

## 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.

## History of 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. 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.
- HTML is a platform independent language that can be used on any platform such as Windows, Linux and so on. HTML documents/pages have a .html extension.
- In 1989 HTML was originally developed by Tim Berners-Lee. It is derived from SGML (Standard Generalized Markup Language).

## HTTP (Hypertext Transfer Protocol)

- 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.

## Elements

- HTML elements generally consist of a pair of angle-bracketed tags surrounding some text.
- The end tag (</Tag>) is just like the start tag (<Tag>).

<Tag> ← Start Tag

-----  
Text that the tag affect

-----  
</Tag> ← End Tag

## Tags

- Html documents are created by combining special markup code called tags.
- The tags define structure of the document and provide the framework for holding the actual content, which can be text, images or other special content.
- All Html tags are enclosed in angle brackets '<' and '>' i.e. Opening tag <HTML> and closing tag </HTML>.
- Tags are not case sensitive.
- There are two type of tags:
  - **Container Tags:**
    - Tags which have both the opening and closing i.e. <TAG> and </TAG>are called container tags.
  - **Empty Tags:**
    - Tags which have only opening and no ending are called empty tags.

## Attributes

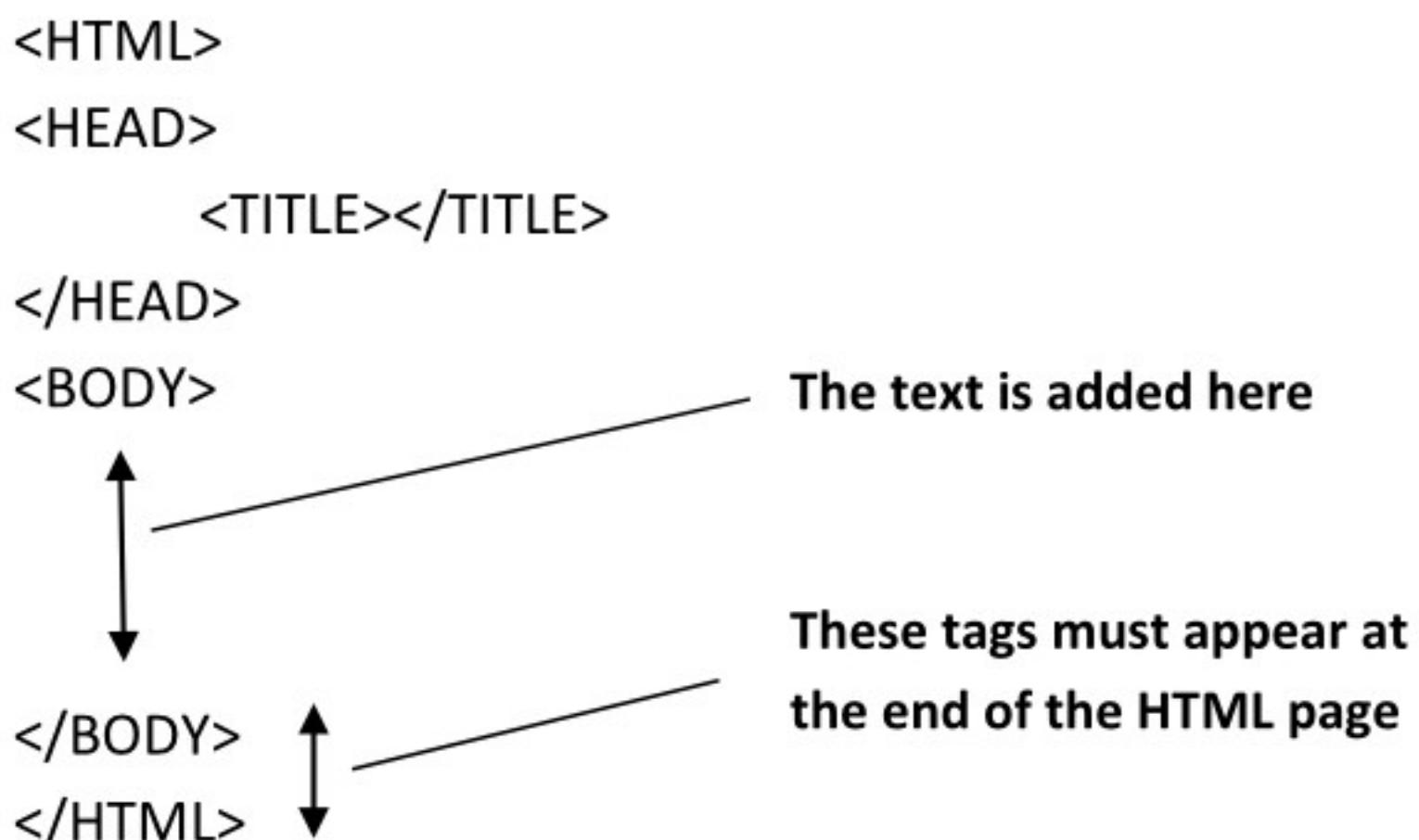
- All HTML elements can have **attributes**.
- Attributes provide **additional information** about an element.
- Attributes are always specified in **the start tag**.
- Attributes usually come in name/value pairs like: **name="value"**.

### Example:

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

- Attribute names and attribute values are case-insensitive.

## Structure of HTML document



## 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.

**Example:**

```
<html>  
    Your title and document goes here.  
</html>
```

## **<head>**

- This tag is used to indicate the header section of the HTML document, which typically includes the `<title>` and `<meta>` tags.

**Example:**

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

## **<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.

**Example:**

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

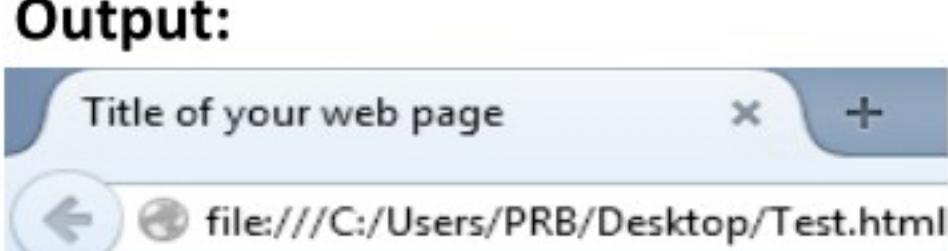
## **<body>**

- The `<body>` tag includes the HTML body of the document. Everything inside the `<body>` tag is displayed on the browser.
- The `<body>` tag may contain several attributes. The most commonly used ones are listed below:
  - **bgcolor:** This is the background color of the entire HTML document, and may be specified either by the color name directly or by the six-digit hex code.
  - **topmargin:** The margin from the top of the browser window.
  - **leftmargin:** The margin from the left of the browser window.

**Example:**

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

**Output:**



HTML web page contents

## <hr>

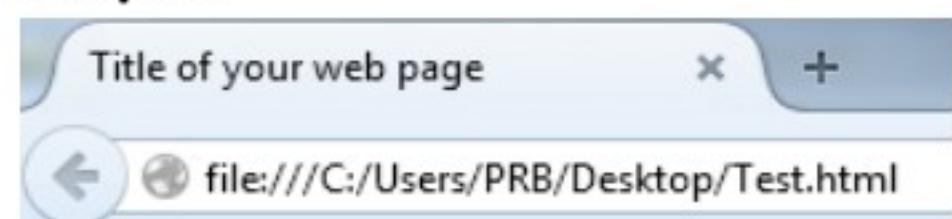
- The <hr> tag is used for creating a horizontal line. This is also called Horizontal Rule in HTML.

Attribute	Value	Description
align	left, right, center, justify	Specifies the alignment of the horizontal rule.
size	pixels or %	Specifies the height of the horizontal rule.
width	pixels or %	Specifies the width of the horizontal rule.

### Example:

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

### Output:



## <p>

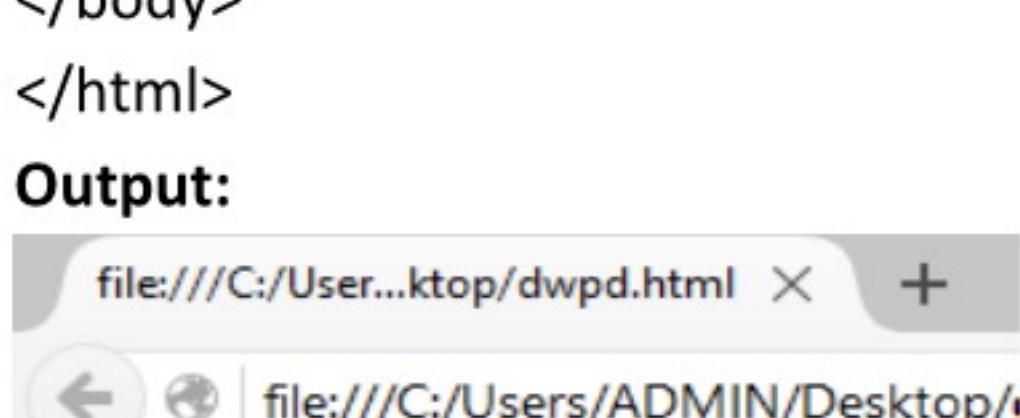
- The <p> tag defines a paragraph of text.

Attribute	Value	Description
align	left, right, center, justify	Specifies text alignment within a paragraph.

### Example:

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

### Output:



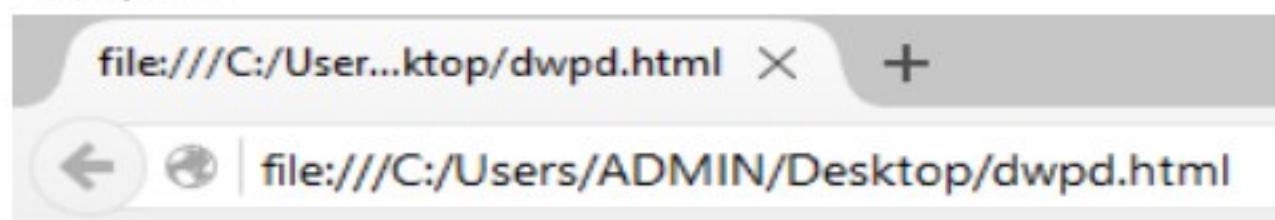
## <h1> to <h6>

- The <h1> to <h6> tags are used to define HTML heading.
- <h1> defines largest heading and <h6> defines smallest heading.

### Example:

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

### Output:



**This is heading 1**

**This is heading 2**

**This is heading 3**

**This is heading 4**

**This is heading 5**

**This is heading 6**

## <img>

- The <img> tag defines an image in an HTML page.

Attribute	Value	Description
src	URL	Specifies the URL of an image
height	pixels or %	Specifies the height of an image
width	pixels or %	Specifies the width of an image
alt	text	Specifies an alternate text for an image
border	pixels or %	Specifies the width of the border around an image

### Example:

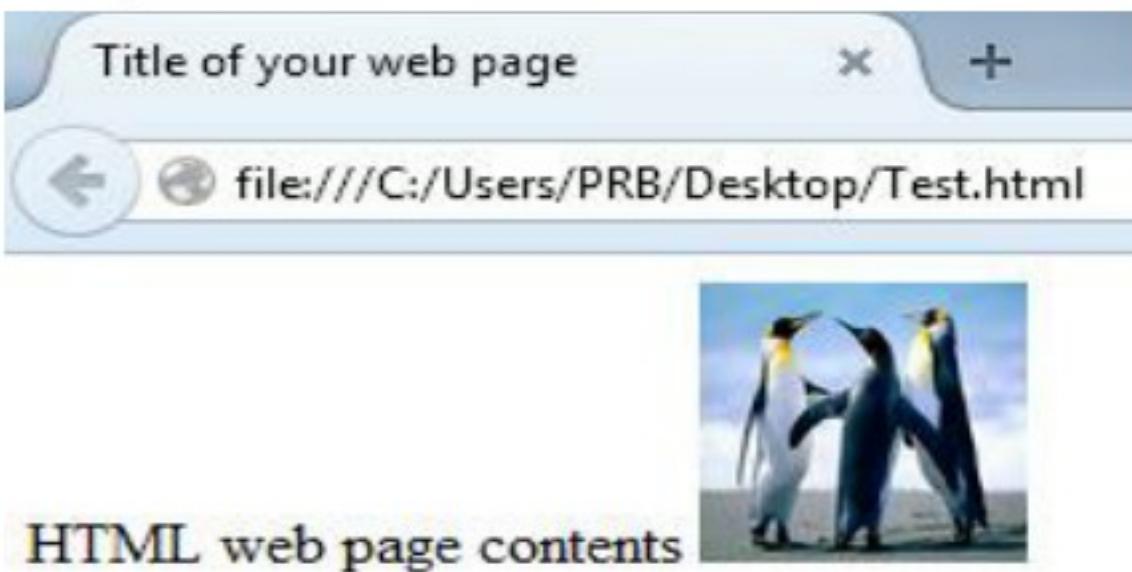
```
<html>
<head>
    <title>Title of your web page</title>
```

```

</head>
<body>HTML web page contents
    
</body>
</html>

```

**Output:**



**<embed>**

- The <embed> tag defines a container for an external application or interactive content (a plug-in).

Attribute	Value	Description
height	pixels or %	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 or %	Specifies the width of the embedded content

**Example:**

```

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

```

**<bgsound>**

- The HTML <bgsound> tag is used to play a soundtrack in the background. This tag is for Internet Explorer only.

Attribute	Value	Description
loop	number	Let's you replay a background soundtrack a certain number of times.
src	URL	Specifies the path of the sound file.

**Example:**

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

## <font>

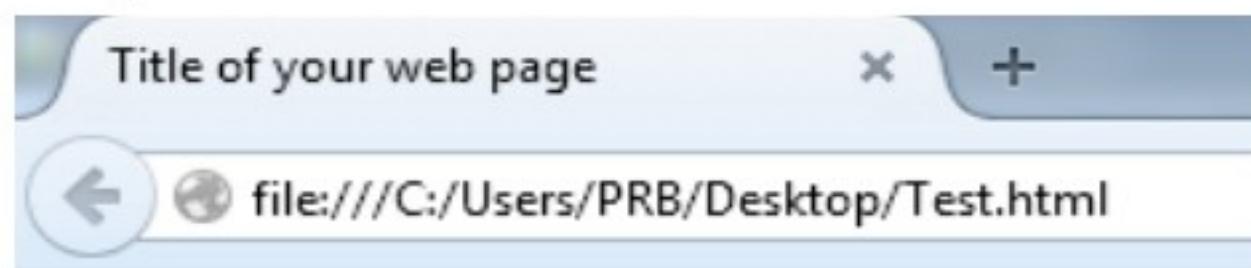
- The <font> tag specifies the font face, font size, and color of text.

Attribute	Value	Description
color	rgb(x,x,x), #xxxxxx, colorname	Specifies the color of text.
face	font_family	Specifies the font of text.
size	number	Specifies the size of text.

### Example:

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

### Output:



HTML web page contents **Hello**

## <center>

- The <center> tag is used to center-align text.

### Example:

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

## <marquee>

- An HTML marquee is a scrolling piece or block of text displayed either horizontally across or vertically down your webpage depending on the settings.

Attribute	Value	Description
width	pixels or %	This specifies the width of the marquee.
height	pixels or %	This specifies the height of the marquee.
direction	up, down, left, right	Defines the direction of scrolling the content.

### Example:

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

## <a>

- The <a> tag defines a hyperlink, which is used to link from one page to another.
- The most important attribute of the <a> element is the href attribute, which indicates the link's destination.
- By default, links will appear as follows in all browsers:
  - An unvisited link is underlined and blue
  - A visited link is underlined and purple
  - An active link is underlined and red

Attribute	Value	Description
href	URL	Specifies the URL of the page the link goes to <b>An absolute URL</b> - points to another web site (href="http://www.example.com/default.html") <b>A relative URL</b> - points to a file within a web site (like href= "default.htm ")
target	_blank _parent _self _top framename	Specifies where to open the linked document

### Example:

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

## List Tags

### <li>

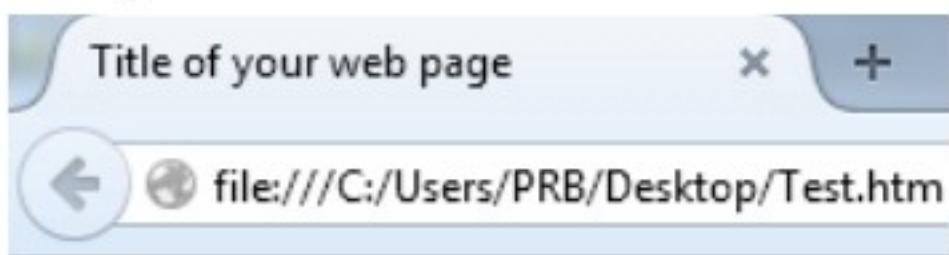
- The <li> tag defines a list item.
- The <li> tag is used in ordered lists (<ol>), unordered lists (<ul>), and in menu lists (<menu>).

Attribute	Value	Description
Type	A, a, I, i, 1, Disc, circle, square	Specifies the type of the list.
value	number	Specifies the value of a list item.

### Example:

```
<ol type="A">
  <li>Cricket</li>
  <li>Football</li>
  <li>Hockey</li>
</ol>
```

## Output:



- A. Cricket
- B. Football
- C. Hockey

## <ul>

- The HTML <ul> tag is used for creating an unordered list.

