**HTML, CSS AND GRID960**

**Introduction**

This is a document which will give you a brief introduction about the web development technologies like HTML5, CSS3 and GRID960.

If you are new in the field of web designing, you might be searching the different sources from where can you study the basics concept of designing a website.

So you may consider this as one of efficient source, which will give you a full 360 view that how we professionally built a website.

This will help you to build a website with ease. Besides of paying anyone, it will make you capable to build a website of your own choice.

By this you will be able to form a base form new technologies like BOOTSTRAP4.

**Index**

|  |  |  |
| --- | --- | --- |
| Sno. | Topic | Page no |
| 1 | Introduction | 2 |
| 2 | Environment Setup | 4 |
| 3 | HTML | 5-14 |
| 4 | CSS | 15-21 |
| 5 | Grid960 | 22-25 |

**2.Environment Setup**

Before starting with the concepts lets first set the environment for web development. You may need the following to work:-

1)**Text Editor**-Notepad and Notepad++ will work fine but we generally use Sublime Text 3 or Atom or Brackets. These are all text editor but the provide content assist which will save your time by typing the code and you can probably focus more on the designing part .

Link for Sublime Text 3: <https://www.sublimetext.com/3>

Link for Atom: <https://atom.io/>

Link for Brackets: <http://brackets.io/>

2)**(Optional)A temporary server**-You may run your code on a browser like Chrome and firefox but if you will test on a server you will get a better way to test your application. For that you may install NodeJs.

Link to install NodeJs for windows: <https://nodejs.org/en/>

For Linux and Mac, run the following command-

sudo apt-get update

sudo apt-get install nodejs

sudo apt-get install npm

Now all the users can check whether NodeJs has been installed successfully by using the commands-

node —version

npm —version

If both are working fine then you may need to install a http server. You cna do that by writing the following command-

sudo npm install -g http-server

And inside the working folder type the following command to initiate a server-

http-server

3)**Grid960 Files**-Go to <https://960.gs/> and download the the zip file from the download option at the left top. Extract it and copy the the files 960.css, reset.css, text.css present inside code->css and paste in to a seprete folder and name that folder as 960.

**Now you are all set to work with web develpment.**

**3.HTML**

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.It is used to define the structure of a webpage and to format he web pages with the help of different elements that it contain.

HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special tags or codes that a web browser uses to interpret and display information on your computer screen. HTML stands for Hyper Text Markup Language. The markup tags tell the Web browser how to display the page. An HTML file must have a html file extension

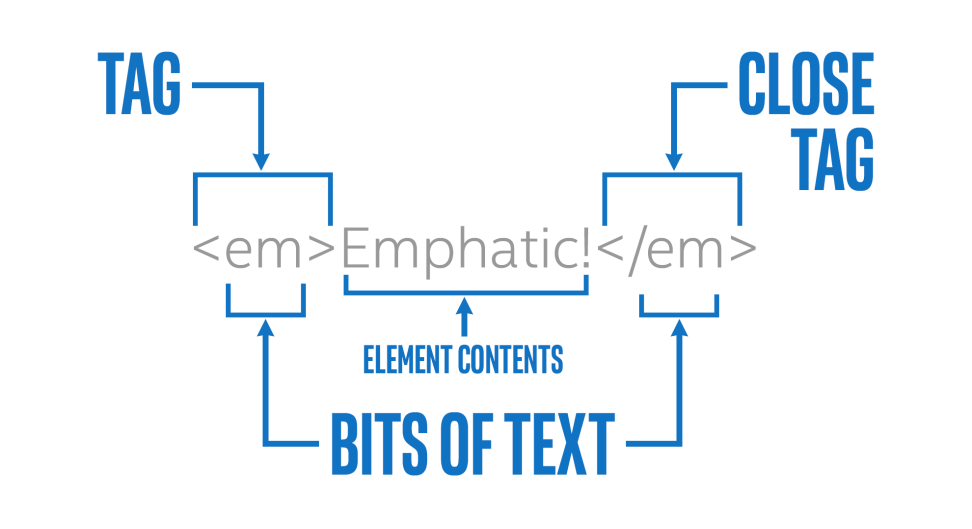
Tag are the main component of any html document. Each tag has a defined starting and ending. There are large number of tags that we can use accordingly.

**3.1 HTML tags:**

* HTML tags are used to mark-up HTML elements. HTML tags are surrounded by the two characters < and > .
* The surrounding characters are called angle brackets.
* The first tag in a pair is the start tag, the second tag is the end tag.
* The text between the start and end tags is the element content**.**

**3.2 HTML element:**

Many people have a confusion that tag and element are same thing but they have a difference. Tags are just a starting and closing tags, but element includes starting and ending tags including the content inside them.



**3.3 HTML Tags List**

|  |  |
| --- | --- |
| Tag | Description |
| [**<!--...-->**](https://www.quackit.com/html_5/tags/html_comments_tag.cfm) | Specifies a comment |
| [**<!DOCTYPE>**](https://www.quackit.com/html_5/tags/html_doctype_tag.cfm) | Specifies the document type |
| [**<a>**](https://www.quackit.com/html_5/tags/html_a_tag.cfm) | Specifies a hyperlink |
| [**<article>**](https://www.quackit.com/html_5/tags/html_article_tag.cfm) | Specifies an article |
| [**<aside>**](https://www.quackit.com/html_5/tags/html_aside_tag.cfm) | Specifies content aside from the page content |
| [**<audio>**](https://www.quackit.com/html_5/tags/html_audio_tag.cfm) | Specifies sound content |
| [**<b>**](https://www.quackit.com/html_5/tags/html_b_tag.cfm) | Specifies bold text |
| [**<blockquote>**](https://www.quackit.com/html_5/tags/html_blockquote_tag.cfm) | Specifies a long quotation |
| [**<body>**](https://www.quackit.com/html_5/tags/html_body_tag.cfm) | Specifies the body element |
| [**<br>**](https://www.quackit.com/html_5/tags/html_br_tag.cfm) | Inserts a single line break |
| [**<button>**](https://www.quackit.com/html_5/tags/html_button_tag.cfm) | Specifies a push button |
| [**<canvas>**](https://www.quackit.com/html_5/tags/html_canvas_tag.cfm) | Define graphics |
| [**<dd>**](https://www.quackit.com/html_5/tags/html_dd_tag.cfm) | Specifies a definition description |
| [**<del>**](https://www.quackit.com/html_5/tags/html_del_tag.cfm) | Specifies deleted text |
| [**<div>**](https://www.quackit.com/html_5/tags/html_div_tag.cfm) | Specifies a section in a document |
| [**<dl>**](https://www.quackit.com/html_5/tags/html_dl_tag.cfm) | Specifies a definition list |
| [**<dt>**](https://www.quackit.com/html_5/tags/html_dt_tag.cfm) | Specifies a definition term |
| [**<em>**](https://www.quackit.com/html_5/tags/html_em_tag.cfm) | Specifies emphasized text |
| [**<fieldset>**](https://www.quackit.com/html_5/tags/html_fieldset_tag.cfm) | Specifies a fieldset |
| [**<footer>**](https://www.quackit.com/html_5/tags/html_footer_tag.cfm) | Specifies a footer for a section or page |
| [**<form>**](https://www.quackit.com/html_5/tags/html_form_tag.cfm) | Specifies a form |
| [**<h1>**](https://www.quackit.com/html_5/tags/html_h1_tag.cfm) | Specifies a heading level 1 |
| [**<h2>**](https://www.quackit.com/html_5/tags/html_h2_tag.cfm) | Specifies a heading level 2 |
| [**<h3>**](https://www.quackit.com/html_5/tags/html_h3_tag.cfm) | Specifies a heading level 3 |
| [**<h4>**](https://www.quackit.com/html_5/tags/html_h4_tag.cfm) | Specifies a heading level 4 |
| [**<h5>**](https://www.quackit.com/html_5/tags/html_h5_tag.cfm) | Specifies a heading level 5 |
| [**<h6>**](https://www.quackit.com/html_5/tags/html_h6_tag.cfm) | Specifies a heading level 6 |
| [**<head>**](https://www.quackit.com/html_5/tags/html_head_tag.cfm) | Specifies information about the document |
| [**<hr>**](https://www.quackit.com/html_5/tags/html_hr_tag.cfm) | Specifies a horizontal rule |
| [**<html>**](https://www.quackit.com/html_5/tags/html_html_tag.cfm) | Specifies an html document |
| [**<i>**](https://www.quackit.com/html_5/tags/html_i_tag.cfm) | Specifies italic text |
| [**<iframe>**](https://www.quackit.com/html_5/tags/html_iframe_tag.cfm) | Specifies an inline sub window (frame) |
| [**<img>**](https://www.quackit.com/html_5/tags/html_img_tag.cfm) | Specifies an image |
| [**<input>**](https://www.quackit.com/html_5/tags/html_input_tag.cfm) | Specifies an input field |
| [**<label>**](https://www.quackit.com/html_5/tags/html_label_tag.cfm) | Specifies a label for a form control |
| [**<legend>**](https://www.quackit.com/html_5/tags/html_legend_tag.cfm) | Specifies a title in a fieldset |
| [**<li>**](https://www.quackit.com/html_5/tags/html_li_tag.cfm) | Specifies a list item |
| [**<link>**](https://www.quackit.com/html_5/tags/html_link_tag.cfm) | Specifies a resource reference |
| [**<meta>**](https://www.quackit.com/html_5/tags/html_meta_tag.cfm) | Specifies meta information |
| [**<ol>**](https://www.quackit.com/html_5/tags/html_ol_tag.cfm) | Specifies an ordered list |
| [**<option>**](https://www.quackit.com/html_5/tags/html_option_tag.cfm) | Specifies an option in a drop-down list |
| [**<output>**](https://www.quackit.com/html_5/tags/html_output_tag.cfm) | Specifies some types of output |
| [**<p>**](https://www.quackit.com/html_5/tags/html_p_tag.cfm) | Specifies a paragraph |
| [**<pre>**](https://www.quackit.com/html_5/tags/html_pre_tag.cfm) | Specifies preformatted text |
| [**<script>**](https://www.quackit.com/html_5/tags/html_script_tag.cfm) | Specifies a script |
| [**<select>**](https://www.quackit.com/html_5/tags/html_select_tag.cfm) | Specifies a selectable list |
| [**<source>**](https://www.quackit.com/html_5/tags/html_source_tag.cfm) | Specifies media resources |
| [**<span>**](https://www.quackit.com/html_5/tags/html_span_tag.cfm) | Specifies a section in a document |
| [**<strong>**](https://www.quackit.com/html_5/tags/html_strong_tag.cfm) | Specifies strong text |
| [**<style>**](https://www.quackit.com/html_5/tags/html_style_tag.cfm) | Specifies a style definition |
| [**<table>**](https://www.quackit.com/html_5/tags/html_table_tag.cfm) | Specifies a table |
| [**<td>**](https://www.quackit.com/html_5/tags/html_td_tag.cfm) | Specifies a table cell |
| [**<textarea>**](https://www.quackit.com/html_5/tags/html_textarea_tag.cfm) | Specifies a text area |
| [**<tfoot>**](https://www.quackit.com/html_5/tags/html_tfoot_tag.cfm) | Specifies a table footer |
| [**<th>**](https://www.quackit.com/html_5/tags/html_th_tag.cfm) | Specifies a table header |
| [**<thead>**](https://www.quackit.com/html_5/tags/html_thead_tag.cfm) | Specifies a table header |
| [**<title>**](https://www.quackit.com/html_5/tags/html_title_tag.cfm) | Specifies the document title |
| [**<tr>**](https://www.quackit.com/html_5/tags/html_tr_tag.cfm) | Specifies a table row |
| [**<u>**](https://www.quackit.com/html_5/tags/html_u_tag.cfm) | Specifies text with a non-textual annotation. |
| [**<ul>**](https://www.quackit.com/html_5/tags/html_ul_tag.cfm) | Specifies an unordered list |
| [**<video>**](https://www.quackit.com/html_5/tags/html_video_tag.cfm) | Specifies a video |

**3.4 Semantic and non-semantic elements-**

There are two types of tag semantic and non-semantic.

Semantic- These are the tags whose purpose can be understood by looking on the name of the tag itself. For eg:-audio, video,form etc.

Non-Semantic- These are the tags whose purpose cannot be understood by looking on the name of the tag. For eg:-em, u, span etc.

**3.5 Structure-**

Lets see a simple structure of a html document-

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>Title of webpage</title>  </head>  <body>  This is my first homepage.  </body>  </html> |

Save the file as mypage.html

**Description-**

The first tag in your html document is <html>. This tag tells your browser that this is the start of an html document. The last tag in your document is </html>. This tag tells your browser that this is the end of the html document.

The text between the tag <head> and the tag </head> is header information. Header information is not displayed in the browser window.

Metadata is data about data.

The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page.

Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

<meta charset="utf-8"> defines the character set to be used in the html document.

The text between the <title> tags is the title of your document. The <title> tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.

The text between <body> and </body> is the content which is displayed on browser.

Nested Tags-

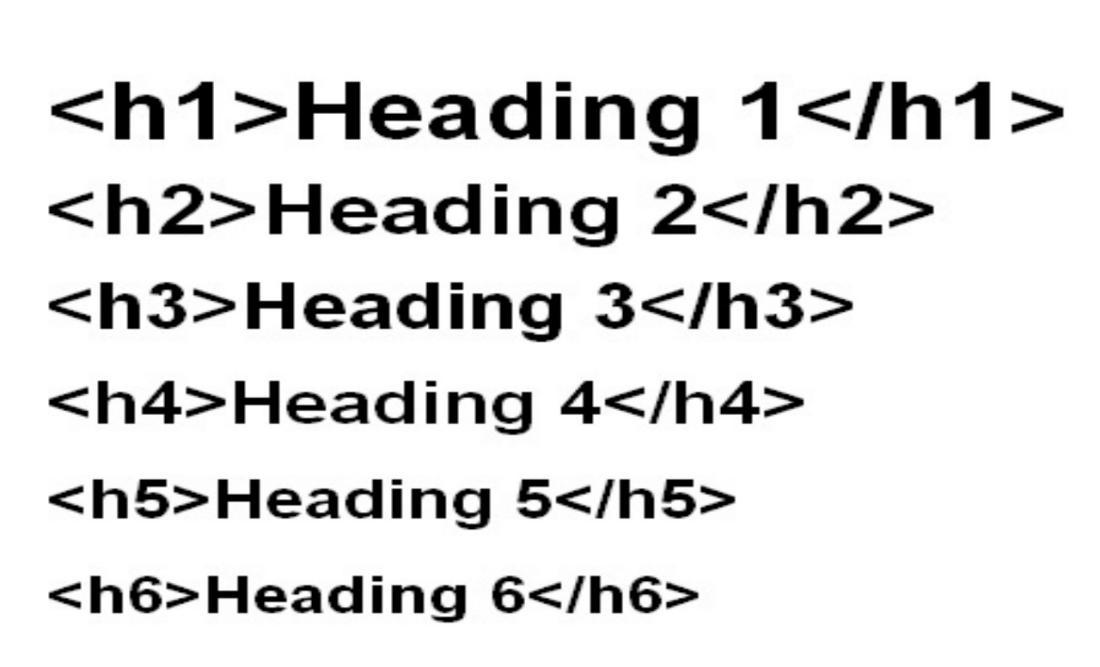
A tag also contains other tags. When you enclose an element in with multiple tags, the last tag opened should be the first tag closed. For example-

<p><strong>This is a bold paragraph</strong></p>

Lets discuss about the most commonly used elements of the html-

**3.6 Heading tag-**

Heading tags, as the name implies, are tags that are used for the creations of headings. The most important tag is the <h1> heading tag and will usually be the title of a post. Heading tags have a top-down hierarchy from <h1> to <h6>.



**3.7 Link**

The anchor element is used to create hyperlinks between a source anchor and a destination anchor. The source is the text, image, or button that links to another resource and the destination is the resource that the source anchor links to.

The anchor element tag is the letter “a” surrounded by angle brackets like this:<a>. Both the opening and closing attributes are required, and all of the content between the tags makes up the anchor source.

|  |
| --- |
| <a href="https://www.google.com" target="\_blank">Visit Google</a> |

The href attribute specifies the destination address of the link.

The link text is the visible part.

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

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

HTML bookmarks are used to allow readers to jump to specific parts of a Web page.

Bookmarks can be useful if your webpage is very long.

To make a bookmark, you must first create the bookmark, and then add a link to it.

When the link is clicked, the page will scroll to the location with the bookmark.

## Example

First, create a bookmark with the id attribute:

|  |
| --- |
| <h2 id="C1">Chapter 1</h2>  Then, add a link to the bookmark, from within the same page:  <a href="#C1">Jump to Chapter 1</a> |

Or, add a link to the bookmark, from another page:

### Example-

|  |
| --- |
| <a href="html\_demo.html#C1">Jump to Chapter 1</a> |

**3.8 Inserting Images**

In HTML, images are defined with the <img> tag.

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

The src attribute specifies the URL (web address) of the image:

<img src="url">

The alt Attribute

The alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

The value of the alt attribute should describe the image:

### Example

<img src="img.jpg" alt="Images">

**3.9 Adding Tables**

An HTML table is defined with the <table> tag.

Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. A table data is defined with the <td> tag.

### Example-

|  |
| --- |
| <table>  <tr>  <th>Name</th>  <th>Place</th>  </tr>  <tr>  <td>X</td>  <td>amb</td>   </tr>  <tr>  <td>Y</td>  <td>mumbai</td>   </tr> </table> |

**3.10 Lists-**

HTML offers three ways for specifying lists of information. All lists must contain one or more list elements.

**3.10.1 Unordered HTML List**

An unordered list starts with the **<ul>** tag. Each list item starts with the **<li>** tag.

The list items will be marked with bullets (small black circles) by default:

### Example

<ul>  
 <li>Item</li>  
 <li>Item</li>  
 <li>Item</li>

### An Unordered List:

* Item
* Item
* Item

**3.10.2 Ordered HTML List**

An ordered list starts with the **<ol>** tag. Each list item starts with the **<li>** tag.

The list items will be marked with numbers by default:

### Example

<ol>  
 <li>First</li>  
 <li>Second</li>  
 <li>THird</li>  
</ol>

### An Ordered List:

1. First
2. Second
3. Third

## **3.10.3 HTML Description Lists-**

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

### Example

|  |
| --- |
| <dl>  <dt>Coffee</dt>  <dd>- black hot drink</dd>  <dt>Milk</dt>  <dd>- white cold drink</dd> </dl> |

**3.11 Form**

The HTML **<**form> element defines a form that is used to collect user input:

|  |
| --- |
| <form> //form elements </form> |

An HTML form contains form elements.

Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

**3.11.1 The <input> Element**

The **<**input**>** element is the most important form element.

## **3.11.1.1 Text Input**

**<**input type="text"**>** defines a one-line input field for text input:

### Example-

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

## **3.11.1.2 Radio Button Input**

**<**input type="radio"**>** defines a radio button.

### Example-

|  |
| --- |
| <form>  <input type="radio" name="gender" value="male" checked> Male<br>  <input type="radio" name="gender" value="female"> Female<br>  <input type="radio" name="gender" value="other"> Other </form> |

## **3.11.2 The Submit Button**

**<**input type="submit"**>** defines a button forsubmittingthe form data.

**3.11.3 The Action Attribute**

The action attribute defines the action to be performed when the form is submitted.

<form action="destination.html">  
If the action attribute is omitted, the action is set to the current page.

**3.11.4 The Target Attribute**

The target attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.

The default value is "\_self" which means the form will be submitted in the current window.

To make the form result open in a new browser tab, use the value "\_blank":

**3.11.5 The Method Attribute**

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

**3.12 Questions**

Now ask the following questions to yourself-

1)What is Markup language.

2)How to differentiate between HTML5 and its previous versions?

3)What is head and its child tags?

4)What is body and its child tags?

5)What is semantic and non-semantic elements?

6)Difference between div and p tag?

7)Use of pre tag?

**CSS**

**1.Prerequisite-**

1)You should know the basic of html.

2)Follow the standard to write a web page which include the basic folder structure and the naming conventions.

**1.1 Naming conventions-**

For folders:

Inside a main parent folder, make three folders:

1)For Html files

2)For images

3)For CSS files

4)Make a README file for giving the description in the parent folder only.

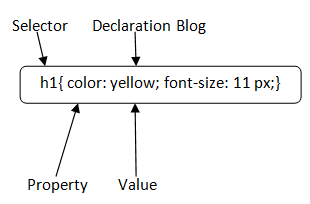
5)Make a index.html which will be your homepage of the website and place it in the parent folder only.

Place all the html files inside html folder, images inside images and css files inside css folder

**2.Introduction-**

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML.

This is a basic Css syntax-



**2.1 Selector:** Selector indicates the HTML element you want to style. It could be any tag like <h1>, <title> etc.

**2.2 Declaration Block:** The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

color: yellow;

font-size: 11 px;

Each declaration contains a property name and value, separated by a colon.

**2.3 Property:** A Property is a type of attribute of HTML element. It could be color, border etc.

**2.4 Value:** Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

**3. Types of CSS-**

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

1.Inline CSS

2.Internal CSS

3.External CSS

## **3.1) Inline CSS**

It is is used to apply CSS on a single line or element.

For example:

|  |
| --- |
| <p style="color:blue"**>**Hello CSS</p> |

**3.2) Internal CSS**

Internal CSS is used to apply CSS on a single document or page.

For example:

|  |
| --- |
| <style>  p{  color: red  }  </style> |

## **3.3) External CSS**

External CSS is used to apply CSS on multiple pages or all pages. We write all the CSS code in a css file. Its extension must be .css

For example:

|  |
| --- |
| p{color:blue}  You need to link this style.css file to your html pages:  **<**link rel="stylesheet" type="text/css" href="style.css"> |

### **3.4.ID**

### Id's are unique.

* Each element can have only one ID.
* Each page can have only one element with that ID.

**Syntax-**

|  |
| --- |
| <html>  <head>  <style> #place { background-color: skyblue;} </style>  </head>  <body>  <h2 class="place">A</h2>  </body>  </html> |

3.**5.Classes**

Classes are not unique.

* You can use the same class on multiple elements.
* You can use multiple classes on the same element.

**Syntax-**

|  |
| --- |
| <html>  <head>  <style> .place {  background-color: ;  color: white;  padding: 10px; }  </style>  </head>  <body> <h2 class="place">A</h2> <h2 class="place">B</h2> <h2 class="place">C</h2> </body>  </html> |

**3.6 Padding-**

CSS Padding property is used to define the space between the element content and the element border.

|  |
| --- |
| <!DOCTYPEhtml>  <html>  <head>  <style>  p {  background-color: pink;  }  p.applypadding {  padding-top: 50px;  padding-right: 100px;  padding-bottom: 150px;  padding-left: 200px;  }  </style>  </head>  <body>  <p>This is a paragraph with no specified padding</p>  <p class="applypadding">This is a paragraph with specified padding</p>  </body>  </html> |

**3.7 Margin-**

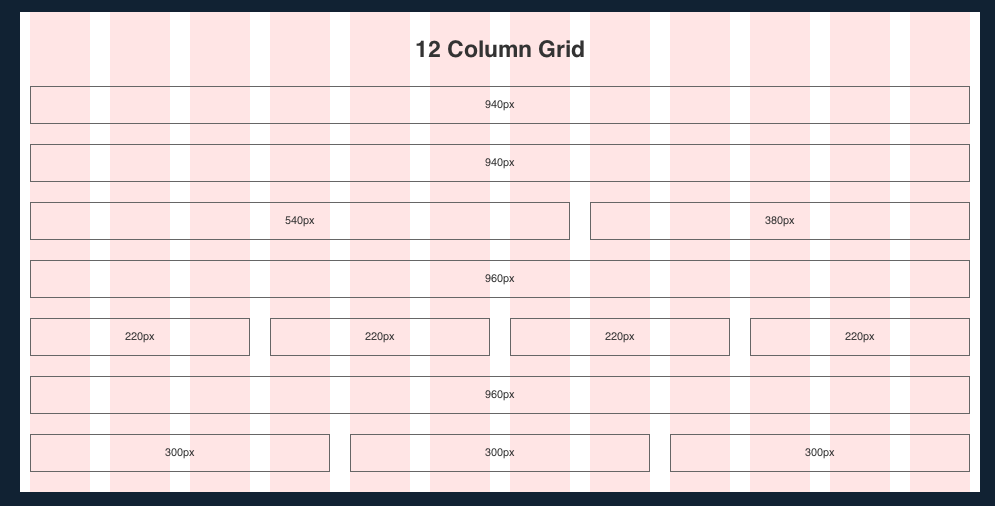
CSS Margin property is used to define the space around elements.

|  |
| --- |
| <!DOCTYPE html**>**  <html>  <head>  <style>  p {  background-color: red;  }  p.margin {  margin-top: 50px;  margin-bottom: 50px;  margin-right: 100px;  margin-left: 100px;  }  </style>  </head>  <body>  <p class="margin">This paragraph is displayed with specified margin.</p>  </body>  </html> |

**1.Grid 960:-**

Grid 960 is a framework which help you to define the structure of your web page in the form of grids.

**1.1) 12 Column Grid-**

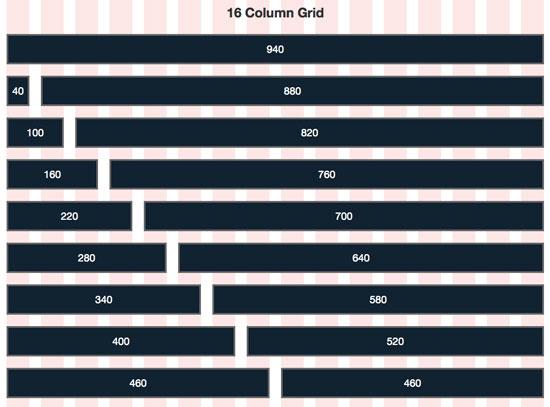


By grid 960 you can easily define the structure within any confusion.

In the 12-column version, the narrowest column is 60 pixels wide. Each column after that increases by 80 pixels.

So the available column widths are: 60, 140, 220, 300, 380, 460, 540, 620, 700, 780, 860 and 940.Similarly, in the 16-column version, the narrowest column is 40 pixels wide and each column after that increases by 60 pixels.

**1.2) 16 Column Grid-**



The whole page is divided into a 12 or 16 column grid. We take a main container which will contain all the elements.

Once defining the container you need to define the grids. You have to do it it such a way in grid 12 that the addition of all the elements in one grid is 12.

We have downloaded the files of grid 960 and placed them into the css folder. Now we need to link them into the html .

Example-

|  |
| --- |
| <link rel="stylesheet" href="css/960/960.css"/>  <link rel="stylesheet" href="css/960/reset.css"/>  <link rel="stylesheet" href="css/960/text.css"/>  Eg.  <!DOCTYPE html>  <html>  <head>  <title>My Profile</title>  <link rel="stylesheet" href="css/960/960.css"/>  <link rel="stylesheet" href="css/960/reset.css"/>  <link rel="stylesheet" href="css/960/text.css"/>  </head>  <body>  <div class="container\_12"> //This will take all the 12 columns  <div class="grid\_12">  <h1>Name</h1>  </div>  <div class="grid\_6">  <h3>Description</h3>  </div> //In this the two div will share a grid equally  <div class="grid\_6">  <h3>Others</h3>  </div>  </body>  </html> |

Thus by using 960 grid we can easily imagine the placement of various element on the screen without actually placing them.

Thus this is the whole overview of Web development using html, css and grid 960.

**Conclusion-**

This is all about the web development with the help of Html, Css and Grid960.

For more information you can refer to-

https://developer.mozilla.org/en-US/docs/Web