a color.	Select your favorite color: <input type="color" name="favcolor">
range	It is used for input fields that should contain a value within a range.	<input type="range" name="points" min="0" max="10">
month	It allows the user to select a month and year.	Birthday (month and year): <input type="month" name="bdaymonth">
week	It allows the user to select a week and year.	Select a week: <input type="week" name="week_year">
time	It allows the user to select a time (no time zone).	Select a time: <input type="time" name="usr_time">

HTML5 Form Input Types

datetim e	It allows the user to select a date and time (with time zone).	Birthday (date and time): <input type="datetime" name="bdaytime">
datetim e-local	It allows the user to select a date and time (no time zone).	Birthday (date and time): <input type="datetime-local" name="bdaytime">
email	It is used for input fields that should contain an e-mail address.	E-mail: <input type="email" name="email">
search	It is used for search fields (a search field behaves like a regular text field).	Search Google: <input type="search" name="googlesearch">
tel	It is used for input fields that should contain a telephone number.	Telephone: <input type="tel" name="usrtel">
url	It is used for input fields that should contain a URL address.	Add your homepage: <input type="url" name="homepage">

HTML5 Form Elements

- **<datalist>**
- The **<datalist>** element specifies a list of pre-defined options for an <input> element.
- Users will see a drop-down list of pre-defined options as they input data.
- The **list** attribute of the <input> element, must refer to the **id** attribute of the <datalist> element.

<html>
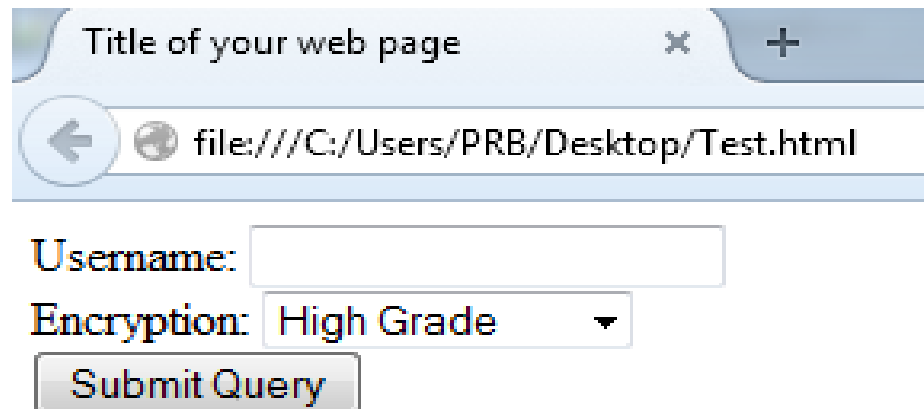<head><title>Title of your web page</title>
</head>
<body>
<input list="browsers" name="browser">
<datalist id="browsers">
<option value="Internet Explorer">
<option value="Firefox">
<option value="Chrome">
<option value="Opera">
<option value="Safari">
</datalist>
</body></html>

HTML5 Form Elements

- ## **<keygen>**
  - The purpose of the **<keygen>** element is to provide a secure way to authenticate users.
  - The <keygen> element specifies a key-pair generator field in a form.
  - When the form is submitted, two keys are generated, one private and one public.
  - The private key is stored locally, and the public key is sent to the server.
  - The public key could be used to generate a client certificate to authenticate the user in the future.

```
<html>
<head><title>Title of your web page</title>
</head>
<body>
<form action="demo_keygen.asp"
method="get"> Username: <input
type="text" name="usr_name">
Encryption: <keygen name="security">
<input type="submit">
</form></body></html>
```
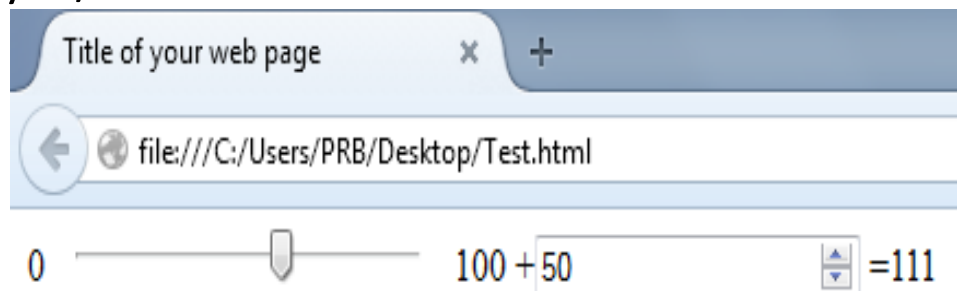
Title of your web page   ×   +

file:///C:/Users/PRB/Desktop/Test.html

Username: [   ]
Encryption: [High Grade   ▼]
[Submit Query]

HTML5 Form Elements

- **<output>**
- The <output> element represents the result of a calculation (like one performed by a script).

Attribute	Description
for	Specifies the relationship between the result of the calculation, and the elements used in the calculation
form	Specifies one or more forms the output element belongs to
name	Specifies a name for the output element

```
<html>
<head><title>Title of your web page</title>
</head><body>
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)"> 0 <input
type="range" id="a" value="50"> 100 +<input type="number" id="b"  value="50">
=<output name="x" for="a b"></output>
</form></body></html>
```

HTML5 Form Elements

- ## HTML5 Video
  - Before HTML5, there was no standard for showing videos on a web page.
  - Before HTML5, videos could only be played with a plug-in (like flash).
  - The HTML5 <video> element specifies a standard way to embed a video in a web page.
  - HTML Video - Media Types: MP4, WebM, Ogg
  - Attribute of video tag:
  - **Autoplay** - To start a video automatically use the autoplay attribute.

<html>
<head><title>Title of your web page</title>
</head>
<body>
<video width="200px" hight="200px" controls autoplay>
<source src="Networking Basics.mp4" type="video/mp4"> Your browser does not support HTML5 video.
</video></body></html>

NIC - network interface card

HTML5 Form Elements

- ## HTML5 Audio

  - Before HTML5, there was no standard for playing audio files on a web page.

  - Before HTML5, audio files could only be played with a plug-in (like flash).

  - The HTML5 <audio> element specifies a standard way to embed audio in a web page.

  - HTML Audio - Media Types: MP3, Ogg, Wav

```
<html>
<head><title>Title of your web page</title>
</head>
<body>
<audio controls>
<source src="A.mp3" type="audio/mp3">
Your browser does not support the audio element.
</audio> </body></html>
```

Title of your web page   ✕   +

file:///C:/Users/PRB/Desktop/Test.html

0:12   5:48
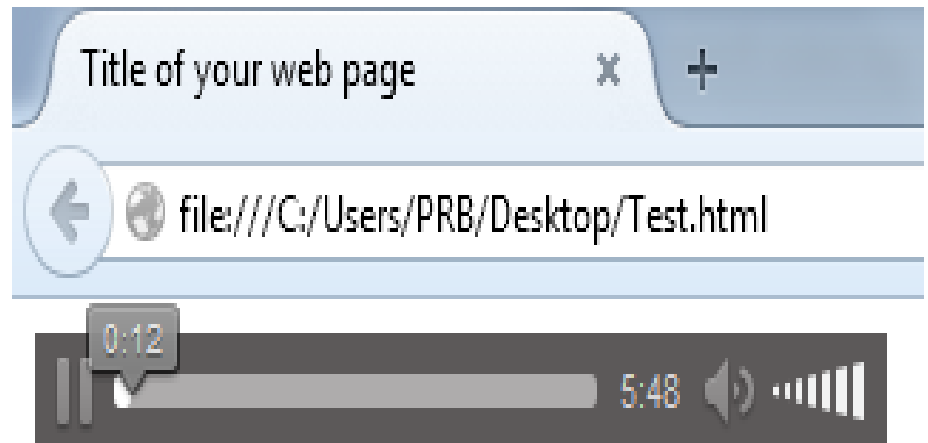
CSS

- ## What is CSS?
  - **CSS** stands for **C**ascading **S**tyle **S**heets
  - CSS defines **how HTML elements are to be displayed**
  - Styles were added to HTML 4.0 **to solve a problem**
  - CSS saves a lot of work
  - External Style Sheets are stored in **CSS files**

