WEB DEVELOPMENT

Introduction to Web Design

Concept of a Web page

A webpage is a text page commonly written in Hypertext Markup Language (HTML) that is accessible through the Internet or other network using an Internet browser.

A webpage is accessible by entering a URL address and may contain text, graphics and hyperlinks to other web pages and files.

Understanding Terminologies

i. World Wide Web (WWW):

Is an information space where documents and other web resources are identified by Uniform Resource Locator (URL), interlinked by hypertext links and can be accessed via the Internet.

ii. Web Server:

Is the actual computer that is setup to serve web pages to the world through the Internet.

iii. Website:

Is a group of web pages that have information in the various pages that contain similar subject material.

iv. Web pages:

Is a text page written in a form of HTML that contains text and links that will display graphics, audio etc.

v. Web browser:

Is a software program used to access and navigate the Internet and enable the user to view web pages.

vi. Hyperlink:

A word, button or image that when clicked will go to another webpage either within the same website or to another web address.

vii. Address Bar:

A horizontal area at the top of the browser window where you enter the web address of the website that you would like to visit.

viii. Doctype:

Is the declaration specifies which version of HTML is used in a document. It has a direct effect on whether your HTML will validate.

ix. FTP Client:

The software program that you use to upload your website to a host server.

x. Hypertext:

Is any computer based text that includes hyperlinks.

xi. HTML (Hypertext Markup Language): This is the base language that is used for creating websites.

xii. Tag: Is a set of markup characters that are used around an element.

Basic HTML Web page Structure

A web page constructed using HTML has a basic and essential structure. Following are important considerations when preparing a webpage structure:

i. Plan Web pages:

-Planning is essential for creating a website, whether your site consists of one page or many.

-Decide on a main topic or theme, what you want to accomplish with your web pages.

ii. Gather Information:

Collect the information you want to include on your web pages such as text, images etc.

iii. Organize Information:

Divide the information you gathered into sections. Each section should be a separate webpage.

iv. Home Page:

-The home page is the main page in a website. The homepage is usually the first page people see when they visit a website.

-Always include a brief summary of your web pages on the home page.

-In order for your home page to load, you will need to name it “index” (all lower case because Linux is case sensitive). If you name your home page home for example, the server will not recognize it as the home page.

What Program Do I Use to Edit my HTML

There are many ways to create Html pages. You can use the File Manager Code edit, a program like ***Dreamweaver***, ***Notepad++, Visual Studio Code (Vs code), Sublime Text***, or any code editor to create an HTML page.

HTML is stored in regular text files. You simply name the file .html or .htm file extension.

HTML Basic: Open and Closed Tags

In HTML, there are two types of tags:

Open and ***Closed*** tag. i.e. **< >** is an open tag and **< />** is a closed tag.

Example

<body> </body>

-The ***<body>*** is the beginning or opening tag and the ***</body>*** is the ending or closing tag. These tags have content or other HTML tags within them.

-Tags within another tag is called "Element". The following example shows the <body> with an "Element" in it.

<body> <p> This is my website.</p></body>

-The beginning <body> tag tells the browser that the body of the page starts there and the </body> tag tells the browser the body of the page stops there. The <p> that is within the <body> tag is called an Element. This is because the <p> tag is nested within the <body> tag.

The Self Closing Tag

The tag that is left open has only one tag. The end of the tag contains a (/) telling the browser that the tag ends. An example of a tag that is left open is: <link href= "/ support/" rel = "" type = ""/>

Creating Web page

In order to create a web page so it is properly read by the web browser, you will need four (4) different HTML tags. Below is the explanation of the necessary tags for your web page.

Important!

<!DOCTYPE html> this tells the browser how the web page is coded. Do not worry about the type of document for now. This is not a tag; however it is needed at very top of the web page code.

Basic Web page Tags

i. **<html></html> :**

The <html> tag tells the browser that the code is an html document.

ii. **<title></title> :**

The <title> tag sets the text of the page for the browser toolbar and search results. This is not seen on the webpage itself. This is required in the head section of your webpage.

iii. **<head></head> :**

The <head> tag is where code is stored that processes styles, scripts and links to internal documents.

iv. **<body></body> :**

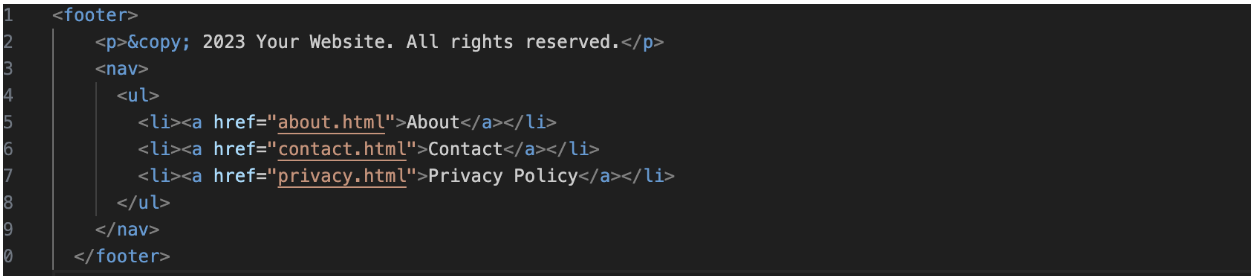
The <body> tag is where all the content for your website is stored. This is what is visible to the visitors of the web page.

v. **<footer></footer>:**

A HTML footer is a section of a webpage that is typically located at the bottom. It provides additional information or navigation links related to the content of the webpage. The footer is placed inside the **<footer>** element in HTML.

The footer can include various elements such as text, links, images, copyright notices, and social media icons. It is often used to display contact information, terms of service, privacy policy, and other important details.

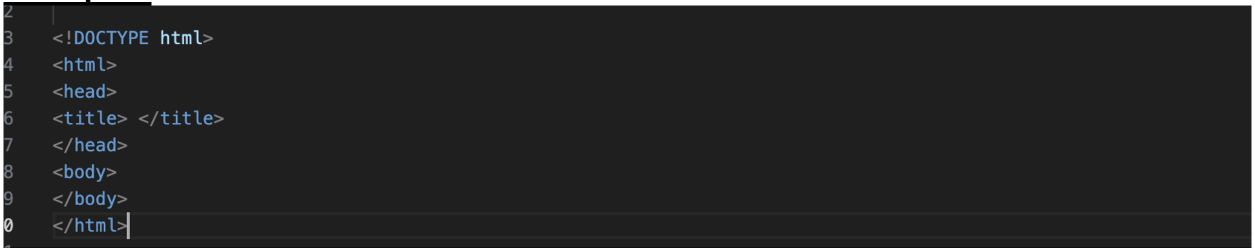
Here is an example code:



```
<footer>
    <p>&copy; 2023 Your Website. All rights reserved.</p>
    <nav>
      <ul>
        <li><a href="about.html">About</a></li>
        <li><a href="contact.html">Contact</a></li>
        <li><a href="privacy.html">Privacy Policy</a></li>
      </ul>
    </nav>
  </footer>
```

-The page always begins with the start tag of the html and always terminate with the end tag of the html as follows.

Example1:



```
<!DOCTYPE html>
<html>
<head>
<title> </title>
</head>
<body>
</body>
</html>
```

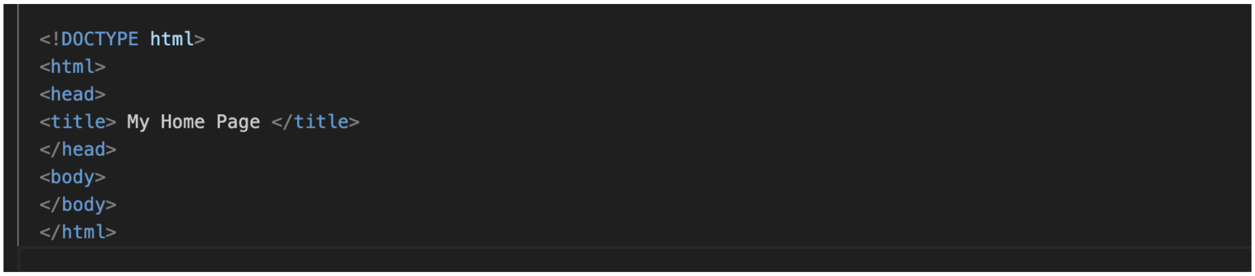
Now, save your web page as form3.html

Adding a Title to your Web page

-Web pages usually have a title that appears in the title bar that runs across the very top of the web page.

-Hence the following HTML code will produce a web page entitled “My Home Page”.

Example2:

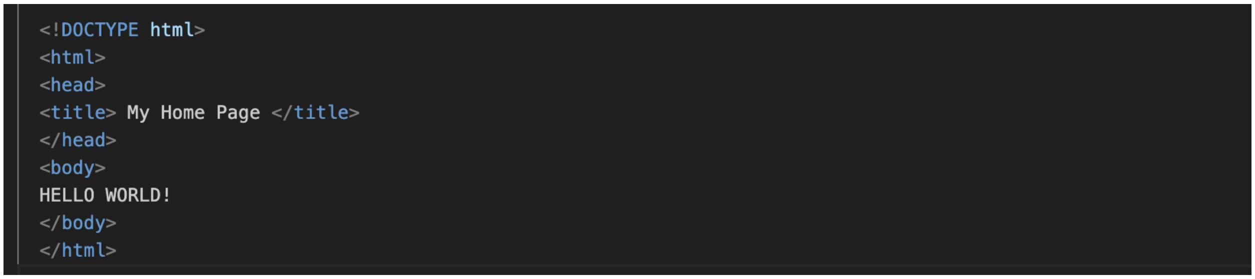


```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
</body>
</html>
```

Adding a Content to your Web page

Now, to add some content to your web page all you have to do is type some text in between the <body>……</body> tags. So let’s, for example put the words “HELLO WORLD!” on youe web page.

Example 3:



```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
HELLO WORLD!
</body>
</html>
```

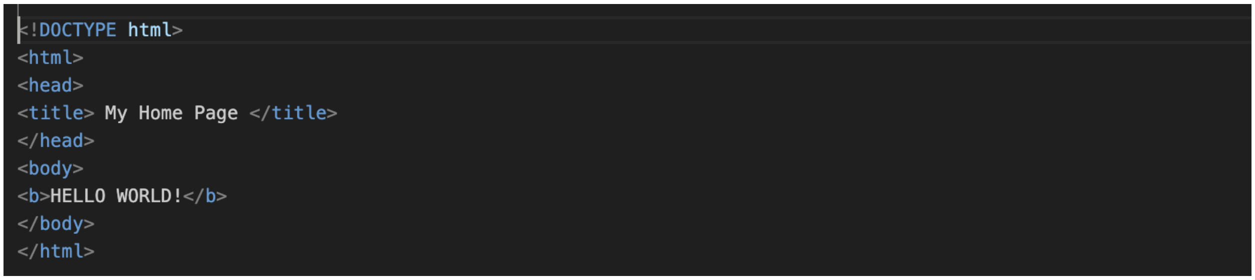
Text Formatting with HTML Code

Text formatting option are controlled by pairs of tags which surround the effected text. Consider the table below:

EFFECT	**HTML TAGS**	**WHAT IT LOOKS LIKE**
Bold face	<b> example</b>	**example**
Italic	<i>example</i>	*Example*
Bold Italic	<b><i>example</i></b>	***Example***
Subscript	H<sub>2</sub>O	H_2O
Superscript	Cm<sup>2</sup>	Cm^2
Color text	<style =“color:#000000”>example</font>	Example

Make the word “HELLOW WORLD!” bold from example 3

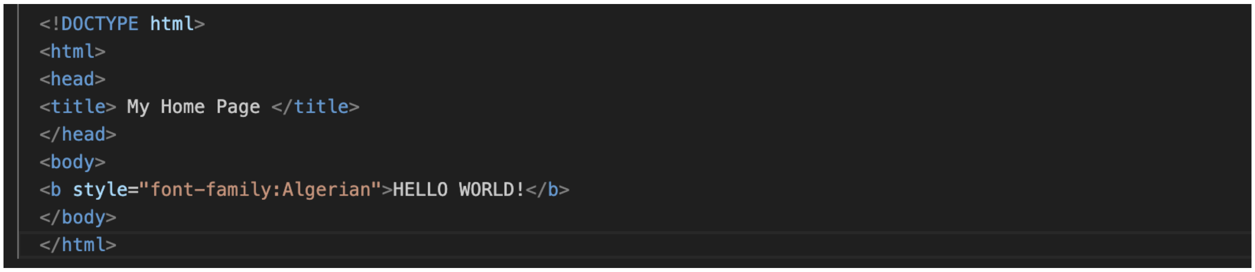
Example 4



```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
<b>HELLO WORLD!</b>
</body>
</html>
```

-Change the font face of the word “HELLO WORLD” to font face Algerian from example 4

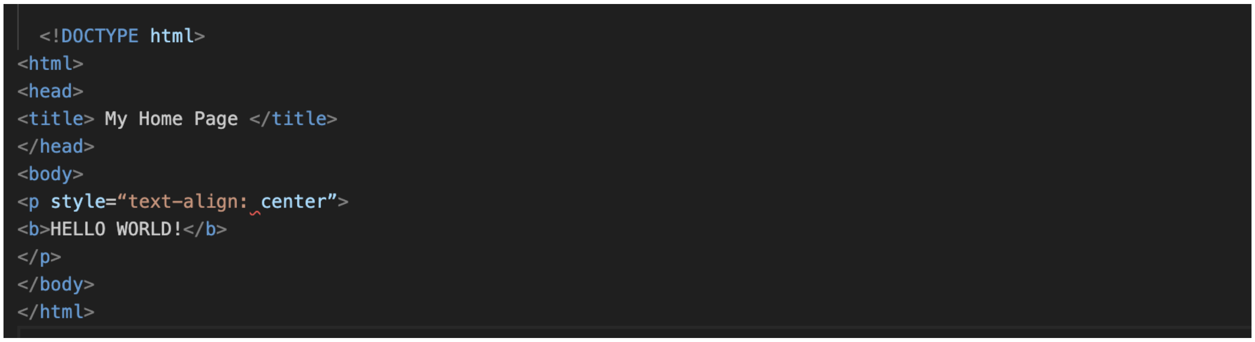
Example 5



```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
<b style="font-family:Algerian">HELLO WORLD!</b>
</body>
</html>
```

-Make centered alignment of the word “HELLO WORLD!” from example 5.

Example 6



```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
<p style=“text-align: center”>
<b>HELLO WORLD!</b>
</p>
</body>
</html>
```

Images

- Images should be in gif (graphics interchange format) or jpg/jpeg (joint photographic expert group) format to ensure cross browser compatibility.
- **png** (Portable Network Graphics) and **bmp** (bitmap image) formats are not supported by all web browsers, where as jpg and gif formats are supported by all browsers.

NOTE

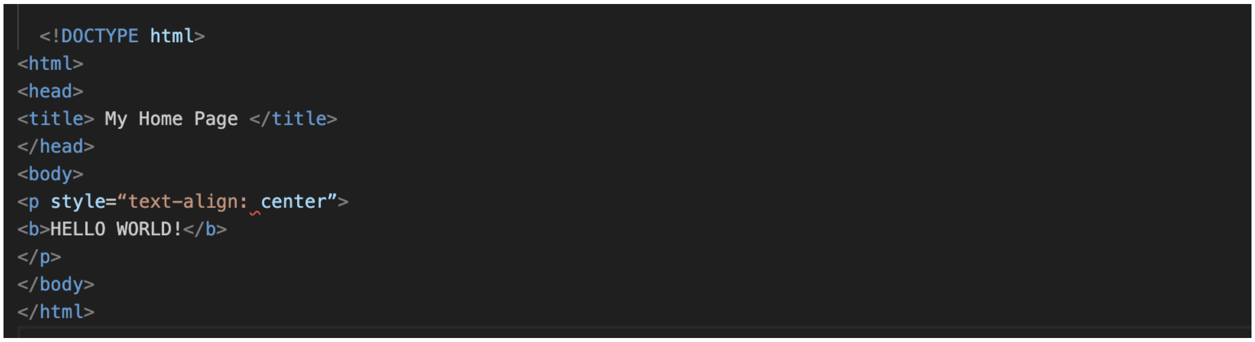
- To avoid issues with the image path, the image file should ideally be located in the same folder/directory as the HTML file itself.
- Web page can export the image and HTML file in one folder to allow for easy uploading and will automatically convert the image paths to match the exported folder.

HTML Code Examples for Image

- **<img src=“test1.gif”/>** will display the GIF file “test.gif” if that file is in the same directory as the HTML file.
- If the image file is in subdirectory/folder, then the subdirectory name must be specified. For example, if the file “test.gif” is in a subdirectory called “img” then the image tag would be: **<img src=“img/test.gif”>.**
- To ensure the file and subdirectory/folder names are exact, as some web server are case sensitive and consider IMAGE.GIF to be a different file from image.gif.

  Adding the Image to the Webpage in example 6.

Example 7



```
 <!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
<p style=“text-align: center”>
<b>HELLO WORLD!</b>
</p>
</body>
</html>
```

Links

- Links are text or images that specify or link to another page, image or other element. Text links usually appear as underlined text on web pages.

HTML Code Examples for Links

- To create a link to the HTML page “about.html” which is stored in the same directory/folder as the current page, the code link tag would be:

  Click here to go to <a href=“about.html”> About Us</a>

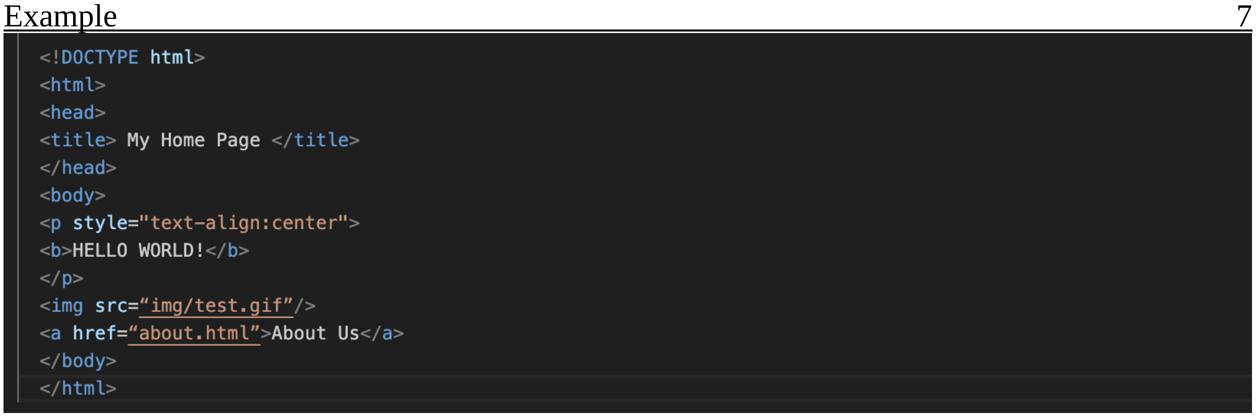
  Would appear on the page like this:

  Click here to go to About Us
- If the about.html is in a subdirectory/folder e.g the support folder, then the folder name must be specified.

  <a href=“support/about.html”>
- If about.html is stored in a higher level folder (one level up from the current folder), then the folder name would be prefixed with**.. e.g <a href=“../about.html”>**

Adding a Link to a Web page in example 7

Example 7

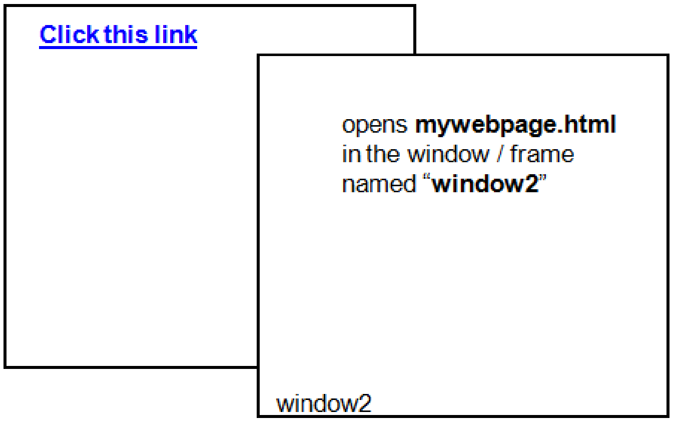


```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
<p style="text-align:center">
<b>HELLO WORLD!</b>
</p>
<img src=“img/test.gif”/>
<a href=“about.html”>About Us</a>
</body>
</html>
```

Hypertext Links

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

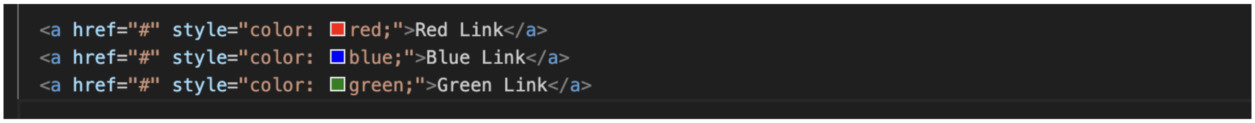
NOTE: target=“_blank” is used if you want to open the link in a new tab.


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

Hypertext Color

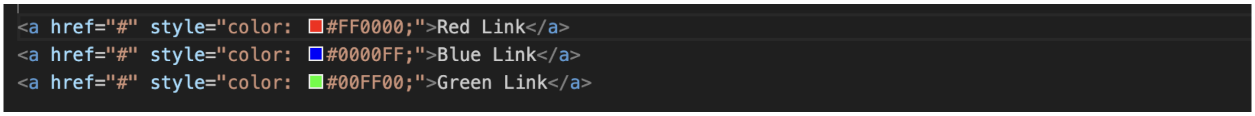
In HTML, you can specify the color of hypertext (links) using various color formats. Here are some examples:

Using Color names:



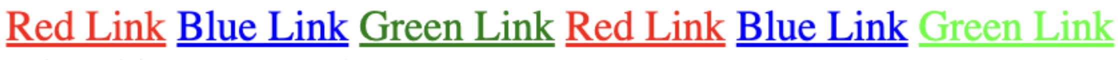
```
<a href="#" style="color: 🟥red;">Red Link</a>
<a href="#" style="color: 🟦blue;">Blue Link</a>
<a href="#" style="color: 🟩green;">Green Link</a>
```

Using hexadecimal color codes:



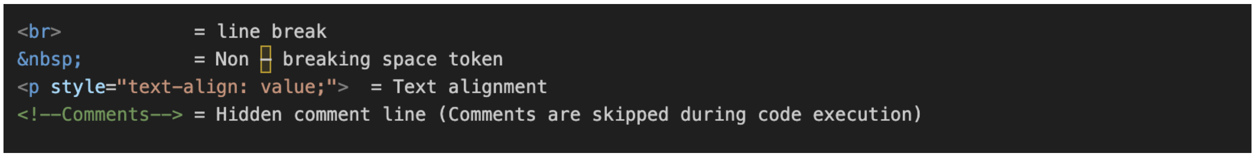
```
<a href="#" style="color: 🟥#FF0000;">Red Link</a>
<a href="#" style="color: 🟦#0000FF;">Blue Link</a>
<a href="#" style="color: 🟩#00FF00;">Green Link</a>
```

The output:



Red Link Blue Link Green Link Red Link Blue Link Green Link

Miscellaneous Formating



```
<br>             = line break
&nbsp;           = Non – breaking space token
<p style="text-align: value;">  = Text alignment
<!--Comments--> = Hidden comment line (Comments are skipped during code execution)
```

Headings

- Headings can be used to create titles in various sizes. Headings are available in 6 sizes. Renders text as a heading, the rendering depending on the level of heading selected.
- Heading should be automatically spaced from the body text.

<h1>Heading 1 level text</h1>
<h2>Heading 2 level text</h2>
<h3>Heading 3 level text</h3>
<h4>Heading 4 level text</h4>
<h5>Heading 5 level text</h5>
<h6>Heading 6 level text</h6>

Lists

- There are three lists types in HTML

i. *Unordered lists* (bulleted i.e. disc, circle or square)

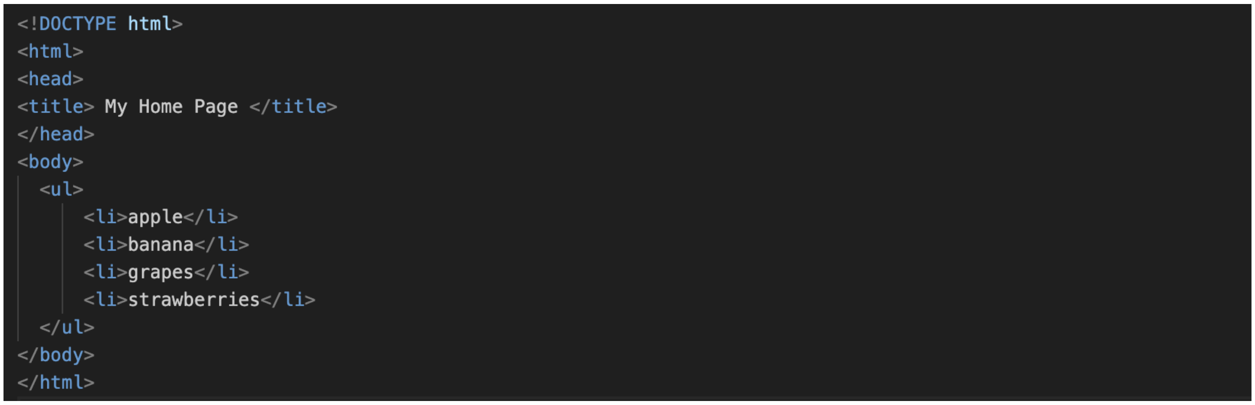
ii. *Ordered lists* (numbered or alphabetized)

iii. *Definition list* (definition format)

I. Unordered List

- Unordered list used to group a set of related items, in no particular order.
- Unordered list used one set of <ul> </ul> tag, wrapped around many sets of <li> </li>.
- Using the type of attribute, you can specify what kind of “bullet type” to display in your list. The value you can use are disc, circle or square.

Example 1.



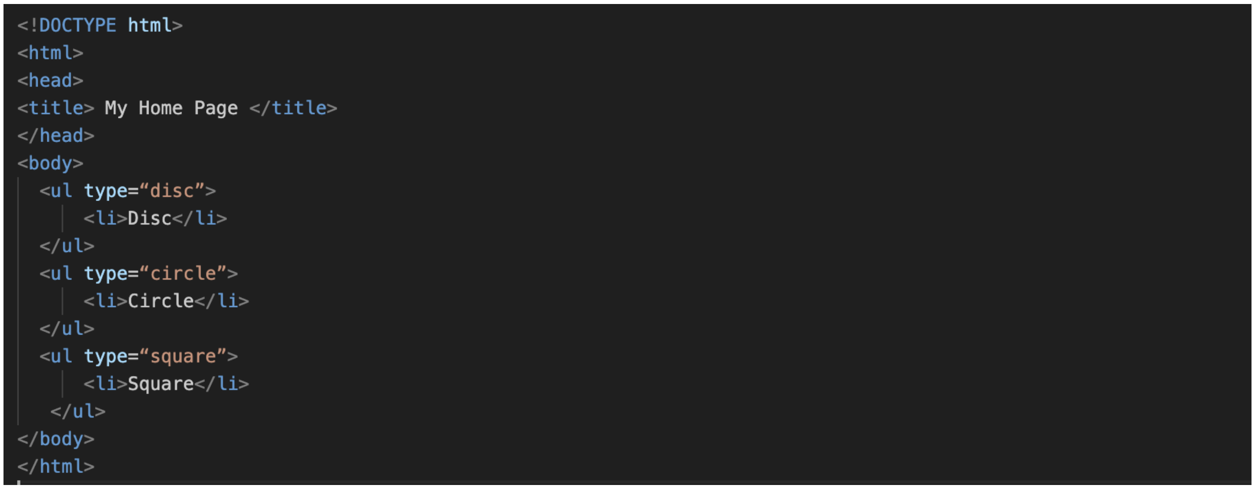
```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
  <ul>
      <li>apple</li>
      <li>banana</li>
      <li>grapes</li>
      <li>strawberries</li>
  </ul>
</body>
</html>
```

The output of the program above becomes:

- apple
- bananas
- grapes
- strawberries

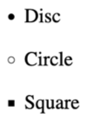
Following is an example displaying three lists containing one list item each and using different values for the type attribute.

Example 2.



```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
  <ul type=“disc”>
      <li>Disc</li>
  </ul>
  <ul type=“circle”>
      <li>Circle</li>
  </ul>
  <ul type=“square”>
      <li>Square</li>
   </ul>
</body>
</html>
```

The output of the program above becomes:



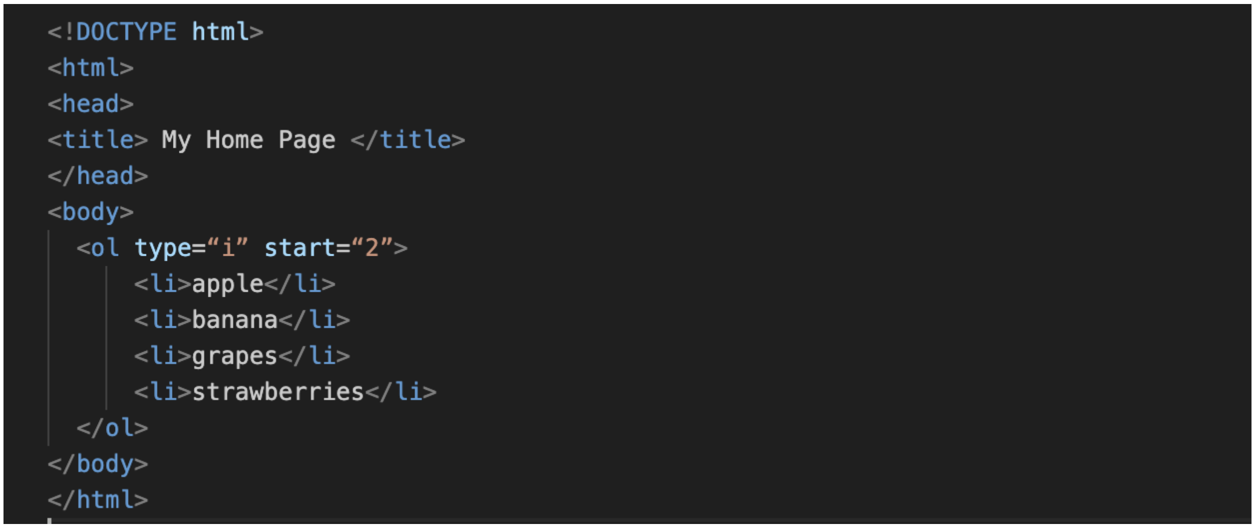
```
• Disc
○ Circle
▪ Square
```

II. Ordered List

- Ordered lists used to group a set of related items in a specific order.
- Ordered lists use one set of <ol> </ol> tags, wrapped around many sets of <li> </li>.
- You can, explicitly define the ordering style by using the type attribute in the <ol> tag. The type attribute employs one of the following values and “orders” the list according to the given table.

Value	**Ordering Style**
1	1, 2, 3, …..
i	i, ii, iii, …..
I	I, II, III, ….
a	a, b, c, ….
A	A, B, C, ….

Example 3.



```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
  <ol type=“i” start=“2”>
      <li>apple</li>
      <li>banana</li>
      <li>grapes</li>
      <li>strawberries</li>
  </ol>
</body>
</html>
```

The output of the program above becomes:

II. apple

III. bananas

IV. grapes

V. strawberries

III. Definition List

- A definition list is used to present a list of terms and their corresponding definitions. Each term is wrapped in a <dt> (definition term) tag, and each definition is wrapped in a <dd> (definition description) tag.

  NOTE
- Definition lists are different from the other kinds of list, as they use names and values instead of list items.
- You wrap a description list in one set of <dl> </dt> (name) and <dd> </dd> (value) tags.
- You must pair at least one <dt> </dt> with at least one <dd> </dd>, a <dt> </dt> should always come first in the source order.

If you are new to this type of List then you must be wondering what exactly is this. don’t worry we got you covered. Here is the light to master this type of list.

<dl>: The <dl> element stands for "definition list." It is used to define a list of terms and their corresponding definitions within an HTML document. It provides a structured way to present paired content, where each term is associated with a definition.

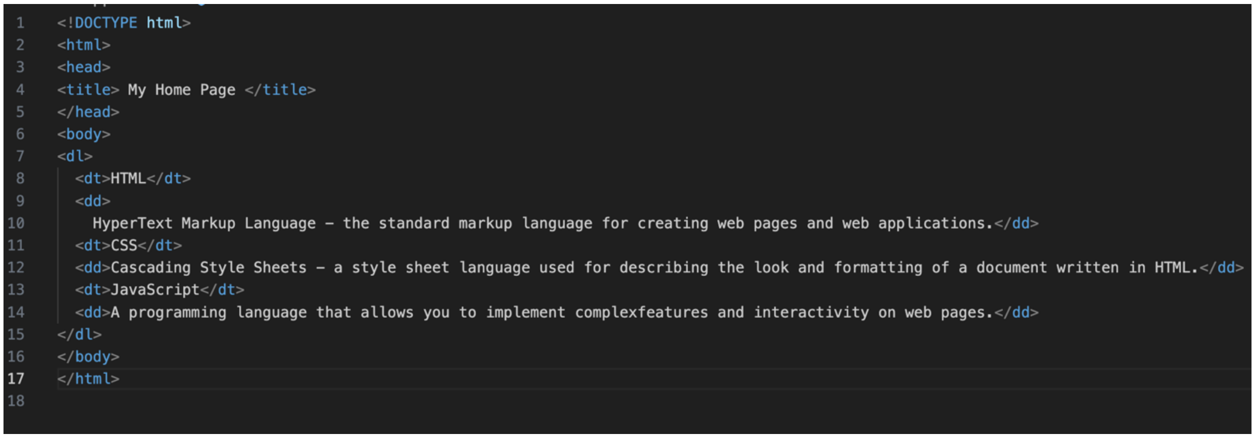
<dt>: The <dt> element stands for "definition term." It is used within a <dl> element to represent a term or a heading for a definition in the list. It typically comes before the corresponding <dd> element. It is often displayed in bold or with a distinct style to visually differentiate it from the definition.

<dd>: The <dd> element stands for "definition description." It is used within a <dl> element to represent the description or definition of a term in the list. It follows the corresponding <dt> element and provides the content or explanation for the term.

In summary, <dl> represents the entire definition list, <dt> represents a term or heading, and <dd> represents the description or definition associated with that term.

Example 4.

This example below is a simple dictionary showing the application of definition list



```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
<dl>
  <dt>HTML</dt>
  <dd>
    HyperText Markup Language - the standard markup language for creating web pages and web applications.</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets - a style sheet language used for describing the look and formatting of a document written in HTML.</dd>
  <dt>JavaScript</dt>
  <dd>A programming language that allows you to implement complexfeatures and interactivity on web pages.</dd>
</dl>
</body>
</html>
```

And here is the result.

HTML

HyperText Markup Language - the standard markup language for creating web pages and web applications.

CSS

Cascading Style Sheets - a style sheet language used for describing the look and formatting of a document written in HTML.

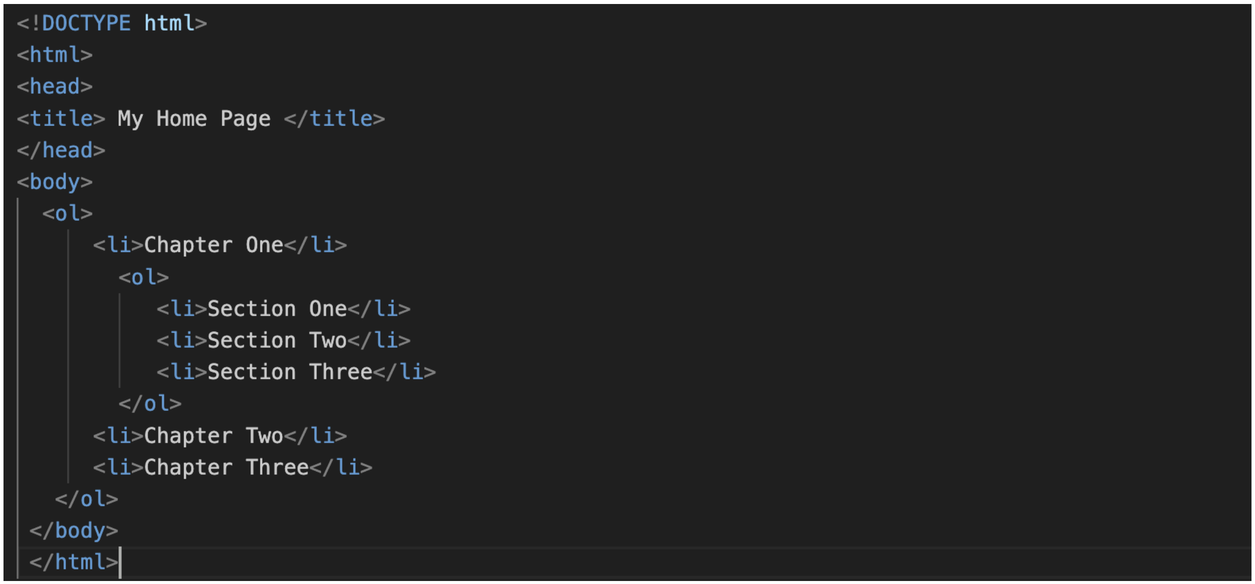
JavaScript

A programming language that allows you to implement complexfeatures and interactivity on web pages.

Nesting List

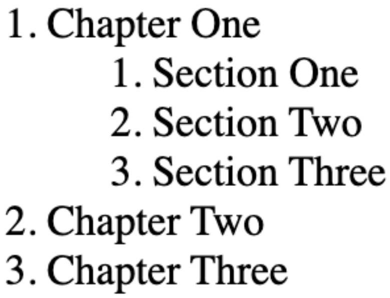
- A list item can contain another entire list. This is known as “nesting” a list. It is useful for things like tables of contents, such as the one at the start of this article.

Example 5.



```
<!DOCTYPE html>
<html>
<head>
<title> My Home Page </title>
</head>
<body>
  <ol>
      <li>Chapter One</li>
        <ol>
           <li>Section One</li>
           <li>Section Two</li>
           <li>Section Three</li>
        </ol>
      <li>Chapter Two</li>
      <li>Chapter Three</li>
   </ol>
 </body>
 </html>
```

The output of the program above becomes:



```
1. Chapter One
        1. Section One
        2. Section Two
        3. Section Three
2. Chapter Two
3. Chapter Three
```

Question

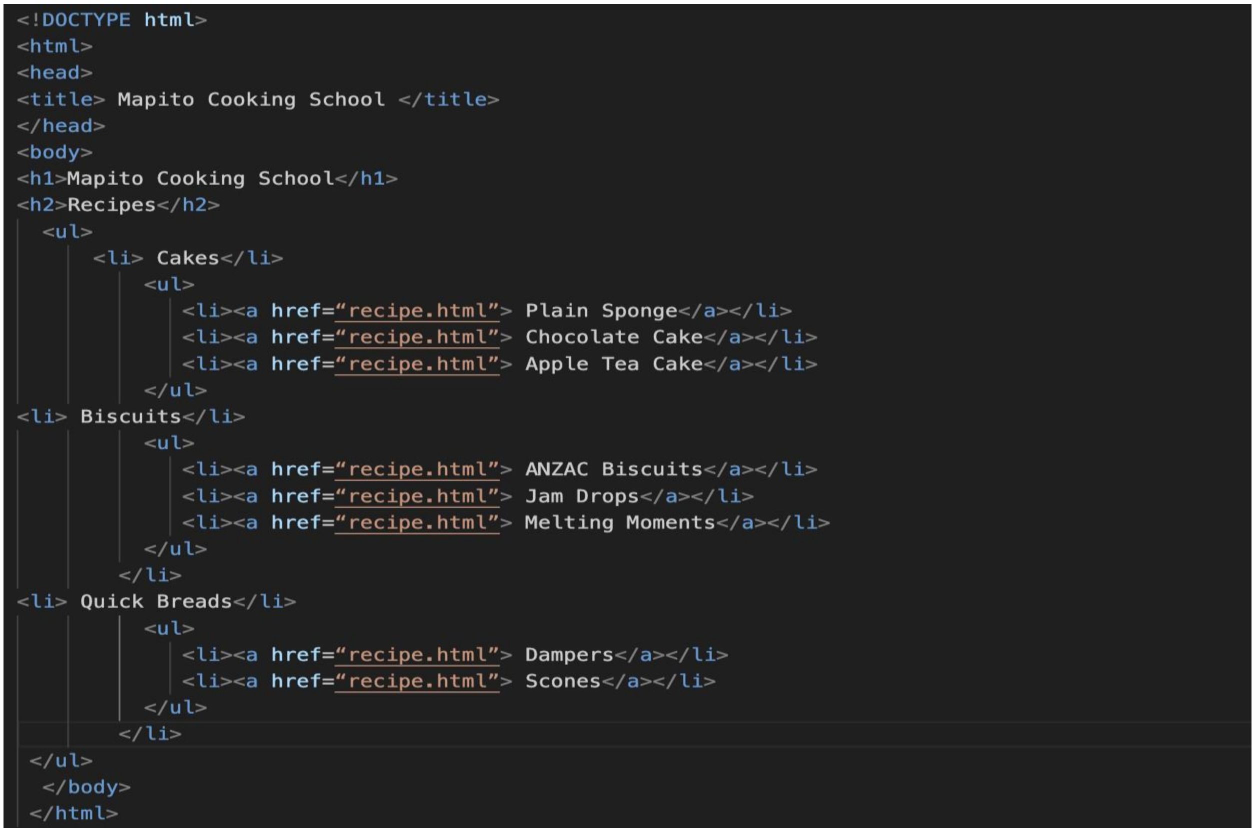
You are creating a small website for the Mapito Cooking School. On the main page you are to show a list of categorized recipes, linking through to recipe pages. Each recipe page lists the ingredients required notes on those ingredients and the preparation method. The three categories are:

- Cakes (Including recipes for Plain Sponge, Chocolate Cake and Apple Tea Cake)
- Biscuits (including recipes for ANZAC Biscuits, Jam Drops and Melting Moments)
- Quick breads (including recipes for Damper and Scones)

The client does not mind what order the categories and recipes are shown; they just want to be sure people know which items are categories and which ones are recipes.

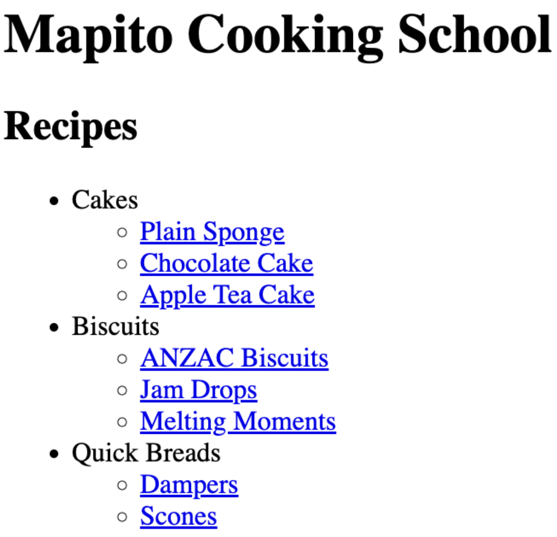
You are required to write a program for that scenario.

Solution



```
<!DOCTYPE html>
<html>
<head>
<title> Mapito Cooking School </title>
</head>
<body>
<h1>Mapito Cooking School</h1>
<h2>Recipes</h2>
  <ul>
      <li> Cakes</li>
          <ul>
             <li><a href=“recipe.html”> Plain Sponge</a></li>
             <li><a href=“recipe.html”> Chocolate Cake</a></li>
             <li><a href=“recipe.html”> Apple Tea Cake</a></li>
          </ul>
<li> Biscuits</li>
          <ul>
             <li><a href=“recipe.html”> ANZAC Biscuits</a></li>
             <li><a href=“recipe.html”> Jam Drops</a></li>
             <li><a href=“recipe.html”> Melting Moments</a></li>
          </ul>
        </li>
<li> Quick Breads</li>
          <ul>
             <li><a href=“recipe.html”> Dampers</a></li>
             <li><a href=“recipe.html”> Scones</a></li>
          </ul>
        </li>
 </ul>
  </body>
 </html>
```

The output of the program above becomes:



Mapito Cooking School

Recipes

- Cakes
  - Plain Sponge
  - Chocolate Cake
  - Apple Tea Cake
- Biscuits
  - ANZAC Biscuits
  - Jam Drops
  - Melting Moments
- Quick Breads
  - Dampers
  - Scones

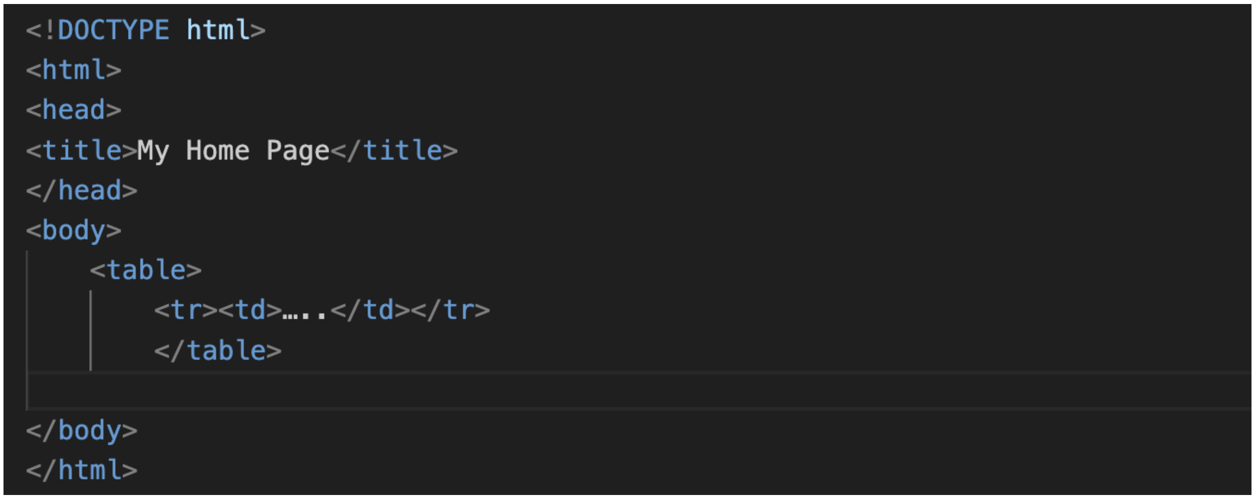
Tables

- Next to the hyperlink, one of the mostly used HTML elements is the table element. Table are created using table element which uses both a start and end tag.
- The <table>…</table> tags are required to contain at least one set of <tr>….</tr> tags which themselves are required to contain at least one set of <td>….</td> tags.
- Each set of <tr>….</tr> tags creates one table row while each set of <td>….</td> tags creates one table data cell.
- An HTML table has two kinds of cells:
    - **i. Header cells:**

      Contain header information (created with the <th> element). The text in <th> elements are bold and centered by default.
    - **ii. Data cells:**

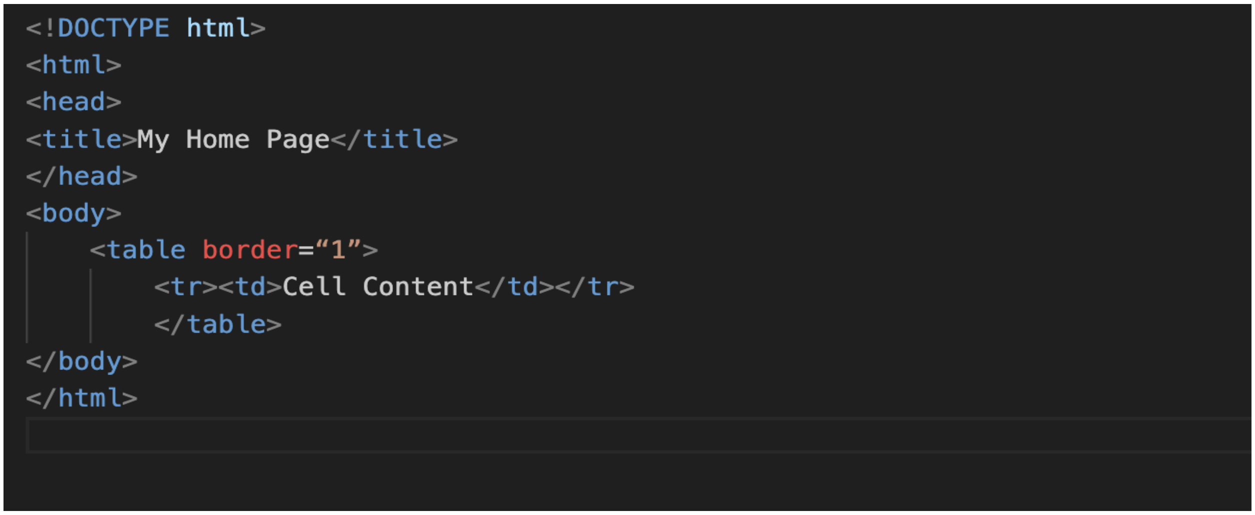
      Contains data (created with the <td> element). The text in <td> elements is regular and left aligned by default.
- Each set of <tr>…..</tr> tags creates one table row while each of set of <td>…</td> tags creates one table data cell.

Syntax formula



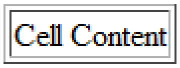
```
<!DOCTYPE html>
<html>
<head>
<title>My Home Page</title>
</head>
<body>
    <table>
        <tr><td>…..</td></tr>
        </table>

</body>
</html>
```

Example 1



```
<!DOCTYPE html>
<html>
<head>
<title>My Home Page</title>
</head>
<body>
    <table border=“1”>
        <tr><td>Cell Content</td></tr>
        </table>
</body>
</html>
```

The output of the program above displays

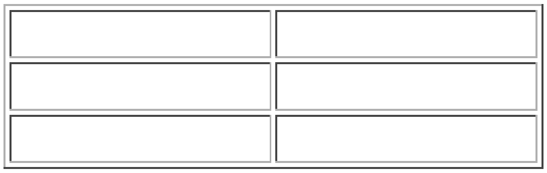


Cell Content

Example 2 creates a table having three rows and two columns with no contents.

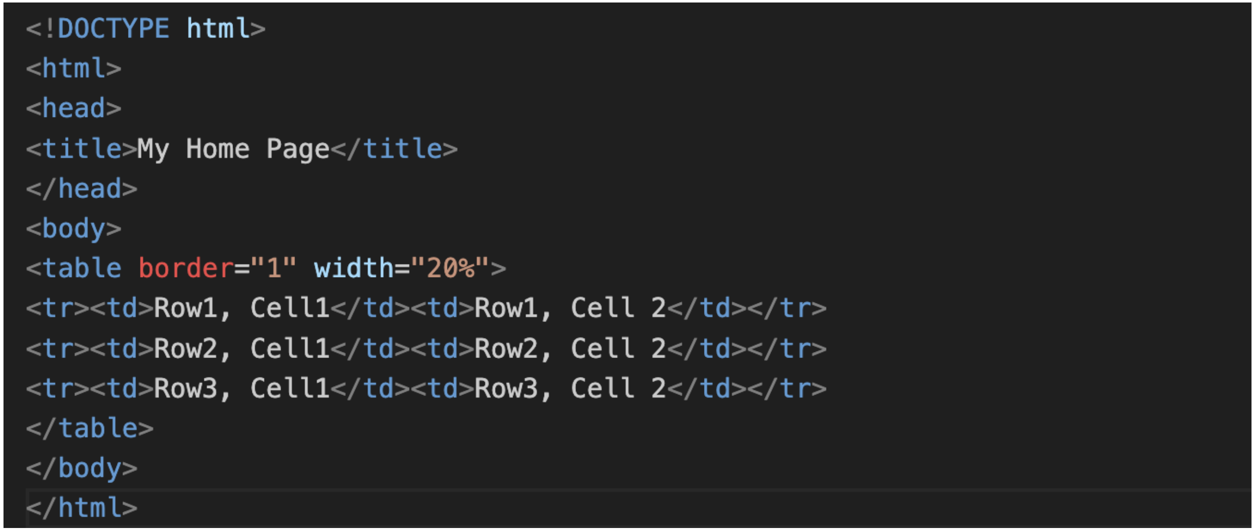
Example 2

The output of the program above displays



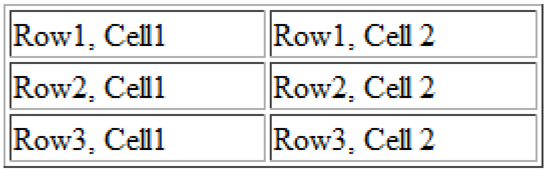
Example 3 creates a table having three rows and two columns with contents.

Example 3:



```
<!DOCTYPE html>
<html>
<head>
<title>My Home Page</title>
</head>
<body>
<table border="1" width="20%">
<tr><td>Row1, Cell1</td><td>Row1, Cell 2</td></tr>
<tr><td>Row2, Cell1</td><td>Row2, Cell 2</td></tr>
<tr><td>Row3, Cell1</td><td>Row3, Cell 2</td></tr>
</table>
</body>
</html>
```

The output of the program above displays



Row1, Cell1	Row1, Cell 2
Row2, Cell1	Row2, Cell 2
Row3, Cell1	Row3, Cell 2

There are a number of attributes which you can apply to the <table> tag to control your table. Following are attributes that can be applied to the table tag:

i. Background and border color

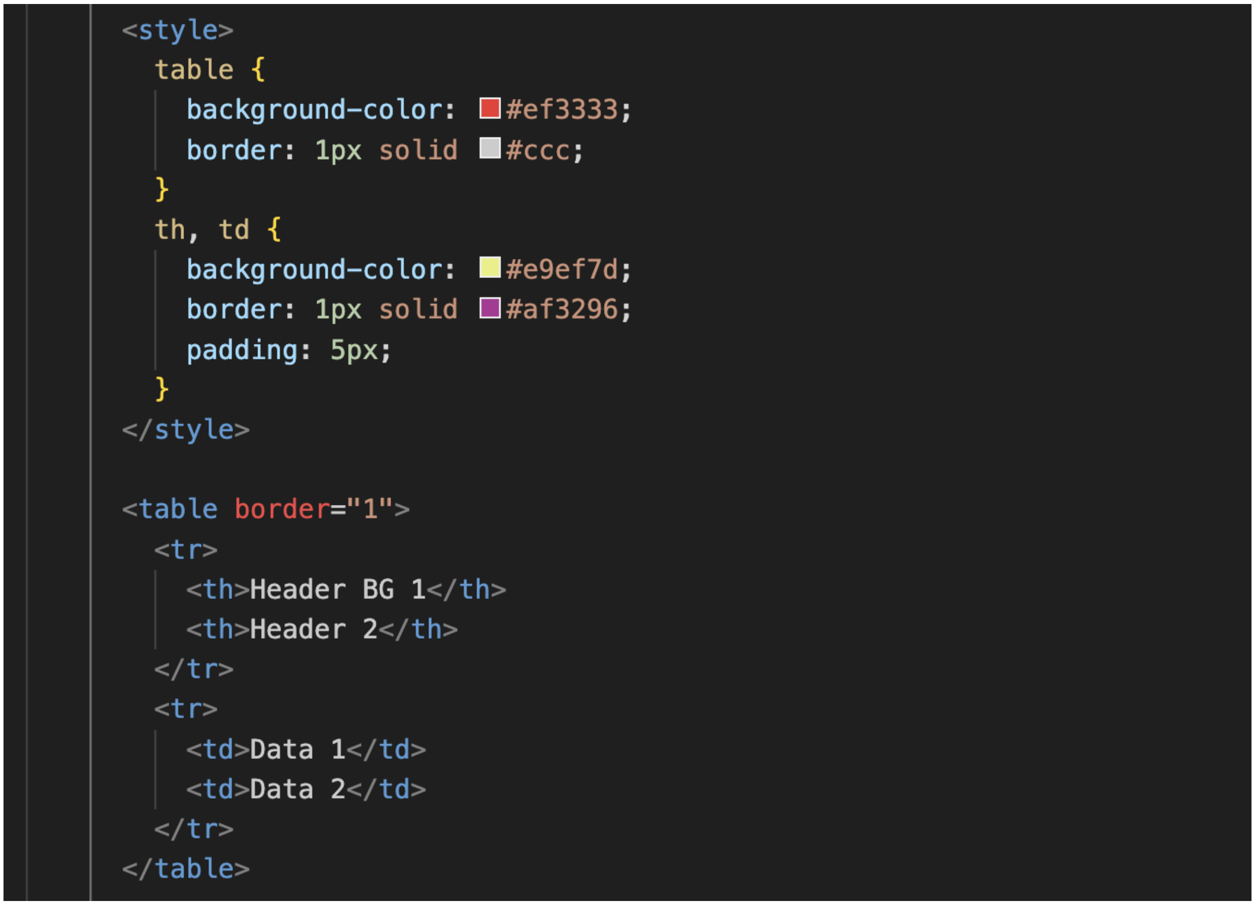
ii. Space and alignment

iii. Rowspan and colspan

iv. Width and height

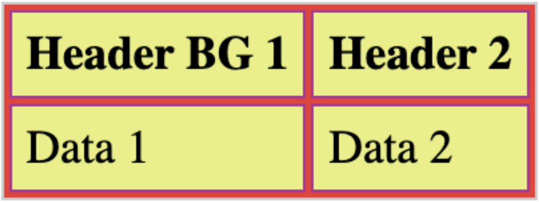
I. Table Cell: Background and Border color

Example 1



```
<style>
  table {
    background-color: #ef3333;
    border: 1px solid #ccc;
  }
  th, td {
    background-color: #e9ef7d;
    border: 1px solid #af3296;
    padding: 5px;
  }
</style>

<table border="1">
  <tr>
    <th>Header BG 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>Data 1</td>
    <td>Data 2</td>
  </tr>
</table>
```

The output of the program above displays



Header BG 1	Header 2
Data 1	Data 2

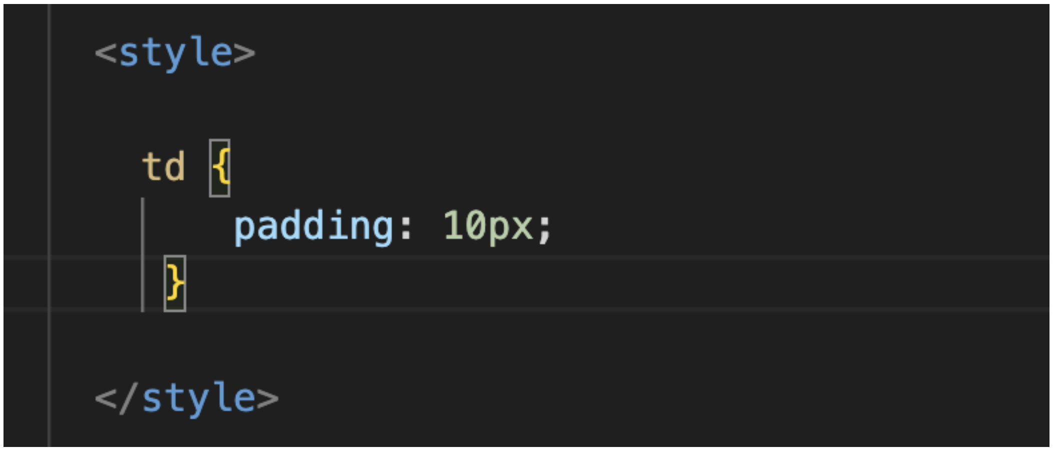
II. Table Cell: Space and Alignment

Refers to the control and arrangement of space and alignment within the cells of an HTML table. It involves adjusting the spacing between the content and the cell borders, as well as aligning the content within the cells.

Space: By applying padding to table cells (<td> or <th> elements), you can control the space between the cell content and the cell borders. Padding adds *space* inside the cell, ***pushing the content away from the borders***. This can be achieved using CSS with the padding property.

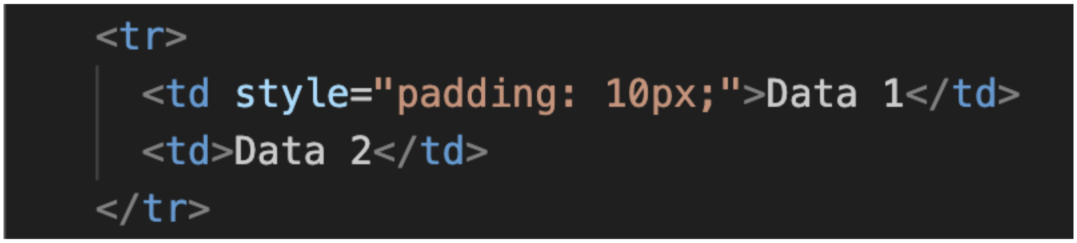
Padding: Refers to the space between the content of an element (such as text or images) and its border. Padding can either be left,right,top and bottom. If you don't specify padding direction by default it will set space in all directions. So if you write only **padding:10px** this will mean both padding-top,bottom,left and right are all set to 10px.

The property below is by using embedded css.



```
<style>

  td {
      padding: 10px;
  }

</style>
```

Here is an example using inline css.



```
<tr>
  <td style="padding: 10px;">Data 1</td>
  <td>Data 2</td>
</tr>
```

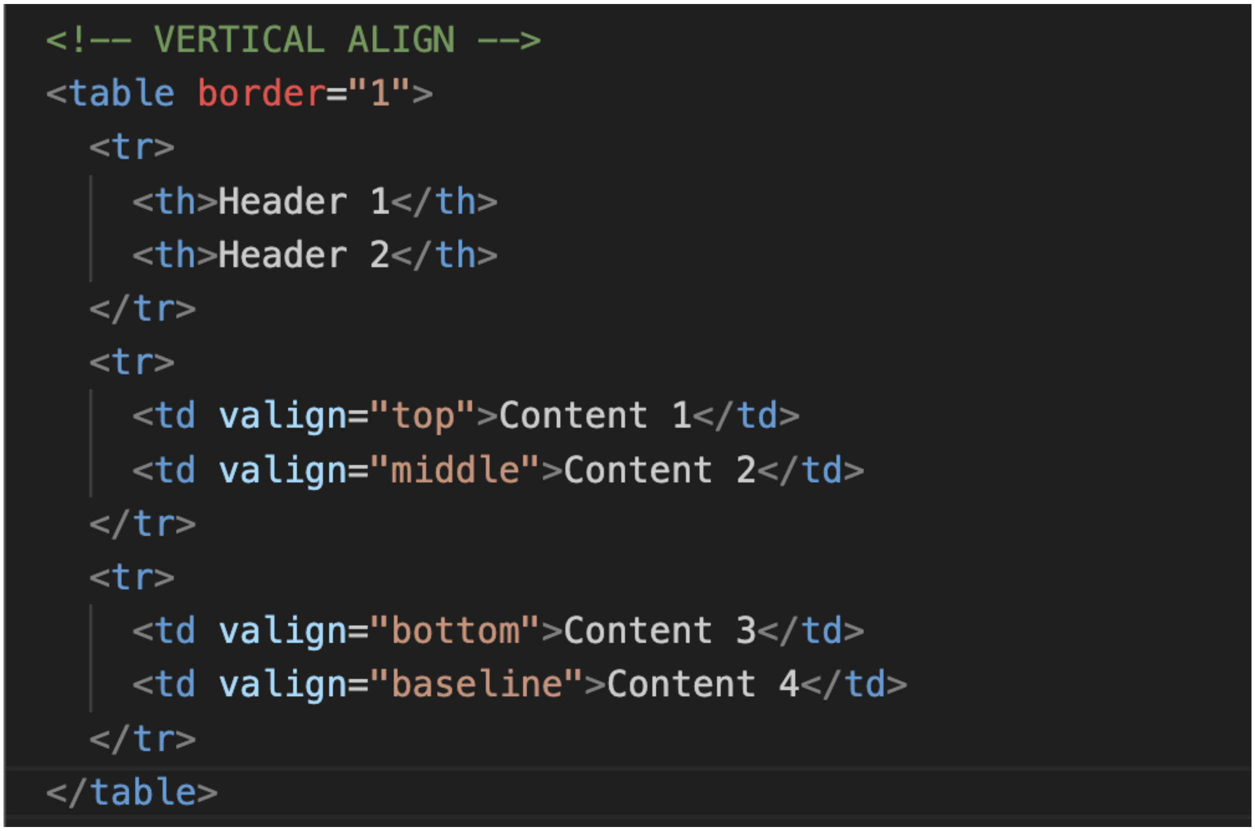
Valign =“top, middle or bottom”

- The valign attribute is used in a <td> tag to set the vertical alignment of the respective cell’s content.

Syntax formula

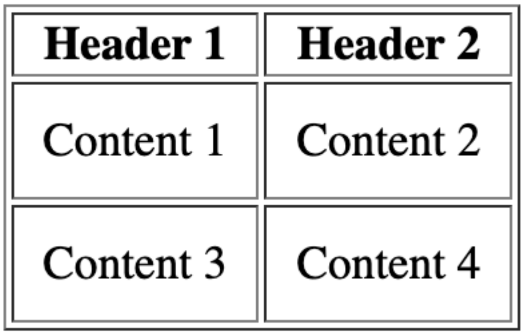
<td vlign=“position”>…..</td>

<u>Example 2</u>



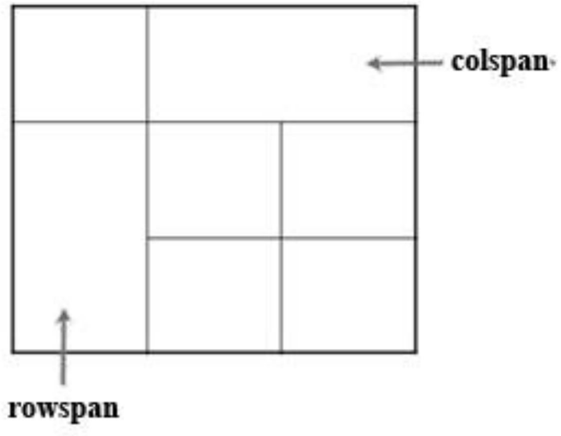
```html
<!-- VERTICAL ALIGN -->
<table border="1">
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td valign="top">Content 1</td>
    <td valign="middle">Content 2</td>
  </tr>
  <tr>
    <td valign="bottom">Content 3</td>
    <td valign="baseline">Content 4</td>
  </tr>
</table>
```

The output of the program above displays



Header 1	Header 2
Content 1	Content 2
Content 3	Content 4

III. Table Cell: Rowspan and Colspan



Syntax formula

<td colspan="number of columns" rowspan="number of rows"/>

NOTE

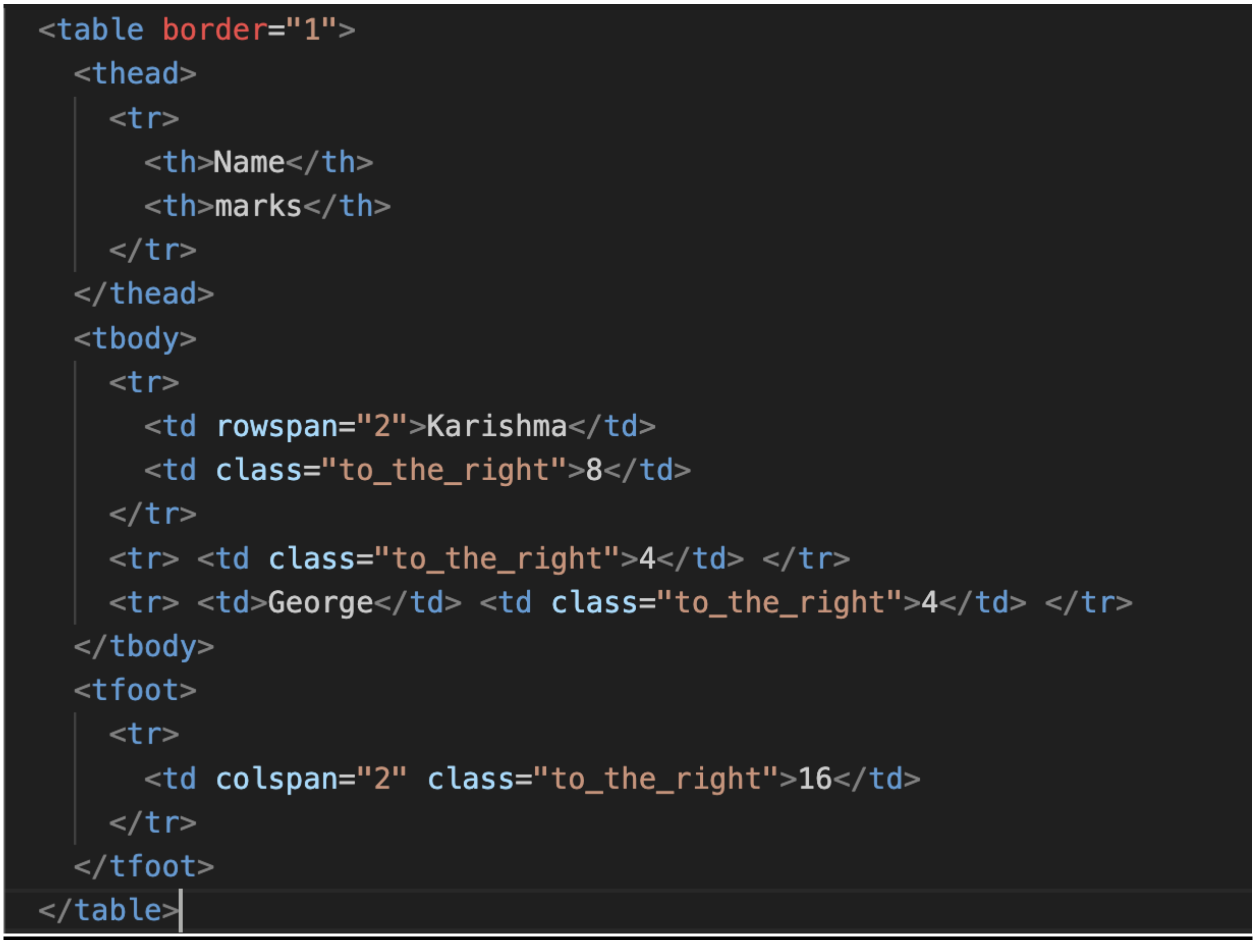
When constructing an HTML table the number of rows are established by the number of <td>…..</td> tags pairs in each row.

In example 1 below, the first row only contain one table cell where ar all the other rows contain two. Hence the table is defined as having two columns and the first row is displayed with its second cell missing.

- **Colspan:-**
  - This defines how many columns cell will span. By default, the number of columns in a table is defined is defined by the number of table data cells appearing in the table row that contains the most.

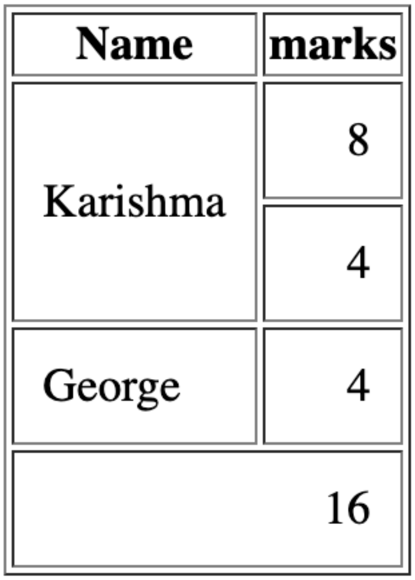
- **Rowspan:-**
  - This defines how many rows cell will span. The row span attribute works just like the colspan attribute and by using the rowspan attribute, you can force a table cell to span the number of rows specified by the respective value.

Example 1



```
<table border="1">
  <thead>
    <tr>
      <th>Name</th>
      <th>marks</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="2">Karishma</td>
      <td class="to_the_right">8</td>
    </tr>
    <tr> <td class="to_the_right">4</td> </tr>
    <tr> <td>George</td> <td class="to_the_right">4</td> </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="2" class="to_the_right">16</td>
    </tr>
  </tfoot>
</table>
```

The output of the program above displays



Name	marks
Karishma	8
	4
George	4
	16

In the following example, the “MONDAY” table cell spreads over three table rows.

Example 2

<!DOCTYPE html>

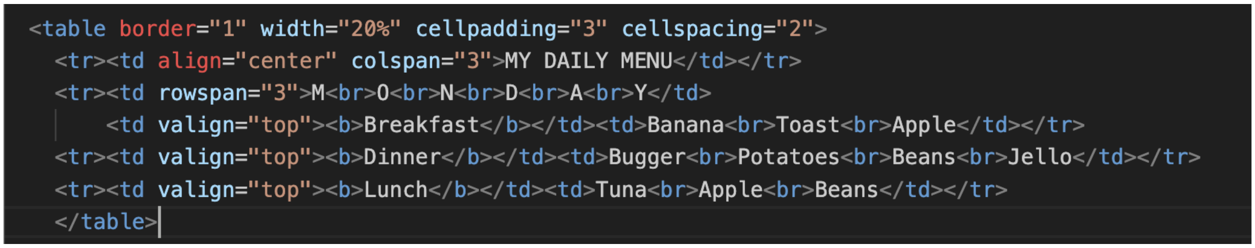
<html>

<head>

<title>My Home Page</title>

</head>

<body>

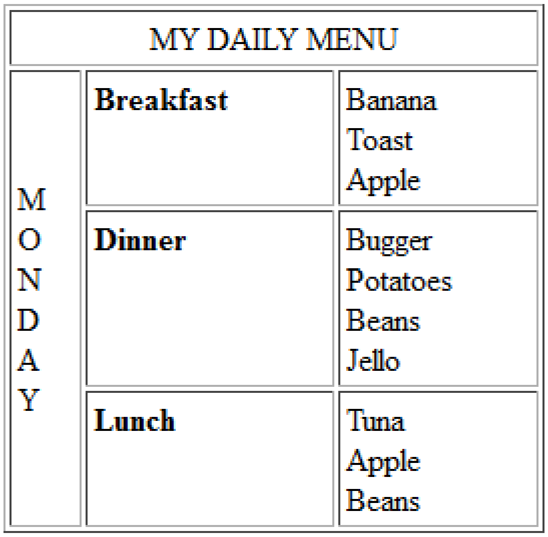


```
<table border="1" width="20%" cellpadding="3" cellspacing="2">
  <tr><td align="center" colspan="3">MY DAILY MENU</td></tr>
  <tr><td rowspan="3">M<br>O<br>N<br>D<br>A<br>Y</td>
      <td valign="top"><b>Breakfast</b></td><td>Banana<br>Toast<br>Apple</td></tr>
  <tr><td valign="top"><b>Dinner</b></td><td>Bugger<br>Potatoes<br>Beans<br>Jello</td></tr>
  <tr><td valign="top"><b>Lunch</b></td><td>Tuna<br>Apple<br>Beans</td></tr>
  </table>
```

</body>

</html>

The output of the program above displays



MY DAILY MENU		
M O N D A Y	**Breakfast**	Banana Toast Apple
	Dinner	Bugger Potatoes Beans Jello
	Lunch	Tuna Apple Beans

IV. Table Cell: Width and Height

You can apply width and height attribute to your <td> tags to explicitly set the dimension of your table cells and thus exercise control over the width and height of rows and columns.

- **Width="pixels or percentage":-**

  Using the width attribute in your <td> tags, you may define either a fixed width in pixels or a relative width (recommended) which is equal to a percentage of the table width.

Example 1 below renders a table that is 500 pixels wide and has two columns. The width of the first table data cell is set to 25%. The width of all the cells in column 2 default to the remainder of tables width (75%)

Example 1

<!DOCTYPE html>

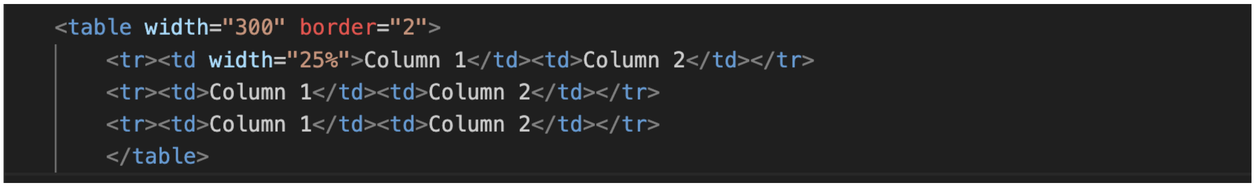
<html>

<head>

<title>My Home Page</title>

</head>

<body>

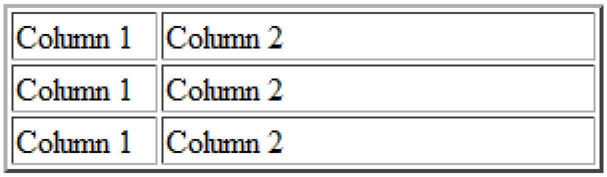


```
<table width="300" border="2">
    <tr><td width="25%">Column 1</td><td>Column 2</td></tr>
    <tr><td>Column 1</td><td>Column 2</td></tr>
    <tr><td>Column 1</td><td>Column 2</td></tr>
    </table>
```

</body>

</html>

The output of the program above displays



Column 1	Column 2
Column 1	Column 2
Column 1	Column 2

- **Height="pixel or percentage":-**

  Using the height attribute in your <td> tags, you may define either a fixed height in pixel (recommended) or a relative height which is equal to a percentage of the table height.

Example 2 renders a table that is 300 pixels wide and has two rows. The height of the first table cell in Row 1 is set to 100 pixels which then sets the height of all cells in Row 1 to 100 pixels.

- The height of Row 2 is not affected by the height specification in Row1.

Example 2

<!DOCTYPE html>

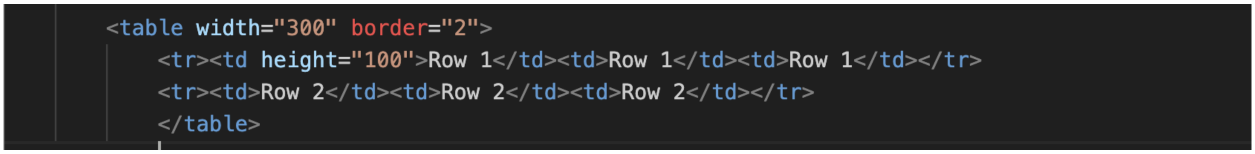
<html>

<head>

<title>My Home Page</title>

</head>

<body>

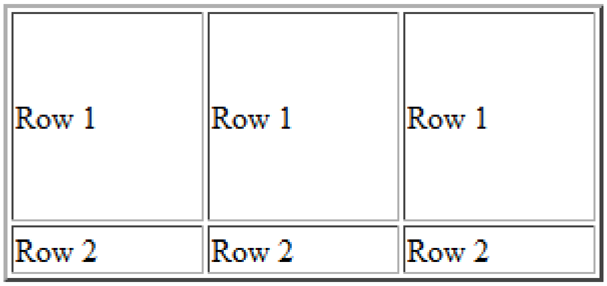


```
<table width="300" border="2">
    <tr><td height="100">Row 1</td><td>Row 1</td><td>Row 1</td></tr>
    <tr><td>Row 2</td><td>Row 2</td><td>Row 2</td></tr>
    </table>
```

</body>

</html>

The output of the program above displays



Row 1	Row 1	Row 1
Row 2	Row 2	Row 2

FORMS

- The HTML <form> element defines a form that is used to collect user input. An HTML form contains form element.

  Syntax form tag element

  <form>

  ..form element…

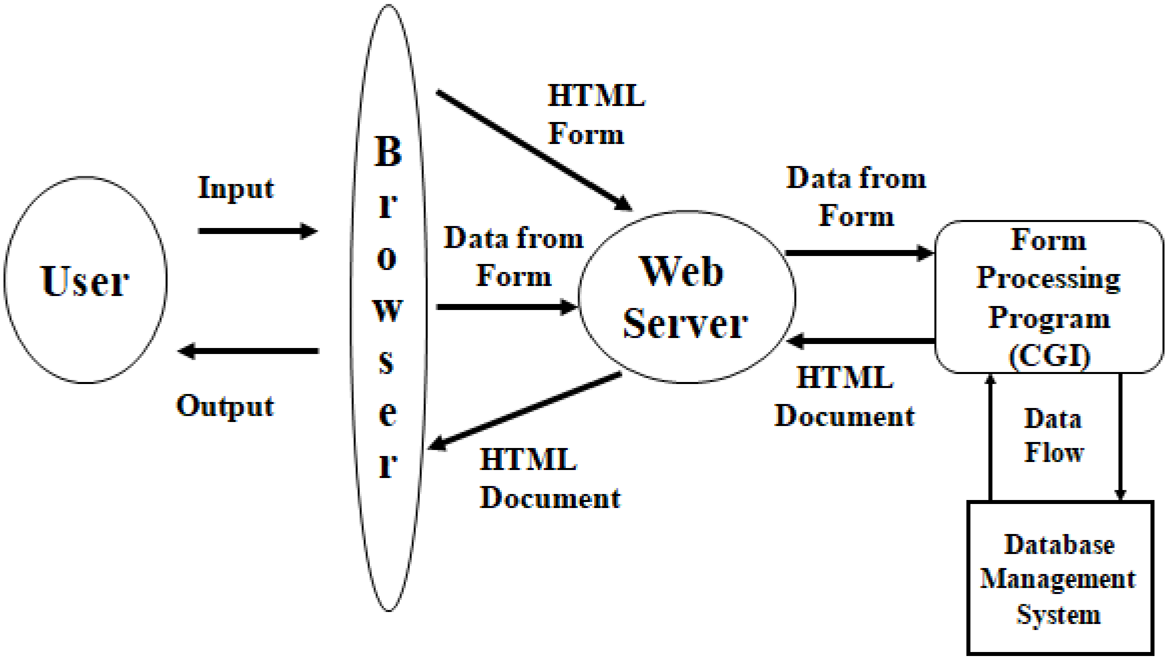
  </form>

- Form elements are different types of input elements, like text field, check boxes, radio buttons, submit buttons and more.

The <input> Elements and Input tag

The <**input**> **tag** specifies an **input field** where the **user** can enter data. <**input**> elements are used within a <**form**> element to declare **input** controls that allow **users** to **input** data. An **input field** can vary in many ways, depending on the type attribute.

Form Processing



Form Input: INPUT

Only used within a FORM element and is denoted by <INPUT>.

<u>ATTRIBUTES:</u>

- *NAME*: The name ‘of the particular element.
- *MAXLENGTH*: The maximum number of characters that can be entered by users in a text field.
- *SIZE*: Specifies the size of the field and depends on its type.
- *SRC*: Denote URL for an image.
- *VALUE*: Contain the initial value displayed to users.
- *TYPE*: Defines the type of data used in the field.
- *CHECKED*: Indicates that a checkbox or radio button is selected.
- *DISABLED*: Prevents the field from receiving focus.
- *ALIGN*: Alignment if image is used.
- *READONLY*: Prevents modification of the contents of the field.

Examples of input element include:

i. <u>Text Input (type= “text”)</u>

- This defines a one- line text input field for text input.
- Can be used with the MAXLENGTH and SIZE attributes (MAXLENGTH >= SIZE)

Syntax text input tag element

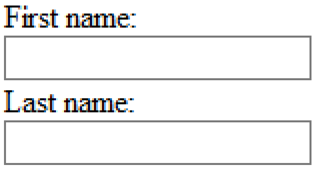
<input type=”text” name= “ ” maxlenght= “” size= “”>

<u>Example 1</u>

```
<!DOCTYPE html>
<html>
<head><title>My Text Field</title></head><br>
<body>
<form>
First name:<br>
<input type= “text” name=“firstname”><br>
Last name:<br>
```

```
<input type=“text” name= “lastname”>
</form>
</body>
</html>
```

The output of the program above displays



First name:

Last name:

NOTE: The default width of a text field is 20 characters.

Example 1: A text field named "text1" that is 30 characters wide.

```
<input type="text" name="text1" size="30">
```

Output:

Example 2: A text field named "text2" that is 30 characters wide but will only accept 20 characters.

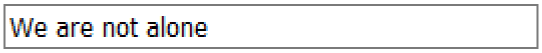
```
<input type="text" name="text2" size="30" maxlength="20">
```

Output:

Example 3: A text field named "text3" that is 40 characters wide with default value.

```
<input type="text" name="text3" size="40" value="We are not alone">
```

Output:



We are not alone

ii. **Text Input (type= “password”)**

Are exactly the same as text input elements, except that when the viewer types in, they see "bullet" characters rather than the letter they are typing. Password text is scrambled during transmission and then unscramble when the form data is received at the server end.

Example 4: A password field named "pass1" that is 30 characters wide

```
<input type="password" name="pass1" size="30">
```

Output:

Example 5: A password field named "pass2" that is 30 characters wide but will only accept 20 characters

```
<input type="password" name="pass2" size="30" maxlength="20" value="MyHo">
```

Output:

iii. Radio Button Input

This defines a radio button (for selecting one of many choices). Radio buttons let a user select one of a limited number of choices.

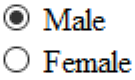
Syntax text input tag element

<input type=”radio” name= “ ” value=“ ”>

Example 2

```
<!DOCTYPE html>
<html>
<head><title>My Radio Buttons</title></head><br>
<body>
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
</form>
</body>
</html
```

The output of the program above displays



```
● Male
○ Female
```

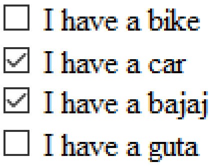
iv. Check Boxes Input

- This defines a check box (for selecting one of many choices). Check boxes let a user select one or more number of choices.

Example 2

```
<!DOCTYPE html>
<html>
<head><title>My Radio Buttons</title></head><br>
<body>
<form>
 <input type="checkbox" name="vehicle" value="bike" > I have a bike<br>
<input type="checkbox" name="vehicle" value="car" checked> I have a car<br>
<input type="checkbox" name="vehicle" value="bajaj" checked> I have a bajaj<br>
<input type="checkbox" name="vehicle" value="guta" checked> I have a guta<br>
</form>
</body>
</html>
```

The output of the program above displays



```
□ I have a bike
☑ I have a car
☑ I have a bajaj
□ I have a guta
```

v. The Submit Button

- This define a submit button (for submitting the form data to a form handler). The form handler is typically a server page with a script for processing input data.
- The form handler is specified in the form's action attribute.

Example 3

```
<!DOCTYPE html>
```

```
<html>
<head><title>My Radio Buttons</title></head><br>
<body>
<form>
 <input type="submit" value="Submit" >
</body>
</html>
```

The output of the program above displays

The Action Attribute

- The action attribute defines the action to be performed when the form is submitted. The value of the action attribute contains the URL to the script that will process the form.
- Normally, the form data is sent to a web page on the server when the user clicks on the submit button.

  Syntax tag element

  <form action=”actionpage.php”>

- The form data is sent to a [age on the server called “actionpage.php”. This page contains a server-side script that handles the form data.
- If the action attribute is omitted, the action is set to the current page.

The Method Attribute

The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data.

Syntax tag element

<form action=”actionpage.php” method=“GET”>

OR

<form action=”actionpage.php” method=“POST”>

<u>When to Use GET?</u>

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.

i.e

actionpage.php?firdstname=Sonia&lastname=George

NOTE

GET must not be used when sending sensitive information. GET is best suited for short, non-sensitive of data, because it has size limitation too.

When to Use POST?

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.

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

The Name Attribute

Each input field must have a more attribute to be submitted. If the name attribute is omitted, the data of that input field will not be sent at all.

Grouping Form Data with <fieldset>

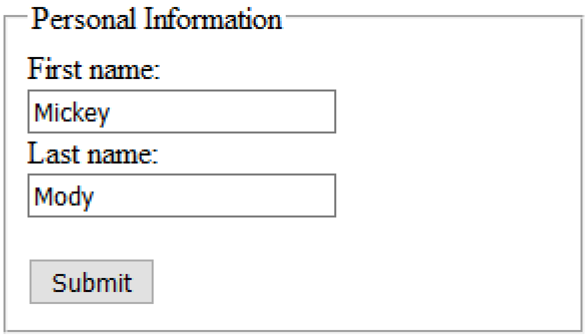
The <fieldset> element is used to group related data in a form. The <legend>. Element defines a caption for the <fieldset> element.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>My Field Set</title>
</head>
<body>
<table width="300">
<tr><td>
<form action="actionpage.php" method="POST">
<fieldset>
<legend>Personal Information</legend>
First name:<br>
```

```
        <input type="text" name="firstname" value="Mickey"><br>
        Last name:<br>
        <input type="text" name="lastname" value="Mody"><br><br>
        <input type="submit" value="Submit">
        </fieldset></td></tr></table>
        </form>
        </body>
        </html>
```

The output of the program above displays



Personal Information

First name: Mickey

Last name: Mody

Text Area

The text area element creates a multi-line text box. It require both a start tag <textarea> and end tag </textarea> and the rows and cols attribute are used in the start tag to define its display size.

Syntax tag element

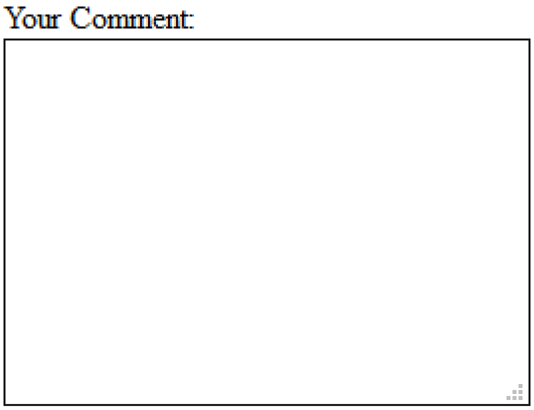
<textarea></textarea>

Example

```
<!DOCTYPE html>
<html>
<head>
<title>My Field Set</title>
</head>
<body>
<form action="actionpage.php" method="POST">
Your Comment:<br>
<textarea name="comments" rows="10" cols="30"></textarea>
```

```
</form>
</body>
</html>
<textarea name=“comments” rows=“15” cols=“50”></textarea>
```

The output of the program above displays



Your Comment: