Module 1:
HTML
Hyper Text Markup Language

Outline

- What is HTML ?
- Different HTML Tags
- Text formatting tags
- Tables, Frames tags
- Use of various HTML Tag for Form Design
- HTML5 Features

HTML: **H**yper**T**ext **M**arkup **L**anguage

- HTML documents are simply text documents with a specific form
  - Documents comprised of **content** and **markup tags**
    - **Content**: actual information being conveyed
    - **markup tags:** tell the Web browser **how to display** the page
  - An HTML file must have an **.htm** or **.html** file extension
  - An HTML file can be created using a **simple text editor**

Creating HTML Pages

- HTML files can be created with text editors:
  - NotePad, NotePad ++,wordpad.

- Or HTML editors (WYSIWYG Editors):
  - Microsoft FrontPage
  - Macromedia Dreamweaver
  - Microsoft Word
  - Visual Studio

First Example

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
This is my first homepage.
</body>
</html>
```

Open this file using a browser, and you will see…

HTML Tags

- HTML tags are used to mark-up HTML elements
  - Surrounded by angle brackets < and >
  - HTML tags normally come in pairs, like <tagname> (start tag) and </tagname> (end tag)
  - The text between the start and end tags is the element content
  - Not case-sensitive
  - Follow the latest web standards: Use lowercase tags

HTML Document: Basic Structure

- Every HTML page begins with a document type declaration that informs the browser which version of HTML the page is being used
- E.g.

  <!Doctype HTML> //in HTML 5

- Begins with `<html>` and ends with `</html>`
- The two primary structural components in HTML are the head and the body

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

HTML Tags

- Headings Tags
- Paragraph Tags
- Formatting Tags
  - Hyperlinks
  - Images

HTML Headings

- Headings are defined with the <h1> to <h6> tags.
- <h1> defines the most important heading. <h6> defines the least important heading.
- Example:

```
<!DOCTYPE html>
<html>
<body>
<h1>Heading 1</h1>          (largest)
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>          (smallest)
</body>
</html>
```

HTML Headings

- **Headings Are Important**
- Search engines use the headings to index the structure and content of your web pages.
- Users skim(scan) your pages by its headings.
- It is important to use headings to show the document structure.
- <h1> headings should be used for main headings, followed by <h2> headings, then the less important <h3>, and so on.

HTML Horizontal Rules <hr>

The <hr> element is used to separate content (or define a change) in an HTML page:

- **Example:**

```
<!DOCTYPE html>
<html>
<body>
<h1>This is heading 1</h1>
<hr>
<h2>This is heading 2</h2>
</body>
</html>
```

HTML Basic Tags

Tag	Description
<html>	Defines the root of an HTML document
<body>	Defines the document's body
<head>	A container for all the head elements (title, scripts, styles, meta information, and more)
<h1> to <h6>	Defines HTML headings
<hr>	Defines a thematic change in the content

HTML Paragraph <p> Tags

- The HTML <p> element defines a **paragraph**:
- **Example:**

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

HTML Line Breaks
 tag

- The HTML
 element defines a **line break**.
- Use
 if you want a line break (a new line) without starting a new paragraph
- **Example:**

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

HTML preformatted <pre> Element

- The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks
- **Example**:

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

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

Summary: HTML Paragraph Tags

Tag	Description
<p>	Defines a paragraph
 	Inserts a single line break
<pre>	Defines pre-formatted text

HTML Attributes

- Tags can have attributes that provide additional information to an HTML element
  - Attributes always come in pairs like: **name = “value”**
  - Attributes are always specified in the start tag
  - **Attribute values** should always be enclosed in quotes. Double quotes are most common.
  - Also **case-insensitive**: however, lowercase is recommended
  - **<tagname a1=“v1” a2=“v2”></tagname>**
  - For example,

  <table border=“0”> is a start tag that defines a table that has no borders

HTML Style Attribute

- Setting the style of an HTML element, can be done with the **style attribute**.
- The HTML style attribute has the following **syntax**:

<tagname style="*property*:*value;*">

HTML Background Color

- The background-color property defines the background color for an HTML element.
- **Example**:

```
<!DOCTYPE html>
<html>
<body style="background-color:blue“>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

HTML Text Color

- The color property defines the text color for an HTML element:
- **Example**:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:yellow">This is a heading</h1>
<p style="color:red">This is a paragraph.</p>
</body>
</html>
```

HTML Text Size

- The font-size property defines the text size for an HTML element
- **Example:**

```
<!DOCTYPE html>
<html>
<body>
<h1 style="font-size:300%“>This is a heading</h1>
<p style="font-size:160%">This is a paragraph.</p>
</body>
</html>
```

HTML Text Alignment

- The text-align property defines the horizontal text alignment for an HTML element:
- **Example:**

```
<!DOCTYPE html>
<html>
<body>
<h1 style="text-align:center">Centered Heading</h1>
<p style="text-align:center">Centered paragraph.</p>
</body>
</html>
```

HTML Text Formatting Tags

- HTML also defines special **elements** for defining text with a special **meaning**.
- Formatting elements were designed to display special types of text:

<b> - Bold text

** - Important text**

<i> - Italic text

** - Emphasized text**

<mark> - Marked text

<small> - Small text

<del> - Deleted text

<ins> - Inserted text

<sub> - Subscript text

<sup> - Superscript text

HTML <b> and Elements

- The HTML <b> element defines **bold** text, without any extra importance.
- The HTML element defines **strong** text, with added semantic "strong" importance.
- **Example**:
  <!DOCTYPE html>

