HTML

Scan or click here for more resources

HTML stands for **H**yper **T**ext **M**arkup **L**anguage.

H yper
T ext
M arkup
L anguage

LET'S BREAK THIS

HYPER - It simply means that when you are on the internet using a browser such as Netscape Navigator or Internet Explorer, you can in fact, go "all over the place". HTML does not hold to that and allows you to jump to any page on the WWW and at any time.

TEXT- We are working with text only files.

MARKUP- "Markup" comes from the fact that in order to create web pages, we will be typing in the text and then "marking up" the text.

LANGUAGE- "Language" means that we are using a language with all its syntax. Note that HTML is not a programming language such as BASIC or FORTRAN. These are linear programming languages and are based on a whole different set of rules and are far more complicated to learn. So HTML is the basic language for creating web pages on a website and, as you will see, is also an easy language to learn.

In HTML, you must put in the tags yourself. If you want to see the tags for this page, just choose **VIEW** from the menu bar of your browser and then choose **SOURCE** or **DOCUMENT SOURCE**.

In other words we can say that “HTML is a Hyper Text Markup Language which is used to write HTML documents. Once you can create an HTML document it can be viewed in any browser. There Once you create your web pages, you can upload these pages to your site using any FTP software.

DIFFERENT VERSIONS OF HTML

HTML 1.0

The original version of HTML was HTML 1.0. The **first version of HTML** was written by Tim Berners-Lee in 1993. It had very limited features which greatly limited what you could do in designing your web pages.

HTML 2.0

HTML 2.0 then arrived and included all the features of HTML 1.0 plus several new features for web page design. Until January, 1997, HTML 2.0 was the standard in web page design.

HTML 3.0

Netscape, the leading browser at that time, introduced new tags and attributes called the **Netscape Extension Tags**. but Netscape did not fully specify their new tags and so these extension tags did not work in most other browsers.

Dave Raggett, introduced the HTML 3.0 draft which included many new and useful enhancements to HTML. The phrase "HTML 3.0 enhanced" quickly became popular on the web but it more often than not referred to documents containing browser specific tags. HTML 3.0 did not make it was because it was so "big".

HTML 3.2 (WILBUR)

The Word Wide Web Consortium (W3C), founded in 1994 to develop common standards for the evolution of the World Wide Web, drafted the ***WILBUR*** standard, which later became known as ***HTML 3.2***. HTML 3.2 early 1996 and became the official standard in January, 1997.

HTML 4.0 (COUGAR)

In the early days, ***HTML 4.0*** was code-named ***COUGAR***. This version introduces new functionality, most of which comes from the expired HTML 3.0. This version became a recommendation in December, 1997 and a standard as of April, 1998. Explorer has done a very good job in implementing the many features of HTML 4.0.

The most widely used **version** throughout the 2000's was **HTML** 4.01, which became an official standard in December 1999.

XHTML

You would think that the next major version after HTML 4.0 would be HTML 5.0 The next version of HTML after HTML 4 is **XHTML**. XHTML, was a rewrite of **HTML** as an XML language.

XHTML stands for E**X**tensible **H**yper **T**ext **M**arkup **L**anguage.

XHTML is the result of the hard working World Wide Web Consortium (the W3C) to bring some sort of **standard** to provide rich high quality web pages through these varied devices. **XHTML** became an **official** W3C **recommendation** in **January, 2000**. **XHTML** is now a **web standard** and is the next generation of HTML.

HTML 5

HTML 5 is the new web standard. It follows HTML 4 and XHTML. HTML5 is the result of cooperation that began in 2006 between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG). While HTML5 is still evolving , the latest browsers do support many of the new features and elements in this version. The basic aim of HTML5 is to provide two things - (1) to improve the language and (2) to support the latest multimedia

WHAT DO I NEED TO BEGIN DESIGNING A HOME PAGE?

YOU NEED A FOLDER (also called a DIRECTORY)
YOU DO NOT NEED TO BE CONNECTED TO THE INTERNET
YOU NEED A BROWSER

YOU NEED A WORD PROCESSOR - you basically:

1. create your web page in a text editor such as NotePad (typing in all the text and tags)
2. save your web page as an HTML file using any appropriate name
3. load the HTML file into the browser to see how your web page looks and works
4. switch back to NotePad to make any corrections, changes, etc.

NAMING YOUR WEB PAGE

When you save your web page for the first time, you need to give it a name. You not only have to give it a name but you also need to add an extension to the name.

THE NAME: If you are **not** running at least Windows 95, your file name is limited to a **maximum of 8 characters**.

THE SUFFIX: The suffix is an **extension** to the name and declares the kind of document that it is. In HTML, the suffix is either "**.htm**" or "**.html**". "Htm" or "html" tells the browser you are working with HTML files - that is, an HTML document.

HTML TAGS

Tag or tag element refers to the HTML codes that define the element in an HTML file, such as headings, images, paragraphs and list. There are two kinds of tags i.e. the container tag and empty tag. HTML tags are inserted into a document between < and > symbols. Tags are not case-sensitive.

Container Tag : These tag which bracket or contain text or other tag elements are called container tags. These consists of two tags , a start tag and an end tag which enclose the text they affect.

Empty Tag : These are standalone are do not bracket or contain text or any other tag.

Attribute : Allows you to specify how web browsers should treat a particular tag. An attribute is included within the actual tag, either within a start tag or an empty tag. End tags should not contain attributes.

Syntax : Attribute = “value”

e.g. align = “center”

HTML
Hyper Text Markup Language

- Documents
  - – Document = page = HTM file = topic
  - – Content (text, images)
  - – Tags (display commands)
- Other terms
  - – Window: browser display window
  - – URL: Uniform Resource Locator
  - – Hyperlink: hypertext jump to a resource
  - – Resource: URL, image, mailto, external file

HTML
Hyper Text Markup Language

- Documents
  - – Document = page = HTM file = topic
  - – Content (text, images)
  - – Tags (display commands)
- Other terms
  - – Window: browser display window
  - – URL: Uniform Resource Locator
  - – Hyperlink: hypertext jump to a resource
  - – Resource: URL, image, mailto, external file

HTML

HTML pages are tag-based documents

- Really plain ASCII text files
- Don't look like documents they represent
- Tags indicate how processing program should display text and graphics
- Processed by browsers “on the fly”
- Tags usually appear in pairs
- Most have reasonable names
- Most can be modified by attributes/values
- Clear text, case insensitive
- Ignores white space
- Comprised of tags   <tag />
- Open tags and closed tags

Starting Your Page Document Tags

All HTML files should include at least these tags :

- HTML Tag
- Head Tag
- Title Tag
- Body Tag

- HTML documents are text documents that contain:
  - formatting instructions, called **tags**
  - the text that is to be displayed on a Web page
  - This tag defines the top-most element, identifying it as an HTML document.

Basic HTML Syntax

1. The HTML Tag : HTML is container tag that has a start and an end tag and all the other tags and texts are nested within it. All HTML documents begin with `<html>` and end with `</html>`

```
Syntax :        <HTML>
                 ........
                </HTML>
```

2. The HEAD Tag : `<head>` tag contains information that is used by the Web browser, and you place it at the start of an HTML document, after the opening `<html>` tag.

- This tag contains information about your HTML file. It may also contain other tags that helps you identify your HTML file to the outside world. The Head Tag is nested within the tag.

- The only tag contained within the head tag is the title tag.

Syntax :

```
<HTML>
<HEAD>
………..
</HEAD>
</HTML>
```

3. **The TITLE Tag :** This tag is nested within the Head Tag. The tag output is displayed on your browser’s title bar but does not appear as part of the page. This tag is a required element that you should include in each and every HTML document.

Syntax :

```
<HTML>
<HEAD>
<TITLE>
My First Web Page
          </TITLE>
          </HEAD>
          </HTML>
```

4. **The BODY Tag :** Following the document head is the **<body>** tag, which contains the document body. This tag is the compliment of the Head tag and contains all of the tags, or elements that a browser actually displays as the body of your HTML document. Both the head tag and the Body tag are nested

within the HTML Tag. BODY tag comes after the head tag.

Syntax :

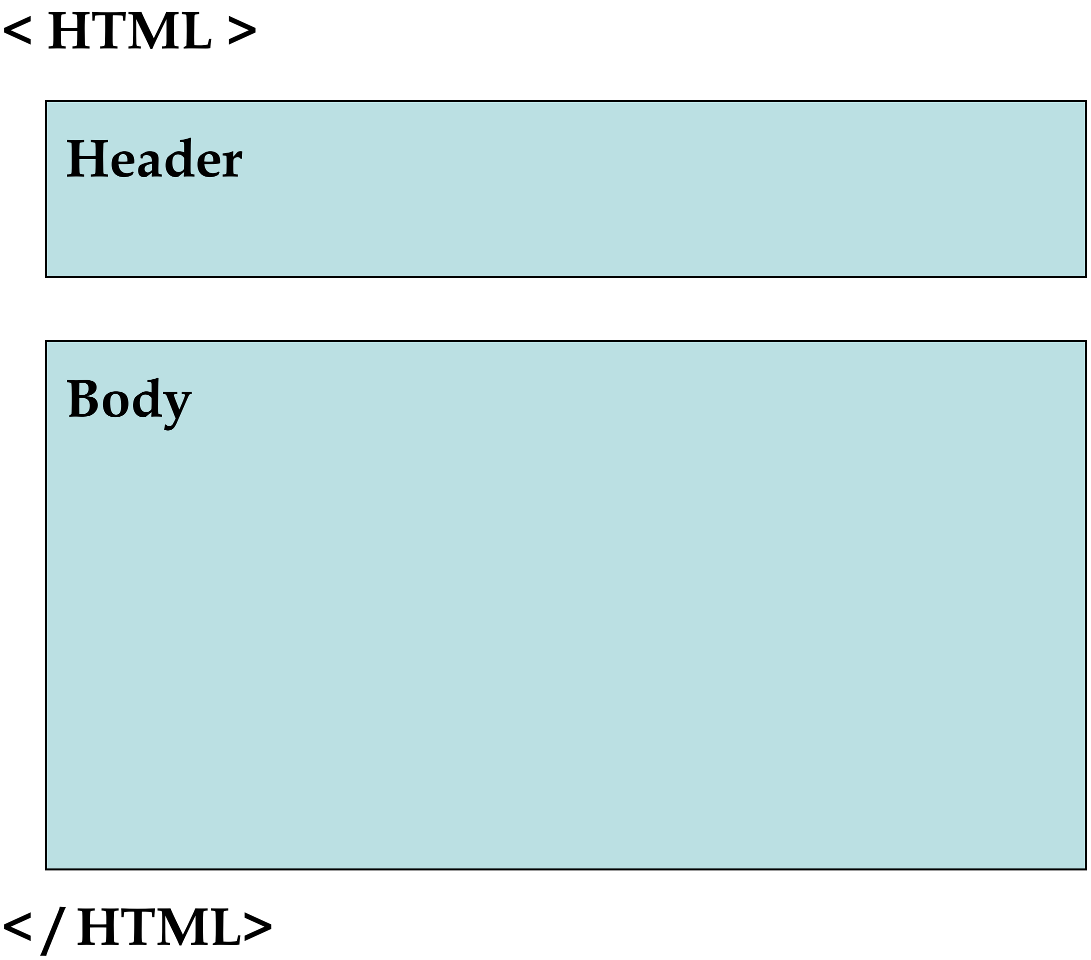
```
<HTML>
<HEAD>
<TITLE>
My First Web Page
        </TITLE>
        </HEAD>
        <BODY>
        ………..
        </BODY>
        </HTML>
```

Note:- 1. The **<body>** tag pair and the text and tags it contains are referred to as the **document body.**