Attribute	Value	Description
type	square, circle, disc	Specifies the style of the list.

## Example:

```
<ul type="square">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Milk</li>
</ul>
```

## Output:



- Cricket
- Football
- Hockey

## <ol>

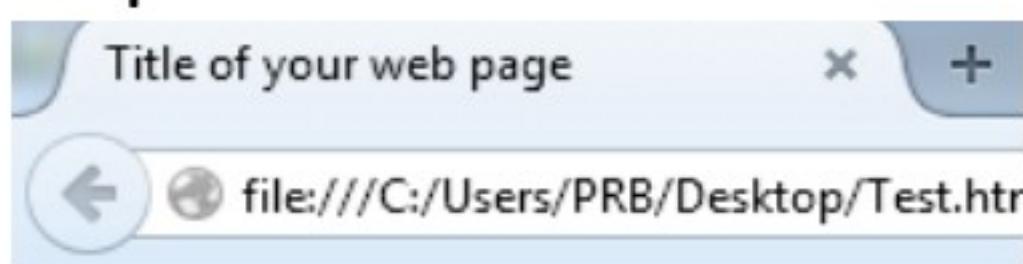
- The HTML <ol> tag is used for creating an ordered list.

Attribute	Value	Description
reversed	reversed	Specifies that the list order should be descending (9,8,7...).
start	number	Specifies the start value of an ordered list.
type	A, a, I, i, 1	Specifies the style of the list.

## Example:

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

## Output:



- 5. Cricket
- 4. Football
- 3. Hockey

## Working with Table

### <table>

- The <table> tag defines an HTML table.
- An HTML table consists of the <table> element and one or more <tr>, <th>, and <td> elements.
- The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.

Attribute	Value	Description
align	right, left, center, justify	Specifies the order of the list (descending).
bgcolor	rgb(x,x,x), #xxxxxxxx, colorname	Specifies the initial number to start the list.
cellpadding	pixels	Specifies the space between the cell wall and the cell content.
cellspacing	pixels	Specifies the space between cells.

### <th>

- The <th> tag defines a header cell in an HTML table.
- The text in <th> elements are bold and centered by default.

Attribute	Value	Description
bgcolor	rgb(x,x,x), #xxxxxxxx, colorname	Specifies the background color of a header cell.
v-align	top, middle, bottom	Vertical aligns the content in a header cell.
align	right, left, center, justify	Aligns the content in a header cell.
colspan	number	Specifies the number of columns a header cell should span.
rowspan	number	Specifies the number of rows a header cell should span.
height	pixels or %	Specifies the height of a header cell.
width	pixels or %	Specifies the width of a header cell.

### <td>

- The <td> tag defines a column in an HTML table. Columns are specified within each row.

Attribute	Value	Description
bgcolor	rgb(x,x,x), #xxxxxxxx, colorname	Specifies the background color of a cell.
v-align	top, middle, bottom	Vertical aligns the content in a cell.

align	right, left, center, justify	Aligns the content in a cell.	
colspan		number	Specifies the number of columns this particular column occupies.
rowspan		number	Specifies the number of rows this particular row occupies.
height		pixels or %	Specifies the height of a cell.
width		pixels or %	Specifies the width of a cell.

## <tr>

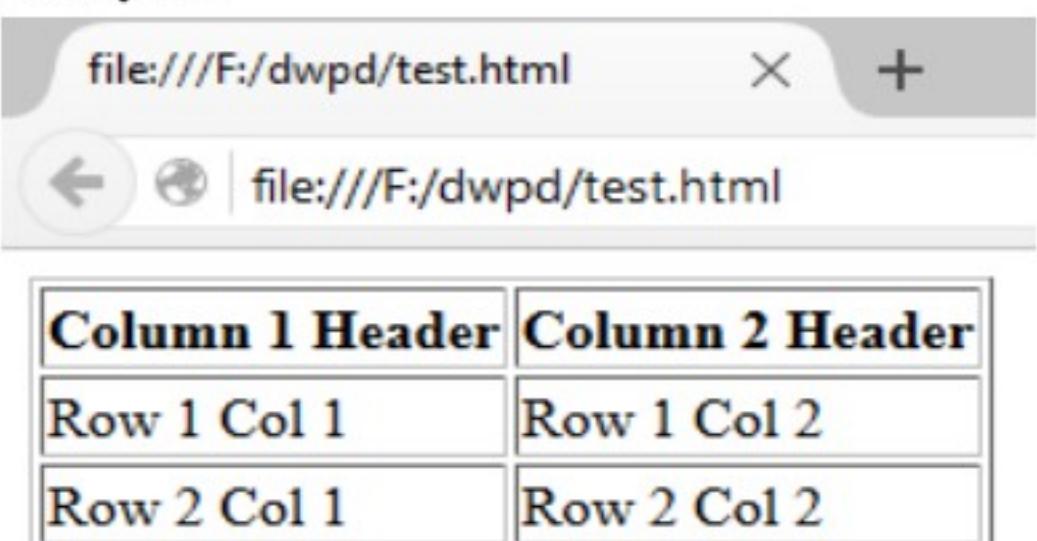
- The <tr> tag defines a row in an HTML table.

Attribute	Value	Description
align	right, left, center, justify	Aligns the content in a table row.
bgcolor	rgb(x,x,x), #xxxxxx, colorname	Specifies a background color for a table row.
valign	top, middle, bottom	Vertical aligns the content in a table row.

### Example 1:

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

### Output:

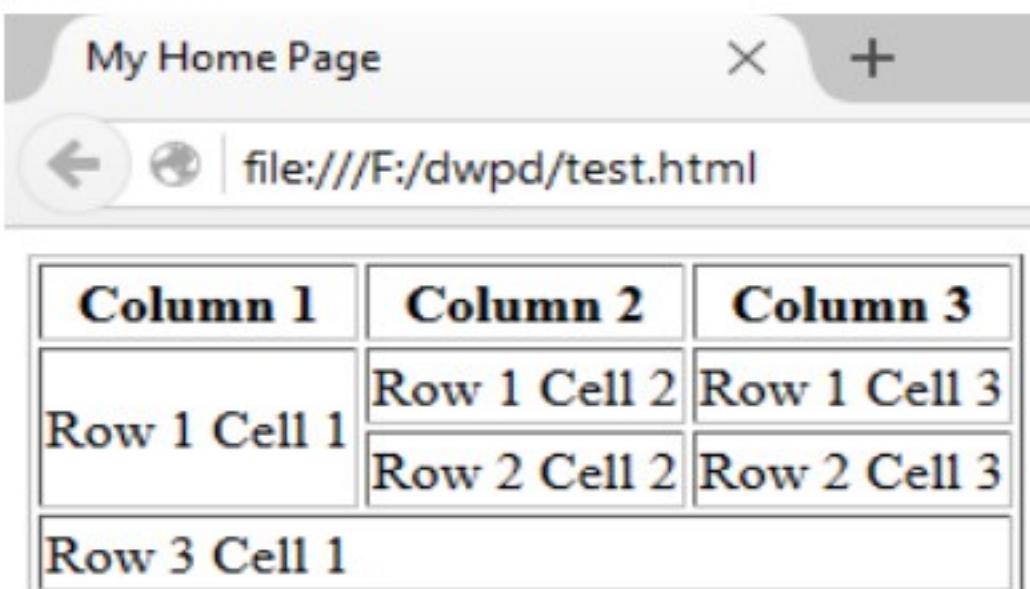


Column 1 Header	Column 2 Header
Row 1 Col 1	Row 1 Col 2
Row 2 Col 1	Row 2 Col 2

## Example 2:

```
<html>
<head>
    <title>My Home Page</title>
</head>
<body>
    <table border= "1">
        <tr>
            <th>Column 1 </th>
            <th>Column 2 </th>
            <th>Column 3</th>
        </tr>
        <tr>
            <td rowspan="2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
        </tr>
        <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
        </tr>
        <tr>
            <td colspan="3">Row 3 Cell 1</td>
        </tr>
    </table>
</body>
</html>
```

## Output:



## <div>

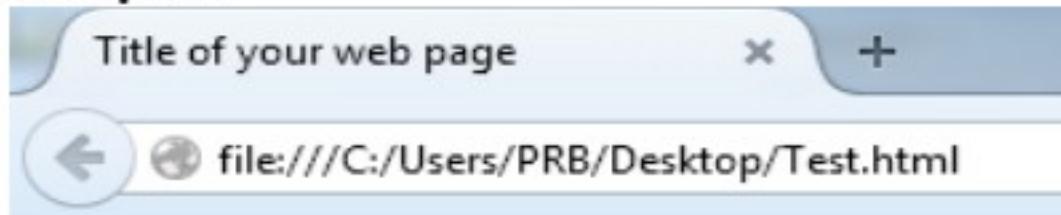
- The HTML <div> tag is used for defining a section of your document. With the div tag, you can group large sections of HTML elements together and format them with CSS.

Attribute	Value	Description
align	right, left, center, justify	Specifies the alignment of the content inside a <div> element.

**Example:**

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

**Output:**



**This is a heading in a div element**

This is some text in a div element.

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

- **ID Selector**

- The “**id**” selector uses the id attribute of an HTML tag to find the specific element.
- **ID’s are unique**. Each element can have only one ID and each page can have only one element with that id.
- To find an element with a specific id, write an **hash(#)** character, followed by the id of the element.

- **Class Selector**

- The “**class**” selector finds elements with the specific class. The class selector uses the HTML class attribute.
- **Classes are not unique**. We can use the same class on multiple elements and we can use multiple classes on the same element.
- To find elements with a specific class, write a **period(.)** character, followed by the name of the class

**Example:**

```
<html>
<head>
    <style>
        #intro{
            color: yellow;
        }
        #intro1{
            color: hotpink;
        }
    </style>
</head>
<body>
    <h1>My First Web Page</h1>
    <p>This is my first web page.</p>
</body>

```

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

**Output:**

Welcome to DWSL material webpage.

Learn the use of selector ID and Class.

Thanking you.

## Special Character

- 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:

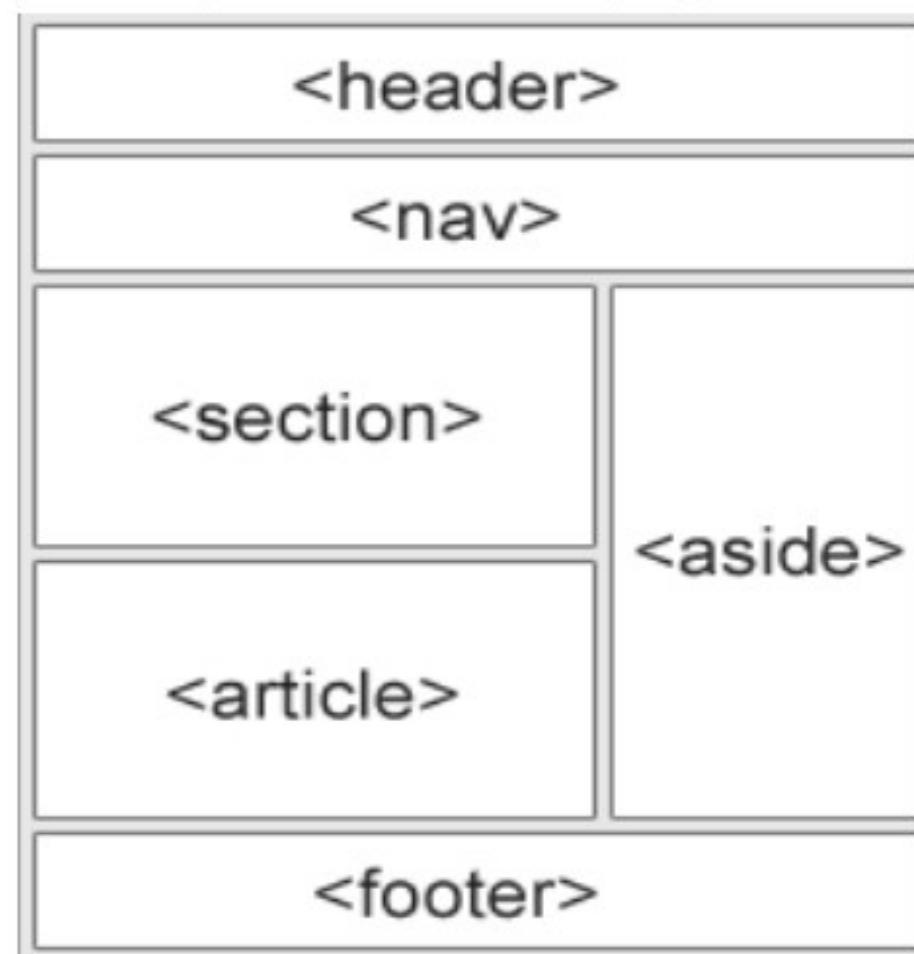
Sr. No.	Result	Description	Entity Name
1		non-breaking space	&nbsp;
2	<	less than	&lt;
3	>	greater than	&gt;
4	&	ampersand	&amp;
5	¢	cent	&cent;
6	£	pound	&pound;
7	¥	yen	&yen;
8	€	euro	&euro;
9	©	copyright	&copy;
10	®	registered trademark	&reg;

## Difference between HTML 4.0 & 5.0

HTML 4.0	HTML 5.0
In HTML4, the DOCTYPE declaration is very hard and lengthy.	The DOCTYPE element has been made very simple.
HTML4 uses the most common webpage structure like header, footer and columns, etc.	HTML5 provides the many inbuilt elements to structure the webpages. For example, header, footer, nav, article and section etc.
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.
HTML 4 cannot handle inaccurate syntax.	HTML 5 is capable of handling inaccurate syntax.
HTML4 does not provide the inbuilt tag for vector graphics.	It provides the inbuilt tag like and canvas which are used for vector graphics.
In HTML4, It was extremely lengthy task to get geographical location of users who visit the website.	In HTML5, It is very easy to get geographical location of users who visit the website.
In HTML4, Browser cache is used for temporary storage.	In this Web <a href="#">SQL</a> database, Application Cache and Web storage are available for 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>



### <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.

## <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:

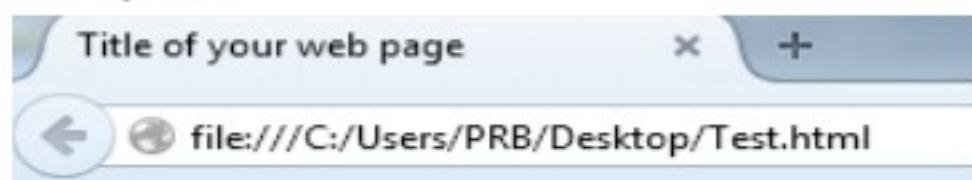
```
<html>
    <head>
        <title>Title of your web page</title>
    </head>
    <body>
        <header>
            <h1>This is page heading</h1>
        </header>
        <nav>
            <ul>
                <li><a href="#">Home</a></li>
```

```

<li><a href="#">About Us</a></li>
<li><a href="#">Contact Us</a></li>
</ul>
</nav>
<article>
    <h1>This is article heading</h1>
    <p>Hello world! Hello world! Hello world!</p>
</article>
<aside>
    <figure>
        
    </figure>
</aside>
<section>
    <h1>This is a section heading</h1>
    <p>Hello world! Hello world! Hello world! </p>
</section>
<footer>
    <hr/>Copyright (C) 2013. All rights reserved.
</footer>
</body>
</html>

```

**Output:**



## This is page heading

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

## This is article heading

Hello world! Hello world! Hello world!



## This is a section heading

Hello world! Hello world! Hello world!

---

Copyright (C) 2013. All rights reserved.

## HTML5 Form Input Types

Input Type	Description	Example
number	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">
datetime-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.

#### Example:

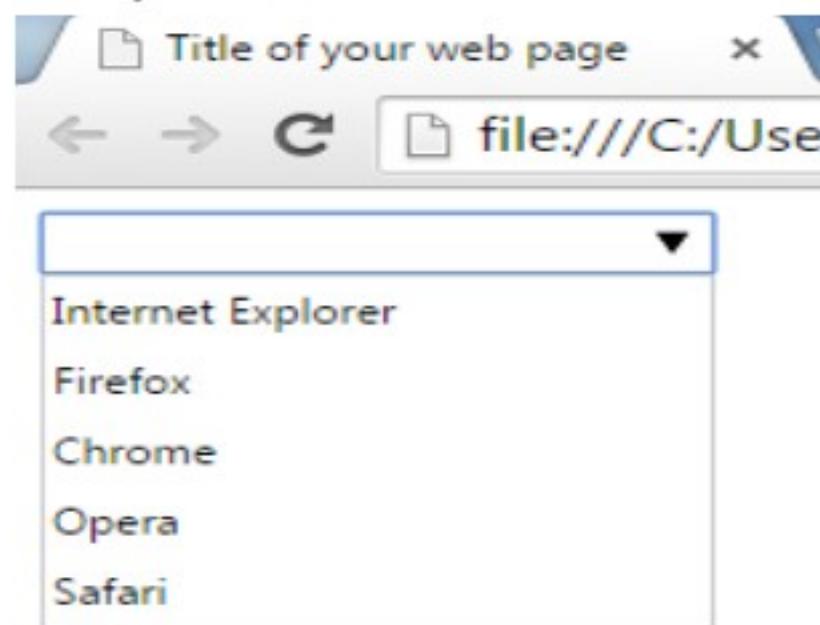
```
<input list="browsers" name="browser">
<datalist id="browsers">
    <option value="Internet Explorer">
```

```

<option value="Firefox">
<option value="Chrome">
<option value="Opera">
<option value="Safari">
</datalist>

```

**Output:**



## <keygen>

- 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.

