* **HTML Introduction**
* **HTML Uses**
* **HTML Tag & Types.**
* **HTML Structure.**
* **HTML Softwares.**
* **HTML Attributes.**
* **HTML Headings.**
* **HTML Scrollings**
* **HTML Paragraphs**
* **HTML Images.**
* **HTML YouTube Videos and Google Maps.**
* **HTML Fonts**
* **HTML Breaks and Horizantal Line.**
* **HTML Source Code Presentation.**
* **HTML Physical Text Formated Tags.**
* **HTML Entities.**
* **HTML Comments.**
* **HTML Colors.**

## INTRODUCTION

* HTML stands for "HYPER TEXT MARKUP LANGUAGE".
* HTML was invented by "TIM BERNERS LEE" in 1990.
* The first internet Web-site developed by "Tim Berners Lee" using "HTML" in 1991 described about "WHAT IS WWWW?" in that web-site.
* HTML is case\_in sensitive language means write html\_program in upper or lowercase or both the combination also.
* HTML is also call as Markup\_Language, that means it has the pre-defined tags.

## USES OF HTML

* Create static web-pages/Websites by having Content.
* Create User-Registartion-Forms/Login-Forms/Search Boxes.
* Create Hyperlinks for Navigation.
* Create List Types.
* Create Tables\_format information.
* Integrate videos/audio/images..etc
* To apply diff text format styles on information.
* Apply scrolling and blinking actions.
* Implement Table Format static Websites also…etc

## HTML TAGS AND TYPES

**Q) What is TAG.**

The process of applying rules and instructions to Change/Convert normal text As Hyper text to place into webpage, by applying Pre-defined HTML Language keywords which is enclosed by angular brackets (i.e < , >) that we can call as TAG.

**Syntax: <keyword>. Ex: <html>, <body>, <table>, <form>,... Etc.**

**TYPES OF TAGS:** These are 2 Types :

**1) Container or parent Tag:** This Purpose of container tag is it contains or hold user-input information to display or present in the webpage.

**Syntax:** <tagname>--content/Information ---</tagname> Like as ( Opening & Closing Tags )

**Ex:** <body>----</body>, <table>----</table>, <html>-----</html>……Etc

**2) Non-Container or Empty Tag (**Self-Closed Tag.**) :**

**Syntax:** <tagname /> **Ex:-** <br />, <hr />, <img />, <input />…..Etc.

## HTML STRUCTURE

To Write and Implement Html Pages or webpages, we have the pre-defined Html structure which is start with <html>- tag. **The Following Structure is :**

**<html>**

**<head>------//header-section part------</head>**

**<body>------//body section part----</body>**

**</html>**

**Note:- All the HTML pages Should be save with file extention is ".html" or ".htm"**

## SOFTWARES FOR HTML

**1) EDITOR S/W:** Notepad, Notepad++, Edit-Plus, Dreamweaver, Netbeans, Ecclipse,..etc

**2) BROWSER S/W:** Mozilla, Chrome, Safari,Opera, Torch, Internet Explorer,..etc

## ATTRIBUTES OF HTML

* **It**  is a modifier of an **HTML** element.
* Attributes provide additional information about HTML elements.
* Attributes come in name/value pairs like: **name="value".**
* **Syntax: A**ttribute= value **Ex:** bgcolor = red;

**RULES OF ATRIBUTES :-**

* Any html\_tag attributes should be write inside of the **starting-tag/Opening** Tag only.
* Atleast maintain one cursor space b/**w tagnname** and **Attribute\_name.**
* Attribute also write **Case In-sensitive Format**.
* One html\_tag attributes doen't work or support into another html\_tags.
* When we are writing more than one attributes inside of the same html\_tag, between that attributes Should maintain atleast one cursor space.
* No need to follow any order format to write Multiple attributes in html-tags.

**BODY TAG ATTRIBUTES:**

* **bgcolor=value :**

Here value means color\_name ,it is used for to fill with body-part of the area with background color.

* **text=value :**

**H**ere value means color\_name,it is used for to fill the entire the body\_document text information with color.

* **background=value :**

**H**ere value means "imagename.extention", it is used for to set the image as background\_image.

# PRE DEFINED HEDING TYPES

**T**o display Text in heading formats into webpage by following the tags are:

**<h1>----</h1> and <h2>----</h2> and <h3>----</h3>**

**<h4>----</h4> and <h5>----</h5> and <h6>----</h6>**

**Attribute: align=left(default)/right/center** :-To present text information in diff positions as horizantal direction.

# SCROLLING TEXT AND IMAGES

To apply scrolling actions on Text information and images also using <marquee>---</marquee>

**Attributes:**

* **bgcolor=color\_name.**
* **direction=left(default)/right/up/down.**
* **scrollamount=value**  To increse and decrese speed of the scrolling.
* **behavior=scroll(default)/alternate/slide** To apply diff scrolling behaviours.
* **loop=value(value means -ve(default)/+ve )**  No.of time scrolling.
* **width=value** To restrict width size of the background body of marquee scrolling.
* **height=value** To restrict height size of the background body of marquee scrolling.

# PARAGRAPGHS FOR TEXT

To divide huge amount of information as multile\_paragraphs by using **<p>-tag** taking no.of times we can see paragraph\_format text. it has only one attribute\_support i.e **align= justify**.

# FONT STYLES FOR TEXT

To apply diff font styles on text\_information i.e text\_color and increase/decrease text\_font size and apply diff font\_style names which is developed by using the attributes are:

* color=colour\_name,
* size=value ( means min-1 and max-7)
* face="font\_names" Ex: arial, times new roman,..Etc

# IMAGES FOR LOGOS/BANNERS/PHOTOS

To integrate images like as Logos/ Photos/ Icons/ banners into webpages.

* src="image\_name.extention" :- To define image sourec path .
* align=left(default)/right
* width=value, height=value
* border=value
* hspace=value :- To get the horizontal space
* vspace=value:- To get the vertical space.

# YOUTUBE VIDEOS AND GOOGLE MAPS

To play YouTube Videos in Website, Then Use <iframe></iframe> Tag.

* src="source path" :- To define source path of video/Map
* align=left(default)/right
* width=value
* height=value

**Q) How To Integrate YouTube Videos into Webpage?**

* Open Youtube website i.e <http://www.youtube.com>
* Select any Video and Right Click On Video .
* Now Choose **GET embed code Option**.
* Now Copy **<iframe>-tag** Code and paste into Webpage.

**Q) How To Integrate Google Maps address into Webpage?**

* Open Google map website i.e <http://www.map.google.com>
* Now Search For address into Textbox.
* Now Click on MENU icon at **Top right side**
* Now Choose Option is Share or Embed Map
* Choose Embed Map Tab Option.
* Now Copy **<iframe>**  Code and paste into Webpage.

# LINE BREAKS

<BR /> tag adds New Break Line Means one New Line Break into webpage.

# CREATE HORIZANTAL LINE

<HR /> Tag adds Horizantal Line into webpage.

**Attributes :**

* **color=color\_name and size=number**
* **width=number and align=center/left/right**

# PHYSICAL TEXT FORMATTED TAGS

* Bold <b>.
* Italics <i>|<var>.
* Underline <u>.
* Strike Through <s>|<strike>|<del>.
* Bigger Print <big>.
* Smaller Print <small>.
* Subscript <sub>.
* Superscript <sup>.
* Abbrivations <acronym>|<abbr>.
* Address Information <address>.

# HTML SOURCE CODE PRESENTATION

The HTML <pre> tag is used for indicating preformatted text. The code tag surrounds the code being marked up. Browsers normally render pre text in a fixed-pitched font, with whitespace in tact, and without word wrap

# HTML ENTITES

* Reserved characters in HTML must be replaced with character entities.
* Characters, not present on your keyboard, can also be replaced by entities.
* If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags.
* Character entities are used to display reserved characters in HTML.
* A character entity looks like this:
* &*entity\_name*; ( OR ) &#*entity\_number*;

# COMMENTS IN HTML

* Comment tags <!-- and --> are used to insert comments in HTML.
* You can add comments to your HTML source by using the following syntax
* <!-- Write your comments here -->

# USEFUL COLORS IN HTML

Select Different Format Colors for Html To apply on Text and background colors i.e

* **Named Colors Ex: red, green, blue,..Etc**
* **Hexa Decimal Colors Ex: #fff, #ff00ff**
* **RGB Colors Ex: rgb(0,0,0)**
* **HTML Lists.**
* **HTML Hyperlinks.**
* **HTML Tables.**
* **HTML Forms.**
* **HTML Header Section.**
* **HTML5 Introduction.**

**HTML LISTS**

To Display list of items either in the form of Sequential or Unsequential Order Format. These lists are divided as 3 categories:

**I) Ordered List : (Use <ol></ol> Tag).**

**II) Unordered List :( Use <ul></ul> Tag).**

**III) Definition List :( Use <dl></dl> Tag).**

**I) Ordered List: To** display list items in sequential order format i.e in numeric or alphabets or roman numbers..etc for both in lower and uppercases. **Attributes:** type=1(default)/a/A/i/I.

**II) Unordered List: It** is used for to display lists in Unsequential order format i.e in symbols notation format.. **Attribute:**  **type=disc/square/circle**

* Display Every list name Enclosed By **<li></li>** For Both Order or Un\_order List.
* Ex:- **<li> ListName </li>.**

**III) Definition List: It** is used for to display list\_item with description of list by following tags are <dt></dt> and <dd></dd>-tags.

**Syntax: <dl>**

**<dt>--Defition Title Name-</dt>**

**<dd>--Description-</dd>**

**</dl>**

**HTML TABLES**

To prepare or make table format structure with data into WEBPAGE.So Create table format to starts with one parent tag is. <table></table>-tag.

**Attributes:**

* border=value.
* bgcolor=colorname.
* background=."image source path ".
* width=value.
* height=value.
* bordercolor=colorname.
* align=left(default)/right/center.
* cellpadding=value:- Maintain spaces inside of cellboxes.
* cellspacing=value:- Maintain spaces between cellboxes.

**SUB\_TAGS FOR <table></table>:**

* **<caption></caption>**: :Caption of the table.
* **<tr></tr>**  :To create number of rows for table.
* **<td></td>** :To define all the table row data information.
* **<th></th> :** To define table columns with Bold Format
* **<thead></thead> :** For Table Header Section Layout.
* **<tfoot></tfoot> :** For Table Footer Section Layout.
* **<tbody>**  :For Table Body Section Layout

**Special Attributes Of <td>, <th>-Tags:-**

* Colspan=number :Merge/Combine Two or More Columns DATA
* Rowspan=number :Merge/Combine Two or More Rows DATA
* Valign=top/middle/bottom :Align Table row data into vertical direction.

***HTML HYPERLINKS*** Create hyperlinks in webpage/websites for navigations, by using one pre-defined html tag. i. e <a></a>-tag. we can call it as anchor Tag.

**Attributes:**

* **href="url/#anchrornameaddress"**:- For Destination address path links.
* **target=”\_\_blank"**: open link webpage in new tab.
* **name="Anchrorname address".**

**Types Of Hyperlinks:**

* **EXTERNAL HYPERLINKS**
* **INTERNAL HYPERLINKS**
* **INLINE HYPERLINKS**

**EXTERNAL HYPERLINKS:**

To navigate from "one website to another website" between 2 different websites using their website url address as Link address. In this Links We create no.of ways it is:

**INTERNAL HYPERLINKS:**

To navigate from "one webpage to another webpage" between number of webpages in same website.

* <a href=”pagename.html”>TEXTLINK</a> |
* <a href=” pagename.html”> <button>ButtonText</button> </a> |
* <a href=” pagename.html”> <img src=”imagesource” /> </a>

**INLINE HYPERLINKS:**

* To navigate or move from Source Link to Target Location point with into same webpage without using scrollbars or pressing up/down arrow-keys through keyboard.
* Create "Target address for destination using name/id attribute", then give that reference address to source point and create hyperlink using href="#anchoraddressname" into <a>-tag.
* <a href=” #anchoraddressname”> SOURCE LINK</a>
* <a name=”anchoraddressname”>---</a>3. HTML H

***HTML FORMS***

To Create User Registration/Login/Search Boxes Forms for webpages using Html Forms Concepts by starts with <FORM></FORM>.The following attributes for <form> Tag are:

* **action**="php/java/.net file\_names with extention
* **method**=get(default)/post
* **target**="\_parent/\_blank"
* **name**="ref\_address”
* **enctype**="multipart/form-data

**Note:- A**ll the <form>-tag attributes, we can use along with php/java/.net

**SUB\_TAGS FOR <FORM>-TAG**

**1) <input />: T**o create Single Line Text Boxes/Radio Buttons/Check Boxes/Buttons/File Fields,..etc for Following Requirements

* **ENTER FIRST\_NAME, LAST\_NAME, USER\_NAME, PASSWORD,**
* **GENDER USING RADIO-BUTTONS**
* **CHECKBOXES FOR HOBBIES**
* **BUTTONS (SUBMIT,RESET ,BUTTON)**
* **FOR MOBILE\_NUMBERS**
* **FOR EMAIL\_ID NAMES..etc**
* **FILE-BROWSING-FORMAT**

**Attributes:**

* **type="text(default)/password/radio/checkbox/button/submit/reset/file"**
* **value="Data" :**To show default\_input data into textbox.
* **maxlength=value :**To controle the maximum textbox\_characters length.
* **readonly=readonly :**To read input\_textbox request as default\_fixed request
* **disabled=disabled** :To apply opactiy format on request
* **checked=checked** :To check radio and checkbox-buttons for defaultselection.

**2) <textarea> </textarea>:** To create Multiple Line Textbox for address input textbox to enter multiple lines information.

**Attributes:**

* **cols=value** :To incre/decre the size of width of the address textbox.
* **rows=value** :To incre/decre the size of height of the address textbox.
* **maxlength=value**

**3) <select> </select>:** To create drop\_down list menu options format by having no.of option names.It has following attributes are:

* **multiple="multiple"**  To select more than one option at a time
* **size=value** To show number of options at a time by default.

**SUB\_TAGS** for **<select></select>**

* **<option>optionname</option>**

To display or present no.of option list names.

* **<optgroup label="groupname">----</optgroup>**

To maintain group\_name for no.of list of option\_names.

**4) <fieldset></fieldset>**: To create border for Form Fields.

**5 )<legend>**</legend>:Caption name of the fieldset.

**6) <label ></label>**: Label names enclose by <label>-tag

**ASSGNMENTS:**

1. GMAIL REGISTRATION FORM ( Both Vertical and Horizantal Format )
2. GMAIL LOGIN FORMS ( Both Vertical and Horizantal Format )

***HTML HEADER SECTION***

To define header\_part section by using <head>-tag and it contains the follwoing tags are:

1) **<title>title name</title> For status of web page tab title.**

2) **<meta /> For Number of uses are there.**

**Attributes:**

* http-equiv="refresh"
* content=value(value means no.of seconds).
* url="webpage\_name"

3) **<style></style>** and **<link />** To integrate css styles to html\_webpage.

4) **<script></script>**:- To integrate js/jquery code to html tags.

**TEST- 1 ) EXAMINATION ON HTML :**

**The following Topics are:**

1. **HTML LISTS**
2. **HTML TABLES**
3. **HTML FORMS**
4. **HTML HYPERLINKS**

**CSS INDEX**

1. **CSS Introduction**
2. **CSS Uses**
3. **CSS Property Declaration**
4. **CSS Style Sheet Levels**

* **Inline CSS Style sheet**
* **Internal CSS Style sheet**
* **External CSS Style sheet**

1. **CSS Selectors Models**

* **Type Selector**
* **Multiple Selector**
* **Decendent Selector**
* **Universal Selector**
* **Class Selector**
* **Id Selector**
* **Pseduo Selector**
* **Attribute Selector**
* **Type with Class Selector**

1. **CSS Properties List**

* **Background Properties**
* **Border Properties**
* **Outline Properties**
* **Dimention Properties**
* **Margin Properties**
* **Padding Properties**
* **Font Properties**
* **Text Properties**
* **Pseduo Properties**
* **Classification Properties**
* **List Properties**
* **Position Properties**
* **Table Properties**
* **Content Properties**

1. **Block Level and Inline Level Elements.**
2. **CSS Div Web Layouts and Convert PSD to Webpage.**
3. **Menus and Image Galleries.**

# CSS INTRODUCTION

* **CSS** stands cascading Style sheet.
* It is not a new language. Its just only Styles for HTML.
* It is style sheet language which is used for to apply styles on HTML webpages.
* CSS can contain properties , that we can call as in HTML "attributes"

# USES OF CSS

* Apply styles on **HTML** **TAGS**
* Reduce **HTML** code size without writing number of **HTML** tags using CSS.
* No need to remember **HTML** tags Attributes if you know about CSS.
* All CSS properties use and apply on any **HTML** tag.
* CSS is RE-USABILITY-CONCEPT: means writing once and use number of times.
* Apply same Styles on more than one **HTML** TAGS. and No.of **HTML** PAGES.
* It is common basic style sheet language for all web-technologies i.e PHP, JAVA, .NET, java Script, jQuery,..etc
* There is no limitations to extend sizes of font using CSS.

**DRAWBACKS OF CSS:**

* while writng CSS styles, then if you make any mistake, it won't return ANY errors.

# CSS PROPERTY DECLARATION

* In HTML, Apply the styles on Tags, We can use Attributes, Like attribute-name=value; But In CSS
* All HTML attributes re updated and Introduced with name is CSS properties with More Features.
* **Syntax: Propertyname : value**

***CSS STYLE SHEETS***

1. **INLINE CSS STYLE SHEET** (Declare Using **style** Attribute)
2. **INTERNAL CSS STYLE SHEET** (Declare Using <style></style>)
3. **EXTERNAL CSS STYLE SHEET.**(Declare Using <link />)

**1) INLINE CSS STYLE SHEET:**

To Apply CSS styles on HTML Tags the use **"style” attribute** which is declaring inside of HTML-tag and with CSS properties.

**Syntax:**

**To apply only single CSS property style using inline CSS style sheet**

<tagname style="property:value" >---</tagname>

**To apply Multiple CSS styles which is seperated by semi-colon (;)**

<tagname style="property:value; property:value; property:value;....">---</tagname>

**2) INTERNAL CSS STYLE SHEET:**

* Apply CSS Styles to Single Web page.
* Using <style></style>-Tag declare into <head></head>-section .
* In This style sheet, Declare Different types of selectors which is writing or declarating with in the <style></style>.

**Syntax:**

<head> <style>-------Declare internal CSS style sheet selectors-----</style> </head>

**3) EXTERNAL CSS STYLE SHEET:**

* Apply SAME CSS Styles to Single or More than one Web pages.
* Then write CSS styles in new page with filename extention is **.CSS**
* Connect External CSS file link to web pages Using <link />Tag and which is declare into <head></head>.
* EXTERNAL CSS FILE LINK SYNTAX: <link href=”filename.css” rel=”stylesheet” />

***TYPES OF CSS PROPERTIES***

* **BACKGROUND Properties**
* **BORDER Properties**
* **OUTLINE Properties**
* **TEXT Properties**
* **FONT Properties**
* **LIST Properties**
* **MARGIN Properties**
* **PADDING Properties**
* **DIMENTION Properties**
* **PSEDUO Properties**
* **CLASSIFICATION Properties**
* **POSITION Properties**
* **TABLE Properties**
* **CONTENT Properties**

**1) BACKGROUND PROPERTIES:**

* background-color:colorname;
* background-image:url(imagesource)
* background-repeat:repeat(default)/no-repeat/repeat-x/repeat-y
* background-position:top/center/bottom left/right/center
* background-attachment:fixed;

**Short hand for background:**

* background: colorname url(source) repeatvalue positionvalue attachmentvalue;

**2) BORDER PROPERTIES:**

* border-top/bottom/left/right-style:stylename;
* border-top/bottom/left/right-color:colorname;
* border-top/bottom/left/right-width:value(px);

**NOTE:**

Border styles are: none(default) /groove/ solid/ double /dotted /dashed /inset /outset /ridge.

**Shorthand for Border style,color,width**

* border-top:stylename colorname widthvalue;
* border-left:stylename colorname widthvalue;
* border-right:stylename colorname widthvalue;
* border-bottom:stylename colorname widthvalue;

**VERY SHORT HAND PROPEWRTIES FOR BORDER**

* border-style:stylename;
* border-color:colorname;
* border-width:value(px)
* border-style:style1 style2; (2 Border styles)
* border-style:style1 style2 style3; (3 Border styles)
* border-style: style1 style2 style3 style4; (4 Border Styles)

**VERY VERY SHORT-HAND PROPERTY FOR BORDER**

* border:stylename colorname widthvalue(px);

**3) OUTLINE PROPERTIES**

* outline-style:stylename
* outline-color:colorname
* outline-width:value(px)

**ShortHand for Outline:**

* outline:stylename colorname widthvalue;

**4) FONT PROPERTIES:**

* font-family:font\_name;
* font-size:value(px)
* font-weight:bold/bolder/smaller/medium/100-900

**5) TEXT PROPERTIES**

* color:colorname;
* text-align:left/right/center
* text-decoration:none/underline/overline/line-thorugh/blink
* direction:ltr(left-to-right)/rtl(right-to-left)
* line-height:value(px)
* letter-spacing:value(px)
* word-spacing:value(px)
* text-indent:value(px)
* text-transform:normal/lowercase/uppercase
* white-space:pre/nowrap/normal
* text-shadow:x-axis y-axis blurvalue colorname;

**6) LIST-PROPERTIES:**

* list-style-type: typename;
* list-style-position: inside/outside
* list-style-image:url(image-name.extention)

**Shorthand for list-style:**

* list-style: typename/url(imagename.ext) position\_name;

**NOTE:**

List types are: disc/square /circle /decimal /decimal-leading-zero/lower-roman/upper-roman /lower -alpha /upper-alpha /lower-greek/lower-latin/upper-latin/katakana..

**7) PSEDUO PROPERTIES:**

* :link
* :active
* :visited
* :hover
* :focus
* :first-line
* :first-letter
* :before
* :after

**8) MARGIN PROPERTIES:**

* margin-top:value(px)
* margin-left:value(px)
* margin-right:value(px)
* margin-bottom:value(px)

**Shorthand for Margin:**

* margin:value; (If single margin value)
* margin: value1 value2; (If 2 margin values)
* margin: value1 value2 value3; (If 3 margin values)
* margin: value1 value2 value3 value4; (If 4 margin values).

**9) PADDING PROPERTIES:**

* padding-top:value;
* padding-left:value;
* padding-right:value;
* padding-bottom:value;

**Shorthand for padding:**

* padding: value; (If Single Padding value)
* padding: value1 value2; (If 2 Padding values)
* padding: value1 value2 value3; (If 3 Padding values)
* padding: value1 value2 value3 value4; (If 4 Padding values)

**10) DIMENTION PROPERTIES:**

* width:value
* height:value
* min-width:value
* min-height:value
* max-width:value
* max-height:value

**11) CLASSIFICATION PROPERTIES:**

* cursor:move/cross-hai/pointer/wait/help/e-resize/...
* overflow:hidden/auto/scroll
* visibility:hidden/visible
* float:left/right/both
* disply:none/block/inline
* z-index:value

**12) POSITION PROPERTIES:**

* position:relative(default)/absolute/fixed;
* top:value(px)
* bottom:value(px)
* left:value(px)
* right:value(px)

**13) TABLE PROPERTIES :**

* border-collapse: *collapse/separate*
* border-spacing: value ( px )
* caption-side: *top/bottom/left/right*
* empty-cells: *show/hide*
* table-layout: *auto/fixed*

**CSS SELECTORS TYPES**

1. **SINGLE/TYPE/ELEMENT SELECTOR**
2. **MULTIPLE SELECTOR**
3. **SELECTOR WITH CLASS\_NAME**
4. **UNIVERSAL SELECTOR**
5. **DECENDENT SELECTOR**
6. **PSEDUO SELECTOR**
7. **ID-SELECTOR**
8. **CLASS SELECTORS**
9. **ATTRIBUTE SELECTOR**

**1) ELEMENT SELECTOR :**

**Apply CSS styles on selected HTML element.**

**Syntax:**

Selector {

propert1:value;

propert2:value;…

}

**2) MULTIPLE SELECTOR :**

* Apply same styles on different tags without writing same properties again.
* Now declare all selected selectors declared by separating with cama (,)

**Syntax:**

selector, selector, selector,...

{

Property:value;

Property:value; ….

}

**3) SELECTOR WITH CLASSNAME SELECTOR:**

If we declare same tag more than once either with same or diff content, then to apply CSS styles differently on CSS tags, along with selector then create developer customize selector i.e class name which is precedded by dot-symbol(.) to seperate both selected selector and class name and which is access by class attribute from CSS-tags.

**Syntax:**

selector.classname {

property:value;

property:value;…

}

**4) UNIVERSAL SELECTOR:** To apply same CSS property style effects on Entire body-section CSS-tags.it is used for to remove gaps or spaces between the CSS tags content using margin and padding properties.

**Syntax:**

\*{

property:value;

property:value;….

}

**5) DECENDENT SELECTOR:**

Apply CSS styles on either child or sub-child or super sub child tags... which is declared inside of their parent tag.

**Syntax:**

parentselector childselector subchildselector....

{

property:value;

property:value;…

}

**6) PSEDUO SELECTOR:** Apply CSS styles with animations.

**Syntax:**

selector:pseudoproperty

{

property:value;

property:value;..

}

**7) CLASS SELECTOR:**

To apply same CSS styles on any CSS-tags using class attribute either once more than once by declaring classname from CSS-tags.

**Syntax:**

.classsname

{

property:value;

property:value;..

}

**8) ID SELECTOR: I**t’s same like as class selector, but we can declare id selector which is preceded with ‘#’symbol. It is used for to create website layouts using <div></div> .

**Syntax**

#idselector

{

property:value;

property:value;……

}

1. **ATTRIBUTE SELECTORS :**

* **Selector[attribute]**
* **Selector[attribute=value]**
* **Selector[attribute~=value]**
* **Selector[attribute|=value]**
* **Selector[attribute^=value]**
* **Selector[attribute$=value]**
* **Selector[attribute\*=value]**

**CSS3 INDEX**

1. **CSS3 VENDOR PREFIXNAMES**
2. **CSS3 NEW BACKGROUNDS**
3. **CSS3 NEW BORDERS**
4. **CSS3 SHADOW EFFECTS**
5. **CSS3 GRADIENTS**
6. **CSS3 NEW COLORS**
7. **CSS3 MULTI COLUMNS**
8. **CSS3 NEW FONTS**
9. **CSS3 2D/3D TRANSFORMATIONS**
10. **CSS3 TRANSITIONS**
11. **CSS3 ANIMATIONS**
12. **CSS3 TEXT EFFECTS**
13. **CSS3 USER INTERFACE**
14. **CSS3 FLEXBOX**
15. **CSS3 FILTERS**
16. **CSS3 NEW PSEDUO PROPERTIES**
17. **CSS3 NEW SELECTORS**
18. **CSS3 MEDIA QUERRIES**
19. **CSS3 RWD (RESPONSIVE WEB DESIGN)**

**CSS3 VENDOR PREFIX NAMES**

| **Prefix** | **Organisation** |
| --- | --- |
| -ms- | Microsoft |
| mso- | Microsoft Office |
| -moz- | Mozilla Foundation ([Gecko-based browsers)](http://en.wikipedia.org/wiki/Gecko_(layout_engine)) |
| -o- | Opera Software |
| -atsc- | Advanced Television Standards Committee |
| -wap- | The WAP Forum |
| -webkit- | Safari (and other [WebKit-based browsers)](http://trac.webkit.org/projects/webkit/wiki/Applications%20using%20WebKit) |
| -khtml- | Konqueror browser |

To See More: <http://stackoverflow.com/questions/5411026/list-of-css-vendor-prefixes>

**CSS3 NEW BACKGROUND PROPERTIES**

**1) Background-size Property:** It allows you to specify the size of background images.

Background-size: value(px); (and) Background-size: value1(px) value2 (px);

**2) Background-clip Property:** It allows you to specifies the painting area of the background.

Background-clip: border-box/ padding-box/ content-box;

**3) Background-origin Property:** This property allows you to specifies where the background image is positioned.

Background-origin: border-box/ padding-box/ content-box;

**4) Multiple Background Images:** This property allows you to one or more background images for an element.

Background-image: url(--), url(--), url(--)…;

**CSS3 NEW BORDER PROPERTIES**

**1) Rounded Borders using Border-radius:** You can give any element "rounded corners.

* Border-radius:value(px); (1 value Rounded Border)
* Border-radius:value(px) value(px); (2 values Rounded Border)
* Border-radius:value(px) value(px) value(px); (3 values Rounded Border)
* Border-radius:value(px) value(px) value(px) value(px); (4 values Rounded Border)

**2) Border Images using border-image:** To specify an image to be used instead of the normal border around an element.

1. **To Specifies the path to the image to be used as a border.**

Border-image-source:url(magesource.extention);

1. **To Specifies how to slice the border image.**

Border-image-slice:value(number/%);

1. **To Specifies whether the border image should be repeated, rounded or stretched.**

Border-image-repeat:round/stretch;

1. **To Specifies the amount by which the border image area extends beyond the border box.**

Border-image-outset:value(px); (1/2/3/4 values possibles)

1. **To Specifies the widths of the border image.**

Border-image-width:value(px);

1. **Shorthand Property :**

Border-image:source slice repeat outset width;

**CSS3 NEW SHADOW EFFECTS PROPERTIES:**

* **Text-Shadow** property applies adds one or more shadow to text

Text-shadow: X-axis Y-axis Blur-value Blur-offset-value color;

* **Box-shadow** property applies adds one or more shadow to elements.

Box-shadow: X-axis Y-axis Blur-value Blur-offset-value color;

**CSS3 GRADIENTS COLORS:**

To display smooth transitions between two or more specified colors.CSS3 defines two types of gradients:

1. **Linear Gradients (goes down/up/left/right/diagonally)**
2. **Radial Gradients (defined by their center)**
3. **LINEAR GRADIENTS :**

You must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an **angle**) along with the gradient effect. **Syntax:** **background: linear-gradient(direction,** **color-stop1,** **color-stop2, ...);**

**REPEATING A LINEAR GRADIENTS:**

repeating-linear-gradient() function is used to repeat linear gradients

background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%);

1. **RADIAL GRADIENTS :**

A radial gradient is defined by its center. To create a radial gradient you must also define at least two color stops. **Syntax:** **background: radial-gradient(shape size****at****position, start-color, ..., last-color);**

The shape parameter defines the shape. It can take the value circle or ellipse. The default value is ellipse.

**REPEATING RADIAL GRADIENTS:**

The repeating-radial-gradient() function is used to repeat radial gradients.

background: -webkit-repeating-radial-gradient(red, yellow 10%, green 15%);

**USE OF DIFFERENT SIZE KEYWORDS :**

The size parameter defines the size of the gradient. It can take four values:

* **Closest-side**
* **Farthest-side**
* **Closest-corner**
* **Farthest-corner**

background: -webkit-radial-gradient(60% 55%, closest-side, red, yellow, black);

background: radial-gradient(closest-side at 60% 55%, red, yellow, black);

background: -webkit-radial-gradient(60% 55%, farthest-side, red, yellow,

background: radial-gradient(farthest-side at 60% 55%, red, yellow, black);

**CSS3 NEW COLORS:**

**Named Color: red,green,..etc**

**RGB Color: rgb(255,0,255)**

**Hexa Decimal Color: #ff00ff or #f0f;**

**RGBA Color: rgba(255,0,255,0.5)**

**OPACTIY: b/w 0 to 1 value for Transperency.**

**HSL Color:** Hue, Saturation and Lightness

1. Hue is a degree on the color wheel (from 0 to 360):
   * 0 (or 360) is red
   * 120 is green
   * 240 is blue
2. Saturation is a percentage value: 100% is the full color.
3. Lightness is also a percentage; 0% is dark (black) and 100% is white.

**HSLA Color:**hsla(hue, saturation, lightness, alpha).Where the alpha parameter defines the opacity.

**CSS3 MULTI COLUMNS**

It allows easy definition of multiple columns of text - just like in newspapers.

The following multi-column properties are:

**To Specifies the number of columns an element should be divided into.**

* **column-count:number**

**To specifies the gap between the columns.**

* **column-gap:number**

**To specifies the style of the rule between columns:**

* **column-rule-style**
* **column-rule-width**
* **column-rule-color**
* **column-rule**

**To specifies how many columns an element should span across.**

* column-span

**To Specifies a suggested, optimal width for the columns**

* column-width

**A shorthand property for setting column-width and column-count**

* columns: 100px 3

**CSS3 WEB FONTS**

Web fonts allow Web designers to use fonts that are not installed on the user's computer. Your "own" fonts are defined within the CSS3 @font-face rule.

**Different Font Formats:**

**i) TrueType Fonts (TTF)**

TrueType is a font standard developed in the late 1980s, by Apple and Microsoft. TrueType is the most common font format for both the Mac OS and Microsoft Windows operating systems.

**ii) OpenType Fonts (OTF)**

OpenType is a format for scalable computer fonts. It was built on TrueType, and is a registered trademark of Microsoft. OpenType fonts are used commonly today on the major computer platforms.

**iii) The Web Open Font Format (WOFF)**

WOFF is a font format for use in web pages. It was developed in 2009, and is now a W3C Recommendation. WOFF is essentially OpenType or TrueType with compression and additional metadata. The goal is to support font distribution from a server to a client over a network with bandwidth constraints.

**iv) The Web Open Font Format (WOFF 2.0)**

TrueType/OpenType font that provides better compression than WOFF 1.0.

**v) SVG Fonts/Shapes**

SVG fonts allow SVG to be used as glyphs when displaying text. The SVG 1.1 specification define a font module that allows the creation of fonts within an SVG document. You can also apply CSS to SVG documents, and the @font-face rule can be applied to text in SVG documents.

**vi) Embedded OpenType Fonts (EOT)**

EOT fonts are a compact form of OpenType fonts designed by Microsoft for use as embedded fonts on web pages. In css3, If You want to apply custom fonts to be loaded on a webpage. Once added to a stylesheet, the rule instructs the browser to download the font from where it is hosted, then display it as specified in the CSS.Then use @font-face rules.

**STEP-1)** Download example Fonts from: "http://webfonts.ffonts.net/"

**STEP-2)** Declare as selector using "@font-face" constant.

@font-face{ ----- }

**STEP-3)** Write the following properties inside of @font-face

@font-face{

font-family:'fontname';

src:url('fontname.ttf.woff') format('woff');

}

**STEP-4)** Now Use Custom Css fonts on Selected Tags

SELECTOR

{

font-family:fontname;

}

**CSS3 2D/3D TRANSFORMATIONS:**

A transformation is an effect that lets an element change shape, size and position.Css3 supports 2D and 3D transformations.

CSS3 transforms allow you to translate, rotate, scale, and skew elements.

**CSS3 2D TRANSFORMS:** The following 2D transformation methods

* rotate()
* scale()
* skew()
* translate()
* matrix()

**CSS3 Transform with rotate() Method**

It rotates an element clockwise or counter-clockwise according to a given degree.

Transform: rotate(+/-value deg);

**CSS3 Transform with scale() Method**

It increases or decreases the size of an element (according to the parameters given for the width and height).

Transform: scale(value); or Transform: scale(value,value);

**CSS3 Transform with skew() Method**

It skews an element along the X and Y-axis by the given angles.

Transform: skew(value deg, value deg);

**CSS3 3D TRANSFORMS:** The following 3D transformation methods:

To change the position on transformed elements. This property must be used together with the [transform](http://www.w3schools.com/cssref/css3_pr_transform.asp) property. The following syntax is:

[**transform-origin**](http://www.w3schools.com/cssref/css3_pr_transform-origin.asp) **: *x-axis y-axis z-axis*;**

Here: *x-axis Means* Defines where the view is placed at the x-axis. Possible values:

* left | center | right | length | %

*Y-axis Means* Defines where the view is placed at the x-axis. Possible values:

* left | center | right | length | %

*Z-axis Means* Defines where the view is placed at the Z-axis. Possible values: *Length value.*

The **transform-style** property specifies how nested elements are rendered in 3D space. The following syntax is:

[transform-style](http://www.w3schools.com/cssref/css3_pr_transform-style.asp): flat|preserve-3d.

**flat** Means Specifies that child elements will NOT preserve its 3D position.

**preserve-3d** Means Specifies that child elements will preserve its 3D position.

The **perspective property** defines how many pixels a 3D element is placed from the view. This property allows you to change the perspective on how 3D elements are viewed

When defining the perspective property for an element, it is the CHILD elements that get the perspective view, NOT the element itself.

**Note:** The perspective property only affects 3D transformed elements

Use this property together with the [perspective-origin](http://www.w3schools.com/cssref/css3_pr_perspective-origin.asp) property, which allows you to change the bottom position of 3D elements. Syntax is: **perspective:** ***length*|none;**

The **perspective-origin** property defines where a 3D element is based in the x- and the y-axis. This property allows you to change the bottom position of 3D elements.

When defining the perspective-origin property for an element, it is the CHILD elements that are positioned, NOT the element itself.

**Note:** This property must be used together with the [perspective](http://www.w3schools.com/cssref/css3_pr_perspective.asp) property, and only affects 3D transformed elements. **perspective-origin:** **X-axis Y-axis;**

**Here: X-axis** Defining where the view is placed at the X-axis The possible value are left| center| right| length| %.

**Y-axis** Defining where the view is placed at the Y-axis The possible value are top | center | bottom | length| %.

The **backface-visibility** property defines whether or not an element should be visible when not facing the screen. This property is useful when an element is rotated, and you do not want to see its backside. Syntax is: **backface-visibility: visible|hidden.**

**CSS3 TRANSITIONS**

CSS3 transitions allows you to change property values smoothly (from one value to another), over a given duration. The following table lists all the transition properties

**1) transition-property: CSS property name | all;**

Specifies the name of the CSS property the transition effect is for

**2) transition-duration: number s;**

Specifies how many seconds or milliseconds a transition effect takes to complete

**3) transition-delay: number s;**

Specifies a delay (in seconds) for the transition effect.

**4) transition-timing-function: ease|linear|ease-in|ease-out|ease-in-out**

Specifies the speed curve of the transition effect.

**5) transition: css-property duration delay timing-function;**

A shorthand property for setting the four transition properties into a single property

**CSS3 ANIMATIONS:**

CSS3 animations allows animation of most HTML elements without using JavaScript or Flash.

To use CSS3 animation, you must first specify some keyframes for the animation. Keyframes hold what styles the element will have at certain times.

The following @keyframes rule and all the animation properties are:

**1)** **@keyframes keyframename { ---- }**

To Specifies the animation code using **From** and **to** selectors.

**2)** **animation-name: keyframename.**

To Specifies the name of the @keyframes animation.

**3)** **animation-duration: number s.**

To Specifies how many seconds|milliseconds an animation takes to complete one cycle.

**4) animation-delay: number s;**

To Specifies a delay for the start of an animation.

**5) animation-direction: reverse|alternate|alternate-reverse**.

To Specifies whether an animation should play in reverse direction or alternate cycles.

**6) animation-iteration-count: number |infinite.**

To Specifies the number of times an animation should be played.

**7)animation-timing-function:ease|linear|ease-in|ease-out|ease-in-out.**

To Specifies the speed curve of the animation.

**8) animation-play-state: play |paused**

To Specifies whether the animation is running or paused.

**CSS3 TEXT EFFECTS**

CSS3 contains several new text features. The following text properties:

1) **text-overflow: clip|ellipsis|*string*|initial|inherit;**

It specifies how overflowed content that is not displayed should be signaled to the user.

2) word-wrap**:value;**

It allows long words to be able to be broken and wrap onto the next line.

**CSS3 USER INTERFACE**

CSS3 has new user interface features such as resizing elements, outlines, and box sizing. The following user interface properties:

**1) resize: none | both| horizontal| vertical;**

**2) outline-offset:** ***length*| initial| inherit;**

**3) box-sizing: content-box| border-box| initial| inherit;**

**CSS3 FLEXBOX**

Flexible boxes, or flexbox, is a new layout mode in CSS3. Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.

For many applications, the flexible box model provides an improvement over the block model in that it does not use floats, nor do the flex container's margins collapse with the margins of its contents. **Syntax** is: **Display:flex; or Display:-prefix-flex;**

**CSS3 FILTERS:**

The CSS filter property adds visual effects (like blur and saturation) to any HTML element - often <img /> elements.

**Syntax:**

**filter: none | blur() | brightness() | contrast() | drop-shadow() | grayscale() | hue-rotate() | invert() | opacity() | saturate() | sepia() | url();**

**1) filter: grayscale(100%)**

Change the color of all images to black and white (100% gray):

**2) filter: blur(30px)**

Applies a blur effect to the image. A larger value will create more blur

**3) filter: brightness(200%);**

Adjusts the brightness of the image. 0% (0) will make the image completely black. 100% (1) is default and represents the original image. Values over 100% will provide brighter results.

**4) filter: contrast(200%)**

Adjusts the contrast of the image. 0% (0) will make the image completely black. 100% (1) is default and represents the original image. Values over 100% will provide results with less contrast.

**5) filter:drop-shadow(h-shadow v-shadow blur color);**

This filter is similar to the box-shadow property.

**6) filter: hue-rotate(90deg);**

Applies a hue rotation on the image. The value defines the number of degrees around the color circle the image samples will be adjusted. 0deg is default, and represents the original image. Maximum value is 360deg.

**7) filter: invert(100%);**

Inverts the samples in the image. 0% (0) is default and represents the original image. 100% (1) will make the image completely inverted. Negative values are not allowed.

**8) filter: opacity(30%);**

Sets the opacity level for the image. The opacity-level describes the transparency-level, where: 0% (0) is completely transparent. 100% (1) is default and represents the original image (no transparency).

**9) filter: saturate(800%);**

Saturates the image. 0% (0) will make the image completely un-saturated. 100% (1) is default and represents the original image. Values over 100% provides super-saturated results.

**10)** **filter: sepia(100%);**

Converts the image to sepia. 0% (0) is default and represents the original image. 100% (1) will make the image completely sepia.

**11) filter: contrast(200%) brightness(150%);**

To use multiple filters, separate each filter with a space.

**CSS3 NEW PSEDUO CLASSES**

**1) :nth-child(N)**

Matches elements on the basis of their positions within a parent element’s list of child elements.

**2) :nth-last-child(N)**

Matches elements on the basis of their positions within a parent element’s list of child elements

**3) :nth-of-type(N)**

Matches elements on the basis of their positions within a parent element’s list of child elements of the same type

**4) :nth-last-of-type(N)**

Matches elements on the basis of their positions within a parent element’s list of child elements of the same type.

**5) :last-child**

Matches an element that’s the last child element of its parent element

**6) :first-of-type**

Matches the first child element of the specified element type.

**7) :last-of-type**

Matches the last child element of the specified element type.

**8) :only-child**

Matches an element if it’s the only child element of its parent.

**9) :only-of-type**

Matches an element that’s the only child element of its type.

**10) :root**

Matches the element that’s the root element of the document

**11) :empty**

Matches elements that have no children

**12) :target**

Matches an element that’s the target of a fragment identifier in the document’s URI.

**13) :enabled**

Matches user interface elements that are enabled

**14) :disabled**

Matches user interface elements that are disabled

**15) :checked**

Matches elements like checkboxes or radio buttons that are checked

**16) :not(S)**

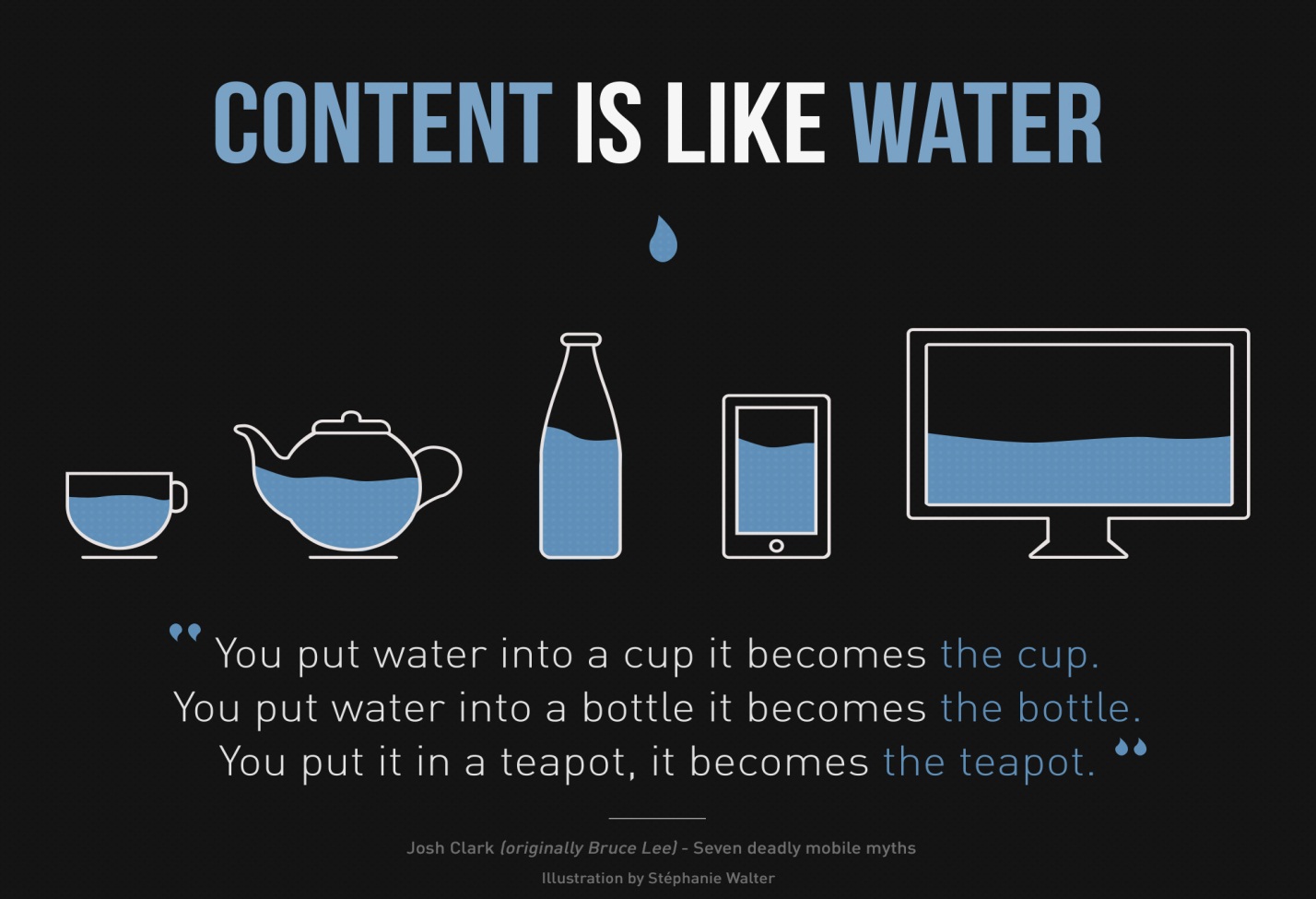
Matches elements that aren’t matched by the specified selector

**RESPONSIVE WEB DESIGN**

**Q) WHAT IS RWD ?**

## Responsive Web Design (RWD) is an approach of laying-out and coding a website such that the website provides an optimal viewing experience — ease of reading and navigation with a minimum of resizing, panning, and scrolling — across a wide range of devices (from desktop computer monitors to mobile phones).

The term [Responsive Web Design](http://www.alistapart.com/articles/responsive-web-design/)was coined by [**Ethan Marcotte**](http://unstoppablerobotninja.com/) and is the practice of using [fluid grids](http://www.alistapart.com/articles/fluidgrids/), [flexible images](http://unstoppablerobotninja.com/entry/fluid-images), and media queries to progressively enhance a web page for different viewing contexts.



**Q) WHY IT IS SO IMPORTANT ?**

There is NO doubt that **Responsive Web Design** is very **important** for today's **webdesign** point of view. Smartphone and tablet adoption rapidly increases, so does the **importance** of mobile-friendly **websites**. Smartphones and tablets have changed the approach toward **design** and user experience.

The designer creating a Responsive Design should ensure that the website’s navigation elements, screen-layouts, text, images, audio/video players and other UI elements re-adjust themselves on a variety of devices. Thus, one need not spend extra time and money in creating and maintaining one “mobile-site version” and another “desktop-site version” of her website.

**Q) Reasons Why Responsive Web Design is the Best Option Now a Days?**

### 1. Super Flexible :

## Responsive web design sites are fluid, meaning the content moves freely across all screen resolutions and all devices. Both the grids and the images are fluid. Just as a liquid spreads out or draws in to allow its content to fill an allotted space and retain its appearance, responsive web design’s fluidity achieves the same result with website content on a device screen.



### 2. Excellent User Experience

## While, content is king and discover ability of content are foremost success metrics, it is the user experience that enables visitors to consume content on any website through the device of their choice and preference, anytime. Thus, responsive web design is about providing the optimal user experience irrespective of whether they use a desktop computer, a smartphone, a tablet or a smart-TV. Responsive web design accommodates the busy professional during the day and the wide-awake college student needing access to your site anytime. No scrolling or resizing is needed for any visitor to access your website from their favorite device.



### 3. Cost Effective

# The advantages of having a single site that conforms to the need of all devices are significant when compared to having two separate websites. One website costs less than two, and the savings can be substantial. Sites designed solely for mobile device traffic don’t offer the advanced navigational techniques found in traditional websites, and they also require the user to maintain two separate web addresses for your site. This is inconvenient for most people and can cause them to check out the competition’s website. Responsive web design enhances SEO efforts by having all your visitors directed to a single site no matter what they prefer to use as a device.

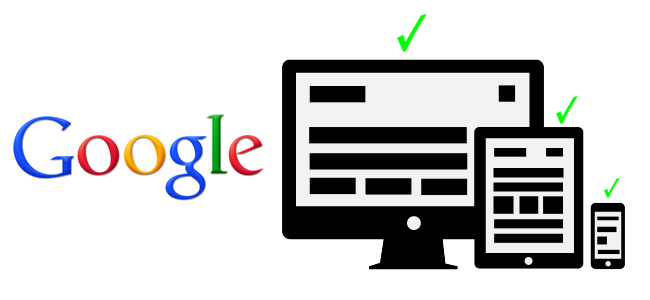


**4. It is Recommended By Google :**

## With 67 percent search market share, when Google speaks, search marketers listen. Google states that responsive web design is its recommended mobile configuration, and even goes so far as to refer to responsive web design as the industry best practice.

## This is because responsive design sites have one URL and the same HTML, regardless of device, which makes it easier and more efficient for Google to crawl, index, and organize content. Contrast this with a separate mobile site which has a different URL and different HTML than its desktop counterpart, requiring Google to crawl and index multiple versions of the same site.

## Additionally, Google prefers responsive web design because content that lives on one website and one URL is much easier for users to share, interact with, and link to than content that lives on a separate mobile site. Take for example a mobile user who shares content from a mobile site with a friend on Facebook who then accesses that content using a desktop, which results in that user viewing a stripped down mobile site on their desktop.

**5. Very Easy to manage :**

## Having a separate desktop and mobile site requires having separate SEO campaigns. Managing one site and one SEO campaign is far easier than managing two sites and two SEO campaigns. This is a key advantage a responsive website has over a separate mobile site. That being said, there are benefits to having a mobile-specific SEO strategy, such as optimizing for keywords that are more likely to be searched when someone is on their smartphone.

## For example, someone performing a mobile search for a local restaurant may be more inclined to use the word “nearby” in their search query. However, a separate mobile site is not a requirement for a mobile SEO strategy, and there’s no reason why mobile-specific keywords can’t be incorporated into a responsive design site as well.

****

### 5. Mobile Usage is on the Raise :

## Currently, more than 58% of American adults own a smartphone and almost 60% of all website traffic is from mobile devices. In fact, there are currently more mobile devices on earth then their are people. And every month mobile usage continues to grow, so every month more and more prospects and customers will view your website from a mobile device. If their experience viewing and interacting with your site is poor, they'll likely have a lower option of your brand, and they'll also be more likely to visit a competitor's site.

### 6. Shopping on Mobile devices is steadly growing :

Online shopping is easier than hopping in the car and driving to the store and it is even easier if you can do it in your favorite chair, while watching TV. 80% of consumers regularly use their smartphones to shop online. And 70% of shoppers now use mobile phones while in stores during the holidays. If your products and services aren't easy to view from a phone, you're missing out on an opportunity.

### 7 Shopping on Mobile devices is steadly growing :

### Over 55% of social media consumption now happens on mobile devices, so sharing links from social media sites like, Facebook, YouTube, Twitter, or Google Plus to your website will mean even more traffic and viewing of your website from mobile devices. So if you have a social marketing strategy and want to leverage social sharing of content, get responsive.

### 8. A Better Bathroom Experience :

Finally, the most disturbing stat about the growth of mobile usage: 75% of Americans bring their phones to the bathroom! That's certainly gross and it may also be an indicator of the downfall of mankind, but it is true. And if people are going to browse from the restroom, you can at least provide them with a positive user experience.

**CSS3 MEDIA QUERRIES**

**Q) What is Media Queries ?**

**Media Queries** is a CSS3 module allowing content rendering to adapt to conditions such as screen resolution (e.g. smartphone screen vs. computer screen). It became a W3C recommended standard in June 2012. and is a cornerstone technology of Responsive web design.

Media Queries ... A collection of inspirational websites using media queries and responsive web design. **Ex: http://mediaqueri.es/** and [**http://mediaqueri.es/all.txt**](http://mediaqueri.es/all.txt)

**For More Explanation Plz Check URLS:**

<http://www.javascriptkit.com/dhtmltutors/cssmediaqueries.shtml>

<http://webdesignerwall.com/tutorials/css3-media-queries>

The **@media** rule is used to define different style rules for different mediatypes/ devices. In CSS2 this was called media types, while in CSS3 it is called media queries.

Media queries look at the capability of the device, and can be used to check many things, such as: width and height of the viewport.

By restricting CSS rules to a certain width of the device displaying a web page, one can tailor the page's representation to devices (i.e. smartphones, tablets, netbooks, and desktops) with varying screen resolution.

## Media Query Syntax :

The syntax of a media query is the same no matter if you use it inside a link element, after an @import instruction, or inside a CSS style sheet. Media queries follow this syntax :

[logic] [media] [and (condition)] [and (condition)] ...

**Here Few Media Query Examples :**

@media only screen and (max-width: 600px)

{

/\* CSS rules for browser widths equal to or less than 600px \*/

body { background-color: #ffffff; }

}

@media only screen and (min-width: 601px) and (max-width: 1200px)

{

/\* CSS rules for browser widths from 601px to 1200 px \*/

body { background-color: #ff0000; }

}

@media only screen and (min-width: 1201px)

{

/\* CSS rules for browser widths from 1201px and up \*/

body { background-color: #0000ff; }   
 }

These three media queries match different browser window widths. The first media query matches all browser widths less than or equal to 600 pixels. That means that all CSS rules listed inside the body of the media query (inside { ... } ) will only be applied if the browser window width is 600px or less.

The second media query matches browser window widths from 601 pixels to 1200 pixels. CSS rules inside the body of the media query will only be applied if the browser window width falls in this interval.

The third media query matches browser window widths from 1201 pixels and up. CSS rules inside the body of the media query will only be applied if the browser window width are 1201 pixels or more.

In the examples each media query only has a single CSS rule inside the body, setting the background color. In a real web application or website you could have more CSS rules.

For the  [logic] token you can use one of the values :

|  |  |
| --- | --- |
| **Value** | **Description** |
| only | The value only means that this media query *only* matches a certain media type (the next parameter in the media query). |
| not | The value not means that this media query matches all other than a certain media type (the next parameter in the media query). |

The the [media]  token you can use one of these values :

|  |  |
| --- | --- |
| **Value** | **Description** |
| screen | Matches all computer screens. Both on desktop computers, laptops, tablets, smart phones and TVs. |
| projection | Matches projection devices (projectors used in meeting rooms etc.). |
| print | Matches when a user clicks "print" for the page. |

There are more media types, but not all of them are supported by the browser at the time of writing. You can find a full list at [**http://www.w3.org/TR/CSS21/media.html**](http://www.w3.org/TR/CSS21/media.html).

The [and (condition)] blocks set conditions for the screen. For instance, you can use these properties inside a condition block.

|  |  |
| --- | --- |
| **Value** | **Description** |
| width | Specifies the width of the browser window this media query matches. |
| min-width | Specifies the minimum browser window width this media query matches. |
| max-width | Specifies the maximum browser window width this media query matches. |
| height | Specifies the height of the browser window this media query matches. |
| min-height | Specifies the minimum browser window height this media query matches. |
| max-height | Specifies the maximum browser window height this media query matches. |
| device-width | Specifies the width of the device (e.g. monitor width or smart phone screen width) this media query matches. |
| min-device-width | Specifies the minimum device width this media query matches. |
| max-device-width | Specifies the maximum device width this media query matches. |
| device-height | Specifies the height of the device (e.g. monitor height or smart phone screen height) this media query matches. |
| min-device-height | Specifies the minimum device height this media query matches. |
| max-device-height | Specifies the maximum device height this media query matches. |
| orientation | Matches the orientation of the device viewing the page. Can be set to landscape orportrait . |
| aspect-ratio | Specifies what aspect ratio of the browser window this media query matches. Aspect ratio is the ratio between the width and height (e.g. 4/3 or 16/9). |
| device-aspect-ratio | Specifies what aspect ratio of the device screen this media query matches. Aspect ratio is the ratio between the width and height (e.g. 4/3 or 16/9). |
| -webkit-device-pixel-ratio | Specifies what device pixel ratio this media query matches. Only works on webkit (Chrome / Safari), but that is both Android and iOS (as far as I know). Several mobile devices have a device pixel ratio of 1.5 or higher, meaning there is a 1.5 relation between the actual device width / height of the device, and the width / height the browser tells your application it has. |
| -webkit-max-device-pixel-ratio | Specifies what maximum device pixel ratio this media query matches. Only works on webkit (Chrome / Safari), but that is both Android and iOS (as far as I know). |
|  |  |

### Q) Types of Declarations For Media Queries :

### CSS media queries can be inserted in your HTML pages in the following ways :

### 1) Inserted into a <link> element which refers to a CSS style sheet.

### 2) Inserted before an @import CSS instruction in CSS style sheet.

### 3) Inserted inside a CSS style sheet.

### Inserted into a <link> element which refers to a CSS style sheet:

<link rel="stylesheet" href="smallscreen.css" media="only screen and (max-width: 480px)" />

### Inserted before an @import CSS instruction in CSS Style Sheet :

@import "smallscreen.css" only screen and (max-width: 480px);

### 3) Inserted inside a CSS style sheet :

<style type="text/css"> @media only screen and (max-width:480px) { ---- } </style>

### Q) What are the screen resolutions do you use while taking screenshots ?

* **Smartphone Sizes ( Between 320px to 480px )**
* **Tablet Sizes ( 768px and Up )**
* **Laptops Sizes ( 1024px )**
* **Desktop Sizes ( 1600px )**

[**http://stackoverflow.com/questions/6370690/media-queries-how-to-target-desktop-tablet-and-mobile**](http://stackoverflow.com/questions/6370690/media-queries-how-to-target-desktop-tablet-and-mobile)

**SET THE VIEW PORT :**

Pages optimized for a variety of devices must include a meta viewport element in the head of the document. A meta viewport tag gives the browser instructions on how to control the page's dimensions and scaling.

* Use the meta viewport tag to control the width and scaling of the browser's viewport.
* Include width=device-width to match the screen's width in device-independent pixels.
* Include initial-scale=1 to establish a 1:1 relationship between CSS pixels and device-independent pixels.

Mobile browsers will render the page at a desktop screen width (usually about 980px, though this varies across devices), and then try to make the content look better by increasing font sizes and scaling the content to fit the screen.

<meta name="viewport" content="width=device-width, initial-scale=1" />

Using the meta viewport value width=device-width instructs the page to match the screen’s width in device-independent pixels. This allows the page to reflow content to match different screen sizes, whether rendered on a small mobile phone or a large desktop monitor.

# MEDIA QUERIES FOR COMMON DEVICE BREAKPOINTS :

The @media query is 1/3 of the recipe for responsive design. It is the**key** ingredient that, in it's simplest form, allows specified CSS to be applied depending on the device and whether it matches the media query criteria.

Before you start copying and pasting below, read [**why you don't need device specific viewports**](https://responsivedesign.is/articles/why-you-dont-need-device-specific-breakpoints).

/\* Smartphones (portrait and landscape) ----------- \*/

@media only screen and (min-device-width : 320px) and (max-device-width : 480px)

{ /\* Styles \*/ }

/\* Smartphones (landscape) ----------- \*/

@media only screen and (min-width : 321px)

{ /\* Styles \*/ }

/\* Smartphones (portrait) ----------- \*/

@media only screen and (max-width : 320px)

{ /\* Styles \*/ }

/\* iPads (portrait and landscape) ----------- \*/

@media only screen and (min-device-width : 768px) and (max-device-width : 1024px)

{ /\* Styles \*/ }

/\* iPads (landscape) ----------- \*/

@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : landscape)

{

/\* Styles \*/

}

/\* iPads (portrait) ----------- \*/

@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : portrait)

{

/\* Styles \*/

}

/\*\*\*\*\*\*\*\*\*\*

iPad 3

\*\*\*\*\*\*\*\*\*\*/

@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : landscape) and (-webkit-min-device-pixel-ratio : 2)

{

/\* Styles \*/

}

@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : portrait) and (-webkit-min-device-pixel-ratio : 2)

{

/\* Styles \*/

}

/\* Desktops and laptops ----------- \*/

@media only screen and (min-width : 1224px)

{

/\* Styles \*/

}

/\* Large screens ----------- \*/

@media only screen and (min-width : 1824px)

{

/\* Styles \*/

}

/\* iPhone 4 ----------- \*/

@media only screen and (min-device-width : 320px) and (max-device-width : 480px) and (orientation : landscape) and (-webkit-min-device-pixel-ratio : 2)

{

/\* Styles \*/

}

@media only screen and (min-device-width : 320px) and (max-device-width : 480px) and (orientation : portrait) and (-webkit-min-device-pixel-ratio : 2)

{

/\* Styles \*/

}

/\* iPhone 5 ----------- \*/

@media only screen and (min-device-width: 320px) and (max-device-height: 568px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 2){

/\* Styles \*/

}

@media only screen and (min-device-width: 320px) and (max-device-height: 568px) and (orientation : portrait) and (-webkit-device-pixel-ratio: 2)

{

/\* Styles \*/

}

/\* iPhone 6 ----------- \*/

@media only screen and (min-device-width: 375px) and (max-device-height: 667px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 2)

{

/\* Styles \*/

}

@media only screen and (min-device-width: 375px) and (max-device-height: 667px) and (orientation : portrait) and (-webkit-device-pixel-ratio: 2)

{

/\* Styles \*/

}

/\* iPhone 6+ ----------- \*/

@media only screen and (min-device-width: 414px) and (max-device-height: 736px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 2)

{

/\* Styles \*/

}

@media only screen and (min-device-width: 414px) and (max-device-height: 736px) and (orientation : portrait) and (-webkit-device-pixel-ratio: 2)

{

/\* Styles \*/

}

/\* Samsung Galaxy S3 ----------- \*/

@media only screen and (min-device-width: 320px) and (max-device-height: 640px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 2)

{

/\* Styles \*/

}

@media only screen and (min-device-width: 320px) and (max-device-height: 640px) and (orientation : portrait) and (-webkit-device-pixel-ratio: 2)

{

/\* Styles \*/

}

/\* Samsung Galaxy S4 ----------- \*/

@media only screen and (min-device-width: 320px) and (max-device-height: 640px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 3)

{

/\* Styles \*/

}

@media only screen and (min-device-width: 320px) and (max-device-height: 640px) and (orientation : portrait) and (-webkit-device-pixel-ratio: 3)

{

/\* Styles \*/

}

/\* Samsung Galaxy S5 ----------- \*/

@media only screen and (min-device-width: 360px) and (max-device-height: 640px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 3)

{

/\* Styles \*/

}

@media only screen and (min-device-width: 360px) and (max-device-height: 640px) and (orientation : portrait) and (-webkit-device-pixel-ratio: 3)

{

/\* Styles \*/

}

**INDEX**

**BASIC BOOTSTRAP ( Use bootstrap.css plugins only )**

1. Bootstrap Introduction
2. Bootstrap Grid System
3. Bootstrap Typography
4. Bootstrap Tables
5. Bootstrap Lists
6. Bootstrap List Groups
7. Bootstrap Forms
8. Bootstrap Input Groups
9. Bootstrap Buttons
10. Bootstrap Button Groups
11. Bootstrap Images
12. Bootstrap Media Objects
13. Bootstrap glyphicons
14. Bootstrap Navs
15. Bootstrap Navbar
16. Bootstrap Panels
17. Bootstrap Breadcrumbs
18. Bootstrap Pagination
19. Bootstrap Labels & Badges
20. Bootstrap Progress Bars
21. Bootstrap Jumbotron
22. Bootstrap Wells
23. Bootstrap Helper Classes

**ADVANCED BOOTSTRAP ( bootstarp.css, bootstrap.js jquery.js )**

1. Bootstrap Modals
2. Bootstrap Dropdowns
3. Bootstrap Tabs
4. Bootstrap Tooltips
5. Bootstrap Popovers
6. Bootstrap Alerts
7. Bootstrap Stateful Buttons
8. Bootstrap Accordion
9. Bootstrap Carousel
10. Bootstrap Typeahead
11. Bootstrap ScrollSpy
12. Bootstrap Affix
13. Resposnive Website

## 1. BOOTSTRAP INTRODUCTION

## What is Twitter Bootstrap

* Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript.
* **Bootstrap** is a free and open-source collection of tools for creating websites and web applications.
* It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

## History of Twitter Bootstrap

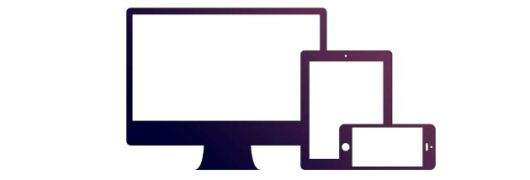
* Bootstrap was developed by *Mark Otto* and *Jacob Thornton* at *Twitter*.
* It was released as an open source product in August 2011 on GitHub.

## Why Use Twitter Bootstrap

* **Mobile first approach** − Bootstrap 3, framework consists of Mobile first styles throughout the entire library instead them of in separate files.
* **Browser Support** − It is supported by all popular browsers.



* **Easy to get started** − With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
* **Responsive design** − Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. More about the responsive design is in the chapter [Bootstrap Responsive Design.](http://www.tutorialspoint.com/bootstrap/bootstrap_responsive_utilities.htm)



* Provides a clean and uniform solution for building an interface for developers.
* It contains beautiful and functional built-in components which are easy to customize.
* It also provides web based customization.
* And best of all it is an open source.

## 11 Reasons to Use Twitter Bootstrap

The Bootstrap framework is the latest innovation to hit the design and development arena, making the creation of websites and apps easier, faster and better in general. If you’re not using Twitter Bootstrap yet, it’s time you took a look. Bootstrap is like a candy store for developers full of a multitude of tools.

## 1. Save Time

The Bootstrap libraries offer readymade pieces of code that can pump life into a website. A web developer doesn’t have to spend time laboriously working out and writing code, but simply find the right piece and fit it into the structure they’re working on

## 2. Customizable

A great aspect of Bootstrap is that you can make it your own This is the main reason many developers feel some tools are completely useless on the Bootstrap while others revere the same.

## 3. Factor in the Design

### The Grid Speaks

Creating page layouts requires a decent grid. Using the platform’s grid isn’t mandatory, but it does make your job a whole lot easier. On default mode the platform provides a 16 column grid which is 940px wide. A single column holds 40px with an additional 20px acting as the gutter. Rows and spans can be manipulated as needed. Styling is completed in auto mode and a developer can get the job done just by dropping content into the HTML they’re working with. Additionally, if you need to work through columns and nesting, then you’ve got the right grid to get the job done.

### LESS is more

LESS has in reality gone ballistic within development circles. With its additions to the Bootstrap platform, you can use LESS mixins and CSS manipulations to customize the built-in grid.

Bootstrap takes some of the most popular CSS3 utilities and brings them to life as a homogenous reaction on all browsers.

### JavaScript

Bootstrap comes equipped with JavaScript libraries that go above and beyond basic structural and styling. JavaScript often becomes an integral part of a web design and developers have to pull in all elements together to see the final form flourish.

With Bootstrap, a developer can easily manipulate modal windows alerts, tooltips, Scrollspy, Popover, Button, Typehead, etc. The best part, however, is that Bootstrap enables you to skip writing the script altogether.

## 4. Consistency

Since it works with a central set of development code, Bootstrap’s results are uniform across platforms. You’ll see the same thing on Internet Explorer, Chrome and Firefox.

## 5. Updates

 updates get to users twice a year. Bootstrap, on the other hand, is ever-evolving with updates coming out on a much more regular and constant basis. As soon as web developers find a problem, the Bootstrap team starts looking to fix it.

## 6. Integration

If you’re working on a site which is already live, but still trying to iron out the creases then Bootstrap can help.  Integration is simple, fast and easy to accomplish and once you’re done you can play with your design to your heart’s content.

## 7. Responsiveness

Bootstrap is responsive. If you shift from a laptop to an iPad and from an iPad to a Mac you won’t have to fret over your work. Bootstrap adapts to the change in platforms with super speed and efficiency.

## 8. Future Compatibility

Bootstrap comes equipped with many elements that are being considered the future of design itself. For instance both HTML5 and CSS3 are things that are going to be big in the future. Since the framework takes into account the future of design and development.

## 9. Competitiveness

Bootstrap isn’t the only one of its kind. It has been pitted against JQuery UI and the [HTML5 Boilerplate](http://www.sitepoint.com/introduction-html5-boilerplate/), but a more realistic competitor is the Zurb Foundation. Bootstrap 2 has a toolset that the Foundation will have a tough time matching up to.  Third party plugins, themes, a plethora of features, codes, etc. are already up for grabs with Bootstrap – which isn’t true for the Foundation at this point. Some of you might feel that Bootstrap has a bloated feel (which the Foundation doesn’t) but that can be easily done away with through the custom Bootstrap download builder.

## 10. The Docs Talk!

Bootstrap’s documentation is impressive. The case with most newer platforms is that they don’t have proper documentation, but Bootstrap gives you a head start because you don’t have to play with everything to learn and instead can find information on just about anything through the documentation.

## 11. For the Bosses

You don’t need to spend a lot of time learning to use Bootstrap as it has taken all modular approaches and good architecture and pulled them together. From the perspective of an employer, if you give a web developer the Bootstrap framework then they won’t need to spend as much time learning, as compared to HTML and CSS. Bootstrap allows for immediate quality work, which makes it great for facilitating efficient, outcome-oriented design communication.

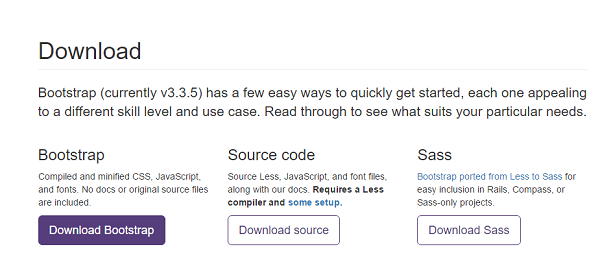
## Download Bootstrap :

You can download the latest version of Bootstrap from [**http://getbootstrap.com/**](http://getbootstrap.com/#_blank)**.** When you click on this link, you will get to see a screen as below:

* **Download Bootstrap** − Clicking this, you can download the precompiled and minified versions of Bootstrap CSS, JavaScript, and fonts. No documentation or original source code files are included.
* **Download Source** − Clicking this, you can get the latest Bootstrap LESS and JavaScript source code directly from GitHub.
* **Download Sass** − Clicking this, you can get the latest Bootstrap LESS to Sass for easy inclusion in Rails, Compass, or Sass-only projects.



If you click on DownloadBootstrap button, The page will redirect to another page. Here you can see Three buttons:



## File Structure of Download Bootstrap

Once the compiled version Bootstrap is downloaded, extract the ZIP file, and you will see the following file/directory structure :



As you can see, there are compiled CSS and JS (bootstrap.\*), as well as compiled and minified CSS and JS (bootstrap.min.\*). Fonts from Glyphicons are included, as it is the optional Bootstrap theme.

## Bootstrap Source Code

If you have downloaded the Bootstrap source code then the file structure would be as follows −



* The files under *less/*, *js/*, and *fonts/* are the source code for Bootstrap CSS, JS, and icon fonts (respectively).
* The *dist/* folder includes everything listed in the precompiled download section above.
* *docs-assets/*, *examples/*, and all *\*.html* files are Bootstrap documentation.

## Bootstrap CDN:

**<!-- Latest compiled and minified CSS -->**

**<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />**

**<!-- Optional theme -->**

**<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-theme.min.css" />**

**<!-- Latest compiled and minified JavaScript -->**

**<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js">**

**</script>**

## Bootstrap Plugins linked to webpage

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Bootstrap 101 Template</title>

<!-- Bootstrap -->

<link href="css/bootstrap.min.css" rel="stylesheet">

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

<!-- Include all compiled plugins (below), or include individual files as needed -->

<script src="js/bootstrap.min.js"></script>

</head>

<body>

<h1>Hello, world!</h1>

</body>

</html>

## Bootstrap CDN Plugins linked to Webpage

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Bootstrap 101 Template</title>

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

**<!-- Latest compiled and minified CSS -->**

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

<!-- Latest compiled and minified JavaScript -->

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

</head>

<body>

<h1>Hello, world!</h1>

</body>

</html>

## 2. BOOTSTRAP GRID SYSTEM

## What is a Grid?

A grid is a structure (usually two-dimensional) made up of a series of intersecting straight (vertical, horizontal) lines used to structure the content. It is widely used to design layout and content structure in print design. In web design, it is a very effective method to create a consistent layout rapidly and effectively using HTML and CSS.

Grids in web design organise and structure content, makes the websites easy to scan and reduces the cognitive load on users.

## What is Bootstrap Grid System?

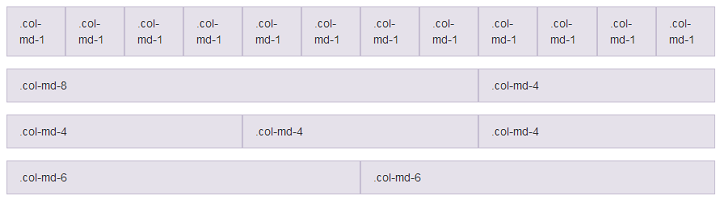
Bootstrap includes a responsive, mobile first fluid grid system that appropriately scales up to 12 columns as the device or viewport size increases. It includes predefined classes for easy layout options, as well as powerful mixins for generating more semantic layouts.

Bootstrap 3 is mobile first in the sense that the code for Bootstrap now starts by targeting smaller screens like mobile devices, tablets, and then “expands” components and grids for larger screens such as laptops, desktops.

## Bootstrap Grid System :

Bootstrap's grid system allows up to 12 columns across the page.

If you do not want to use all 12 column individually, you can group the columns together to create wider columns:



## Grid Classes :

The Bootstrap grid system has four classes:

* col-xs-number (for phones)
* col-sm-number (for tablets)
* col-md-number (for Laptops)
* col-lg-number (for larger desktops)

The classes above can be combined to create more dynamic and flexible layouts.

**Tip:** Each class scales up, so if you wish to set the same widths for xs and sm, you only need to specify xs.

## Grid System Rules:

Some Bootstrap grid system rules:

* Rows must be placed within a .container (fixed-width) or .container-fluid (full-width) for proper alignment and padding.
* Use rows to create horizontal groups of columns.
* Content should be placed within columns, and only columns may be immediate children of rows.
* Predefined classes like .row and .col-sm-4 are available for quickly making grid layouts.
* Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on .rows.
* Grid columns are created by specifying the number of 12 available columns you wish to span. For example, three equal columns would use three .col-sm-4.

## Working of Bootstrap Grid System :

Grid systems are used for creating page layouts through a series of rows and columns that house your content. Here's how the Bootstrap grid system works :

* Rows must be placed within a **.container** class for proper alignment and padding.
* Use rows to create horizontal groups of columns.
* Content should be placed within the columns, and only columns may be the immediate children of rows.
* Predefined grid classes like **.row and** **.col-xs-4** are available for quickly making grid layouts. LESS mixins can also be used for more semantic layouts.
* Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and the last column via negative margin on **.rows**.
* Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three **.col-xs-4**.

## Media Queries :

Media query is a really fancy term for "conditional CSS rule". It simply applies some CSS, based on certain conditions set forth. If those conditions are met, the style is applied.

Media Queries in Bootstrap allow you to move, show and hide content based on the viewport size. Following media queries are used in LESS files to create the key breakpoints in the Bootstrap grid system.

/\* Extra small devices (phones, less than 768px) \*/

/\* No media query since this is the default in Bootstrap \*/

/\* Small devices (tablets, 768px and up) \*/

@media (min-width: @screen-sm-min) { ... }

/\* Medium devices (desktops, 992px and up) \*/

@media (min-width: @screen-md-min) { ... }

/\* Large devices (large desktops, 1200px and up) \*/

@media (min-width: @screen-lg-min) { ... }

@media (max-width: @screen-xs-max) { ... }

@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }

@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }

@media (min-width: @screen-lg-min) { ... }

## 3. BOOTSTRAP TYPOGRAPHY

You will learn about the styling and formatting of text content like headings, paragraphs, blockquotes, etc. with Bootstrap.

## Working with Headings

You can also utilize the heading classes **.h1**through**.h6** on other elements, if you want to apply the style on element's text same as headings.

Moreover you can use the [<small>](http://www.tutorialrepublic.com/html-reference/html-small-tag.php) tag or [<span>](http://www.tutorialrepublic.com/html-reference/html-span-tag.php) tag with **.small**class to display the secondary text of any heading in a smaller and lighter variation.

## Creating Page Headers

You can make your <h1> heading appear differently than rest of the headings on a page using the **page-header** class.

## Working with Paragraphs

You can also make a paragraph stand out by just adding the class **.lead.** You can easily align text inside a paragraph and other elements using text alignment classes.

text-left, text-center, text-right, text-justify, text-nowrap.

## Bootstrap Text Formatting

You are free to use text formatting tags like [<strong>](http://www.tutorialrepublic.com/html-reference/html-strong-tag.php), [<i>](http://www.tutorialrepublic.com/html-reference/html-i-tag.php), [<small>](http://www.tutorialrepublic.com/html-reference/html-small-tag.php) to make your text bold, italic, small and so on, in the same way you do in simple HTML.

## Text Transformation Classes

You can also transform the text to lowercase, uppercase or make them capitalize.

text-lowercase, text-uppercase, text-capitalize.

## Text Emphasis Classes

Colors are the powerful method of conveying important information in website design. Bootstrap has handful of emphasis utility classes that can be used for this purpose such as showing success message, warning or error message in different color Styles i.e

.**text-muted** in Light color, .**text-primary** in **Thick Blue** color, .**text-success** in **Green** color

.**text-info** in **Lightblue** color, .**text-warning** in **purple** color, .**text-danger** in **red** color

## 4.BOOTSTRAP TABLES

* You will learn how to create elegant tables with Bootstrap.
* The HTML tables are used to present data in grid manner like row and columns. Using Bootstrap you can greatly improve the appearance of table in a simple way

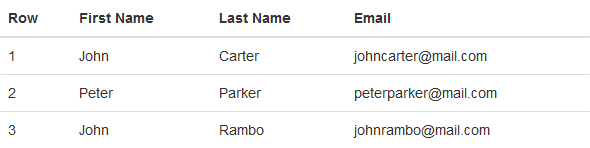
## Supported Table Elements in Bootstrap :

The following table lists the supported HTML table elements and how they should be used.

| **Tag** | **Description** |
| --- | --- |
| [<table>](http://www.tutorialrepublic.com/html-reference/html-table-tag.php) | Wrapper element for displaying data in a tabular format. |
| [<caption>](http://www.tutorialrepublic.com/html-reference/html-caption-tag.php) | The title or summary of what the table holds. |
| [<thead>](http://www.tutorialrepublic.com/html-reference/html-thead-tag.php) | Container element for table header rows (<tr>) defines headings for table columns. |
| [<tbody>](http://www.tutorialrepublic.com/html-reference/html-tbody-tag.php) | Container element for table rows (<tr>) that defines the body of a table. |
| [<tr>](http://www.tutorialrepublic.com/html-reference/html-tr-tag.php) | Container element for a set of table cells (<td> or <th>) that appears in a single row. |
| [<th>](http://www.tutorialrepublic.com/html-reference/html-th-tag.php) | Special table cell for column headings. |
| [<td>](http://www.tutorialrepublic.com/html-reference/html-td-tag.php) | Default table cell for placing data. |

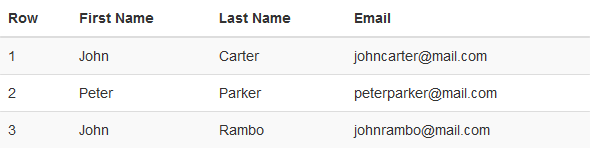
## Creating a Simple Table with Bootstrap :

## You can create tables with basic styling that has horizontal dividers and small cell padding (8px by default), by just adding the Bootstrap's class .table to the <table> element.



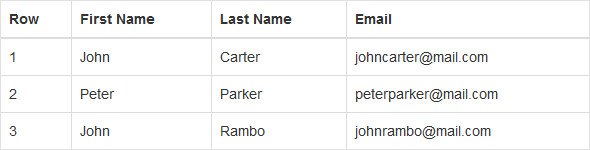
## Tables with Alternate Background :

You can create table with alternate background like zebra-strips by simply adding the Bootstrap's class **.table-striped** to the **.table** base class.



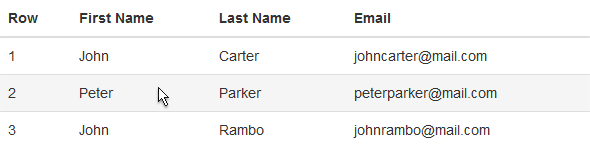
## Table with Borders :

You can also add borders to the all table cells by adding an extra Bootstrap's class **.table-bordered**  to the **.table** base class.



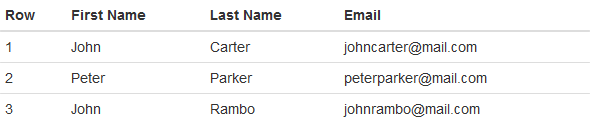
## Enable Hover State on Table Rows

You can also enable a hover state on table rows within a [<tbody>](http://www.tutorialrepublic.com/html-reference/html-tbody-tag.php) element by adding the Bootstrap's class .**table-hover** to the **.table** base class.



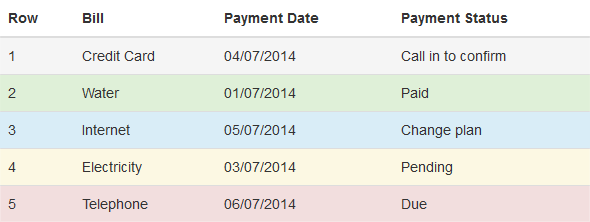
## Condensed or Compact Table

You can also make your tables more compact and save the space through adding an extra class **.table-condensed** to the **.table** base class. The class .table-condensed makes the table compact by cutting the cell padding in half.



## Optional Emphasis Classes for Table Rows

There are some contextual classes to emphasize the row or individual cells data like **.active, .success, .warning, .danger, .info classes.** through coloring its background.



## Creating Responsive Tables with Bootstrap

To make any table responsive just place the table inside a [<div>](http://www.tutorialrepublic.com/html-reference/html-div-tag.php) element and apply the class **.table-responsive** on it.

With Bootstrap 3 you can also create responsive tables to enable horizontal scrolling on small devices (screen width under 768px). However, viewing responsive tables on other devices having screen width larger than 768px, you will not see any difference.

## 5. BOOTSTRAP LISTS

## Unstyled Ordered and Unordered Lists

Sometimes you might need to remove the default styling form the list items. You can do this by simply applying the class **.list-unstyled** to the respective <ul> or <ol> elements. **Example is:**



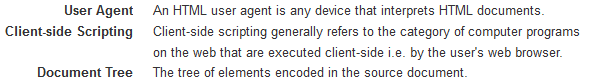
## Placing Ordered and Unordered List Items Inline

If you want to create a horizontal menu using ordered or unordered list you need to place all list items in a single line i.e. side by side. You can do this by simply applying the Bootstrap's class **.list-inline** to the respective <ul> or <ol> elements. The **.list-inline** class also adds some left and right padding (5px by default) to the all list items. **Example is:**



## Creating Horizontal Definition Lists

The terms and descriptions in definition lists can also be placed side-by-side using the Bootstrap's class .**dl-horizontal.** The terms in horizontal definition lists will be truncated if is too long to fit in the left column (160px by default), however in narrower viewports they will change to the default stacked layout. **Example is:**



## 6. BOOTSTRAP LIST GROUPS

## Creating List Groups with Bootstrap

The list groups are very useful and flexible component for displaying lists of elements in a beautiful manner. In most basic form a list group is simply an [unordered list](http://www.tutorialrepublic.com/html-tutorial/html-lists.php) with the class **.list-group** whereas, the list items having the class **.list-group-item. Example is:**

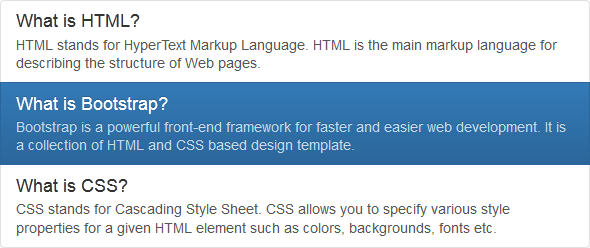


## List Group with Linked Items

You can also [hyperlink](http://www.tutorialrepublic.com/html-tutorial/html-links.php) list group items with the little change in HTML markup. Just replace the [<li>](http://www.tutorialrepublic.com/html-reference/html-li-tag.php) with [<a>](http://www.tutorialrepublic.com/html-reference/html-a-tag.php) tag and use [<div>](http://www.tutorialrepublic.com/html-reference/html-div-tag.php) element as a parent instead of [<ul>](http://www.tutorialrepublic.com/html-reference/html-ul-tag.php). You can also add [icons](http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-icons.php) and [badges](http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-labels-and-badges.php) to this list group to make it more elegant. The badges component will automatically be positioned on the right. **Example is:**

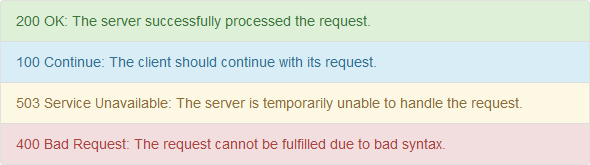


**You can also add other HTML elements like headings and paragraph within these list groups.**



## List Group with Contextual States

Tip: You can use the Bootstrap list group component for creating the sidebar navigation menu, e.g. displaying the list of products or categories on your website.



## 7. BOOTSTRAP FORMS

## Creating Forms with Bootstrap :

HTML forms are the integral part of the web pages and applications, but styling the form controls manually one by one with CSS are often boring and tedious.

Bootstrap greatly simplifies the process of styling and alignment of form controls like labels, input fields, selectboxes, textareas, buttons, etc. through predefined set of classes. Bootstrap provides three different types of form layouts:

* **Vertical Form (default form layout)**
* **Horizontal Form**
* **Inline Form**

## Creating Vertical Form Layout :

This is the default Bootstrap form layout in which styles are applied to form controls without adding any base class to the [<form>](http://www.tutorialrepublic.com/html-tutorial/html-forms.php) element or any large changes in the markup.

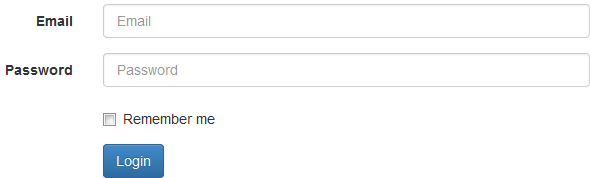
**Note:** In Bootstrap 3 all textual elements like [<input>](http://www.tutorialrepublic.com/html-reference/html-input-tag.php), [<textarea>](http://www.tutorialrepublic.com/html-reference/html-textarea-tag.php), and [<select>](http://www.tutorialrepublic.com/html-reference/html-select-tag.php)with the class **.form-control** are 100% wide by default. To use them inline, you'll have to set a width on the element the form controls used within.

****

## Creating Horizontal Form Layout

In horizontal form layout labels are right aligned and floated to left to make them appear on the same line as form controls. The horizontal form layout requires the various markup changes from a default form layout.

* **Add the class .form-horizontal to the <form> element.**
* **Wrap labels and form controls in a <div> element and apply the class .form-group.**
* **Use Bootstrap's predefined grid classes to align labels and form controls.**
* **Add the class .control-label to the <label> element.**

****

## Creating Inline Form Layout.

Sometimes you might require to place the form controls side-by-side to compact the layout. You can do this easily by adding the Bootstrap class .**form-inline** to the [<form>](http://www.tutorialrepublic.com/html-tutorial/html-forms.php) element.

It is recommended to include a label for every form inputs otherwise screen readers will have trouble with your forms. However in case of inline form layout you can hide these labels using the .sr-only class.



## Static Form Control :

There might be a situation when you need to place just plain text next to a form label instead of a form control. You can do this within a horizontal form layout by using the .**form-control-static** class.

## Height Sizing of Inputs and Select Boxes :

You can easily control the height of your input and select boxes to match the [button sizes](http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-buttons.php). The Bootstrap's relative sizing classes like **.input-lg, .input-sm**can be used both on[<input>](http://www.tutorialrepublic.com/html-reference/html-input-tag.php) and [<select>](http://www.tutorialrepublic.com/html-reference/html-select-tag.php) boxes to create it's larger or smaller sizes.

## Column Sizing of Inputs, Textareas and Select Boxes :

You can also match the sizes of your form controls to the Bootstrap grid column sizes. Just wrap your form controls (i.e. [<input>](http://www.tutorialrepublic.com/html-reference/html-input-tag.php), [<textarea>](http://www.tutorialrepublic.com/html-reference/html-textarea-tag.php), and [<select>](http://www.tutorialrepublic.com/html-reference/html-select-tag.php)) in grid columns, or any custom element and apply the [grid classes](http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php) on it.

## Height Sizing of Form Groups :

You can add the relative form sizing classes such as **.form-group-lg** or **.form-group-sm** to the **.form-group** itself to make both the labels and form controls larger or smaller at the same time within the horizontal form layouts.

## Placing Help Text Around Form Controls :

Placing help text for the form controls in an efficient way to guide users to enter the correct data in a form. You can place block level help text for the form controls using the class **.help-block**. The help text is typically displayed at the bottom of the control.

## Bootstrap Form Validation States :

* Bootstrap provides easy to use and powerful mechanism for styling input controls to present different validation states.
* Bootstrap includes validation styles for error, warning, and success messages. To use, just add the appropriate class to the surrounding **.form-group.**
* You can also add optional feedback icons to your inputs using the class .has-feedback on.form-group and the right glyphicon

## 8. BOOTSTRAP INPUT GROUPS

Bootstrap input group component is very flexible and powerful component for creating the interactive form controls, however it is limited to textual input only.

In the following sections you'll see how to extend the text based [<input>](http://www.tutorialrepublic.com/html-reference/html-input-tag.php) by adding the text or buttons before, after, or on both sides of it to make your form more attractive.

## Creating Prepended and Appended Inputs

You can add text and icons or buttons before or after any text-based input. To prepend or append text and icons to an input:

* Wrap the text or icon within a <span> element having the class .**input-group-addon** and place it before or after the <input> element.
* Wrap both the <span> and text-based <input> element within a <div> element and apply the class .**input-group** on it.
* **Note:** Bootstrap's prepending or appending feature is only available to text-based inputs. It does not support <select> or <textarea> elements.



## Checkboxes and Radio Buttons Addons

Similarly, you can place checkbox or radio button within input group's addon instead of text.



## Buttons Addons for Text Inputs

You can also prepend or append buttons instead of text. Wrap buttons within the <span>element and apply the class **.input-group-btn**, instead of .**input-group-addon**.



## Adding Button Dropdowns to Text Inputs

You can also create dropdown buttons if you want to do more than one action from a button



## Adding Segmented Dropdown Button Groups

Similarly, you can define the segmented dropdown button group where dropdown button is placed besides the other buttons.



## Height Sizing of Input Groups

You can also add the relative form sizing classes such as .input-group-lg or .input-group-sm to the .input-group itself to make it larger or smaller.

The contents within .input-group will automatically resize — there is no need for repeating the form control size classes on each element.

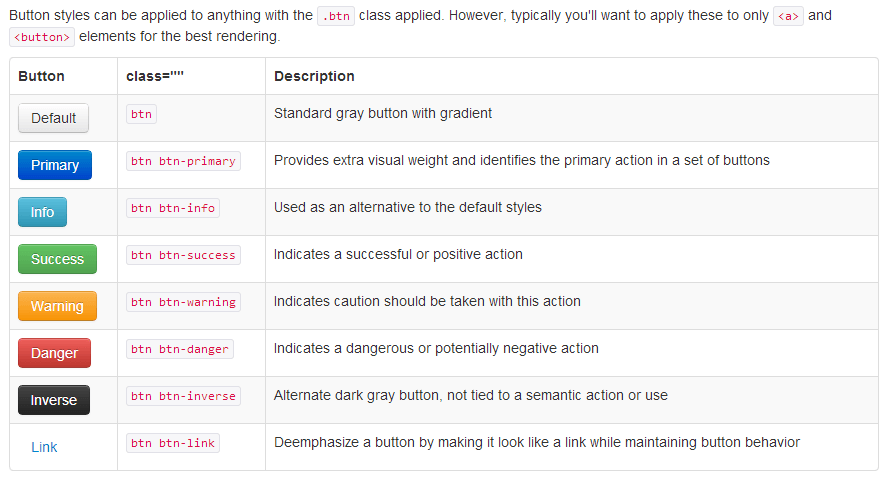
## 9. BOOTSTRAP BUTTONS

Buttons are the integral part of a website and application. They are used for various purposes like, submit or reset an [HTML form](http://www.tutorialrepublic.com/html-tutorial/html-forms.php), performing interactive actions such as showing or hiding something on a web page on click of the button, etc. The Bootstrap button CSS provides the quick and easy way to create and customize the buttons.

## Bootstrap Button Styles

Different classes are available in Bootstrap for styling the buttons as well as to indicate the different states. Button styles can be applied to any element. However, it is applied normally to the [<a>](http://www.tutorialrepublic.com/html-reference/html-a-tag.php), [<input>](http://www.tutorialrepublic.com/html-reference/html-input-tag.php), and [<button>](http://www.tutorialrepublic.com/html-reference/html-button-tag.php) elements for the best rendering.

The following table lists the different buttons which are available in the Bootstrap:



## Changing the Sizes of Buttons

Bootstrap gives you option further to scaling a button up or down. You can a make buttons larger or smaller through adding an extra class .**btn-lg,  .btn-sm, or  .btn-xs**.

## BOOTSTRAP BUTTON GROUPS

## Creating Button Group with Bootstrap

Bootstrap however, also allows you to group a series of buttons together in a single line through the button group component. To create a button groups just wrap a series of buttons in a [<div>](http://www.tutorialrepublic.com/html-reference/html-div-tag.php) element and apply the class.**btn-group** on it.



You can also make the button groups appear vertically stacked rather than horizontally. To do this just replace the class .btn-group with the .**btn-group-vertical**.

## Creating Button Toolbar :

You can also combine sets of button groups together for creating more complex components like button toolbar. To create button toolbar just wrap sets of button groups in a [<div>](http://www.tutorialrepublic.com/html-reference/html-div-tag.php) element and apply the class .**btn-toolbar** on it.



## Height Sizing of Button Groups

1. You can also apply relative sizing classes like .btn-group-lg, .btn-group-sm or .btn-group-xs on
2. button groups to create larger or smaller button groups. Just add these button sizing classes
3. directly to the .btn-group, instead of applying to every button.

## Creating Justified Button Groups

You can also make a group of buttons stretch at the same size to span the entire width of its parent by appling an extra class .btn-group-justified to the .btn-group base class.

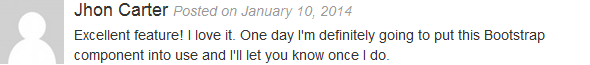


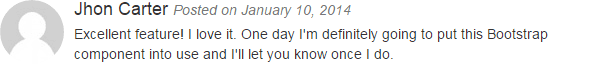
## 11. BOOTSTRAP MEDIA OBJECTS

## Using the Bootstrap Media Object

If you want to create a layout that contains left- or right-aligned media object like images or video alongside the textual content such as blog comments, Tweets, etc.

You can do this easily using the newly introduced Bootstrap media component using base class **.media class to <div>-tag. And also child classes are .media-left and .media-right which is declared to <div>-tag and .media-object Declared to <img /> and .media-body declared to </div> and .media-heading declared to <h1>--<h6>**

****



## Alignment of Media Objects

You can also align the images or other media at the middle or bottom of the content block using the class .**media-middle** or .**media-bottom**. By default it is top aligned.

## Creating Nested Media List

You can also create a list of media object or nested media object using the media list component. It can be useful for comment threads or article lists using **.media-list.**

## 12. BOOTSTRAP IMAGES

## Styling Images with Bootstrap

1. Images are very common in modern web design. So styling images and placing it properly on the
2. web pages is very important for improving the user experience.
3. Using the Bootstrap built-in classes you can easily style images such as making the round
4. cornered or circular images, or give them effect like thumbnails using **.img-rounded** and **.img-**
5. **circle** and **.img-thumbnail** in <img />.



## Creating Responsive Images and Videos :

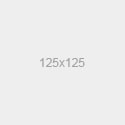
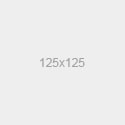
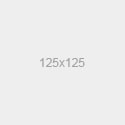
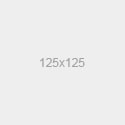
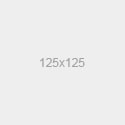
In Bootstrap you can make the images responsive too. Just add the class **.img-responsive** to the [<img>](http://www.tutorialrepublic.com/html-reference/html-img-tag.php) tag.

You can also make the [video](http://www.tutorialrepublic.com/html-tutorial/html5-video.php) or slideshow embedded in a web page responsive without affecting its original aspect ratio. The Bootstrap responsive embed classes can be applied directly to the [<iframe>](http://www.tutorialrepublic.com/html-reference/html-iframe-tag.php), [<embed>](http://www.tutorialrepublic.com/html-reference/html5-embed-tag.php), [<video>](http://www.tutorialrepublic.com/html-reference/html5-video-tag.php), and [<object>](http://www.tutorialrepublic.com/html-reference/html-object-tag.php) elements. Using base class **.embed-responsive.**

We've created the two responsive videos with two different aspect ratios (**16:9** and **4:3**) by adding the classes **.embed-responsive-16by9** and **.embed-responsive-4by3** to their containing blocks respectively and the class **.embed-responsive-item** to the descendant <iframe> element.

## Bootstrap Thumbnails

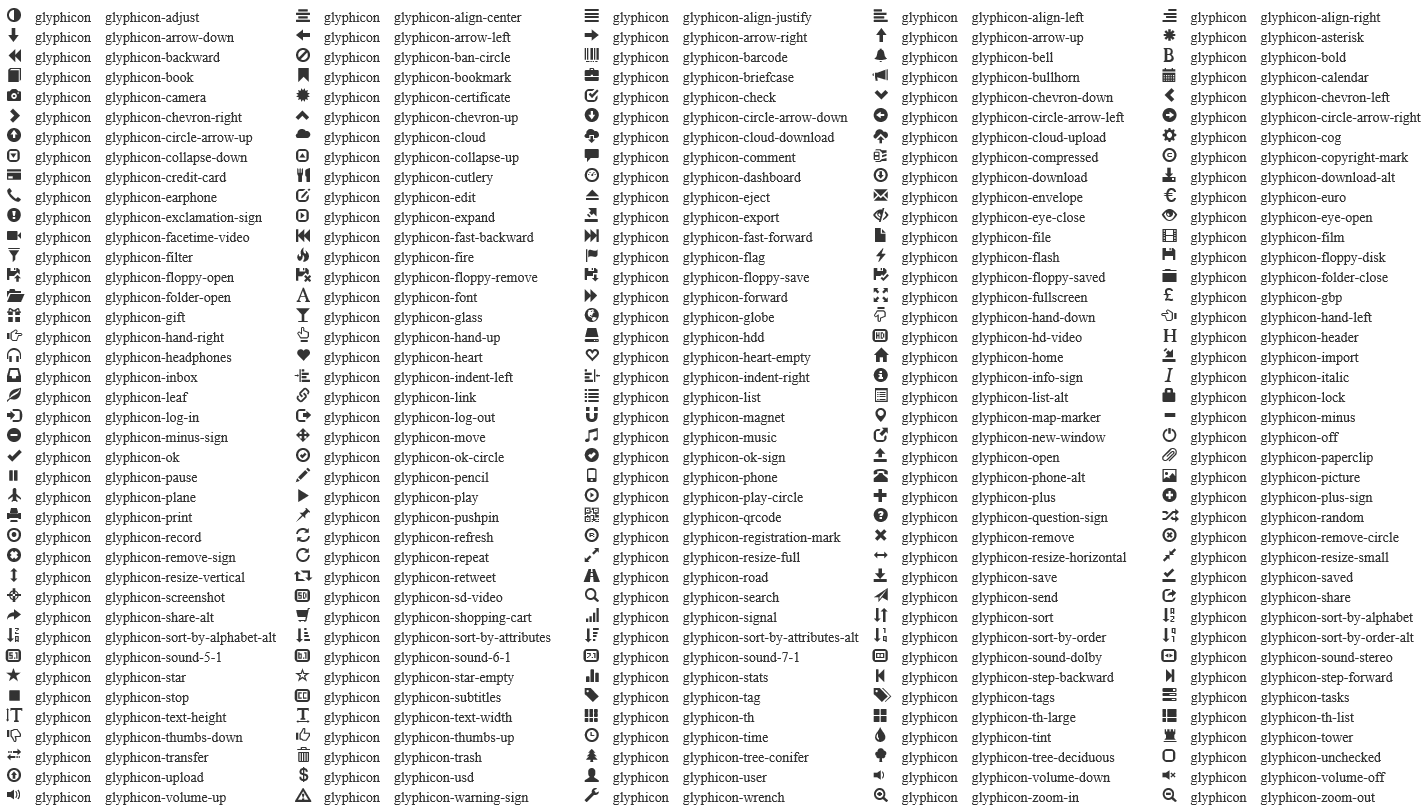
The Bootstrap thumbnail component is very useful for creating grids of images or videos, lists of products, portfolios, and much more.You can also insert HTML content like headings, paragraphs, or buttons into thumbnails using **.thumbnail class to <div> or <a>-tag.**

**13. BOOTSTRAP ICONS**

Bootstrap 3.3.6 includes more than 250 Glyphicons. As opposed to [previous sprite version](http://www.tutorialrepublic.com/twitter-bootstrap-2.3.2-tutorial/bootstrap-icons.php)these icons are available in font format for better usability and scalability.

Since these Glyphicons are font based now you can create icons of any color just applying the CSS [color](http://www.tutorialrepublic.com/css-reference/css-color-property.php) property on the specific element.See <http://getbootstrap.com/components/>.



## 14. BOOTSTRAP NAVS

Bootstrap provides an easy way to create basic nav components like tabs and pills. All the Bootstrap's nav components—tabs and pills—share the same base markup and styles through the base .**nav** class.

## Creating Basic Tabs with Bootstrap

To create a basic tab component using base class **.nav** and sub class **nav-tabs to <ul>-tag.**



## Adding Icons to Tabs

You can also add icons to your tabs to make it more attractive using bootstrap icons classes to <span>-tag in List tag and place before listname.



## Creating Basic Pills Nav with Bootstrap

Similarly you can create a basic pill based navigation using the base class .**nav-pills** instead of .**nav-tabs**, without any further change in markup.



## Adding Icons to Pills Nav

You can also add icons to your pills nav to make it more attractive.



## Stacked Pills Nav

Pills navigations are horizontal by default. To make them appear vertically stacked, just add an extra class **.nav-stacked** to the [<ul>](http://www.tutorialrepublic.com/html-reference/html-ul-tag.php) element. Like as class="nav nav-pills nav-stacked".



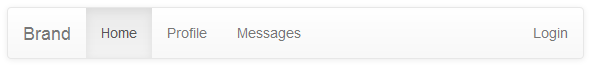
## 15. BOOTSTRAP NAVBAR

## Creating a Simple Navbar with Bootstrap

You can use the Bootstrap navbar component to create responsive navigation header for your website or application.These responsive navbar initially collapsed on devices having small viewports like cell-phones but expand when user click the toggle button.However, it will be horizontal as normal on the medium and large devices like laptop or desktop.

Use the following classes are: class=”navbar navbar-default” to <div> and navbar-header, navbar-toggle, navbar-brand, navbar-collapse, navbar-right.

**Note:** Use the classes .navbar-left or .navbar-right instead of .pull-left or.pull-right to align the nav links, forms, buttons or text inside the navbar.



## Bootstrap Fixed Navbar

Bootstrap also provides mechanism to create navbar that is fixed on the top or bottom of the viewport and will scroll with the content on the page.

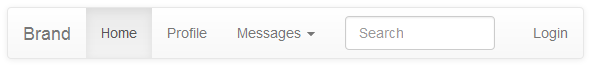
Add an extra class .**navbar-fixed-top** in addition to the **.navbar** and .**navbar-default** base class to create navbars that is fixed on the top.

Similarly to create navbars that is fixed at the bottom add the class **.navbar-fixed-bottom.**

## Bootstrap Static Top Navbar :

You can also create full-width navbar that appears on the top but scrolls away with the page by adding the class .**navbar-static-top.** Unlike the .navbar-fixed-top class, you do not need to change any padding on the [<body>](http://www.tutorialrepublic.com/html-reference/html-body-tag.php) element.

**Bootstrap Navbar with Search Form :** Search form is very common component of the navbars and you have seen it on various websit equite often. Search form can be placed inside the navbar using the class .**navbar-form** on the [<form>](http://www.tutorialrepublic.com/html-reference/html-form-tag.php) element.



You can also create inverted variation of the Bootstrap navbar by adding an extra class **.navbar-inverse**to the .navbar base class, without any further change in markup.

## 16. BOOTSTRAP PANELS

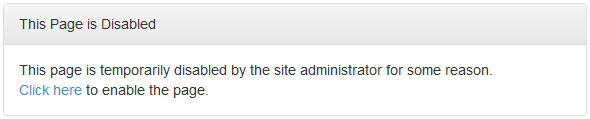
Sometimes you might require to put your content in box for certain reason. In such situation the Bootstrap panel component can be very useful. In most basic form the panel component applies some [border](http://www.tutorialrepublic.com/css-reference/css-border-property.php) and [padding](http://www.tutorialrepublic.com/css-reference/css-padding-property.php) around the content. Use **class="panel panel-default".**

****

## Panels with Heading

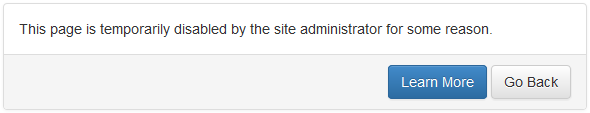
You can also add a heading to your panel with .panel-heading and **.panel-body** class.

You can also include heading elements from [<h1>](http://www.tutorialrepublic.com/html-reference/html-headings-tag.php) to [<h6>](http://www.tutorialrepublic.com/html-reference/html-headings-tag.php) with a .panel-title class.



## Panels with Footer

You can also add footer section to your panels using the .panel-footer class. The panel's footer can be used to wrap buttons or secondary text.



## Panels with Contextual States

you also add contextual state classes like .panel-primary, .panel-success, .panel-info, .panel-warning, or .panel-danger on the panel components to make it more meaningful and drawing attention of the user.

## 17. BOOTSTRAP PAGINATION

## Creating Pagination with Bootstrap

Pagination is the process of organizing content by dividing it into separate pages using pre-defined class is **.pagination. to <ul>-tag.**



## Pagination with Disabled and Active States

Use the class **.disabled** for making the links unclickable and .**active** to indicate the current page to <li>-tag

## Changing the Sizes of Pagination

You can also change the sizes of pagination to increase or decrease the clickable area.

Add the relative sizing classes like .**pagination-lg**, or .**pagination-sm** to the .**pagination** base class for creating larger or smaller pagination.

## Bootstrap Pager

Sometimes you may simply require previous and next links on your website to provide simple and quick navigation to the user instead of the complex pagination. This can be done using the Bootstrap class **.pager to <ul>-tag.**

****

## Alignment of Pager

By default pager are aligned horizontally center but you align previous link to left and next link right using the class **.previous**and **.next** respectively to <li>.



## 18. BOOTSTRAP BREADCUMBS

A breadcrumb is a navigation scheme that indicates current page's location to the user within a website or application.

Breadcrumb navigation can greatly enhance the accessibility of the websites having a large number of pages.

You can create static breadcrumbs layouts with Bootstrap simply using the class .breadcrumb on the [unordered lists](http://www.tutorialrepublic.com/html-tutorial/html-lists.php).



## 19. BOOTSTRAP LABELS & BADGES

## Creating Inline Labels:

Inline labels are generally used to indicate some valuable information on the web pages such as important notes, warning messages, etc using base class is .**label** with sub base class is .**label-default to <span>-tag.** There are some contextual classes to emphasize these inline labels.

**class**="label label-default/label-primary/label-success/label-info/label-warning/label-danger".



## Creating Inline Badges:

Similarly you can create inline badges to provide important notification to the user such as number received or unread messages, number of friend requests etc. They're most commonly found in email client and social networking websites use **class="badge" with <span>-Tag.**



## 20.BOOTSTRAP PROGRESS BARS

## Creating Progress Bar with Bootstrap

Progress bars can be used for showing the progress of a task or action to the users. you to create a simple progress bar with vertical gradient. Use class="progress" and class="progress-bar" classes.



## Creating Stripped Progress Bar

To create the stripped progress bar just add an extra class .progress-striped to the **.progress** base class.



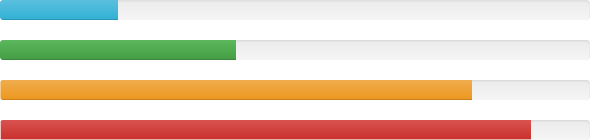
## Creating Stacked Progress Bar

You can also place multiple bars into the same progress bar to stack them.



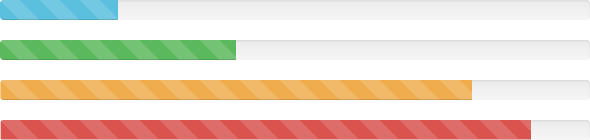
## Progress Bars with Emphasis Classes

Bootstrap also provides some emphasis utility classes for progress bars that can be further used to convey meaning through color.



## Striped Progress Bars with Emphasis Classes

Similar to the solid colors, you can also create varied striped progress bars.



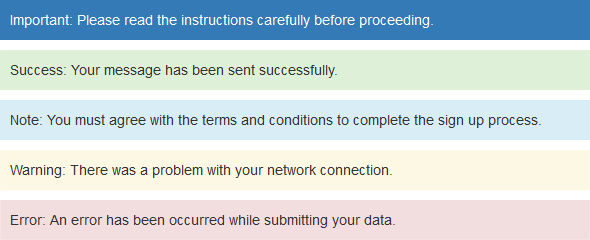
## 21. BOOTSTRAP HELPER CLASSES

## Contextual Colors

You can use the contextual color classes like .text-succes, .text-info, .text-warning etc. to emphasize the text(  [Bootstrap typography](http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-typography.php#contextual-color-classes) ).

## Contextual Backgrounds

you can use the contextual background color classes to set the [background-color](http://www.tutorialrepublic.com/css-tutorial/css-background.php) of an element to apply extra emphasis on them using following classes are: class="bg-primary/bg-success/bg-info/bg-warning/bg-danger".



## Close Icon

Bootstrap provides a generic close icon that can be used for dismissing modals and alerts using class="close"



## Caret Icon

Bootstrap provides a generic caret icon to indicate the dropdown functionality. The direction of the caret icon will reverse automatically in dropup menus.



## Center Alignment of Content Blocks

You can use the Bootstrap class .**center-block** to align the content block horizontally center.

## Quick Floats

You can quickly float an element to the left or right using the .pull-left and .pull-right classes.

## Clearfix

The .clearfix class clears the float on any element. This class is widely used for fixing the [collapsing parent](http://www.tutorialrepublic.com/css-tutorial/css-alignment.php#collapsing-parent) issue, where parent element contains floated boxes.

## Showing and Hiding Content

You can force an element to be shown or hidden on all the devices using the **.show** and **.hidden** classes. you can use the class .invisible to toggle only.

## Content Specific to Screen Readers

The special .**sr-only** class hides an element to all devices except screen readers.

## Hide Text Only:

## You can use the class .text-hide to hide the text content of an element.

## 22. BOOTSTRAP WELLS

## Placing Content inside Wells

The Bootstrap well component provides a quick way to apply a simple inset effect to an element. It will be very helpful if you just want to place some content inside a box to make it look like different from rest of the contents. To use this just enclose the contents with a[<div>](http://www.tutorialrepublic.com/html-reference/html-div-tag.php) element and apply the class .**well** on it.



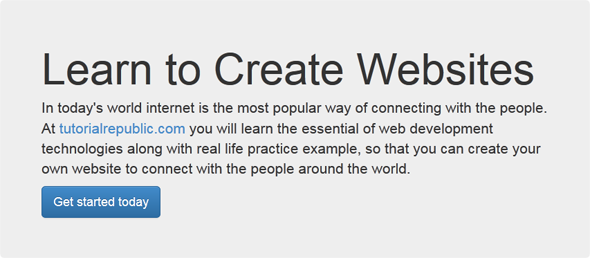
## Varying Well Sizes

You can further control the padding and rounded corners of the wells using the two optional modifier classes **.well-lg and .well-sm to .well** class.

## 23. BOOTSTRAP JUMBOTRON

## Showcasing Contents with Jumbotron

The Bootstrap jumbotron component provides an excellent way to showcase the key content or information on a web page. Just wrap your featured content like heading, descriptions etc. In [<div>](http://www.tutorialrepublic.com/html-reference/html-div-tag.php) element and apply the class .jumbotron on it.

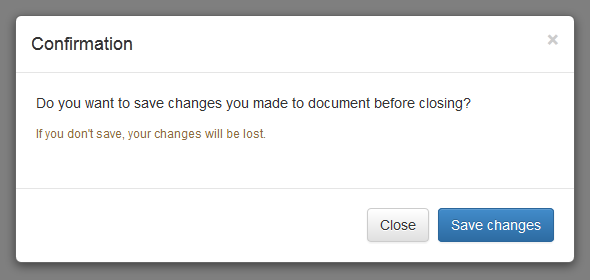


## 24. BOOTSTRAP MODELS

## Creating Modals with Bootstrap :

Modals are basically a dialog box that is used to provide important information to the user or prompt user to take necessary actions before moving on.

Modal windows are widely used to warn users for situations like session time out or to receive their final confirmation before going to perform any critical actions such as saving or deleting important data. You can easily create very smart and flexible dialog boxes with the Bootstrap modal plugin.



## Activate Modals via Data Attributes :

You can activate a Bootstrap modal by clicking on the button or link via data attributes without writing any JavaScript code.

## Activate Modals via JavaScript :

You may also activate a Bootstrap modal window via JavaScript — just call the modal()Bootstrap method with the modal "id" or "class" [selector](http://www.tutorialrepublic.com/css-tutorial/css-selectors.php) in your JavaScript code.

## Changing the Sizes of Modals :

Bootstrap gives you option further to scaling a modal up or down. You can a make modals larger or smaller by adding an extra class .modal-lg or .modal-sm on .modal-dialog.

## 25. BOOTSTRAP DROPDOWNS

## Creating Dropdown Menus with Bootstrap :

1. The dropdown menu is typically used inside the navigation header to display a list of related links when a user mouse hover or click on the trigger element.
2. You can use the Bootstrap dropdown plugin to add toggleable dropdown menus (i.e. open and close on click) to almost anything such as links, buttons or button groups, navbar, tabs and pills nav etc. without even writing a single line of JavaScript code.

## Adding Dropdowns via Data Attribute:

1. Bootstrap provides an easy and elegant mechanism for adding the dropdown menu to an element
2. via data attributes.

The following example will show you the minimum markup required for

1. adding a dropdown menu to the hyperlink via data attributes.

## Dropdowns within a Navbar

The following examples will show you how to add dropdowns to navbar.



## Dropdowns within Navs

The following example will show you how to add dropdowns to pills navs.



## Dropdowns within Buttons :

## The following examples will show you how to add dropdowns to buttons.



## Bootstrap Split Button Dropdowns

## 

## Dropdowns Inside Button Groups

To create dropdown menus inside a button group just place a .btn-group along with the dropdown markup within another .btn-group.



## Adding Headers to Dropdown Items

1. You can optionally add a menu header to label a section of menu items inside a dropdown menu
2. By adding the class .dropdown-header to the list element.

## 26. BOOTSTRAP TABS

In this tutorial you will learn how to create dynamic tabs to toggle between the content using the Bootstrap tabs plugin.

## Creating Tabs with Bootstrap

Tab based navigations provides an easy and powerful mechanism to handle huge amount of content within a small area through separating content into different panes where each pane is viewable one at a time. The user can quickly access the content through switching between the panes without leaving the page.



## Creating Dynamic Tabs via Data Attributes

1. You can activate a tab component without writing any JavaScript — simply specify the data
2. toggle="tab" on each tab, as well as create a .tab-pane with unique ID for every tab and wrap
3. them in .tab-content. The following example will show you how to create a basic tabbable
4. tabs via data attributes in Bootstrap.

## Creating Dynamic Tabs via JavaScript

You may also enable tabs via JavaScript. Each tab needs to be activated individually and You can activate individual tabs in several ways.

## 27. BOOTSTRAP TOOLTIPS

## Creating Tooltips With Bootstrap :

1. A tooltip is a small pop up that appears when user places the mouse pointer over an element such
2. as link or buttons to provide hint or information about the element being hovered.

Tooltips can be very helpful for the new visitors of your website because they enable the user to know the purpose of icons and links by placing the mouse pointer over them.

## 

## Triggering the Tooltips :

Tooltips can be triggered via JavaScript — just call the tooltip() Bootstrap method with the"id" or "class" [selector](http://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the target element in your JavaScript code.

## Setting the Directions of Tooltips :

You can set tooltips to appear on top, right, bottom and left sides of an element.

### Positioning of Tooltips via Data Attributes :

The following example will show you how to set the direction of tooltips via data attributes.

### Positioning of Tooltips via JavaScript :

The following example will show you how to set the direction of tooltips via JavaScript.

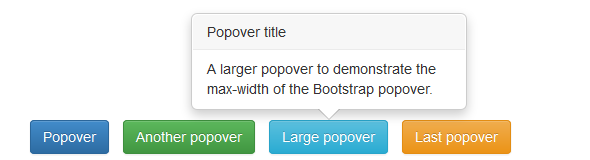
## 28. BOOTSTRAP POP OVERS

## Creating Popovers with Bootstrap

Popover is a small overlay of content that is used to display secondary information of any element when it is clicked by a user, like those on the iPad.

## Triggering the Popovers

Popovers can be triggered via JavaScript — just call the popover() Bootstrap method with the"id" or "class" [selector](http://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the required element in your JavaScript code.



## Setting the Directions of Popovers

You can set popovers to appear on top, right, bottom and left sides of an element.

### Positioning of Popovers via Data Attributes

The following example will show you how to set the direction of popovers via data attributes.

### Positioning of Popovers via JavaScript

The following example will show you how to set the direction of popovers via JavaScript.

## Hiding the Popovers on Next Click

1. By default popovers are not hiding until you click the trigger element once again. You can use
2. the focus trigger to hide the popovers when the user makes the next click.

## 29. BOOTSTRAP ALERTS

## Creating Alert Messages with Bootstrap

1. Alert boxes are used quite often to stand out the information that requires immediate attention of
2. the end users such as warning, error or confirmation messages.With Bootstrap you can easily
3. create elegant alert messages box for various purposes. You can also add an optional close button
4. to dismiss any alert.

## Warning Alerts

## You can create a simple Bootstrap warning alert message box by adding the contextual

## class .alert-warning to the .alert base class, like this.

## 

## Error or Danger Alerts

Add the class **.alert-danger** to the .alert base class to create error or danger alerts.



## Success or Confirmation Alerts

Likewise, to create the success or confirmation alert message box add the contextual class **.alert-success** to the .alert base class.



## Information Alerts

For information alert messages add the class .alert-info to the .alert base class.



## 30. BOOTSTRAP ACCORDION

## Creating Accordion Widget with Bootstrap :

Accordion widgets and menus are widely used on the websites to manage the large amount of content and navigation lists. With Bootstrap collapse plugin you can either create accordion or a simple collapsible panel without writing any JavaScript code.

## 

## Expanding and Collapsing Elements via Data Attributes

You can use the Bootstrap collapse feature for expanding and collapsing any specific element via data attributes without using the accordion markup.

## Expanding and Collapsing Elements via JavaScript

You may also expand and collapse an individual element manually via JavaScript, just call thecollapse() Bootstrap method with the "id" or "class" [selector](http://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the collapsible element in your JavaScript code.

## 31. BOOTSTRAP TYPEHEAD

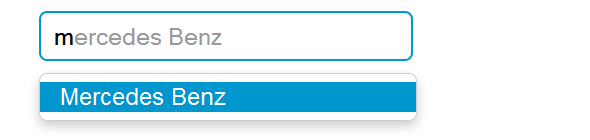
## Creating Typeaheads with Bootstrap

The typeahead input fields are very popular in modern web forms. The main purpose of using typeahead is to improve the user experience by supplying hints or a list of possible choices based on the text they've entered while filling a from or searching something — like the Google instant search. It also saves time and reduces the number of potential errors, because the user has less likelihood of making a spelling mistake.

Twitter typeaheads is a fast and fully-featured autocomplete library inspired by twitter.com's autocomplete search functionality. To create Twitter typeaheads first download [typeahead.js](http://twitter.github.io/typeahead.js/#_blank)from GitHub and include in your project after that you can turns any text-based [<input>](http://www.tutorialrepublic.com/html-reference/html-input-tag.php)element (i.e. input[type="text"]) into a typeahead.

## Creating Twitter Typeahead with Local Dataset

The following example will show you how to create Twitter typeahead with local dataset.



## Creating Twitter Type ahead External Dataset

You can also specify external dataset through a URL pointing to a JSON file containing an array of datums. The individual units that compose datasets are called datums.

## 32. BOOTSTRAP CAROUSEL

## Creating Carousels with Bootstrap

The carousel also known as slideshow or image slider is some of the best way of showcasing the huge amount of contents within a small space on the web pages. It is a dynamic presentation of contents where text and images are made visible or accessible to the user by cycling through several items.

The following example will show you how to build a simple carousel like image rotator using the Bootstrap carousel plugin. You can also add captions such as heading or description to the individual slides of the carousel.

## 

## Activate Carousels via Data Attributes

With Bootstrap you can create carousels very easily via data attributes without writing a single line of JavaScript code.

## Activate Carousels via JavaScript

You may also activate a carousel manually using the JavaScript, just call the carousel()method with the "id" or "class" [selector](http://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the wrapper element in your JavaScript code.

## 33. BOOTSTRAP SCROLLSPY & AFFIX

## Creating Scroll Spy with Bootstrap

The Bootstrap scroll spy is a navigation mechanism that automatically highlights the nav links based on the scroll position to indicate the visitor where they are currently on the page. The scroll spy will make your web page more elegant and accessible, if you are using the bookmark links for directing the visitors to the different sections of a page that has a huge amount of content. Here's a typical example of Bootstrap scroll spy.

## Creating Scroll Spy via Data Attributes

You can easily add scrollspy behavior to your top bar navigation via data attributes without writing a single line of JavaScript code. Let's check out the following example:

## Creating Scroll Spy via JavaScript

You may also add scrollspy manually using the JavaScript — just call the **scrollspy()** Bootstrap method with the "id" or "class" [selector](http://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the navbar in your JavaScript code.

## Creating Pinned Element with Bootstrap

In this tutorial we will learn how to create a pinned ([fixed positioned](http://www.tutorialrepublic.com/css-tutorial/css-position.php)) element and toggle

its pinning on and off using Bootstrap affix plugin. The pinning of an element is enabled

through changing the value of its [position](http://www.tutorialrepublic.com/css-reference/css-position-property.php) CSS property from static to fixed

To do this, the affix plugin toggles between three classes: **.affix, .affix-top, and .affix-bottom**. Each class represents a particular state.

* Initially, the plugin adds .**affix-top**or **.affix-bottom** class to indicate the element is in its top-most or bottom-most position.
* When the element scrolling past the offset limit provided by the "data-offset-"attribute the plugin replaces the .affix-top or .affix-bottom class with the .affixclass (sets position: fixed;), which trigger the actual affixing.
* At this point the appropriate CSS [top](http://www.tutorialrepublic.com/css-reference/css-top-property.php) or [bottom](http://www.tutorialrepublic.com/css-reference/css-bottom-property.php) property is required to determine the position of affix element on the viewport.

## Enable Affix via Data Attributes

You can easily add affix behavior to any element — just add data-spy="affix" to the element you want to spy on. Then use "data-offset-" attributes to define when to toggle the pinning of an element 'on' and 'off'.

## Enable Affix via JavaScript

You may also enable the affix plugin manually using the JavaScript — just call the affix()method with the "id" or "class" [selector](http://www.tutorialrepublic.com/css-tutorial/css-selectors.php) of the required element in your JavaScript code.

## 34. BOOTSTRAP STATEFULL BUTTONS

## Controlling Button States

In the previous section you've learnt about the Bootstrap button styling and the modifications as well as how to create button groups and toolbars. However, with Bootstrap you can do more with buttons like controlling the states of buttons, make checkbox and radio inputs behaves like toggle buttons, etc. Let's discuss about them in detail.

## Creating Single Toggle Button

You can activate toggling (i.e. change the normal state of a button to a push state and vice versa) on a single button by simply adding the data attribute data-toggle="button".



## Add Loading State on Buttons

You can change the normal state of a button to a loading state by simply adding the data attribute data-loading-text="Loading..." to a button.



## Creating Buttons Checkbox And Radio

You can add the attribute data-toggle="buttons" to a group of checkboxes and radio butttons for checkbox and radio styles toggling on button groups, like this: