UNIT-I

HTML :Basic Syntax, Standard HTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, HTML5.
CSS: Levels of Style Sheets, Style Specification Formats, Selector Forms, The Box Model, Conflict Resolution.

- HTML stands for Hyper Text Markup Language
- HTML was developed by Tim Berners Lee.
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML formatting is specified by using tags.
- A tag is format name surrounded by angle brackets. End tag which switch a format off also contains a "forward slash">.
- EX:- <tagname>Content goes here...</tagname>
- HTML tags are not case sensitive.
- EX:-<head>,<HEAD>,<HEad> are equivalents.

STRUCTURE OF HTML DOCUMENT:

Structure of HTML document consists of two sections. One is Head section and the other is body section.

- All HTML document follow the basic structure. They have a head which contains control information used by the browser and server and a large body.
- The body contains the content that the display on the screen and tags which control how the content is formatted by the browser.
- The basic document is :

```
<!DOCTYPE html>
< html >
< head >
< title > An HTML DOCUMENT < /title >
</head>
<body>
<h1> My First Heading </h1>
<p> My first paragraph.</p>
</body>
</html>
```

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

HTML Element:

An HTML element is defined by a start tag, some content, and an end tag:

<tagname>Content goes here...</tagname>

The HTML **element** is everything from the start tag to the end tag:

<h1>My First Heading</h1>

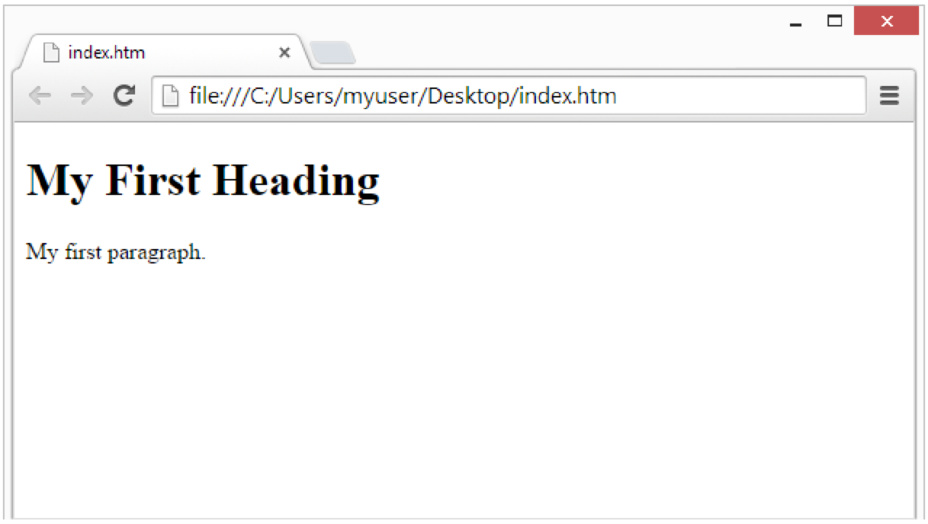
<p>My first paragraph.</p>

Start tag	**Element content**	**End tag**
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
<br>	*none*	*none*

Note: Some HTML elements have no content (like the <br> element). These elements are called empty elements. Empty elements do not have an end tag!

Web Browser:

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.
A browser does not display the HTML tags, but uses them to determine how to display the document:

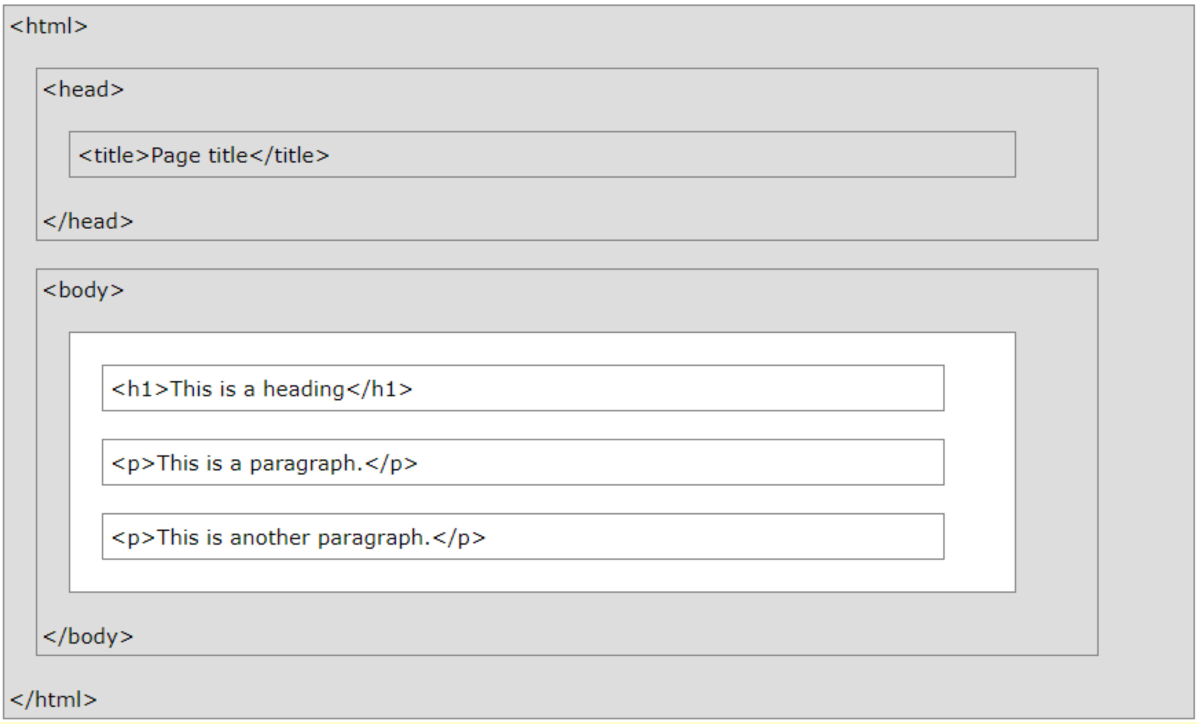


My First Heading

My first paragraph.

HTML Page Structure :

Below is a visualization of an HTML page structure:



```
<html>
    <head>
        <title>Page title</title>
    </head>
    <body>
            <h1>This is a heading</h1>
            <p>This is a paragraph.</p>
            <p>This is another paragraph.</p>
    </body>
</html>
```

Note: Only the content inside the <body> section (the white area above) will be displayed in a browser.

HTML Using Notepad:

Learning HTML we recommend a simple text editor like Notepad (PC) . We believe in that using a simple text editor is a good way to learn HTML. Follow the steps below to create your first web page with Notepad.

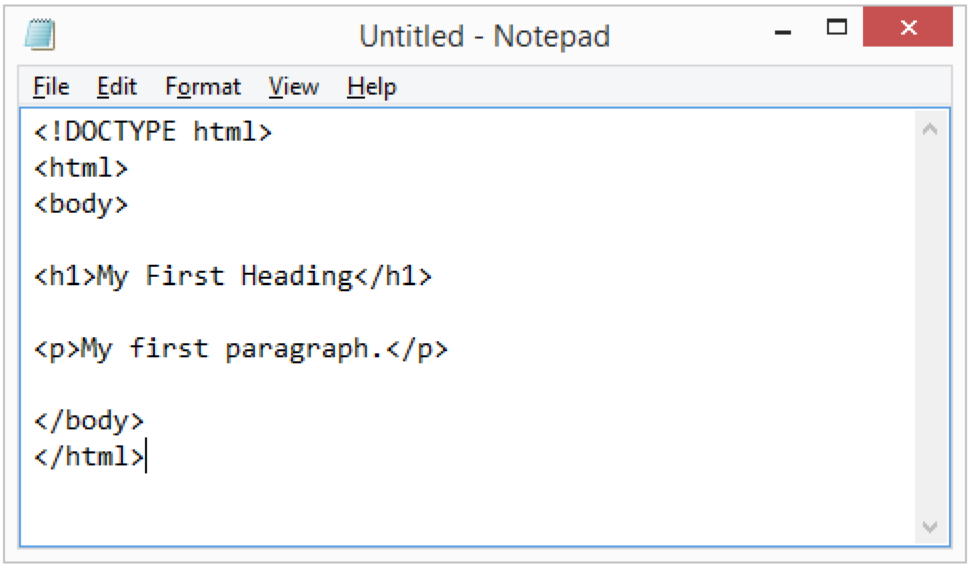
Step 1: Open Notepad (PC)

Start > **Programs** > **Accessories** > **Notepad**

Step 2: Write Some HTML

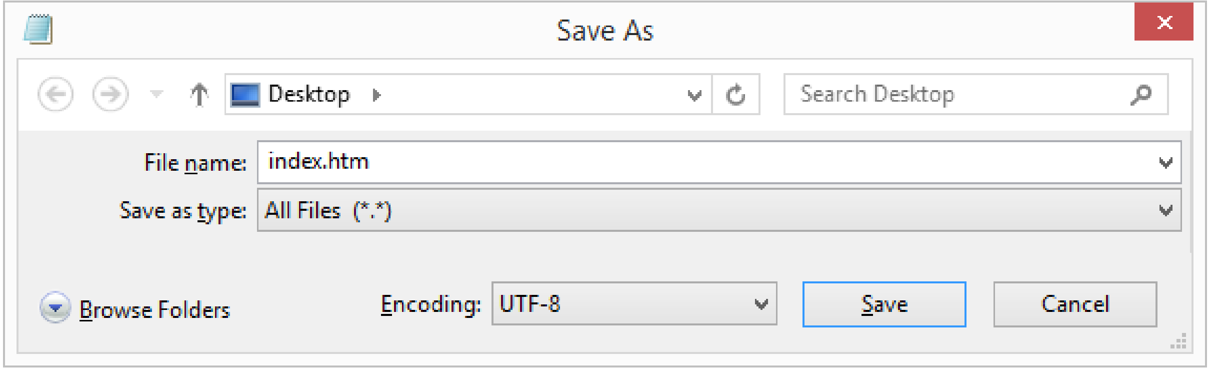
Write or copy the following HTML code into Notepad:

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

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```



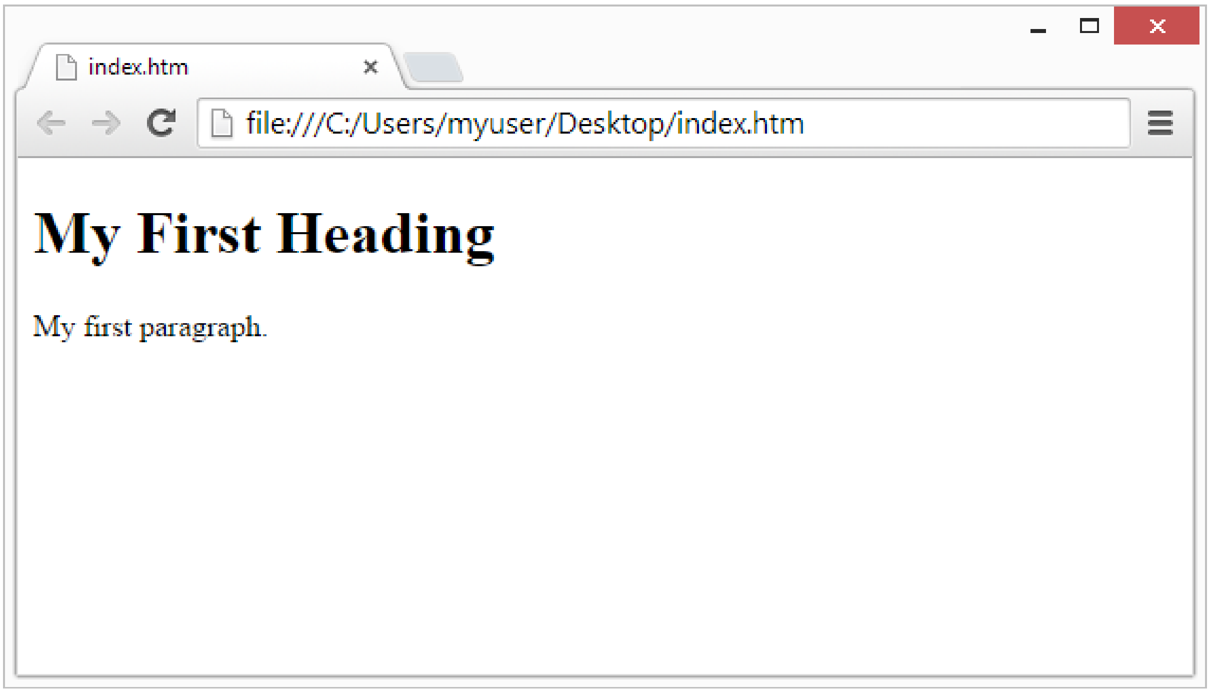
```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Step 3: Save the HTML Page
Save the file on your computer. Select **File** > **Save as** in the Notepad menu.
Name the file **"index.htm"** and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).



File name: index.htm
Save as type: All Files (*.*)
Encoding: UTF-8

Step 4: View the HTML Page in Your Browser
Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").
The result will look much like this:



file:///C:/Users/myuser/Desktop/index.htm

My First Heading

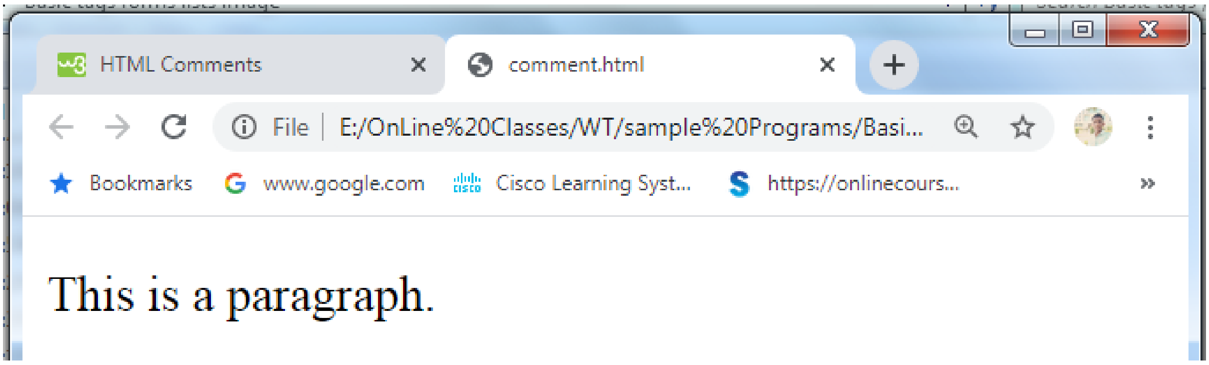
My first paragraph.

HTML Comments:

HTML comments are not displayed in the browser, but they can help document your HTML source code. You can add comments to your HTML source by using the following syntax. Notice that there is an exclamation point (!) in the start tag, but not in the end tag.

```
<!-- Write your comments here -->
```

```
<!DOCTYPE html>
<html>
<body>
<!-- This is a comment -->
<p>This is a paragraph.</p>
<!-- Comments are not displayed in the browser -->
</body>
</html>
```



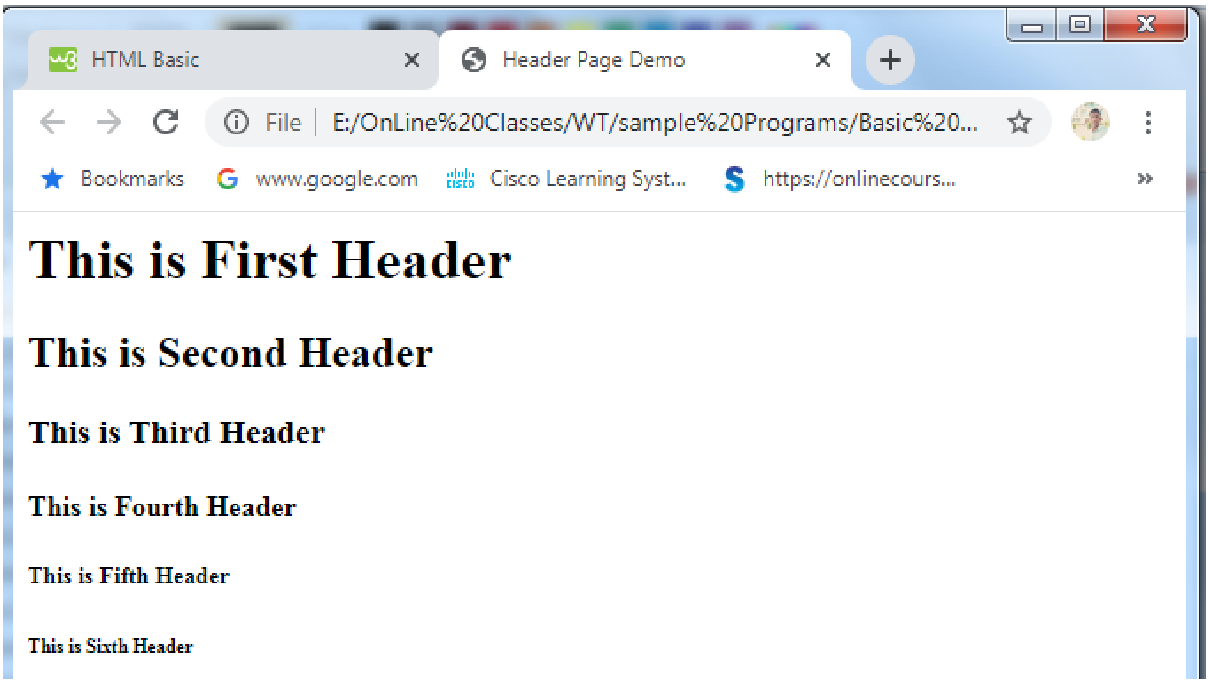
This is a paragraph.

Basic Text Markup:

HTML Headings:

HTML headings are defined with the <h1> to <h6> tags. HTML headings are titles or subtitles that you want to display on a webpage. <h1> defines the most important heading. <h6> defines the least important heading. <h1> headings should be used for main headings, followed by <h2> headings, then the less important <h3>, and so on. Each HTML heading has a default size. Browsers automatically add some white space (a margin) before and after a heading.

```
<!DOCTYPE html>

<html>
<head><title>Header Page Demo</title></head>
<body>
<h1>This is First Header</h1>
<h2>This is Second Header</h2>
<h3>This is Third Header</h3>
<h4>This is Fourth Header</h4>
<h5>This is Fifth Header</h5>
<h6>This is Sixth Header</h6>
</body>
</html>
```



This is First Header

This is Second Header

This is Third Header

This is Fourth Header

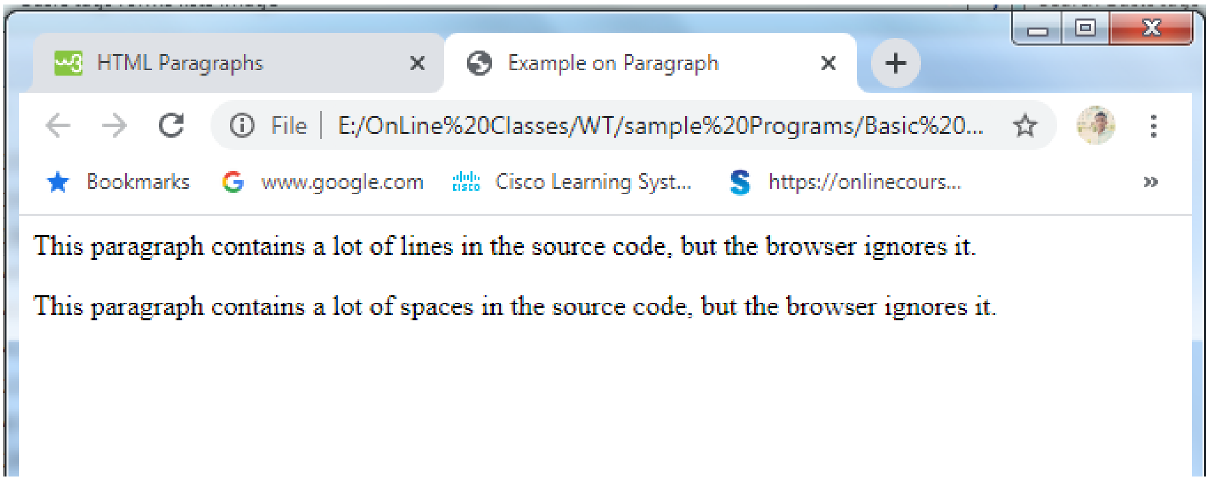
This is Fifth Header

This is Sixth Header

HTML Paragraphs:

The HTML <p> element defines a paragraph. A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph. The browser will automatically remove any extra spaces and lines when the page is displayed. A paragraph always starts on a new line, and is usually a block of text.

```
<!DOCTYPE html>
<html>
<head><title>Example on Paragraph</title></head>
<body>
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>

<p>
This paragraph
contains         a lot of spaces
in the source         code,
but the        browser
ignores it.
</p>
</body>
</html>
```



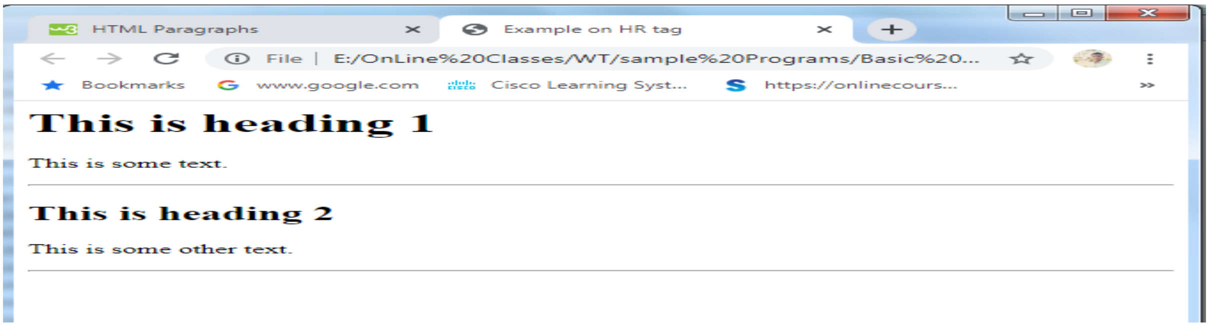
This paragraph contains a lot of lines in the source code, but the browser ignores it.

This paragraph contains a lot of spaces in the source code, but the browser ignores it.

HTML Horizontal Rules:

The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule. The <hr> element is used to separate content (or define a change) in an HTML page. The <hr> tag is an empty tag, which means that it has no end tag.

```
<!DOCTYPE html>
<html>
<head><title>Example on HR tag</title></head>
<body>
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
</body>
</html>
```



This is heading 1

This is some text.

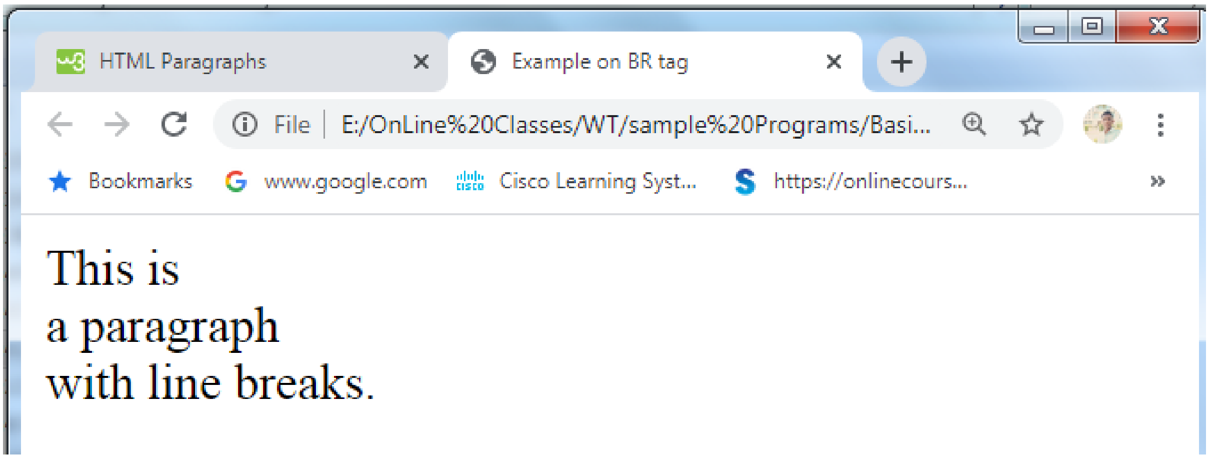
This is heading 2

This is some other text.

HTML Line Breaks:

The HTML <br> element defines a line break. Use <br> if you want a line break (a new line) without starting a new paragraph. The <br> tag is an empty tag, which means that it has no end tag.

```
<!DOCTYPE html>
<html>
<body>
<p>This is<br>a paragraph<br>with line breaks.</p>
</body>
</html>
```

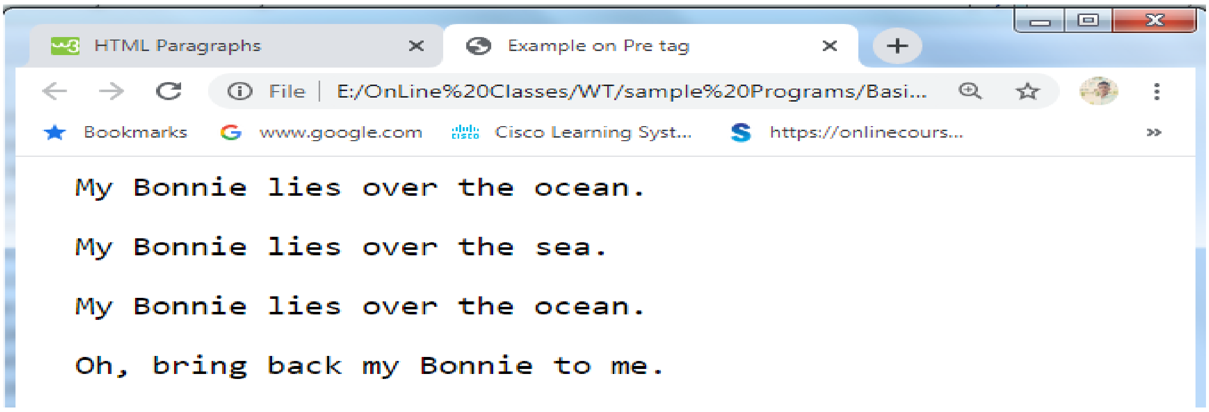


```
This is
a paragraph
with line breaks.
```

HTML <pre> Element:

The HTML <pre> element defines preformatted text. The text inside a <pre> element is displayed in a fixed-width font, and it preserves both spaces and line breaks.

```
<!DOCTYPE html>
<html>
<body>
<p>The pre tag preserves both spaces and line breaks:</p>
<pre>
  My Bonnie lies over the ocean.

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.
</pre>
</body>
</html>
```



```
  My Bonnie lies over the ocean.

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.
```

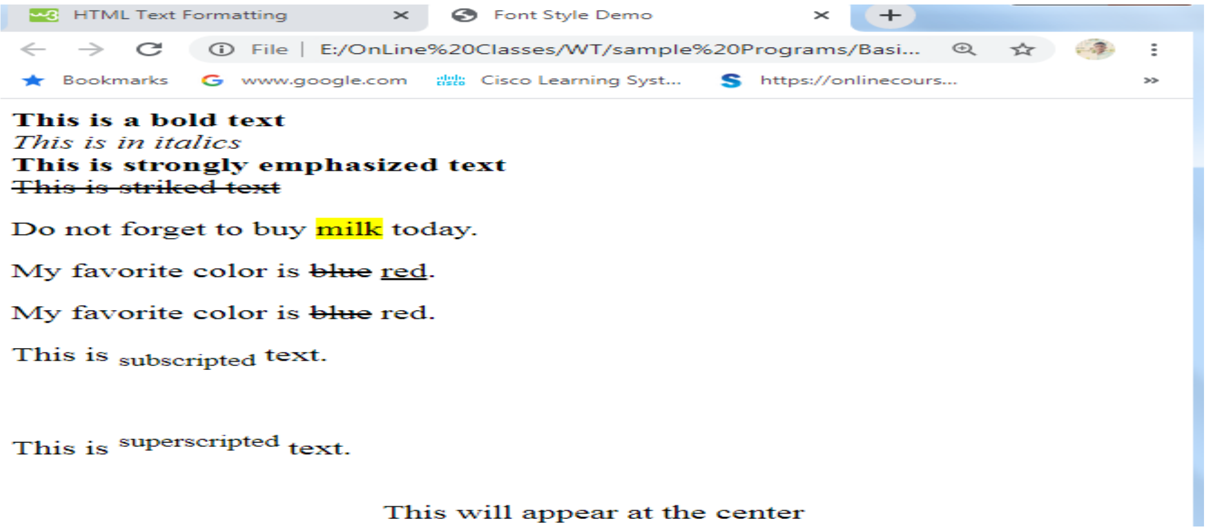
HTML Text Formatting:

HTML contains several elements for defining text with a special meaning. Formatting elements were designed to display special types of text.

- **<b> - Bold text** - The HTML <b> element defines bold text
- **<strong> - Important text-** The HTML <strong> element defines text with strong importance.
- **<i> - Italic text** - The HTML <i> element defines a part of text the content inside is typically displayed in italic.

- **<em> - Emphasized text -** The HTML <em> element defines emphasized text. The content inside is typically displayed in italic. A screen reader will pronounce the words in <em> with an emphasis, using verbal stress.
- **<mark> - Marked text-** The HTML <mark> element defines text that should be marked or highlighted
- **<small> - Smaller text-** The HTML <small> element defines smaller text.
- **<del> - Deleted text -** The HTML <del> element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text:
- **<ins> - Inserted text-** The HTML <ins> element defines a text that has been inserted into a document. Browsers will usually underline inserted text:
- **<sub> - Subscript text-** The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H_2O.
- **<sup> - Superscript text-** The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW[1].

```
<!DOCTYPE html>
<html>
<head><title>Font Style Demo</title></head>
<body>
<b>This is a bold text</b><br/>
<i>This is in italics</i><br/>
<strong>This is strongly emphasized text</strong><br/>
<strike>This is striked text</strike><br/>
<p>Do not forget to buy <mark>milk</mark> today.</p>
<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
<p>My favorite color is <del>blue</del> red.</p>
<p>This is <sub>subscripted</sub> text.</p></br>
<p>This is <sup>superscripted</sup> text.</p></br>
<center>This will appear at the center</center>
</body>
</html>
```



This is a bold text
This is in italics
This is strongly emphasized text
~~This is striked text~~

Do not forget to buy milk today.

My favorite color is ~~blue~~ red.

My favorite color is ~~blue~~ red.

This is $_{\text{subscripted}}$ text.

This is $^{\text{superscripted}}$ text.

This will appear at the center

Images:

To insert an image in your web page by using **<img>** tag. Images can improve the design and the appearance of a web page. Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image. You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in src attribute. Image name is always case sensitive.

The <img> tag is empty, it contains attributes only, and does not have a closing tag.

The <img> tag has two required attributes:

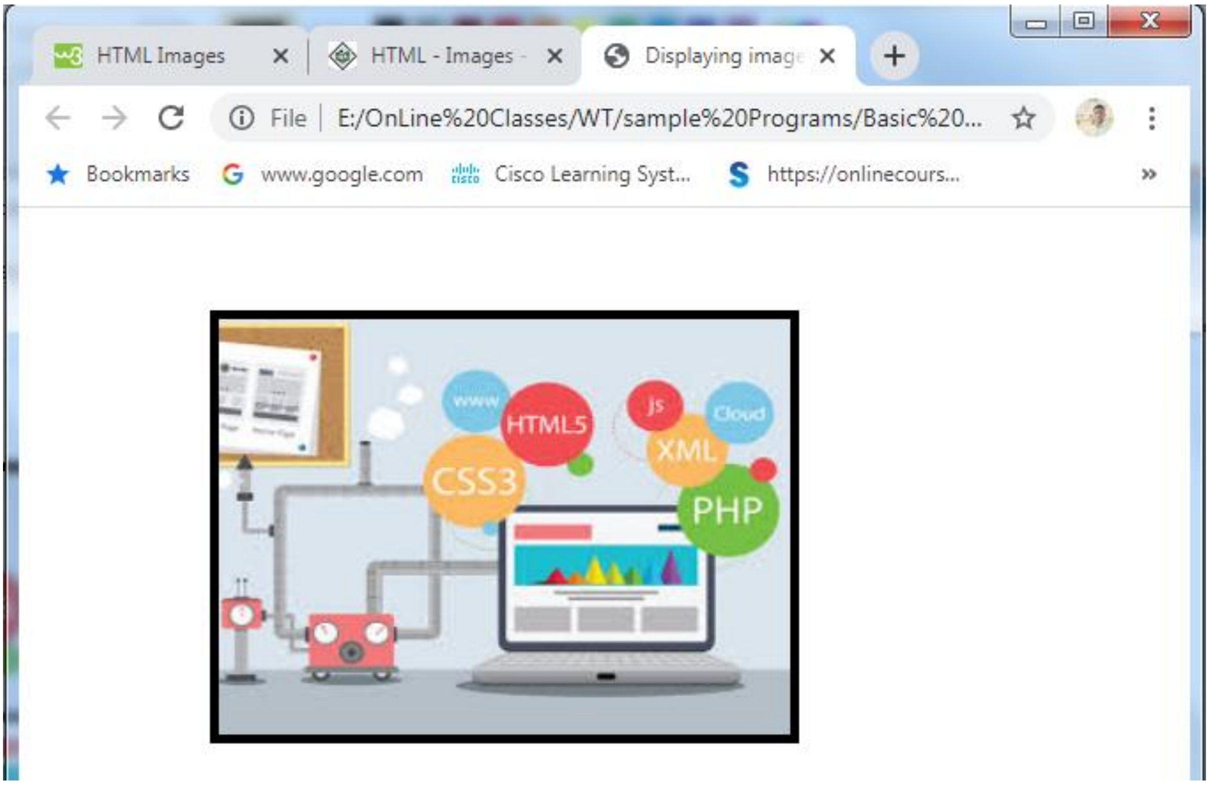
- src - Specifies the path to the image
- alt - Specifies an alternate text for the image

Syntax

<img src="url" alt="*alternatetext*" border="5" hspace="100" vspace="50" width="50%" height="50% >

- **SRC** attribute specifies the path (URL) to the image.
- **ALT** attribute is used to specify the text to be displayed in place of an image for browsers that can not handle graphics or have graphics disabled.
- **ALIGN** sets the alignment of text, which follows the IMG reference relative to the image on the screen. Possible settings are LEFT, RIGHT, TOP, MIDDLE, and CENTER Alignment of the image in the browser window and the text that follows will wrap around the image according to the setting used.
- **BORDER** sets whether or not the image has a border, and if so, how thick the border is.
- **HSPACE** sets the horizontal spacing (both left and right sides) around the image in pixels.
- **VSPACE** sets the vertical spacing (both top and bottom sides) around the image in pixels.
- **WIDTH and HEIGHT -** sets image width and height based on your requirement using width and height attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

```
    <!DOCTYPE html>
<html>
<head><title>Displaying image demo</title></head>
<body >
<img src="download.jpg" alt="web technologies image" border="5" hspace="100"
vspace="50" width="50%" height="50%">
</body>
</html>
```



Hypertext Links:

A webpage can contain various links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks. Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus you can create hyperlinks using text or images available on a webpage. You can click on a link and jump to another document. When you move the mouse over a link, the mouse arrow will turn into a little hand.

A link is specified using HTML tag <a>. This tag is called **anchor tag** and anything between the opening <a> tag and the closing </a> tag becomes part of the link and a user can click that part to reach to the linked document.

Following is the simple syntax to use <a> tag.

<a href = "Document URL" ... attributes-list target=”where to display”>Link Text</a>

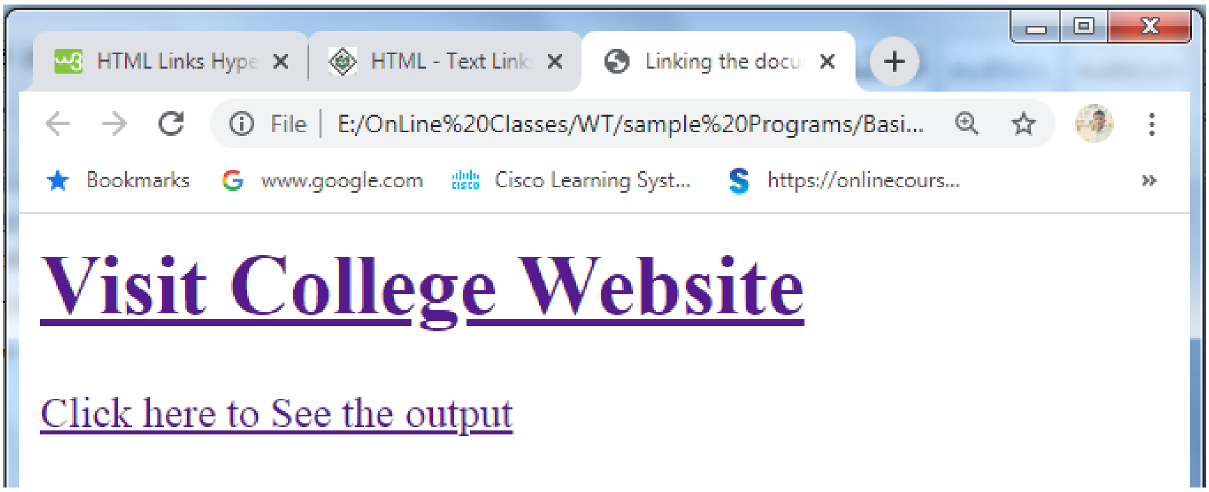
The target Attribute

We have used **target** attribute in our previous example. This attribute is used to specify the location where linked document is opened. Following are the possible options −

Sr.No	Option & Description
1	**_blank** Opens the linked document in a new window or tab.
2	**_self** Opens the linked document in the same frame.

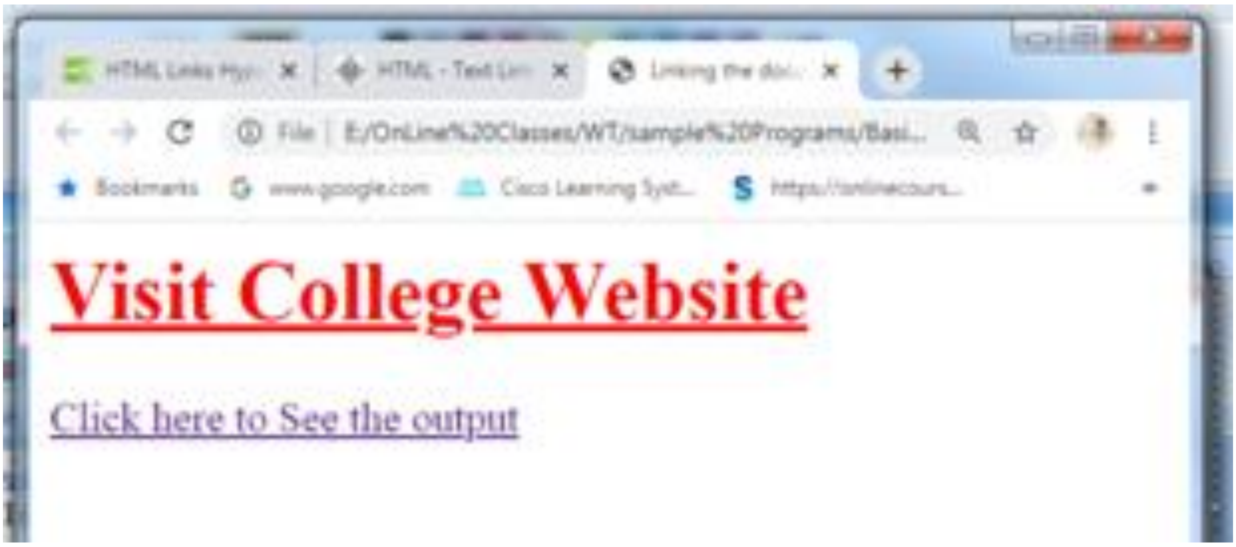
3	**_parent** Opens the linked document in the parent frame.
4	**_top** Opens the linked document in the full body of the window.
5	**targetframe** Opens the linked document in a named *targetframe*.

```
<!DOCTYPE html>


<html>
<head><title>Linking the document</title></head>
<body >
<a href="https://www.vvitguntur.com/" target="_blank"><h1>Visit College Website</h1></a>
<a href="1 first.html">Click here to See the output</a>
</body>
</html>
```



Visit College Website

Click here to See the output



Visit College Website

Click here to See the output

Lists:

Lists is the collection of elements or items. HTML lists allow web developers to group a set of related items in lists. In HTML, you can display information in the form of lists.

There are three varieties of lists they are.

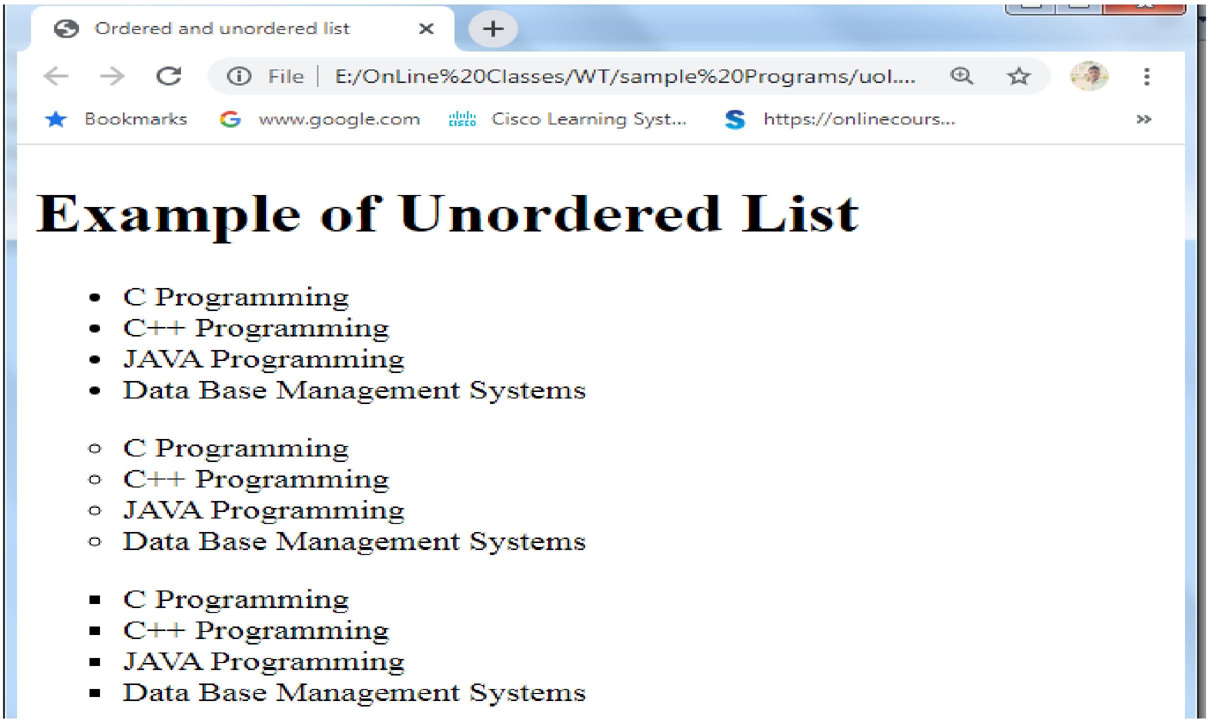
- Unordered HTML List
- Ordered HTML List
- HTML Description Lists

Unordered HTML List:

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag. The list items will be marked with bullets (small black circles) by default. Three values are present for the attribute type 1.Disc 2.Circle 3.Square.

Syntax:- <UL type=”disc”/”square”/”circle” > ------ - - - - </UL>

```
<!DOCTYPE html>
<HTML>
 <HEAD>   <TITLE> Ordered and unordered list </TITLE>  </HEAD>
 <BODY>
  <h1> Example of  Unordered List</h1>
  <ul>
  <li>C Programming</li>
  <li>C++ Programming</li>
  <li> JAVA Programming</li>
  <li>Data Base Management Systems</li>
  </ul>

  <ul type="circle">
  <li>C Programming</li>
  <li>C++ Programming</li>
  <li> JAVA Programming</li>
  <li>Data Base Management Systems</li>
  </ul>
<ul type="square">
  <li >C Programming</li>
  <li>C++ Programming</li>
  <li> JAVA Programming</li>
  <li>Data Base Management Systems</li>
  </ul>
  </BODY>
</HTML>
```


Ordered and unordered list
File
E:/OnLine%20Classes/WT/sample%20Programs/uol....
Bookmarks
www.google.com
Cisco Learning Syst...
https://onlinecours...

Example of Unordered List

- C Programming
- C++ Programming
- JAVA Programming
- Data Base Management Systems

○ C Programming
○ C++ Programming
○ JAVA Programming
○ Data Base Management Systems

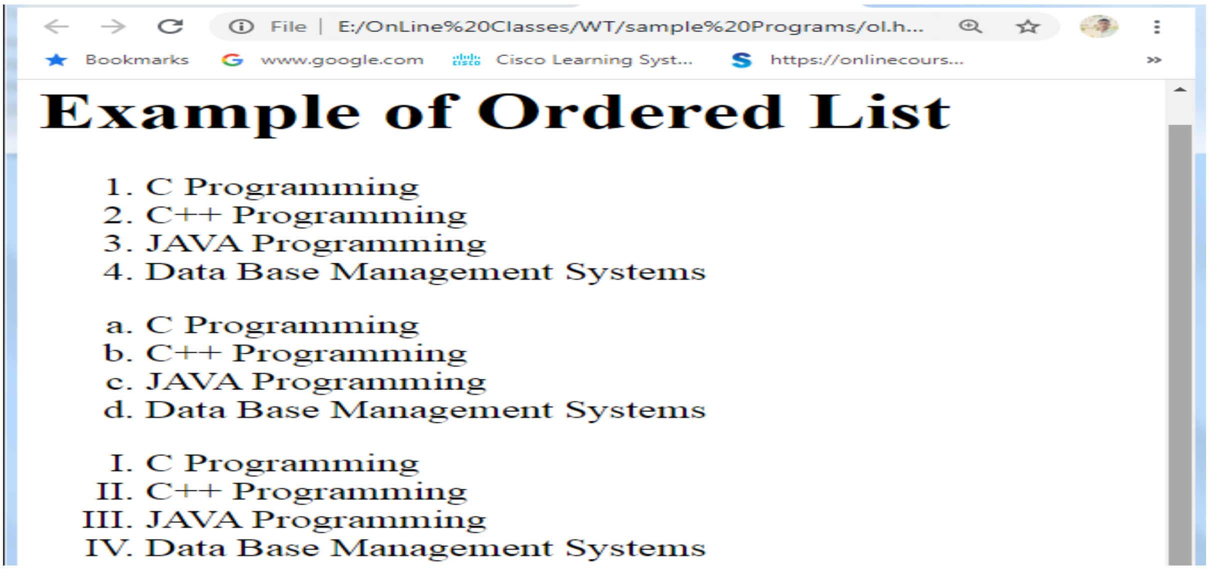
▪ C Programming
▪ C++ Programming
▪ JAVA Programming
▪ Data Base Management Systems

Ordered HTML List:

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag. The list items will be marked with numbers by default. An ordered list elements must be enclosed with in <li> - - - - - </li> tags.

Syntax:- <ol type=”i”/”a”/”A”/”a” start =”n” >- - - - - - </ol>

```
<!DOCTYPE html>
<HTML>
 <HEAD>   <TITLE> Ordered and unordered list </TITLE>  </HEAD>
 <BODY>
  <h1> Example of  Ordered List</h1>
  <ol>
  <li>C Programming</li>
  <li>C++ Programming</li>
  <li> JAVA Programming</li>
  <li>Data Base Management Systems</li>
  </ol>

  <ol type="a">
  <li>C Programming</li>
  <li>C++ Programming</li>
  <li> JAVA Programming</li>
  <li>Data Base Management Systems</li>
  </ol>
<ol type="I">
  <li >C Programming</li>
  <li>C++ Programming</li>
  <li> JAVA Programming</li>
  <li>Data Base Management Systems</li>
  </ol>
  </BODY>
</HTML>
```



Example of Ordered List

1. C Programming
2. C++ Programming
3. JAVA Programming
4. Data Base Management Systems

a. C Programming
b. C++ Programming
c. JAVA Programming
d. Data Base Management Systems

I. C Programming
II. C++ Programming
III. JAVA Programming
IV. Data Base Management Systems

HTML Description Lists:

A description list is a list of terms, with a description of each term. The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term.

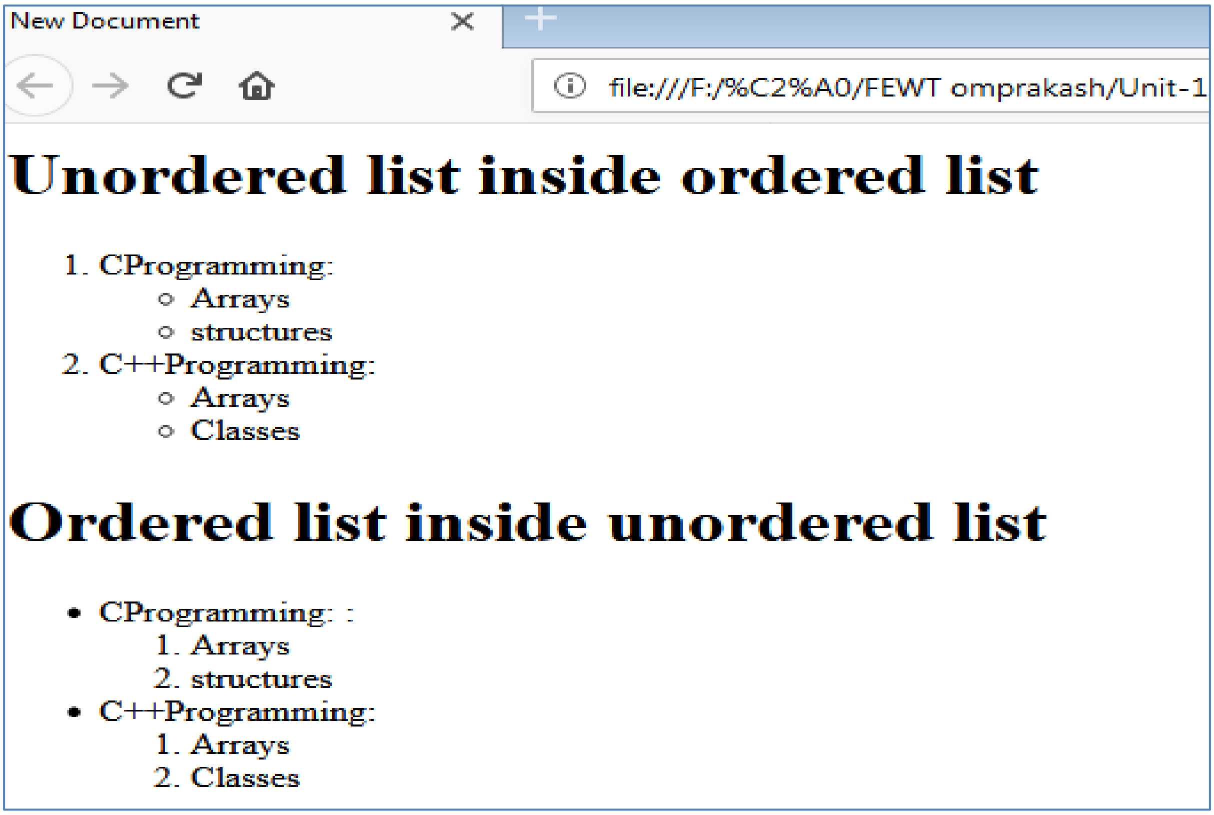
Syntax :

```
<dl>
  <dt>term</dt>
  <dd>description</dd>
</dl>
```

Nested Lists:

A nested list or a sublist is a list within a list. The trick to marking nested lists up correctly in HTML is to recognize that the sublist is actually a child of a list item and not of a list. Notice that the sublist is a child and not a sibling of an <li> tag.

```
<!DOCTYPE html>
<HTML>
 <HEAD>
  <TITLE> Nesting of Lists </TITLE>
   </HEAD>
 <BODY>
      <h1> Unordered list inside ordered list</h1>
 <ol>
  <li>CProgramming:<ul><li>Arrays</li> <li>structures</li></ul></li>
  <li>C++Programming:<ul><li>Arrays</li> <li>Classes</li></ul></li>
  </ol>
<h1> Ordered list inside unordered list</h1>
  <ul>
  <li>CProgramming:<ol><li>Arrays</li> <li>structures</li></ol></li>
  <li>C++Programming:<ol><li>Arrays</li> <li>Classes</li></ol></li>
    </ul>
 </BODY>
 </HTML>
```



Unordered list inside ordered list

1. CProgramming:
   - Arrays
   - structures
2. C++Programming:
   - Arrays
   - Classes

Ordered list inside unordered list

- CProgramming: :
  1. Arrays
  2. structures
- C++Programming:
  1. Arrays
  2. Classes

Tables:

HTML tables allow web developers to arrange data into rows and columns. Using table element you can display the content of a webpage in the tabular format. The <TABLE> element is used along with TR, TH and TD elements.

You can set the width and height of the table columns and rows using the style attribute in the table element. Using the style attribute you can also define borders, font style, font size of the contents of the table.

<CAPTION> element can be used to specify the table caption.

The <table> tag defines an HTML table.

Each table row is defined with a <tr> tag. Each table header is defined with a <th> tag. Each table data/cell is defined with a <td> tag.

By default, the text in <th> elements are bold and centered.

By default, the text in <td> elements are regular and left-aligned.

The <td> elements are the data containers of the table. They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

- SYNTAX:

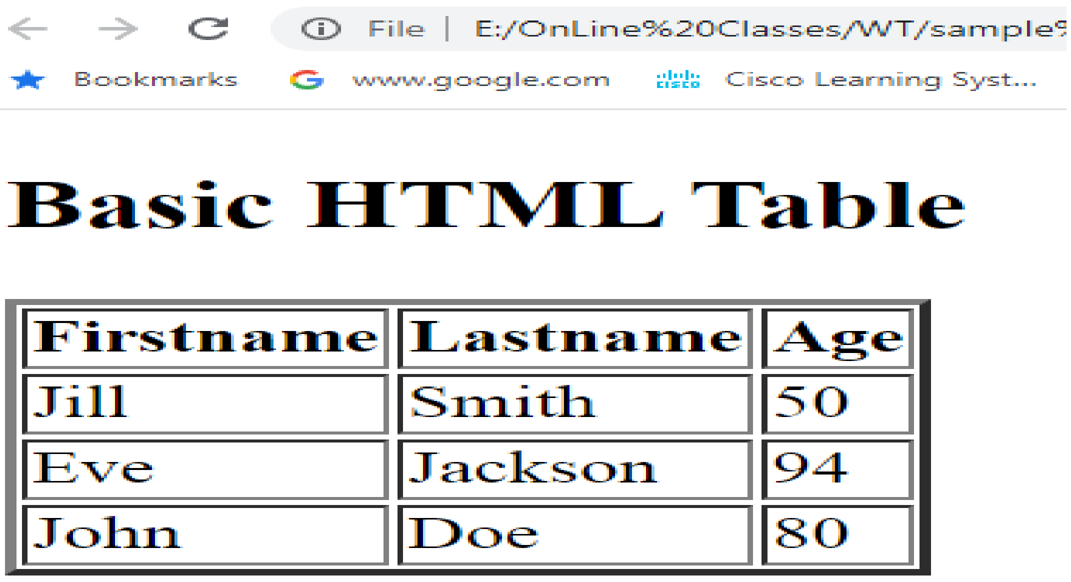
```
<table>
<tr>
        <th> </th>
        <th> </th>
</tr>
<tr>
        <td> </td>
        <td> </td>
</tr>
  </table>
```

Attributes For Table:

Following are the attributes of the <Table> element:

- **ALIGN**: Specifies the horizontal alignment of the table in the browser window. Set to *LEFT, CENTER or RIGHT*.
- **BACKGROUND**: Specifies the URL of a background image to be used as a background for the table. All cell contents are displayed over this image.
- **BGCOLOR**: Sets the background color of the table cells. Set to a RGB triplet or a predefined color name.
- **BORDER**: Sets the borer width in pixels. *0 indicates no border*.
- **BORDERCOLOR**: sets the external border color for the entire table
- **CELLPADDING**: Sets the spacing between cell walls and cell contents in pixels
- **CELLSPACING**: Sets the distance between cells in pixels
- **HEIGHT**: Gives the height of the whole table in pixels
- **WIDTH**: Sets the width of the table; sets to a pixel value or a percentage of the display area.

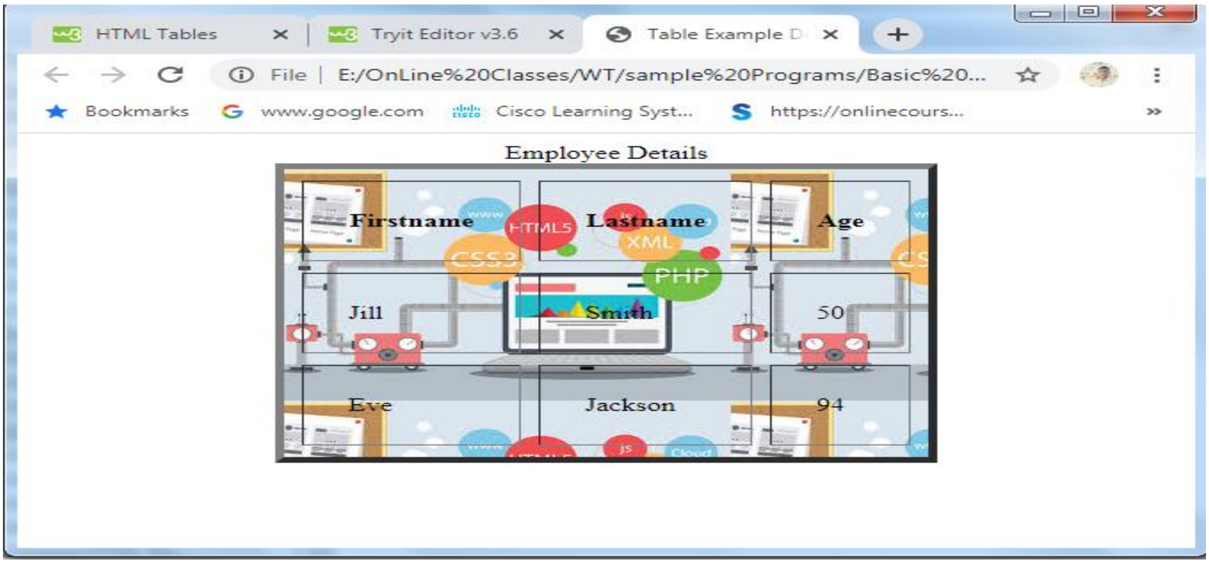
```
<!DOCTYPE html>
<html>
<body>

<h2>Basic HTML Table</h2>

<table border="2" >
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
</html>
```



Basic HTML Table

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

```
<!DOCTYPE html>
<html>
<head><title>Table Example Demo</title>
</head>
<body>
<table border="5"   cellpadding="25" cellspacing="10" align="center" bgcolor="green"
background="download.jpg" width="25%" height="25%">
<caption>Employee Details</caption>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
</body>
</html>
```



Employee Details

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94

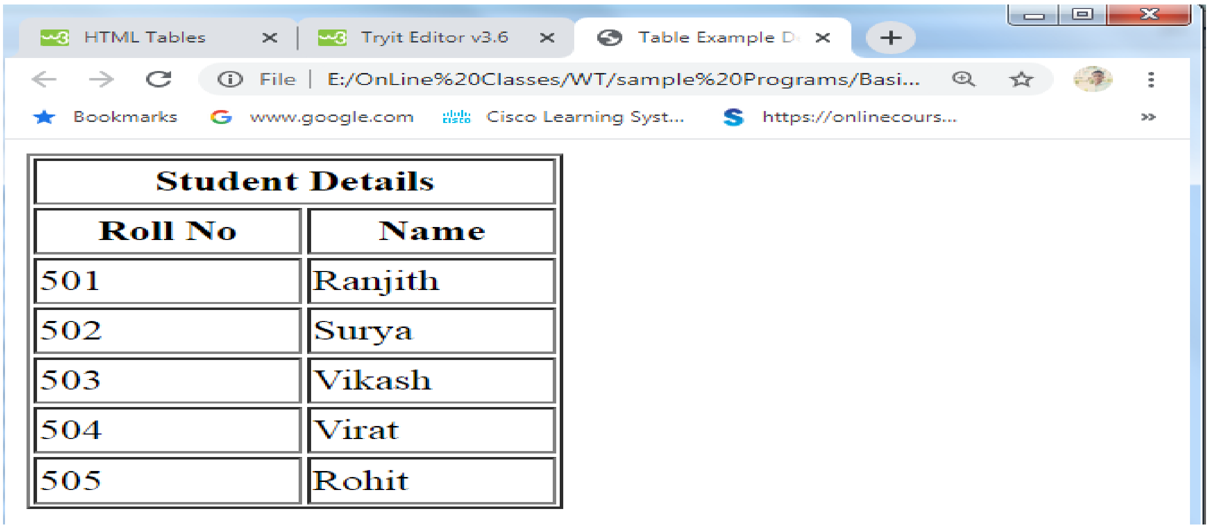
Spanning Rows and Columns:

- Sometimes it is required to combine one or more adjacent cells into one cell.
- The process of combining one or more cells is known as spanning.
- spanning can be done either vertically or horizontally.
- Vertical spanning is called as row spanning and horizontal spanning is called as column spanning.

colspan example

```
<!DOCTYPE html>
<html>
<head><title>Table Example Demo</title>
</head>
<body>
<table border="1" height="200px" width="200px">
  <tr>
<th colspan="2">Student Details</th>
</tr>
  <tr>
<th>Roll No </th>
 <th>Name</th>
</tr>
  <tr>
<td>501</td>
<td>Ranjith</td>
</tr>
<tr>
<td>502</td>
<td>Surya</td>
</tr>
<tr>
<td>503</td>
<td>Vikash</td>
</tr>
<tr>
<td>504</td>
<td>Virat</td>
</tr>
<tr>
<td>505</td>
```

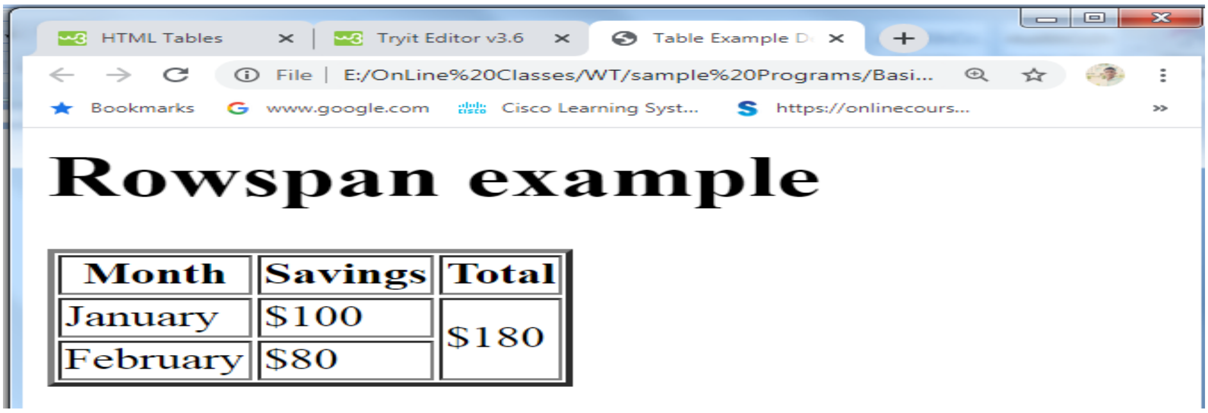
```
 <td>Rohit</td>
</tr>
</table>
</body>
</html>
```



Student Details	
Roll No	**Name**
501	Ranjith
502	Surya
503	Vikash
504	Virat
505	Rohit

Rowspan example

```
<!DOCTYPE html>
<html>
<head><title>Table Example Demo</title>
</head>
<body>
<h1>Rowspan example</h1>
 <table border="2">
  <tr>
    <th>Month</th>
    <th>Savings</th>
    <th>Total</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
    <td rowspan="2">$180</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$80</td>
  </tr>
</table>
</body>
</html>
```



Rowspan example

Month	Savings	Total
January	$100	$180
February	$80	

Frames:

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages −

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back* button might not work as the user hopes.

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Note − The <frame> tag deprecated in HTML5. Do not use this element.

The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag −

Sr.No	Attribute & Description
1	**cols** Specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of the four ways − Absolute values in pixels. For example, to create three vertical frames, use *cols = "100, 500, 100"*. A percentage of the browser window. For example, to create three vertical frames, use *cols = "10%, 80%, 10%"*. Using a wildcard symbol. For example, to create three vertical frames, use *cols = "10%, *, 10%"*. In this case wildcard takes remainder of the window. As relative widths of the browser window. For example, to create three vertical frames, use *cols = "3*, 2*, 1*"*. This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.
2	**rows**

	This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example, to create two horizontal frames, use *rows = "10%, 90%"*. You can specify the height of each row in the same way as explained above for columns.
3	**border** This attribute specifies the width of the border of each frame in pixels. For example, border = "5". A value of zero means no border.
4	**frameborder** This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example frameborder = "0" specifies no border.
5	**framespacing** This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing = "10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

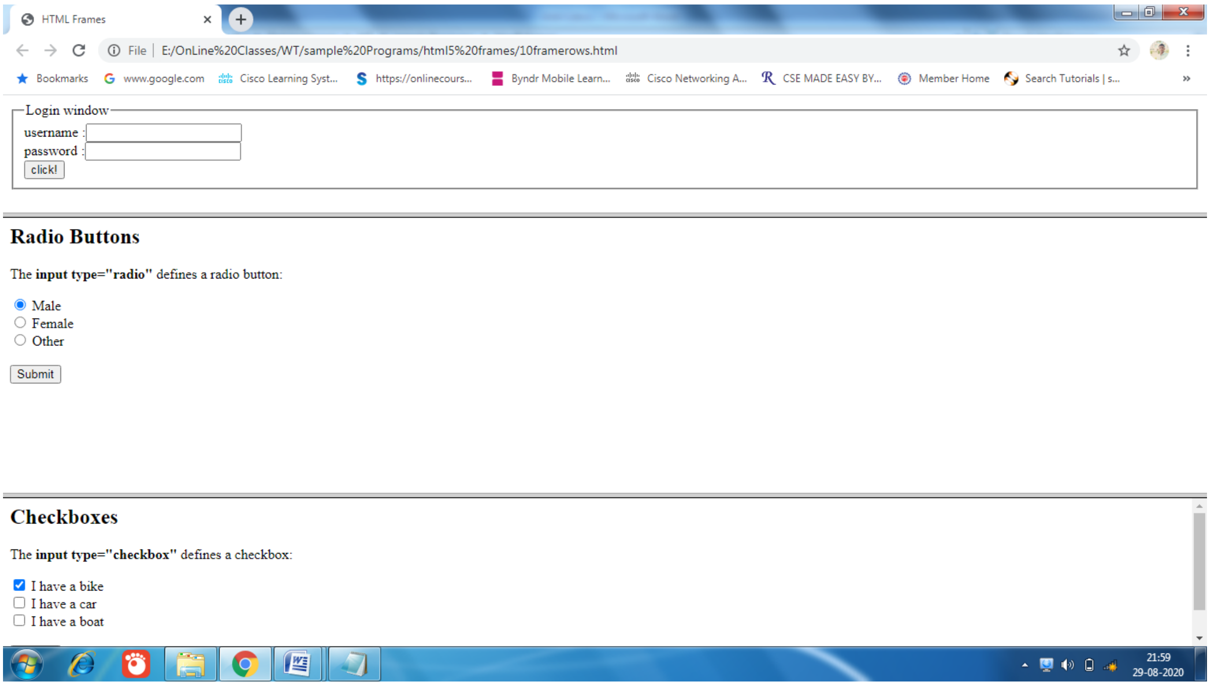
Following are the important attributes of <frame> tag −

Sr.No	Attribute & Description
1	**src** This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src = "/html/top_frame.htm" will load an HTML file available in html directory.
2	**name** This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into an another frame, in which case the second frame needs a name to identify itself as the target of the link.
3	**frameborder** This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).
4	**marginwidth**

	This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth = "10".
5	**marginheight** This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight = "10".
6	**noresize** By default, you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize = "noresize".
7	**scrolling** This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling = "no" means it should not have scroll bars.
8	**longdesc** This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc = "framedescription.htm"

Example to create three horizontal frames:

```
<!DOCTYPE html>
<html>
<head>
 <title>HTML Frames</title>
</head>
<frameset rows = "10%,80%,10%">
          <frame name = "top" src = "17 legend.html" />
          <frame name = "main" src = "18.radio.html" />
          <frame name = "bottom" src = "19 checkbox.html" />
               <noframes>

                    <body>Your browser does not support frames.
                    </body>
                     </noframes>
  </frameset>
 </html>
```



Login window

username :

password :

click!

Radio Buttons

The **input type="radio"** defines a radio button:

● Male
○ Female
○ Other

Submit

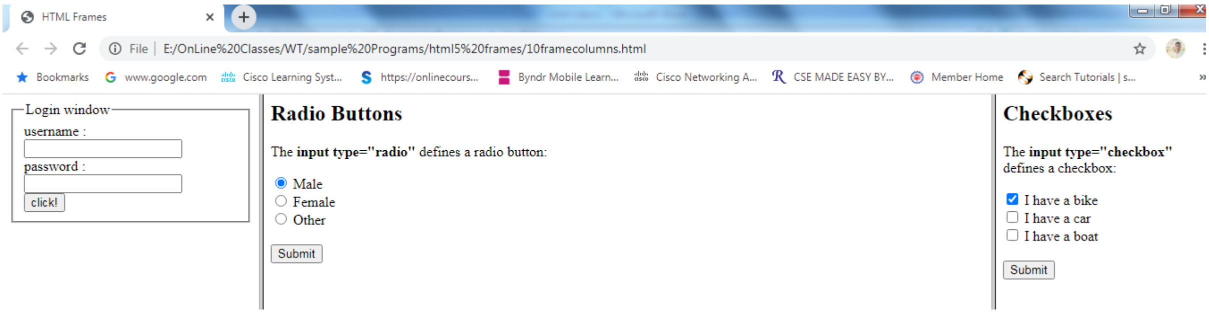
Checkboxes

The **input type="checkbox"** defines a checkbox:

☑ I have a bike
☐ I have a car
☐ I have a boat

Example to create three vertical frames:

```
<!DOCTYPE html>
<html>
<head>
      <title>HTML Frames</title>
   </head>
<frameset cols = "10%,80%,10%" border="5" frameborder = "1" framespacing="100">

            <frame name = "top" src = "17 legend.html"   scrolling = "no" />
            <frame name = "main" src = "18.radio.html" scrolling = "no" />
            <frame name = "bottom" src = "19 checkbox.html" scrolling = "no" />

  <noframes>
<body>Your browser does not support frames.
</body>
 </noframes>
   </frameset>
 </html>
```



Login window

username :

password :

click!

Radio Buttons

The **input type="radio"** defines a radio button:

● Male
○ Female
○ Other

Submit

Checkboxes

The **input type="checkbox"** defines a checkbox:

☑ I have a bike
☐ I have a car
☐ I have a boat

Submit

Forms:

An HTML form is used to collect user input. The user input is most often sent to a server for processing. The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

```
<form>
        …..
        form elements
        ……
</form>
```

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

The HTML <form> tag is used to create an HTML form and it has following syntax −

```
<form action = "Script URL" method = "GET|POST">
  form elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes −

Sr.No	Attribute & Description
1	**action** Backend script ready to process your passed data.
2	**method** Method to be used to upload data. The most frequently used are GET and POST methods.
3	**target** Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form .

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms –

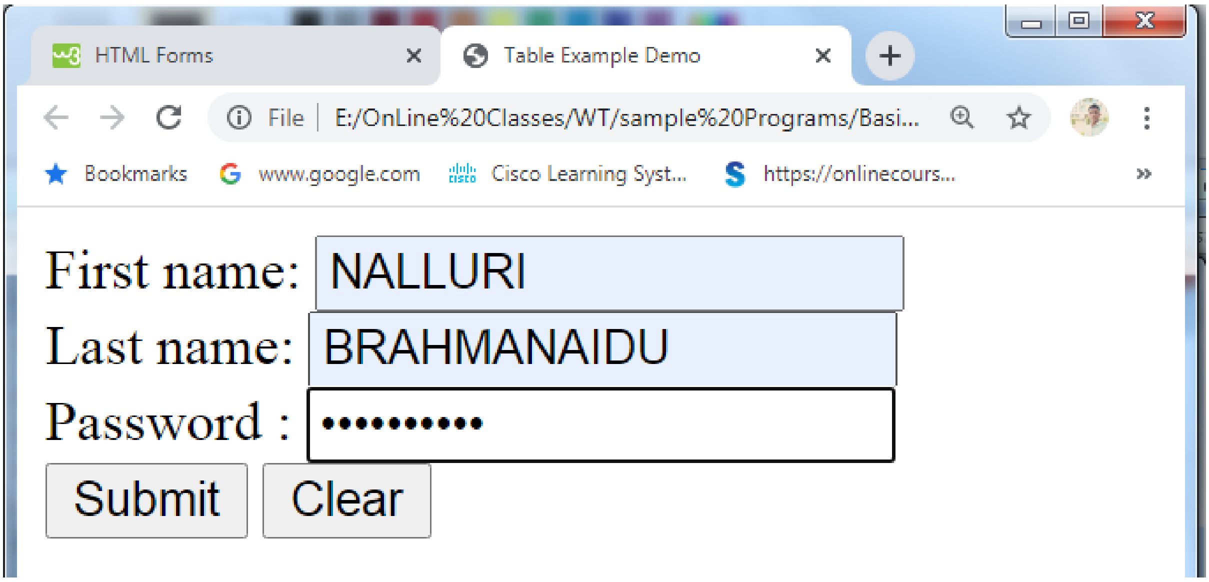
- **Single-line text input controls** – This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
- **Password input controls** – This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
- **Multi-line text input controls** – This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Here is a basic example of a single-line text input

```
<html>
<head><title>Table Example Demo</title>
</head>
<body>
<form  action="html_form_action.php" method="get">
First name: <input type="text" name="firstname"> <br>
Last name: <input type="text" name="lastname"> <br>
Password : <input type="password" name="pwd"><br>
<input type="submit" value="Submit">
 <input type="reset" value="Clear">
</form>
</body>
</html>
```



First name: NALLURI
Last name: BRAHMANAIDU
Password : ••••••••••

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to **password**.

Attributes

Following is the list of attributes for <input> tag for creating text field.

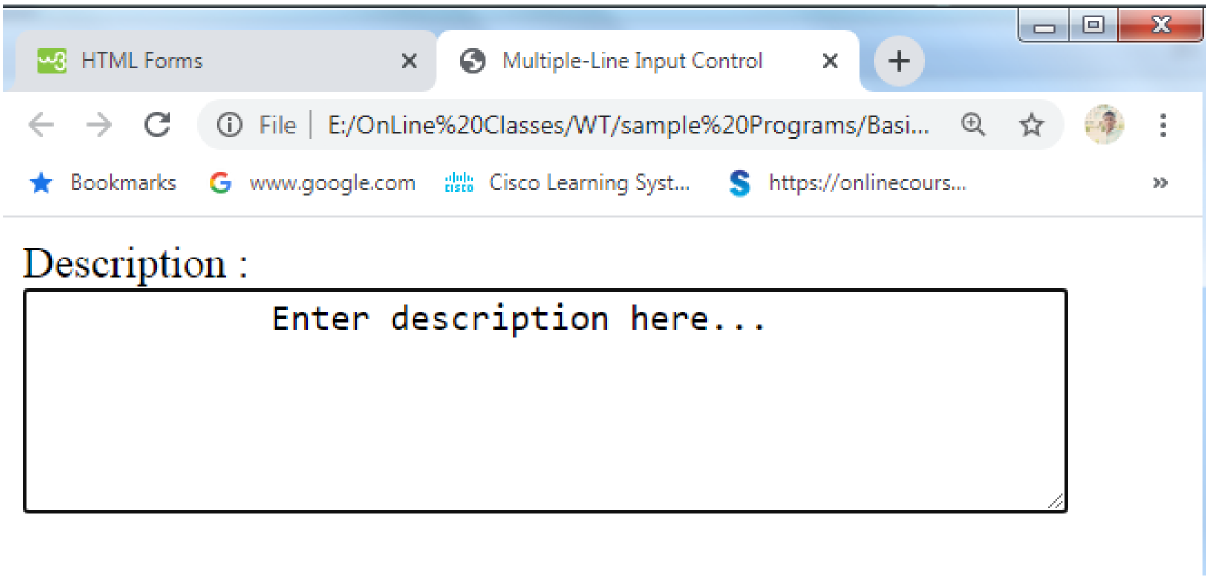
Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for text input control it will be set to **text**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Here is a basic example of a multi-line text input

```
<!DOCTYPE html>
<html>
  <head>
    <title>Multiple-Line Input Control</title>
  </head>

  <body>
    <form>
      Description : <br />
      <textarea rows = "5" cols = "50" name = "description">
        Enter description here...
      </textarea>
    </form>
  </body>

</html>
```


HTML Forms
Multiple-Line Input Control
File | E:/OnLine%20Classes/WT/sample%20Programs/Basi...
Bookmarks
www.google.com
Cisco Learning Syst...
https://onlinecours...

Description :

Enter description here...

Attributes

Following is the list of attributes for <textarea> tag.

Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**rows** Indicates the number of rows of text area box.
3	**cols** Indicates the number of columns of text area box

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox.**

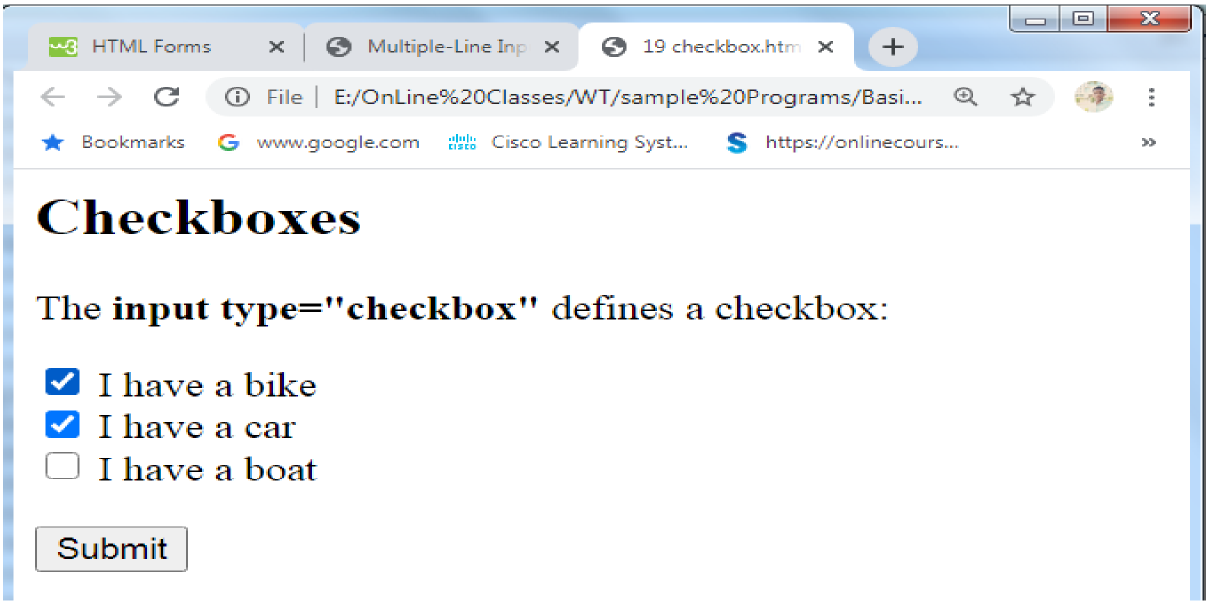
Attributes

Following is the list of attributes for <checkbox> tag.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set

	to **checkbox..**
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the checkbox is selected.
4	**checked** Set to *checked* if you want to select it by default.

```
<html>
<body>

<h2>Checkboxes</h2>
<p>The <strong>input type="checkbox"</strong> defines a checkbox:</p>

<form action="/action_page.php">
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike" checked>
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label><br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```



Checkboxes

The **input type="checkbox"** defines a checkbox:

☑ I have a bike
☑ I have a car
☐ I have a boat

Submit

Radio Button Control

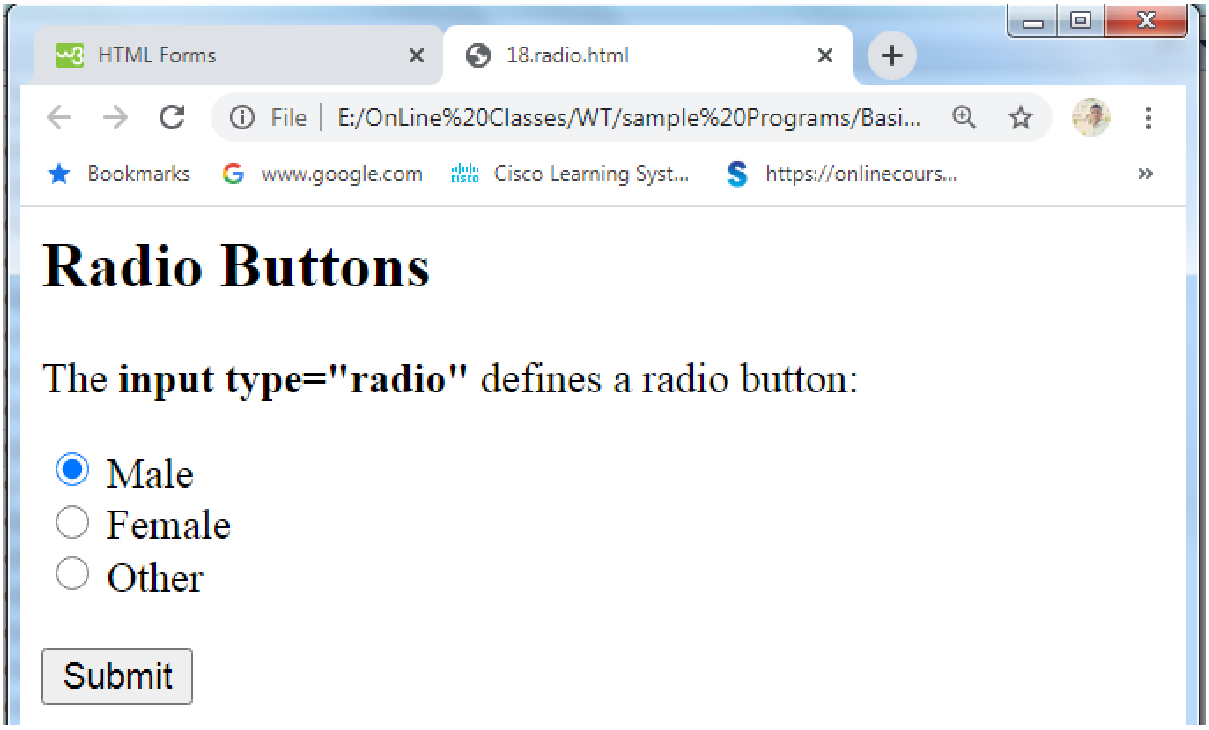
Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

Attributes

Following is the list of attributes for radio button.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set to radio.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the radio box is selected.
4	**checked** Set to *checked* if you want to select it by default.

```
<html>
<body>

<h2>Radio Buttons</h2>

<p>The <strong>input type="radio"</strong> defines a radio button:</p>

<form action="/action_page.php">
  <input type="radio" id="male" name="gender" value="male" checked>
  <label for="male">Male</label><br>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label><br>
  <input type="radio" id="other" name="gender" value="other">
  <label for="other">Other</label><br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```



Radio Buttons

The **input type="radio"** defines a radio button:

● Male
○ Female
○ Other

Submit

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

➢ **The <select> and <option> Elements:**

- The <select> element defines a drop-down list:
- The <option> elements defines an option that can be selected.
- By default, the first item in the drop-down list is selected.
- To define a pre-selected option, add the selected attribute to the option:

Attributes

Following is the list of important attributes of <select> tag −

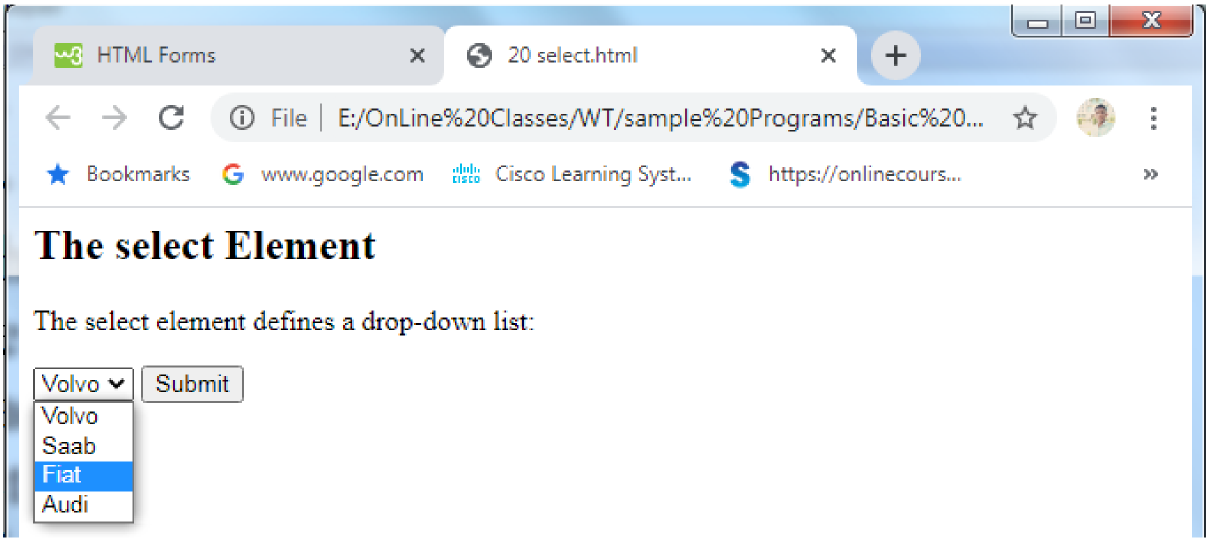
Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**size** This can be used to present a scrolling list box.
3	**multiple**

	If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag –

Sr.No	Attribute & Description
1	**value** The value that will be used if an option in the select box box is selected.
2	**selected** Specifies that this option should be the initially selected value when the page loads.
3	**label** An alternative way of labeling options

```
<html>
<body>
<h2>The select Element</h2>
<p>The select element defines a drop-down list:</p>
<form action="/action_page.jsp">
  <select name="cars" size="1" >
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat" >Fiat</option>
    <option value="audi" >Audi</option>
  </select>
  <input type="submit">
</form>

</body>
</html>
```



The select Element

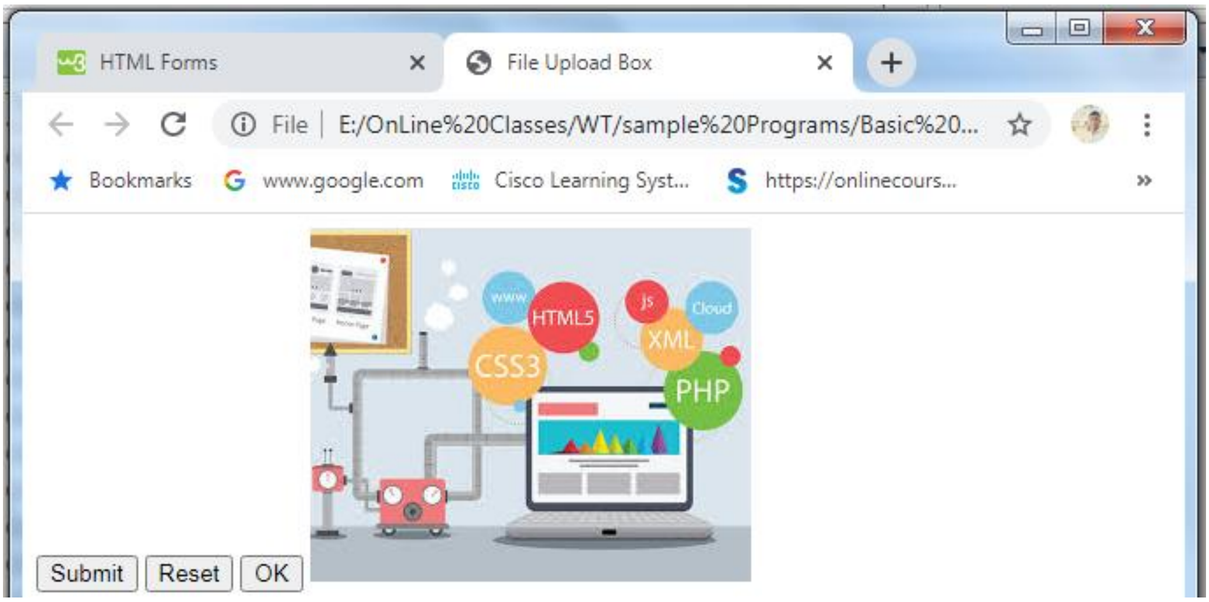
The select element defines a drop-down list:

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to **button**. The type attribute can take the following values −

Sr.No	Type & Description
1	**submit** This creates a button that automatically submits a form.
2	**reset** This creates a button that automatically resets form controls to their initial values.
3	**button** This creates a button that is used to trigger a client-side script when the user clicks that button.
4	**image** This creates a clickable button but we can use an image as background of the button.

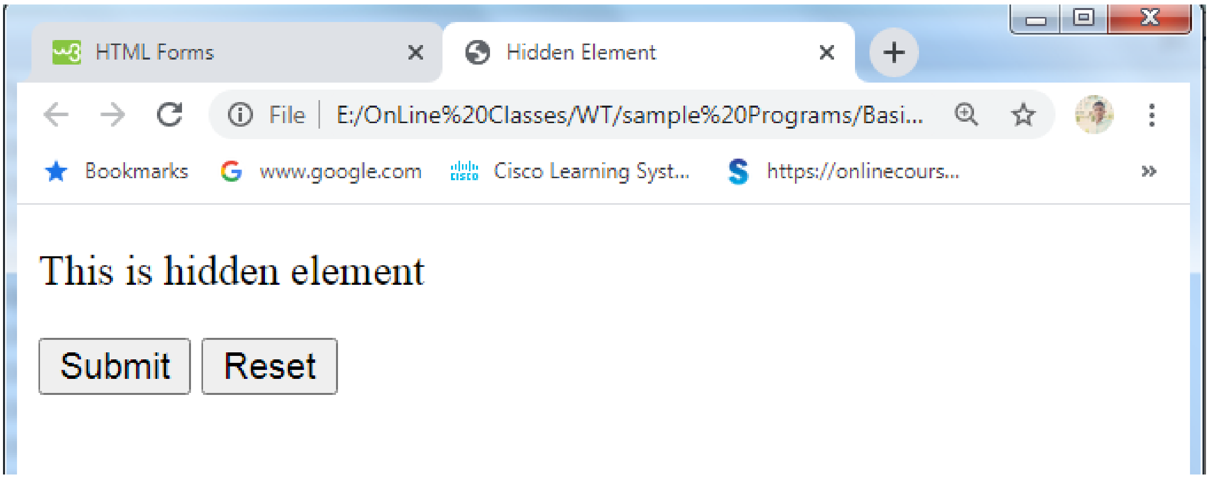
```
<!DOCTYPE html>
<html>

   <head>
      <title>Button Demo</title>
   </head>

   <body>
      <form>
         <input type = "submit" name = "submit" value = "Submit" />
         <input type = "reset" name = "reset"  value = "Reset" />
         <input type = "button" name = "ok" value = "OK" />
         <input type = "image" name = "imagebutton" src = "download.jpg" />
      </form>
   </body>

</html>
```



Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Hidden Element</title>
  </head>

  <body>
    <form>
      <p>This is hidden element</p>
      <input type = "hidden" name = "pagename" value = "10" />
      <input type = "submit" name = "submit" value = "Submit" />
      <input type = "reset" name = "reset"  value = "Reset" />
    </form>
  </body>

</html>
```

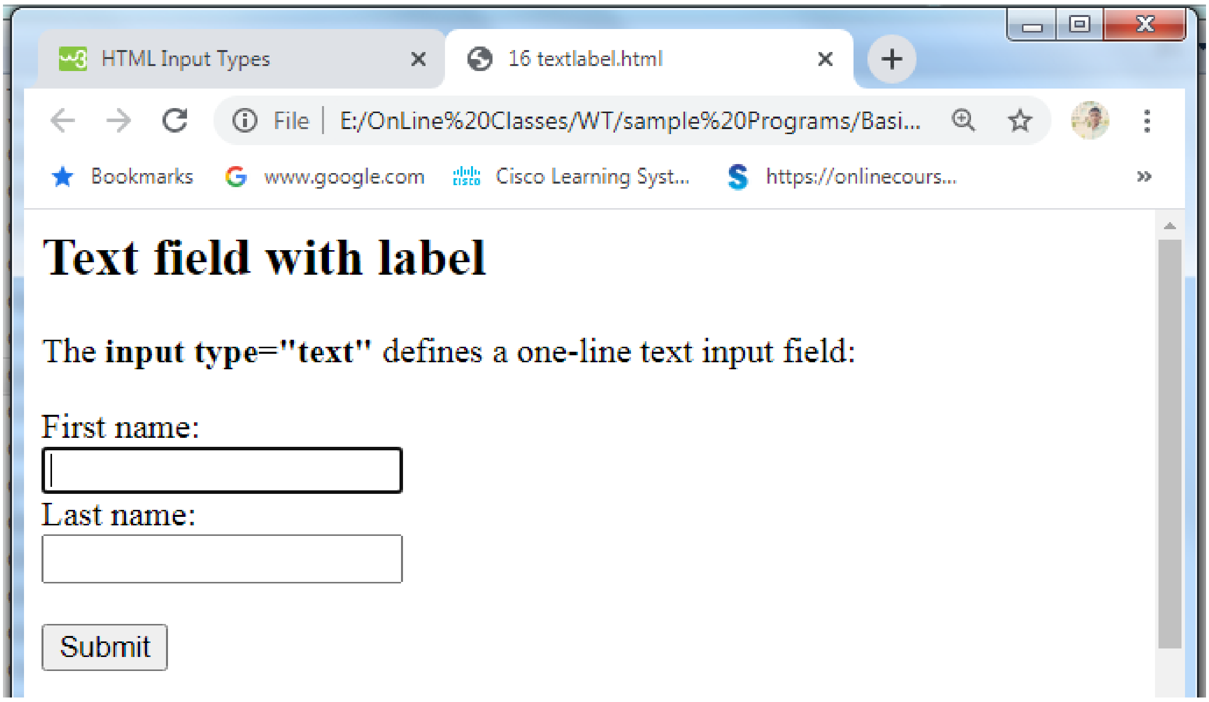


This is hidden element

The <label> Element

- The <label> element defines a label for several form elements.
- The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.
- The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.
- The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

```
<html>
<body>

<h2>Text field with label</h2>
<p>The <strong>input type="text"</strong> defines a one-line text input field:</p>

<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>

<p>Note that the form itself is not visible.</p>
<p>Also note that the default width of a text field is 20 characters.</p>

</body>
</html>
```



Text field with label

The **input type="text"** defines a one-line text input field:

First name:

Last name:

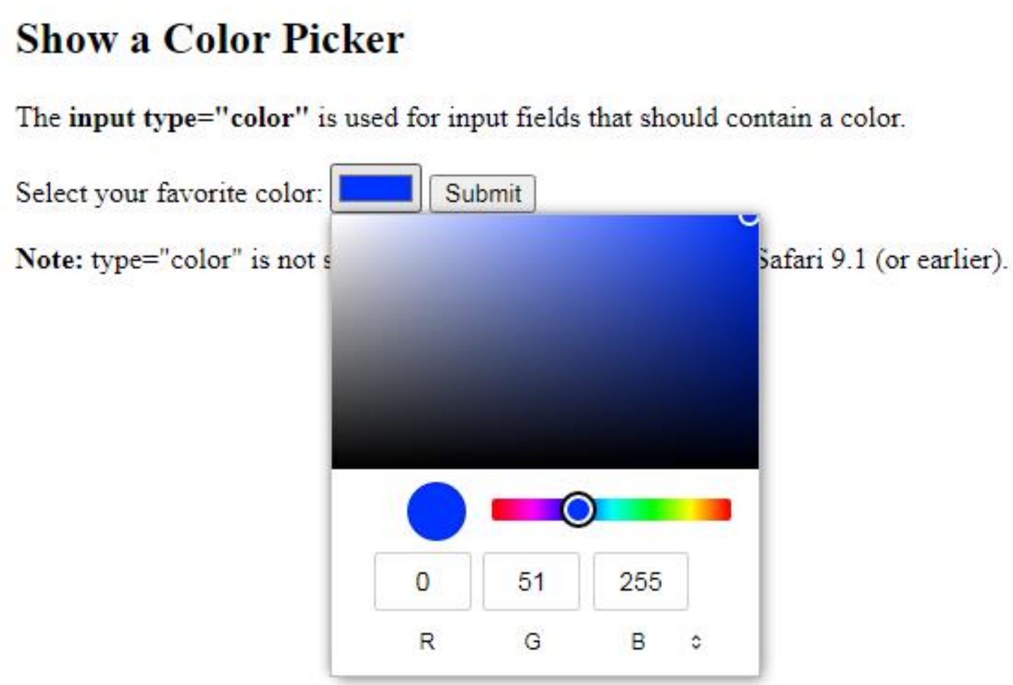
Submit

Input Type Color

The <input type="color"> is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

```
<!DOCTYPE html>
<html>
<body>
<h2>Show a Color Picker</h2>
<p>The <strong>input type="color"</strong> is used for input fields that should contain a color.</p>
<form action="/action_page.php">
  <label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor" value="#ff0000">
  <input type="submit" value="Submit">
</form>
</body>
</html>
```



Show a Color Picker

The **input type="color"** is used for input fields that should contain a color.

Select your favorite color: Submit

Note: type="color" is not s Safari 9.1 (or earlier).

Submitted Form Data

Your input was received as:

favcolor=#0033ff

The server has processed your input and returned this answer.

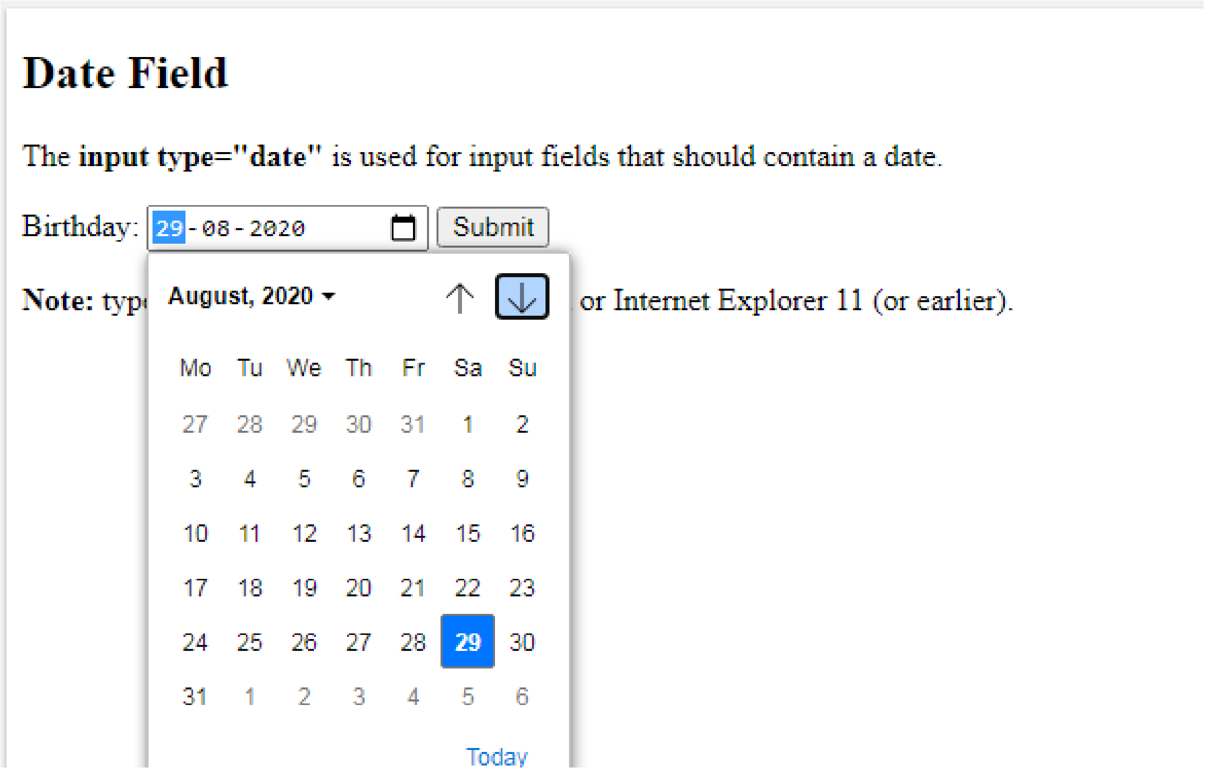
Input Type Date

The <input type="date"> is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

You can also use the `min` and `max` attributes to add restrictions to dates:

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

<h2>Date Field</h2>

<p>The <strong>input type="date"</strong> is used for input fields that should contain a
date.</p>

<form action="/action_page.php">
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
  <input type="submit" value="Submit">
</form>

<p><strong>Note:</strong> type="date" is not supported in Safari or Internet Explorer 11 (or
earlier).</p>

</body>
</html>
```



Date Field

The **input type="date"** is used for input fields that should contain a date.

Birthday: 29-08-2020

Note: typ or Internet Explorer 11 (or earlier).

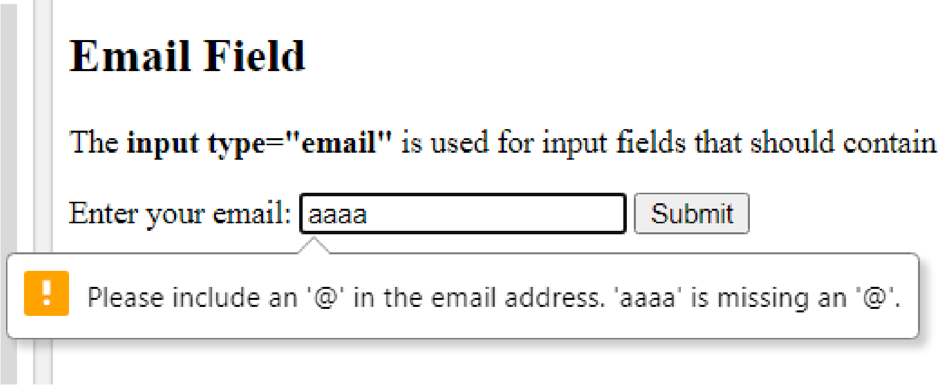
Submitted Form Data
Your input was received as:
birthday=2020-08-29

Input Type Email

The <input type="email"> is used for input fields that should contain an e-mail address. Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

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

<h2>Email Field</h2>

<p>The <strong>input type="email"</strong> is used for input fields that should contain an
e-mail address:</p>

<form action="/action_page.php">
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email">
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

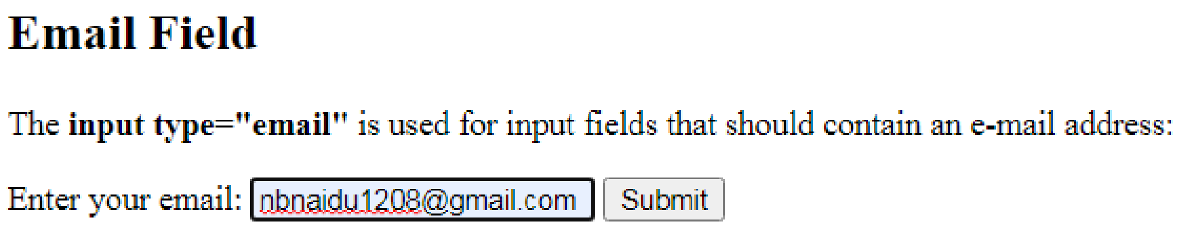


Email Field

The **input type="email"** is used for input fields that should contain an e-mail address:

Enter your email: aaaa

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



Email Field

The **input type="email"** is used for input fields that should contain an e-mail address:

Enter your email: nbnaidu1208@gmail.com

Submitted Form Data

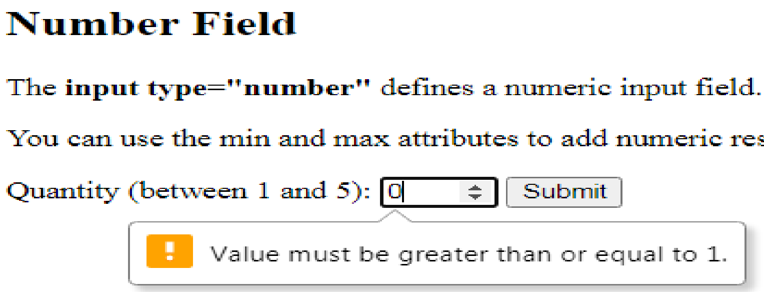
Your input was received as:

email=nbnaidu1208@gmail.com

Input Type Number

The <input type="number"> defines a numeric input field. You can also set restrictions on what numbers are accepted. The following example displays a numeric input field, where you can enter a value from 1 to 5.

```
<!DOCTYPE html>
<html>
<body>
<h2>Number Field</h2>
<p>The <strong>input type="number"</strong> defines a numeric input field.</p>
<p>You can use the min and max attributes to add numeric restrictions in the input field:</p>
<form action="/action_page.php">
  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
  <input type="submit" value="Submit">
</form>
</body>
</html>
```



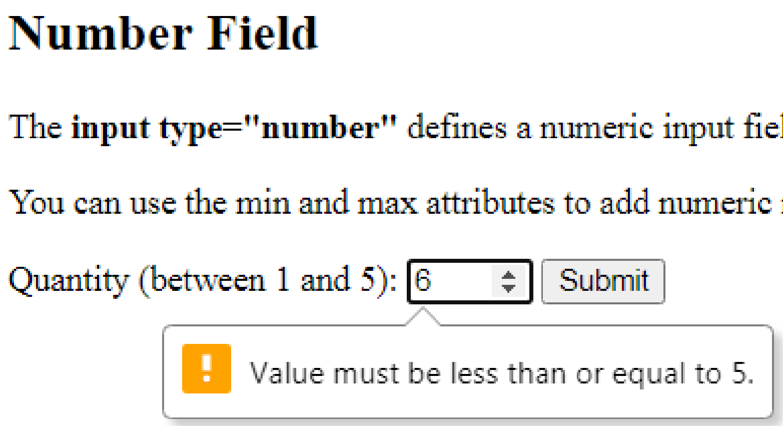
Number Field

The **input type="number"** defines a numeric input field.

You can use the min and max attributes to add numeric restrictions in the input field:

Quantity (between 1 and 5): 0

Value must be greater than or equal to 1.



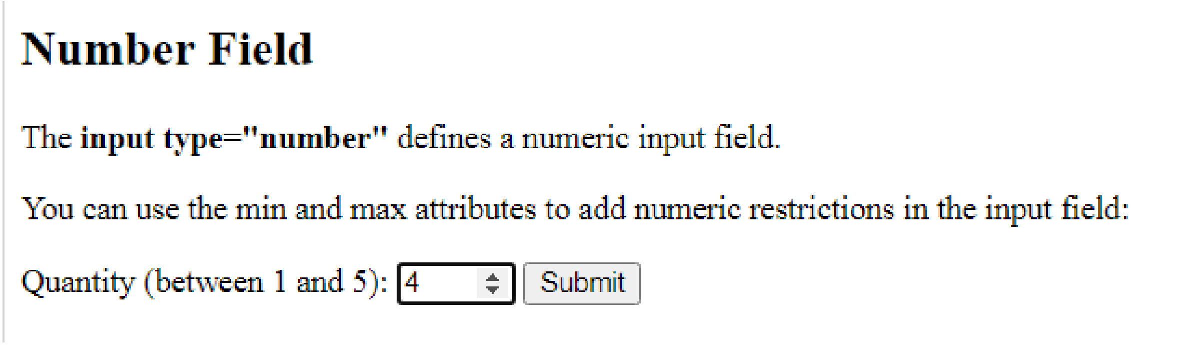
Number Field

The **input type="number"** defines a numeric input field.

You can use the min and max attributes to add numeric restrictions in the input field:

Quantity (between 1 and 5): 6

Value must be less than or equal to 5.



Number Field

The **input type="number"** defines a numeric input field.

You can use the min and max attributes to add numeric restrictions in the input field:

Quantity (between 1 and 5): 4

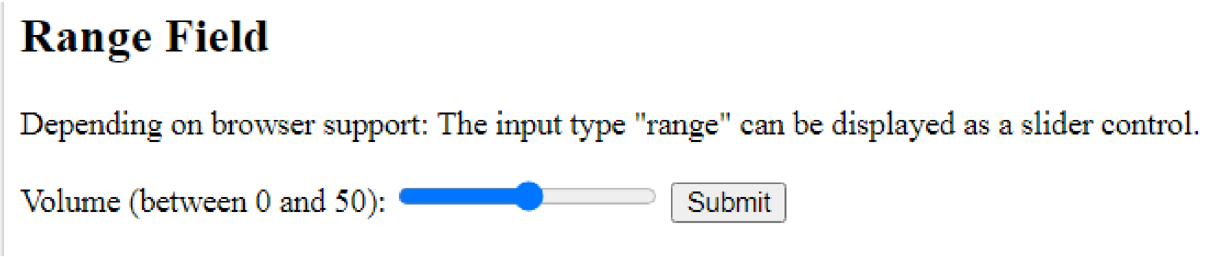
Submitted Form Data

Your input was received as: **quantity=4**

Input Type Range

The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the `min`, `max`, and `step` attributes:

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

<h2>Range Field</h2>

<p>Depending on browser support: The input type "range" can be displayed as a slider control.<p>

<form action="/action_page.php" method="get">
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
  <input type="submit" value="Submit">
</form>

</body>
</html>
```



Range Field

Depending on browser support: The input type "range" can be displayed as a slider control.

Volume (between 0 and 50):

Submitted Form Data

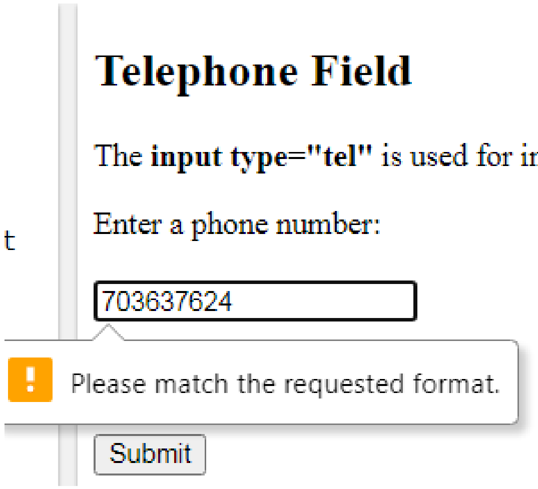
Your input was received as: vol=25 .

Input Type Tel

The <input type="tel"> is used for input fields that should contain a telephone number.

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

<h2>Telephone Field</h2>

<p>The <strong>input type="tel"</strong> is used for input fields that should contain a telephone number:</p>

<form action="/action_page.php">
  <label for="phone">Enter a phone number:</label><br><br>
  <input type="tel" id="phone" name="phone" placeholder="123-456-6789" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}" required><br><br>
  <small>Format: 123-456-6789</small><br><br>
  <input type="submit" value="Submit">
```

```
</form>

</body>
</html>
```



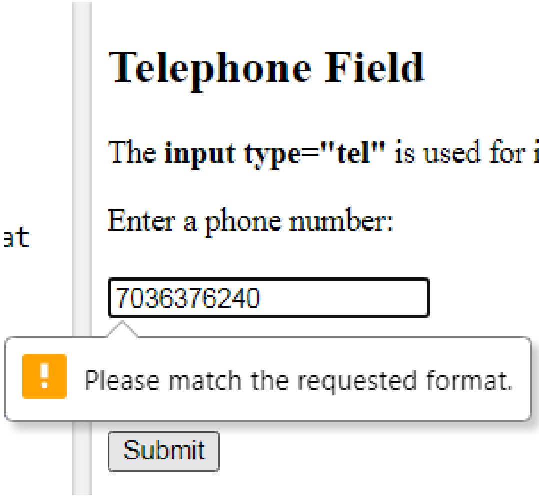
Telephone Field

The **input type="tel"** is used for input fields that should contain a telephone number:

Enter a phone number:

703637624

Please match the requested format.



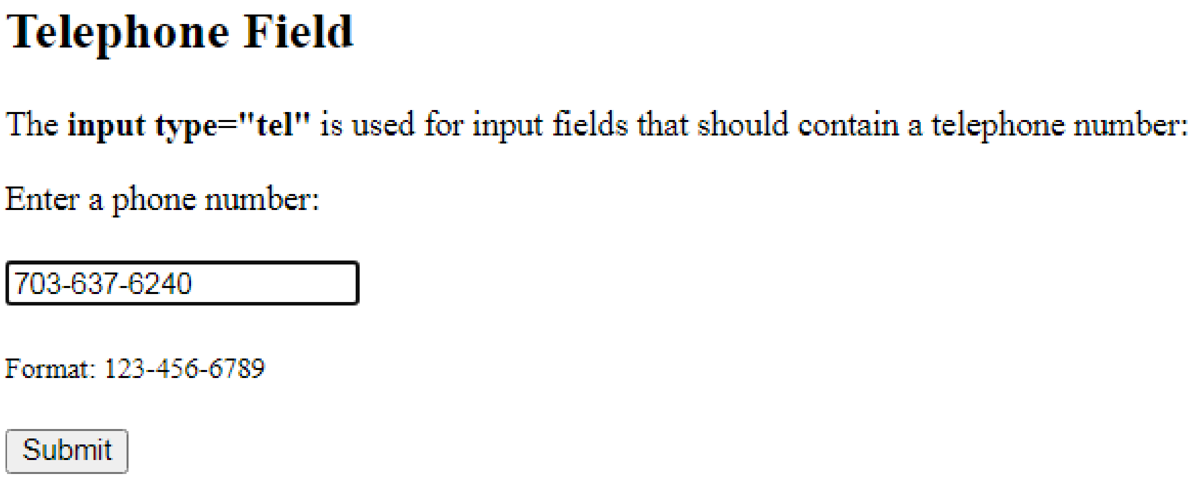
Telephone Field

The **input type="tel"** is used for input fields that should contain a telephone number:

Enter a phone number:

7036376240

Please match the requested format.



Telephone Field

The **input type="tel"** is used for input fields that should contain a telephone number:

Enter a phone number:

703-637-6240

Format: 123-456-6789

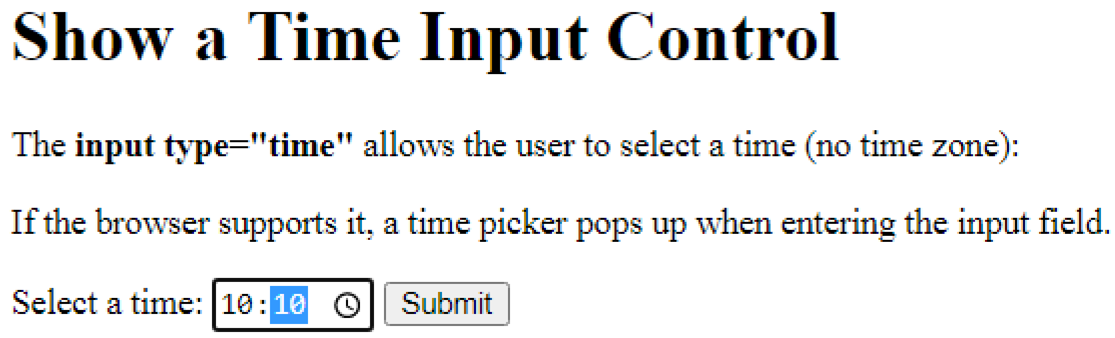
Submitted Form Data

Your input was received as: **phone=703-637-6240**

Input Type Time

The <input type="time"> allows the user to select a time (no time zone).
Depending on browser support, a time picker can show up in the input field.

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

<h1>Show a Time Input Control</h1>

<p>The <strong>input type="time"</strong> allows the user to select a time (no time
zone):</p>

<p>If the browser supports it, a time picker pops up when entering the input field.</p>

<form action="/action_page.php">
  <label for="appt">Select a time:</label>
  <input type="time" id="appt" name="appt">
  <input type="submit" value="Submit">
</form>

<p><strong>Note:</strong> type="time" is not supported in Safari or Internet Explorer 12
(or earlier).</p>

</body>
</html>
```



Show a Time Input Control

The **input type="time"** allows the user to select a time (no time zone):

If the browser supports it, a time picker pops up when entering the input field.

Select a time: 10:10 Submit

Submitted Form Data

Your input was received as:

appt=10:10

HTML5

HTML5 is the latest and most enhanced version of HTML. Technically, HTML is not a programming language, but rather a markup language.
HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 Page Structure

A basic HTML page starts with the Document Type Declaration or doctype. That is a way to inform the browser what type of document it is. The doctype is always the first item at the top of any HTML file.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.01//EN“http://www.w3.org/TR/html4/strict.dtd">

The HTML5 has done more! The best solution for now is this short declaration:
<!Doctype html>

The <html> Element

The <html> element follows the doctype information, which is used to inform the browser that this is an HTML document. You can use the lang attribute with the **"en"** value to specify that the document is in English.

```
<!DOCTYPE HTML>
<html lang="en">
</html>
```

The <head> section

The next part is the <head> section. The <head> element contains metadata (document title, character set, styles, links, scripts), specific information about the web page that is not displayed to the user.

```
<head>
<title>W3Docs - Learn Programming Languages Online.</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="Author" content="W3docs">
<link rel="stylesheet" type="text/css" href="style.css">
 </head>
```

The <body> element

The <body> of a document contains the content of the document. The content may be presented by a user agent in different ways. E.g., the content can be text, images, links, colors, graphics, etc.

```
<body>
...
</body>
```

Contains the main content of the document.
Placed after the closing tag of the head element.

```
<!DOCTYPE HTML>
        <HTML>
        <HEAD>
             <title>HTML5 Document</title>
       </HEAD>
             <BODY>
                  Content of the HTML document
             </BODY>
        </HTML>
```

HTML5 Semantic elements:

Semantic Elements: Semantic elements have meaningful names which tells about type of content. For example header, footer, table, … etc. HTML5 introduces many semantic elements as mentioned below which make the code easier to write and understand for the developer as well as instructs the browser on how to treat them.

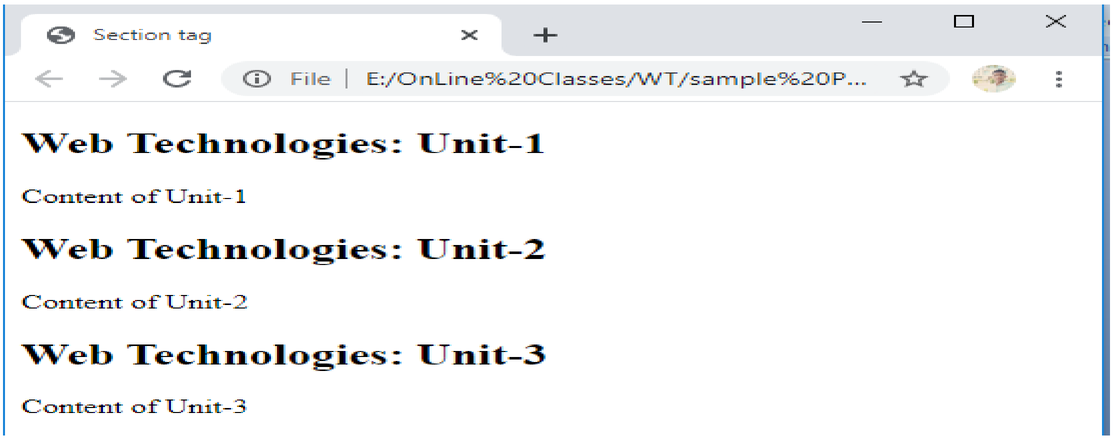
Section element:

Section tag defines the section of documents such as chapters, headers, footers or any other sections. The section tag divides the content into section and subsections. The section tag is used when requirements of two headers or footers or any other section of documents needed. Section tag grouped the generic block of related contents.

Syntax:

<section> Section Contents </section>Section tag is used to distribute the content i.e, it distribute the sections and subsections.

```
<!DOCTYPE html>
<html>
   <head>
      <title>Section tag</title>
   </head>
   <body>
      <section>
         <h1>Web Technologies: Unit-1</h1>
         <p>Content of Unit-1</p>
      </section>
      <section>
         <h1>Web Technologies: Unit-2</h1>
         <p>Content of Unit-2</p>
      </section>
      <section>
         <h1>Web Technologies: Unit-3</h1>
         <p>Content of Unit-3</p>
      </section>
   </body>
</html>
```



Web Technologies: Unit-1

Content of Unit-1

Web Technologies: Unit-2

Content of Unit-2

Web Technologies: Unit-3

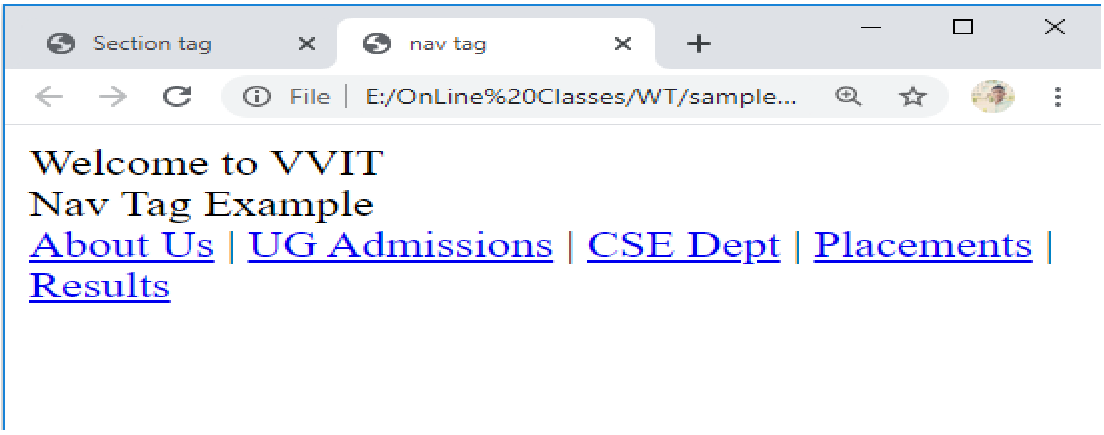
Content of Unit-3

Nav element:

The HTML <nav> element represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes.

```
<!DOCTYPE html>
<html>
  <head>
    <title>nav tag</title>
  </head>
  <body>
    <div class = "gfg">Welcome to VVIT</div>
    <div class = "nav_tag">Nav Tag Example</div>
    <nav>
```

```
      <a href = "https://www.vvitguntur.com/about/about-us">About Us</a> |
      <a href = "https://www.vvitguntur.com/adm/adm-ug">UG Admissions</a> |
      <a href = "https://www.vvitguntur.com/departments/cse-home">CSE Dept</a> |
      <a href = "https://www.vvitguntur.com/placements-home">Placements</a> |
      <a href = "https://www.vvitguntur.com/aca-examcell/latest-results">Results</a>
    </nav>
  </body>
</html>
```

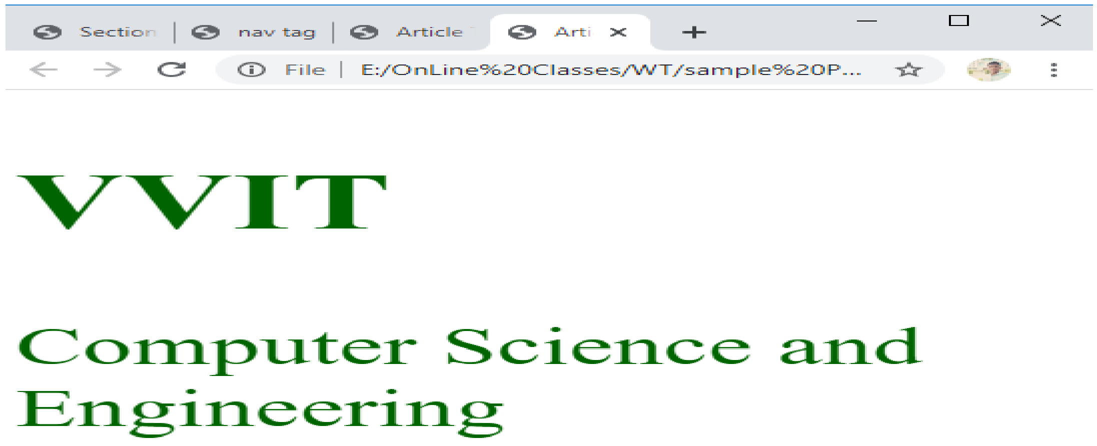


Welcome to VVIT
Nav Tag Example
About Us | UG Admissions | CSE Dept | Placements |
Results

Article Element:

The **HTML <article> element** represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication). Examples include: a forum post, a magazine or newspaper article, or a blog entry.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Article Tag</title>
    <style>
      article {
      Color:#006400;
      font-size:50px;
      Text-align:left;
      }

    </style>
  </head>
  <body>
    <article>
      <h1>VVIT</h1>
      <p> Computer Science and Engineering</p>
    </article>
  </body>
</html>
```



VVIT

Computer Science and Engineering

Aside Element:

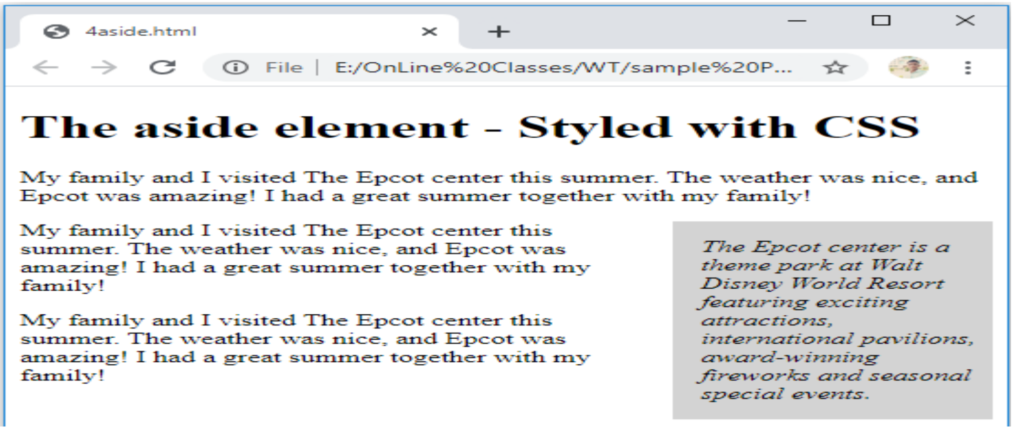
Create a section used to display information about the content of other elements. Such as time and date, current news and weather report.

The <aside> tag is used to describe the main object of the web page in a shorter way like a highlighter. It basically identifies the content that is related to the primary content of the web page **but does not constitute the main intent** of the primary page. The <aside> tag contains mainly author information, links, related content and so on.

```
<aside> <h3>Contents...</h3> <p>Contents...</p> </aside>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
aside {
  width: 30%;
  padding-left: 15px;
  margin-left: 15px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
</style>
</head>
<body>

<h1>The aside element - Styled with CSS</h1>

<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot
was amazing! I had a great summer together with my family!</p>

<aside>
<p>The Epcot center is a theme park at Walt Disney World Resort featuring exciting
attractions, international pavilions, award-winning fireworks and seasonal special events.</p>
</aside>

<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot
was amazing! I had a great summer together with my family!</p>
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot
was amazing! I had a great summer together with my family!</p>
```

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



The aside element – Styled with CSS

My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!

My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!

My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!

The Epcot center is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.

Header Element:

The <header> tag in HTML is used to define the header for a document or a section.

The header tag contains information related to the title and heading of the related content.

The <header> element is intended to usually contain the section’s heading (an h1-h6 element or an <hgroup> element), but this is not required.

The <header> element can also be used to wrap a section’s table of contents, a search form, or any relevant logos.

The <header> tag is a new tag in HTML5 and it requires a starting tag as well as an end tag.

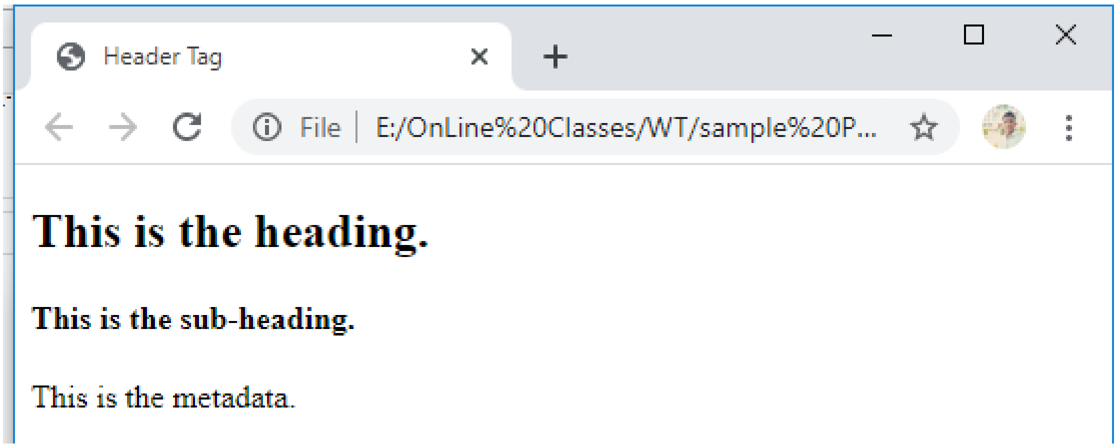
There can be several <header> elements in one document.

A <header> tag cannot be placed within a <footer>, <address> or another <header> element.

Syntax:

<header> ...</header>

```
<!DOCTYPE html>
<html>
   <head>
      <title>Header Tag</title>
   </head>
   <body>
      <article>
         <header>
            <h1>This is the heading.</h1>
            <h4>This is the sub-heading.</h4>
            <p>This is the metadata.</p>
         </header>
      </article>
   </body>
</html>
```



This is the heading.

This is the sub-heading.

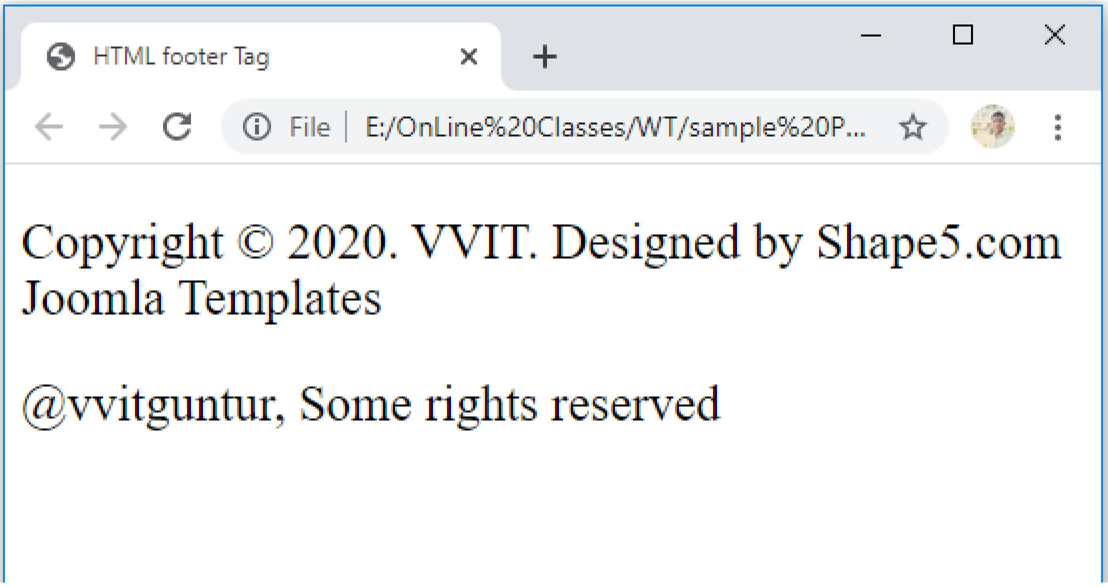
This is the metadata.

Footer Element:

The <footer> tag in HTML is used to define a footer of HTML document. This section contains the footer information (author information, copyright information, carriers, etc). The footer tag is used within the body tag. The <footer> tag is new in the HTML5. The footer elements require a start tag as well as an end tag.

Syntax :
<footer> ... </footer>A footer element typically contains authorship information, copyright information, contact information, sitemap, back to top links, related documents, etc.

```
<!DOCTYPE html>
<html>
   <head>
      <title>HTML footer Tag</title>
      <style>
         a {
            font-size:25px;
            text-decoration:none;
         }
         p {
            font-size:25px;
         }
      </style>
   </head>
   <body>
      <footer>

            <p>
               Copyright © 2020. VVIT. Designed by Shape5.com Joomla Templates
            </p>

         <p>@vvitguntur, Some rights reserved</p>
      </footer>
   </body>
</html>
```



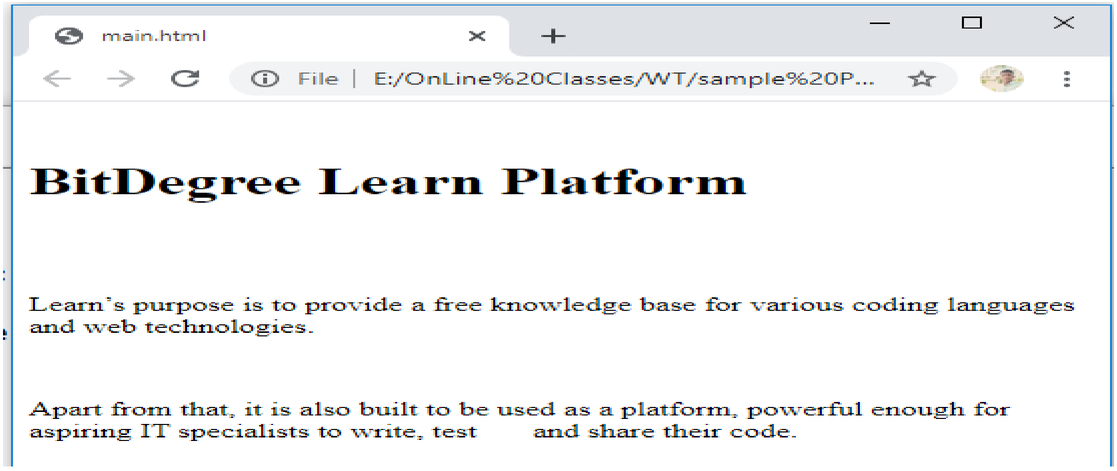
Copyright © 2020. VVIT. Designed by Shape5.com Joomla Templates

@vvitguntur, Some rights reserved

HTML <main> Tag

- The <main> element is used to denote the content of a webpage that relates to the central topic of that page or application. It should include content that is unique to that page and should not include content that is duplicated across multiple webpages, such as headers, footers, and primary navigation elements.
- The <main> section is also useful when non-standard devices, such as screen readers, interpret the content of your page. It allows the screen reading device to detect and find the important content rapidly, while it skips over the code at the top of the file.
- The <main> element can only be used once in each HTML file. It is followed by a closing tag, </main>, which should be placed at the end of the content.
- <main> “represents the main content of the body of a document or application”, while <body> “represents the content of the document”.

```
 <!DOCTYPE html>
<html>

<head>
  <script type="text/javascript" src="scripts.js">
  </script>
  <link rel="stylesheet" href="styles.css">
</head>

<body>

  <h1>BitDegree Learn Platform</h1>

  <main role="main">
    <p>LEARN’s purpose is to provide a free knowledge base for various coding languages
and web technologies.</p>

    <p>Apart from that, it is also built to be used as a platform, powerful enough for aspiring
IT specialists to write, test
      and share their code.</p>
  </main>

</body>

</html>
```



BitDegree Learn Platform

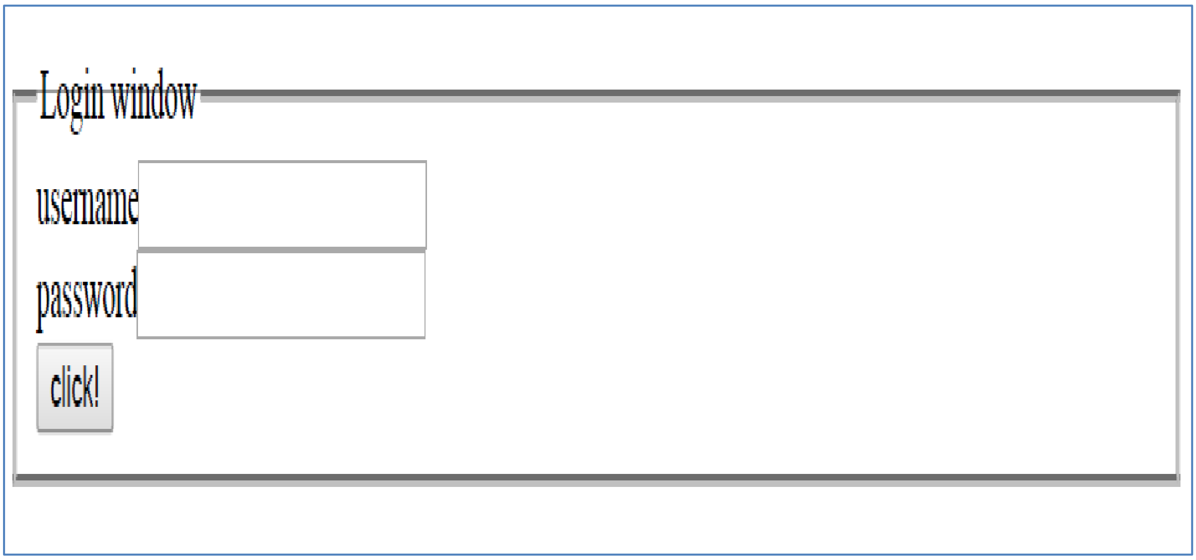
Learn’s purpose is to provide a free knowledge base for various coding languages and web technologies.

Apart from that, it is also built to be used as a platform, powerful enough for aspiring IT specialists to write, test and share their code.

- **FIELDSET and LEGEND Elements:**

- Fieldset element is used to group related controls in a single box and the legend element provide the caption for that box. Example is shown below.

```
<!DOCTYPE html>
<html>
<body>
<form>
<fieldset>
<legend>Login window</legend>
<label>
username<input type="text"/>
</label>
<br/>
<label>
password<input type="password"/>
</label>
<br/>
<label>
<input type="submit" name="submit" value="click!">
</label>
</fieldset>
</form>
</body>
</html>
```



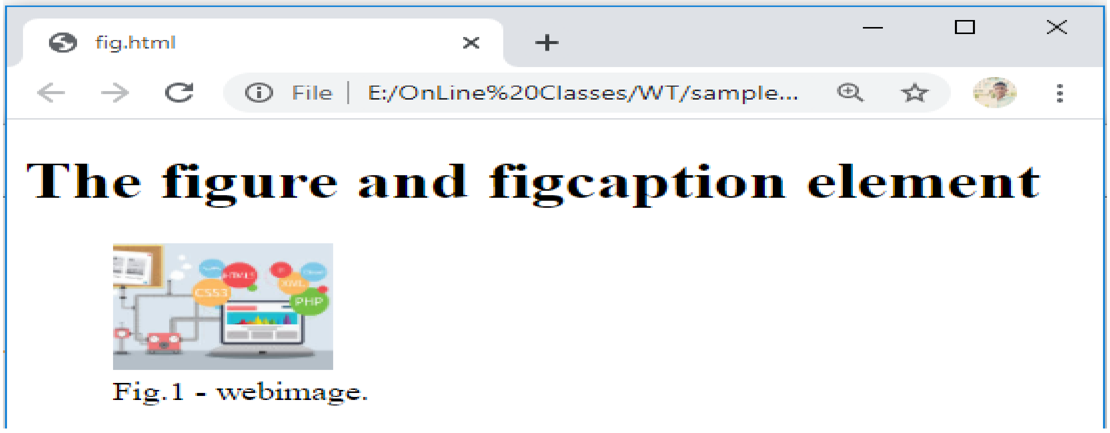
HTML <figure> Tag

Use a <figure> element to mark up a photo in a document, and a <figcaption> element to define a caption for the photo. The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

While the content of the <figure> element is related to the main flow, its position is independent of the main flow, and if removed it should not affect the flow of the document. The <figcaption> element is used to add a caption for the <figure> element.

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

```
<h1>The figure and figcaption element</h1>

<figure>
  <img src="download.jpg" alt="webimage" style="width:100%">
  <figcaption>Fig.1 - webimage.</figcaption>
</figure>

</body>
</html>
```



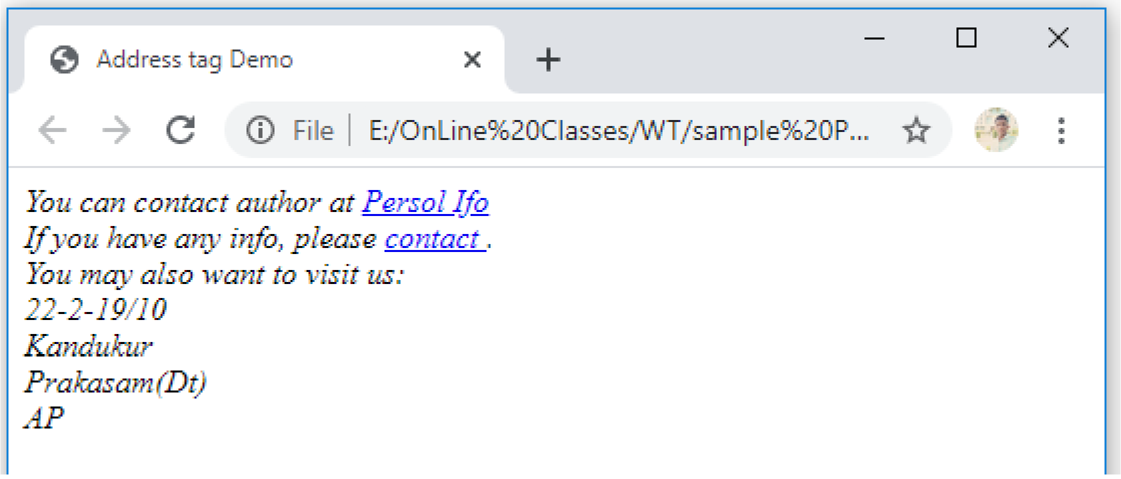
The figure and figcaption element

Fig.1 - webimage.

Address Element:

The **HTML <address> element** indicates that the enclosed HTML provides contact information for a person or people, or for an organization. Display the content information for a document or part of document.
The <address> element should include the name of the person, people, or organization to which the contact information refers.
Content of this element appears in italics of the web- browser.
<address> can be used in a variety of contexts, such as providing a business's contact information in the page header, or indicating the author of an article by including an <address> element within the <article>.

```
<!DOCTYPE html>
<html>
<head>
<title> Address tag Demo </title>
</head>
<body>
 <address>
   You can contact author at <a href="https://www.vvitguntur.com/mr-n-brahmanaidu-m-tech-asst-professor">
   Persol Ifo</a><br>
   If you have any info, please <a href="mailto:nbnaidu1208@gmail.com">
   contact </a>.<br>
   You may also want to visit us:<br>
   22-2-19/10<br>
   Kandukur<br>
   Prakasam(Dt)<br>
   AP
  </address>
<body>
</html>
```

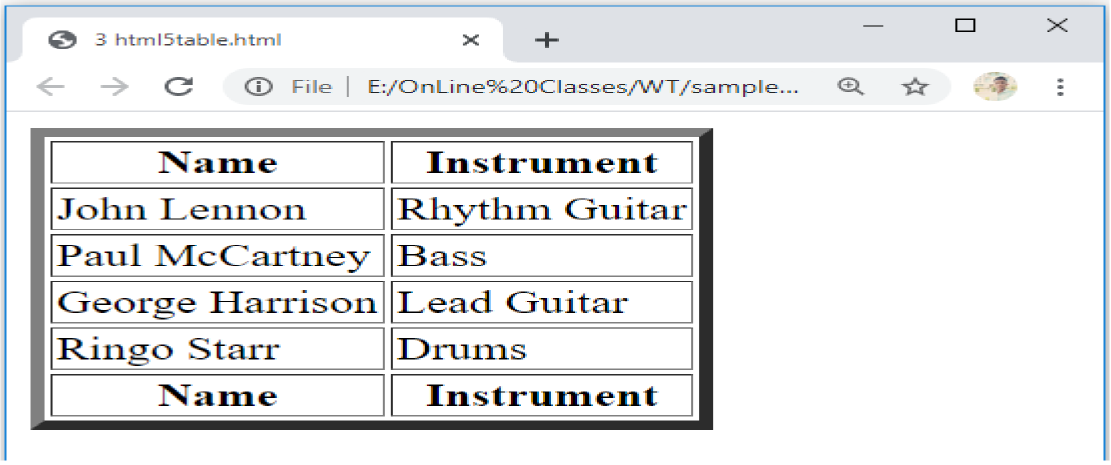


You can contact author at [Persol Ifo]
If you have any info, please [contact] .
You may also want to visit us:
22-2-19/10
Kandukur
Prakasam(Dt)
AP

HTML5 <table> Elements:

- Tbody- it is used to group the rows of a table and is use in conjunction with thead and tfoot elements.
- Thead-used to define the header for the table.
- Tfoot- used to define the footer for the table.
- Tr-defines row of a table.
- Td-defines cell of a table.
- Th-define a table header.

```
<!DOCTYPE html>
<html>
<body>
<table border="5">
  <thead>
    <tr>
      <th>Name</th>
      <th>Instrument</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th>Name</th>
      <th>Instrument</th>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>John Lennon</td>
      <td>Rhythm Guitar</td>
    </tr>
    <tr>
      <td>Paul McCartney</td>
      <td>Bass</td>
    </tr>
    <tr>
      <td>George Harrison</td>
      <td>Lead Guitar</td>
    </tr>
    <tr>
      <td>Ringo Starr</td>
      <td>Drums</td>
    </tr>
  </tbody>
</table>
</body>
</html>
```



Name	Instrument
John Lennon	Rhythm Guitar
Paul McCartney	Bass
George Harrison	Lead Guitar
Ringo Starr	Drums
Name	**Instrument**

Cascading Style Sheets (CSS) :

- Cascading Style Sheets (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.
- Cascading style sheet (CSS) is a text file with .CSS extension and is commonly used to define styles and layouts of webpages written in HTML and extensible Hypertext markup language(X HTML).
- CSS simplifies the task of maintaining a web document by separating its styles information such as font-size, font-color and line-width and background color.

This separation allows us to

- Apply the same style rules to multiple web pages.
- Apply style multiple times in a single website.

Evolution of CSS:

- CSS was introduced in late 1996 on the recommendation of World Wide Web consortium (W3c).
- The CSS level 1 (CSS 1) was published in Dec-1996.
- CSS level 2 (CSS 2) was published on nov-4th, 1997.
- CSS 3 development was started in the year 2005 and still is under development.

CSS: Style Specification Formats

- CSS uses syntax to apply CSS rules in an HTML document.
- CSS syntax is divided into two different parts:

Selector and Declaration

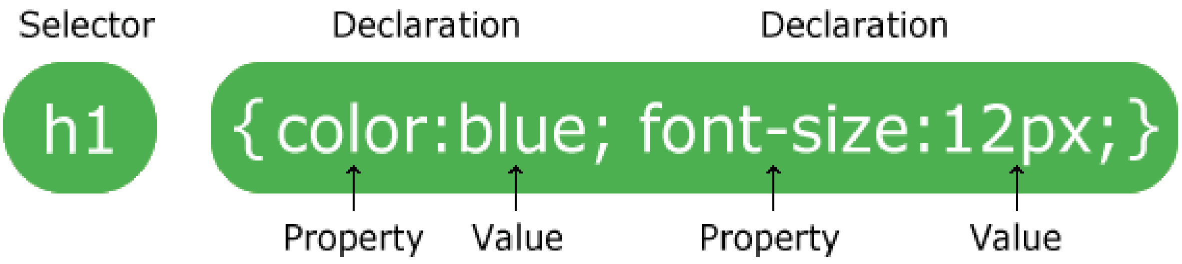
- Selector defines an HTML element to which the style is applied.
- Declaration contains the CSS properties as well as the value of these properties.

```
Selector {
    1st property : value;
    2nd property: value;
    3rd property: value;
    4th property : value;
```

```
                    :
                    :
                    :
          nth property : value;
                  }
```

CSS Syntax:

- A CSS rule-set consists of a selector and a declaration block:
- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.



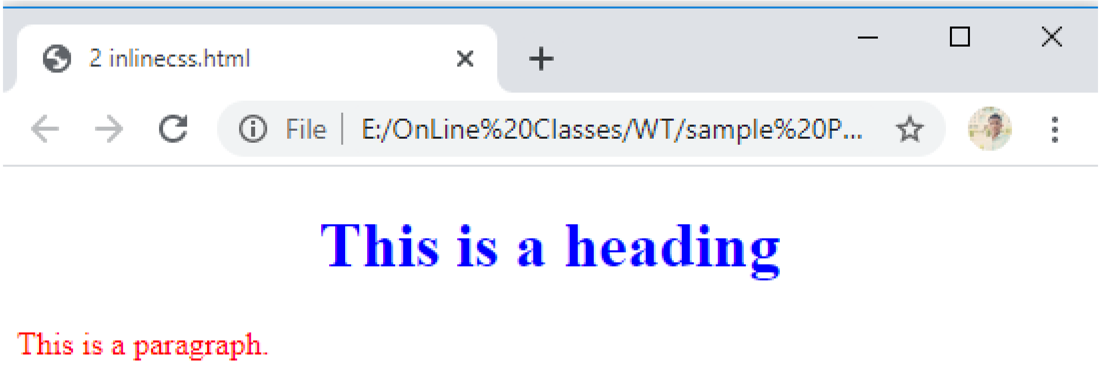
Levels of Style Sheets:

- A CSS style sheet can be linked to an HTML document in a variety of ways.
- 3 ways to apply CSS style to the HTML document:
  - The Inline style
  - The Internal style sheet
  - The External style sheet

- **Inline Style:**
- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

- The inline style properties are written in a simple line separated by semicolons.
- These properties are placed inside the style attribute of the HTML element, on which you want to apply the style.

```
          < p style = “ background : # cccccc;
                 color : # ffffff;
                 border : solid black 1px; ” >
                   Text in the paragraph</p>
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```



This is a heading

This is a paragraph.

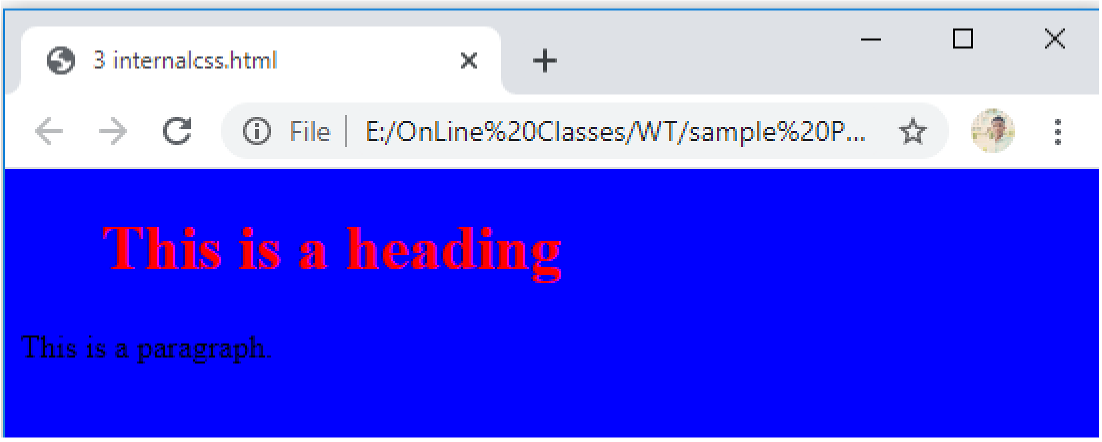
Advantages of using inline style sheets:

- Provides highest precedence over internal and external style sheets. Therefore, if you want some styles to be compulsorily applied, use the inline style.
- Provides an easy and quick approach to add a style sheet in a web page.

Internal style sheet:

- Written with in the Head element of the HTML document.
- This style is applied only to the document in which it is defined and not referenced by any other web document.
- Usage:
  ```
  <style type = “text/CSS”>
  selector { property : value; }
  </style>
  ```
- Advantages of using internal style sheets:
  - Affects only the page in which they are placed.
  - Allows to change the style of the same HTML file in which you are working.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: blue;
}

h1 {
  color: red;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



This is a heading

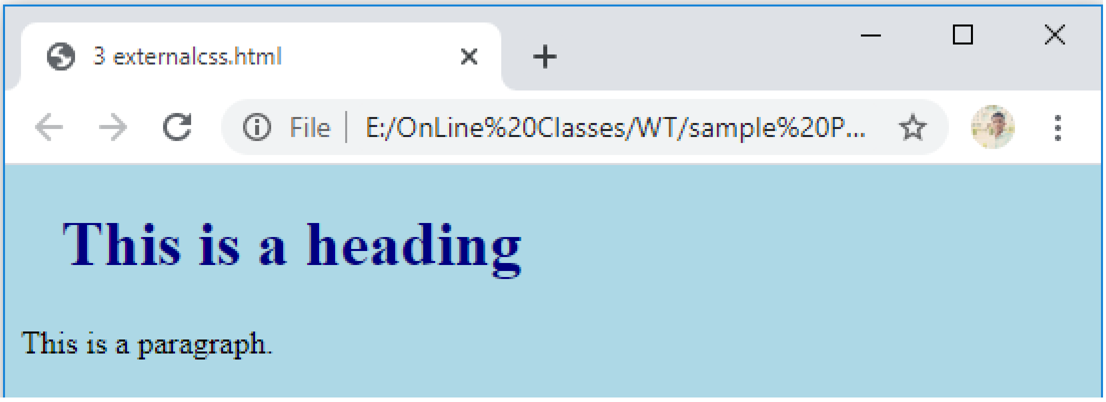
This is a paragraph.

External style sheet:

- The CSS file is written outside the HTML document.
- Reference of the file is placed in the HTML document.
- In an external style sheet, the style sheet rules are saved into a text file with the .CSS extension.

firstcss.css

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

h1 {
  color: navy;
  margin-left: 20px;
}

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="firstcss.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



This is a heading

This is a paragraph.

- We can link external style sheet with the web page using Linking:
- Refers to the HTML LINK element which is used to link a style sheet.
- This element has 3 attributes:
  - rel
  - type
  - href
- The rel attribute specifies what you are linking (style sheet).
- The type specifies the MIME type for the browser.
- The href specifies the path of the .CSS file.

  EX: < Link rel = " stylesheet " type = " text/CSS " href = "firstcss.css " / >

➢ Importing:

- Importing helps us in accessing the style rules from other CSS style sheets.
- The @ import keyword is used followed by the uniform resource identifier (URI)of the style sheet to which you want to import the style rules.

Usage:

```
<style type="text/css">
@import url("firstcss.css")
h1{color:blue}
</style>
```

Advantages of using external style sheets:

- Allows us to control the look and feel of several documents in one go and do not need to define a specific style for every element.
- Allows us to easily group our styles in a more efficient way.

Disadvantages of using external style sheets:

- Increases the download time as the entire css file has to be loaded to apply style to the HTML document.
- Displays the web page only after the entire style sheet is loaded

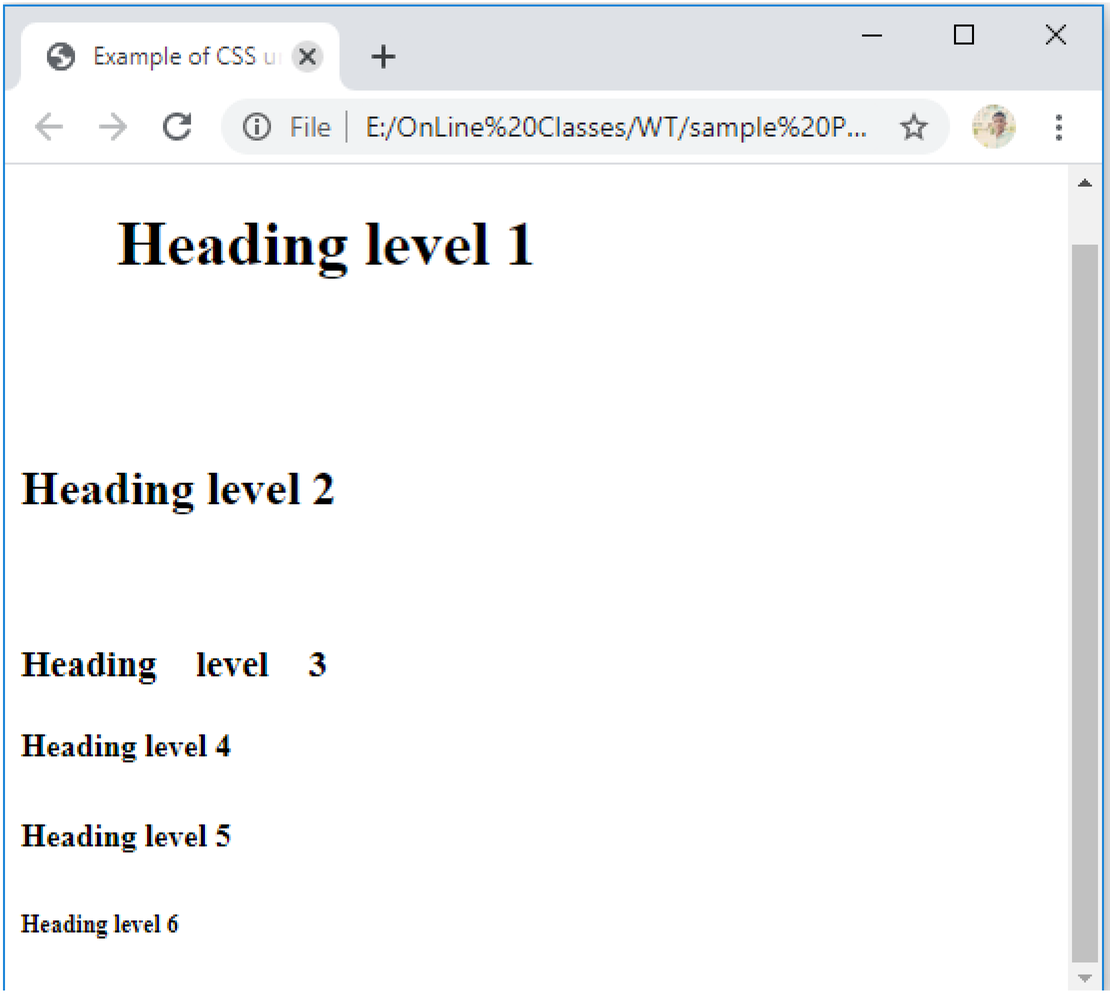
CSS Units:

- CSS has several different units for expressing a length.
- Many CSS properties take "length" values, such as width, margin, padding, font-size, etc.
- Length is a number followed by a length unit, such as 10px, 2em, etc.
- A whitespace cannot appear between the number and the unit. However, if the value is 0, the unit can be omitted.
- For some CSS properties, negative lengths are allowed.
- There are two types of length units: absolute and relative.

Absolute Lengths:

- Absolute length units are fixed in relation to each other. They are highly dependent on the output medium, so are mainly useful when the output environment is known.
- The absolute units consist of the physical units (in, cm, mm, pt, pc) and the px unit.
- Absolute physical units such as in, cm, mm, etc. should be used for print media and similar high-resolution devices.
- Whereas, for on-screen display such as desktop and lower-resolution devices, it is recommended to use the pixel or em units.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example of CSS units</title>
  <style>
    h1 { margin: 0.5in; }       /* inches */
    h2 { line-height: 3cm; }     /* centimeters */
    h3 { word-spacing: 4mm; }    /* millimeters */
    h4 { font-size: 12pt; }      /* points */
    h5 { font-size: 1pc; }       /* picas */
    h6 { font-size: 12px; }      /* pixcles */
  </style>
</head>
<body>
    <h1>Heading level 1</h1>
    <h2>Heading level 2</h2>
    <h3>Heading level 3</h3>
    <h4>Heading level 4</h4>
    <h5>Heading level 5</h5>
    <h6>Heading level 6</h6>
</body>
</html>
```



Heading level 1

Heading level 2

Heading   level   3

Heading level 4

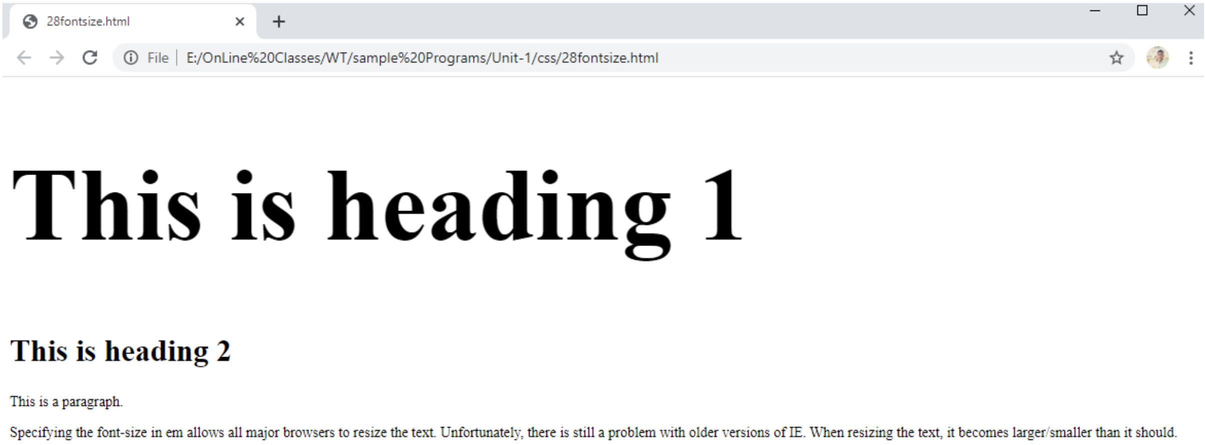
Heading level 5

Heading level 6

Relative Lengths:

- Relative length units specify a length relative to another length property. Relative length units scales better between different rendering mediums.
- Relative length units specify a length relative to another length property.
- The em and ex units depend on the font size that's applied to the element.
- The default size for the fonts in all modern browsers is 16px.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  font-size: 6.25em; /* 100px/16=6.25em */
}

h2 {
  font-size: 1.875em; /* 30px/16=1.875em */
 }

p {
  font-size: 0.875em; /* 14px/16=0.875em */
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>Specifying the font-size in em allows all major browsers to resize the text.
Unfortunately, there is still a problem with older versions of IE. When resizing the text, it
becomes larger/smaller than it should.</p>

</body>
</html>
```


28fontsize.html
File
E:/OnLine%20Classes/WT/sample%20Programs/Unit-1/css/28fontsize.html

This is heading 1

This is heading 2

This is a paragraph.

Specifying the font-size in em allows all major browsers to resize the text. Unfortunately, there is still a problem with older versions of IE. When resizing the text, it becomes larger/smaller than it should.

css selector forms:

CSS selectors are used *to select the content you want to style*. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

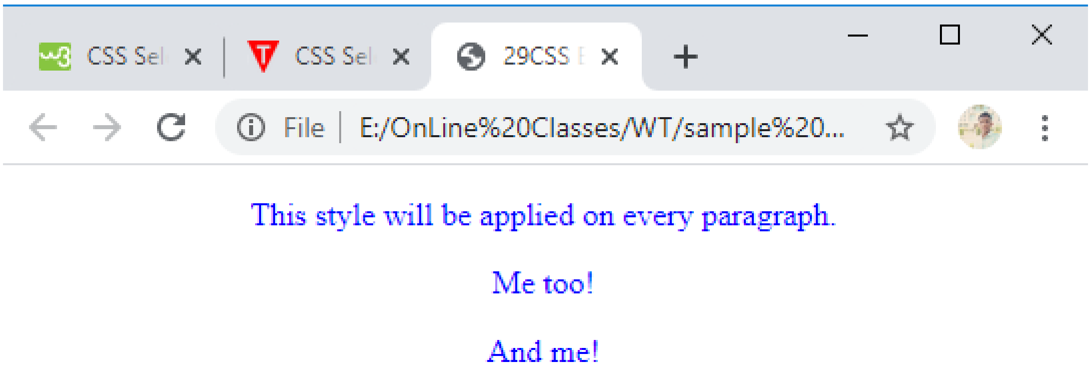
There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

1) CSS Element Selector

The element selector selects the HTML element by name. This style will be applied on every paragraph.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```



This style will be applied on every paragraph.

Me too!

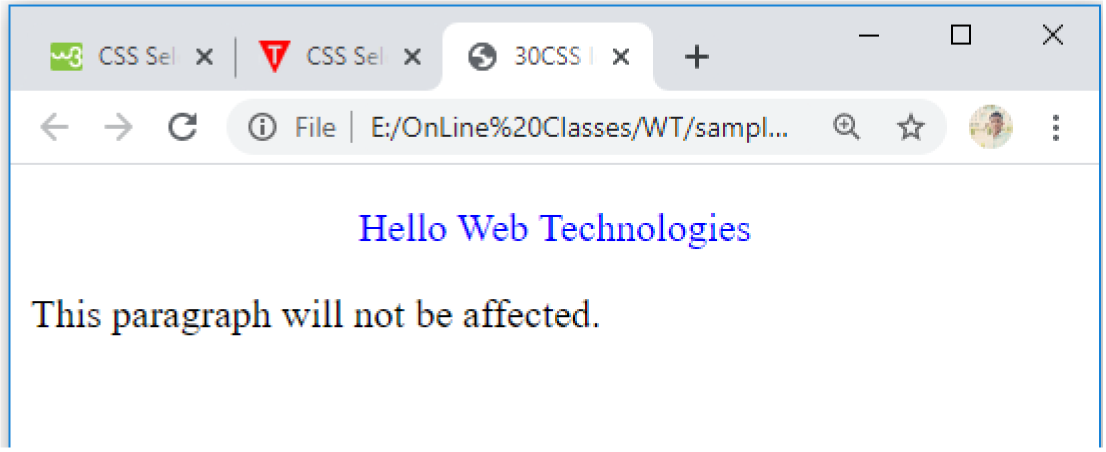
And me!

2) CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element. It is written with the hash character (#), followed by the id of the element.

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p id="para1">Hello Web Technologies</p>
```

```
<p>This paragraph will not be affected.</p>
</body>
</html>
```



Hello Web Technologies

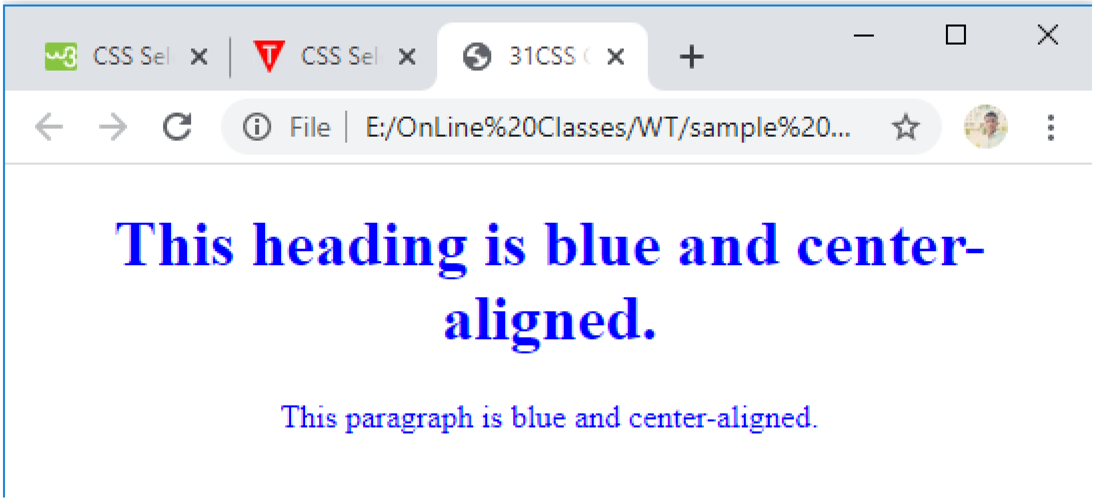
This paragraph will not be affected.

3) CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

Note: A class name should not be started with a number.

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is blue and center-aligned.</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>
```



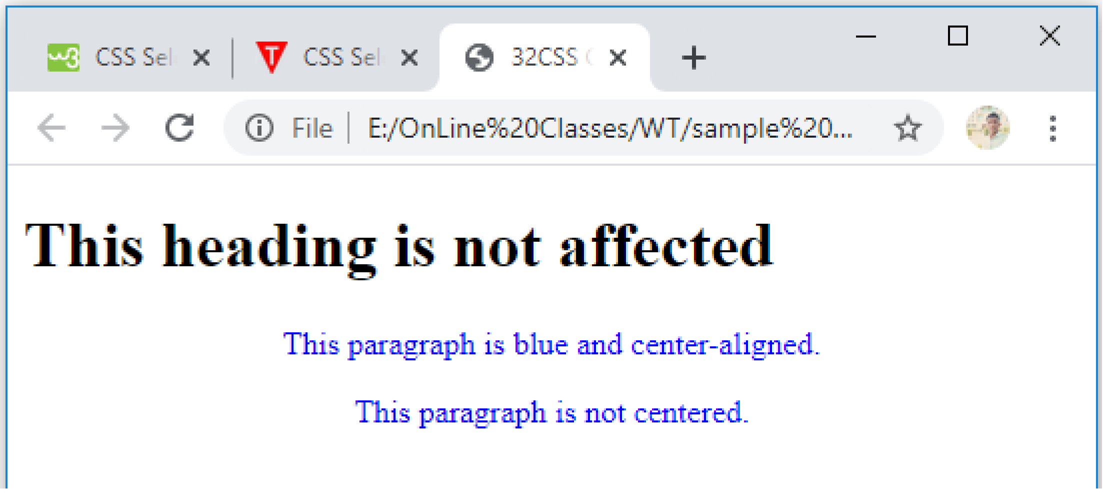
This heading is blue and center-aligned.

This paragraph is blue and center-aligned.

4)CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
   text-align: center;
   color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is not affected</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
<p class="center">This paragraph is not centered.</p>
</body>
</html>
```



This heading is not affected

This paragraph is blue and center-aligned.

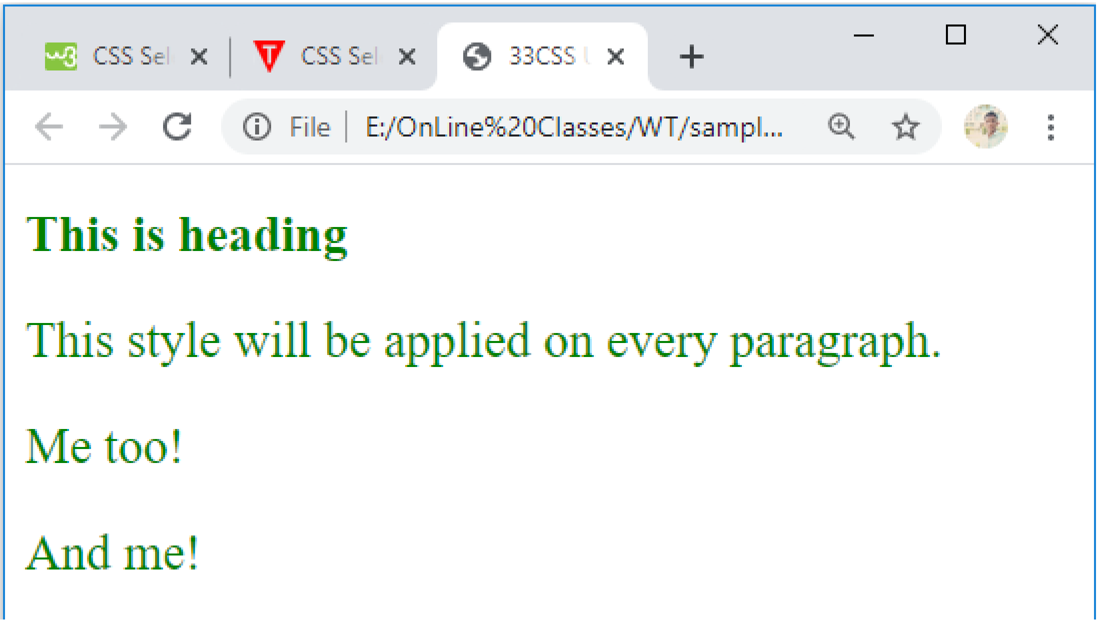
This paragraph is not centered.

5) CSS Universal Selector

The universal selector is used as a wildcard character. It selects all the elements on the pages.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  color: green;
  font-size: 20px;
}
</style>
</head>
<body>
<h2>This is heading</h2>
<p>This style will be applied on every paragraph.</p>
```

```
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```



This is heading

This style will be applied on every paragraph.

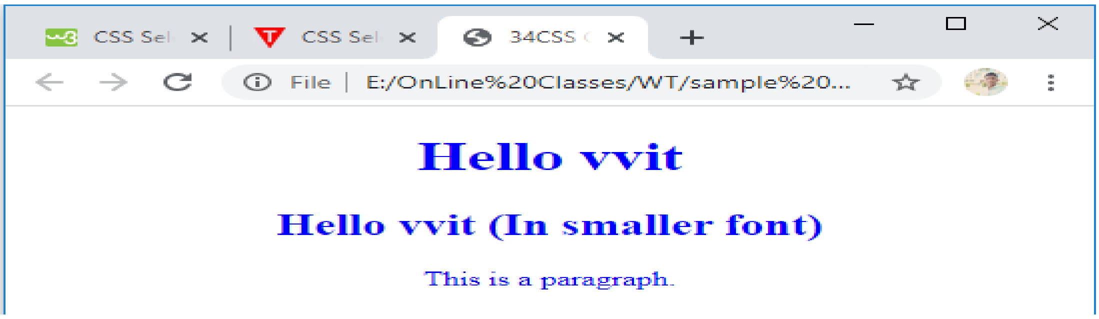
Me too!

And me!

6) CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions. Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
   text-align: center;
   color: blue;
}
</style>
</head>
<body>
<h1>Hello vvit</h1>
<h2>Hello vvit (In smaller font)</h2>
<p>This is a paragraph.</p>
</body>
</html>
```



Hello vvit

Hello vvit (In smaller font)

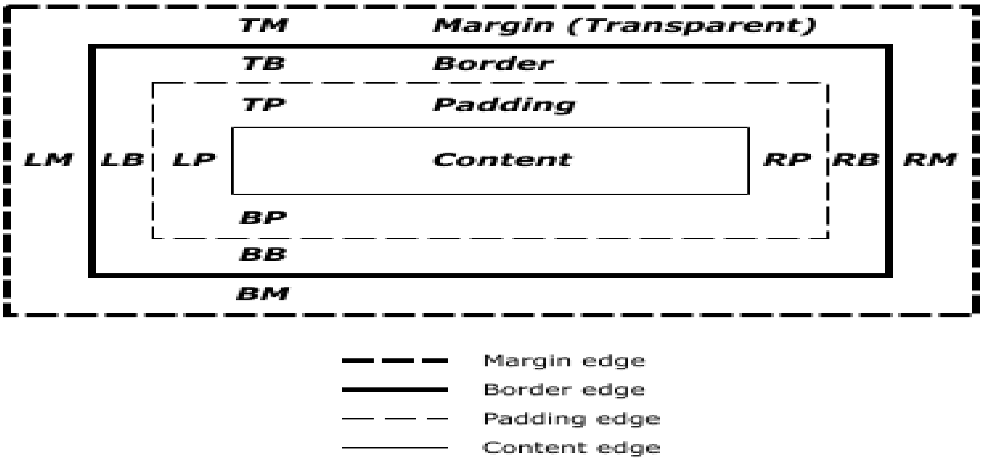
This is a paragraph.

CSS Box model:

CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model. Box-Model has multiple properties in CSS. Some of them are given below:

- borders
- margins
- padding
- Content

The following figure illustrates the box model.



Border Area: It is the area between the box's padding and margin. Its dimensions are given by the width and height of border.

Margin Area: This area consists of space between border and margin. The dimensions of Margin area are the margin-box width and the margin-box height. It is useful to separate the element from its neighbors.

Padding Area: It includes the element's padding. This area is actually the space around the content area and within the border box. Its dimensions are given by the width of the padding-box and the height of the padding-box.

Content Area: This area consists of content like text, image, or other media content. It is bounded by the content edge and its dimensions are given by content box width and height.

Conflict Resolution:

If there are two or more **conflicting CSS** rules that point to the same element, the browser follows some rules to determine which one is most specific and therefore wins out.

- **Cascade**
- **Specificity**
- **Inheritance**

Cascade:

In CSS, styles sheets **cascade** by order of importance. If rules in different style sheets conflict with one another, the rule from the most important style sheet wins.

When there are two different values for the same property on the same element, there is an obvious conflict that the browser must resolve.

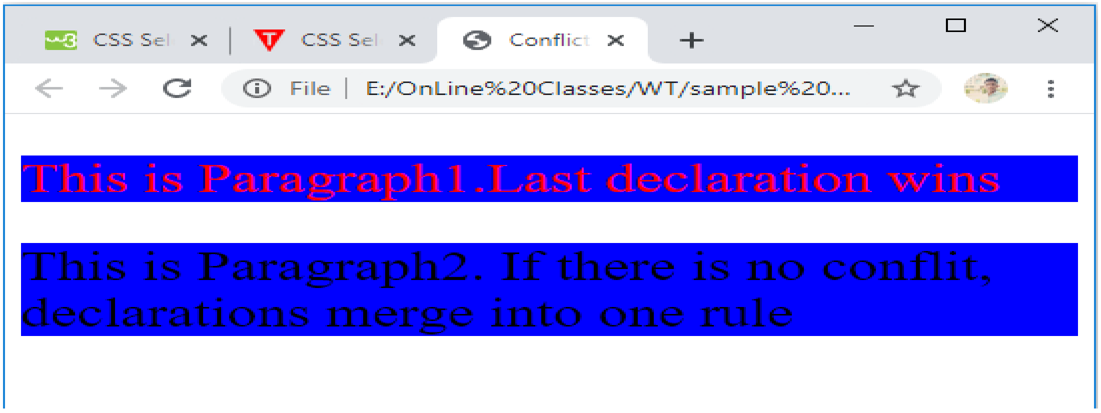
The conflict is resolved by the precedence of the three different levels of style sheets.

In-line have precedence over internal and external style sheets. Internal style sheet have precedence over external style sheet.

external.css

```
p {

    background-color : blue;

    color: red;

    font-size : 30px;

}
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Conflict Demo</title>
<link rel="stylesheet" href="external.css">
    <style>

    p { color:red  ; }
    </style>


  </head>
  <body>
<p>This is Paragraph1.Last declaration wins</p>
<p style="color:black">This is Paragraph2. If there is no conflit, declarations merge into one rule</p>
</header>
    </body>
</html>
```



This is Paragraph1.Last declaration wins

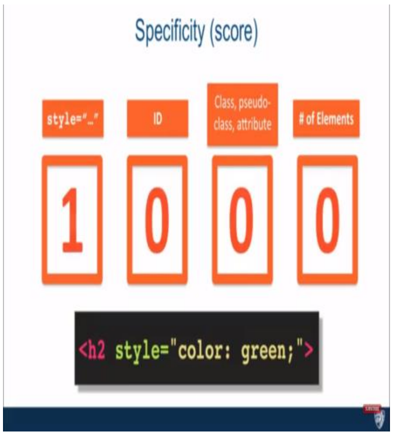
This is Paragraph2. If there is no conflit, declarations merge into one rule

Specificity:

- **Specificity -** CSS rules often conflict with one another. In fact, this is what we want. The trick is understanding how conflicting rules will apply.
- Specificity is one of the ways that conflicting rules are applied.

Here's how the scoring system works:

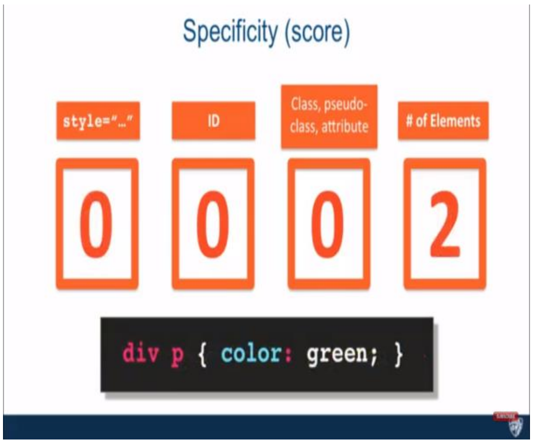
- Add one to A for the style attribute
- Add one to A for each ID in the selector
- Add one to B for each class or pseudo class in the selector
- Add one to C for each element name



Specificity (score)

style="..."	ID	Class, pseudo-class, attribute	# of Elements
1	0	0	0

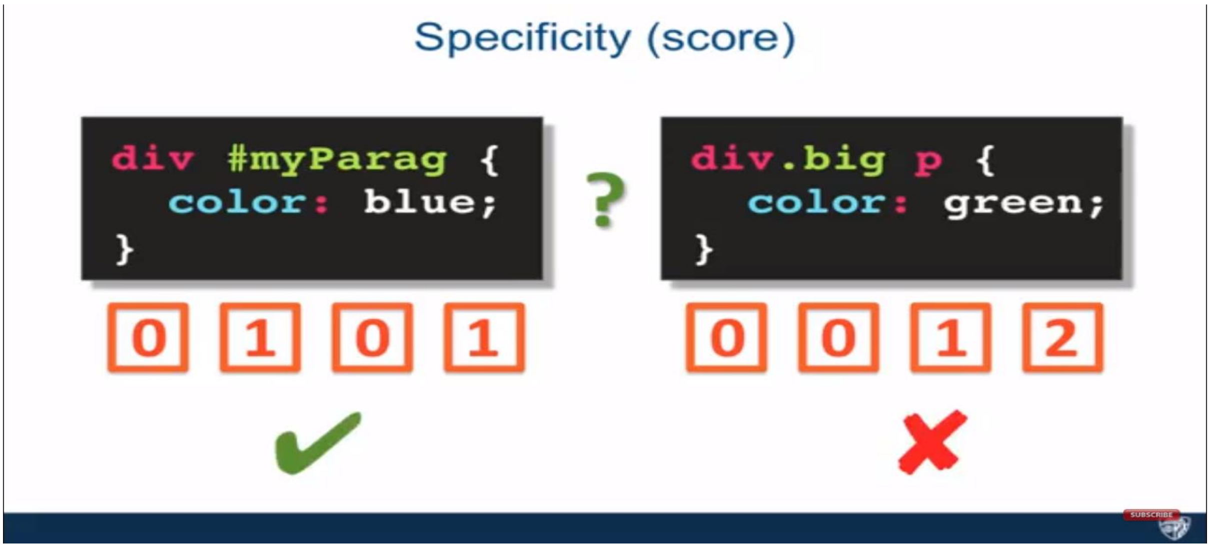
```
<h2 style="color: green;">
```



Specificity (score)

style="..."	ID	Class, pseudo-class, attribute	# of Elements
0	0	0	2

```
div p { color: green; }
```



Specificity (score)

```
div #myParag {
   color: blue;
}
```

0 1 0 1 ✔

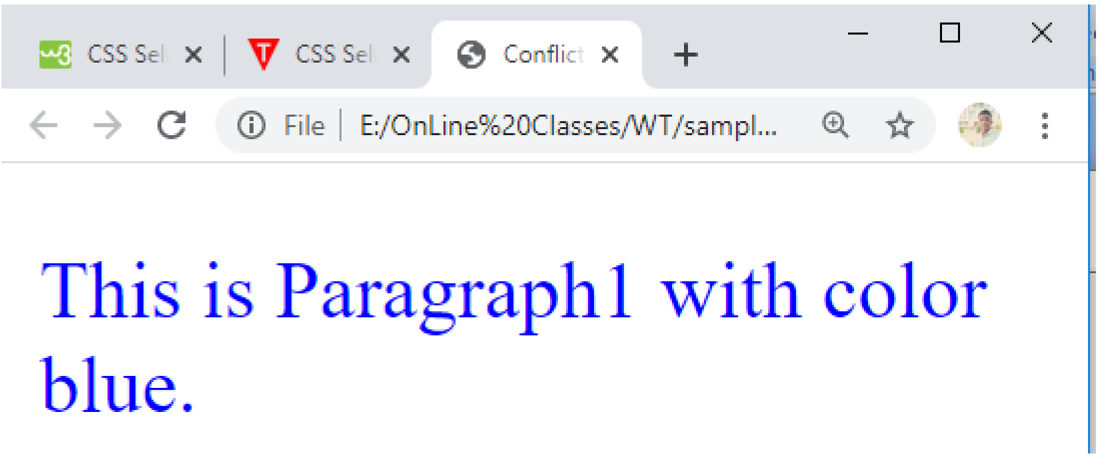
?

```
div.big p {
   color: green;
}
```

0 0 1 2 ✘

```
<!DOCTYPE html>
<html>
   <head>
      <title>Conflict Demo</title>
      <style>
      header.a1 p{ color : blue; }  /* this has the high specificity*/
      p.a2 { color : red; }
      /*p { color:green  !important } */
      </style>
    </head>
   <body>
<header class="a1">
<p class="a2"> This is Paragraph1 with color blue.< /p>
```

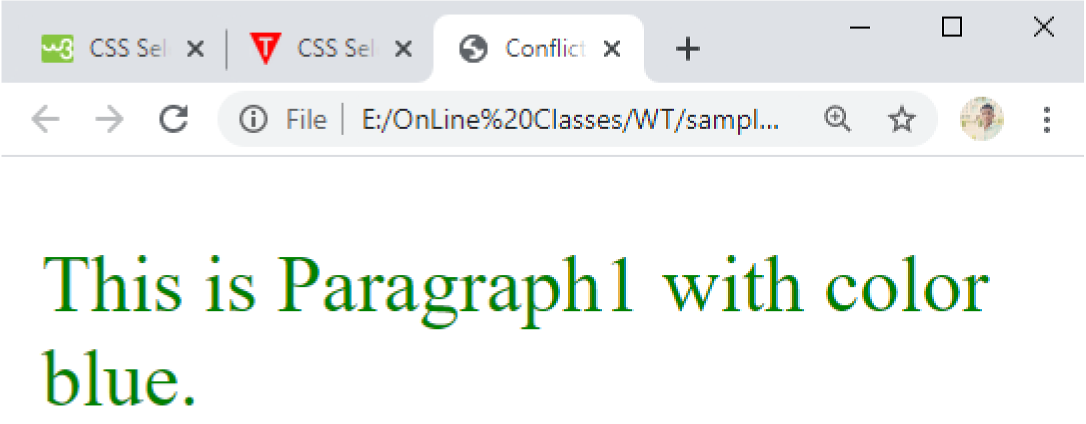
```
</header>
    </body>
</html>
```



This is Paragraph1 with color blue.

- To override the specificity we have to use important.

```
<style>
            header.a1 p{ color : blue; }
            p.a2 { color : red; }
            p { color:green !important; }
    </style>
<!DOCTYPE html>
<html>
  <head>
    <title>Conflict Demo</title>
    <style>
    header.a1 p{ color : blue; }
    p.a2 { color : red; }
    p { color:green  !important }
    </style>
   </head>
  <body>
<header class="a1">
<p class="a2"> This is Paragraph1 with color blue.< /p>

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



This is Paragraph1 with color blue.