```
<html>
<body>
<p>This text is normal.</p>
<p><b>This text is bold.</b></p>
<p><strong>This text is strong.</strong></p>
</body>
</html>
```

HTML <i> and Elements

- The HTML <i> element defines *italic* text, without any extra importance.
- The HTML element defines *emphasized* text, with added semantic importance.
- **Example:**

```
<!DOCTYPE html>
<html>
<body>
<p><i>This text is italic.</i></p>
<p><em>This text is emphasized.</em></p>
</body>
</html>
```

HTML <small> Element

- The HTML <small> element defines smaller text:
- **Example**:

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML <small>Small </small> Formatting</h2>
</body>
</html>
```

HTML <mark>, <del> Element

- The HTML <del> element defines deleted (removed) text.
- The HTML <mark> element defines marked or highlighted text
- **Example:**

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML <mark>Marked </mark> Formatting</h2>
<p>My favorite color is <del> blue </del> red.</p>
</body>
</html>
```

HTML <sub> , <sup>Element

- The HTML <sub> element defines $_{\text{subscripted}}$ text.
- The HTML <sup> element defines $^{\text{superscripted}}$ text.
- **Example:**

```
<!DOCTYPE html>
<html>
<body>
<p>This is <sub> subscripted </sub> text.</p>
<p>This is <sup> superscripted </sup> text.</p>
</body>
</html>
```

Summary: HTML Text Formatting Tags

Tag	Description
<b>	Defines bold text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines smaller text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

HTML Comment Tags

- You can add comments to your HTML source by using the following syntax:

<!-- Write your comments here -->

- **Example**:

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

HTML Image Tags

- In HTML, images are defined with the <img> tag.
- The **<img>** tag is empty, it contains attributes only, and does **not have a closing tag.**
- The src attribute specifies the URL (web address) of the image:

<img src="*url*">

- **alt** Attribute: provides an alternate text for an image, if the user for some reason cannot view it
- The value of the alt attribute should describe the image:

<img src="img_chania.jpg" alt="Flowers in Chania">

HTML Image Tags(contd..)

- The width and height attributes always defines the width and height of the image in pixels.
- **Note:** the browser expects to find the image in the same folder as the web page.
- **Example**:

```
<!DOCTYPE html>
<html>
<body>
<h2>Image Size</h2>
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
</body>
</html>
```

HTML Links - Hyperlinks

- HTML links are hyperlinks.
- You can click on a link and jump to another document.
- **Syntax:**

link text

- **Example:**

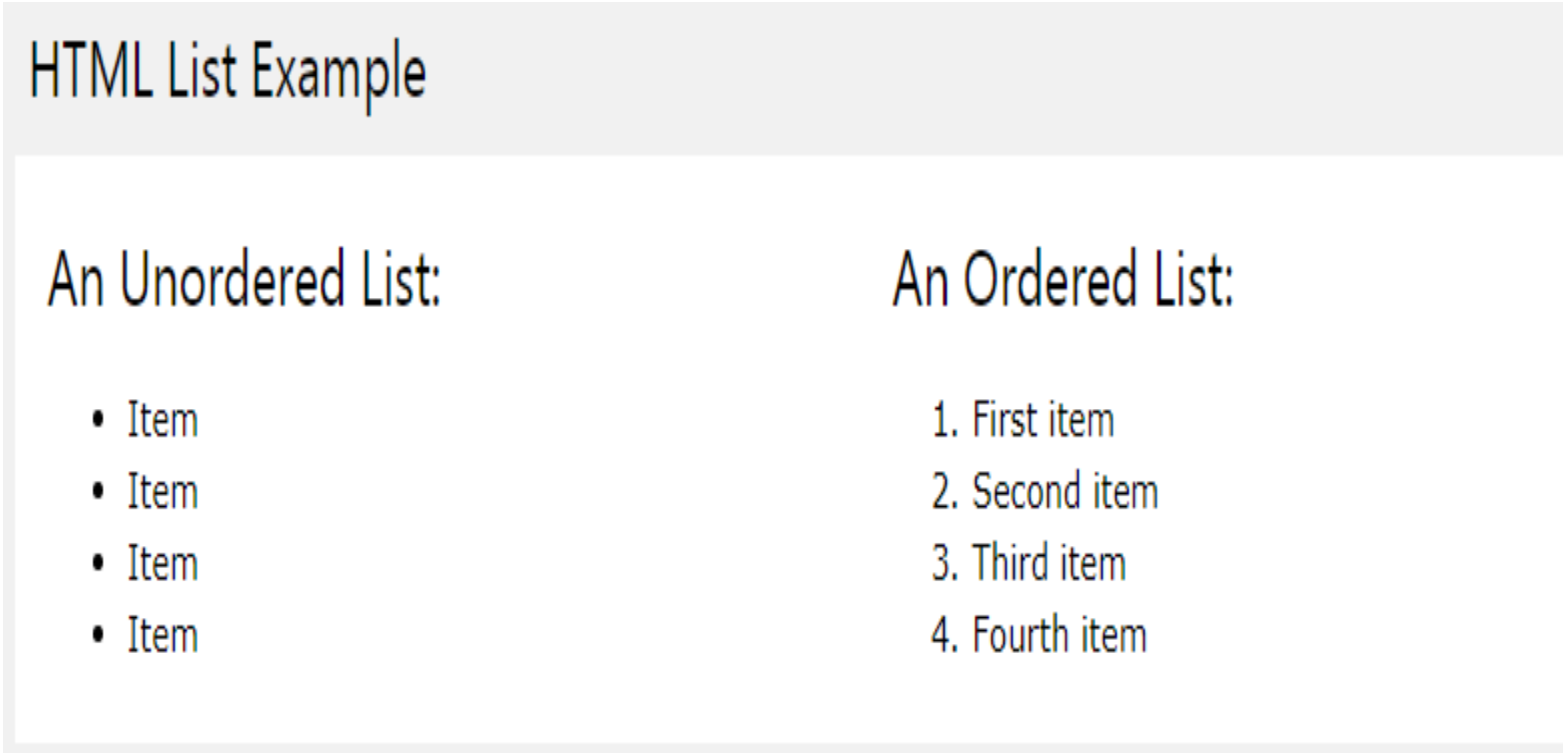
```
<!DOCTYPE html>
<html>
<body>
<a href="https://www.google.com">Visit our Search Engine</a>
</body>
</html>
```

HTML Links - Image as Link

- It is common to use images as links:

```
<!DOCTYPE html>
<html>
<body>
<h2>Image Links</h2>
<a href=“www.google.com">
<img src="smiley.gif" alt="HTML tutorial“
   style="width:42px;height:42px;border:0">
</a>
</body>
</html>
```

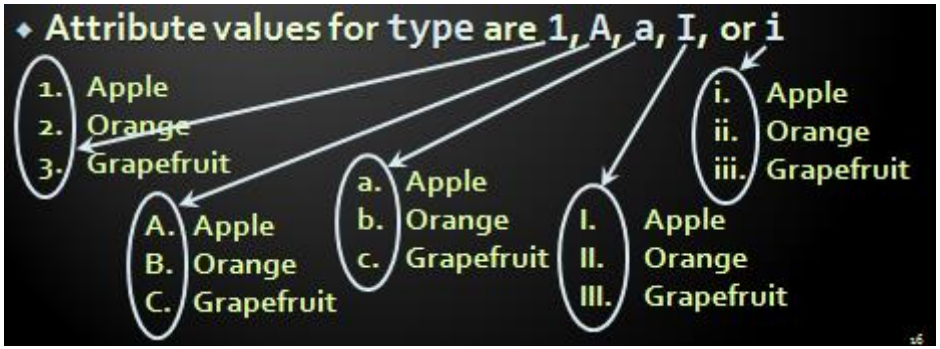
HTML Lists


HTML List Example
An Unordered List:
Item
Item
Item
Item
An Ordered List:
1. First item
2. Second item
3. Third item
4. Fourth item

Ordered Lists: <ol> Tag

- Create an Ordered List using :

```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ol>
```



Ordered HTML List - The Type Attribute

The `type` attribute of the `<ol>` tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

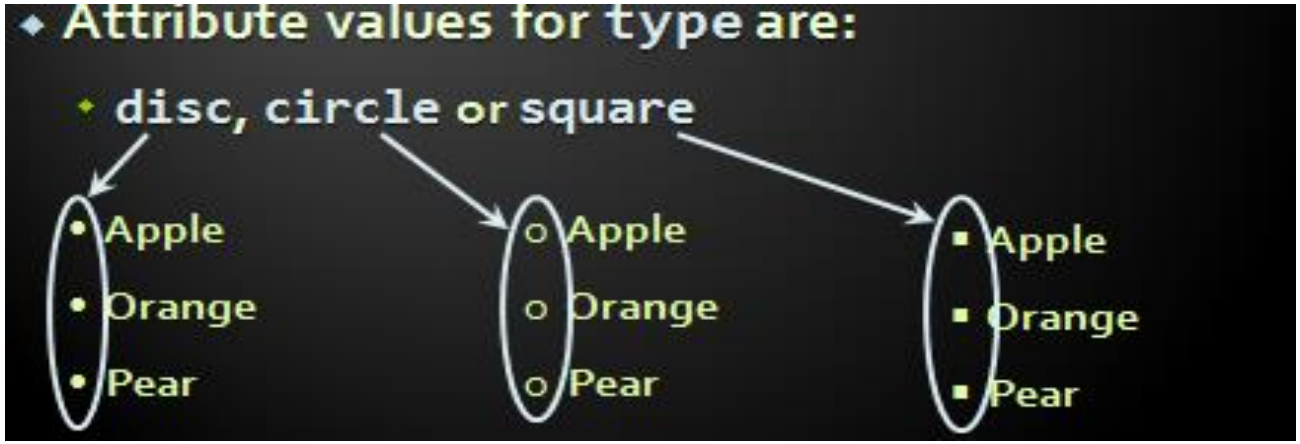
Example

```
<!DOCTYPE html>
<html>
<body>
<h2>An ordered HTML list</h2>
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
</body>
</html>
```

Unordered Lists: <ul> Tag

- Create an Unordered List using :

```
<ul type="disk">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ul>
```



Unordered HTML List - Choose List Item Marker

The CSS `list-style-type` property is used to define the style of the list item marker

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Unordered List

```
<!DOCTYPE html>
<html>
<body>
<h2>Unordered List with Disc Bullets</h2>
<ul style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
</body>
</html>
```

Definition lists: <dl> tag

- Create definition lists using <dl>
  - Pairs of text and associated definition; text is in <dt> tag, definition in <dd> tag

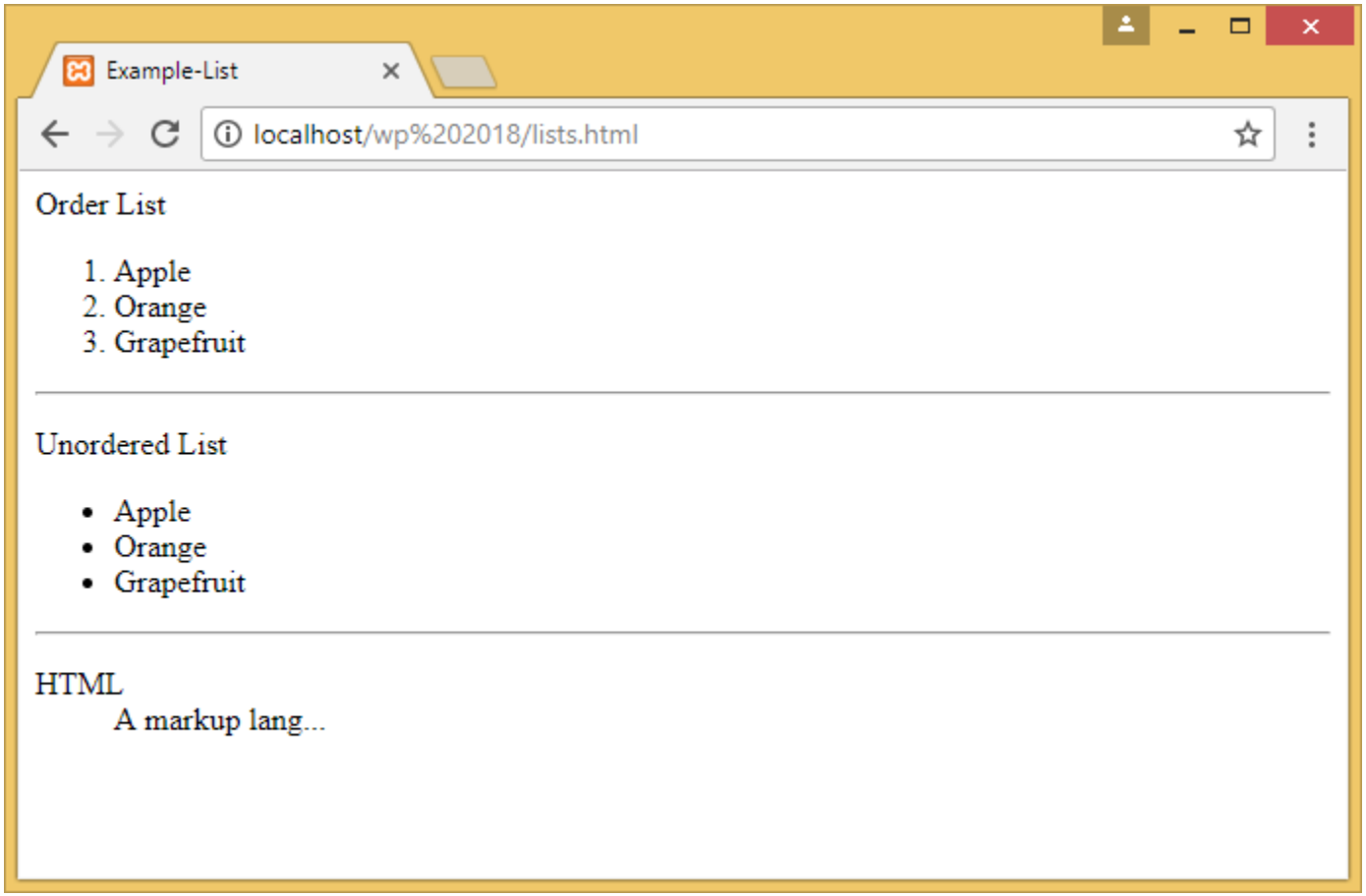
```
<dl>
   <dt>HTML</dt>
   <dd>A markup language …</dd>
   <dt>CSS</dt>
   <dd>Language used to …</dd>
</dl>
```

  - Renders without bullets
  - Definition is indented

Lists – Example

```
<p> Order List</p>
<ol type="1">
    <li>Apple</li>
    <li>Orange</li>
    <li>Grapefruit</li>
</ol>
<hr>
<p>Unordered List</p>
<ul type="disc">
    <li>Apple</li>
    <li>Orange</li>
    <li>Grapefruit</li>
</ul>
<hr>
<dl>
    <dt>HTML</dt>
    <dd>A markup lang...</dd>
</dl>
```

Output:


Example-List
localhost/wp%202018/lists.html
Order List
1. Apple
2. Orange
3. Grapefruit
Unordered List
• Apple
• Orange
• Grapefruit
HTML
A markup lang...

HTML Table

- An HTML table is defined with the <table> tag.
- Each table row is defined with the <tr> tag.
- A table header is defined with the <th> tag.
- By default, table headings are bold and centered.
- A table data/cell is defined with the <td> tag.

Example

```
<!DOCTYPE html>
<html>
<body>
<table style="width:100%">
 <tr>
       <th>Firstname</th>
       <th>Lastname</th>
       <th>Age</th>
 </tr>
<tr>
   <td>Jill</td>
   <td>Smith</td>
   <td>50</td>
  </tr>
  <tr>
</table>
</body>
</html>
```

HTML Table - Adding a Border

- If you do not specify a border for the table, it will be displayed without borders.

- A border is set using the CSS border property need to be specified inside <head> tag within <style> tag

```
<head>
<style>
table, th, td {
border: 1px solid black;
}
</style>
</head>
```

HTML Table - Adding Cell Padding

- Cell padding specifies the space between the cell content and its borders.
- If you do not specify a padding, the table cells will be displayed without padding.

```
<head>
<style>
th, td {
  padding: 15px;
}
</style>
</head>
```

HTML Table - Left-align Headings

- By default, table headings are bold and centered.
- To left-align the table headings, use the CSS text-align property:

```
<head>
<style>
th {
    text-align: left;
  }
</style>
</head>
```

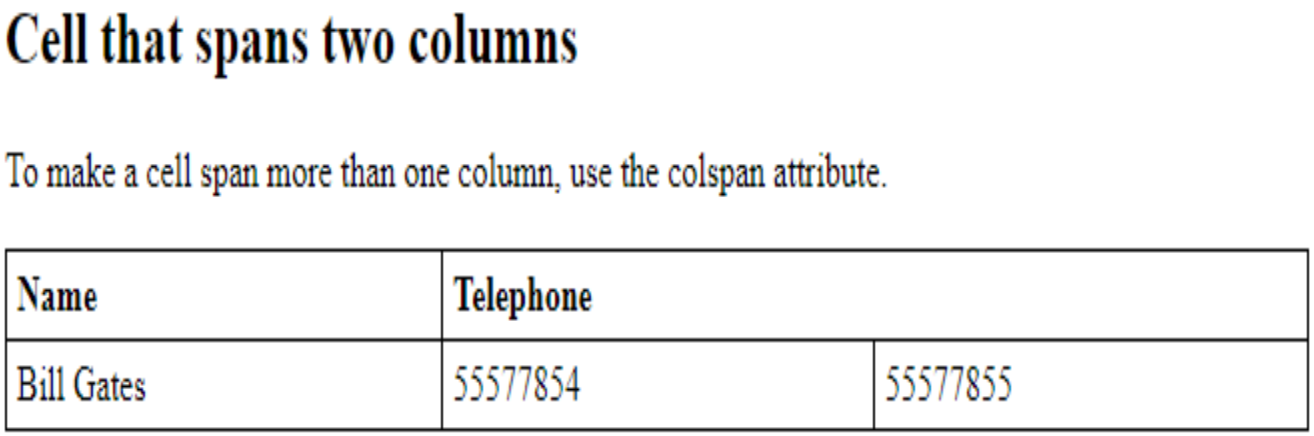
HTML Table - Adding a Caption

- To add a caption to a table, use the <caption> tag:
- **<caption> tag should be specified immediately after table tag in html document**
- Example:
  <caption>Monthly savings</caption>

```
<!DOCTYPE html>
<html>
<head> <style>
table, th, td {
  border: 1px solid black;
}
th, td {
  padding: 5px;
  text-align: left;
}
</style> </head>
<body>

<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table> </body> </html>
```

HTML Table - Cells that Span Many Columns

- To make a cell span more than one column, use the colspan attribute:



Cell that spans two columns

To make a cell span more than one column, use the colspan attribute.

Name	Telephone	
Bill Gates	55577854	55577855

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<table style="width:100%">
 <tr>
   <th>Name</th>
   <th colspan="2">Telephone</th>
 </tr>
<tr>
   <td>Bill Gates</td>
   <td>55577854</td>
   <td>55577855</td>
 </tr>
</table>
</body>
</html>
```

HTML Table - Cells that Span Many Rows

- To make a cell span more than one row, use the rowspan attribute:

Cell that spans two rows

To make a cell span more than one row, use the rowspan attribute.

Name:	Bill Gates
Telephone:	55577854
	55577855

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
</body>
</html>
```

HTML: 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.

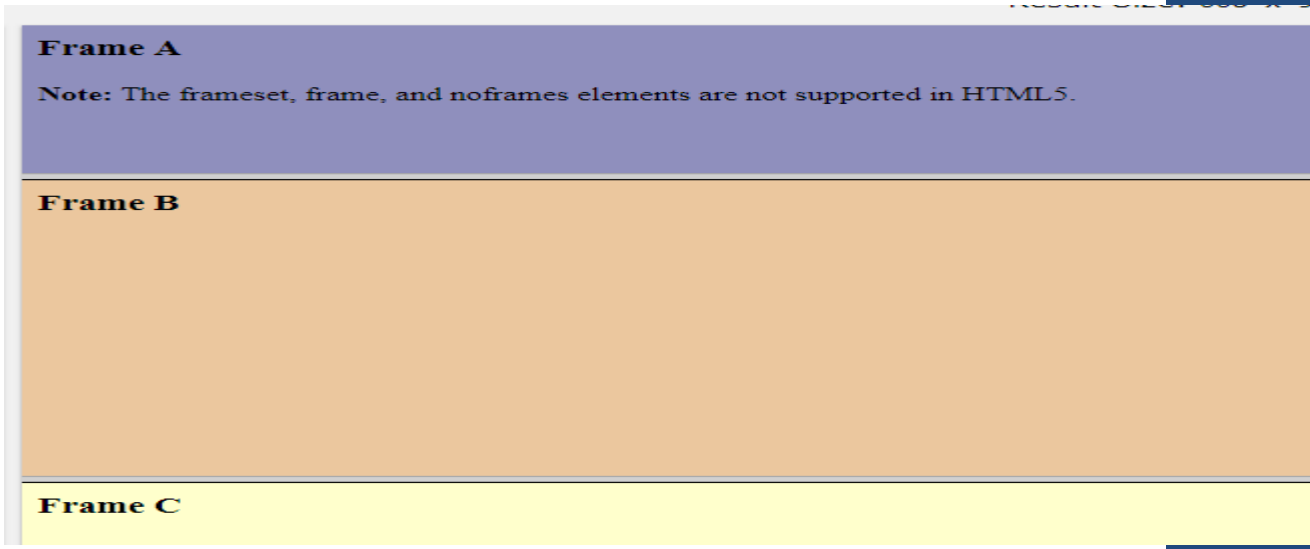
Frame Tag

- The <frameset> tag is not supported in HTML5.
- The <frameset> tag defines a frameset.
- The <frameset> element holds one or more <frame> elements.

  Each <frame> element can hold a separate document.
- The <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.

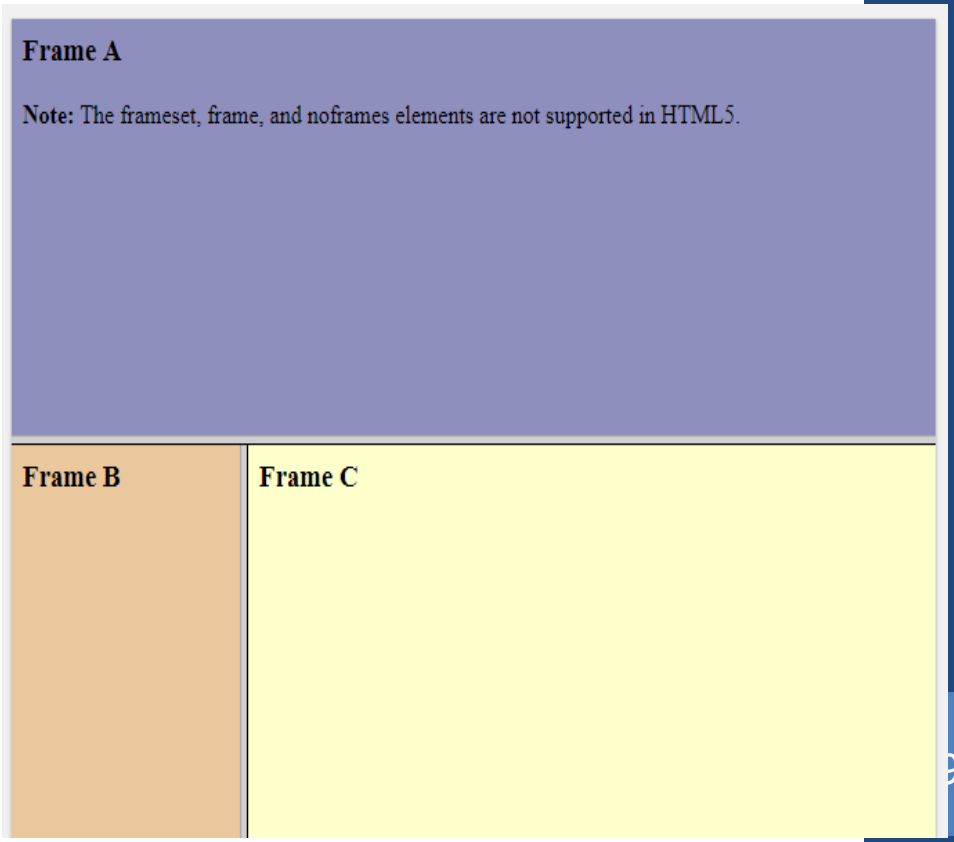
Horizontal Frameset Example

```
<!DOCTYPE html>
<html>
<frameset rows="25%,*,25%">
  <frame src="frame_a.html">
  <frame src="frame_b.html">
  <frame src="frame_c.html”>
</frameset>
</html>
```



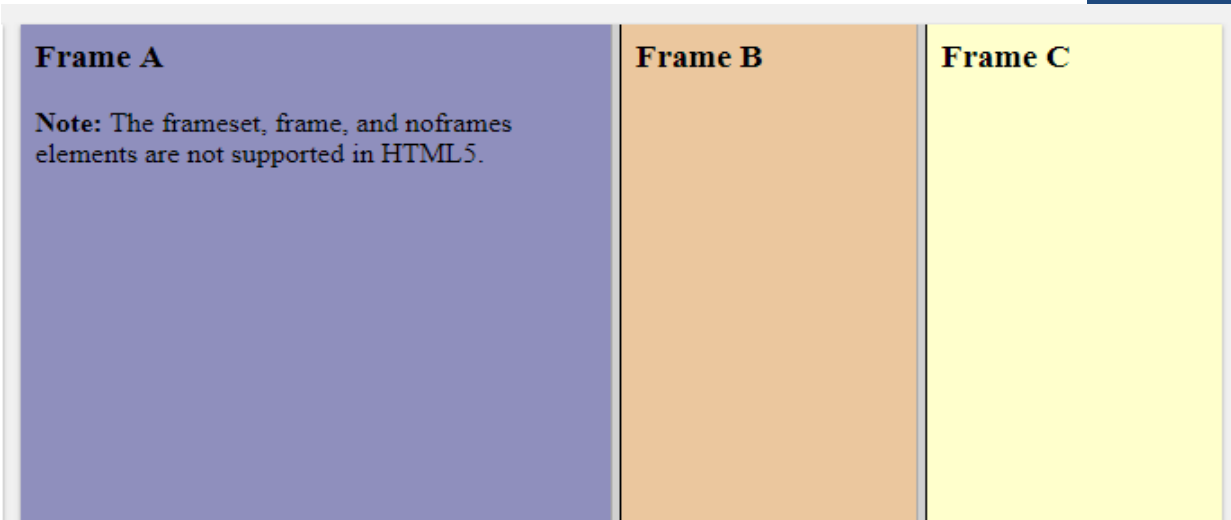
Mixed Frameset Example

```
<!DOCTYPE html>
<html>

<frameset rows="50%,50%">
  <frame src="frame_a.html">
  <frameset cols="25%,75%">
    <frame src="frame_b.html">
    <frame src="frame_c.html">
  </frameset>
</frameset>

</html>
```



Frameset with NoResize Example

```
<!DOCTYPE html>
<html>
<frameset cols="50%,*,25%">
  <frame src="frame_a.htm" noresize="noresize">
  <frame src="frame_b.htm">
  <frame src="frame_c.htm">
</frameset>
</html>
```

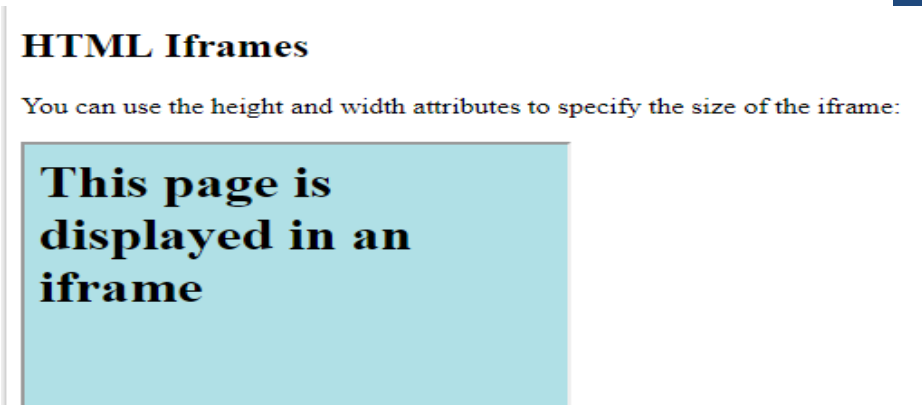


iframes Tag

- You can define an **inline frame** with HTML tag **<iframe>**.
- Can appear anywhere in your document.
- The <iframe> tag defines a **rectangular region within the document in which the browser can display a separate document, including scrollbars and borders.**
- An inline frame is used to embed another document within the current HTML document.
- The **src** attribute is used to specify the URL of the document that occupies the inline frame.

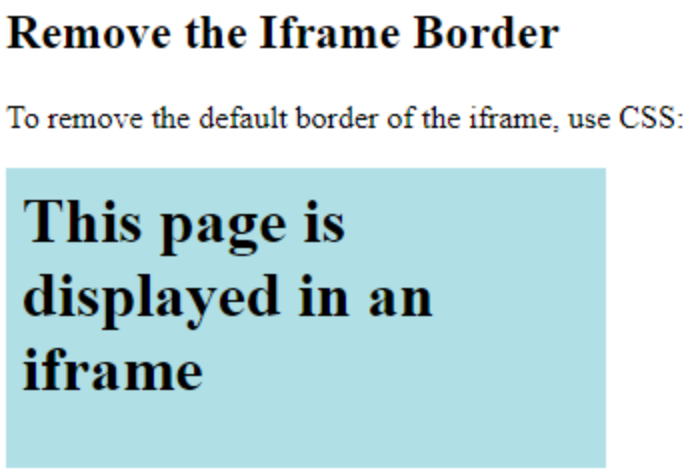
iframe Tag Example

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML Iframes</h2>
<p>You can use the height and width attributes to specify the size
   of the iframe:</p>

<iframe src="demo_iframe.htm" height="200“
   width="300"></iframe>

</body>
</html>
```



iframe Tag Example

```
<!DOCTYPE html>
<html>
<body>
<h2>Remove the Iframe Border</h2>
<p>To remove the default border of the iframe, use CSS:</p>
<iframe src="demo_iframe.htm“ style="border:none;"></iframe>
</body>
</html>
```



iframe Tag Example

```
<!DOCTYPE html>
<html>
<body>
<h2>Custom Iframe Border</h2>
<p>With CSS, you can also change the size, style and color of the
   iframe's border:</p>

<iframe src="demo_iframe.htm" style="border:2px solid
   red;"></iframe>

</body>
</html>
```

tag

- The tag defines a division or a section in an HTML document.
- The element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.

div Tag Example

```
<!DOCTYPE html>
<html>
<body>
<p>This is some text.</p>
<div style="background-color:lightblue">
<h3>This is a heading in a div element</h3>
<p>This is some text in a div element.</p>
</div>
<p>This is some text.</p>
</body>
</html>
```

HTML Forms

HTML Forms

- HTML Forms are required, when you **want to collect some data from the site visitor.**
- For example, during user registration you would like to collect information such as name, email address, credit card, etc.
- 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.
- There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

Forms Syntax

<form **action** = "Script URL“ **method** = "GET/POST">

form elements like input, textarea etc.

</form>

action

- Backend script ready to process your passed data.

method

- Method to be used to upload data. The most frequently used are GET and POST methods.

Forms Syntax

Form elements

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

target

- Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.

GET Method

- The default method when submitting form data is GET.
- However, when GET is used, the submitted form data will be **visible in the page address field**:

When to use GET??

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user wants to bookmark the result
- GET is better for non-secure data, like query strings in Google

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

<p>This form will be submitted using the GET method:</p>

<form method="GET“   target="_blank" >
  First name:<br>
  <input type="text" name="firstname" value="Mickey">
  <br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
  <br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

POST Method

- Always use POST if the form data contains sensitive or personal information.
- The POST method does not display the submitted form data in the page address field.
- **When to use POST??**

 POST has no size limitations, and can be used to send large amounts of data.

 Form submissions with POST cannot be bookmarked

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

<p>This form will be submitted using the POST method:</p>

<form method="POST“ target="_blank" >
  First name:<br>
  <input type="text" name="firstname" value="Mickey">
  <br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
  <br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

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

- **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.

1. Single-line text input controls

```
<!DOCTYPE html>
 <html>
<head>
<title>Text Input Control</title>
 </head>
 <body>
 <form > First name: <input type = "text" name = "first_name"
   /> <br>
 Last name: <input type = "text" name = "last_name" />
 </form>
 </body> </html>
```

1. Single-line text input controls

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.

2. Password input controls

```
<!DOCTYPE html>
<html>
 <head>
 <title>Password Input Control</title>
</head>
 <body>
 <form >
User ID : <input type = "text" name = "user_id" /> <br>
Password: <input type = "password" name = "password" />
</form>
 </body> </html>
```

2. Password input controls

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

3. Multiple-Line Text Input Controls

```
<!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>
```

3. Multiple-Line Text Input Controls

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.**.

```
<!DOCTYPE html>
 <html>
 <head>
 <title>Checkbox Control</title>
 </head>
 <body>
 <form>
 <input type = "checkbox" name = "maths" value = "on"> Maths
 <input type = "checkbox" name = "physics" value = "on"> Physics
   </form>
</body> </html>
```

Checkbox Control

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.

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**.

```
<!DOCTYPE html>
 <html>
<head>
<title>Radio Box Control</title
</head>
 <body>
<form> <input type = "radio" name = "subject" value =
   "maths"> Maths
 <input type = "radio" name = "subject" value = "physics">
   Physics
 </form> </body> </html>
```

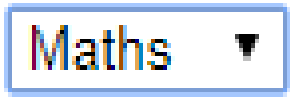
Radio Button Control

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.

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.

```
<!DOCTYPE html>
 <html>
 <head>
<title>Select Box Control</title>
</head>
 <body>
 <form>
<select name = "dropdown">
 <option value = "Maths" selected>Maths</option>
<option value = "Physics">Physics</option> </select>
</form> </body> </html>
```



Select Box Control

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.

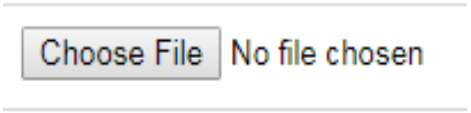
Select Box Control

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

File Upload Box

- If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to **file**.

```
<!DOCTYPE html>
 <html>
 <head> <title>File Upload Box</title> </head>
<body>
 <form>
<input type = "file" name = "fileupload" accept = "image/*" />
</form>
 </body> </html>
```



File Upload Box

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	**accept** Specifies the types of files that the server accepts.

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**.

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.

Button Controls

```
<!DOCTYPE html>
 <html>
<head>  </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 =
   "/html/images/logo.png" />
 </form>
 </body> </html>
```

The Target Attribute

- The target attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.
- The default value is "_self" which means the form will be submitted in the current window.
- To make the form result open in a new browser tab, use the value "_blank":

The Target Attribute

```
<!DOCTYPE html>
<html> <body>
<p>When submitting this form, the result will be opened in a new browser
   tab:</p>

<form target="_blank">
  First name:<br>
  <input type="text" name="firstname" value="Mickey">
  <br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
  <br><br>
  <input type="submit" value="Submit">
</form>

</body> </html>
```

HTML 5 Features

New Input Types

- HTML5 introduces several new <input> types to improve the user experience and to make the forms more interactive
- **color** input type :allows the user to select a color from a drop-down color picker and returns the hex value for that color

```
<form>
 <label>
Select Color: <input type="color" name="mycolor">
</label>
</form>
```

New Input Types

- **date** input type : allows the user to select a date from a drop-down calendar.
- **datetime** input type : allows the user to select a date and time along with time zone.
- **email** input type : allows the user to enter e-mail address.
- **month** input type: allows the user to select a month and year from a drop-down calendar.
- **number** input type : used for entering a numerical value. You can also restrict the user to enter only acceptable values using the additional attributes min, max, and step.

New Input Types

- **search** input type **:** used for creating search fields
- **tel** input type : used for entering a telephone number.
- **time** input type : used for entering a time.
- **url** input type : can be used for entering web addresses
- **week** input type : allows the user to select a week and year from a drop-down calendar.