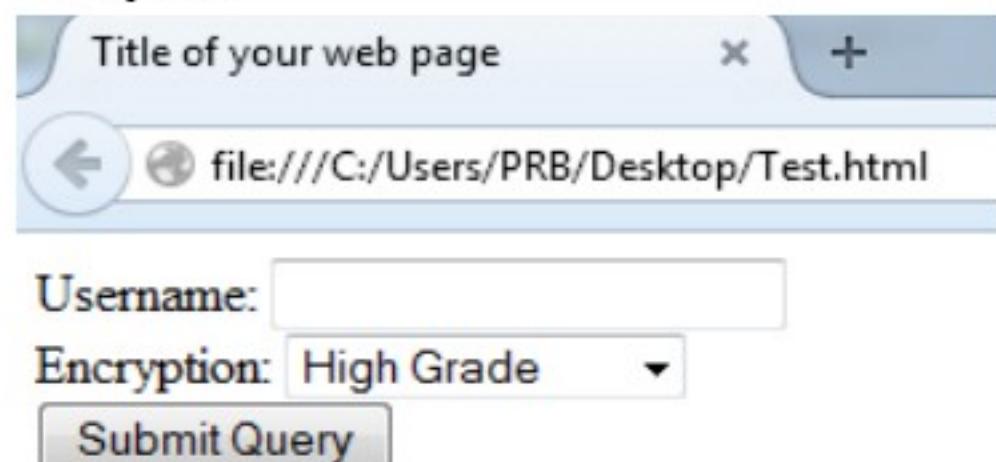
**Example:**

```

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

```

**Output:**



## <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

**Example:**

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

**Output:**



## HTML5 Video

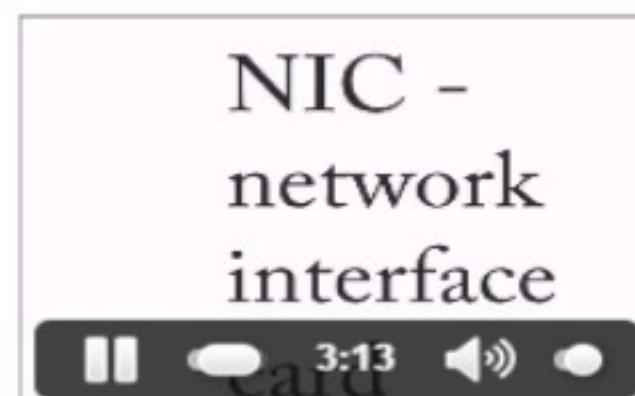
- Before HTML5, there was no standard for showing videos on a web page.
- Before HTML5, videos could only be played with a plug-in (like flash).
- The HTML5 `<video>` element specifies a standard way to embed a video in a web page.
- HTML Video - Media Types: MP4, WebM, Ogg.

Attribute	Description
autoplay	Specifies that the video will start playing as soon as it is ready.
src	Specifies the URL of the video file.
controls	Specifies that video controls should be displayed (such as a play/pause button etc).
loop	Specifies that the video will start over again, every time it is finished.
height	Sets the height of the video player.
width	Sets the width of the video player.

**Example:**

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

**Output:**



## HTML5 Audio

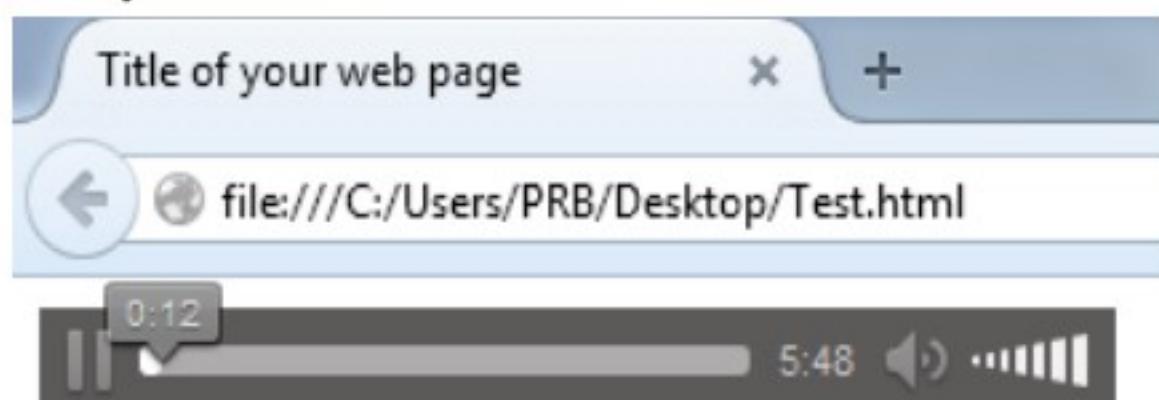
- Before HTML5, there was no standard for playing audio files on a web page.
- Before HTML5, audio files could only be played with a plug-in (like flash).
- The HTML5 `<audio>` element specifies a standard way to embed audio in a web page.
- HTML Audio - Media Types: MP3, Ogg, Wav.

Attribute	Description
autoplay	Specifies that the audio will start playing as soon as it is ready
src	Specifies the URL of the audio file
controls	Specifies that audio controls should be displayed (such as a play/pause button etc)
loop	Specifies that the audio will start over again, every time it is finished

**Example:**

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

**Output:**



## What is CSS? And Define CSS.

- CSS stands for **Cascading Style Sheets**.
- CSS defines **how HTML elements are to be displayed**.
- CSS saves a lot of work this means that the same content can be used with many styles without being modified-a separation of presentation and content.
- Style sheets are capable of specifying that presentation without interfacing with the content of the page.

## 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 **minimum 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**.

## 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

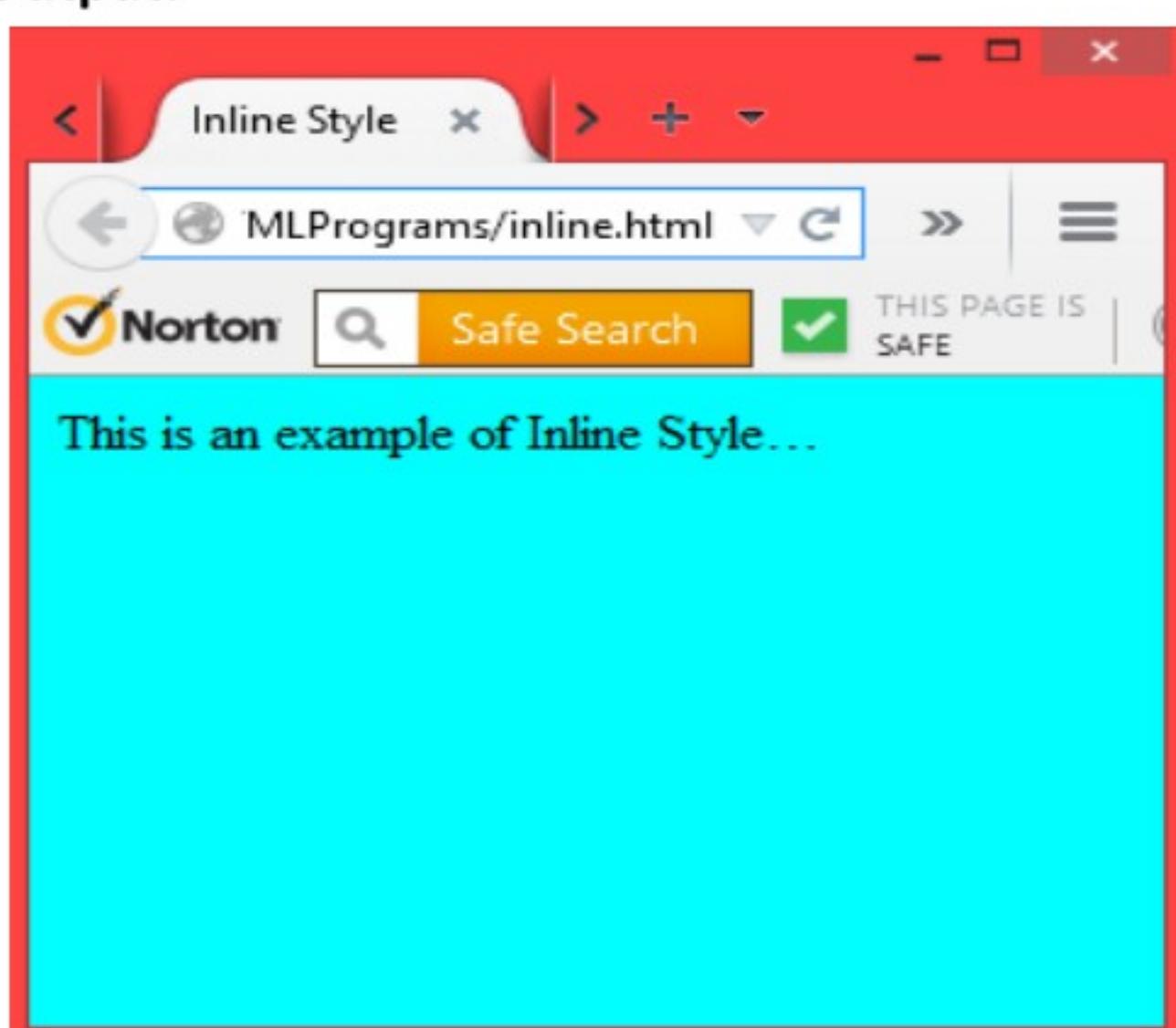
### 1) Inline Style

- 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>

#### Example: Using Inline Stylesheet

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

#### Output:



## 2) Embedded/Internal Style

- When using internal/embedded CSS, you must add a new tag, `<style>`, inside the `<head>` tag.
- Embedded styles affect only the tags on the page they are embedded in.
- **Syntax:**

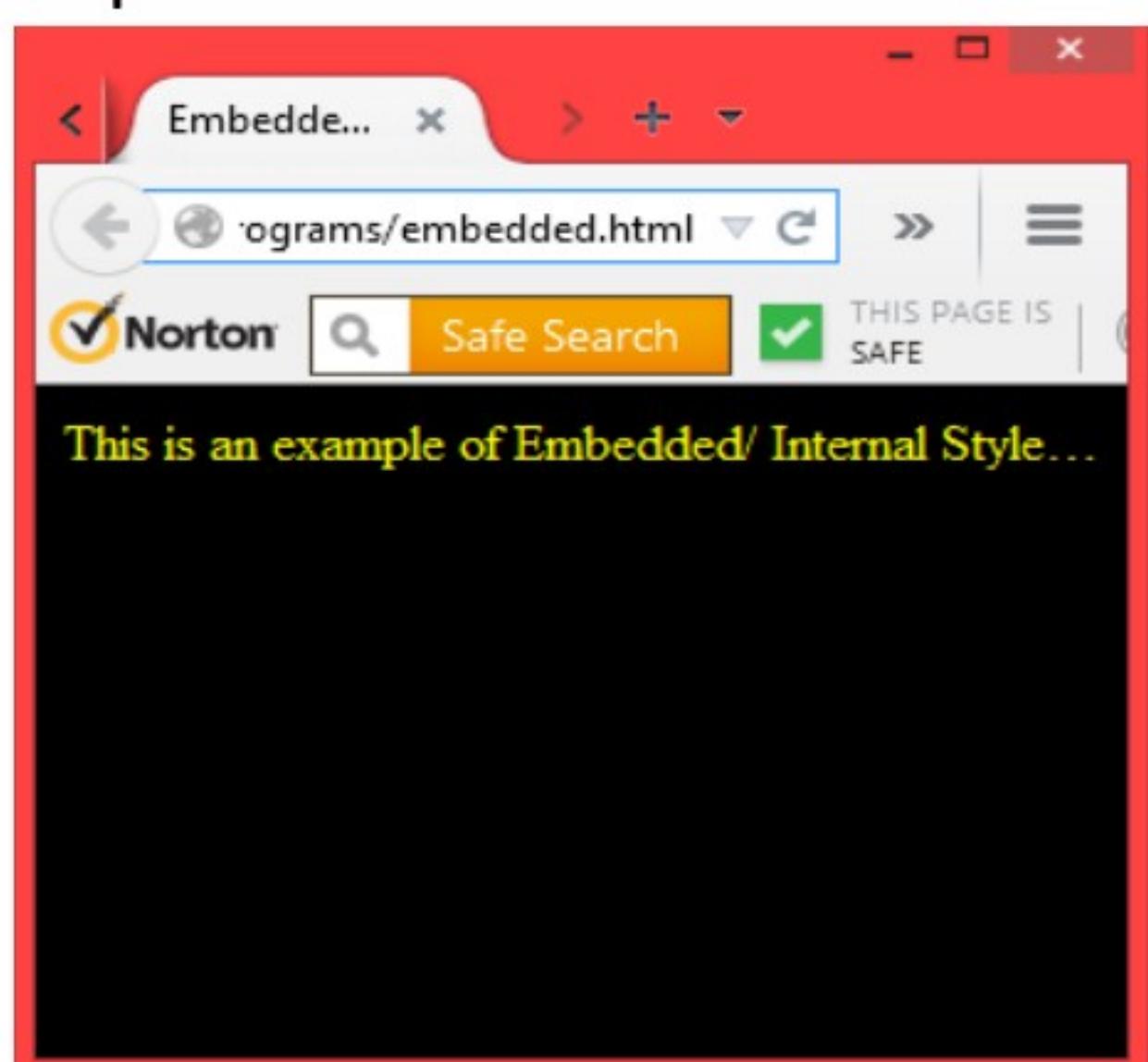
```
<style type="text/css">
    Selector {property: value;}
    Selector {property: value;}
    .....
</style>
```

- The type attribute allows browsers to treat this code as CSS.

### Example: Embedding Stylesheet

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

### Output:



### 3) External/Linked Style Sheet

- External styles 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 your HTML document using the **<link>** tag instead of **<style>**.
- For example, let's create style.css file having below code.

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

p
{ color: blue; }

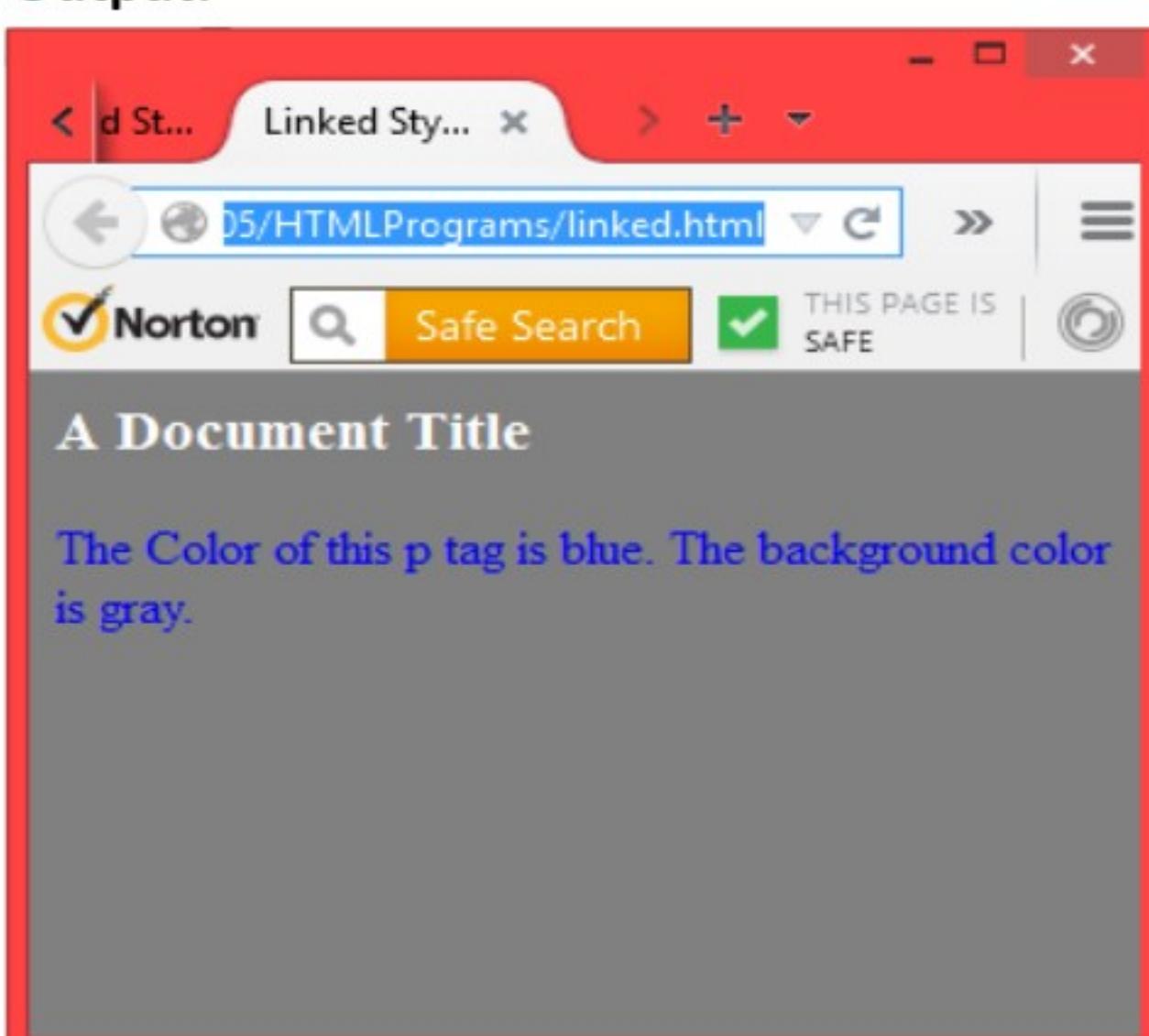
h3
{ color: white; }
```

- Now create a new HTML webpage and link it with style.css as shown in below example.

#### Example: Using Linked Stylesheet

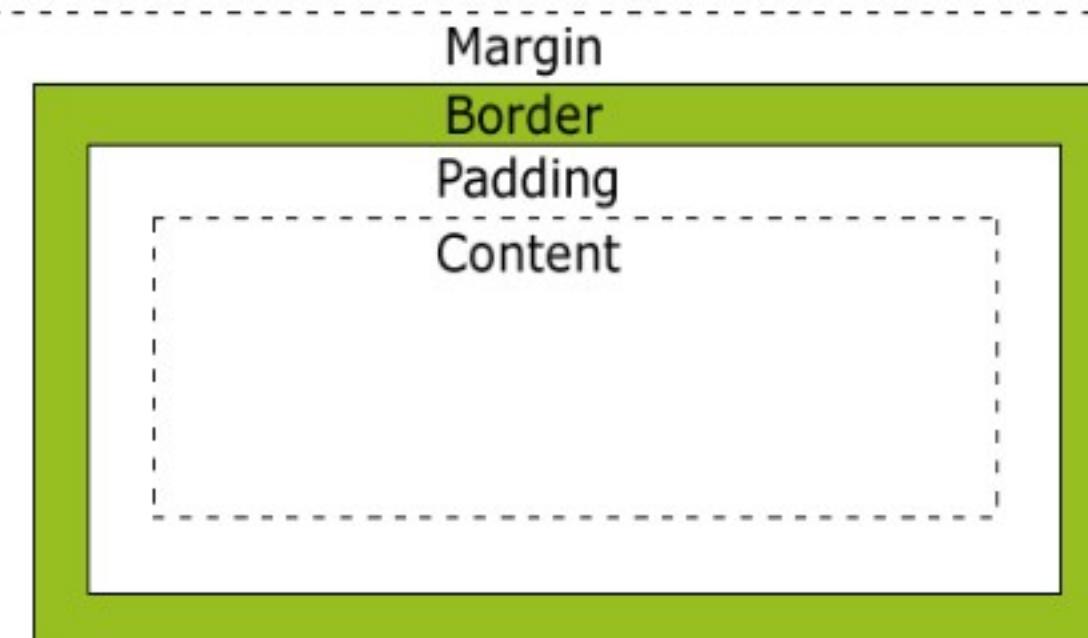
```
<html>
<head>
    <title>Linked Style Sheets</title>
    <link rel="stylesheet" href="style.css" type="text/css"/>
</head>
<body>
    <h3> A Document Title </h3>
    <p> The Color of this p tag is blue. The background color is gray. </p>
</body>
</html>
```

#### Output:



## The CSS Box Model

- All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.
- 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.



### Example: Using Linked Stylesheet

```
<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>
```

#### Output:

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

## CSS Margin

- The CSS margin properties define 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.

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

### Example: Applying margins for paragraph tag.

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

- The margin property can have from one to four values.
  - 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

## 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.

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

### Example: Applying padding for paragraph tag.

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

- The padding property can have from one to four values.
  - 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

## CSS Text

### All CSS Text 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 <b>RGB</b> value like "rgb(255,0,0)", a <b>Color</b> 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 <b>centered</b> , or <b>aligned to the left or right</b> , or <b>justified</b> .	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; }
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.	h1 { text-shadow: 2px 2px 8px #FF0000;

	<p>(2) v-shadow(Required): The position of the vertical shadow.</p> <p>(3) blur-radius(Optional): The blur radius. Default value is 0.</p> <p>(4) color(Optional): The color of the shadow.</p> <p>(5) none: Default value. No shadow.</p>	}
letter-spacing	This property increases or decreases the space between characters in a text.	<pre>h1 { letter-spacing: 2px; } h2 { letter-spacing: -3px; }</pre>
line-height	This property specifies the line height.	<pre>p.small { line-height: 70%; }</pre>

## CSS Font

### All CSS Font Properties

Property	Description	Example
font	Sets all the font properties in one declaration.	<pre>p { font: italic bold 12px/30px Georgia, serif; }</pre>
font-family	Specifies the font family for text. There are two types of font family names: (1) <b>family-name</b> like "times", "courier", "arial", etc. (2) <b>generic-family</b> like "serif", "sans-serif", etc.	<pre>p.serif { font-family: "Times New Roman", Times, serif; }</pre>
font-size	Specifies the font size of text. Different Property values for font-size are <b>medium</b> , <b>xx-small</b> , <b>x-small</b> , <b>small</b> , <b>large</b> , <b>x-large</b> , <b>xx-large</b> , <b>smaller</b> , <b>larger</b> , <b>length</b> , <b>initial</b> and <b>inherit</b> ;	<pre>h1 { font-size:150%; } div { font-size:larger; }</pre>
font-style	Specifies the font style for text. This property has three values: (1) <b>normal</b> : The text is shown normally (2) <b>italic</b> : The text is shown in italics (3) <b>oblique</b> : This text is very similar to italic but less supported.	<pre>p.normal { font-style: normal; }</pre>
font-variant	Specifies whether or not a text should be displayed in a small-caps font. It has four property values: (1) <b>normal</b> : Displays a normal font. (2) <b>small-caps</b> : Displays a small-caps font	<pre>p.normal { font-variant: normal; } p.small</pre>

	<p>(3) <b>initial</b>: Sets this property to its default value.</p> <p>(4) <b>inherit</b>: Inherits this property from its parent element.</p>	<pre>{ font-variant: small-caps; }</pre>
font-weight	<p>The font-weight property sets how thick or thin characters in text should be displayed. It has following property values: <b>normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, initial and inherit</b>.</p>	<pre>p.normal { font-weight: normal; }  p.thick { font-weight: bold; }</pre>

## CSS Link

- Links can be styled with any CSS property (e.g. color, font-family, background, etc.). In addition, links can be styled differently depending on what state they are in.
- 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:**

<b>/* unvisited link */</b> a:link { color: #FF0000; }	<b>/* visited link */</b> a:visited { color: #00FF00; }	<b>/* mouse over link */</b> a:hover { color: #FF00FF; }	<b>/* selected link */</b> 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

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

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

### 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.

#### Example:

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

```
<div class="fixed"> This div element has position: fixed; </div>
</body>
</html>
```

### Output:

This div element has position: static;

This div element has position: fixed;

### Position: relative

- 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 absolute properties** can work to overlap two div on webpage. **An absolute position** element is positioned relative to the first parent element that has a position other than static.
- If no such element is found, the containing block is <html>. Absolutely positioned elements are removed from the normal flow. Absolutely positioned elements can overlap other elements.

#### Example:

```
<html>
<head>
    <style>
        div.relative
        {
            position: relative;
            width: 400px;
            height: 200px;
            border: 3px solid #8AC007;
        }
        div.absolute
        {

