**LEARN HTML5,CSS3 &**

**960 GRID SYSTEM**

(IN AN HOUR)

# INTRODUCTION

In this article we are going to learn how to make a webpage using HTML, CSS & 960 Grid System. From here on I am going to need your perfect attention as we are going to learn it all in an hour.

Before starting, first of all be sure you have a **google chrome** browser latest version installed in your system as here we are going to deal with HTML5 & CSS3 which are the latest version and not every browser supports them.

# WORKING

Also, we are going to use notepad editor and whatever you code save the file as **.html** file and run it with google chrome. So here we have the basic understanding of the software that we needed and how to run a html file.

# PREREQUISITES

I have a good news for you guys cause there is not any prerequisite before working with html all you have to do is to be alert and be creative.

# Table Of Contents

[**INTRODUCTION**](#_a6sz7xwvbe0r) **2**

[**WORKING**](#_8053vuiz9lr) **2**

[**PREREQUISITES**](#_9k2hwasjb1qr) **2**

[**Table Of Contents**](#_yhoo89wtmdch) **3**

[**HTML**](#_r2rchkkfgur5) **6**

[HTML Elements/Tags](#_abwnovc5q5n7) 6

[Basic HTML Document](#_q0gffnqpxigm) 7

[<head> Tag](#_j53tjaei5xr2) 7

[Basic Tags in HTML](#_aodo1j4n3rvf) 7

[Basic Formatting Tags](#_wexjettuwm9) 8

[<img> tag](#_oo7gx4n9qv5q) 9

[Anchor tag](#_diq5yvk7caw5) 9

[LIST](#_z42igqh9nu8j) 10

[<Table> tag](#_oe5cb7v2v7pz) 11

[<fieldset> and <legend>](#_ggyf4odzk9he) 11

[Block and inline elements](#_jmglegbbg1hs) 12

[Form Elements](#_wjj3z65f43sl) 12

[Form tag](#_17lut6jbqcey) 12

[TEXTFIELD](#_g160ulxufz0c) 13

[Password](#_ir50j4tgdpvv) 13

[Radio Button](#_pma3497syl41) 13

[Checkbox](#_3sehnt5yfssl) 14

[Dropdown](#_m02g5mtlmm6x) 14

[Textarea](#_4witnehjjame) 14

[Submit/Reset button](#_d8kc8v6i7p69) 15

[Button](#_olz9yr9wqjph) 15

[Structural Elements](#_rvli9nmjann8) 15

[<header> tag](#_q1grrl8xabod) 15

[<nav> tag](#_d1vkmfbkb7rd) 16

[<article> tag](#_45svo7xc6n7m) 16

[<section> tag](#_ra36mj4lao18) 16

[<aside> tag](#_4vlc5lnjm77t) 16

[<footer> tag](#_uvz63x94y8xk) 16

[Media Elements](#_ubzh2p5s7uph) 17

[<video> element](#_kju7o9vfpxzw) 17

[<audio> element](#_2dcapgogkxqz) 18

[Embed Videos](#_xw0yjxmgijm) 18

[**Cascading Style Sheets(CSS)**](#_szfe7ublvmvm) **19**

[Introduction to CSS](#_4r4ezhfua8f7) 19

[Advantages of CSS](#_z5s8s12sy4km) 19

[Style and structure](#_49bt2qom2bbp) 19

[Types of CSS](#_n0g612324t45) 20

[Internal Stylesheet](#_n49zl525dj8j) 20

[External Stylesheet](#_o35i1t5z1z5o) 21

[Inline Stylesheet](#_63ov74w03dmp) 21

[CSS Property](#_i6ph7k6urkgi) 22

[CSS – Background](#_5suijos4mtpk) 22

[CSS – Fonts](#_9a4dvz8e6d69) 23

[CSS – Text](#_9evd3cy2kpyv) 23

[Using class](#_i9b5tokzuxv4) 25

[Using id](#_k036vd30kyn0) 25

[CSS – Links](#_8ase72yrkax1) 25

[CSS – Floating](#_e8f722vt3moy) 26

[CSS – List](#_p76s6kkp0x3h) 26

[CSS – Display](#_5gzr6hgwtbot) 26

[CSS – Positioning](#_fourjrja1qig) 27

[Positioning – static](#_oryapf9qgbcp) 27

[Positioning – relative](#_ejvjfpbxaz9e) 27

[Positioning – absolute](#_tl9n2m9l6v1i) 28

[Positioning – fixed](#_fl4q61pe0904) 28

[CSS Box Model](#_g3sgprspy22r) 29

[**960 GRID SYSTEM**](#_b2e6x4zuvws) **30**

[Dimensions](#_rjw0zna0ddtl) 30

[Purpose](#_q0ngsl9xp0lp) 30

[Example](#_z8xwwvi0s22e) 30

# 

# 

# HTML

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages. Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext. As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display. Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