2. A Web browser's process of assembling and formatting an HTML document is called **parsing** or **rendering.**

HTML – Fundamentals

Document Structure



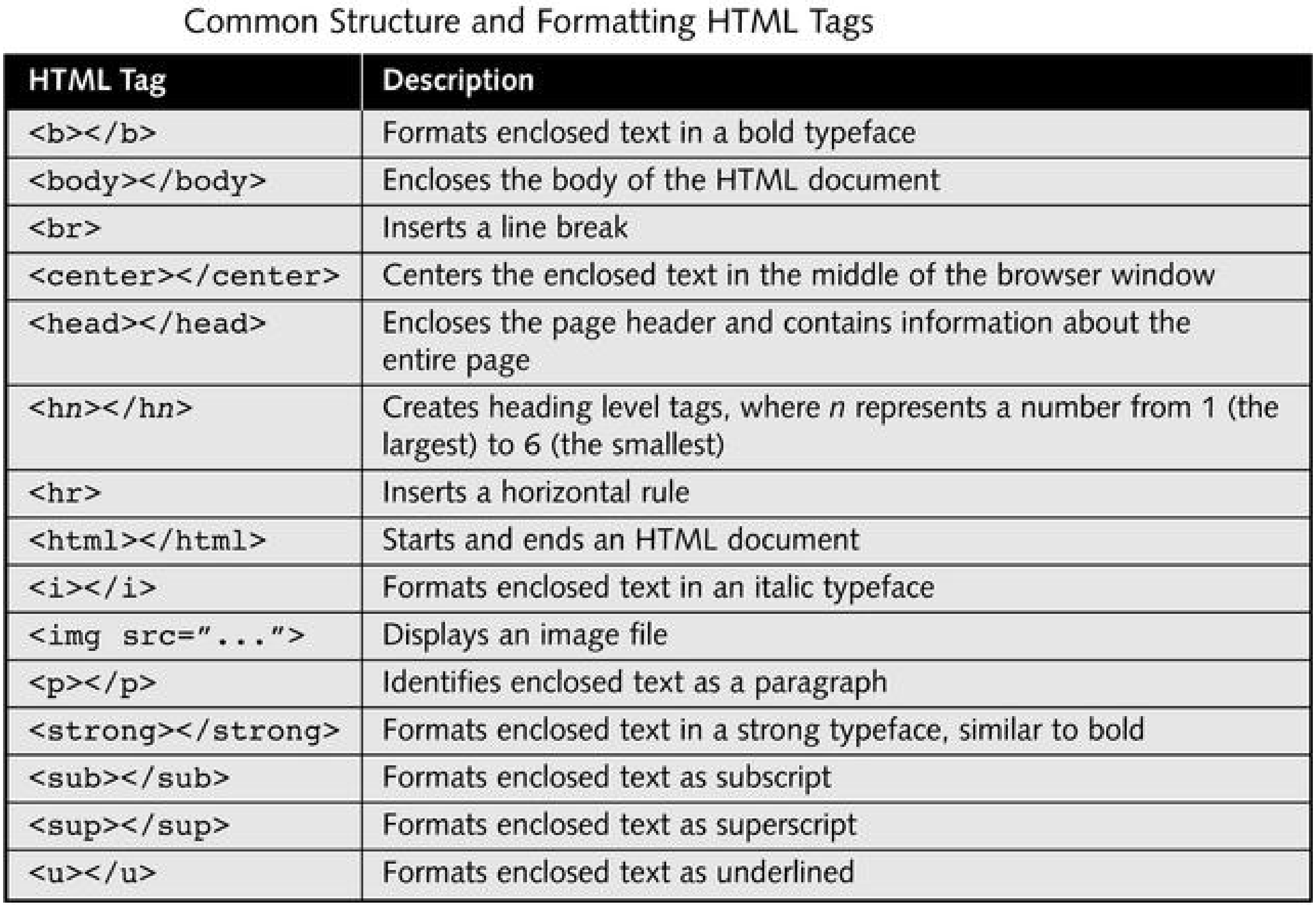
< HTML >

Header

Body

< / HTML>

Common Structure and Formatting HTML Tags

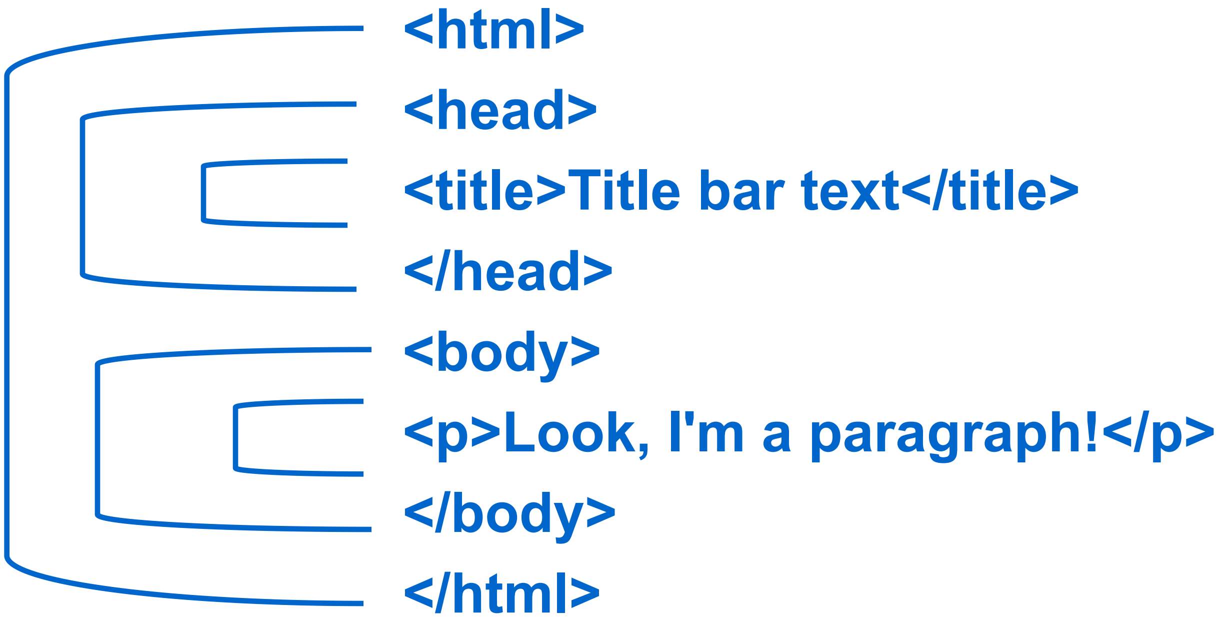


Common Structure and Formatting HTML Tags

HTML Tag	Description
<b></b>	Formats enclosed text in a bold typeface
<body></body>	Encloses the body of the HTML document
<br>	Inserts a line break
<center></center>	Centers the enclosed text in the middle of the browser window
<head></head>	Encloses the page header and contains information about the entire page
<h*n*></h*n*>	Creates heading level tags, where *n* represents a number from 1 (the largest) to 6 (the smallest)
<hr>	Inserts a horizontal rule
<html></html>	Starts and ends an HTML document
<i></i>	Formats enclosed text in an italic typeface
<img src="...">	Displays an image file
<p></p>	Identifies enclosed text as a paragraph
<strong></strong>	Formats enclosed text in a strong typeface, similar to bold
	Formats enclosed text as subscript
	Formats enclosed text as superscript
<u></u>	Formats enclosed text as underlined

Basic Structure of HTML document

Example of basic tag positioning



```
<html>
<head>
<title>Title bar text</title>
</head>
<body>
<p>Look, I'm a paragraph!</p>
</body>
</html>
```

Formatting Elements

The various elements that are used formatting blocks of text within an HTML document. Formatting elements should all be located within the BODY element of the document.

1. **FONT Element**

   This element requires the following attributes :

SIZE – The SIZE attribute is represented as <FONT SIZE=value>. The valid value ranges from 1-7 and the default size is 3.

e.g. <FONT SIZE =5> change the font to 5 </FONT>
      <FONT SIZE =+2> change the font to<BASEFONT SIZE…>+2</FONT>

COLOR – The COLOR attribute changes the color of the text.

e.g. <FONT COLOR=red> This text is also red</FONT>

FACE – This attribute sets the typeface that will be used to display the text on the screen provided the typeface is already installed on the user's machine.

e.g. <FONT FACE=“ Times Roman”>
      </FONT>

2. HEADING ELEMENTS

HTML defines six levels of headings. The headings range from <H1> to <H6>

e.g. <H1>This is heading</H1>

3. **CENTRE Element**
All lines of text between the beginning and end of <CENTRE> elements are centered between the current left and right margins.

```
Syntax-  <CENTRE>any text
         </CENTRE>
```

4. **COMMENT Element –** This is a comment entry and it should not be displayed on the screen.

```
Syntax- <COMMENT>
          ……………
          </COMMENT>
```

5. **PARAGRAPH Element –** This element indicates a paragraph. The text between <P> and </P> elements is surrounded by a vertical space of one line or half a line.

6. Pre-formatted text Element – This elements blocks of text in fixed-width font. The <PRE> element may be used with the optional WIDTH attribute. The WIDTH attribute specifies the maximum number of characters for a line and allows the browser to select a suitable font and indention. If there is no WIDTH attribute, a width of 80 characters is assumed. If the WIDTH attribute is specified, the width should be specified as 40, 80 and 132 characters.

```
e.g. <PRE WIDTH="80">
This is an example code.
       </PRE>
```

7. HORIZONTAL Element

The <HR> element is a divider between sections of text such as a full width horizontal rule. It accepts the following attributes-

SIZE- The Size attributes determines the thickness of the horizontal rule.

WIDTH– With the WIDTH attribute specify an exact width in pixels.

ALIGN– The alignment of the rule can be set. It can be left, right or center.

NOSHADE– If a solid bar is required, the NOSHADE attribute specifies that the horizontal rule should not be shaded at all.

COLOR Attribute– The color of the horizontal rule can be set.

e.g.

```
<HR SIZE=5 WIDTH=200 ALIGN=LEFT>
<HR SIZE=5 WIDTH=200 ALIGN=RIGHT>
<HR SIZE=5 WIDTH=200 ALIGN=CENTER NOSHADE>
```

8. **MARQUEE Element –** The <MARQUEE> element allows to create a scrolling text marquee. It highlights the scrolling text. Marquee can be left or right aligned and have a number of attributes, some of them –

ALIGN, BEHAVIOR, BGCOLOR, DIRECTION, HEIGHT, WIDTH, HSPACE, LOOP etc.

e.g. <MARQUEE>This text will scroll from left to right slowly</MARQUEE>

<MARQUEE HEIGHT=50% WIDTH=20%>This marquee is half the height of the screen and 20% of the screen width</MARQUEE>

<BR>, <P> and <HR> TAGS

<BR>

<BR> tells your browser to go to the beginning of the next line. BR stands for line BReak. <BR> acts in the same way as the ENTER key on your keyboard.

<P>

<P> for **P**aragraph tells your browser to insert a blank or empty line and then begin a new line (a new paragraph). <BR> tells the browser when a line has ended while <P> tells the browser to leave a blank line and begin a new paragraph.

<HR>

<HR> puts a line across the page. **HR** stands for **H**orizontal **R**ule. The two lines you see below were put there with <HR> tags.

HTML – Fundamentals
Basic Structure

```
<html>
    <head>
        <title> The title of your html page </title>
    </head>
    <body>

        <! - - your web page content and markup  - ->

    </body>
</html>
```

HTML - Fundamentals

```
header
<body>

    Hello world

</body>
```

HTML - Fundamentals

```
header
<body>


    <b>Hello</b><br>
    World<br>



</body>
```

HTML Links – Hyperlinks

Links are found in nearly all web pages. Links allow users to click their way from page to page. HTML links are hyperlinks. 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.

You use links to:

1. Jump from section to section within the same web page (also called Page Jump).
2. Link to a different page within your own website (such as my linking this lesson to the next lesson or to my home page).
3. Link to another web page or website anywhere in the world.

To create a link, you wrap <a> and </a> tags around the content you want to use for the link, and supply the URL to link to in the <a> tag's href attribute.

HTML Links - Syntax

Here's how to create some text that links to www.example.com:

<a href="http://www.example.com/">Visit this great website!</a>

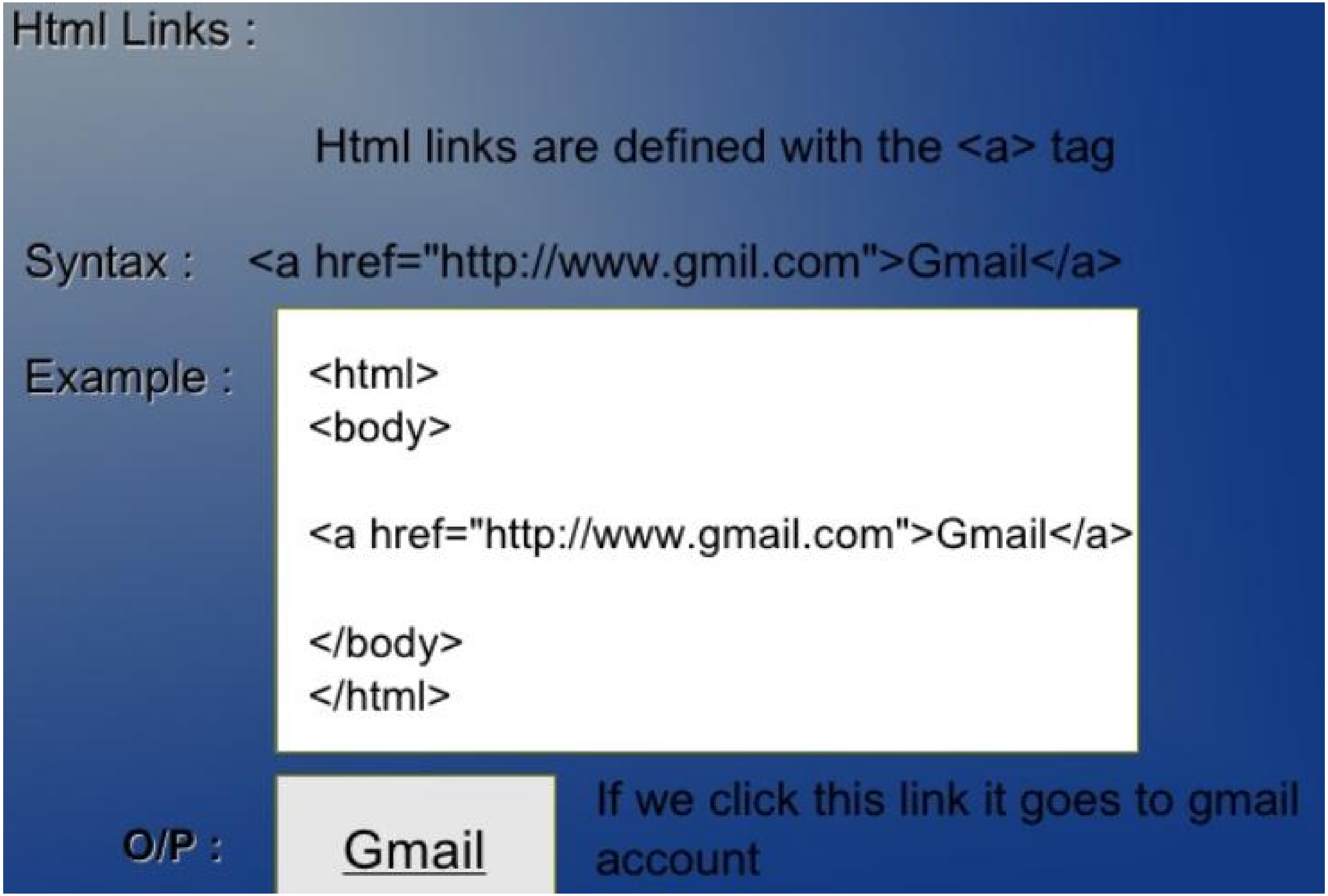
<a href="*url*">*link text*</a>

ANCHORS (Hypertext Link)

<A href=“url” attributes>Displayed text
</A>

Attributes

- NAME = “text”
- TITLE = "text"
- TARGET = “frame_name|window_name”



Html Links :

Html links are defined with the <a> tag

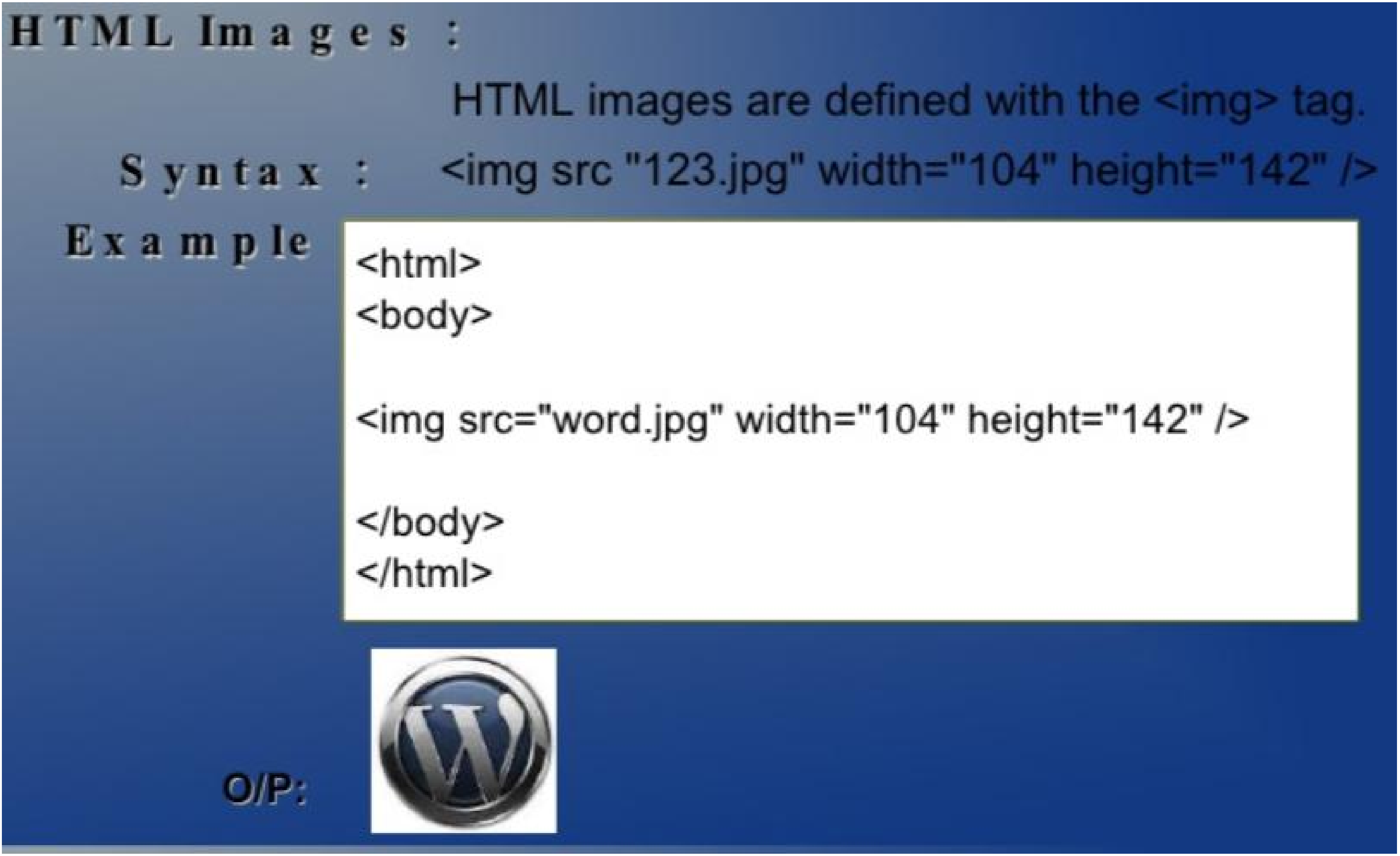
Syntax : <a href="http://www.gmil.com">Gmail</a>

Example :

```
<html>
<body>

<a href="http://www.gmail.com">Gmail</a>

</body>
</html>
```

O/P :

If we click this link it goes to gmail account



HTML Images :

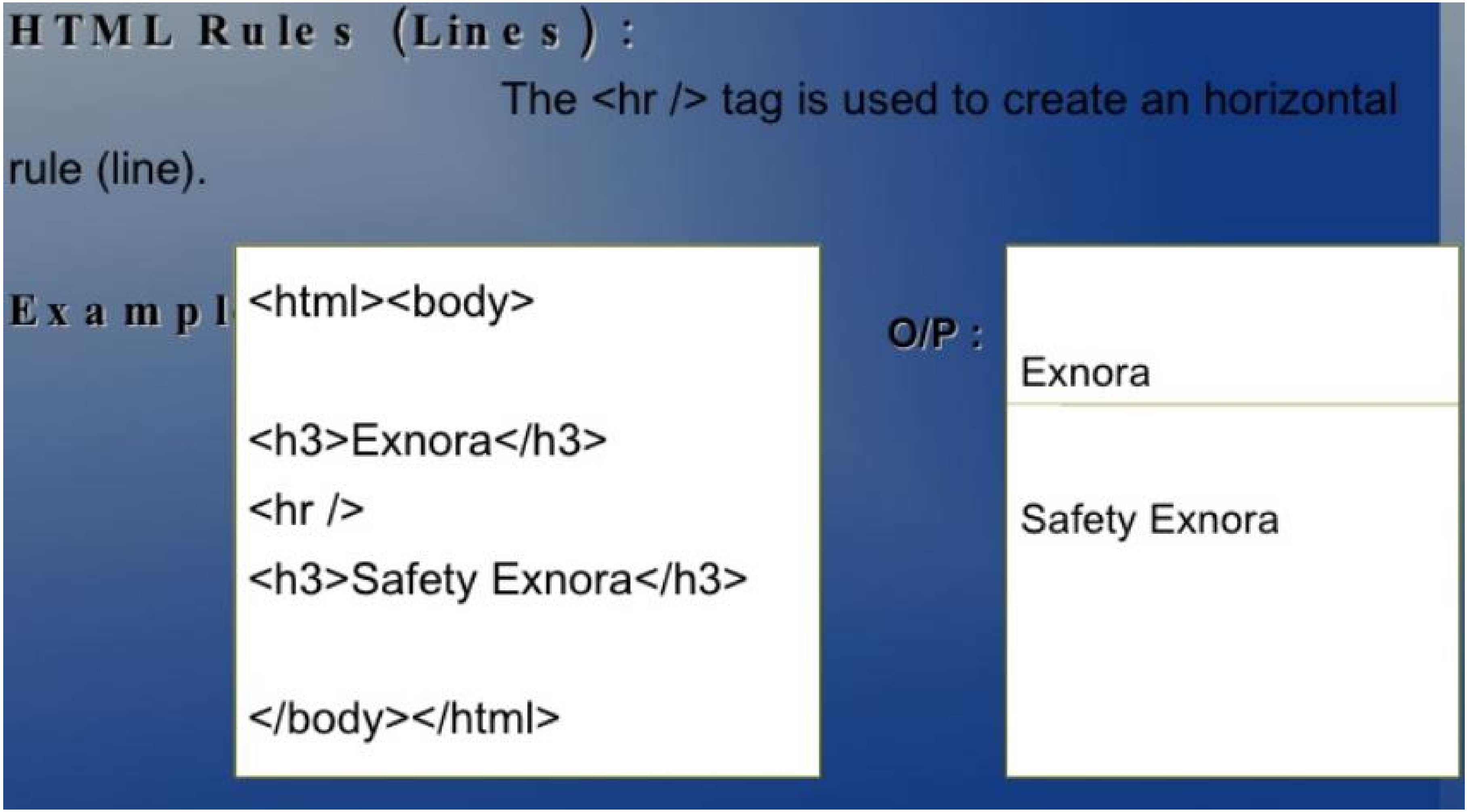
HTML images are defined with the <img> tag.

Syntax : <img src "123.jpg" width="104" height="142" />

Example

```
<html>
<body>

<img src="word.jpg" width="104" height="142" />

</body>
</html>
```

O/P:



HTML Rules (Lines) :

The <hr /> tag is used to create an horizontal rule (line).

Exampl

```
<html><body>


<h3>Exnora</h3>
<hr />
<h3>Safety Exnora</h3>


</body></html>
```

O/P :

```

Exnora


Safety Exnora
```

HTML Comments :

Comments can be inserted in the HTML code to make more readable and understandable. Comments are ignored by the browse and are not displayed.

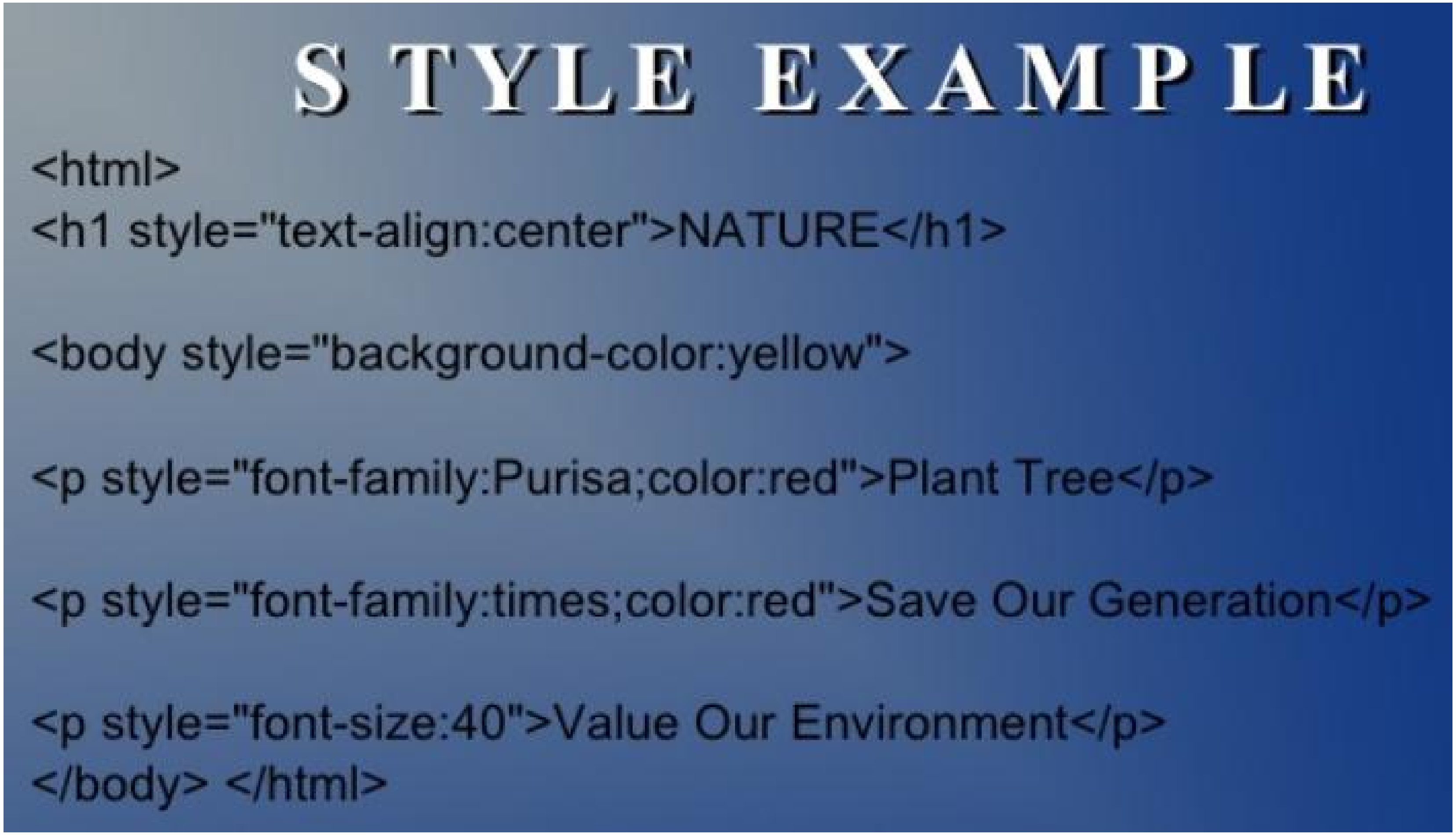
Syntax : `<!-- some text →`

Example

```
<html><body>


<!--It will not be displayed-->
<h3>Plant Trees </h3>


</body></html>
```

O/P :

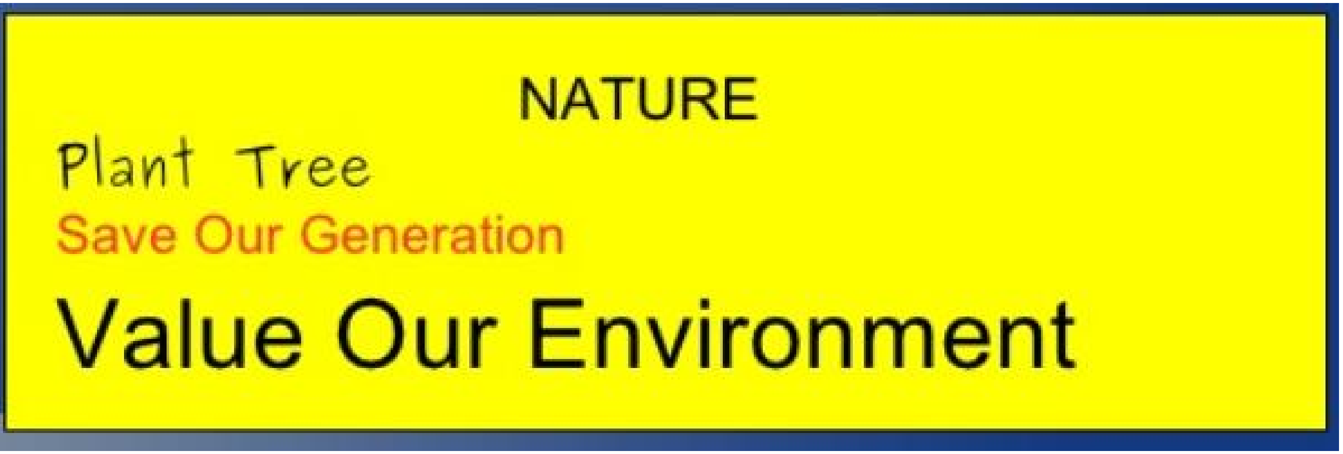
```
Plant Trees
```



STYLE EXAMPLE

```
<html>
<h1 style="text-align:center">NATURE</h1>

<body style="background-color:yellow">

<p style="font-family:Purisa;color:red">Plant Tree</p>

<p style="font-family:times;color:red">Save Our Generation</p>

<p style="font-size:40">Value Our Environment</p>
</body> </html>
```

O/P :



NATURE

Plant Tree

Save Our Generation

Value Our Environment

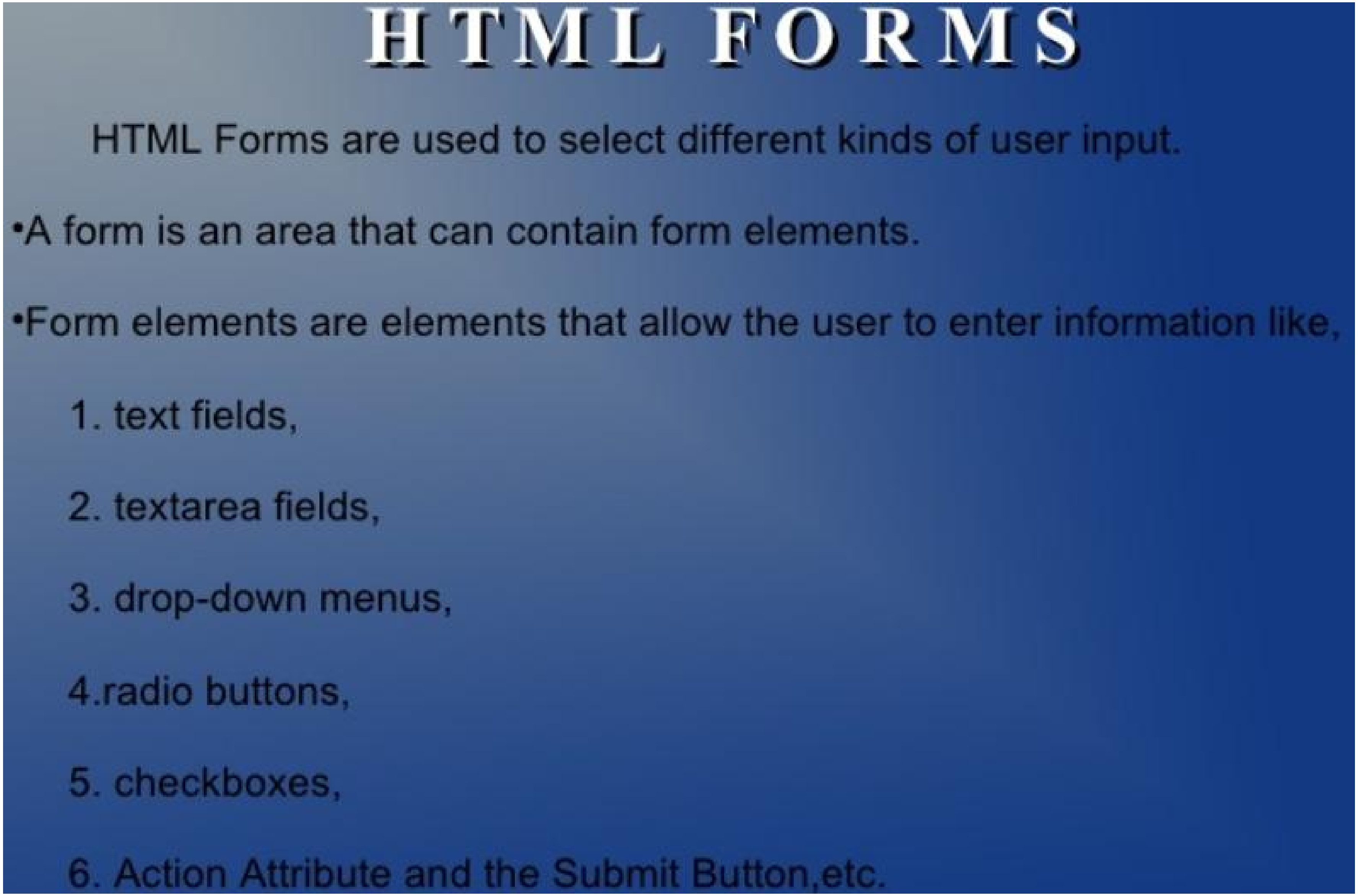
HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

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. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

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>
```



HTML FORMS

HTML Forms are used to select different kinds of user input.

- A form is an area that can contain form elements.
- Form elements are elements that allow the user to enter information like,

1. text fields,
2. textarea fields,
3. drop-down menus,

4.radio buttons,

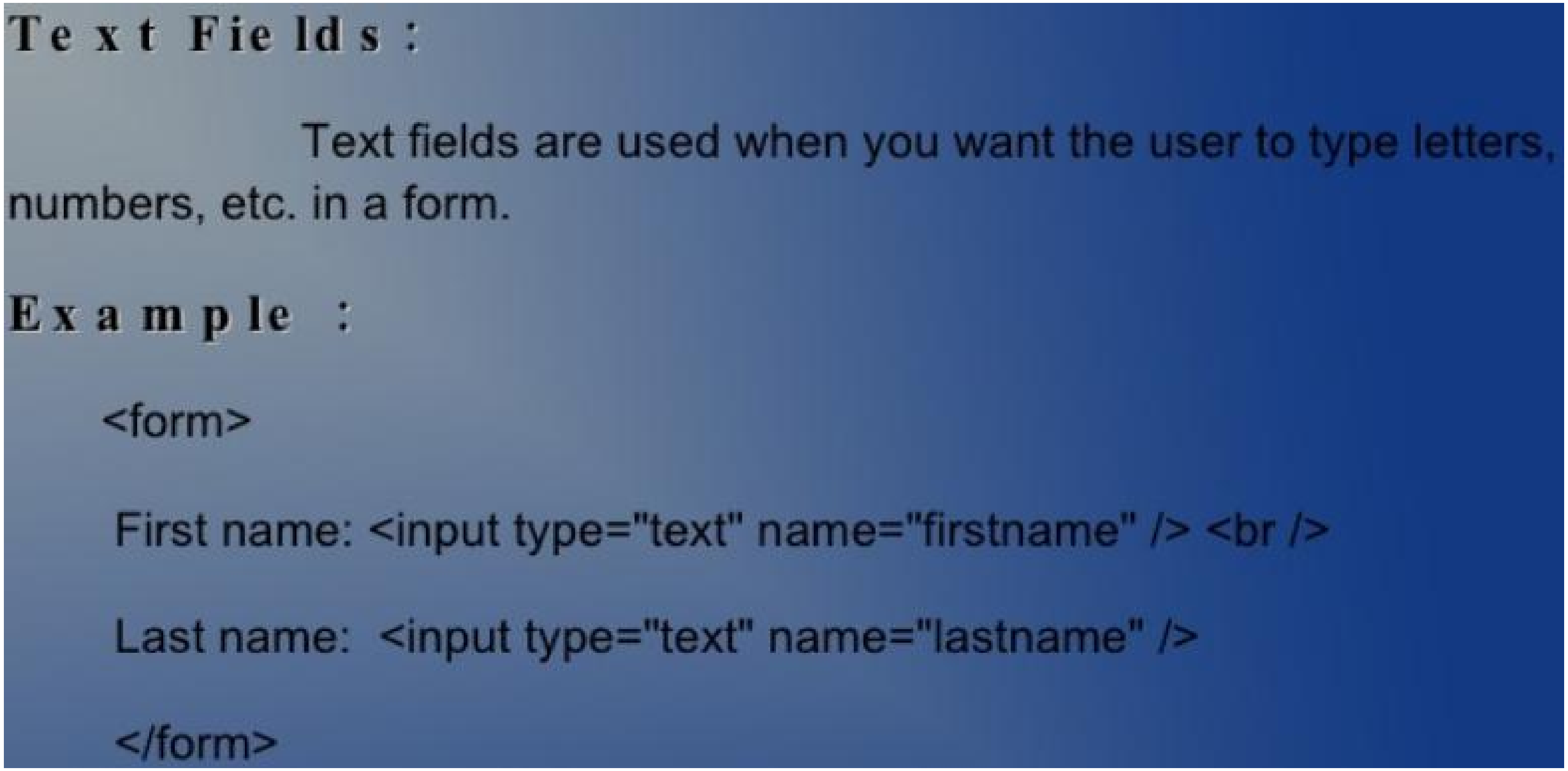
5. checkboxes,
6. Action Attribute and the Submit Button,etc.

The <input> Element

The HTML `<input>` element is the most used form element. An `<input>` element can be displayed in many ways, depending on the `type` attribute.

Here are some examples:

Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

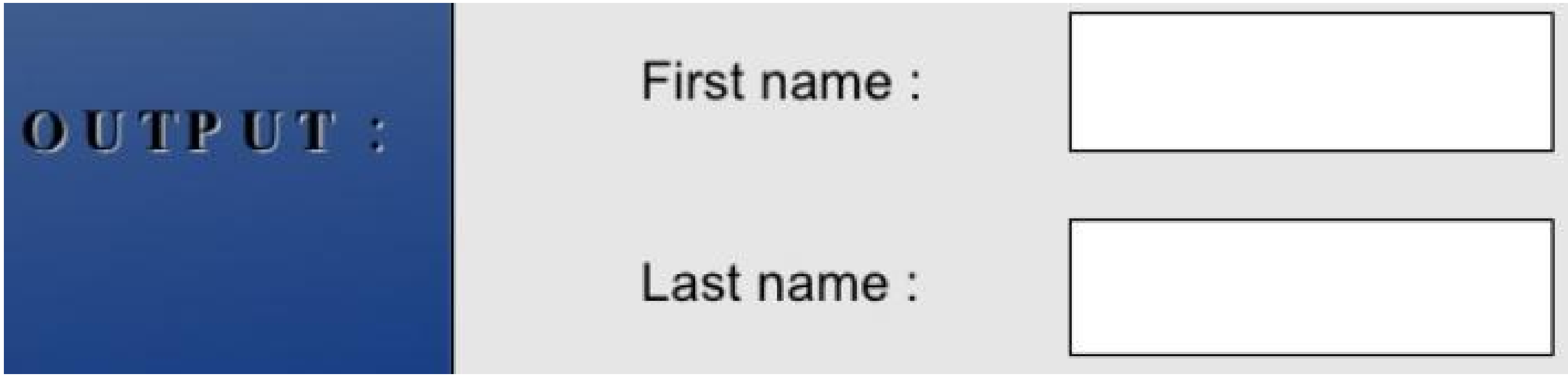


Text Fields :

Text fields are used when you want the user to type letters, numbers, etc. in a form.

Example :

```
<form>
 First name: <input type="text" name="firstname" /> <br />
 Last name:  <input type="text" name="lastname" />
 </form>
```



OUTPUT :

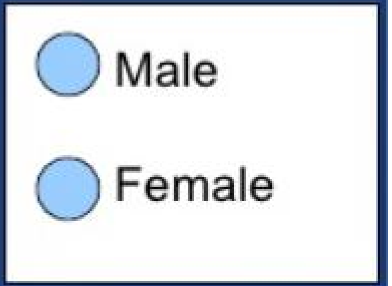
First name :

Last name :

RADIO & CHECK BOX

Radio Buttons :

```
<form>

<input type="radio" name="sex" value="male" /> Male <br />

<input type="radio" name="sex" value="female" /> Female

</form>
```

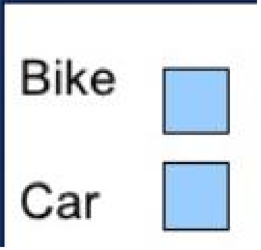


○ Male

○ Female

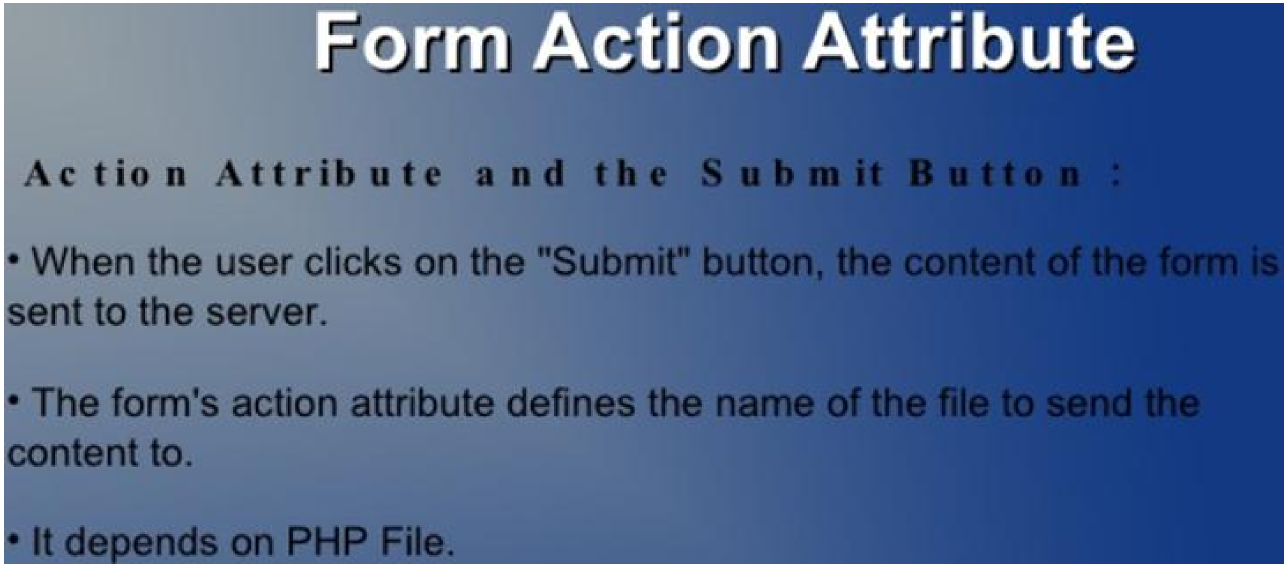
Checkboxes :

```
<form>

Bike: <input type="checkbox" name="vehicle" value="Bike"/> <br />

Car: <input type="checkbox" name="vehicle" value="Car"/><br />

</form>
```



Bike □

Car □



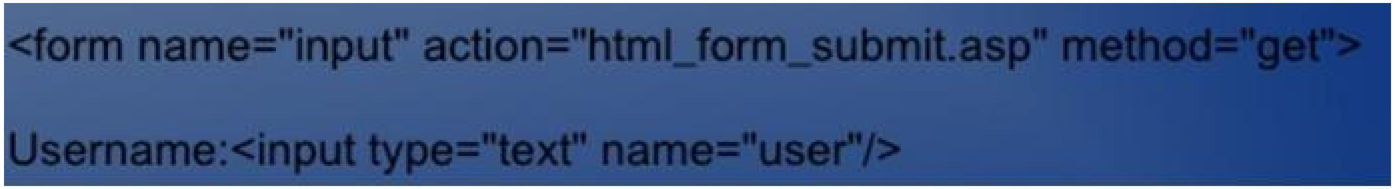
Form Action Attribute

Action Attribute and the Submit Button :

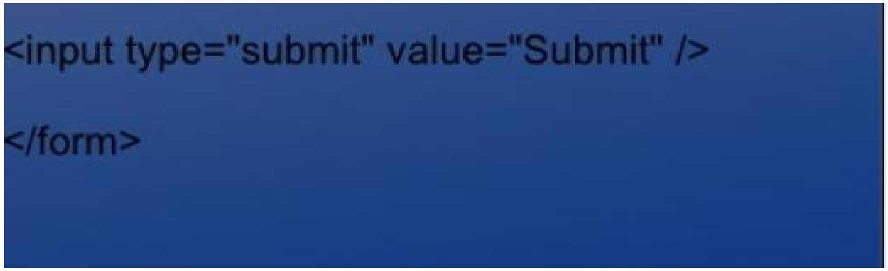
- When the user clicks on the "Submit" button, the content of the form is sent to the server.
- The form's action attribute defines the name of the file to send the content to.
- It depends on PHP File.

The Submit Button

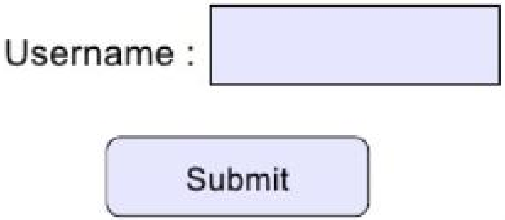
The `<input type="submit">` defines a button for submitting the form data to a form-handler.
The form-handler is typically a file on the server with a script for processing input data.
The form-handler is specified in the form's `action` attribute.



```
<form name="input" action="html_form_submit.asp" method="get">

Username:<input type="text" name="user"/>
```



```
<input type="submit" value="Submit" />

</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.
4	**enctype** You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are − **application/x-www-form-urlencoded** − This is the standard method most forms use in simple scenarios. **mutlipart/form-data** − This is used when you want to upload binary data in the form of files like image, word file 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

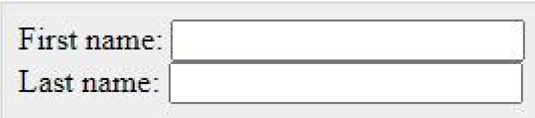
This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name −

```
<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>
```

This will produce the following result −



First name:

Last name:

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.

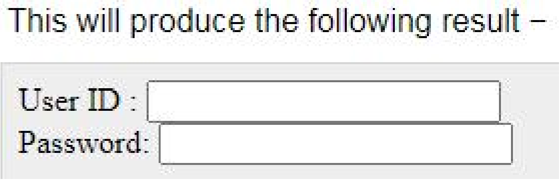
Password input controls

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**.

Example

Here is a basic example of a single-line password input used to take user password −

```
<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>
```



This will produce the following result −

User ID :

Password:

Attributes

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

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.

Multiple-Line Text Input Controls

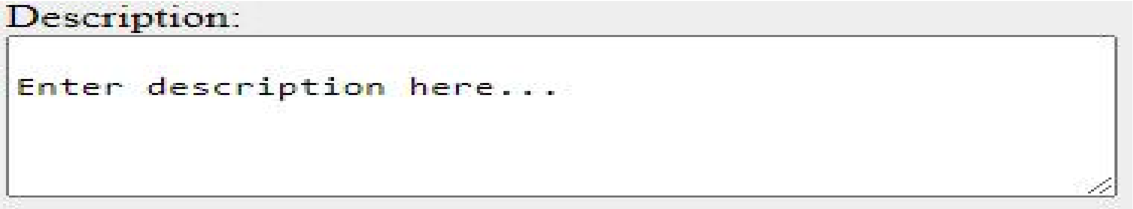
This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description −

```
<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>
```

This will produce the following result −



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

Select Box Control(Drop down box)

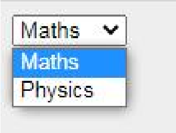
A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```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>
```

This will produce the following result –



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.

Attributes

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

File Upload Box

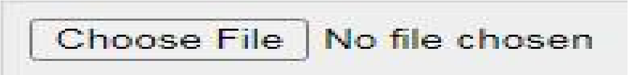
If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to **file**.

Example

Here is example HTML code for a form with one file upload box −

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

This will produce the following result −



Attributes

Following is the list of important attributes of file upload box −

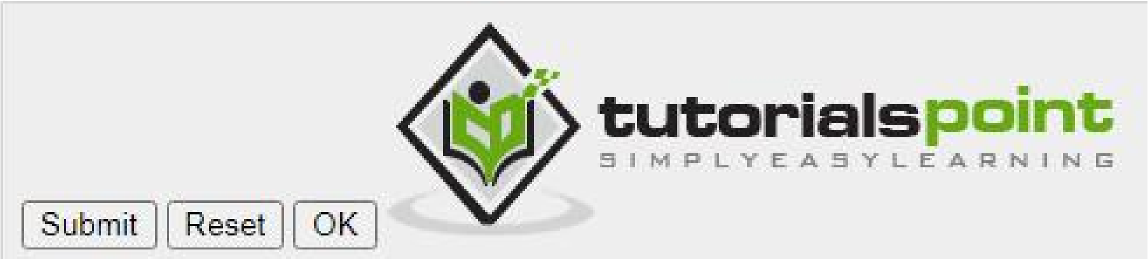
Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**accept** Specifies the types of files that the server accepts.

Example

Here is example HTML code for a form with three types of buttons –

```html
<html>
<head>
<title>File Upload Box</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 =
"/html/images/logo.png" />
</form>
</body>
</html>
```

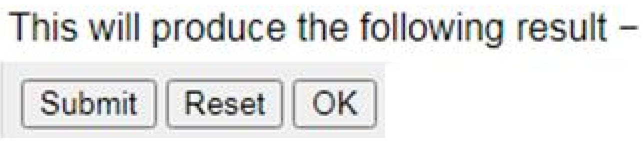
This will produce the following result –



Example

Here is example HTML code for a form with three types of buttons −

```
<html>
<head>
<title>File Upload Box</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" />
</form>
</body>
</html>
```



This will produce the following result −

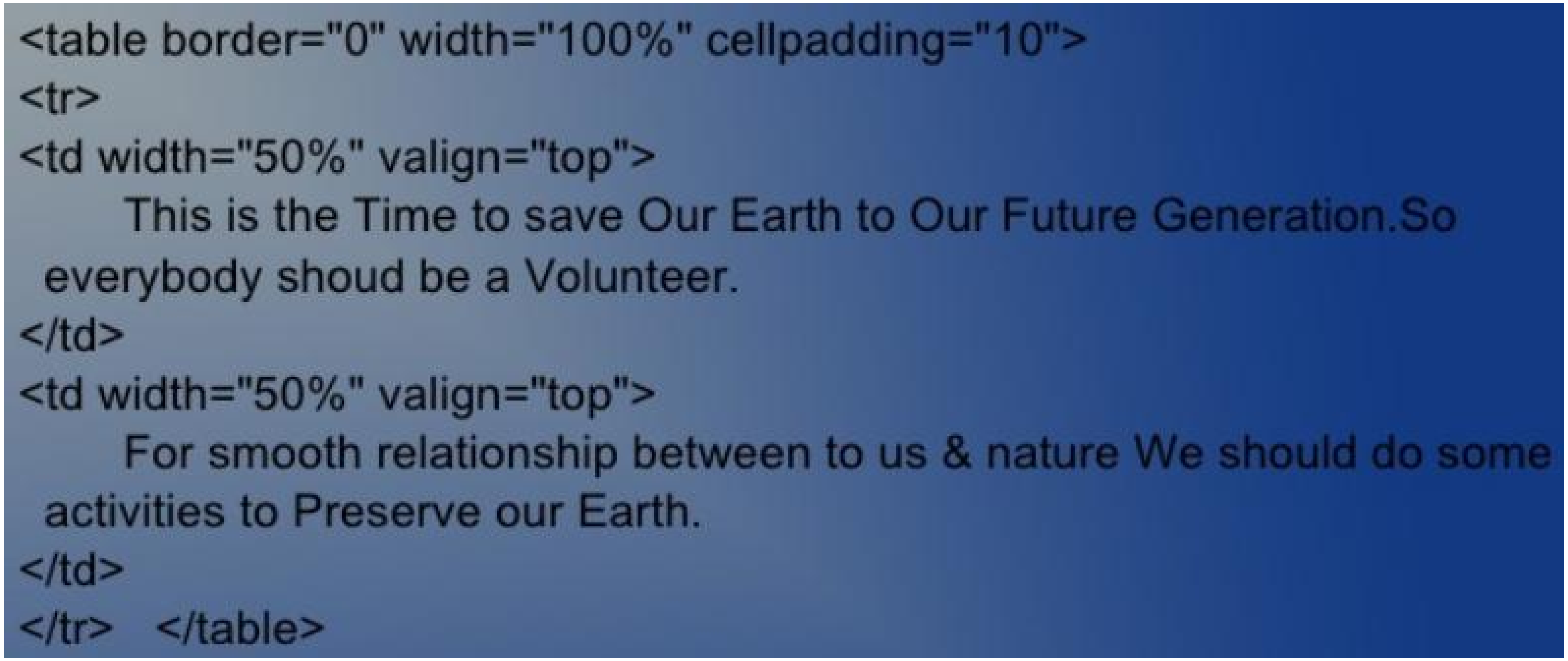
Button Controls

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 −

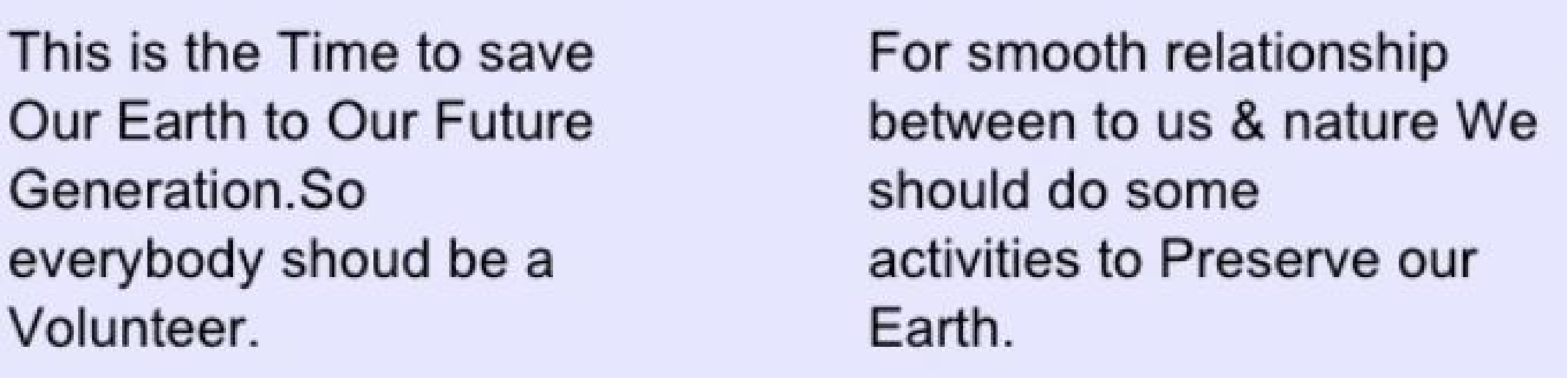
Sr.No	Type & Description
1	**submit** This creates a button that automatically submits a form.
2	**reset** This creates a button that automatically resets form controls to their initial values.
3	**button** This creates a button that is used to trigger a client-side script when the user clicks that button.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

HTML LAYOUT

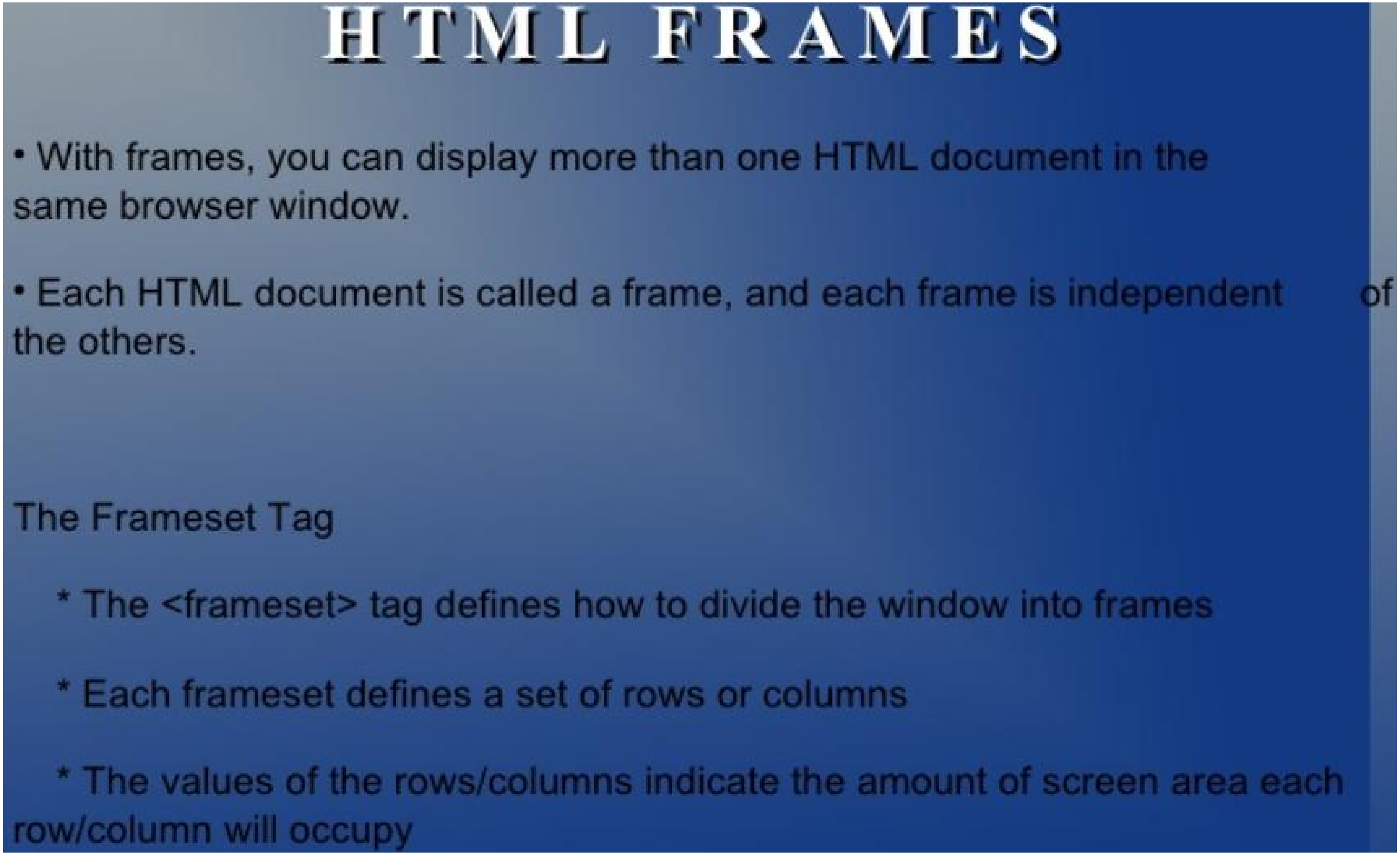
- A part of this page is formatted with two columns, like a newspaper page.
- The trick is to use a table without borders, and maybe a little extra cell-padding.
- No matter how much text you add to this page, it will stay inside its column borders.



```
<table border="0" width="100%" cellpadding="10">
<tr>
<td width="50%" valign="top">
      This is the Time to save Our Earth to Our Future Generation.So
  everybody shoud be a Volunteer.
</td>
<td width="50%" valign="top">
      For smooth relationship between to us & nature We should do some
  activities to Preserve our Earth.
</td>
</tr>   </table>
```



This is the Time to save Our Earth to Our Future Generation.So everybody shoud be a Volunteer.	For smooth relationship between to us & nature We should do some activities to Preserve our Earth.



HTML FRAMES

- With frames, you can display more than one HTML document in the same browser window.

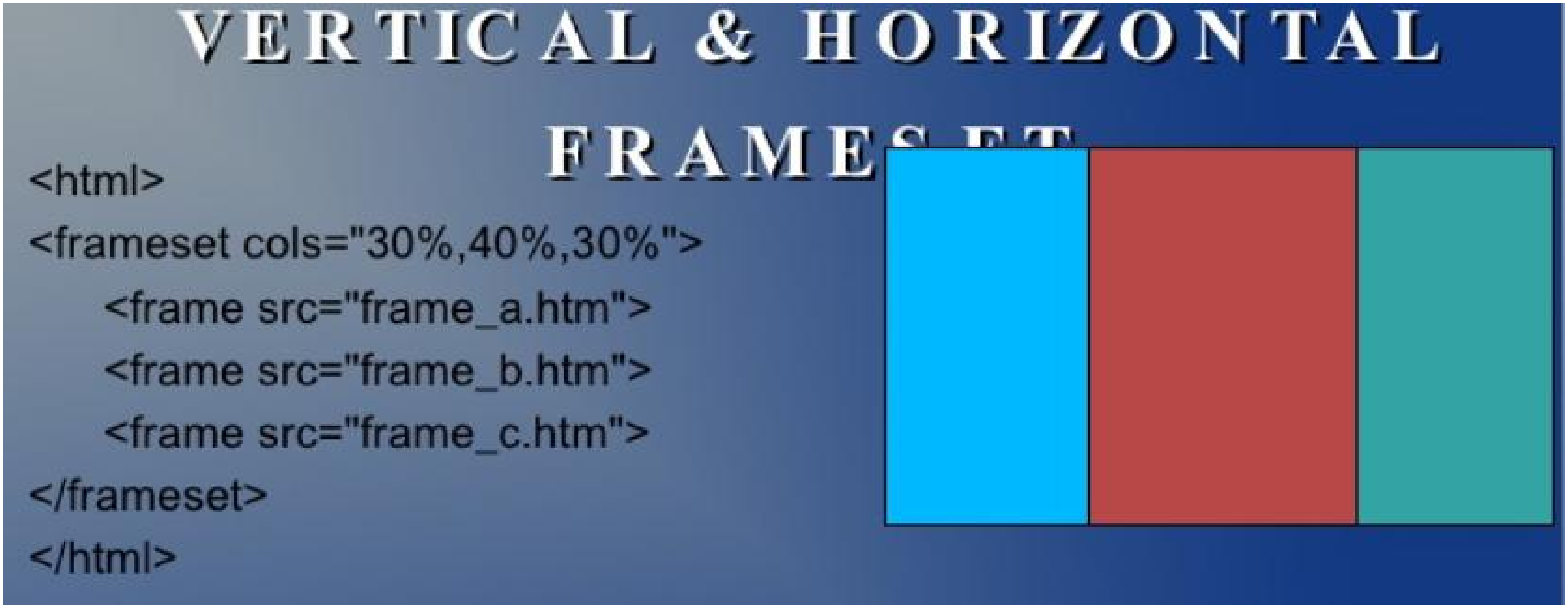
- Each HTML document is called a frame, and each frame is independent of the others.

The Frameset Tag

* The <frameset> tag defines how to divide the window into frames

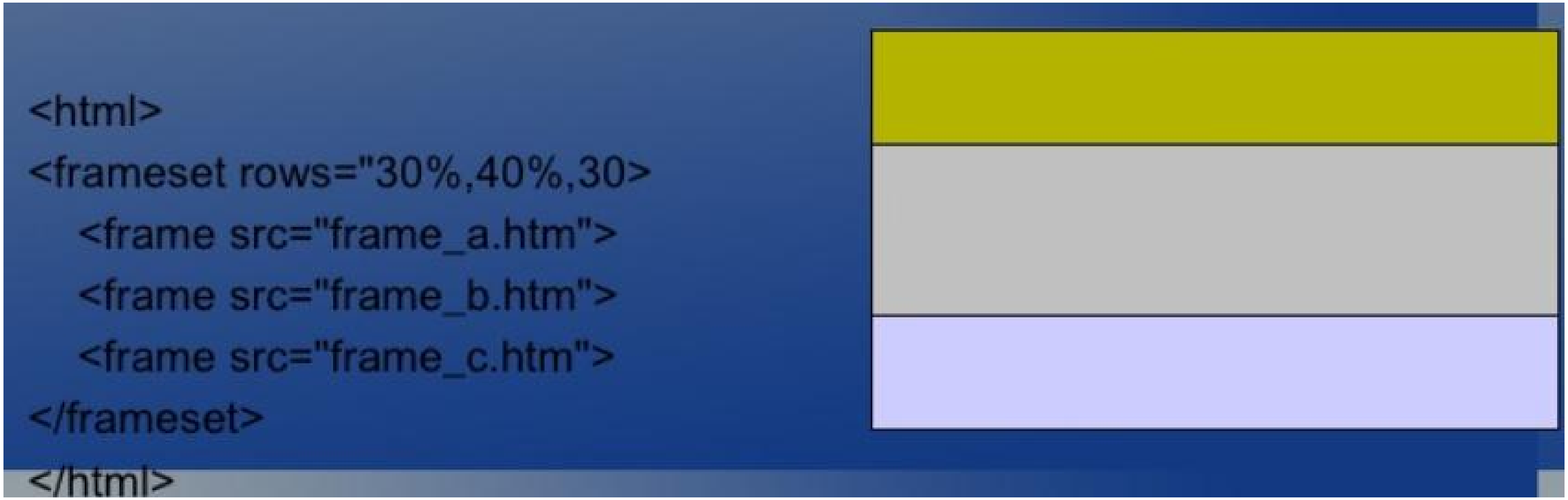
* Each frameset defines a set of rows or columns

* The values of the rows/columns indicate the amount of screen area each row/column will occupy



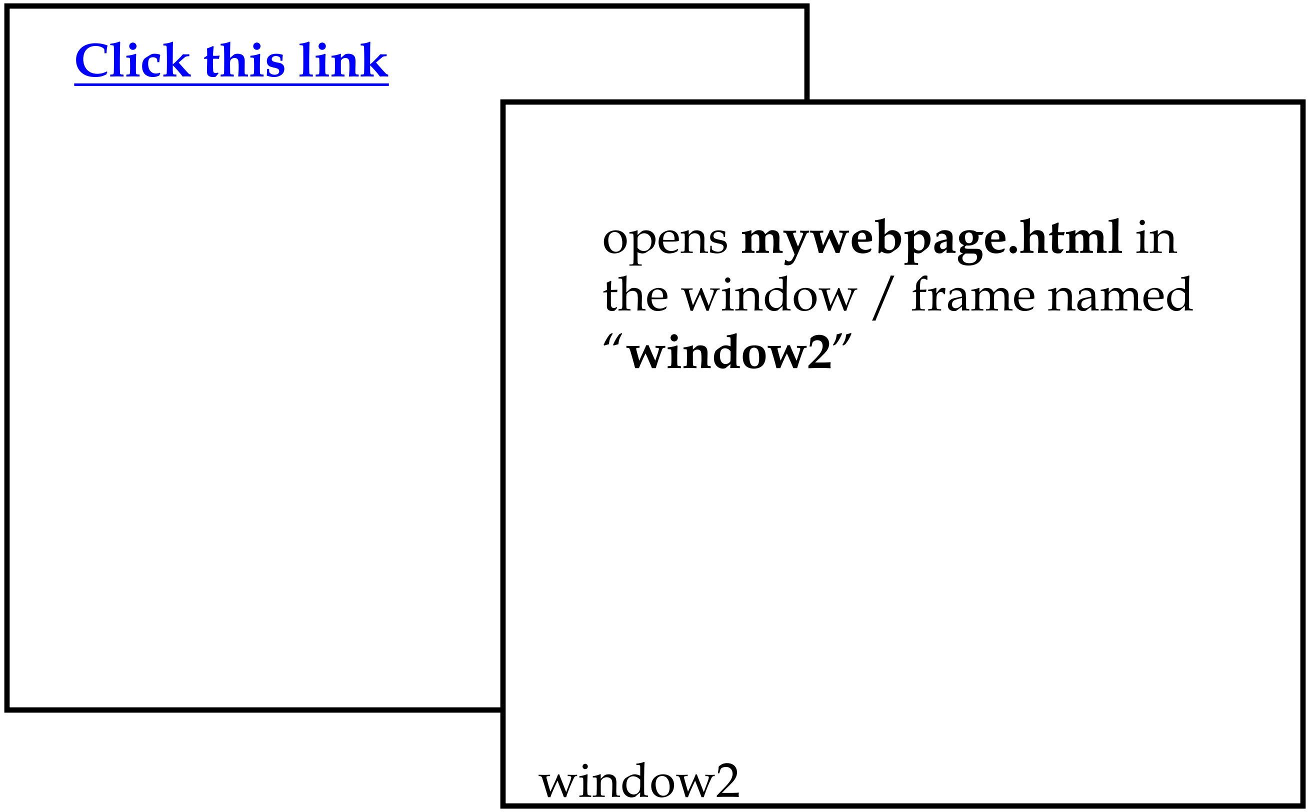
VERTICAL & HORIZONTAL FRAMESET

```
<html>
<frameset cols="30%,40%,30%">
    <frame src="frame_a.htm">
    <frame src="frame_b.htm">
    <frame src="frame_c.htm">
</frameset>
</html>
```



```
<html>
<frameset rows="30%,40%,30>
   <frame src="frame_a.htm">
   <frame src="frame_b.htm">
   <frame src="frame_c.htm">
</frameset>
</html>
```

```
<a href=“mywebpage.html” target=“window2” >Click this link </a>
```

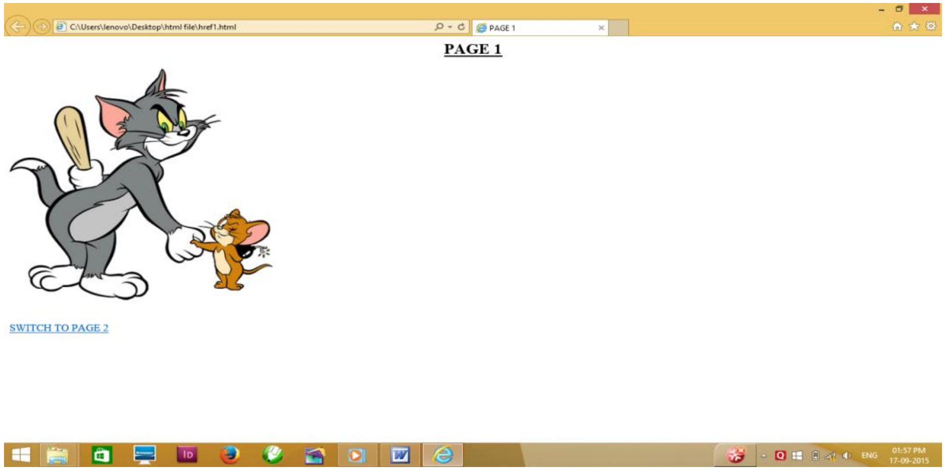


Click this link

opens **mywebpage.html** in the window / frame named “**window2**”

window2

```
<html>
<head>
<title>PAGE 1</title>
</head>
<body>
<center>
<h2><u>PAGE 1</u></h2>
</center>
<br>
<img src=C:\Users\lenovo\Pictures\taj.png>
<br>
<br>
<br>
<a href=C:\java_prog\applet\href2.html>SWITCH TO PAGE 2</a>
</body>
</html>
```



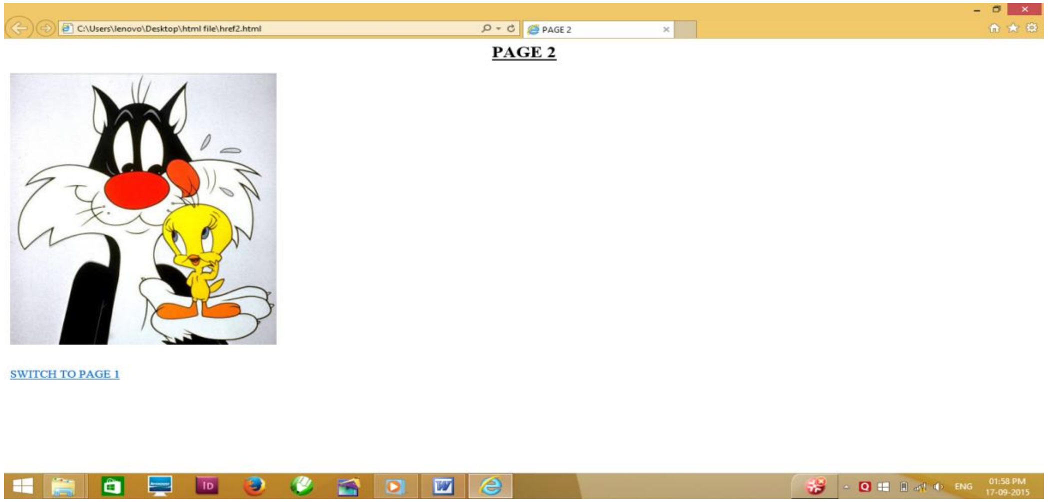
C:\Users\lenovo\Desktop\html file\href1.html

PAGE 1

PAGE 1

SWITCH TO PAGE 2

```
<html>
<head>
<title>PAGE 2</title>
</head>
<body>
<center>
<h2><u>PAGE 2</u></h2>
</center>
<br>
<img src=C:\Users\lenovo\Pictures\sat.jpeg>
<br>
<br>
<br>
<a href=C:\java_prog\applet\href1.html>SWITCH TO PAGE 1</a>
</body>
</html>
```



PAGE 2

SWITCH TO PAGE 1

```
<a href="https://www.rbsmtc.in">This is a link</a>
```

The link's destination is specified in the href attribute.

Attributes are used to provide additional information about HTML elements.

Below is an example of a very simple page:

This is the code used to make the page:

```
<HTML>
 <HEAD>

<TITLE>Your Title Here</TITLE>

</HEAD>

<BODY BGCOLOR="FFFFFF">

<CENTER>
<IMG SRC="clouds.jpg" ALIGN="BOTTOM"> </CENTER>
 <HR>
 <a href="http://somegreatsite.com">Link Name</a>
```

```
is a link to another nifty site

<H1>This is a Header</H1>

<H2>This is a Medium Header</H2>

Send me mail at <a href="mailto:support@yourcompany.com">

support@yourcompany.com</a>.

<P> This is a new paragraph!

<P> <B>This is a new paragraph!</B>

<BR> <B><I>This is a new sentence without a paragraph break, in bold
italics.</I></B>

<HR>

</BODY>

</HTML>
```

Example – Creating a directory or table of contents that will be displayed at the top of the web page and that will then link to subheading sections in the same document.

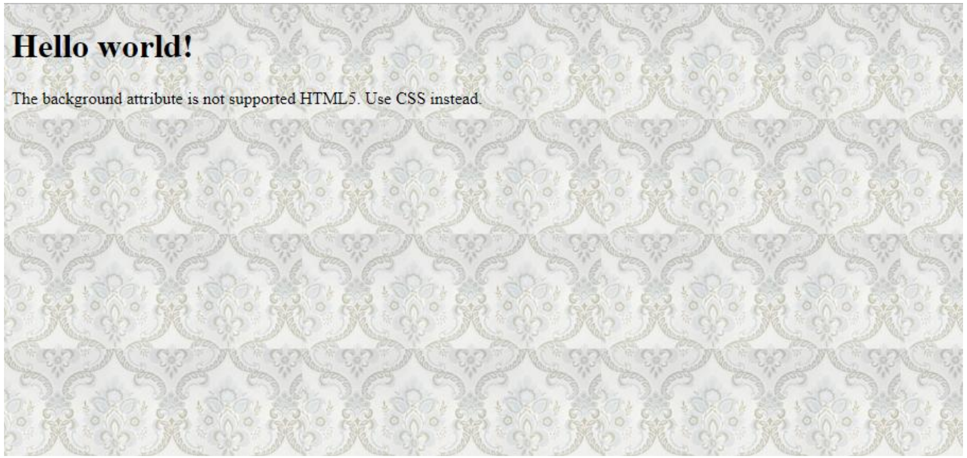
Sol. :-

```
<HTML>
<HEAD>
<TITLE>Linking Pages></TITLE>
</HEAD>
<BODY>
<H2>USING HYPERTEXT LINKS</H2>
<P>
<A HREF=“#PAGE”>LINKING TO ANOTHER PAGE OR FILE </A><BR>
<A HREF=“#LOCAT”>LINKING TO A PLACE ON THE SAME PAGE</A><BR>
<A HREF=“#LOCPG”>LINKING TO A PLACE ON ANOTHER PAGE </A><BR>
<H3>
<A NAME=“PAGE”>LINKING TO ANOTHER PAGE OF FILE</A></H3>
<P>
```

```
<P>
YOU CAN FORM A LINK WITH ANYTHING ON THE WEB THAT HAS AN
ADDRESS
<BR>

<H3>
<A NAME=“LOCAT”>LINKING TO A PLACE ON THE SAME PAGE</A></H3>
<P>
YOU CAN FORM A LINK WITH ANOTHER PLACE ON THE SAME PAGE BY
LINKING AN HREF AND A NAME ANCHOR
<BR>

<H3>
<A NAME =“LOCPG”>LINKING TO A PLACE ON THE SAME PAGE</A></H3>
<P>
YOU CAN NOT ONLY LINK TO ANOTHER HTML FILE, BUT TO A PLACE IN
THAT FILE        </BODY>          </HTML>
```

```
<html>

<body background="tyles.jpg">

<h1>Hello world!</h1>

<p>The background attribute is not supported HTML5. Use CSS instead.</p>

</body>

</html>
```

Result:



Hello world!

The background attribute is not supported HTML5. Use CSS instead.

Local Links

A local link (link to the same web site) is specified with a relative URL (without http://www....).

Example

```
<a href="html_images.asp">HTML Images</a>
```

Example

```
<a href="http://www.google.com/html/">Visit our HTML tutorial</a>
```

The **href** attribute specifies the destination address (http://www.google.com/html/) of the link.

The **link text** is the visible part (Visit our HTML tutorial).

Clicking on the link text will send you to the specified address.

Note : The example above used an absolute URL (A full web address).

HTML Link Colors

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

You can change the default colors, by using styles:

Example

```
<style>
a:link    {color:green; background-color:transparent; text-decoration:none}
a:visited {color:pink; background-color:transparent; text-decoration:none}
a:hover   {color:red; background-color:transparent; text-decoration:underline}
a:active  {color:yellow; background-color:transparent; text-decoration:underline}
</style>
```

<BODY LINK=*color*, VLINK=*color*, ALINK=*color* >

<BODY LINK=*“blue”*, VLINK=*“purple”*, ALINK=*“red”* >

<BODY LINK=*“#0000FF”*, VLINK=*“#FF00FF”*, ALINK=*“#FF0000”* >

HTML Links - The target Attribute

The **target** attribute specifies where to open the linked document.

The target attribute can have one of the following values:

- _blank - Opens the linked document in a new window or tab
- _self - Opens the linked document in the same window/tab as it was clicked (this is default)
- _parent - Opens the linked document in the parent frame
- _top - Opens the linked document in the full body of the window
- framename - Opens the linked document in a named frame

This example will open the linked document in a new browser window/tab:

Example

```
<a href="http://www.google.com/" target="_blank">Visit google</a>
```

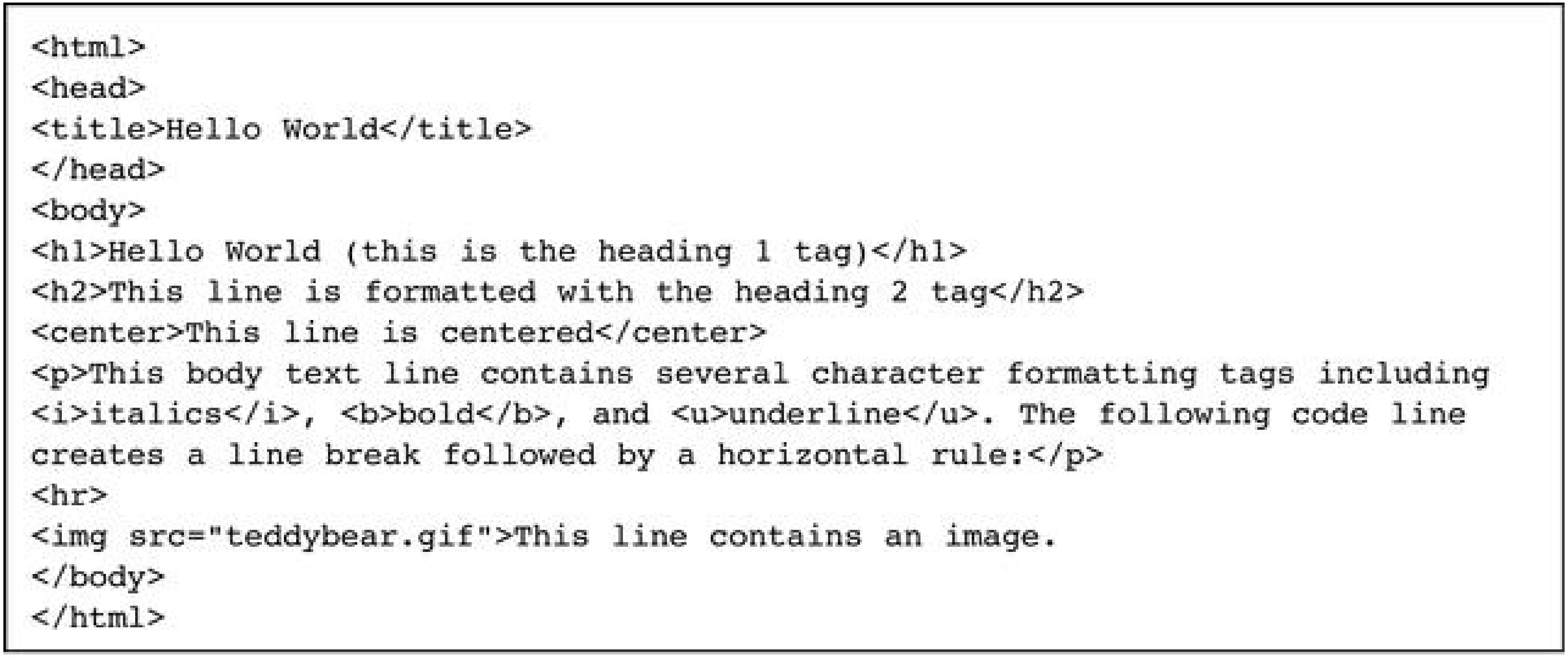
HTML Links - Image as Link

It is common to use images as links:

Example

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;border:0;">
</a>
```

Basic HTML Syntax



```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<h1>Hello World (this is the heading 1 tag)</h1>
<h2>This line is formatted with the heading 2 tag</h2>
<center>This line is centered</center>
<p>This body text line contains several character formatting tags including
<i>italics</i>, <b>bold</b>, and <u>underline</u>. The following code line
creates a line break followed by a horizontal rule:</p>
<hr>
<img src="teddybear.gif">This line contains an image.
</body>
</html>
```

A simple HTML document

HTML - Fundamentals

header

```
<body>


   <font face=“Arial”  color=“red” size=“3”>
   <b>Hello</b><br>
   World<br>
   </font>


   </body>
```

HTML - Fundamentals

```
header
<body>
    <p align=‘center’>
    <font face=“Arial”  color=“red” size=“3”>
    <b>Hello</b><br>
    World<br>
    </font>
    </p>
    </body>
```

HTML - Fundamentals

```
header
<body>
    <p align=‘center’>
    <font face=“Arial”  color=“red” size=“3”>
    <b>Hello</b><br>
    World<br>
    </font>
    </p>
    <img src=‘f:/images/images1.jpg’/>

</body>
```

HTML – Fundamentals

Images

<img src=“images/pic1.jpg” width=75px, height=50px />

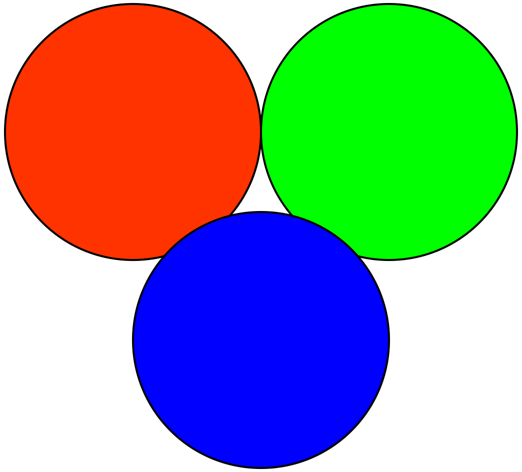
Alt = “ “

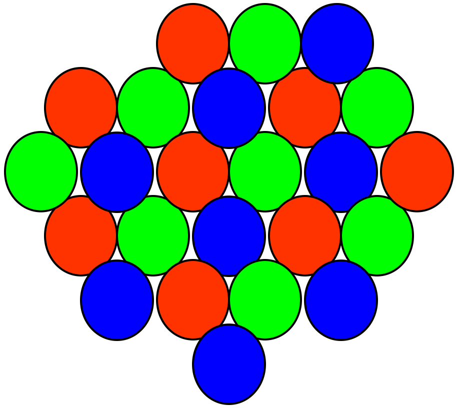
Border = value

HTML - Fundamentals

HTML – Fundamentals
Colors

- Cathode Ray Tubes (CRT)





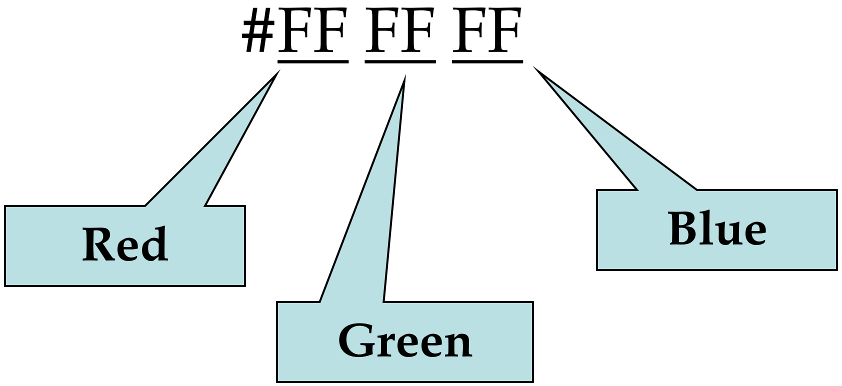
HTML – Fundamentals
Colors

color = “red” (Browser compatibility issues)

color = “#FF0000”

values vary from 00 to FF (hexadecimal)

0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f



#FF FF FF

Red

Green

Blue

Creating list

Heading and paragraph text elements are used more commonly than lists. Many web pages are nothing but lists of hypertext links. You can create two types of list : Ordered and unordered. An ordered list is a numbered list and an unordered list is a bulleted list.

The UL (Unordered List) tag – The unordered list element is used to present a list of items which is separated by white space or is marked by bullets. An unordered list starts with the <UL> element followed by the <LI>element .

The OL (Ordered List) tag – The ordered list element is used to present a numbered list of items in the order of importance or the items are sorted by sequence. An ordered list starts with the <OL> element followed by the <LI>element .

Unordered list

```
<ul>
    <li>apples</li>
    <li>bananas</li>
    <li>grapes</li>
    <li>strawberries</li>
</ul>
```

Ordered list

```
<ol>
    <li>apples</li>
    <li>bananas</li>
    <li>grapes</li>
    <li>strawberries</li>
</ol>
```

Unordered list

- apples
- bananas
- grapes
- strawberries

Ordered list

II. apples
III. bananas
IV. grapes
V. strawberries

Nesting list

You can nest a list inside another list. The browser automatically indents nested list levels. You can nest the same or different kinds of lists.

Example-

```
<html>
<head>
<title> The title of your html page </title>
</head>
<body>
<UL>
<LI>SOFTWARE
    <OL>
    <LI>COMPUTER CONCEPTS
    <LI>MS-WINDOWS
    <LI>MS-EXCEL
    <LI>MS-WORD
    </OL>
```

```
<LI>HARDWARE
<OL>
        <LI>CPU
        <LI>INPUT DEVICES
        <LI>OUTPUT DEVICES
        <LI>HARD DISK
    </OL>
    </UL>
     </body>
    </html>
```

OUTPUT-

- SOFTWARE
  1. COMPUTER CONCEPTS
  2. MS-WINDOWS
  3. MS-EXCEL
  4. MS-WORD
- HARDWARE
  1. CPU
  2. INPUT DEVICES
  3. OUTPUT DEVICES
  4. HARD DISK

Controlling how Ordered list are displayed

The TYPE attribute allows you to specify the number type for an OL.

Specifying the Number Type : You can use the TYPE attribute to specify the number type for an OL. The values that you can use with the TYPE attribute are “A”, “a”, “I”, “i” and “1” for specifying uppercase letters, lowercase letters, uppercase roman numerals, lowercase roman numerals or Arabic numbers respectively,

eg.-

```
<html>
<head>
<title> The title of your html page </title>
</head>
<body>
<OL TYPE =“I”>

<LI>LEVEL ONE OUTLINE LEVEL

<OL TYPE =“A”>

<LI>LEVEL TWO OUTLINE LEVEL
```

```
<OL TYPE =“1”>

<LI>LEVEL THREE OUTLINE LEVEL

<OL TYPE =“A”>

<LI>LEVEL FOUR OUTLINE LEVEL

<LI>LEVEL FOUR OUTLINE LEVEL

</OL>

<LI>LEVEL THREE OUTLINE LEVEL

</OL>

<LI>LEVEL TWO OUTLINE LEVEL

</OL>

<LI>LEVEL ONE OUTLINE LEVEL

</OL>

</body>          </html>
```

Output –

```
   I.  LEVEL ONE OUTLINE LEVEL
       A.     LEVEL TWO OUTLINE LEVEL
              1.     LEVEL THREE OUTLINE LEVEL
                     A.     LEVEL FOUR OUTLINE LEVEL
                     B.     LEVEL FOUR OUTLINE LEVEL
              2.     LEVEL THREE OUTLINE LEVEL
       B.     LEVEL TWO OUTLINE LEVEL
   II. LEVEL ONE OUTLINE LEVEL
```

Using START and Value Attributes in an OL

The START attribute in an OL start tag is used to start the numbering sequence at a particular number. You can use the VALUE attribute in an LI tag to restart the numbering sequence at a particular number.

eg.- first start the numbering sequence at 3 and then restart it at 8, then enter the following commands

```
<OL START =“3”>
 <LI>THIS SHOULD BE NUMBERED AS 3
 <LI>THIS SHOULD BE NUMBERED AS 4
 <LI VALUE =“8”> THIS SHOULD BE NUMBERED AS 8
 <LI>THIS SHOULD BE NUMBERED AS 9
 </OL>
```

Controlling how Unordered list are displayed

The TYPE attribute allows you to specify the number type of bullet for an unordered list. The values that you can use with the TYPE attribute are “disc”, “circle”, and “square”.eg.-   amitkohli2k1@yahoo.com

```
<html>

<head>

<title> The title of your html page </title>

</head>

<body>

<UL TYPE =“SQUARE”>

<LI>SOFTWARE

<UL TYPE =“DISC”>

<LI>COMPUTER CONCEPTS

<LI>MS-WINDOWS

<LI>MS-EXCEL

<LI>MS-WORD                </UL>
```

```
<LI>HARDWARE
<UL TYPE=“CIRCLE”>
<LI>CPU
<LI>INPUT DEVICES
<LI>OUTPUT DEVICES
<LI>HARD DISK
</UL>
</body>
</html>
```

Creating Definition Lists

The DL tag allows you to create glossaries or lists of terms and definitions. A glossary consists of three tag elements : a tag to define the list (DL), a tag to define the term (DT) and a tag to define the definitions (DD).

```
<html>
<head>
<title>DEFINITIONS LIST</title>
</head>
<body>
<dl>
<dt>apples</dt>
  <dd>A fruit of red color</dd>
<dt>grapes</dt>
  <dd>Small green color fruit</dd>
</dl>
```

Output :-

```
apples
        A fruit of red color
grapes
        Small green color fruit
```

Create Tables

Tables are used to display data in a tabular format on the screen. Tables are useful in communicating data in a meaningful way to the end-user. All other tags or text included in your table should nest inside the Table tag i.e. <Table> …………</Table>

Creating Columns and Rows :- The table row (TR) and table data (TD) tags are used to create a grid of rows and columns. The no. of rows in a table depends on how many <TR> elements are contained with in the table.

The <TD> elements stands for table data. Table data cell must only appear within table rows.

HTML – Fundamentals
Tables

The <TR> element can have the following attributes –

- BORDER=*value*
- ALIGN=*left|right|center*
- CELLSPACING=*value*
- CELLPADDING=*value*
- WIDTH=*value|percent*

Adding a Border : By default, the table does not contain a border. To include a border to the table, specify a BORDER attribute inside the TABLE tag.

Adding a Caption : The CAPTION tag is used to insert a caption for the table . The <CAPTION> element by default is center-aligned at the top of the table.

eg. <TABLE BORDER="1" CELLSPACING="6" CELLPADDING="6">
    <CAPTION ALIGN="BOTTOM">

Adding Column Headings : The Table Heading (TH) tag is used to define a “cell” as a heading cell rather than as an ordinary data cell. To create the headings at the top of the table, first create a row using the TR tag and then use TH tag to define the cells instead of using the TD tags.

Adding Spacing and Padding : The CELLSPACING attribute adds space between the cells whereas CELLPADDING attribute adds space within each cell.

Setting the Table Width and Height : The WIDTH or HEIGHT attribute can be used to specify the size to the table. You can use either absolute value (number of pixels) or relative values (percentages).
Eg. <TABLE BORDER=“1” CELLSPACING=“6” CELLPADDING=“6” WIDTH=“80%”>

Add Row Headings : To create a row heading add TH cell instead of a TD cell at the start of a table row.

Aligning Cell Contents : The ALIGN attribute is used to align the contents of TH (Table Heading) and TD (Table Data) cells by assigning “left”, “center” or “right” as the ALIGN value.

Setting Column Width : By inserting a WIDTH attribute in the top cell of a column, you can specify the width of the entire column. To ensure that all of the columns are of equal width, you need to set the same WIDTH attribute value, either as a percentage for each column.

```
Eg. <TR>
<TH WIDTH=“20%”>ROLL NO</TH>
<TH WIDTH=“25%”>FIRST NAME</TH>
<TH WIDTH=“30%”>LAST NAME</TH>
<TH WIDTH=“25%”>FEES PAID</TH>
</TR>
```

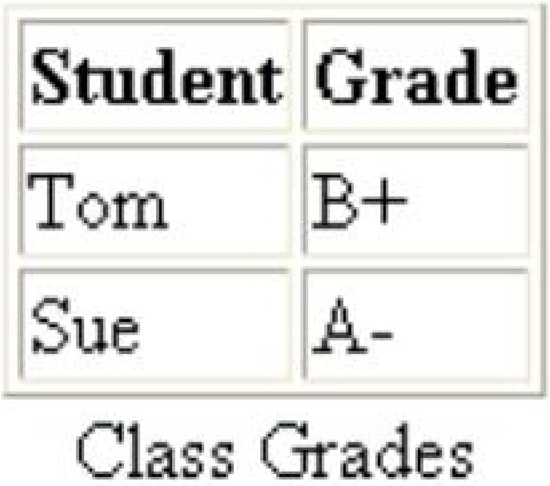
Centering a Table : To center the table, put it inside a CENTER tag.

```
<CENTER> ………………</CENTER>
```

```
<table>
    <tr>
        <td>Item 1</td>
        <td>Item 2</td>
        <td>Item 3</td>
    </tr>
    <tr>
        <td>Item 1</td>
        <td>Item 2</td>
        <td>Item 3</td>
    </tr>
    <tr>
        <td>Item 1</td>
        <td>Item 2</td>
        <td>Item 3</td>
    </tr>
</table>
```

Item 1	Item 2	Item 3
Item 1	Item 2	Item 3
Item 1	Item 2	Item 3

```
<TABLE>
    <CAPTION ALIGN="bottom">Class Grades</CAPTION>
    <TR>
        <TH>Student</TH>
        <TH>Grade</TH>
    </TR>
    <TR>
        <TD>Tom</TD>
        <TD>B+</TD>
    </TR>
    <TR>
        <TD>Sue</TD>
        <TD>A-</TD>
    </TR>
</TABLE>
```



Student	**Grade**
Tom	B+
Sue	A-

Class Grades

```
<TABLE BORDER=“1” WIDTH=“50%" CELLPADDING=“6”
    CELLSPACING=“2” ALIGN="RIGHT">
    <CAPTION ALIGN="bottom">Class Grades</CAPTION>
    <TR>
        <TH>Student</TH>
        <TH>Grade</TH>
    </TR>
    <TR>
        <TD>Tom</TD>
        <TD>B+</TD>
    </TR>
    <TR>
        <TD>Sue</TD>
        <TD>A-</TD>
    </TR>
</TABLE>
```

Student	Grade
Tom	B-
Sue	A+

Class Grades

Spanning Columns (COLSPAN) : This attribute can appear within any table cell and it specifies the number of columns the cell can span. The default COLSPAN for any cell is 1. For example COLSPAN="4" indicates that 4 cells are merged to become one cell.

Spanning Rows (ROWSPAN) : This attribute specifies the number of rows the cell can span. The default span for any cell is 1.

eg

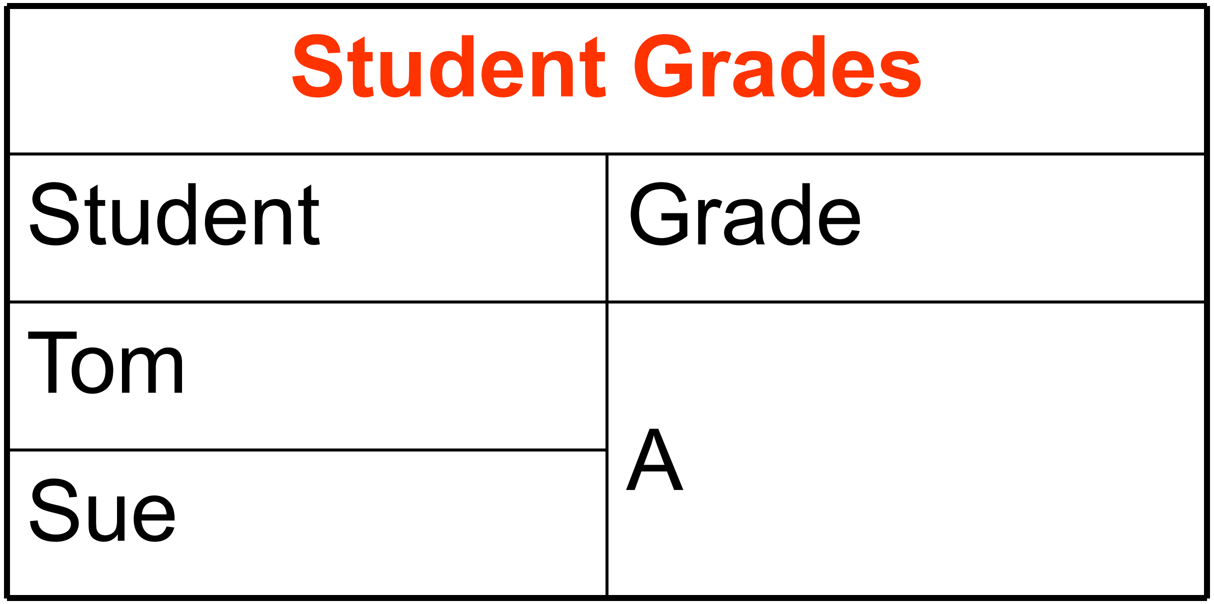
rowspan and colspan

<table>
<tr><td></td><td colspan="2"></td></tr>
<tr><td rowspan="2"></td><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>

HTML – Fundamentals

```
<TABLE BORDER=1 WIDTH="50%" CELLPADDING=5 ALIGN="center">
    <TR>
        <TD colspan=2 align='center'>
                <font color="red"><b>Student Grades</b></font>
        </TD>
    </TR>
    <TR>
        <TD><b>Student</b></TD>
        <TD><b>Grade</b></TD>
    </TR>
    <TR>
        <TD>Tom</TD>
        <TD rowspan=2>A</TD>
    </TR>
    <TR>
        <TD>Sue</TD>
    </TR>
</TABLE>
```

HTML – Fundamentals

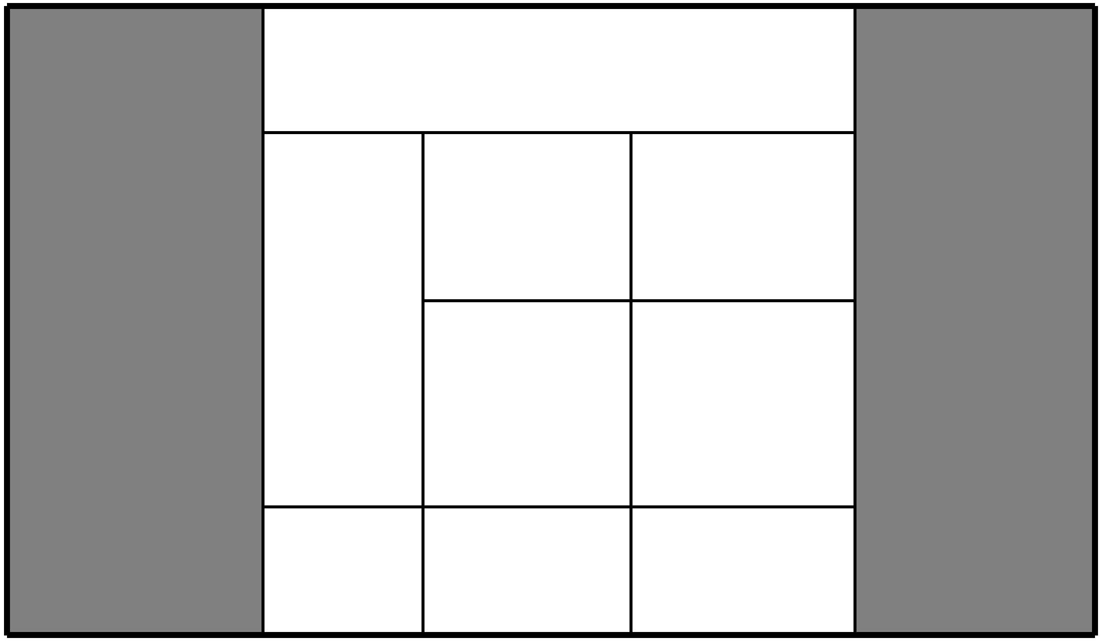


Student Grades

Student	Grade
Tom	A
Sue	

HTML – Fundamentals
Tables

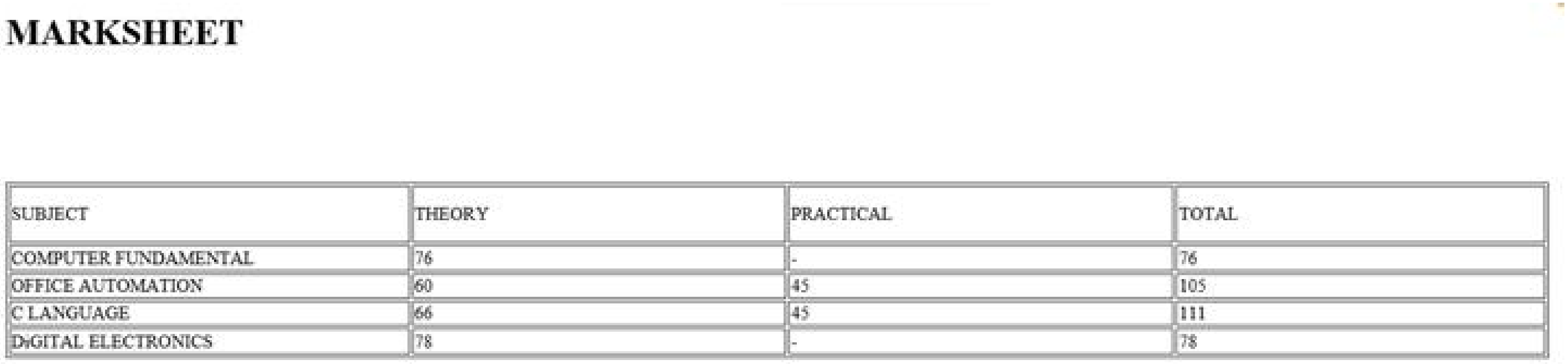
- Tables are frequently used to layout the basic web page design.



CREATING TABLE IN HTML

```
<HTML>
<head>
</head>
<body>
<h1>MARKSHEET</h1>
<br><br><br><br><br>
<table border=1>
<tr>
    <td width=500 height=50>SUBJECT</td>
    <td width=500>THEORY</td>
    <td width=500>PRACTICAL</td>
    <td width=500>TOTAL</td>
</tr>
<tr>
    <td width=500>COMPUTER FUNDAMENTAL</td>
    <td width=500>76</td><td width=500>-</td>
    <td width=500>76</td>
</tr>
<tr>
   <td width=500>OFFICE AUTOMATION</td><td width=500>60</td>
   <td width=500>45</td><td width=500>105</td>
</tr>
```

```
<tr>
   <td width=500>C LANGUAGE</td><td width=500>66</td>
   <td width=500>45</td><td width=500>111</td>
</tr>
<tr>
   <td width=500>DiGITAL ELECTRONICS</td>
   <td width=500>78</td><td width=500>-</td>
   <td width=500>78</td>
</tr>
</table>
</body>
</HTML>
```



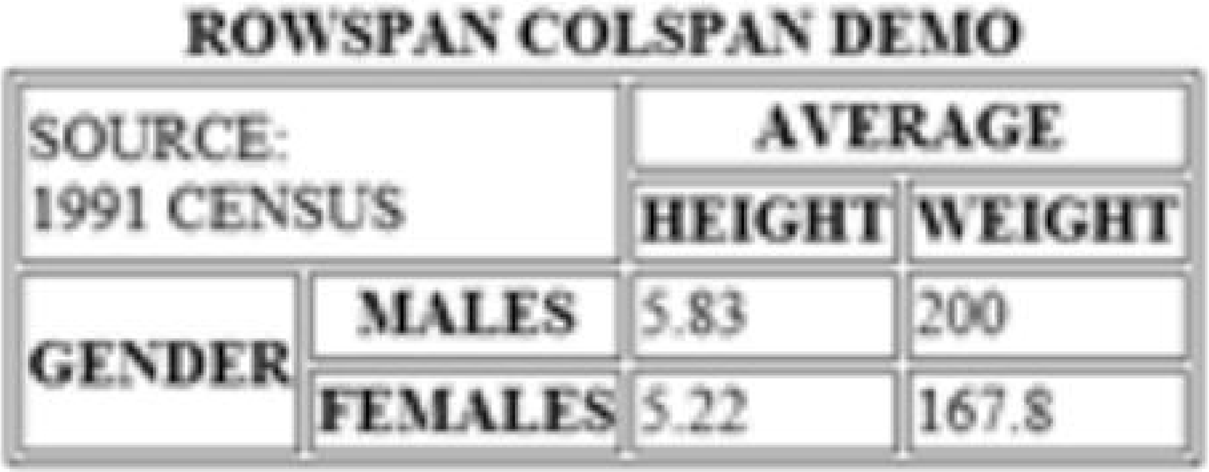
MARKSHEET

SUBJECT	THEORY	PRACTICAL	TOTAL
COMPUTER FUNDAMENTAL	76	-	76
OFFICE AUTOMATION	60	45	105
C LANGUAGE	66	45	111
DiGITAL ELECTRONICS	78	-	78

CREATING ROWSPAN AND COLSPAN IN HTML

```
<html>
<head>
</head>
<body>
<center>
<table border=1>
<caption><b>ROWSPAN COLSPAN DEMO</b></caption>
<tr>
   <td colspan=2 rowspan=2>SOURCE:<br>1991 CENSUS</td>
   <th colspan=2>AVERAGE</th>
</tr>
<tr>
   <th>HEIGHT</th>
   <th>WEIGHT</th>
</tr>
<tr>
   <th rowspan=2>GENDER</th>
   <th>MALES</th>
   <td>5.83</td>
   <td>200</td>
```

```
</tr>
    <th>FEMALES</th>
    <td>5.22</td>
    <td>167.8</td>
</tr>
</table>
</center>
</body>
</html>
```



ROWSPAN COLSPAN DEMO

SOURCE: 1991 CENSUS		AVERAGE	
		HEIGHT	WEIGHT
GENDER	MALES	5.83	200
	FEMALES	5.22	167.8

Setting a specific text size

Again, this one uses the same basic code logic, but uses the element "font-size." Put your font sizes in pixels, or px.

```
<p style="font-size:36px;">Make this paragraph size 36 font.</p>
<p style="font-size:12px;">Make this paragraph size 12 font.</p>
```

Make this paragraph size 36 font.

Make this paragraph size 12 font.

Marked or highlighted text

I bet you didn't know that you could highlight text through an HTML code, did you? It's so cool and so easy. Wrap the text to be highlighted in <mark> tags </mark> so that you get a cool highlighted feature.

```
Only highlight the <mark>most important notes</mark>.
```

Only highlight the most important notes.

Setting a specific text alignment

Left, centered, right, or justified. How do you like your text? Make it any way you'd like with "text-align.“

```
<p style="text=align:center;">This paragraph should be centered.</p>
<p style="text-align:right;">This paragraph should be right aligned.</p>
```

This paragraph should be centered.

This paragraph should be right aligned.

Setting a specific text color

This one uses the same code type as the previous example, but instead of using font-family, you use "color." You can experiment with actual colors (blue, red, orange, etc.), or you can also insert hex colors to customize your text to your brand.

```
<p style="color:blue;">The sky is really blue today.</p>
<p style="color:#ff471a;">The fire is a reddish-orange.</p>
```

The sky is really blue today.
The fire is a reddish-orange.

Short and long quotations

By now, you may be wondering how I'm putting all of my examples in a text box. Well, surprise! There's a code for that, too. It's actually called a blockquote or a long quote. You can see the difference between a long quote and a short quote (normal quotation marks) below.

```
<blockquote>All of this text will be in a blockquote like the rest of the
examples.</blockquote>
<q>I'm quoting this because I'm saying it out loud.</q>
```

All of this text will be in a blockquote like the rest of the examples.
I'm quoting this because I'm saying it out loud.

Setting a specific font

The next few are going to get a bit trickier, so try to stay with me. Now that you know how to create a heading, a paragraph, and stylized text, it'll be useful for you to know that you can easily change the font using the element "font-family."
Don't forget all of the small details, such as the equal sign, quotation marks, and semi-colon. Check out the examples below.

```
<h4 style="font-family:Georgia;">I want to change this header to Georgia font.</h4>
<p style="font-family:Verdana;">I want to change this paragraph to Verdana font.</p>
```

I want to change this header to Georgia font.
I want to change this paragraph to Verdana font.

<div> … </div> — A block-level container for content

The div element is a generic container that you can use to add more structure to your page content. For example, you might group several paragraphs or headings that serve a similar purpose together inside a div element. Typically, div elements are used for things like:

a. Page headers and footers
b. Columns of content and sidebars
c. Highlighted boxes within the text flow
d. Areas of the page with a specific purpose, such as ad spots
e. Image galleries

Here’s an example that uses a div to contain the content for a sidebar in the page:

```
<div id="sidebar">
  <h1>Sidebar Heading</h1>
  <p>Sidebar text</p>
  <p>More sidebar text</p>
</div>
```

HTML – Fundamentals
Anchors

- External linking

<A href="filename.htm">hyperlink text </A>

<A Name ="location name">

<A href="filename.html#location name">
hyperlink text</A>

- Internal linking

<A Name ="location name">

<A href="#location name">
hyperlink text</A>

HTML – Fundamentals
Images

- Image links

```
<A HREF=“filename.html”> <IMG SRC =“image.jpg”></A>
```

- Image Maps

```
<MAP NAME = “mapname”>
<area>
```

shape – rect,	circle,	polygon,	default
Coords – x1,y1,x2,y2	x,y,radius	3 or pair	no coords

href

```
<MAP NAME =“map1”>
    <AREA SHAPE=“rect” COORD=“24,62,112,84” HREF=“first.html”>
    <AREA SHAPE=“circle” COORD=“12,82,10” HREF=“second.html”>
</MAP>
```

```
 <IMG  SRC =“image.jpg” USEMAP=“#map1”>
```

HTML – Fundamentals
Frames

- Frames let you divide a screen into windows with each window viewing a different web page.

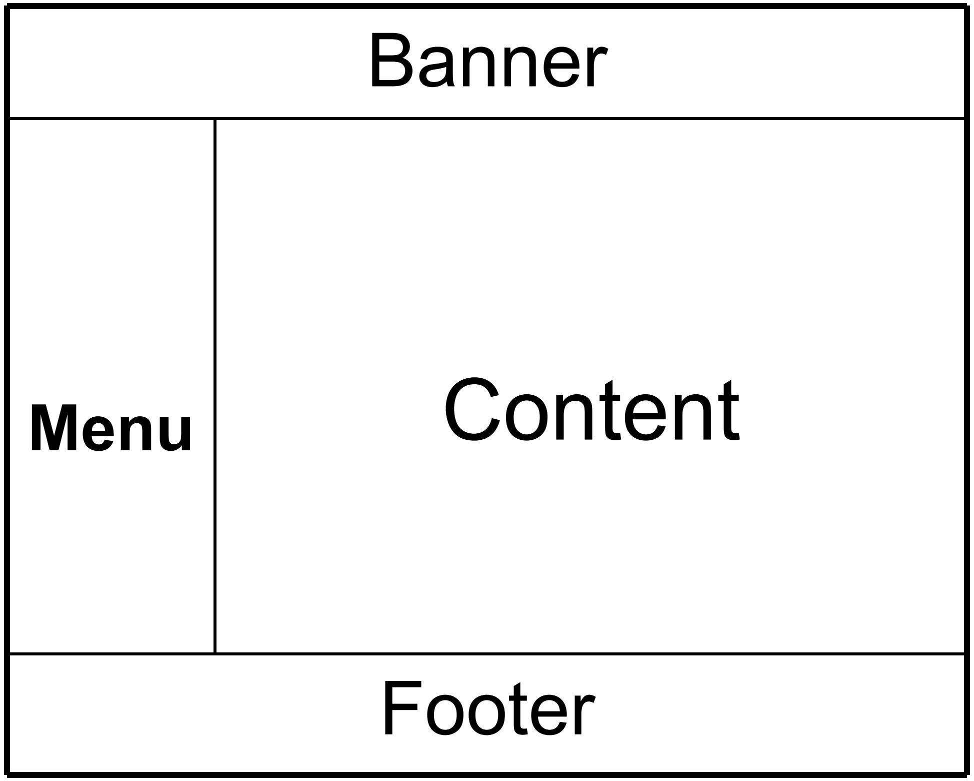
```
<frameset> ..</frameset>
<frame >   </frame>
```

Basic attributes

```
cols = “values”
rows = “values”

name = “frame_name”
src = “frame_source(url)”
```

HTML – Fundamentals

Basic Frames



Banner

Menu

Content

Footer

HTML – Fundamentals

Basic Frames

Individual FRAME attributes

- SCROLLING="yes|no|auto"
- NORESIZE
- MARGINWIDTH=pixels
- MARGINHEIGHT="pixels"
- BORDERCOLOR="color"
- FRAMESPACING="pixels"
- FRAMEBORDER="yes|no"|0
- NAME=“frame_name”

HTML – Fundamentals
Forms

- <FORM Method= “get/post” Action =>

<INPUT NAME=“” TYPE=“” >

TEXT – SIZE, MAXLENGTH

PASSWORD

BUTTON – VALUE

SUBMIT

RESET

CHECKBOX – VALUE, CHECKED

RADIOBUTTON – VALUE, CHECKED

HIDDEN

HTML – Fundamentals
Forms

```
<TEXTAREA  NAME=“”  ROWS=“” COLS=“”>  ……………………</TEXTAREA>
```

```
<SELECT NAME=“” SIZE=“” MULTIPLE>
<OPTION>………….
<OPTION SELECTED>……..
</SELECT>
```

HTML – Fundamentals

CASCADING STYLE SHEETS (CSS)

- Styles enable you to define a consistent 'look' for your documents by describing **once** how headings, paragraphs, quotes, etc. should be displayed.

- Style sheet syntax is made up of three parts:

```
tag {attribute: value}
```

Limitations of HTML

- Predefined tags
- Static
- Does not support Event Handling
- Unstructured
- Lacks syntactic checking