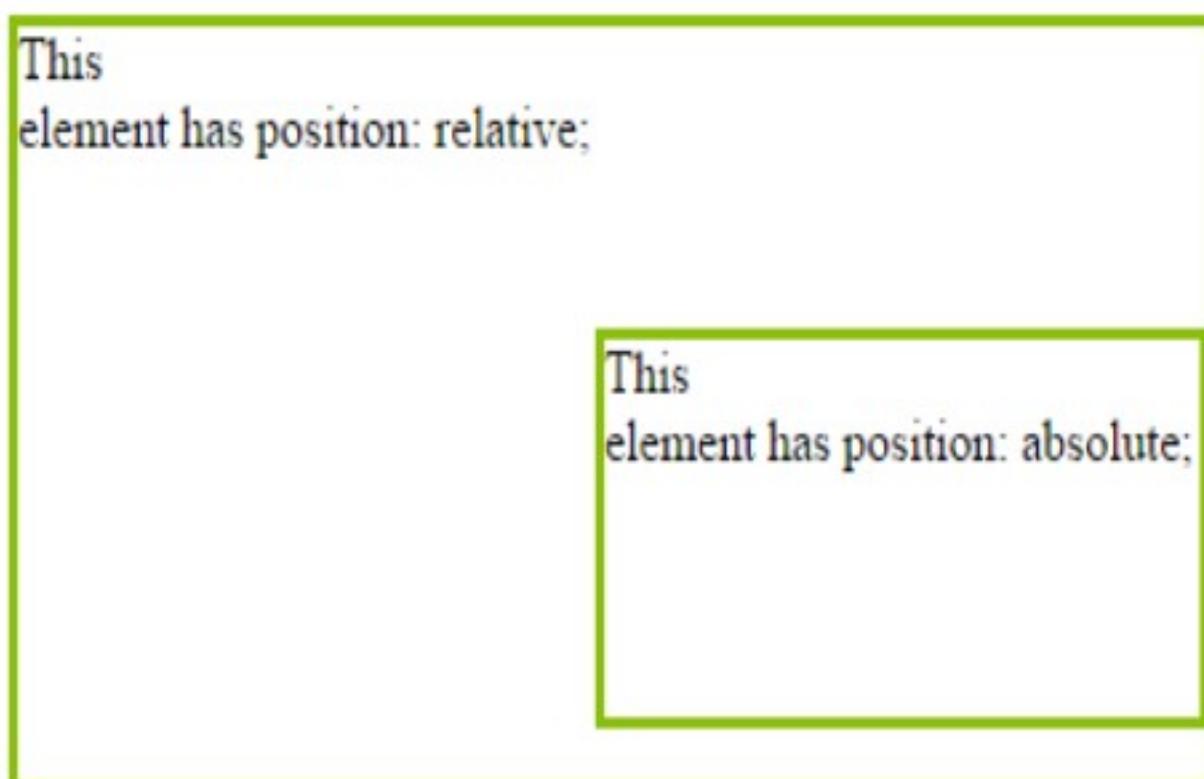
```

```

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

```

**Output:**



## 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.

**Example:**

```

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

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

```

## Output:



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

## Float Properties

- The float property specifies whether or not a box (an element) should float.
- Syntax:** float: none|left|right|initial|inherit;

### Example:

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

### Output:



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

## Difference between CSS2 and CSS3

- CSS3 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 module 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.
- **Syntax:** opacity: number

**Example:**

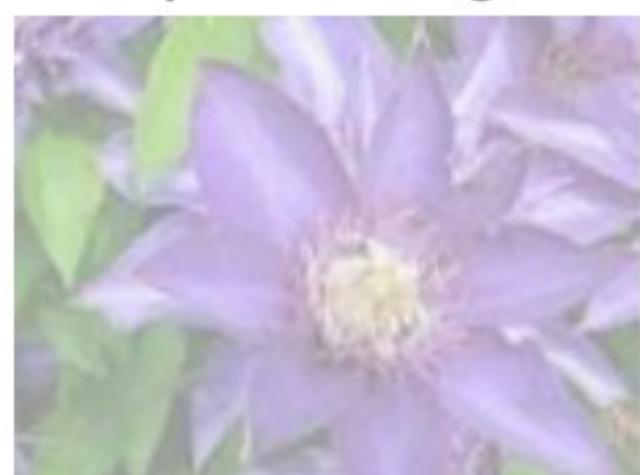
```
<html>
<head>
<style>
    img { opacity: 0.4; }
</style>
</head>
<body>
    
</body>
</html>
```

**Output:**

**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;

**Example:**

```
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;

**Example:**

```

<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**.

## 1) 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, ...);

**Example:** Linear Gradients

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

**Output:**



## 2) Radial Gradients

- **Radial Gradient** 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);

**Example:** Radial Gradients

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

**Output:**



## Transition Animation

- **CSS3 transitions** allows you to change property values smoothly (from one value to another), over a given duration.
- **The transition attribute** takes several parameters:
  - **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.

**Example:**

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

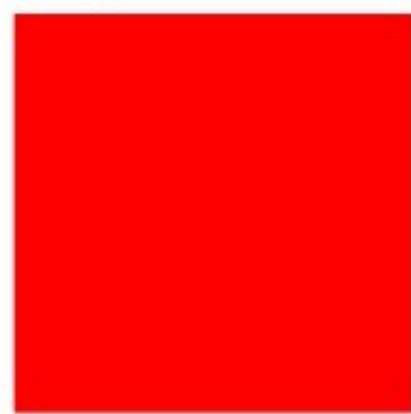
```

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

```

**Output:**

Original Box



After Hover Effect



## Transformations

- 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.
- 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).
- **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 takes six parameters, containing mathematical functions, which allows you to rotate, scale, move (translate), and skew elements.

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

**Example:**

**In CSS:**

```
#box1, #box2, #box3, #box4, #box5, #box6{
    width: 75px;
    height: 40px;
    border: 3px solid black;
    background-color: yellow; }

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

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

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

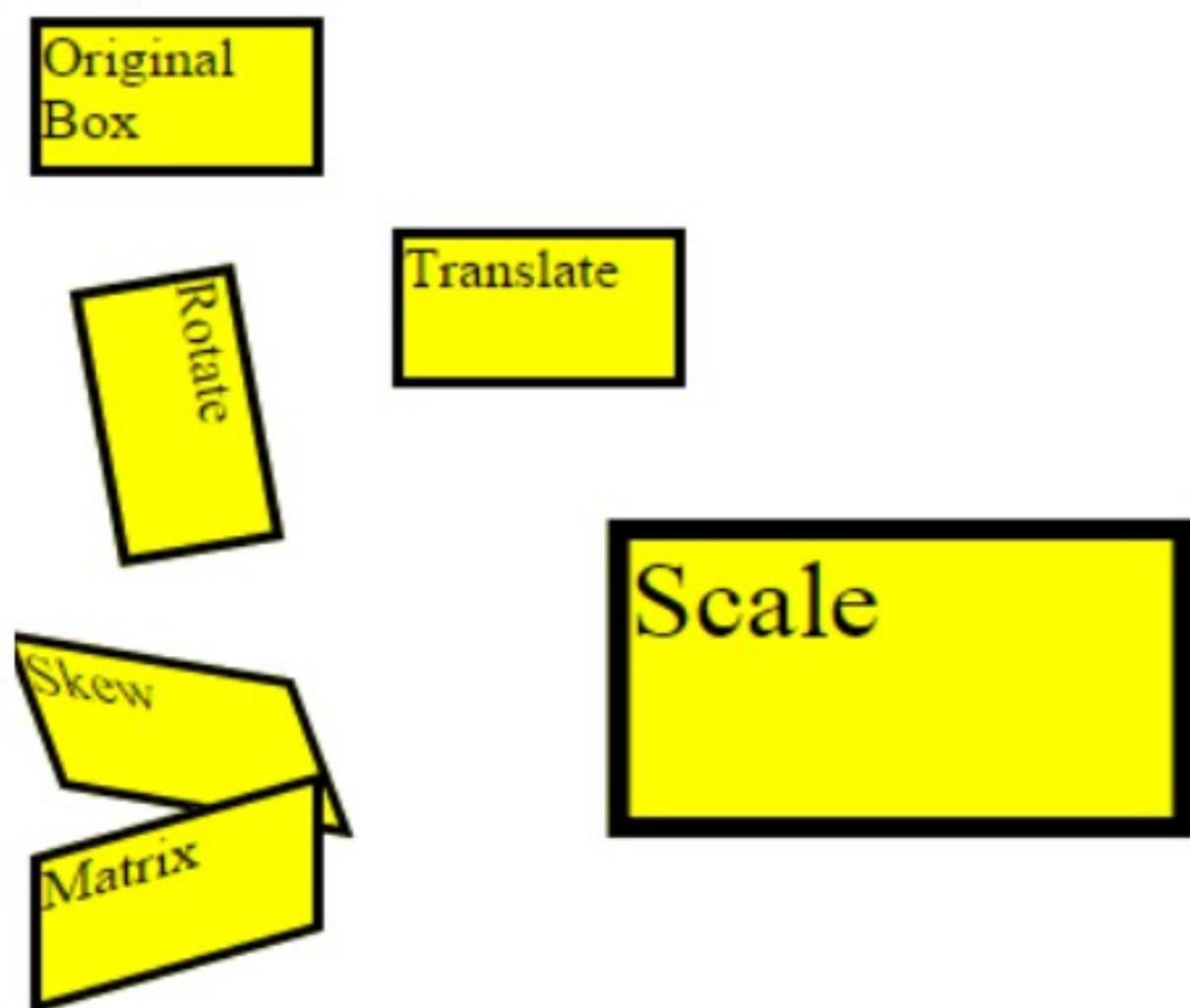
#box5 {
    transform: skew(20deg,10deg);}

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

**In Body Tag:**

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

## Output:



## Animations

- 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;}}

### Example:

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

```
}
```

```
@keyframes example
```

```
{
```

```
    from
```

```
    {
```

```
        background-color: red;
```

```
    }
```

```
    to
```

```
    {
```

```
        background-color: green;
```

```
    }
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
    <div>Red to green</div>
```

```
</body>
```

```
</html>
```

**Output:**

Red to green