## HTML Elements/Tags

* Elements give the structure of html documents
* All elements have the same format begins with “<” tag end with “>”.
* Elements represents how to be present data in the web browser.
* The tag can be
* Empty **<br/>**
* With body content **<b>Hello World</b>**
* With attributes **<input type=”text”>**
* With body contents and attributes

|  |
| --- |
| <form name=”form1”>  <input type=”text” name=”username”>  </form> |

## 

## 

## 

## Basic HTML Document

In its simplest form, following is an example of an HTML document:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>This is document title</title>  </head>  <body>  <h1>This is a heading</h1>  <p>Document content goes here.....</p>  </body>    </html> |

Here, In the above example <!DOCTYPE html> specifies the browser that this is a HTML5 document. In earlier versions different DOCTYPE is used which is of no concern for us. After this we have <html> tag which is used to define the starting of html document. In the head section we define the header part of html that is the information given to the browser. In the body section we define the child tags and main content that is going to be represented in the browser. After that we have all the closing tags.

## <head> Tag

It has following child tags:

* **Title:** specifies the title for webpage
* **Meta:** -used to specify content type, character set and author details

-used by search engines.

* **Link:** used to call an external CSS page
* **Style:** specifies that css written inside this tag
* **Script:** specifies that javascript is written inside the tag

## Basic Tags in HTML

* Basic Formatting Tags
* Image <img> tag
* Anchor <a> Tag
* Table
* Form Tags

## Basic Formatting Tags

* <b> - Bold text

|  |
| --- |
| <b>This text is bold</b> |

* <strong> - Important text

|  |
| --- |
| <strong>This text is strong</strong> |

* <i> - Italic text

|  |
| --- |
| <i>This text is italic</i> |

* <em> - Emphasized text

|  |
| --- |
| <em>This text is emphasized</em> |

* <mark> - Marked text

|  |
| --- |
| <h2>HTML <mark>Marked</mark> Formatting</h2> |

* <small> - Small text

|  |
| --- |
| <h2>HTML <small>Small</small> Formatting</h2> |

* <del> - Deleted text

|  |
| --- |
| <p>My favorite color is <del>blue</del> red.</p> |

* <ins> - Inserted text

|  |
| --- |
| <p>My favorite <ins>color</ins> is red.</p> |

* <sub> - Subscript text

|  |
| --- |
| <p>This is <sub>subscripted</sub> text.</p> |

* <sup> - Superscript text

|  |
| --- |
| <p>This is <sup>superscripted</sup> text.</p> |

## <img> tag

Images are defined with the <img> tag

* Has attributes as

– src(source)

– alt (alternative text )

– width

– height

|  |
| --- |
| <img src="smiley.gif" alt="Smiley face" height="42" width="42"> |

## Anchor tag

* To open a link or a move to a different webpage use **<a>** tag
* The website address is specified using **href** attribute**.**
* Use **target=“\_blank”** to open the url in a new window

|  |
| --- |
| <a href="https://www.google.com">Visit googe!</a> |

* To propagate in the same page also **<a>** tag is used **eg:**
* On the top of the page write ***< a name = “top”>***
* On the bottom of the page use ***< a href=“#top>Goto Top</a>*** When you click **Goto Top** it will take to the top of the page

## LIST

* List Unordered
* uses <ul> tag
* adds the list value using <li> tag
* Has attribute –

type(square/disc/circle)

|  |
| --- |
| <ul>   <li>Coffee</li>   <li>Tea</li>   <li>Milk</li> </ul> |

* List Ordered
  + - uses ***<ol>*** tag
    - adds the list value using ***<li>*** tag
    - Has attribute –

– start(specifies start number)

– type(number or alphabet)

|  |
| --- |
| <ol>   <li>Coffee</li>   <li>Tea</li>   <li>Milk</li> </ol> |

* List – Description List
* Description List:<dl>
* Description term:<dt> - name or term
* Description data:<dd> - can be paragraph or list

|  |
| --- |
| <dl>   <dt>Coffee</dt>   <dd>Black hot drink</dd>   <dt>Milk</dt>   <dd>White cold drink</dd> </dl> |

## <Table> tag

* **<table>** is used to display the tabular data as rows and columns.
* Three different elements are used to insert data in the table
* **<tr>** - represents **table row**
* **<td>** - represents **table data** each cell value of the table.
* **<th> -** represents table head
* **<table>** has attributes like border ,cellpadding , cellspacing
* **<td>** has attributes like

– **colspan** to merge one or more columns

– **rowspan** to merge one or more rows

|  |
| --- |
| <table>   <tr>     <th>Month</th>     <th>Savings</th>   </tr>   <tr>     <td>January</td>     <td>$100</td>   </tr> </table> |

## <fieldset> and <legend>

* Used to differentiate the tables with names

|  |
| --- |
| <form>   <fieldset>     <legend>Personalia:</legend>     Name: <input type="text"><br>     Email: <input type="text"><br>     Date of birth: <input type="text">   </fieldset> </form> |

## Block and inline elements

* A block element is an element that takes up the full width available.

**Example for block elements**

– <h1>, <p>, <li>,<div>

**Example of inline elements**

– <span>, <a>

## Form Elements

* HTML Forms are used to collect user input.
* **<form>** element defines an HTML
* formForm elements are :

– Textfield

– Password

– Radio

– Checkbox

– Select(dropdown)

– Textarea

– Submit

– Reset

– Button

## Form tag

* **<form>** is used for creating form tag
* All form elements MUST be inside form tag
* Has attributes like

– **name** used while validating a form

– **method** specifies whether **get/post** is used to carry form values

– **action** specifies the address that gets called on clicking submit

|  |
| --- |
| <form action="/action\_page.php" method="get">   First name: <input type="text" name="fname"><br>   Last name: <input type="text" name="lname"><br>   <input type="submit" value="Submit"> </form> |

## TEXTFIELD

* <input> is used for one line input field for text input
* Has attributes like

– **type** specifies the field type

– **name** is used to retrieve value of form field in the backend(servlet)

– **id** is used to retrieve value of form field in javascript

– **value** specifies the default value

– **placeholder** is seen in the background as text

|  |
| --- |
| <form action="/action\_page.php" method="get">   First name: <input type="text" name="fname"><br>   Last name: <input type="text" name="lname"><br>   <input type="submit" value="Submit"> </form> |

### Password

* <input tag is used for one line input field for text input
* Has attributes like

– **type** specifies the field type

– **name** is used to retrieve value of form field in the backend (servlet)

– **id** is used to retrieve value of form field in javascript

– **value** specifies the default value

– **placeholder** is seen in the background as text

|  |
| --- |
| <input type="password" name="password"> |

### Radio Button

* Allows user to select ONE in a limited number of choice
* Has attributes like

– **type** specifies the field type

– **name** is used to retrieve value in the backend(has to be same name)

– **id** is used to retrieve value of form field in javascript

– **value** specifies the value of the radio button

|  |
| --- |
| <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> |

### Checkbox

* Allows the user to select ZERO or MORE options of a limited number of choices.
* Has attributes like

– **type** specifies the field type

– **name** is used to retrieve value in the backend(has to be same name)

– **id** is used to retrieve value of form field in javascript

– **value** specifies the value of the checkbox

|  |
| --- |
| <form action="">  <input type="checkbox" name="vehicle" value="Bike">I have a bike<br>  <input type="checkbox" name="vehicle" value="Car">I have a car  </form> |

### Dropdown

* **<select>** tag defines a **drop-down** list
* Has a child tag <option> which specifies the list
* **<select>** has attributes like

– **name** is used to retrieve value in the backend

– **multiple** allows to select multiple values

– **size** specifies the number of elements that will be shown

|  |
| --- |
| <select>   <option value="volvo">Volvo</option>   <option value="saab">Saab</option>   <option value="mercedes">Mercedes</option>   <option value="audi">Audi</option> </select> |

### Textarea

* **<textarea>** is used for multi-line input
* Has attributes like **rows** and **cols**

|  |
| --- |
| <textarea rows="4" cols="50">  At w3schools.com you will learn how to make a website. We offer free tutorials in all web development technologies.  </textarea> |

### Submit/Reset button

* **<input type="submit">** defines a button for **submitting** a form to aform-handler.
* Form-handler is a server page with a script for processing input data.
* The form-handler is specified in the form's **action** attribute
* **<input type=“reset">** will clear all the values entered in the form

|  |
| --- |
| <form action="/action\_page.php" method="get">  First name: <input type="text" name="fname"><br>  Last name: <input type="text" name="lname"><br>  <button type="submit" value="Submit">Submit</button>  <button type="reset" value="Reset">Reset</button>  </form> |

### Button

* **<button>** is used for creating button
* Can be used when you want to handle events using javascript
* Will not submit the values to the server page

|  |
| --- |
| <button type="button" onclick="alert('Hello world!')">Click Me!</button> |

## Structural Elements

* HTML5 has elements for better document structure
* header
* nav
* article
* section
* aside
* footer

### <header> tag

* Represents a group of introductory or navigational aids.

**e.g.,**

* table of contents, a search form, any relevant logos

### <nav> tag

* Represents a section of a page that links to other pages or to parts within the page

**e.g.,**

* a section with navigation links

### <article> tag

* Represents a self-contained composition in a document, page, application, or site that is reusable.

**e.g.,**

* a forum post, a magazine or newspaper article, a blog entry, user-submitted comment, any other independent item of content

### <section> tag

* Represents a generic document or application section.

**e.g.,**

* Chapters, various tabbed pages in a tabbed dialog box
* A Web site's home page could be split into sections for an introduction,
* news items, & contact information

### <aside> tag

* Represents a section of a page that consists of content that is tangentially related to the content around the aside element,
* Are separate from the main content and represented as sidebars in printed typography

**e.g.,**

* author profile in a blog

### <footer> tag

* Represents a footer for its nearest ancestor sectioning content.
* Is not a sectioning content

**e.g.,**

* contact details of author, copyright information

## Media Elements

* Multimedia can be in different formats.

**eg.** Pictures, music, sound, video, films, animations, etc...

* Web page contains multimedia elements of different formats

**Media Elements**

* **<audio>** - to add audio content
* **<video>** - to add video content
* **<source>** - to define source for audio and video
* **<track>** - to define tracks for audio and video
* **<iframe>** - to embed you tube videos in a container

### <video> element

* <video> element specifies a standard way to embed a video/movie on a web page
* **controls** attribute adds video controls like **play**, **pause**, **volume**
* Supported video formats : mp4, WebM, and ogg
* <source> element allows you to specify alternative video files which the browser may choose fro.
* Has DOM methods, properties and events.
  + **Methods** –play(), pause(), load()
  + **Properties** –videoWidth, videoHeight
  + **Events** –play, pause, empty

|  |
| --- |
| <video width="400" controls>  <source src="mov\_bbb.mp4" type="video/mp4">  <source src="mov\_bbb.ogg" type="video/ogg">  Your browser does not support HTML5 video.  </video> |

### <audio> element

* <audio> element specifies a standard way to embed a audio on a web page
* **controls** attribute adds audio controls like **play**, **pause**, **volume**
* Supported audio formats : mp3, wav, and ogg
* <source> element allows you to specify alternative audio files which the browser may choose from

Has DOM methods, properties and events.

* **Methods** –play(), pause(), load()
* **Properties** –videoWidth, videoHeight
* **Events** –play, pause, empty

|  |
| --- |
| <audio controls>  <source src="horse.ogg" type="audio/ogg">  <source src="horse.mp3" type="audio/mpeg">  Your browser does not support the audio element.  </audio> |

### Embed Videos

* Embed videos from youtube using iframe

|  |
| --- |
| <iframe width="854" height="480" src="https://www.youtube.com/embed/4qEh8bwDW0I" frameborder="0" allow="autoplay; encrypted-media" allowfullscreen></iframe> |

# Cascading Style Sheets(CSS)

### Introduction to CSS

* CSS means “Cascading Style Sheet”
* Handles the look and feel part of a web page.
* Change the text color, font style, the paragraph spacing
* Can be combined with the markup languages HTML or XHTML
* Easy to learn and understand
* Provides powerful control over the presentation of a HTML document

### Advantages of CSS

* **Content and Style Separation**

– Separates HTML content from the style and layout of that document

* **Saves time**

– write CSS once and reuse the same code to group of HTML elements or across multiple HTML pages.

* **Easy maintenance**

– By making changes in the css file, the elements in all the web pages get updated automatically. Also helps to maintain consistency across multiple documents

* **Superior style to html**

– has more presentation capabilities than HTML. **e.g.,** Adding opacity, gradients, rounded corners, animation

* **Multi device Compatibility**

– Allow the HTML document to be optimized and rendered in more than one type of device or media such as desktops, laptops, mobiles etc

### Style and structure

* Has style rules that are interpreted by the browser.
* **Selector** : is a HTML element for which style will be applied. **e.g., <h1>**,**<p>**
* **Property:** Is a type of attribute of HTML tag like color, font etc
* **Value :** the value assigned to a property

## Types of CSS

* Internal CSS
* External CSS
* Inline CSS

### Internal Stylesheet

* is used in the same html page
* Is defined in head section and inside the <style> tags

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <style>  body {  background-color: linen;  }  h1 {  color: maroon;  margin-left: 40px;  }  </style>  </head>  <body>  <h1>This is a heading</h1>  <p>This is a paragraph.</p>  </body>  </html> |

### External Stylesheet

* Is written in a separate file and saved in ***anyname.css***
* is ideal when same style has to be applied to many pages.
* Can change the look of an entire web site by changing a single css file

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <link rel="stylesheet" type="text/css" href="mystyle.css">  </head>  <body>  <h1>This is a heading</h1>  <p>This is a paragraph.</p>  </body>  </html> |

### Inline Stylesheet

* Is used inside any tag using **style** attribute
* **style** attribute can contain any CSS property

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <h1 style="color:blue;margin-left:30px;">This is a heading</h1>  <p>This is a paragraph.</p>  </body>  </html> |

## CSS Property

* Background
* Text
* Fonts
* Links
* Lists
* Display
* Floating
* Position

### CSS – Background

* **background-color:** to set the background color of an element.
  + background-color : pink;
* **background-image:** to set the background image of an element.
  + background-image : url (“butterfly.gif");
* **background-position**: to control the position of an image in the background.
  + background-position : top left;
* **background-attachment** property is used to control the scrolling of an image
  + background-attachment : scroll;
* **background-repeat** is used to control the repetition of an image.
  + background-repeat : repeat;
* **background** is shorthand to specify a number of other background property
  + background: #FFCC66 url("butterfly.gif") repeat scroll ;

|  |
| --- |
| body {  background-image: url("img\_tree.png");  background-repeat: no-repeat;  background-position: right top;  margin-right: 200px;  background-attachment: fixed;  } |

### CSS – Fonts

* **font-family** is used to change the face of a font.
  + font-family: sans-serif;
* **font-style** used to make a font italic or oblique.
  + font-style: italic;
* **font-variant** used to create a small-caps effect.
  + font-variant : small – caps ;
* **font-weight** used to increase or decrease the boldness for a font .
  + font-weight: bold;
* **font-size** used to increase or decrease the size of a font.
  + font-size: 20px;
* **font** is a shorthand to specify a number of other font properties.
  + font: 15px bold sans-serif;

### CSS – Text

* **color** is used to set the color of a text.
  + color: red;
* **Direction** is used to set the text direction.
  + direction : rlt;
* **letter-spacing** is used to add or subtract space between the letters in a word.
  + letter-spacing: 5px;
* **word-spacing** is used to add/subtract space between the words of a sentence.
  + word-spacing : 5px;
* **text-indent** is used to indent the text of a paragraph.
  + text-indent: 1cm;
* **text-align** is used to align the text of a document.
  + text-align : right;
* **text-decoration** is used to underline, overline, or strikethrough text.
  + text-decoration : underline;
* **text-transform** is used to capitalize text or convert text to uppercase orlowercase letters.
  + text-transform: uppercase;
* **white-space** is used to control the flow and formatting of text.
  + white-space : pre;
* **text-shadow** is used to set the text shadow around a text.
  + text-shadow : 4px 4px blue;

|  |
| --- |
| <style>  h1 {  text-decoration: overline;  }  h2 {  text-decoration: line-through;  }  h3 {  text-decoration: underline; |

## Using class

* To give same formatting to different tags
* It is referred by ***.***

|  |
| --- |
| <div class="widget"></div>  <div class="widget"></div>  <div class="widget"></div> |

## Using id

* To give unique formatting for an individual tag
* It is referred by **#**

|  |
| --- |
| <li id="comment-27299" class="item"> |

## CSS – Links

* **:link**

for unvisited hyperlinks

**a : link {color: orange}**

* **:visited**

for visited hyperlinks

**a : visited { color: black}**

* **:active**

Is when the user is currently clicking the link

**a : active {color: green}**

* **:hover**

happens when the user's mouse pointer hover over the element.

**a : hover { color: yellow}**

|  |
| --- |
| <html>  <head>  <style type="text/css">  a:link {color:#000000}  </style>  </head>  <body>  <a href="">Link</a>  </body>  </html> |

## CSS – Floating

* Elements are floated horizontally, either left or right only.
* Floating an image to the right/left of text.
* Do not float up or down in CSS.
* Elements before the floating element will not be affected.

– float: right; float: left;

* Turning off float – using clear

– clear: both;

## CSS – List

* **list-style-type** control the shape or appearance of the marker.

– **list- style- type: circle;**

* **list-style-image** an image for the marker rather than a bullet point or number.

– **list- style-image: url (butterfly.gif);**

## CSS – Display

* Hiding an element can be done by setting the property

– **display :none, visibility :hidden**

* + **visibility: hidden**

-Hides an element but it takes up the same space as before the element will be hidden

* + **display :none**

-Hides an element, and it will not use up any space

## CSS – Positioning

* This specifies how an element is positioned in a document.
* The elements are then positioned using top, bottom, left and right properties
* The position values are

– static

– fixed

– relative

– absolute

### Positioning – static

* Static positioning is the default positioning model for elements.
* They are positioned according to the normal flow of the page
* The top, left, right and bottom properties have no value

### Positioning – relative

* Relative positioning allows to specify an offset (top, right, bottom, left) which is relative to the element's normal position in HTML flow

|  |
| --- |
| <html>  <head>  </head>  <body>  <div style="**position:relative;left:80px;top:2px;**background-color:yellow;">  This div has relative positioning.  </div>  </body>  </html> |

### Positioning – absolute

* An element that is absolutely positioned is taken out of the flow. The other elements are positioned as if it did not exist.
* The element is positioned relative to the nearest positioned ancestor.
* They are bound by the viewport and will cause scrolling
* If there are no ancestors, then positioned relative to the entire document body.

|  |
| --- |
| <html>  <head>  </head>  <body>  <div style="**position:absolute; left:80px; top:20px;** background-color:yellow;">  This div has absolute positioning.  </div>  </body>  </html> |

### Positioning – fixed

* This restricts an element to a specific position in the viewport, which always stays in the same place even if the page is scrolled

|  |
| --- |
| <html>  <head>  </head>  <body>  <div style="**position:fixed; left:80px; top:20px;** background-color:yellow;">  This div has fixed positioning.  </div>  </body>  </html> |

## CSS Box Model

* CSS box model is a box that wraps around every HTML element.
* It has margins, borders, padding, and the actual content.

**Content** - The content of the box, where text and images appear

**Padding** - Clears an area around the content. It is transparent

**Border** - A border that goes around the padding and content

**Margin** - Clears an area outside the border and is transparent

**Total element width** = width + left padding + right padding + leftborder + right border + left margin + right margin

**Total element height** = height + top padding + bottom padding +top border + bottom border + top margin + bottom margin

|  |
| --- |
| <html>  <head>    <style>  .div1 {  width: 300px;  height: 100px;  border: 1px solid blue;  box-sizing: border-box;  }  .div2 {  width: 300px;  height: 100px;  padding: 50px;  border: 1px solid red;  box-sizing: border-box;  }  </style>    </head>  <body>  <div class="div1">TutorialsPoint.com</div></br>  <div class="div2">TutorialsPoint.com</div>  </body>  </html> |

# 960 GRID SYSTEM

The 960 Grid System is an effort to streamline web development workflow by providing commonly used dimensions, based on a width of 960 pixels. There are two variants: 12 and 16 columns, which can be used separately or in tandem

## Dimensions

The 12-column grid is divided into portions that are 60 pixels wide. The 16-column grid consists of 40 pixel increments. Each column has 10 pixels of margin on the left and right, which create 20 pixel wide gutters between columns.

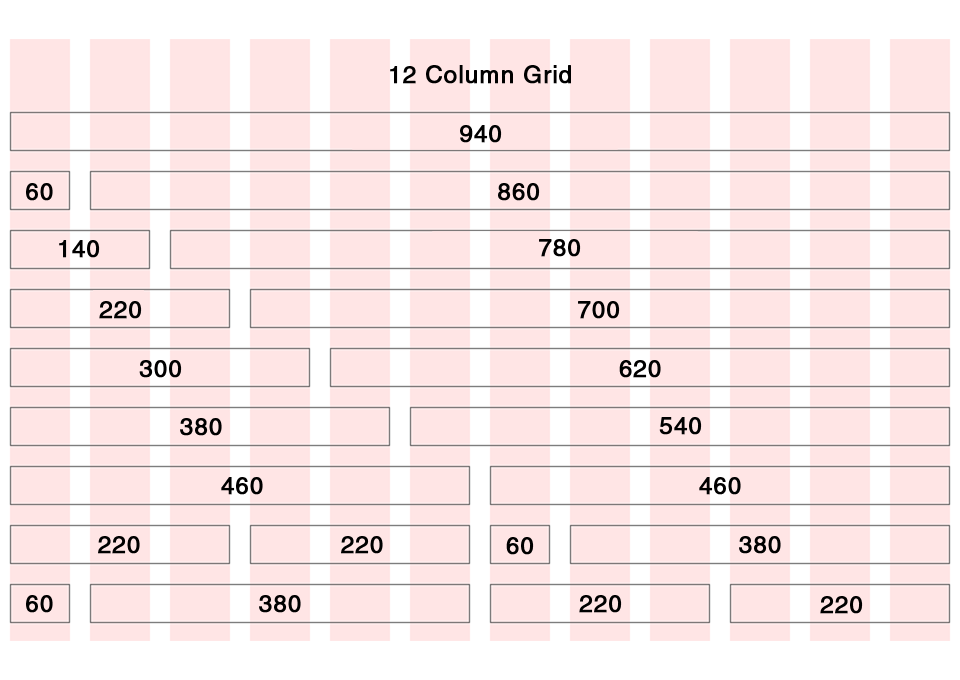
## Purpose

The premise of the system is ideally suited to rapid prototyping, but it would work equally well when integrated into a production environment. There are printable sketch sheets, design layouts, and a CSS file that have identical measurements

## Example

|  |
| --- |
| <h1><a href="http://960.gs/">960 Grid System</a></h1>  <div class="container\_12">  <h2>12 Column Grid</h2>  <div class="grid\_12">  <p>940px</p>  </div>  <!-- end .grid\_12 -->  <div class="clear">&nbsp;</div>  <div class="grid\_9">  <p>700px</p>  </div>  <div class="grid\_3">  <p>220px</p>  </div>  <div class="grid\_12">  <p>940px</p>  </div>  </div>  <!-- end .container\_12 -- |

As we can see in the above example it is very easy to work with grid system. It will make your css coding and placement of divisions very easy. It comes with predefined columns in which you have to work. You can make a drawing in a simple page and craft your webpage as you want. All you have to do is take care of the height of the columns and as you define it, it will place the divisions accordingly.



**GRID DIVISIONS**