- ## Uses of CSS
  - **CSS saves time** because with CSS, we only have to specify content setting details once for any element. CSS automatically apply the specified styles whenever that element occurs.
  - The file size of the CSS is very small hence your website takes **minimal loading time**.
  - **Easy maintenance**: To change the style of an element, we only have to make an edit in one place only.
  - CSS has much wider array of attribute than HTML so by creating the CSS; **you can make the web design flexible.**

CSS

- **Syntax of CSS**
  - The CSS syntax is made up of three parts: **a selector, a property and a value.**
  - **Syntax:** selector {property: value;}
  - The selector is normally the HTML element/tag that you want to define.
  - The property is the attribute that you want to change and each property can take a value. The property and value separated by a colon.
  - A CSS declaration always ends with a semicolon and declaration groups are enclosed by curly braces.
  - By using the below code, all <p> elements will be center-aligned, with a green text color.
  - **Example**: p { color: red; text-align: center; }
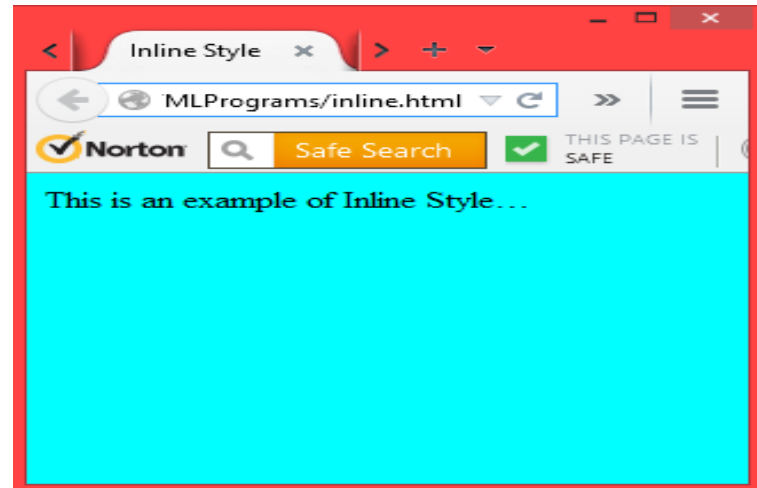- **Types of CSS**
  - There are three ways to implement css command into your site:
    1. Inline style
    2. Embedded/Internal style
    3. External/Linked style sheet

CSS: Inline Style

- Inline styles are styles that are **written directly in the tag on the document**. Inline styles affect only the tag they are applied to.

- **Syntax:** <tag style="property1: value; property2: value;"> </tag>

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

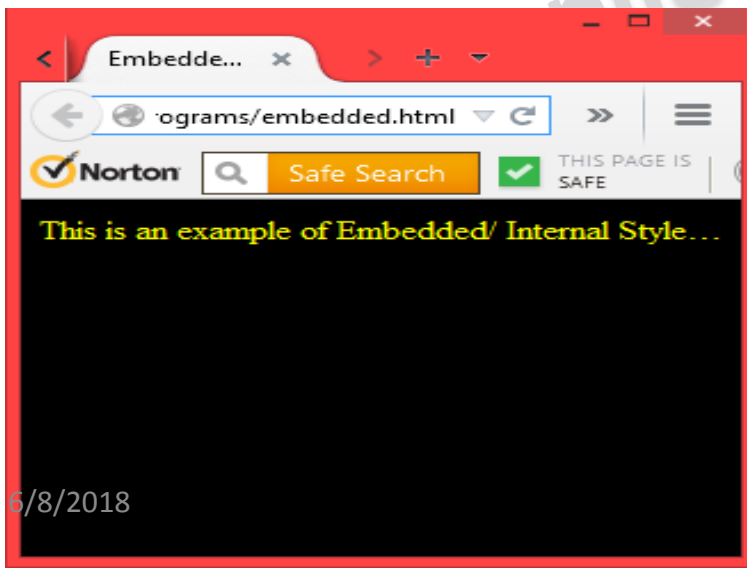# CSS: Embedded/Internal Style

- When using internal/embedded CSS, you must add a new tag, <style>, inside the <head> tag.

- **Syntax:**

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

- …….

- </style>

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

CSS: External/Linked Style Sheet

- External styles are styles that are written in a **separate document and then attached to various Web documents**. External style sheets can affect any document they are attached to.

- External CSS file that contains only CSS code and is saved with **a ".css" file extension**. This file is used in you HTML document using the **<link>** tag instead of <style>.

- let's create **style.css** file having below code.

- Body { background-color: gray; }

- p{ color: blue; }

- H3 { color: white; }

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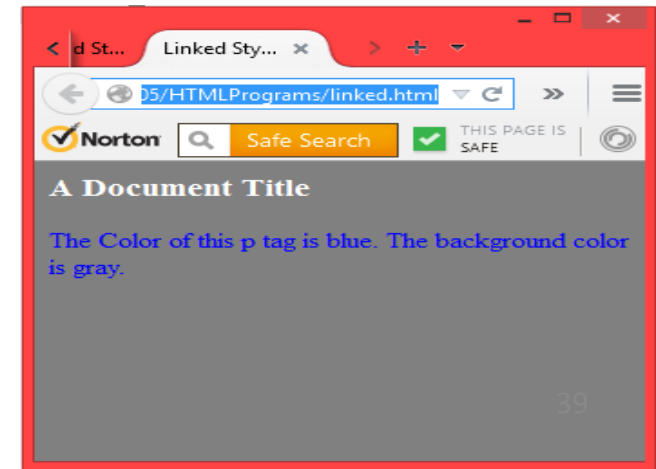
The CSS Box Model

- The CSS box model is a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

- The box model allows us to add a border around elements, and to define space between elements.

- **Content** - The content of the box, where text and images appear.

- **Padding** - Clears an area around the content.

- **Border** - A border that goes around the padding and content.

- **Margin** - Clears an area outside the border.
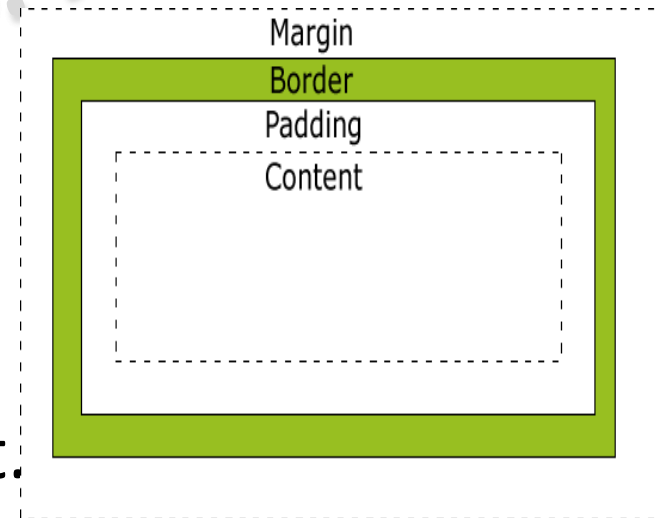
The CSS Box Model
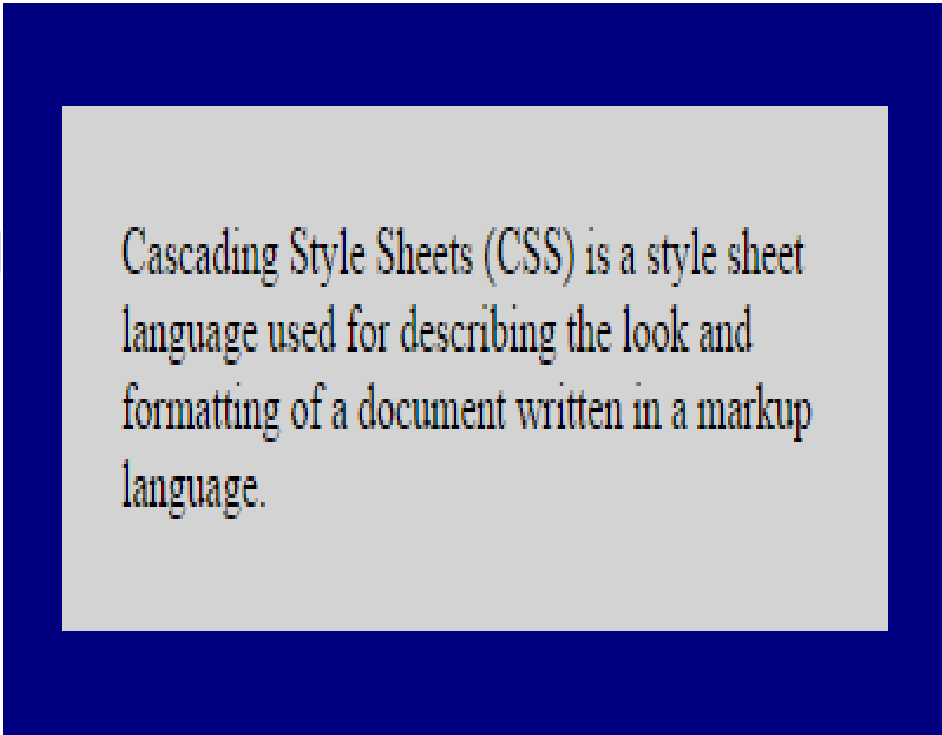
```
<style>
div {
background-color: lightgrey;
    width: 300px;
padding: 25px;
border: 25px solid navy;
    margin: 25px;
}
</style>
<div>
Cascading Style Sheets (CSS)
    is a style sheet language
    used for describing the look
    and formatting of a
    document written in a
    markup language.
</div>
```

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

CSS Margin

- The CSS margin defines the space around elements. The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent

- **Example: Applying margins for paragraph tag.**

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

— **margin: 25px 50px 75px 100px;** //top margin is 25px, right margin is 50px, bottom margin is 75px and left margin is 100px

— **margin: 25px 50px 75px;** // top margin is 25px, right and left margins are 50px and bottom margin is 75px

— **margin**: **25px 50px;** // top and bottom margins are 25px and right and left margins are 50px

— **margin: 25px;** // all four margins are 25px

Value	Description
auto	The browser calculates a margin
length	Specifies a margin in px, pt, cm, etc. Default value is 0px
%	Specifies a margin in percent of the width of the containing element
inherit	Specifies that the margin should be inherited from the parent element

CSS Padding

- The CSS padding properties define the space between the element border and the element content. The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element.

- Example: Applying padding for paragraph tag.

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

— **padding: 25px 50px 75px 100px;** //top padding is 25px, right padding is 50px,bottom padding is 75px and left padding is 100px

— **padding: 25px 50px 75px;** //top padding is 25px, right and left paddings are 50px and bottom padding is 75px

— **padding: 25px 50px;** //top and bottom paddings are 25px and right and left paddings are 50px

— **padding: 25px;** //all four paddings are 25px

Value	Description
length	Defines a fixed padding (in pixels, pt, em, etc.)
%	Defines a padding in % of the containing element

CSS Text and its properties

Property	Description	Example

color	It is used to set the color of the text. With CSS, a color is specified by:
	a HEX value like "#ff00ff"
	a **RGB** value like "rgb(255,0,0)" a **Color**
	name like "black"
		body { color: black; } h1 {
		color: #ff00ff;
		}

text-align	It is used to specify the horizontal alignment of the text. The text can be **centered,** or **aligned to the left or right,** or **justified.**	p.main { text-align: justify; } h1 { text-align: right; }

text-decoration	It is used to set or remove decoration from text. It is mostly used to remove underlines from links for design purposes.
	This property has following values:
	(1) none: Defines a normal text. This is default.
	(2) underline: Defines a line below the text.
	(3) overline: Defines a line above the text.
	(4) line-through: Defines a line through the text.
		a {
		text-decoration: none;
		}
		h1 {
		text-decoration: underline;
		}

text-transform	It is used to controls the capitalization of text. This property specify uppercase and lowercase letters in a text. This property has following values:
	(1) none: No capitalization. The text renders as it is.
	(2) capitalize: Transforms the first character of each word to uppercase.
	(3) uppercase: Transforms all characters to uppercase.
	(4) lowercase: Transforms all characters to lowercase.
		p.uppercase {
		text-transform: uppercase;
		}
		p.lowercase {
		text-transform: lowercase;
		}
		p.capitalize {
		text-transform: capitalize;
		}

CSS Text and its properties

text-indent	It is used to specifies the indentation of the first line in a text-block.	p { text-indent: 50px; }

text-shadow	This property adds shadow to text. This property has following values:
	(1) h-shadow(Required): The position of the horizontal shadow.
	(2) v-shadow(Required): The position of the vertical shadow.
	(3) blur-radius(Optional): The blur radius. Default value is 0.
	(4) color(Optional): The color of the shadow.
	(5) none: Default value. No shadow.
		h1 {
		text-shadow: 2px 2px 8px #FF0000;
		}

letter-spacing	This property increases or decreases the space between characters in a text.	h1 { letter-spacing: 2px; } h2 { letter-spacing: -3px; }
line-height	This property specifies the line height.	p.small { line-height: 70%; }

CSS Font

Property	Description	Example
font	Sets all the font properties in one declaration.	p { font: italic bold 12px/30px Georgia, serif; }

font-family	Specifies the font family for text. There are two types of font family names:
	(1) **family-name** like "times", "courier", "arial", etc.
	(2) **generic-family** like "serif", "sans-serif", etc.
		p.serif {
		font-family: "Times New Roman", Times, serif;
		}

font-size	Specifies the font size of text. Different Property values for font- size are **medium, xx-small, x-small small, large, x-large, xx-large, smaller, larger, length, initial and inherit;**	h1 { font-size:150%; } div {
		font-size:larger;
		}

font-style	Specifies the font style for text. This propery has three values:
	(1) **normal**: The text is shown normally
	(2) **italic**: The text is shown in italics
	(3) **oblique**: This text is very similar to italic but less supported.
		p.normal {
		font-style: normal;
		}

CSS Font

Property	Description	Example

font-variant	Specifies whether or not a text should be displayed in a small-caps font. It has four property values:
	(1)normal: Displays a normal font**.**
	(2)small-caps: Displays a small-caps font
	(3)initial: Sets this property to its default value
	(4)inherit: Inherits this property from its parent element.
		p.normal {
		font-variant: normal;
		}
		p.small {
		font-variant: small-caps;
		}

font-weight	The font-weight property sets how thick or thin characters in text should be displayed. It has following property values: normal, bold, bolder, lighter, 100, 200, 300, 400,
	500, 600, 700, 800, 900, initial and inherit.
		p.normal {
		font-weight: normal;
		}
		p.thick {
		font-weight: bold; }

CSS Link

- Links can be styled with any CSS property (e.g. color, font-family, background, etc.).
  - It has four links states:
    - **a:link**: an unvisited link.
    - **a:visited**: a link the user has visited.
    - **a:hover**: a link when the user mouse over it.
    - **a:active**: a link the moment it is clicked.

  Example:

```
/* unvisited link */
a:link {
color: #FF0000;
}
```

```
/* visited link */
a:visited {
color:
#00FF00;
}
```

```
/* mouse over link */
a:hover  { color:
#FF00FF;
}
```

```
/* selected link */
a:active  { color:
#0000FF;
}
```

CSS Background

Property	Description	Example
background	Sets all the background properties in one declaration	background: #00ff00 url("smiley.gif")
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	background- attachment: fixed;
background-color	Sets the background color of an element	background-color: yellow;
background-image	Sets the background image for an element	background-image: url("paper.gif");
background-position	Sets the starting position of a background image	background-position: center;
background-repeat	Sets how a background image will be repeated	background-repeat: no- repeat;

CSS Border Properties

- **Border Style**

- The border-style property specifies what kind of border to display. border-style values:

none: Defines no border

dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

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

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

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

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

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

CSS Border Properties

- ## **Border Width**
  - The border-width property is used to set the width of the border.
  - The width is set in pixels, or by using one of the three pre-defined values: thin, medium, or thick.
  - **Example:** border-width: medium;

- ## **Border Color**
  - The border-color property is used to set the color of the border. The color can be set by:
    - name - specify a color name, like "red"
    - RGB - specify a RGB value, like "rgb(255,0,0)"
    - Hex - specify a hex value, like "#ff0000"
  - **Example:** border-color: red;

CSS Border Properties

- ## Border Width
  - The border-width property is used to set the width of the border.
  - The width is set in pixels, or by using one of the three pre-defined values: thin, medium, or thick.
  - **Example:** border-width: medium;

- ## Border Color
  - The border-color property is used to set the color of the border. The color can be set by:
    - name - specify a color name, like "red"
    - RGB - specify a RGB value, like "rgb(255,0,0)"
    - Hex - specify a hex value, like "#ff0000"
  - **Example:** border-color: red;

The position Property

- The position property specifies the type of positioning method used for an element.
- There are four different position values: static, relative, fixed, absolute
- Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first.

- **Position: static**
  - HTML elements are positioned static by default.
  - Static positioned elements are not affected by the top, bottom, left, and right properties.
  - An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page.

- **Position: fixed**
  - An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.
  - A fixed element does not leave a gap in the page where it would normally have been located.

The position Property

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

This div element has position: static;

This div element has position: fixed;

The position Property

- **Position: relative**
  - An element with position: relative; is positioned relative to its normal position.
  - Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

- **Position: absolute**
  - An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
  - However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

The position Property

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

This
element has position: relative;

This
element has position: absolute;

Overlapping Elements

- When elements are positioned, they can overlap other elements.

- The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

- An element can have a positive or negative stack order.

```
<html>
<head>
<title>Linked Style Sheets</title>
<style>
img {  position: absolute; left: 0px; top: 0px; z-index: -1; }
</style>
<img src="Penguins.jpg" width="100" height="140">
<p>Because the image has a z-index of -1, it will be placed behind the text. </p></body>
</html>
```

Float Properties

- Float property can be used to wrap text around images.
- All CSS Float Properties

Property	Description	Values
clear	Specifies on which sides of an element where floating elements are not allowed to float	left, right, both, none, inherit
float	Specifies whether or not an element should float	left, right, none, inherit

```
<html>
<head>
<title>Linked Style Sheets</title>
<style> img { float: left;       } </style>
<p><img src="Penguins.jpg"
alt="W3Schools.com" width="100"
height="140">
Cascading Style Sheets (CSS) is a style sheet
language used for describing the look and
formatting of a document written in a
markup language. </p>></body>
</html>
```

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

Difference between CSS2 and CSS3

- CSS3 is divided into many different documents called Modules. Every module adds new capability or extends features defined in CSS2. Work on CSS3 started around the time of publication of the original CSS2 recommendation.

- Because of the modularization in CSS3, every modules has different stability and is in different status.

- The CSS3 version supports many more browsers than CSS2.

- CSS3 has other added features such as new combinator, new CSS selectors, new pseudo-elements and new style properties

Opacity (Transparency)

- **The opacity** property sets the opacity level for an element. The CSS3 property for transparency is opacity.

- IE9, Firefox, Chrome, Opera, and Safari use the property opacity for transparency. The opacity property can take a value from 0.0 - 1.0. A lower value makes the element more transparent.

- The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see- through, and 0 is completely transparent

```
<html>
<head>
<style>
img {  opacity: 0.4; }
</style></head>
<body>
<img src="klematis.jpg" width="150"
height="150">
</body></html>
```

Original Image: Transparent Image:

Box-shadow (Shadows)

- **The box-shadow property** allows designers to easily implement multiple drop shadows (outer or inner) on box elements, text, etc by specifying values for color, size, blur and offset. The property of box- shadow is a comma-separated list of shadows, each specified by 2-4 length values, an optional color, and an optional inset keyword. Omitted lengths are 0.

- **Syntax:** box-shadow: none|h-shadow v-shadow blur spread color |inset|initial|inherit;

div {
width: 300px; height: 100px; padding: 15px;
background-color: yellow;
box-shadow: 10px 10px 5px grey;
}
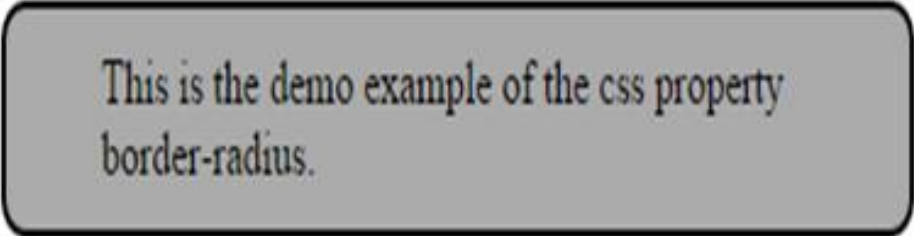<div>This is a div element with a box-shadow</div>

Output:

Border-radius (Rounded Corners)

- **The border-radius** property is a shorthand property for setting the four border -*-*-radius properties. This property allows you to add rounded borders to elements.

- **Syntax:** border-radius: 1-4 length|% / 1-4 length|%|initial|inherit;

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

Output:

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

Gradients

- **A gradient is a sequence of colors**. Simple gradients flow from a foreground to a background color, but gradients can contain many other colors. **CSS3 gradients** let you display smooth transitions between two or more specified colors.

- By using CSS3 gradients you can reduce download time and bandwidth usage. In addition, elements with gradients look better when zoomed, because the **gradient is generated by the browser**.

- CSS3 defines **two types of gradients: (1). Linear Gradient and (2). Radial Gradient.**

• Linear Gradients

- **Linear Gradient** which flows in a straight line from one color to another.

- To create a linear gradient you **must define at least two color stops**. Color stops are the colors you want to render smooth transitions among. You can also **set a starting point and a direction (or an angle) along with the gradient effect.**

- **Syntax:** background: linear-gradient(direction, color-stop1, color-stop2, …);

Linear Gradients

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

Radial Gradients

- **Radial Gradient** where one color is concentrated at a specific point, and the other colors are visible farther from that point. By default, shape is ellipse, size is farthest-corner, and position is center. To create a radial gradient you must also define at least two color stops.

- Syntax: background: radial-gradient(shape size at position, start-color, ..., last-color);

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


Radial Gradient - Circle Shape

Transition Animation

- **CSS3 transitions** allows you to change property values smoothly (from one value to another), over a given duration.

  –**transition-property**: The type of animation defined by this tag. The default value is all, but other types are expected to work, including color, length, width, percentage, opacity, and number.

  –**transition-duration**: The length of the animation in seconds.

  –**transition-timing-function**: If you want the animation to occur at a constant speed, use linear.

  –**transition-delay**: If you include a second time value, this will be considered a delay. The animation will not begin until after the delay.

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

Output:
Original Box

After Hover Effect

Transformations

- **A Transformation** is an effect that lets an element change shape, size and position.

- **The transform property** applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skewX, skewY and skew, etc., elements.

- **Syntax :** transform: none|transform-functions|initial|inherit;

- When you apply transform to an element, you need to apply one or more of the following parameters to describe the type of transformation:

  - **translate:** The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).
  - **rotate:** The rotate() method rotates an element clockwise or counter-clockwise according to a given degree. Using negative values will rotate the element counter-clockwise.
  - **scale:** The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).

Transformations

- **skew:** The skew() method skews an element along the X and Y-axis by the given angles.

- **skewX:** The skewX() method skews an element along the X-axis by the given angle.

- **skewY:** The skewY() method skews an element along the Y-axis by the given angle.

- **matrix:** The matrix() method combines all the 2D transform methods into one. The matrix() method take six parameters, containing mathematic functions, which allows you to rotate, scale, move (translate), and skew elements.

Transformations

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

height: 40px;

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

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

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

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

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

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

In Body Tag:

<div id="box1">Original Box</div>

<div id="box2">Translate</div>

<div id="box3">Rotate</div>

<div id="box4">Scale</div>

<div id="box5">Skew</div>

<div id="box6">Matrix</div>

Output:

Original Box

Translate

Rotate

Scale

Skew

Matrix

Animations

- An animation lets an element gradually change from one style to another.

- You can change as many CSS properties you want, as many times you want.

- To use CSS3 animation, you must first specify some keyframes for the animation.

- Keyframes hold what styles the element will have at certain times.

Key Frames

- **The @keyframes** rule specifies the animation code. The animation is created by gradually changing from one set of CSS styles to another.

- During the animation, you can change the set of CSS styles many times.

- Specify when the style change will happen in percent, or with the keywords "from" and "to", which is the same as 0% and 100%. **0% is the beginning of the animation, 100% is when the animation is complete.**

- For best browser support, you should always define both the 0% and the 100% selectors.

- **Syntax:** @keyframes animationname {keyframes-selector {css-styles;}}

Key Frames

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

Output: