UNIT I

CHAPTER 1

Introduction to Web Technologies

University Prescribed Syllabus

HTML: Getting started with HTML, Why HTML, Tags and Elements, Attributes, Properties, Headings list, Links, Tables, Images, HTML Form, Media (Audio, Video), Semantic HTML5 Elements.

CSS: Why CSS, Types of CSS, How to use CSS, Properties, Classes, Child-Class (Nested CSS), Colors, Text, Background, Border, Margin, Padding, Positioning (flex, grid, inline, block), Animation, Transition.

BOOTSTRAP: Why Bootstrap, CSS over Bootstrap, How to Use Bootstrap, Bootstrap Grid System, Bootstrap Responsive, Bootstrap Classes, Bootstrap Components (i.e., Button, Table, List, etc.), Bootstrap as a Cross Platform.

W3C: What is W3C , How W3C handles/Supports Web Technologies.

1.1 HTML 1-4

GQ. What is HTML ? 1-4

1.1.1 Structure of HTML Document 1-4

GQ. Explain the basic structure of HTML document. 1-4

1.1.2 HTML Versions 1-5

1.1.3 Features of HTML 1-5

1.1.4 Building Blocks of HTML 1-5

1.2 HTML ELEMENTS / TAGS 1-6

GQ. List various tags in HTML with simple example for a web page. 1-6

1.3 HTML Attribute 1-9

1.4 HTML Formatting 1-9

1.5 HTML Heading 1-11

GQ. Explain heading element in HTML. 1-11

1.6 Colors and Fonts 1-12

1.6.1 Colors 1-12

GQ. Explain different color model in HTML. 1-12

1.6.2 Font 1-13

GQ. Explain setting fonts in HTML. 1-13

1.7 Links – Hyperlinks 1-13

GQ. Explain following HTML Tags with suitable example <a> 1-13

GQ. What are the Hyperlinks and Anchors in the HTML? Explain with suitable example. 1-13

1.8 Frames 1-15
GQ. Explain how frames are constructed in HTML document. Explain with program. 1-15
GQ. How to create a frame in HTML? Explain with example. 1-15
1.8.1 <iframe> tag 1-18
GQ. Explain the concept of iframe with suitable example. 1-18
1.9 Lists 1-19
GQ. Explain the Lists in HTML. 1-19
1.10 Tables 1-21
GQ. How to create Tables on the web pages using HTML? 1-21
1.10.1 Table Tag with Attributes, TABLE, TR, TH, TD Tags 1-21
1.10.2 Borders, Cell Spacing, Cell Padding, Width, Align, Bgcolor Attribute 1-22
1.10.3 Colspan and Rowspan Attributes 1-23
1.11 Images 1-24
1.12 HTML Form 1-27
GQ. Explain Form element in HTML 1-27
1.13 HTML5 controls 1-31
GQ. Explain semantic elements in HTML% with an example. 1-31
1.13.1 Semantic Elements in HTML5 1-31
1.13.2 Media Elements 1-34
1.13.3 HTML5 New <input> types 1-35
1.14 CSS (Cascading Style Sheet) 1-36
GQ. What is CSS? 1-36
1.14.1 Advantages of CSS 1-36
GQ. State the advantages of CSS. 1-36
1.14.2 CSS Syntax 1-36
1.15 Types of CSS 1-37
GQ. What are the types of CSS? Explain with example. 1-37
1.16 CSS Selector 1-38
1.17 CSS Comments 1-40
1.17.1 CSS Colors 1-40
1.18 CSS Properties 1-42
1.18.1 Text 1-42
GQ. Write note on manipulating text in CSS with example. 1-42
1.18.2 Background 1-46
GQ. How to set background images in CSS? 1-46
1.18.3 Borders 1-47
1.18.4 Border Images 1-49
1.18.5 Margin 1-49
GQ. How to set margins in CSS? 1-49
1.19 Padding Lists 1-50
GQ. Write note on Padding lists in CSS with suitable example. 1-50
1.20 CSS Properties for Positioning an Element 1-50
GQ. What is positioning in CSS? Explain it with example. 1-50

1.21 CSS Display 1-52
1.22 CSS flex Property 1-55
1.23 CSS grid property 1-57
1.23.1 CSS3 Transformations 1-58
1.24 CSS Transitions 1-65
1.25 CSS Animations 1-68
1.26 Bootstrap 1-72
GQ. What is Bootstrap ? 1-72
1.26.1 Uses of Bootstrap 1-73
1.27 CSS Vs. Bootstrap 1-73
1.28 How to use Bootstrap in webpage ? 1-74
1.29 Bootstrap Grid System 1-76
GQ. What is Bootstrap Grid System? 1-76
1.30 Bootstrap Responsive Layout 1-83
GQ. What is Responsive Web Design. 1-83
1.30.1 Creating Responsive Layout with Bootstrap 1-83
1.31 Bootstrap Classes 1-85
1.32 Bootstrap Components 1-94
1.32.1 Bootstrap Buttons 1-94
1.32.2 Bootstrap Tables 1-97
1.32.3 Bootstrap Lists 1-112
1.32.4 Bootstrap Navbar 1-114
1.32.5 Bootstrap Forms 1-120
1.33 Cross-Browser Compatibility 1-125
1.34 W3C 1-125
GQ. What is W3C? 1-125
1.34.1 Characteristics of W3C 1-126
1.34.2 Advantages of W3C 1-126
1.34.3 Disadvantages of W3C 1-126
1.34.4 Importance of W3C Standards in Web Technologies 1-126
GQ. How W3C standards supports web technologies? 1-126
❑ **Chapter Ends** 1-127

1.1 HTML

GQ. What is HTML ?

HTML stands for **Hyper Text Markup Language.** Hypertext is nothing but the way in which the different web pages are linked with each other. Such links are called as Hypertext.

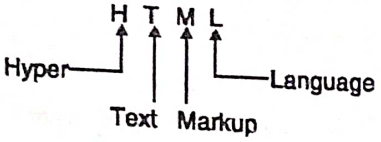


Fig. 1.1.1

It consists of main three elements.

1. **Hyper Text :** HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.
2. **Markup Language :** A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.
3. **Web Page :** A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. With the help of HTML only, we can create static web pages.

HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

1.1.1 Structure of HTML Document

GQ. Explain the basic structure of HTML document.

- All the elements are included in the main opening and closing <html> tags.
- An HTML document contains two main parts:
  1. **Head :** The head element contains title and metadata of a web document. This section is used to declare variables and functions in scripting languages. These variables and functions are then accessible throughout the page.
  2. **Body :** The body element contains the information which we like to display on a web page. All text to be displayed and control creation is done in this section.

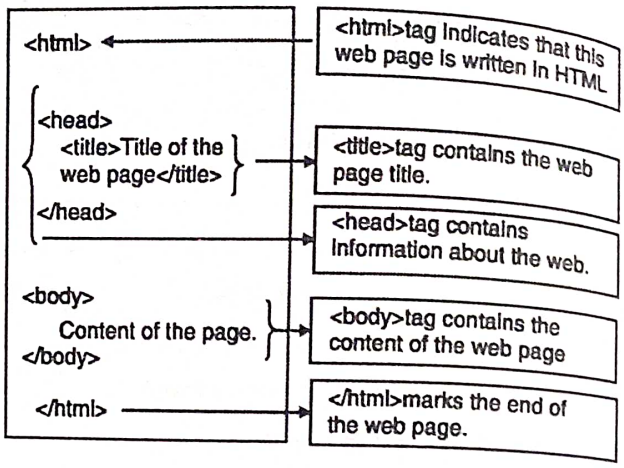


Fig. 1.1.2 : Structure of HTML Documents

Example

```
<!DOCTYPE>
<html>
<head>
<title>Web page title</title>
</head>
<body>
<h1>Write Your First Heading</h1>
<p>Write Your First Paragraph.</p>
</body>
</html>
```

▶ Description of HTML Example

1. **<!DOCTYPE> :** It defines the document type or it instruct the browser about the version of HTML.
2. **<html > :** This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except <!DOCTYPE>.
3. **<head> :** It should be the first element inside the <html> element, which contains the metadata (information about the document). It must be closed before the body tag opens.
4. **<title> :** I0t is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close immediately.
5. **<body> :** Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.

6. <h1> : Text between <h1> tag describes the first level heading of the webpage.
7. <p> : Text between <p> tag describes the paragraph of the webpage.

1.1.2 HTML Versions

Since the time HTML was invented, there are lots of HTML versions in market, the brief introduction about the HTML version is given below:

1. **HTML 1.0 :** The first version of HTML was 1.0, which was the barebones version of HTML language, and it was released in1991.
2. **HTML 2.0 :** This was the next version which was released in 1995, and it was standard language version for website design. HTML 2.0 was able to support extra features such as form-based file upload, form elements such as text box, option button, etc.
3. **HTML 3.2 :** HTML 3.2 version was published by W3C in early 1997. This version was capable of creating tables and providing support for extra options for form elements. It can also support a web page with complex mathematical equations. It became an official standard for any browser till January 1997. It is practically supported by most of the browsers.
4. **HTML 4.01 :** HTML 4.01 version was released on December 1999, and it is a very stable version of HTML language. This version is the current official standard, and it provides added support for stylesheets (CSS) and scripting ability for various multimedia elements.
5. **HTML5 :** HTML5 is the newest version of HyperText Markup language. The first draft of this version was announced in January 2008. There are two major organizations one is W3C (World Wide Web Consortium), and another one is WHATWG (Web Hypertext Application Technology Working Group) which are involved in the development of HTML 5 version, and still, it is under development.

1.1.3 Features of HTML

1. It is a very easy and simple language. It can be easily understood and modified.
2. It is very easy to make an effective presentation with HTML because it has a lot of formatting tags.
3. It is a markup language, so it provides a flexible way to design web pages along with the text.
4. It facilitates programmers to add a link on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.
5. It is platform-independent because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.
6. It facilitates the programmer to add Graphics, Videos, and Sound to the web pages which makes it more attractive and interactive.
7. HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

1.1.4 Building Blocks of HTML

An HTML document consist of its basic building blocks which are:

1. **Tags :** An HTML tag surrounds the content and apply meaning to it. It is written between < and > brackets.
2. **Attribute :** An attribute in HTML provides extra information about the element, and it is applied within the start tag. An HTML attribute contains two fields: name & value.

Syntax

```
<tag name attribute_name= " attr_value"> content </ tag name>
```

3. **Elements :** An HTML element is an individual component of an HTML file. In an HTML file, everything written within tags are termed as HTML elements.

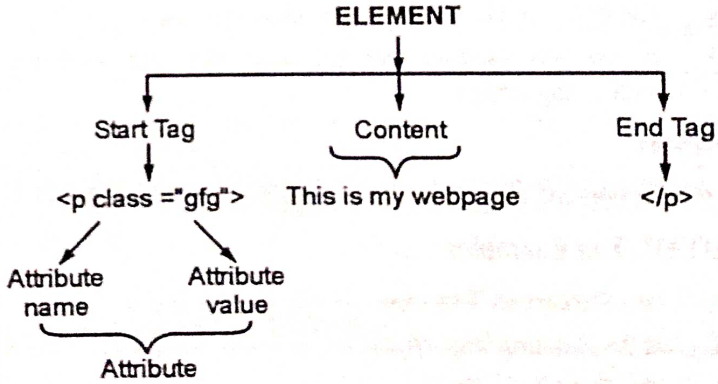


Fig. 1.1.3 : HTML Element

Example

```
<!DOCTYPE html>
<html>
 <head>
  <title>The basic building blocks of HTML</title>
 </head>
 <body>
```

```
    <h2>The building blocks</h2>
    <p>This is a paragraph tag</p>
    <p style= "color: red">The style is attribute of
paragraph tag</p>
    <span>The element contains tag, attribute and
content</span>
  </body>
</html>
```

1.2 HTML ELEMENTS / TAGS

GQ. List various tags in HTML with simple example for a web page.

- HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content.
- HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.
- When a web browser reads an HTML document, browser reads it from top to bottom and left to right.
- HTML tags are used to create HTML documents and render their properties.
- Each HTML tags have different properties. An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text.
- All HTML tags must enclose within < > these brackets.
- Every tag in HTML performs different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag>.

Syntax

```
<tag> content </tag>
```

HTML Tag Examples

1. Paragraph Tag
2. <h2> Heading Tag </h2>
3. Bold Tag
4. Italic Tag
5. <u> Underline Tag</u>

HTML tags are divided into following types

1. Unclosed HTML Tags

Some HTML tags are not closed, for example br and hr.

1.
 Tag: br stands for break line, it breaks the line of the code.
2. <hr> Tag: hr stands for Horizontal Rule. This tag is used to put a line across the webpage.

2. HTML Meta Tags

DOCTYPE, title, link, meta and style tag.

3. HTML Text Tags

, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , , <abbr>, <acronym>, <address>, <bdo>, <blockquote>, <cite>, <q>, <code>, <ins>, <del>, <dfn>, <kbd>, <pre>, <samp>, <var> and

4. Link Tags

<a> and <base>

5. HTML Image and Object Tags

<img>, <area>, <map>, <param> and <object>

6. HTML List Tags

<ul>, <ol>, <li>, <dl>, <dt> and <dd>

7. HTML Table Tags

table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption

8. HTML Form Tags

form, input, textarea, select, option, optgroup, button, label, fieldset and legend

9. HTML Scripting Tags

script and noscript

• NOTES •

HTML Tag List

Tag Name	Description
<!-- -->	This tag is used to apply comment in an HTML document.
<!DOCTYPE>	This tag is used to specify the version of HTML
<a>	It is termed as anchor tag and it creates a hyperlink or link.
<abbr>	It defines an abbreviation for a phrase or longer word.
<address>	It defines the author's contact information of the HTML article
<area>	It defines the area of an image map.
<b>	It is used to make a text bold.
<base>	This tag defines the base URL for all relative URL within the document.
<bdo>	It is used to override the current text direction.
<blockquote>	It is used to define a content which is taken from another source.
<body>	It is used to define the body section of an HTML document.
 	It is used to apply single line break.
<button>	It is used to represent a clickable button
<caption>	It is used to define a caption for a table.
	It is used to align the content in center.
<cite>	It is used to define the title of the work, book, website, etc.
<code>	It is used to display a part of programming code in an HTML document.
<col>	It defines a column within a table which represent common properties of columns and used with the <colgroup> element.
<colgroup>	It is used to define group of columns in a table.
<dd>	It is used to provide definition/description of a term in description list.
<del>	It defines a text which has been deleted from the document.
	It defines a division or section within HTML document.
<dl>	It is sued to define a description list.
<dt>	It is used to define a term in description list.
	It is used to emphasis the content applied within this element.
<fieldset>	It is used to group related elements/labels within a web form.
<font>	It defines the font, size, color, and face for the content.
<form>	It is used to define an HTML form.
<frame>	It defines a particular area of webpage which can contain another HTML file.
<frameset>	It defines group of Frames.
<h1> to <h6>	It defines headings for an HTML document from level 1 to level 6.
<head>	It defines the head section of an HTML document.
<hr>	It is used to apply thematic break between paragraph-level elements.
<html>	It represents root of an HTML document.
<i>	It is used to represent a text in some different voice.
<iframe>	It defines an inline frame which can embed other content.
<img>	It is used to insert an image within an HTML document.

Tag Name	Description
<input>	It defines an input field within an HTML form.
<label>	It defines a text label for the input field of form.
<legend>	It defines a caption for content of <fieldset>
<li>	It is used to represent items in list.
<link>	It represents a relationship between current document and an external resource.
<marquee>	It is used to insert the scrolling text or an image either horizontally or vertically.
<menu>	It is used for creating a menu list of commands.
<meta>	It defines metadata of an HTML document.
<noframes>	It provides alternate content to represent in browser which does not support the <frame> elements.
<noscript>	It provides an alternative content if a script type is not supported in browser.
<object>	It is used to embed an object in HTML file.
<ol>	It defines an ordered list of items.
<optgroup>	It is used to group the options of a drop-down list.
<option>	It is used to define options or items in a drop-down list.
<p>	It represents a paragraph in an HTML document.
<param>	It defines parameter for an <object> element
<pre>	It defines preformatted text in an HTML document.
<script>	It is used to declare the JavaScript within HTML document.
<select>	It represents a control which provides a menu of options.
<small>	It is used to make text font one size smaller than document?s base font size.
	It is used for styling and grouping inline.
<strike>	It is used to render strike through the text.
	It is used to define important text.
<style>	It is used to contain style information for an HTML document.
<sub>	It defines a text which displays as a subscript text.
<sup>	It defines a text which represent as superscript text.
<svg>	It is used as container of SVG (Scalable Vector Graphics).
<table>	It is used to present data in tabular form or to create a table within HTML document.
<tbody>	It represents the body content of an HTML table and used along with <thead> and <tfoot>.
<td>	It is used to define cells of an HTML table which contains table data
<template>	It is used to contain the client-side content which will not display at time of page load and may render later using JavaScript.
<textarea>	It is used to define multiple line input, such as comment, feedback, and review, etc.
<tfoot>	It defines the footer content of an HTML table.
<th>	It defines the head cell of an HTML table.
<thead>	It defines the header of an HTML table. It is used along with <tbody> and <tfoot> tags.
<u>	It is used to render enclosed text with an underline.
<ul>	It defines unordered list of items.

1.3 HTML ATTRIBUTE

HTML attributes are special words which provide additional information about the elements or attributes are the modifier of the HTML element. Each element or tag can have attributes, which defines the behaviour of that element.

- Attributes should always be applied with start tag.
- The Attribute should always be applied with its name and value pair.
- The Attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.
- You can add multiple attributes in one HTML element, but need to give space between two attributes.

Syntax

```
<element attribute_name="value">content</element>
```

Example

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <h1> This is Style attribute</h1>
  <p style="height: 50px; color: blue">It will add style
property in element</p>
  <p style="color: red">It will change the color of
content</p>
</body>
</html>
<p style="height: 50px; color: blue">It will add style
property in element</p>
```

In the above statement, we have used paragraph tags in which we have applied style attribute. This attribute is used for applying CSS property on any HTML element. It provides height to paragraph element of 50px and turns it colour to blue.

```
<p style="color: red">It will change the color of
content</p>
```

In the above statement we have again used style attribute in paragraph tag, which turns its colour red.

☞ Title attribute in HTML

- The title attribute is used as text tooltip in most of the browsers. It displays its text when user move the cursor over a link or any text. You can use it with any text or link to show the description about that link or text.
- Following example shows paragraph tag and heading tag.

Example

```
<!DOCTYPE html>
<html>
 <head>
</head>
<body>
 <h1 title="This is heading tag">Example of title
attribute</h1>
  <p title="This is paragraph tag">Welcome to Title
attribute</p>
 </body>
</html>
```

☞ href attribute in HTML

- The href attribute is the main attribute of <a> anchor tag. This attribute gives the link address which is specified in that link. The href attribute provides the hyperlink, and if it is blank, then it will remain in same page.

Example

```
With link address:
<a href="https://www.loopholetech.com/html-anchor">This
is a link</a>
Without link address:
<a href="">This is a link</a>
```

☞ src Attribute

- The src attribute is one of the important and required attribute of <img> element. It is source for the image which is required to display on browser. This attribute can contain image in same directory or another directory. The image name or source should be correct else browser will not display the image.

Example

```
<img src="tulip.jpg" height="400" width="600">
```

1.4 HTML FORMATTING

- HTML Formatting is a process of formatting text for better look and feel. HTML provides us ability to format text without using CSS. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined.

- In HTML the formatting tags are divided into two categories:
  1. **Physical tag :** These tags are used to provide the visual appearance to the text.
  2. **Logical tag :** These tags are used to add some logical or semantic value to the text.
- Following is the list of HTML formatting text.

Element name	Description
	This is a physical tag, which is used to bold the text written between it.
	This is a logical tag, which tells the browser that the text is important.
	This is a physical tag which is used to make text italic.
	This is a logical tag which is used to display content in italic.
<mark>	This tag is used to highlight text.
<u>	This tag is used to underline text written between it.
<sup>	It displays the content slightly above the normal line.
<sub>	It displays the content slightly below the normal line.
<del>	This tag is used to display the deleted content.
<ins>	This tag displays the content which is added
<big>	This tag is used to increase the font size by one conventional unit.
<small>	This tag is used to decrease the font size by one unit from base font size.

1. Bold Text

- HTML and are formatting elements.
- The HTML element is a physical tag which display text in bold font, without any logical importance. If you write anything within ..... element, is shown in bold letters.

Example

```
<p> <b>Write Your First Paragraph in bold text. </b></p>
```

- The HTML tag is a logical tag, which displays the content in bold font and informs the browser about its logical importance. If you write anything between ..... , is shown important text.

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>formatting elements</title>
</head>
<body>
<h1>Explanation of formatting element</h1>
<p><strong>This is an important content</strong>, and this is normal content</p>
</body>
</html>
```

2. Italic Text

HTML and are formatting elements.

- The HTML element is physical element, which display the enclosed content in italic font, without any added importance. If you write anything within ..... element, is shown in italic letters.

Example

```
<p> <i>Write Your First Paragraph in italic text. </i></p>
```

- The HTML tag is a logical element, which will display the enclosed content in italic font, with added semantics importance.

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>formatting elements</title>
</head>
<body>
<h1>Explanation of italic formatting element</h1>
<p><em>This is an important content</em>, which displayed in italic font.</p>
</body>
</html>
```

3. Marked formatting

If you want to mark or highlight a text, you should write the content within <mark>.....</mark>.

Example

```
<h2> I want to put a <mark> Mark</mark> on your face</h2>
```

4. Underlined Text

If you write anything within <u>.....</u> element, is shown in underlined text.

Example

 <u>Write Your First Paragraph in underlined text. </u>

5. Strike Text

Anything written within <strike>.....</strike> element is displayed with strikethrough. It is a thin line which cross the statement.

Example

 <strike>Write Your First Paragraph with strikethrough</strike>.

6. Monospaced Font

If you want that each letter has the same width then you should write the content within <tt>.....</tt> element.

Example

Hello <tt>Write Your First Paragraph in monospaced font. </tt>

7. Superscript Text

If you put the content within ^{.....} element, is shown in superscript; means it is displayed half a character's height above the other characters.

Example

Hello ^{Write Your First Paragraph in superscript.}

8. Subscript Text

If you put the content within _{.....} element, is shown in subscript; means it is displayed half a character's height below the other characters.

Example

Hello _{Write Your First Paragraph in subscript.}

1.5 HTML HEADING

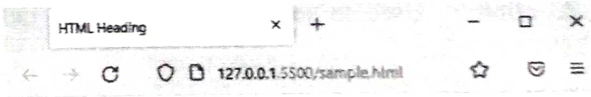
GQ. Explain heading element in HTML.

- **Use :** In general a document starts with heading. The heading elements are used to give title or subtitles to be displayed on the webpage. The different headings may require different sizes as per their importance. HTML has six levels of headings. Six elements namely <h1>, <h2>, <h3>, <h4>, <h5>, and <h6> are used to give headings. When any of the heading tag is used, one line before and one line after that heading is added by the browser.
- <h1> is the biggest heading tag while <h6> is the smallest heading tag. Hence h1 should be used for the most important heading text and h6 should be used for least important heading text.

Program 1.5.1 : Write a program to display all heading tags from <h1> to <h6>.

```
<!DOCTYPE html>
<html>

<head>
        <title>HTML Heading</title>
</head>

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

</html>
```

Output



Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

1.6 COLORS AND FONTS

1.6.1 Colors

GQ. Explain different color model in HTML.

The look and feel of a website is dependent upon the colors used in a webpage for text and backgrounds. The colors can be specified at page level with the help of <body> tag or can be set for individual tags.

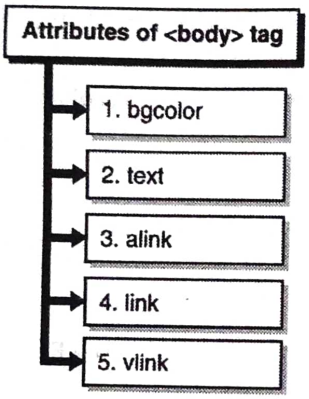


Fig. 1.6.1 : Attributes of <body> tag

The <body> tag has number of attributes which are used to set colors to different entities.

- **1. bgcolor :** Used to set color to the background of the page.
- **2. text :** Used to set color to the body text.
- **3. alink :** Used to set color to the active links or selected links.
- **4. link :** Used to set color to the linked text.
- **5. vlink :** Used to set color to the visited links – that is, for linked text that you have already clicked on.

Color Coding Methods in HTML

There are different methods to set colors in the web page as follows :

(i) **Color names** – We can directly specify color name like red, green, blue etc.

(ii) **Hex codes** – It is the six-digit code which represents the expected color.

(iii) **Color decimal or percentage values** – This is also called as RGB color scheme. Here we have to specify the amount of colors red, green and blue. The mixture of this gives expected color.

1. HTML Colors - Color Names

- In this method the color names are directly assigned to set color for text or background. The W3C (World Wide Web Consortium) has listed 16 standard color names for HTML. But maximum of browsers supports more than 200 color names.
- The W3C Standard 16 Colors are as follows:
- Black, Yellow, Red, Maroon, Gray, Lime, Green, Olive, Silver, Aqua, Blue, Navy, white, Fuchsia, purple, Teal.

2. HTML Colors - Hex Codes

- This is the six digit representation of a color. The initial two digits (RR) represent the red value, the next two digits represent green value (GG), and the last two digits represent the blue value (BB).
- There are various graphics software like MS Paint, Adobe Photoshop or Paintshop Pro where we can get the hexadecimal value.
- Each hexadecimal code is preceded with pound or hash sign (#).

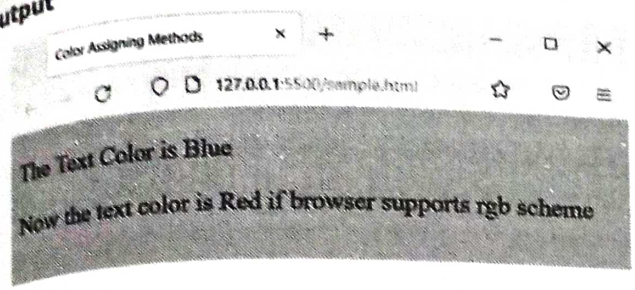
3. HTML Colors - RGB Values

- The RGB method takes three values as arguments for Red, Green and Blue colors. These values are in integer format which range from 0 to 255 depending upon the required intensity of color.
- Not all the browsers support this method, hence not suggestive.

Program 1.6.1 : Write a simple code to display the text in various color format.

```
<!DOCTYPE html>
<html>

<head>
        <title>Color Assigning Methods</title>
</head>

<body text="blue" bgcolor="#00FF00">
        <p>The Text Color is Blue</p>
        <font color="rgb(255,0,0)">Now the text color
is Red if browser supports rgb scheme</font>
</body>

</html>
```

Output



1.6.2 Font

GQ. Explain setting fonts in HTML.

- User friendliness is the basic need of any website. This can be achieved by making it readable with the help of <font> tag. The text in a webpage can be formatted by setting the < font > ..... < /font > tag and various font attributes.
- The <font> tag has following attributes :

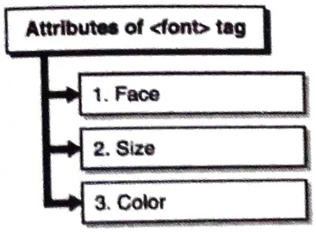


Fig. 1.6.2 : Attributes of <font> tag

1. Face

- **Use :** The Face attribute is used to specify the name of font for the text.

```
<font face="Arial"> Arial Font </font>
```

- The specified font must be installed on the machine where the web page is running; otherwise the text will be displayed in default font of the web browser.
- To the face attribute, we can give multiple font names separated by a comma.

```
<font face="Sans serif, Comic Sans MS, Lucida Console"> Multiple fonts</font>
```

- The first specified font is taken into consideration by the browser. But if it is not available on the machine then second font will be applied and if second font is not available then third font will be applied. Even this is also not available then the default font is applied to the text.

2. Size

- **Use :** The size of a font can be set using the size attribute. The allowed range of values for font size is from 1(smallest) to 7(largest). 3 is the default font size.

```
<font size=5>Size is 5</font>
```

- It is also possible to set the relative font size. That means how many sizes greater or smaller than the current font size.

```
<font size="-2">Less by 2</font>
<font size="+2">Greater by 2</font>
```

3. Color

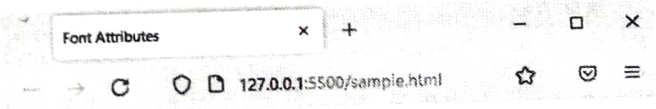
- **Use :** Used to set the color of font. This attribute can accept value as standard name of a color or color code.

Program 1.6.2 : Write a program to display font size, font face and color.

Program to display the font size, color and font face

```
<!DOCTYPE html>
<html>
<head>
        <title>Font Attributes</title>
</head>
<body>
        <font face="Times New Roman,Sans serif"
size="5" color="red">
                "Extraordinary things are always
hiding in places people never think to look."
        </font>
</body>
</html>
```

Output



"Extraordinary things are always hiding in places people never think to look."

1.7 LINKS – HYPERLINKS

GQ. Explain following HTML Tags with suitable example <a>

GQ. What are the Hyperlinks and Anchors in the HTML? Explain with suitable example.

- **Hyperlinks** are used to navigate in the website. That means to move from one webpage to another. A webpage can have various hyperlinks which can take us directly to another pages and even specific parts of the current page. For hyper linking words, phrases or even images can be used.

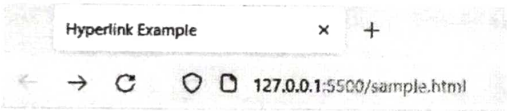
- The **anchor tag** <a> is used to specify the hyperlink. The text written in opening <a> and closing</a> tag is known as hypertext. When this hypertext is clicked, the target webpage get opened. The hypertext has default formatting. It is in blue color and has underline.

Program 1.7.1 : Write a program to display hyperlink example.

```
<!DOCTYPE html>
<html>
<head>
        <title>Hyperlink Example</title>
</head>
<body>
        <p>Click the following link</p>
        <a
href="https://www.sjcem.edu.in">SJCEM</a>
</body>
</html>
```

The href attribute is used to specify the address of webpage which we want to display.

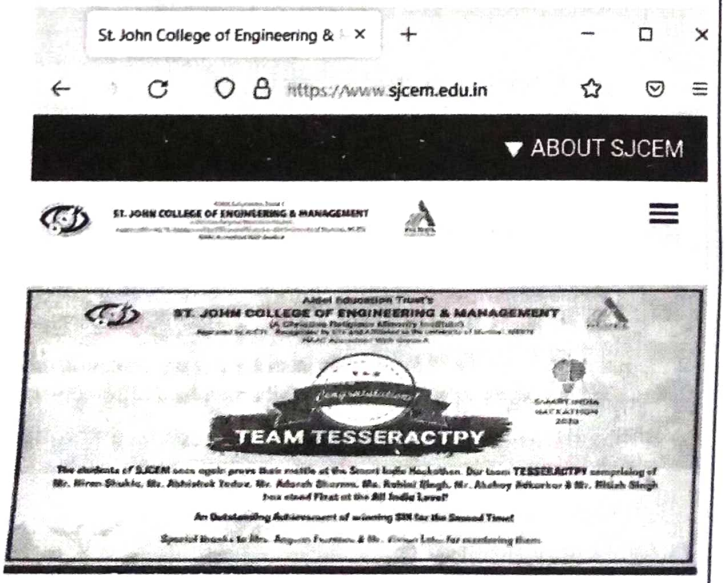
Output



Click the following link

SJCEM

When this link is clicked, the home page of website will display as shown in following screenshot.



The Target Attribute

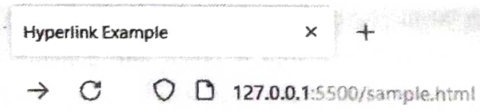
The target attribute is used to specify location where the linked webpage should be opened. Different values for target attribute are as follows :

Sr. No	Option	Description
1	_blank	The linked webpage is opened in new window or tab.
2	_self	The linked webpage is opened in the same frame.
3	_parent	The linked webpage is opened in the parent frame.
4	_top	The linked webpage is opened in the full body of the window.
5	targetframe	The linked webpage is opened in a named targetframe.

Program 1.7.2 : Write a program to display the hyperlinks using target attribute.

```
<!DOCTYPE html>
<html>
<head>
        <title>Hyperlink Example</title>
</head>
<body>
        <p>Click any of the following links</p>
        <a href="https://spcasjcem.herokuapp.com/"
target="_blank">Opens in New</a> <br>
        <a href="https://spcasjcem.herokuapp.com/"
target="_self">Opens in Self</a> <br>
        <a href="https://spcasjcem.herokuapp.com/"
target="_parent">Opens in Parent</a> <br>
        <a href="https://spcasjcem.herokuapp.com/"
target="_top">Opens in Body</a>
</body>
</html>
```

Output



Click any of the following links

Opens in New
Opens in Self
Opens in Parent
Opens in Body

Use of Base Path

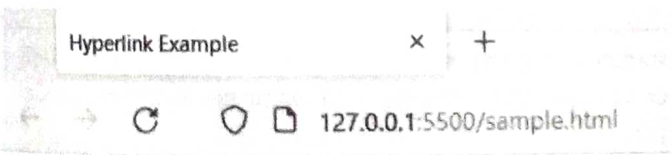
When web pages related to same website are linked, then it is not necessary to specify the complete path of the webpage. This path is known as URL. In document header we can use <base> tag to set base path for all the links. The browser will concatenate the given relative path with this base path and a complete URL is generated.

Program 1.7.3 : Write a simple code for <base> tag.

In this program we will use <base> tag to specify the base URL and afterwards we can use the relative path for all the links rather than giving full URL for every link.

```
<!DOCTYPE html>
<html>
 <head>
  <title>Hyperlink Example</title>
  <base href = " https://spcasjcem.herokuapp.com/">
 </head>
 <body>
  <p>Click following link</p>
  <a href = "https://spcasjcem.herokuapp.com/events/"
target = "_blank">Student Association:SPCA</a>
 </body>
</html>
```

Output



Click following link

Student Association:SPCA

Linking to a Page Section

It is also possible to link specific portion of the webpage with the help of name attribute of anchor <a> tag. There are two steps to implement it.

Step 1

Initially we have to create a link where we want to go within the webpage and give name to it using <a> tag as follows:

```
<h1>Link Section <a name = "top"></a></h1>
```

Step 2

Now we have to create hyperlink to link the webpage and place where we want to reach –

```
<a href = "/html/html_text_links.htm#top">Go to the
Top</a>
```

This will generate following link, where we can click to reach to the top of the webpage.

Go to the Top

1.8 FRAMES

GQ. Explain how frames are constructed in HTML document. Explain with program.

GQ. How to create a frame in HTML? Explain with example.

- Usually we can display only one webpage at a time in the browser window.
- If we wish to display multiple pages at the same time on the browse window, then we can use the frames to divide the browser window into different sections and in each section we can display a separate webpage. A group of frames in the browser window is called as a frameset.
- While dividing the window, it is considered as a table and organized in rows and columns.
- In the frameset application, basic tag of html like <body> is not used. Instead of it, <frameset> tag is used. This tag defines how to divide the window into different sections. There are two main attributes of frameset tag.
  1. **Rows** – Defines the horizontal frames.
  2. **Cols** – Defines the vertical frames
- The <frame> is sub-tag of <frameset> tag. The <frame> tag has attribute src (source) to which we have to specify the URL of webpage which we want to display in particular frame.

☞ Attributes of the <frameset> tag

Following are the important attributes of tag <frameset>:

Sr. No	Attribute	Description
1.	Cols	Specifies the number of columns contained in the frameset and also the size of all the columns. Width of each column can be specified in four ways: (1) Absolute values can be given in pixels. For example, to generate three vertical frames, we can set cols = "200, 400, 100". (2) A percentage amount of the current browser window. For example, to generate three vertical frames, use cols = "20%, 70%, 10%". (3) Using the special wildcard symbol. For example, to generate three vertical frames, use cols = "20%, *, 20%". Here the wildcard takes remaining percentage of the window. (4) As relative widths of current browser window. For example, to generate three vertical frames, use cols = "3*, 2*, 1*". This is substitute of percentages. The relative widths of the browser window can be used. This divides the window as: first column occupies half of the window, the second occupies one third, and the third occupies one sixth.
2.	rows	This attribute is same as of the cols attribute and accepts the same values, but it is used to mention the rows in the frameset. For example, to generate two horizontal frames, use rows = "20%, 80%". The height of each row can be specified in the same way as we have seen for columns.
3.	border	The width of the border of each frame can be specified with border attribute. For example, border = "3". A value of zero indicates frame without border.
4.	frameborder	This attribute mentions whether to display a three-dimensional border between frames or not. It takes value either 1 (true) or 0 (false). For example frameborder = "1" specifies frame with three-dimensional border.
5.	Framespacing	The amount of space between different frames in a frameset can be specified with the help of this attribute. It accepts any integer value. For example framespacing = "5" indicates that there should be 5 pixels spacing between the frames.

☞ The <frame> Tag Attributes

Following are the important attributes of <frame> tag :

Sr. No	Attribute	Attribute & Description
1.	Src	This attribute specifies the URL of web page that should be loaded in the frame. **Example :** src = "/html/index.htm".
2.	Name	This attribute specifies the name to a frame. This is usually important when we want to establish links in one frame that load web pages into another frame. In this case the second frame must have a name to identify itself as the target for the link.
3.	Frameborder	This attribute is used to set whether or not the borders of the frame should be displayed; it overwrites the value specified in the frameborder attribute of the <frameset> tag if it is specified. It takes value either 1 (true) or 0 (false).
4.	Marginwidth	This attribute specifies the width of the space between the left and right borders of the frame and the content of the frame. The value is specified in pixels. **Example :** marginwidth = "5".

Sr. No	Attribute	Attribute & Description
5.	Marginheight	This attribute specifies the height of the space between the top and bottom borders of the frame and its contents. The value is specified in pixels. For example marginheight = "5".
6.	Noresize	The borders of the frames can be resized by default. If we want to restrict user from resizing the frame then we can set : noresize = "noresize".
7.	Scrolling	This attribute specifies the behavior of the scrollbars which appear on the frame. It accepts value either "yes", "no" or "auto". **Example :** scrolling = "yes" indicates that the frame should have scroll bars.
8.	longdesc	This attribute specifies link to another page which contains a long description about the contents of the frame. **Example :** longdesc = "details.htm"

Browser Support for Frames

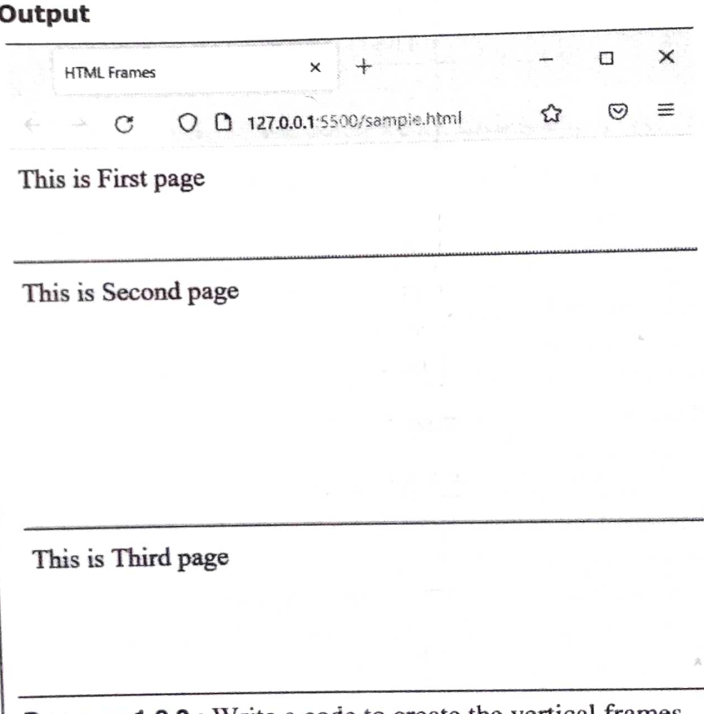
- If any user has old browser or such a browser which does not have support for frames then the <noframes> element is shown on the browser.
- In such case we have to place the <body> element inside the <noframes> element since the <frameset> element should be replaced by the <body> element.

Program 1.8.1 : Write a code to create horizontal frames.

```
<!DOCTYPE html>
<html>
 <head>
  <title>HTML Frames</title>
 </head>
 <frameset rows = "20%,50%,30%">
  <frame name = "top" src = "First.html" />
  <frame name = "main" src = "Second.html" />
  <frame name = "bottom" src = "Third.html" />
  <noframes>
   <body>Your browser does not support frames.</body>
  </noframes>
 </frameset>
</html>
```

Note : Here it is considered that pages First.html, Second.html and Third.html are already created.

Output

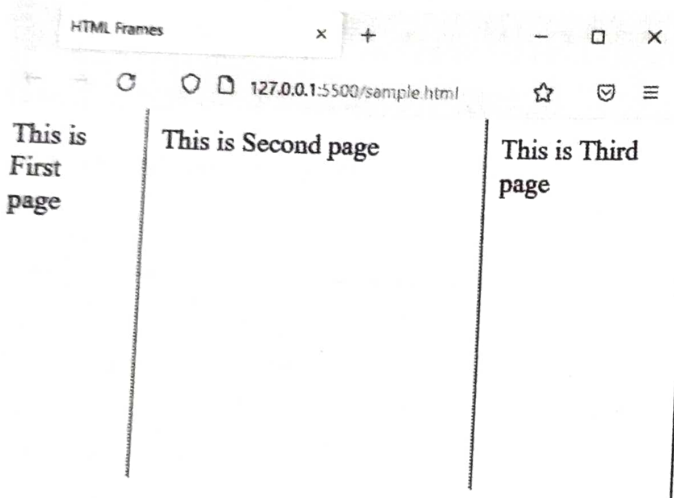


Program 1.8.2 : Write a code to create the vertical frames.

```
<!DOCTYPE html>
<html>
 <head>
  <title>HTML Frames</title>
 </head>
 <frameset cols = "20%,50%,30%" border=10
 framespacing=200>
  <frame name = "left" src = "First.html" />
  <frame name = "center" src = "Second.html" />
```

```
    <frame name = "right" src = "Third.html"
scrolling="no"/>
    <noframes>
      <body>Your browser does not support
frames.</body>
    </noframes>
  </frameset>
</html>
```

Output



☞ Frame's name and target attributes

One of the best uses of frame is to place navigation bars in one frame and then load the link web pages into a separate frame.

Program 1.8.3 : Write a code to divide the browser window into two frames.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Target Frames</title>
  </head>
  <frameset cols = "200, *">
    <frame src = "menu.html" name = "menu_page" />
    <frame src = "main.html" name = "main_page" />
  </frameset>
</html>
```

Here the browser window is divided into two frames. The first frame is small and will contain the only the navigation menu options of menu.html page. The second column occupies near about 80% of size and contain the main part of the page and it is implemented by main.htm file. In the links of the menu we will specify the target frame as main_page. Whenever any link in the menu is clicked, the related page will be opened in main page.

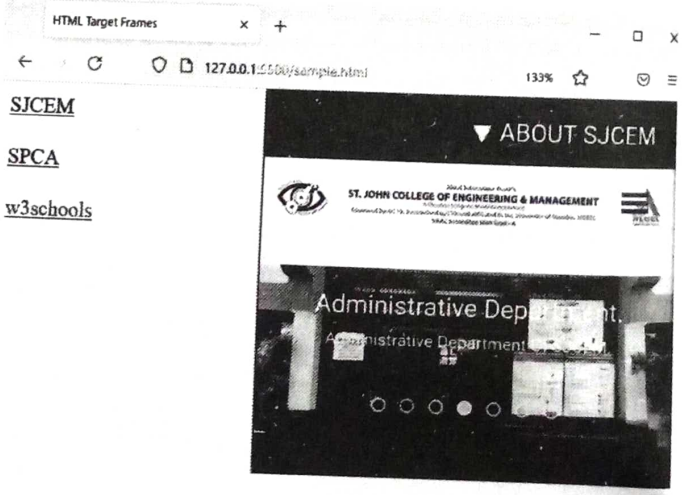
menu.html file

```
<!DOCTYPE html>
<html>
  <body>
    <a href = "https://www.sjcem.edu.in/"
   target = "main_page">SJCEM</a>
    <br />
    <br />
    <a href = "https://spcasjcem.herokuapp.com/events/"
   target = "main_page">SPCA</a>
    <br />
    <br />
    <a href = "https://www.w3schools.com/"
   target = "main_page">w3schools</a>
  </body>
</html>
```

main.html file

```
<!DOCTYPE html>
<html>
  <body bgcolor = "skyblue">
    <h3>Welcome Web Technology</h3>
    <p>Click any link and page will be displayed here.</p>
  </body>
</html>
```

Output



1.8.1 <iframe> tag

GQ. Explain the concept of iframe with suitable example.

- **iframe stands for inline frame.** It is defined using tag <iframe>.

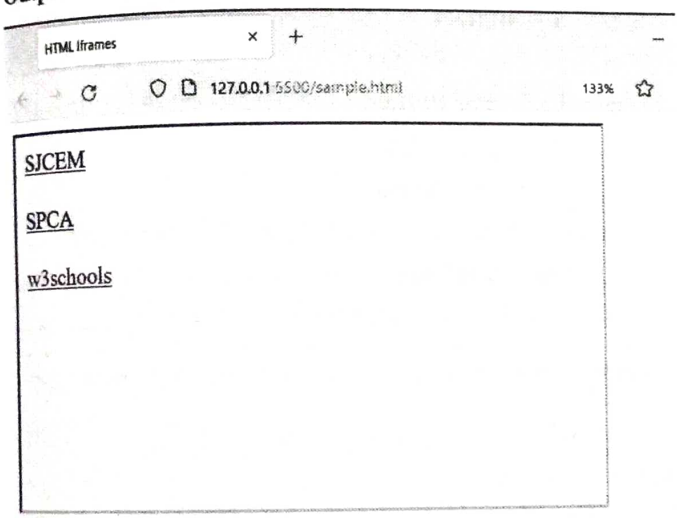
Use : The <iframe> tag defines a region with rectangular shape inside the main page where the browser should display another webpage with scrollbars and borders.

The URL of webpage which occupies the inline frame is specified by the src attribute.

Program 1.8.4 : Write a code to define <iframe> tag.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Iframes</title>
  </head>
  <body>
    <iframe src = "menu.html" width = "400"  height =
"250">
    </iframe>
    <p>Document content go here...</p>
  </body>
</html>
```

Output



1.9 LISTS

GQ. Explain the Lists in HTML.

- Lists are used everywhere on the websites.
- There are number of things like articles, website navigation menus, and product features on e-commerce websites which makes frequent use of lists.
- HTML provides three types ways to give the lists :

1. An unordered list <ul>
2. An ordered list <ol>
3. A definition list <dl>

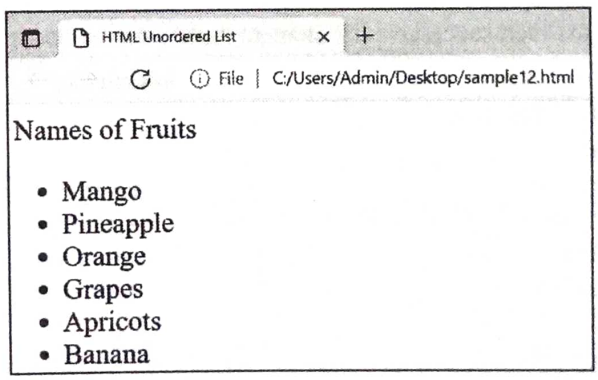
1. HTML Unordered Lists

- In this list, plain bullets are used for the list items. Generally this option is used when there is no any standard sequence or order of the list items. The <ul> tag is used to give this list. Every element in the list is marked by a bullet. Different types of bullets are available.
- The <li> which is the sub-tag of <ul> is used for individual list elements.

Program 1.9.1 : Write a code for displaying the list elements.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Unordered List</title>
  </head>
  <body>
<font size=5>
Names of Fruits
    <ul>
      <li>Mango</li>
      <li>Pineapple</li>
      <li>Orange</li>
      <li> Grapes </li>
      <li> Apricots</li>
      <li> Banana</li>
    </ul>
</font>
  </body>
</html>
```

Output



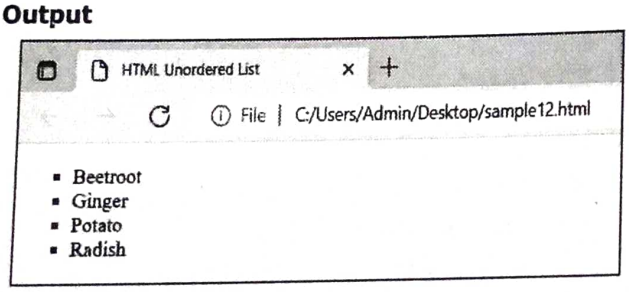
- The <ul> tag has attribute *TYPE* which is used to specify the type of bullet for the list items. The default type is disc which is displayed in above output.
- There following options for the attribute *TYPE* of <ul> tag.

  (1) Square (2) Circle (3) Disc
- See the Program 1.9.2 which illustrates the use of attribute *TYPE*.

Program 1.9.2 : Write a simple code demonstrating the use of <ul> tag.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Unordered List</title>
  </head>
  <body>
    <ul type = "square">
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ul>
  </body>
</html>
```

Output



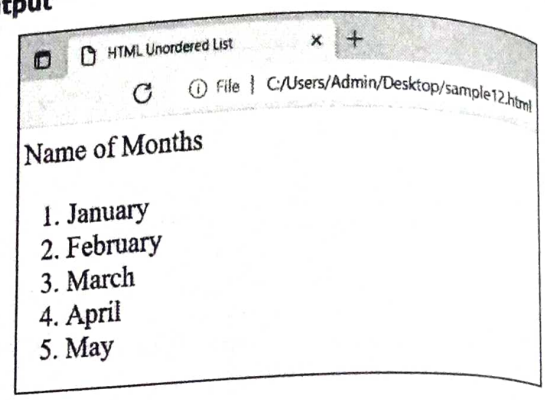
2. HTML Ordered Lists

- Ordered lists are generally used when we want to specify numbers instead of bullets. The <ol> tag is used for such list.
- The number starts from 1 and also incremented by one for each successive list element in the ordered list.

Program 1.9.3 : Write a script to display ordered list.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Unordered List</title>
  </head>
  <body>
<font size=5>
Name of Months
    <ol>
      <li>January</li>
      <li>February</li>
      <li>March</li>
      <li>April</li>
      <li>May</li>
    </ol>
</font>
  </body>
</html>
```

Output

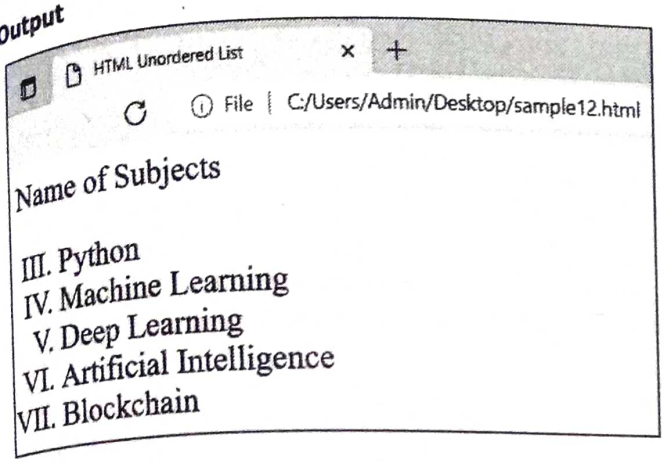


- The <ol> tag has attribute *TYPE* which is used to specify the type of number for the list items. The default type is 1 which is displayed in above output.
- There following options for the attribute *TYPE* of <ul> tag.
  - 1 - Number
  - I – Upper Roman
  - i – Lower Roman
  - A – Upper Alpha
  - a – Lower Alpha
- Program 1.9.4 illustrates the use of attribute *TYPE*.
- **The start Attribute :** This attribute of <ol> tag is used to specify the starting point of numbering.

Program 1.9.4 : Display the ordered list using roman type.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Unordered List</title>
  </head>
  <body>
<font size=5>
Name of Subjects
    <ol type="I" start="3">
      <li>Python</li>
      <li>Machine Learning</li>
      <li>Deep Learning</li>
      <li>Artificial Intelligence</li>
      <li>Blockchain</li>
    </ol>
</font>
</body>
</html>
```

Output



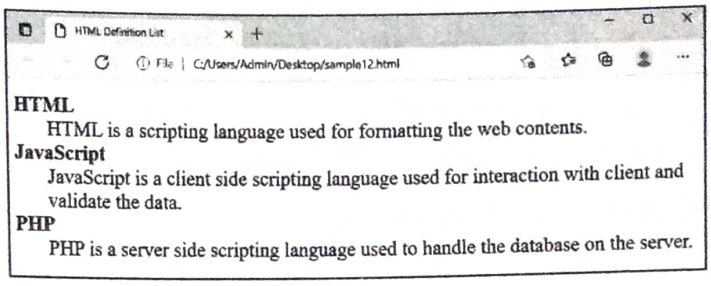
3. HTML Definition Lists

- Definition Lists are used to give the list of definitions just like the dictionary or encyclopedia. In general this is considered as the ideal way to represent a glossary, list of terms, or some kind of name/value pair list.
- Following tags are used in definition list.
  (1) **<dl>** – Defines the beginning of the list
  (2) **<dt>** – Specify a term
  (3) **<dd>** – Specify term definition
  (4) **</dl>** – Defines the end of the list

Program 1.9.5 : Write a script using the HTML definition list.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Definition List</title>
  </head>
  <body>
<font size=5>
    <dl>
      <dt><b>HTML</b></dt>
      <dd>HTML is a scripting language used for formatting
the web contents.</dd>
      <dt><b>JavaScript</b></dt>
      <dd>JavaScript is a client side scripting language used
for interaction with client and
validate the data.</dd>
<dt><b>PHP</b></dt>
<dd>PHP is a server side scripting language used to handle
the database on the server.</dd>
</font>
    </dl>
  </body>
</html>
```

Output



1.10 TABLES

GQ. How to create Tables on the web pages using HTML?

- HTML provides the ability to developers to add the data in table format so that it will help the visitors to understand the information quickly.
- HTML provides the <table> tag for adding the table into webpage. Let's see some information about this tag.

1.10.1 Table Tag with Attributes, TABLE, TR, TH, TD Tags

The <table> tag is used to present the data in tabular format, means in rows and columns. <table> is the main tag which is used to create the table.

Table shows the list of table related tags

Tag	Description
<table>	It defines a table.
<tr>	Table row - It defines a row in a table.
<th>	Table heading - It defines column headings in the table.
<td>	Table Data - It defines a cell in a table.
<caption>	It defines the caption of the table.
<colgroup>	It specifies a group of one or more columns in a table for formatting.
<col>	It is used in combination with <colgroup> element to specify column properties for each column.
<tbody>	It is used to group the body content in a table.
<thead>	It is used to group the header content in a table.
<tfooter>	It is used to group the footer content in a table.

1.10.2 Borders, Cell Spacing, Cell Padding, Width, Align, Bgcolor Attribute

Also some other attributes are provided to format the table such as specifying the color, border, height, width, etc.

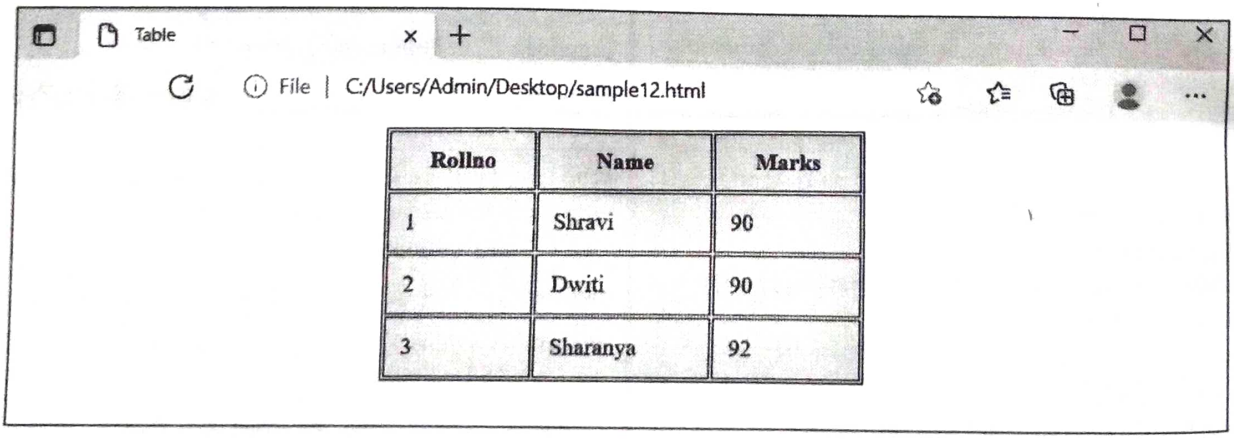
Attributes of <table> tag

Attribute	Value	Description
align	left center right	Specifies the alignment of a table with respect to surrounding text
bgcolor	rgb(x,x,x) #xxxxxx colorname	Specifies the color for a table background
border	Any number	Specifies size of border
cellpadding	pixels	Specifies the space between the cell wall and the cell content
cellspacing	pixels	Specifies the space between cells
width	pixels %	Specifies the width of a table
height	pixels %	Specifies the height of a table
Bordercolor	rgb(x,x,x) #xxxxxx colorname	Sets color of the table border

Program 1.10.1 : Write HTML code to display student data in table format.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Table</title>
  </head>
      <body>
<table border=1 bordercolor="red" width=40% height = 40% cellspacing=2 cellpadding=10 bgcolor="lightgray" align="center">
<tr>
<th>Rollno</th>
<th>Name</th>
<th>Marks</th>
</tr>
<tr>
<td>1</td>
<td>Shravi</td>
<td>90</td>
</tr>
<tr>
<td>2</td>
<td>Dwiti</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>Sharanya</td>
<td>92</td>
</tr>
</table>
</body>
</html>
```

Output



Rollno	Name	Marks
1	Shravi	90
2	Dwiti	90
3	Sharanya	92

1.10.3 Colspan and Rowspan Attributes

- A table is divided into rows and every row is divided into cells.
- Sometimes we require the table cells span across (merged) more than one column or row. Here we can use Colspan and Rowspan attributes.

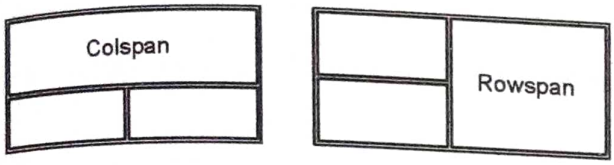


Fig. 1.10.1

(i) Colspan

- The colspan attribute is used to define the number of columns a cell should span (or merge) horizontally. That means, we like to merge two or more Cells in a row into a single Cell.

```
<td colspan=2 >
```

- This given code will merge two Cells into one Cell horizontally.

Before Colspan

First Cell	Second Cell
Third Cell	Forth Cell

After Colspan

Merged	
Third Cell	Forth Cell

Fig. 1.10.2

- In Fig. 1.10.2 there are two tables. In the first figure 2 rows and 2 columns in each row are present.
- In the second figure, first two cells are merged horizontally with the help of Colspan attribute.

Program 1.10.2 : Write a program to demonstrate use of colspan.

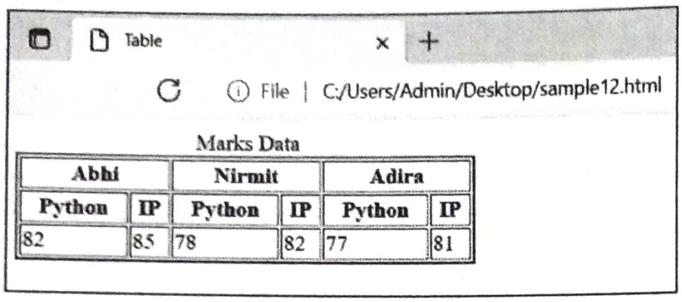
```
html>
<body >
  <table border=1 >
    <tr>
      <td colspan=2 >
        Merged
      </td>
    </tr>
    <tr>
      <td>
        Third Cell
      </td>
      <td>
        Forth Cell
      </td>
    </tr>
  </table>
</body>
</html>
```

- Colspan (Column Span) merges the Cells horizontally that means from left to right.
- The default value for the Colspan is 1.

Program 1.10.3 : Write a program to display the marks of students using colspan.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Table</title>
  </head>
  <body>
<table border=2 width="40%">
<caption>Marks Data</caption>
<tr>
<th colspan="2">Abhi</th>
<th colspan="2">Nirmit</th>
<th colspan="2">Adira</th>
</tr>
<tr>
<th>Python</th>
<th>IP</th>
<th>Python</th>
<th>IP</th>
<th>Python</th>
<th>IP</th>
</tr>
<tr>
<td>82</td>
<td>85</td>
<td>78</td>
<td>82</td>
<td>77</td>
<td>81</td>
</tr>
</table>
</body>
</html>
```

Output



(ii) Rowspan

- The rowspan attribute is used to define the number of rows a cell should span in vertical manner. That means two or more Cells in the same column are merged as a single Cell vertically.

```
<td rowspan=2 >
```

- This given code will merge two Cells into one Cell vertically.
- In Fig. 1.10.3, we can see there are two tables. In the first figure there are 2 rows and each row has 2 columns.
- In the second figure there are 2 rows in the first column and only 1 row in the second column. The Rowspan attribute is used vertically in the second column.

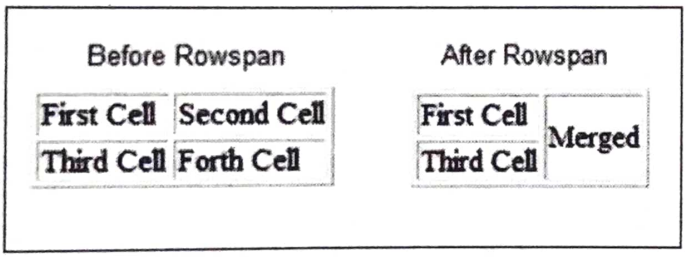


Fig. 1.10.3

Program 1.10.4 : Write a program to demonstrate use of rowspan.

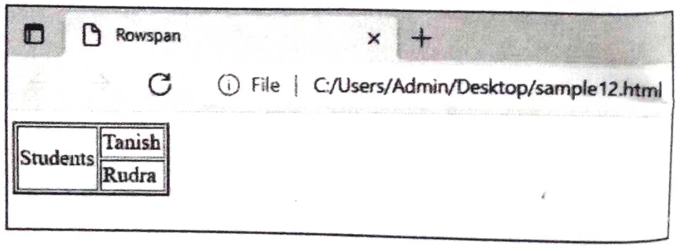
```
<html>
<body >
   <table border=1 >
      <tr>
         <td>
            First Cell
         </td>
         <td rowspan=2 >
            Merged
         </td>
      </tr>
      <tr>
         <td valign=middle>
            Third Cell
         </td>
      </tr>
   </table>
</body>
</html>
```

- Rowspan merged the Cells in vertical manner that means from top to bottom.

Program 1.10.5 : Write a program to merge the data in a row.

```
<html>
<head>
<title>
Rowspan
</title>
</head>
<body>
<table border=2>
   <tr>
      <td rowspan="2">
      Students
      </td>
      <td>
   Tanish
      </td>
   </tr>
   <tr>
      <td>
      Rudra
      </td>
   </tr>
</table>
</body>
</html>
```

Output



1.11 IMAGES

- Images are very important to improve the design and the appearance of a web page.
- In HTML, images are defined with the **<img>** tag.

- Inserting an image in web page is achieved by using <img> tag.

Syntax

```
<img src="Image_URL" />
```

- Here the **src** attribute specifies the web address of the image. The src attribute's value should contain the correct image file name with its extension. PNG, JPEG or GIF images can be displayed on webpage.

The alt attribute

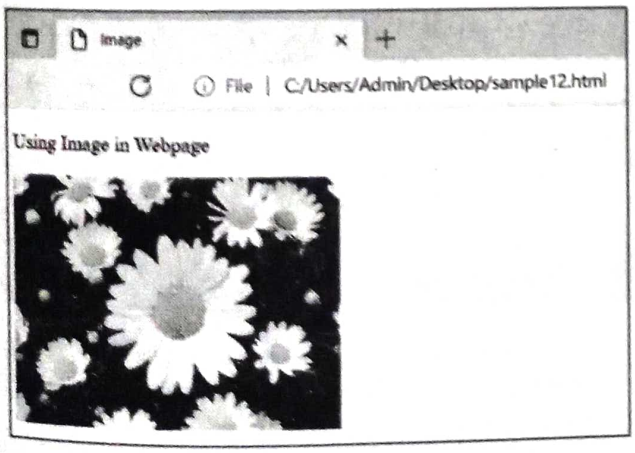
- The **alt** attribute specifies the alternate text for the image. If the image can't displayed on the webpage. So it is a mandatory attribute.
- The value of the alt attribute is a text which describes the image :

```
<img src=" Image_URL" alt="description_text">
```

Program 1.11.1 : Write a program to add an image in HTML document.

```
<!DOCTYPE html>
<html>
<head>
<title>Image </title>
</head>
<body>
<p>Using Image in Webpage</p>
<img src=" flower.jpg" alt="Flower" />
</body>
</html>
```

Output



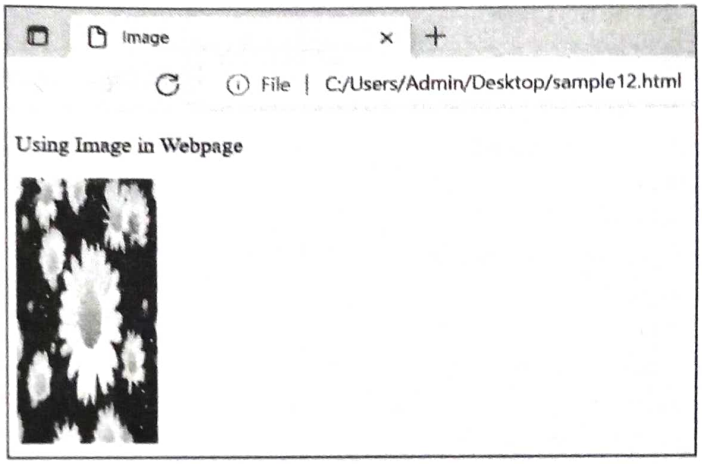
The width and height attributes

- The width and height attributes are used to set the size of the image to be displayed on the webpage.
- These attributes of <img> tag defines the width and height of the image in pixels.

Program 1.11.2 : Write program to set the size of image using width and height attributes of <img> tag.

```
<!DOCTYPE html>
<html>
<head>
<title>Image</title>
</head>
<body>
<p>Using Image in Webpage</p>
<img src=" flower.jpg " alt="Flower "
width="100"height="200">
</body>
</html>
```

Output



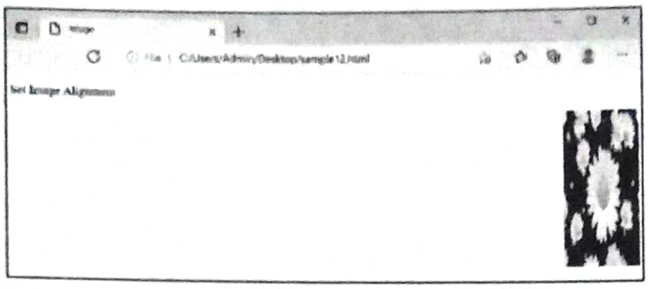
The align attribute

- The **align** attribute is used for setting the alignment of image to center or right, because by default, image will align at the left side of the webpage.
- Following Program shows how to align the image to right.

Program 1.11.3 : Write a program to set the alignment for image.

```
<!DOCTYPE html>
<html>
<head>
<title> Image </title>
</head>
<body>
<p>Set Image Alignment</p>
<img src=" flower.jpg " alt="Flower " width="100"
height="200" align="right">
</body>
</html>
```

Output



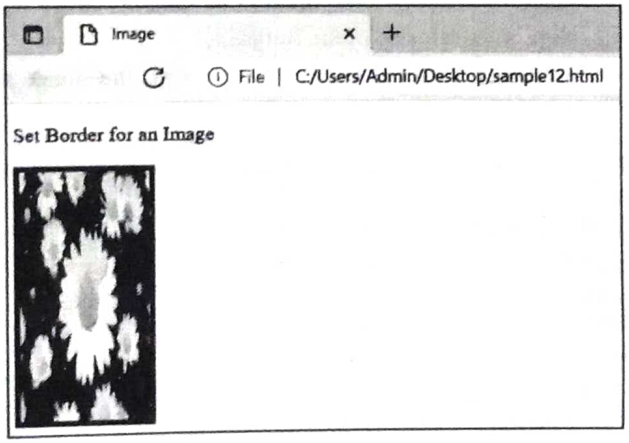
The border attribute

The border attribute has the value which specifies the border thickness in pixels. By default, image will have a border around it, and the 0 pixel thickness means no border around the image.

Program 1.11.4 : Write a program to add an image and set the border=4 pixel to it.

```
<!DOCTYPE html>
<html>
<head>
<title>Image</title>
</head>
<body>
<p>Set Border for an Image </p>
<img src=" flower.jpg " alt="Flower " width="100"
height="200" border="4">
</body>
</html>
```

Output



Inserting Images as a Page Background

By default the webpage background is white. HTML provides following two ways to beautify the webpage background.

- The **background** attribute is used with HTML elements to set the image as its background.

Syntax

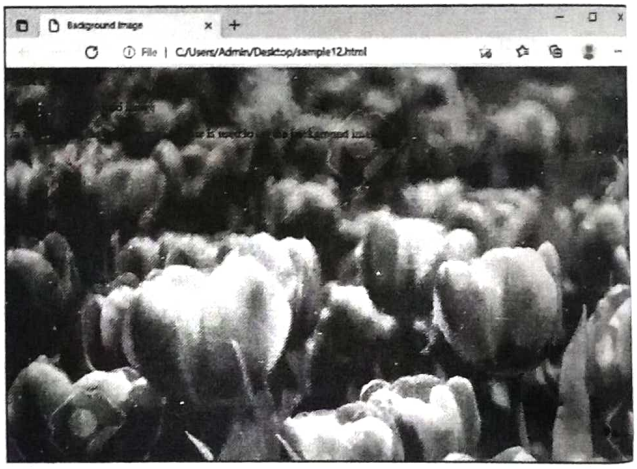
```
<tagname background="Image URL"...>
...
</tagname>
```

- For setting background the JPEG, GIF and PNG image formats are used.

Program 1.11.5 : Write a program to set the background image for body tag using background attribute.

```
!DOCTYPE html>
<html>
<head>
<title>Background Image</title>
</head>
<body background="tulip.jpg">
<p>Setting the background image </p>
<p>in the body tag the background attribute is used to set
the background image</p>
</body>
</html>
```

Output



- Also using the background-image property of the style attribute we can add the image as background of an HTML element :

```
<body style="background-image:url('tulip.jpg')">
```

- It will produce same result as shown above.
- It is possible to set the background image for other HTML elements such as paragraph or table.
- Following example shows how to set background image on the paragraph as shown below :

```
<body>
<p style="background-image:url('tulip.jpg')">
...
</p>
</body>
```

1.12 HTML FORM

GQ. Explain Form element in HTML.

- The HTML form is a part of a document which may have general content, special elements known as controls like text fields, checkboxes, submit button, radio buttons etc.
- Users usually "complete" a form by modifying its elements. This modification may include entering text, selecting controls like checkbox etc. before submitting the form to the next page for processing.
- This data is posted to an application like CGI, ASP Script or PHP script etc. This application will then perform essential processing on the accepted data based on defined business logics of the application.

Use

- HTML Forms are generally necessary when we want to accept some data from the website visitors. For example e-mail account form, credit card form where we may like to accept information like name, email address, phone number, gender etc.
- The HTML **<form>** tag creates an HTML form

Syntax

```
<form action = "Script URL" method = "GET/POST">
 form elements like input, button etc.
</form>
```

Tag attributes

Following is the list of form tag attributes

Sr. No	Attribute	Description
1.	Action	Url of next page(script) to which we want to submit the data
2.	Method	Method which will upload the data. The common methods used are GET and POST.
3.	target	Sets the name of the target window or frame where we want to display the result. It takes values like _blank, _self, _parent etc.
4.	enctype	Specify the way to encode the data before sending it to server

HTML Form Controls

HTML provides various controls which can be used to collect the data using HTML form

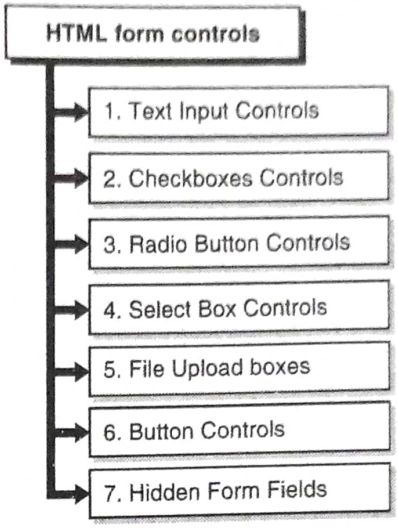


Fig. 1.12.1 : HTML form controls

1. Text Input Controls

Three types of text input controls are used on forms :

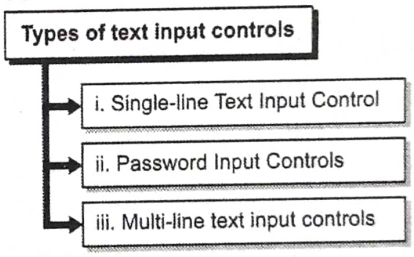


Fig. 1.12.2 : Types of text input controls

(i) Single-line Text Input Control

This control is used for one line of user input. It is created using HTML **<input>** tag.

```
<input type= "text" name= "txtUser"/>
```

(ii) Password Input Controls

This is too a single-line text input control except that it masks the character when user enters it. It is also created using the same <input> tag.

```
<input type= "password" name= "txtPass"/>
```

Attributes of input tag

Sr. No	Attribute	Description
1.	Type	Specifies the type of control.
2.	Name	Used to assign a name to the control which is sent to the server to be recognized and retrieve the value.
3.	Value	This is used to set default value to the control.
4.	size	Sets the width of the control in terms of characters.
5.	maxlength	Specifies the maximum number of characters which user can give in the text box.

(iii) Multi-line text input controls

This is used to accept detailed information from user. It is created using HTML **<textarea>** tag.

```
<textarea rows=5 cols=30/>
```

Attributes of <textarea> tag

Sr. No	Attribute	Description
1.	name	Used to assign a name to the control which is sent to the server to be recognized and retrieve the value.
2.	Rows	Specifies the number of rows of text area.
3.	Cols	Specifies the number of columns of text area.

2. Checkbox Control

This control is used to give list of items where multi-selection is allowed. It is also created using <input> tag but the value for type attribute is set as **checkbox.**

```
<input type= "checkbox" name= "chk"/>
```

3. Radio Button Control

This control is used to give list of items where single-selection is allowed. It is also created using <input> tag but where the value for type attribute is set as **radio.**

```
<input type= "radio" name= "opt"/>
```

4. Select Box Control

A select box is also known as drop down box. It provides option to list number of options in the form of drop down list. Here the user can select one or multiple options.

```
<select name= "sel">
<option>--</option>
-----
</select>
```

Attributes of <select> tag

Sr. No	Attribute	Description
1.	name	Used to assign a name to the control which is sent to the server to be recognized and retrieve the value.
2.	size	This can be used to specify a scrolling list box.
3.	multiple	If this option is set, it creates listbox and user can select multiple elements. If not then combobox is created where single selection is allowed.

Attributes of <option> tag

Sr. No	Attribute	Description
1.	value	The value that will be used to check whether the option is selected or not.
2.	Selected	It sets the option by default selected
3.	Label	Sets label to options

5. File Upload Box

This is used to give option to user for file upload. This is created using <input> tag. The type attribute is set to **file**.

Attributes of file upload box

Sr. No	Attribute	Description
1.	accept	Specifies the types of files that the server accepts.

6. Button Controls

HTML provides different types of buttons. The <input> tag is used to create buttons.

Types of Buttons

Sr. No	Type	Description
1.	Submit	This button submits a form.
2.	Reset	This button resets (clears) form controls.
3.	Button	This button calls a client-side script (function).

7. Hidden Form Fields

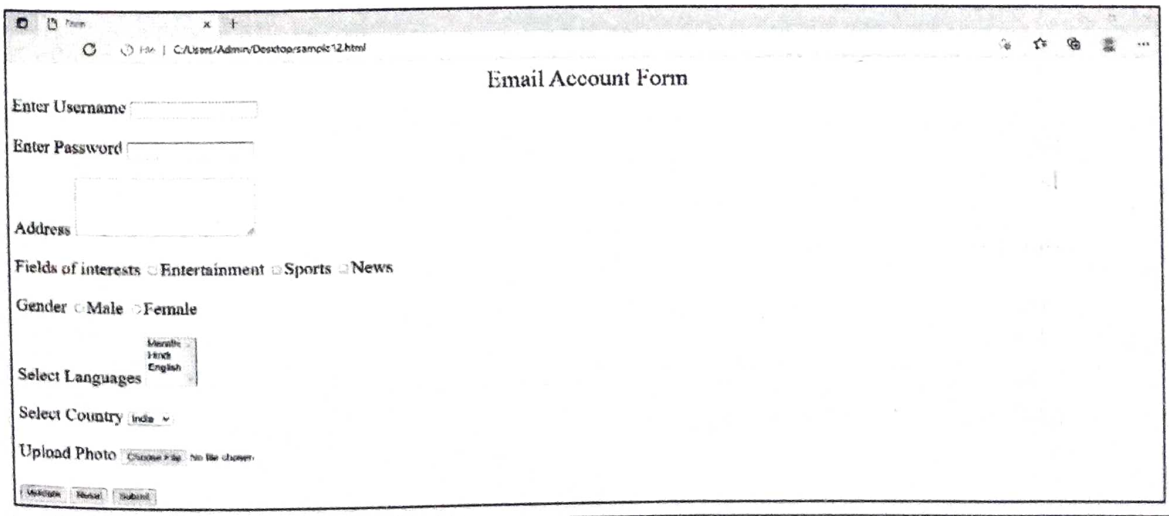
This element is used to hide the data while sending it to the server. This control as name suggests does not display on the form.

Program 1.12.1 : Write a program to display the form containing all controls.

Program displaying the form containing all records

```
<html>
<head>
<title>
Form
</title>
</head>
<body>
<font size=6>
<center>
Email Account Form
</center>
</font>
<font size=5>
Enter Username <input type="text" name ="txtUser"
maxlength=30/><br><br>
Enter Password <input type="password" name
="txtPass"/><br><br>
Address <textarea name="add" rows=5
cols=30></textarea><br><br>
Fields of interests
<input type="checkbox" name ="chk1"/>Entertainment
<input type="checkbox" name ="chk2"/>Sports
<input type="checkbox" name ="chk3"/>News<br><br>
Gender
<input type="radio" name ="optGender"/>Male
<input type="radio" name
="optGender"/>Female<br><br>
Select Languages
<select name="lang" multiple>
<option>Marathi</option>
<option>Hindi</option>
<option>English</option>
</select>
<br><br>
Select Country
<select name="cntry">
<option>India</option>
<option>USA</option>
<option>Japan</option>
<option>UK</option>
</select>
<br><br>
Upload Photo
<input type = "file" name = "fileupload" accept = "image/*"
/>
<br><br>
<input type="button" name = "vld" value="Validate">
<input type="reset" name = "rst" value="Reset">
<input type="submit" name = "sub" value="Submit">
<input type = "hidden" name = "pagenm" value = "11" />
</font>
</body>
</html>
```

Output



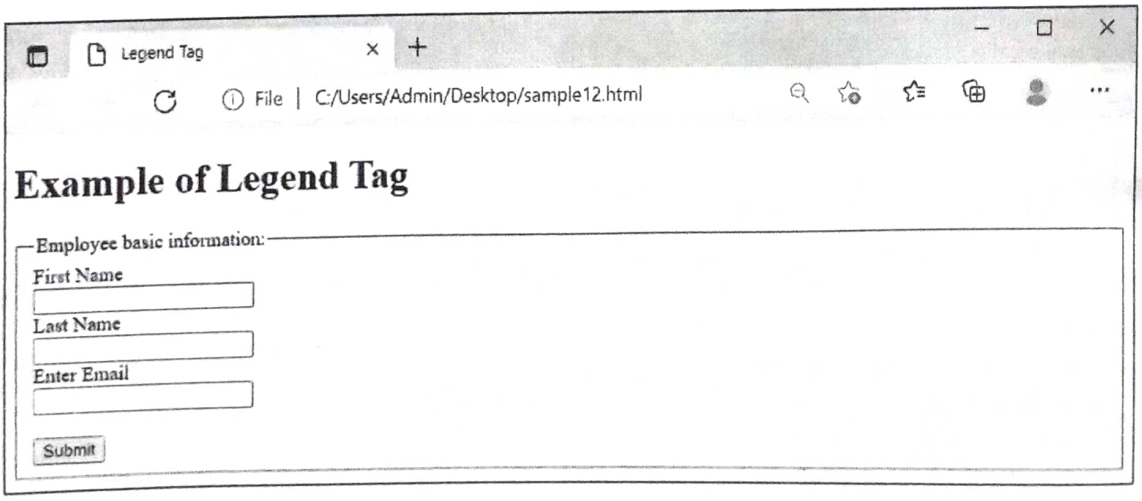
HTML <fieldset >and <legend> tag

- HTML <legend> tag is used to insert a title or caption to its parent element such as <fieldset>. The <legend> element must be the first child of <fieldset > element.
- By using <legend> tag with <form> elements, it is easy to understand the purpose of grouped form elements.

Program 1.12.2

```
<!DOCTYPE html>
<html>
<head>
<title>Legend Tag</title>
</head>
<body>
 <h1>Example of Legend Tag</h1>
<form>
        <fieldset>
    <legend>Employee basic information:</legend>
    <label>First Name</label><br>
    <input type="text" name="fname"><br>
    <label>Last Name</label><br>
    <input type="text" name="lname"><br>
    <label>Enter Email</label><br>
    <input type="email" name="email"><br><br>
    <input type="Submit"><br>
        </fieldset>
</form>
</body>
</html>
```

Output



1.13 HTML5 CONTROLS

GQ. Explain semantic elements in HTML% with an example.

1.13.1 Semantic Elements in HTML5

- In HTML4, developers have to use their own id/class names to style elements: header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, etc.
- This is so difficult for search engines to identify the correct web page content. Now in HTML5 elements (<header> <footer> <nav> <section> <article>), this will become easier. It now allows data to be shared and reused across applications, enterprises, and communities.
- Semantic elements can increase the accessibility of your website, and also helps to create a better website structure.

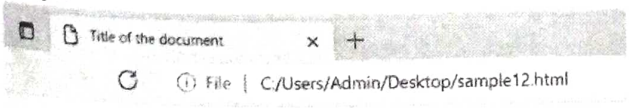
A) HTML5 <article> Element

HTML <article> element defines article content within a document, page, application, or a website. It can be used to represent a forum post, a magazine, a newspaper article, or a big story.

Program 1.13.1

```
<!DOCTYPE html>
<html>
 <head>
  <title>Title of the document</title>
 </head>
 <body>
  <article>
   <h1>Title of the article</h1>
   <p>Text of the article</p>
  </article>
 </body>
</html>
```

Output



Title of the article

Text of the article

B) HTML5 <aside> Element

The <aside> element represent the content which is indirectly giving information to the main content of the page. It is frequently represented as a sidebar.

Program 1.13.2

```
<!DOCTYPE html>
<html>
 <head>
  <title>Title of the document</title>
 </head>
 <body>
  <h2>My last year memories</h2>
  <p>I have visited Paris with my friends last month. This
was the memorable journey and i wish to go there again.</p>
   <aside>
    <h4>Paris</h4>
    <p>Paris, France's capital, is a major European city and
a global center for art, fashion, gastronomy and culture</p>
   </aside>
   </body>
</html>
```

Output

My last year memories

I have visited Paris with my friends last month. This was the memorable journey and i wish to go there again.

Paris

Paris, France's capital, is a major European city and a global center for art, fashion, gastronomy and culture

C) HTML5 <section> Element

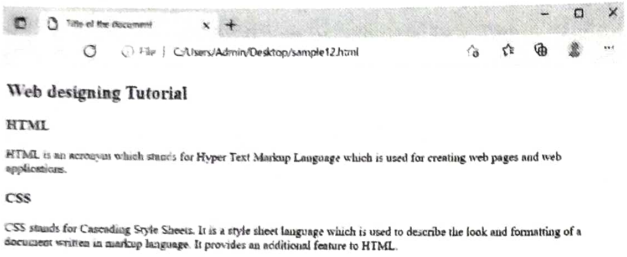
The <section> element is used to represent the standalone section within an HTML document. A page can have various sections and each section can contain any content, but headings for each section is not mandatory.

Program 1.13.3

```
<!DOCTYPE html>
<html>
 <head>
  <title>Title of the document</title>
 </head>
 <body>
  <h2>Web designing Tutorial</h2>
<section>
  <h3>HTML</h3>
```

```
  <p>HTML is an acronym which stands for Hyper Text Markup Language which is used for creating web pages and web applications.</p>
 </section>
 <section>
  <h3>CSS</h3>
  <p>CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML.</p>
 </section>
  </body>
</html>
```

Output



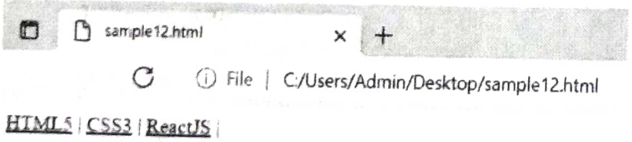
D) HTML5 <nav> Element

The HTML <nav> element is used to define a set of navigation links.

Program 1.13.4

```
<!DOCTYPE html>
<html>
<body>
<nav>
 <a href="HTML.html">HTML5</a> |
 <a href="CSS.html">CSS3</a> |
 <a href="React.html">ReactJS</a> |
</nav>
</body>
  </html>
```

Output



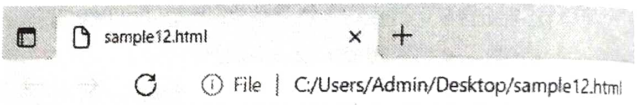
E) HTML5 <header> Element

The <header> element represent the header of the document which can contain introductory content or navigation links.

Program 1.13.5

```
<!DOCTYPE html>
<html>
<body>
  <header>
    <h1>Welcome to HTML5</h1>
    <nav>
      <ul>
        <li>Home </li>
        <li>About us </li>
        <li>Contact us</li>
      </ul>
    </nav>
  </header>
</body>
</html>
```

Output



Welcome to HTML5

- Home
- About us
- Contact us

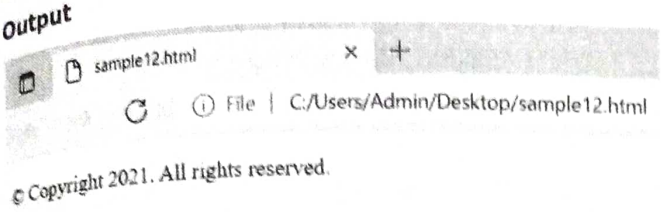
F) HTML5 <footer> Element

The <footer> tag defines the footer of an HTML document or page.

Program 1.13.6

```
<!DOCTYPE html>
<html>
<body>
  <footer>
    <p>© Copyright 2021. All rights reserved. </p>
  </footer>
</body>
</html>
```

Output



G) HTML5 <figure> and <figcaption> tag

- The <figcaption> element is used to provide a caption to an image.
- It is an optional tag and can appear before or after the content within the <figure> tag.
- Only one <figcaption> element can be nested within a <figure> tag although the <figure> element itself may contain multiple other elements like <img> or <code>.
- The <figcaption> element is used with <figure> element and it can be placed as the first or last child of the <figure> element.

Program 1.13.7

```
<!DOCTYPE html>
<html>
<body>
  <figure>
    <img src="tajmahal.jpg" alt="Taj Mahal"/>
   <figcaption>Fig.1.1 - A front view of the great Taj Mahal in Agra.</figcaption>
   </figure>
</body>
</html>
```

Output

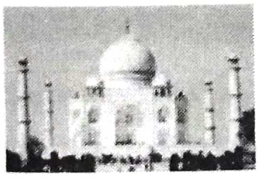


Fig. P. 1.13.7 : A front view of the great Taj Mahal in Agra

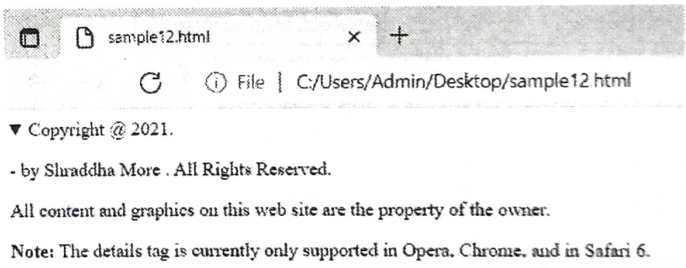
H) HTML <details> Tag

- HTML <details> tag is used to specify the additional details on the web page that the user can view or hide on demand.
- it is used as a disclosure widget from which user can retrieve additional information or control.
- It is used together with a relevant tag known as <summary>. Technically, there is no need of summary tag, but if you ignore this then the browser will use some default text.
- The <details> tag wraps all the content which you want to show or hide and the <summary> tag contains the summary and the title of the section.

Program 1.13.8

```
<!DOCTYPE html>
<html>
<body>
  <details>
     <summary>Copyright @ 2021.</summary>
     <p> - by Shraddha More . All Rights Reserved.</p>
     <p>All content and graphics on this web site are the property of the owner.</p>
     </details>
     <p><b>Note:</b> The details tag is currently only supported in Opera, Chrome, and in Safari 6.</p>
</body>
</html>
```

Output



☞ List of Semantic/Structural elements

Controls	Description
<article>	Defines an article.
<aside>	Defines content aside from the page content.
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside of it.
<command>	Defines a command button that a user can invoke.
<details>	Specifies additional details that the user can view or hide on demand.
<figcaption>	Defines a caption for a <figure> element.

Controls	Description
<figure>	Specifies self-contained content.
footer>	Defines a footer for a document or section.
<header>	Defines a header for a document or section.
<hgroup>	Group heading (<h1> to <h6>) elements.
<mark>	Defines marked/highlighted text.
<meter>	Defines a scalar measurement within a known range, or a fractional value.
<nav>	Defines navigation links.
<output>	Represents the result of a calculation.
<progress>	Represents the progress of a task.
<rp>	Defines what to show in browsers that do not support ruby annotations (for East Asian typography).
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography).
<ruby>	Defines a ruby annotation (for East Asian typography).
<section>	Defines a section in a document.
<summary>	Defines a visible heading for the <details> element. The heading can be clicked to view/hide the details.
<time>	Defines a date/time.

1.13.2 Media Elements

- HTML5 provides the feature of native audio and video support without the need of Flash.
- For this purpose HTML5 provide tags such as <audio> and <video> which adds media to a website.
- These tags contain **src** attribute to identify the media source and **controls** attribute so that user will be able to play and pause the media.

A) Embedding Audio

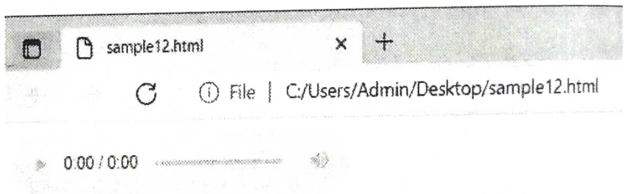
- HTML5 provides <audio> tag which helps to embed content related to sound in an HTML or XHTML document as follows.

```
<audio src="foo.wav" controls autoplay>
  Your browser does not support the <audio> element.
</audio>
```

- The specification of current HTML5 draft does not clearly mention the allowed audio formats browsers should support in the audio tag.
- But frequently preferred audio formats are **ogg, mp3** and **wav**.
- The <source> tag is used to specify media along with media type and various attributes.
- Multiple source elements are allowed in an audio element and the first recognized format will be used by the browser.

```
<!DOCTYPE HTML>
<html>
  <body>
    <audio controls autoplay>
      <source src="/songs/s1.ogg" type="audio/ogg" />
      <source src="/songs/s1.wav" type="audio/wav" />
    </audio>
  </body>
</html>
```

Output



Audio Attribute Specification

- The HTML5 audio tag provides following attributes to control the look and feel as well as several functionalities of the control :

Attribute	Description
autoplay	It is of type Boolean. If mentioned, the audio will automatically start to play back as soon as possible without waiting to finish loading the data.
autobuffer	It is of type Boolean. If mentioned, the audio will automatically start buffering even if it's not set to automatically play.
controls	It allows the user to control audio playback, including volume, seeking, and pause/resume playback.

Attribute	Description
loop	It is of type Boolean. If mentioned, will allow audio to automatically starts again when reach to the end.
preload	If this attribute is mentioned, then the audio will be loaded at page load, and ready to run. It will be ignored in case autoplay is present.
src	The URL of the audio to embed.

B) Embedding Video

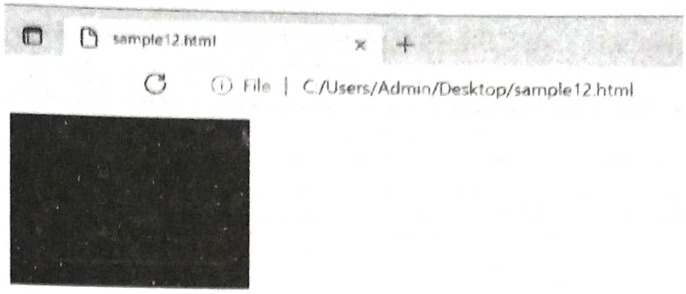
- HTML5 provides <video> tag which helps to embed content related to video in an HTML or XHTML document as follows.

```
<video src="s1.mp4"  width="200" height="150" controls>
</video>
```

- The specification of current HTML5 draft does not clearly mention the allowed video formats browsers should support in the audio tag.
- But frequently preferred audio formats are **ogg,** and **mpeg4**.
- The <source> tag is used to specify media along with media type and various attributes.
- Multiple source elements are allowed in video element and the first recognized format will be used by the browser.

```
<!DOCTYPE HTML>
<html>
  <body>
    <video  width="200" height="150" controls autoplay>
      <source src="/vsong/s1.ogg" type="video/ogg" />
      <source src="/vsong/s1.mp4" type="video/mp4" />
    </video>
  </body>
</html>
```

Output



1.13.3 HTML5 New <input> types

Controls	Description
color	Used for input fields that should contain a color.
date	Used for input fields that should contain a date.
datetime	Allows the user to select a date and time (with time zone).
datetime-local	Allows the user to select a date and time (no time zone).
email	Used for input fields that should contain an e-mail address.
month	Allows the user to select a month and year.
number	Used for input fields that should contain a numeric value.
range	Used for input fields that should contain a value within a range.
search	Used for search fields (a search field behaves like a regular text field).
tel	Used for input fields that should contain a telephone number.
time	Allows the user to select a time (no time zone).
url	Used for input fields that should contain a URL address.
week	Allows the user to select a week and year.

Syllabus Topic : CSS: Why CSS, Types of CSS, How to use CSS, Properties, Classes, Child-Class (Nested CSS), Colors, Text, Background, Border, Margin, Padding, Positioning (flex, grid, inline, block), Animation, Transition

1.14 CSS (CASCADING STYLE SHEET)

GQ. What is CSS?

- CSS stands for Cascading Style Sheets.
- It is a style sheet language which is used to describe the look and formatting of a document written in markup language.
- It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces.
- CSS is used to manage the look and feel of the web pages. CSS is used to control the color of the text, the style of fonts, the way columns are sized and laid out, different background images or colors used, the spacing between paragraphs, layout designs, and variations in the display for various devices, screen sizes or resolutions and variety of other effects.
- The CSS is very easy to learn and understand but it has strong control over the presentation of web page. In general, CSS is integrated with the markup languages like HTML or XHTML.
- In simple words, CSS is nothing but declaration of style sheets which can be repeatedly used. It resembles to the function concept of C language. That means define once and use anytime anywhere repeatedly.

1.14.1 Advantages of CSS

GQ. State the advantages of CSS.

1. **CSS Saves Time**

   Once written, CSS can be used in multiple HTML documents repeatedly. It helps to define style for different HTML elements and apply it to multiple web pages as per requirement.

2. **Pages Load Faster**

   When CSS is used, there is no need to write HTML tag attributes repeatedly. Just write only once and can be applied to all the occurrences of that tag in the webpage. As code is less, it speed up the download process.

3. **Easy Maintenance**

   Just change in the style sheet effects all the elements in different web pages.

4. **Superior Styles to HTML**

   CSS has a wider range of attributes which helps to make the look of webpage far better than HTML.

5. **Multiple Device Compatibility**

   Style sheet contents are compatible for multiple types of devices. With use of same HTML document, we can present various versions of a website.

6. **Offline Browsing**

   CSS helps to store the web applications on local machines by using an offline cache. This helps to view offline websites. The cache provides fast loading of the website and improved overall performance.

7. **Platform Independence**

   The CSS supports reliable platform independence and also supports all the latest browsers.

1.14.2 CSS Syntax

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document.

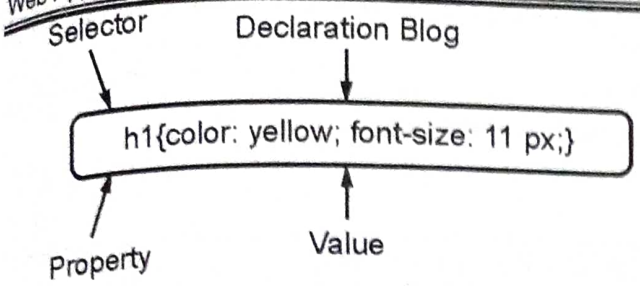
A CSS rule set contains a selector and a declaration block.

A style rule is made of three parts:

1. **Selector :** A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
2. **Property :** A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.
3. **Value :** Values are assigned to properties. For example, color property can have value either red or #F1F1F1 etc.

CSS Style Rule Syntax

```
Selector{ property: value } OR
Selector{Property1: value1; Property2: value2; .....;}
```



(1P2)**Fig. 1.14.1**

Example

table {border :1px solid #C00;}

Here table is a selector and border are a property and given value 1px solid #C00 is the value of that property.

1.15 TYPES OF CSS

GQ. What are the types of CSS? Explain with example.

There are three ways of inserting a style sheet in the webpage :

1. Inline Style Sheet
2. Internal Style Sheet
3. External Style Sheet

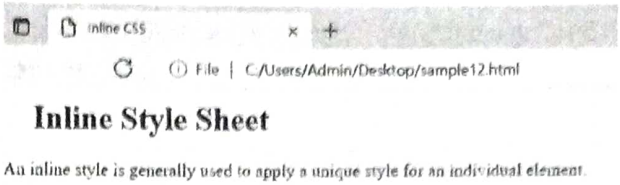
1. Inline Style Sheet

- An inline style is generally used to apply a unique style for an individual element.
- The Inline style is specific for the individual tag. The HTML "style" attribute is used by the inline style to style a particular tag. The Inline style generally useful for an individual CSS change which we do not want to use repeatedly in the site.

Program 1.15.1

```
<html>
<head>
<title>
Inline CSS
</title>
</head>
<body>
<h1 style="color:blue;margin-left:30px;">SJCEM</h1>
<p style="color:red;font-size:18px"> An inline style is
generally used to apply a unique style for an individual
element.
</p>
</body>
</html>
```

Output



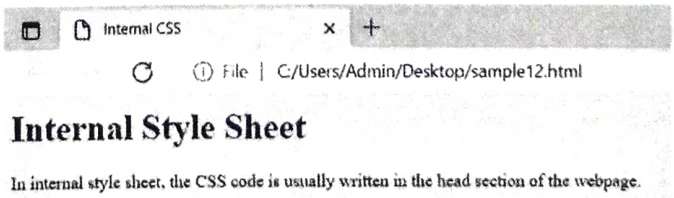
2. Internal Style Sheet

- In internal style sheet, the CSS code is usually written in the head section of the webpage. This simplifies the applications of styles like classes or id's in contrast to repeated use of the code.
- While creating an internal style sheet in the web page, we have to use the <style></style> HTML tags in the Head section of the webpage. The entire code of the Internal CSS style sheet is included between the <head></head> section of the websites.
- Following is an example of how an internal style sheet looks like.

Program 1.15.2

```
<html>
<head>
<title>
Internal CSS
</title>
<style type="text/css">
body {background-color: linen}
h1 {color:blue;margin-left:30px>
p {color:red;font-size:18px>
</style>
</head>
<body>
<h1>Internal Style Sheet</h1>
<p>In internal style sheet, the CSS code is usually written in
the head section of the webpage.</p>
</body>
</html>
```

Output



3. External Style Sheet

- The External Style sheet is a file with extension .css which we link to website. This CSS file is used to declare the style sheets. Whenever any change is made in this css file, it gets reflected in all the web pages of the website. This simplifies our work and avoids making change in each and every page where the css is used.
- Now a day most of the websites use external style sheets. These are the styles which are written in a separate document or file and then linked to various web pages. External style sheets affects all the documents which are attached to, this means that if we have a 100-page website where all the pages use the same style sheet, just change in the style sheets will make visual change in all the web pages which are linked to it.

Program 1.15.3

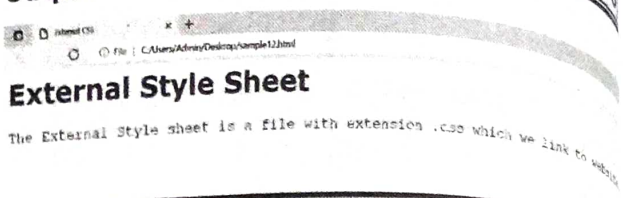
CSS file: style.css

```
h1 {
   color: blue;
   font-family: verdana;
   font-size: 300%;
}
p {
   color: red;
   font-family: courier;
   font-size: 160%;
}
```

HTML file: style.html

```
<html>
<head>
<title>
External CSS
</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<h1>External Style Sheet </h1>
<p> The External Style sheet is a file with extension .css
which we link to website.</p>
</body>
</html>
```

Output



1.16 CSS SELECTOR

CSS selectors are used to select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type attribute etc.

There are several different types of selectors in CSS :

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

1. CSS Element Selector

The element selector selects the HTML element by name.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<p>This style will be applied on every paragraph. </p>
<p id="para1">Hello </p>
<p>Welcome </ p>
</body>
</html>
```

2. CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element.

Example with the id "para1".

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<p id="para1">Hello </p>
<p>This paragraph will not be affected. </p>
</body>
</html>
```

3. CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period (.) followed by the class name.

Example with a class "center".

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is blue and center-aligned. </h1>
<p class="center">This paragraph is blue and center-aligned. </p>
</body>
</html>
```

CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is not affected</h1>
<p class="center">This paragraph is blue and center-aligned. </p>
</body>
</html>
```

4. CSS Universal Selector

The universal selector is used as a wildcard character. It selects all the elements on the pages.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  color: green;
  font-size: 20px;
}
</style>
</head>
<body>
<h2>This is heading</h2>
<p>This style will be applied on every paragraph. </p>
<p id="para1">Hello World! </p>
<p>Welcome to CSS!! </p>
</body>
</html>
```

5. CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

Example

```
h1 {
   text-align: center;
   color: blue;
}
h2 {
   text-align: center;
   color: blue;
}
p {
   text-align: center;
   color: blue;
}
```

As you can see, you need to define CSS properties for all the elements. It can be grouped in following ways:

```
h1, h2, p {
   text-align: center;
   color: blue;
}
```

Example of CSS group selector

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
   text-align: center;
   color: blue;
}
</style>
</head>
<body>
<h1>Hello SJCEM</h1>
<h2>Hello World! </h2>
<p>This is a paragraph. </p>
</body>
</html>
```

1.17 CSS COMMENTS

- CSS comments are generally written to explain your code. It is very helpful for the users who reads your code so that they can easily understand the code.
- Comments are ignored by browsers.
- Comments are single or multiple lines statement and written within /*.....*/ .

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
   color: blue;
   /* This is a single-line comment */
   text-align: center;
}
/* This is
a multi-line
comment */
</style>
</head>
<body>
<p>Hello Javatpoint.com</p>
<p>This statement is styled with CSS. </p>
<p>CSS comments are ignored by the browsers and not shown in the output. </p>
</body>
</html>
```

1.17.1 CSS Colors

- The color property in CSS is used to set the color of HTML elements. Typically, this property is used to set the background color or the font color of an element.
- In CSS, we use color values for specifying the color. We can also use this property for the border-color and other decorative effects.
- We can define the color of an element by using the following ways :

1. RGB format.
2. RGBA format.
3. Hexadecimal notation.
4. HSL.
5. HSLA.
6. Built-in Color

1. RGB Format

- RGB format is the short form of 'RED GREEN and BLUE' that is used for defining the color of an HTML element simply by specifying the values of R, G, B that are in the range of 0 to 255.

- The color values in this format are specified by using the rgb() property. This property allows three values that can either be in percentage or integer (range from 0 to 255).
- This property is not supported in all browsers; that's why it is not recommended to use it.

Syntax

```
color: rgb (R, G, B);
```

2. RGBA Format

- It is almost similar to RGB format except that RGBA contains A (Alpha) that specifies the element's transparency.
- The value of alpha is in the range 0.0 to 1.0, in which 0.0 is for fully transparent, and 1.0 is for not transparent.

Syntax

```
color: rgba (R, G, B, A);
```

3. Hexadecimal notation

- Hexadecimal can be defined as a six-digit color representation. This notation starts with the # symbol followed by six characters ranges from 0 to F. In hexadecimal notation, the first two digits represent the red (RR) color value, the next two digits represent the green (GG) color value, and the last two digits represent the blue (BB) color value.
- The black color notation in hexadecimal is #000000, and the white color notation in hexadecimal is #FFFFFF. Some of the codes in hexadecimal notation are #FF0000, #00FF00, #0000FF, #FFFF00, and many more.

Syntax

```
color: #(0-F) (0-F) (0-F) (0-F) (0-F) (0-F);
```

Short Hex codes

It is a short form of hexadecimal notation in which every digit is recreated to arrive at an equivalent hexadecimal value.

For example, #7B6 becomes #77BB66 in hexadecimal.

The black color notation in short hex is #000, and the white color notation in short hex is #FFF. Some of the codes in short hex are #F00, #0F0, #0FF, #FF0, and many more.

4. HSL

It is a short form of Hue, Saturation, and Lightness.

- **Hue :** It can be defined as the degree on the color wheel from 0 to 360. 0 represents red, 120 represents green, 240 represents blue.
- **Saturation :** It takes value in percentage in which 100% represents fully saturated, i.e., no shades of gray, 50% represent 50% gray, but the color is still visible, and 0% represents fully unsaturated, i.e., completely gray, and the color is invisible.
- **Lightness :** The lightness of the color can be defined as the light that we want to provide the color in which 0% represents black (there is no light), 50% represents neither dark nor light, and 100% represents white (full lightness).

Syntax

```
color: hsl (H, S, L);
```

5. HSLA

It is entirely similar to HSL property, except that it contains A (alpha) that specifies the element's transparency. The value of alpha is in the range 0.0 to 1.0, in which 0.0 indicates fully transparent, and 1.0 indicates not transparent.

Syntax

```
color: hsla (H, S, L, A);
```

6. Built-in Color

As its name implies, built-in color means the collection of previously defined colors that are used by using a name such as red, blue, green, etc.

Syntax

```
color: color-name;
```

Example

```
<html>
  <head>
    <title>CSS hsl color property</title>
    <style>
      h1{
        text-align:center;
      }
      #rgb{
       color:rgb(255,0,0);
      }
      #rgba{
       color:rgba(255,0,0,0.5);
      }
```

```
      #hex{
       color:#EE82EE;
      }
      #short{
       color: #E8E;
      }
      #hsl{
       color:hsl(0,50%,50%);
      }
      #hsla{
       color:hsla(0,50%,50%,0.5);
      }
      #built{
       color:green;
      }
    </style>
  </head>
  <body>
    <h1 id="rgb">
      Hello World. This is RGB format.
    </h1>
    <h1 id="rgba">
      Hello World. This is RGBA format.
    </h1>
    <h1 id="hex">
    Hello World. This is Hexadecimal format.
  </h1>
  <h1 id="short">
  Hello World. This is Short-hexadecimal format.
  </h1>
  <h1 id="hsl">
  Hello World. This is HSL format.
  </h1>
  <h1 id="hsla">
  Hello World. This is HSLA format.
  </h1>
  <h1 id="built">
  Hello World. This is Built-in color format.
  </h1>
    </body>
  </html>
```

1.18 CSS PROPERTIES

1.18.1 Text

GQ. Write note on manipulating text in CSS with example.

Text Properties are given as below:

Property	Description
letter-spacing	Used for incrementing or decrementing the letter-spacing between characters in a text
line-break	Used to break the lines
line-height	Used for setting the height of line
tab-size	Used to specify the tab-character length
text-align	Used to specify the horizontal-text-alignment
text-indent	Used to specify the text-indentation of the first line of text block.
text-justify	Used to specify text-align is "justify"
text-transform	Used to Control the text-capitalization
white-space	Used to handle the white-space within an element
Hyphens	Used to separate the words
hanging-punctuation	Used to specify whether a punctuation character can be placed outer of the line box
overflow-wrap	Used to handle the overflow by breaking lines within words when a string is large enough and can't fit in given box.

Property	Description
word-break	Used to Specify the rule for line breaking in non-CJK scripts
word-spacing	Used to specify whether to increase or decrease the space between words in a text
word-wrap	Used to adjust the words so as to handle the overflow by breaking the long and unbreakable words to wrap them into next line

☞ Text decoration properties

To decorate the text following Text Decoration Properties are used.

Property	Description
text-decoration	Used to specify the decoration applied on the text.
text-decoration-color	Used to specify the color for decorating the text.
text-decoration-line	Used to specify the line-type in a text-decoration
text-decoration-style	Used to specify the style for decorating the text.
text-shadow	Used to add shadow to text
text-underline-position	Used to specify the position of the underline set by text-decoration property.

Following are the CSS properties used to manipulate text:

1. Color

This property is used for setting the color of a text. We have studied this property in previous section.

2. Direction

- This property is used for setting the text direction.
- The values of this property are ltr or rtl means left to right or right to left respectively.

3. Letter-spacing

- This property is used to insert or remove spaces between the letters of a word.
- The values of this property are normal or a number specifying the space.
- The following Program 1.18.2 shows how to set the space between characters.

4. Word-spacing

- This property is used to insert or remove spaces between the words of a sentence.
- The values of this property are normal or a number specifying space
- The following Program 1.18.3 shows how to set the space between words.

5. Text-indent

- This property is used to indent the text of a paragraph.
- The values of this property are % or a number specifying indent space. Program 1.18.4 shows how to indent the first line of a paragraph.

6. Text-align

- This property is used to align the text of a document.
- The values of this property are left, right, center, justify.

7. Text-decoration

- This property is used to decorate the text using underline, over-line, and strikethrough.
- The values of this property are none, underline, over-line, line-through, and blink.

8. Text-transform

- This property is used to set the text cases to uppercase or lowercase letters.
- The values of this property are none, capitalize, uppercase, lowercase.

9. White-space

- This property is used handling the white space present inside an element.
- The values of this property are normal, pre, nowrap.
- The following Program 1.18.8 shows whether the white-space inside the element affects the text or not.

10. Text-shadow

- This property is used to set the text shadow around a text.
- This property is supported by few browsers.
- For example the text-shadow property has following format :

```
text-shadow:4px 4px 8px blue
```

- Where first two values indicate the position of shadow from the text, and the third value indicate the opacity of shadow. 1px indicates dark shadow. And the fourth value indicates the shadow color.

Program 1.18.1 : Write a code to demonstrate use of direction property.

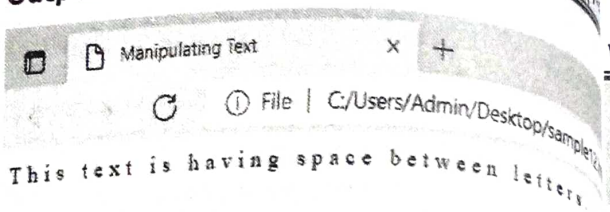
```
<html>
  <head>
    <title>Manipulating Text</title>
    <style>
        p { direction: rtl; }
    </style>
  </head>
  <body>
    <p>
    This text will be rendered from right to left
    </p>
  </body>
</html>
```

Output

Program 1.18.2 : Write a code to demonstrate the use of letter-spacing property.

```
<html>
  <head>
    <title>Manipulating Text</title>
    <style>
        p { letter-spacing:5px; }
    </style>
  </head>
  <body>
    <p>
    This text is having space between letters.
    </p>
  </body>
</html>
```

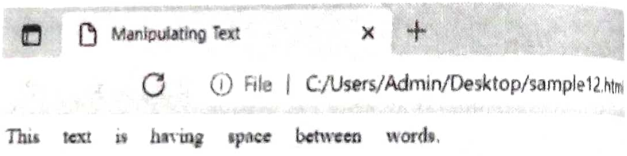
Output



Program 1.18.3 : Write a code to demonstrate the use of word-spacing property.

```
<html>
  <head>
    <title>Manipulating Text</title>
    <style>
        p { word-spacing:15px; }
    </style>
  </head>
  <body>
    <p>
    This text is having space between words.
    </p>
  </body>
</html>
```

Output

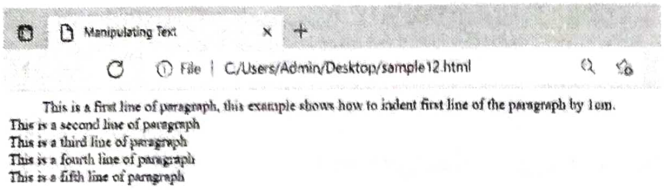


Program 1.18.4 : Write a code to demonstrate the use of text-indent property.

```
<html>
<title>Manipulating Text</title>
  <head>
      <style>
      p { text-indent:1cm; }
      </style>
  </head>
  <body>
    <p >
    This is a first line of paragraph, this example shows how to
indent first line of the paragraph by 1cm. <br>
    This is a second line of paragraph <br/>
    This is a third line of paragraph <br/>
    This is a fourth line of paragraph <br/>
    This is a fifth line of paragraph <br/>
```

```
    </p>
  </body>
</html>
```

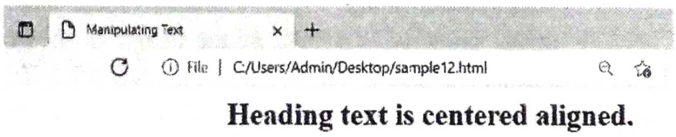
Output



Program 1.18.5 : Write a code to demonstrate the use of text-align property.

```
<html>
<title>Manipulating Text</title>
    <head>
        <style>
        h1 { text-align: center; }
        p { text-align: justify; }
        </style>
  </head>
  <body>
      <h1>
        Heading text is centered aligned.
      </h1>
    <p >
        Alignment of paragraph text is set to justify by using CSS text-align property.
    </p>
  </body>
</html>
```

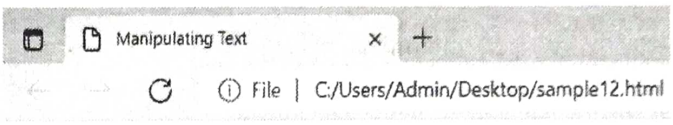
Output



Program 1.18.6 : Write a code to demonstrate the use of text-decoration property.

```
<html>
<title>Manipulating Text</title>
    <head>
        <style>
        h1 { text-decoration: underline; }
        b { text-decoration: overline; }
        </style>
  </head>
  <body>
      <h1>
        Heading text is underlined.
      </h1>
    <p >
        The text written in <b> bold</b> is overlined by using CSS text-decoration property.
    </p>
  </body>
</html>
```

Output



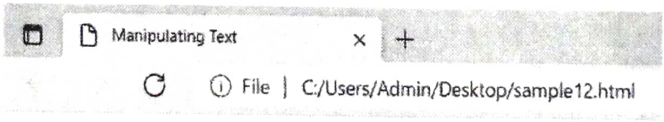
Heading text is underlined.

The text written in **bold** is overlined by using CSS text-decoration property.

Program 1.18.7 : Write a code to demonstrate the use of text-transform property.

```
<html>
<title>Manipulating Text</title>
    <head>
        <style>
        h1 { text-transform:capitalize; }
        p  { text-transform:lowecase;}
        b { text-transform:uppercase; }
        </style>
  </head>
  <body>
      <h1>
        Heading text is underlined.
      </h1>
    <p >
        The text written in <b> bold</b> is overlined by using  CSS text-decoration property.
    </p>
  </body>
</html>
```

Output



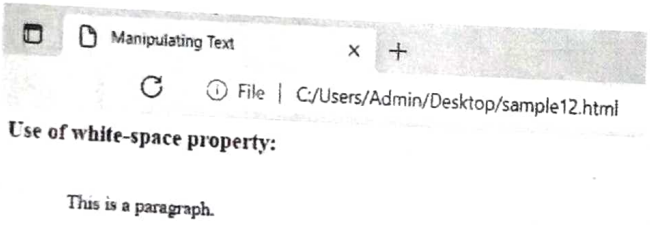
Heading Text Is Underlined.

The text written in **BOLD** is overlined by using CSS text-decoration property.

Program 1.18.8 : Write a code to demonstrate the use of white-space property.

```
<html>
<title>Manipulating Text</title>
    <head>
        <style>
        h6 { white-space:nowrap; }
        p  { white-space:pre;}
        b { text-transform:uppercase; }
        </style>
  </head>
  <body>
     <h3>
       Use of white-space property:
     </h3>
    <p >
       This is a paragraph.
    </p>
  </body>
</html>
```

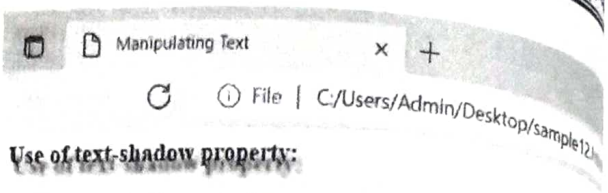
Output



Program 1.18.9 : Write a code to demonstrate the use of text-shadow property.

```
<html>
<title>Manipulating Text</title>
    <head>
        <style>

        h3 { text-shadow:4px 8px 3px red;}
        b { text-shadow:2px 4px 3px maroon;}
        </style>
  </head>
  <body>
     <h3>
       Use of text-shadow property:
     </h3>
    <p >
       In this paragraph, the text-shadow property is applied on the <b>bold</b> text .
    </p>
  </body>
</html>
```

Output



1.18.2 Background

GQ. How to set background images in CSS?

- There are several CSS properties used for setting background images of an element. To set background for entire page these properties are with <body> element.
- Following are the CSS background- properties which are used to set an image as background of an element
  - **background-image :** The value of this property an image URL to be used as background. This used to specify the source URL for background image.
  - **background-repeat :** This property is used specify whether and how the background image should repeat or tile.
  - **background-attachment :** This property is used to specify whether and how the background image should scroll with the Webpage content.
  - **background-position :** This property is used specify the position of background image relative to the element.
  - **background-image :** The value of background image property is a URL of the image as shown below :

```
body { background-image: url(/folder/subfolder/image.jpg);
```

- We can write image path in url() in either of three ways :
  - **Relative path based on the root domain of the CSS file :** when the image is in root directory of CSS file this path is used. The path begins with slash (/) indicating the root directory.

e.g.: url(/folder/subfolder/image.jpg)

  - **Relative path based on the immediate directory of the CSS file :** when the image is in the same directory of CSS file this path is used.

e.g.: (url(folder/subfolder/image.jpg))

  - **Absolute path :** when the image stored externally.

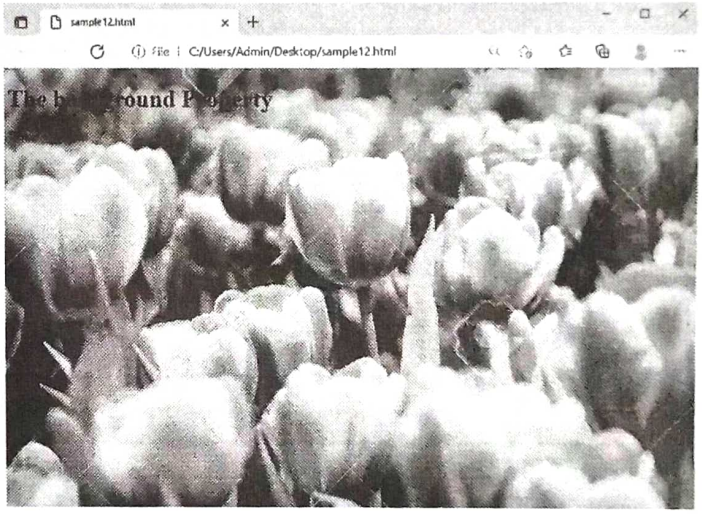
e.g.: url(http://example.com/folder/subfolder/image.jpg)

- **background-repeat :** By default this property is set to repeat the image as both horizontally and vertically. Following are the values which can be used for this property :
  - **repeat :** By default this value is set for background-repeat property. This value repeats the background image in both directions.
  - repeat-x : This value repeats the background image only on the x-axis i.e. horizontally across the element.
  - repeat-y : This value repeats the background image only on the y-axis i.e. vertically down to the element.
  - no-repeat : This value doesn't repeats the background image.
- **background-attachment :** This property specifies how the background image responds to scrolling.
  - **fixed :** This specifies that the background image is fixed relative to the browser window means scrolling the entire page or contents of the element doesn't leads to move the background image.
  - **scroll :** This specifies that the background image is fixed relative to the element.
  - **local :** This specifies that the background image is fixed relative to the contents of the image means if the contents of the image are scrolled, then the background image also scroll with them. And also if the entire page is scrolled by which the element content moved, then the background image also moves too.
- **background-position :** This is used to specify the position of a background image.
- The easiest way to position the background image is to use the positioning keywords left, right, top, bottom, and center.
- Combining this keywords gives nine possibilities like :
  - Left top
  - Left center
  - Left bottom
  - Center top
  - Center center
  - Center bottom
  - Right top
  - Right center
  - Right bottom

Program 1.18.10 : Set different background properties in one declaration:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
 background: url("tulip.jpg") no-repeat fixed center;
}
</style>
</head>
<body>
<h1>The background Property</h1>
<p>This is some text</p>
</body>
</html>
```

Output



1.18.3 Borders

The CSS border properties allow you to specify the style, width, and color of an element's border.

The border-style property specifies what kind of border to display.

The following values are allowed:

- **dotted** - Defines a dotted border
- **dashed** - Defines a dashed border
- **solid** - Defines a solid border
- **double** - Defines a double border
- **groove** - Defines a 3D grooved border. The effect depends on the border-color value
- **ridge** - Defines a 3D ridged border. The effect depends on the border-color value

- **inset** - Defines a 3D inset border. The effect depends on the border-color value
- **outset** - Defines a 3D outset border. The effect depends on the border-color value
- **none** - Defines no border
- **hidden** - Defines a hidden border

Program 1.18.11

```
<!DOCTYPE html>
<html>
<head>
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
</style>
</head>
<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>

<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>
<p class="mix">A mixed border.</p>
</body>
</html>
```

Output

The border-style Property

This property specifies what kind of border to display:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

No border.

A hidden border.

A mixed border.

1.18.4 Border Images

The CSS border-image property allows you to specify an image to be used instead of the normal border around an element.

The property has three parts:

- The image to use as the border
- Where to slice the image
- Define whether the middle sections should be repeated or stretched

For border-image to work, the element also needs the border property set.

Border Image Properties

- **border-image:** A shorthand property for setting all the border-image-* properties
- **border-image-source:** Specifies the path to the image to be used as a border
- **border-image-slice:** Specifies how to slice the border image
- **border-image-width:** Specifies the widths of the border image
- **border-image-outset:** Specifies the amount by which the border image area extends beyond the border box
- **border-image-repeat:** Specifies whether the border image should be repeated, rounded or stretched

Program 1.18.12

```
<!DOCTYPE html>
<html>
<head>
<style>
#borderimg {
 border: 10px solid transparent;
 padding: 15px;
 border-image: url(border.png) 30 round;
}
</style>
</head>
<body>
<h1>The border-image Property</h1>
<p>Here, the middle sections of the image are repeated to
create the border:</p>
<p id="borderimg">border-image: url(border.png) 30
round;</p>
<p>Here is the original image:</p><img
src="border.png">
</body>
</html>
```

Output

The border-image Property

Here, the middle sections of the image are repeated to create the border:

border-image: url(border.png) 30 round;

Here is the original image:

1.18.5 Margin

GQ. How to set margins in CSS?

- The margin property is used to define the space around elements outside its borders.
- CSS allows setting margins for individual sites of the element using margin-left, margin-right, margin-top, and margin-bottom properties or using a shorthand property margin.

CSS syntax

margin: auto | *length* | initial | inherit;

Margin Properties

All the margin properties have the following values:

1. **auto :** This value is used to indicate that the browser itself calculates the margin for the element.
2. **length :** This value is used to indicate margin for the element in px, pt, cm, etc.
3. **initial :** This value is used to set margin for the element to its default values.
4. **inherit :** This value is used to indicate that the margin for the element must be inherited from its parent element.
5. **%** : This value is used to specify the margin for the element in % of the width of its parent element

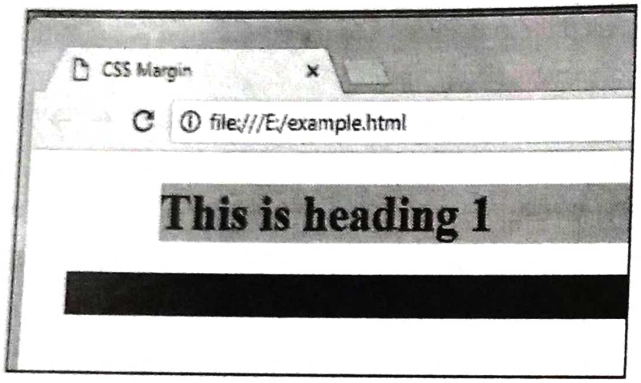
Following program shows how the margin property is set for an element:

Program 1.18.13 : Write a code to demonstrate use of CSS margin property.

```
<html>
<title>CSS Margin</title>
    <head>
    <style>
    h1{ margin: 20px 5px 10px 80px;
```

```
    background: orange;
    }
    h2 {margin-left:20px;
    background: red;
    }
    </style>
  </head>
  <body>
    <h1>
      This is heading 1
    </h1>
    <h2>
      This is heading 2
    </h2>
  </body>
</html>
```

Output



1.19 PADDING LISTS

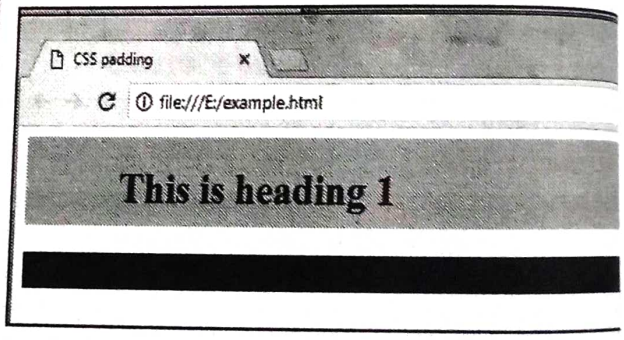
GQ. Write note on Padding lists in CSS with suitable example.

- Unlike margin, the padding doesn't affect the distance between elements instead it specifies the distance inside the element i.e. the distance between the border and the content of the element.
- CSS allows setting padding for individual sites of the element using padding-left, padding-right, padding-top, and padding-bottom properties or using shorthand property padding.
- Following program shows how to set padding for HTML elements.

Program 1.19.1 : Write a code to demonstrate the use of padding property.

```
<html>
<title>CSS padding</title>
  <head>
    <style>
    h1{ padding: 20px 5px 10px 80px;
    background: orange;
    }
    h2 {padding-left:20px;
    background: red;
    }
    </style>
  </head>
  <body>
    <h1>
      This is heading 1
    </h1>
    <h2>
      This is heading 2
    </h2>
  </body>
</html>
```

Output



1.20 CSS PROPERTIES FOR POSITIONING AN ELEMENT

GQ. What is positioning in CSS? Explain it with example

- The CSS positioning properties specifies the position of an HTML element on the Webpage.
- This can allow overlapping the HTML elements by positioning an HTML element behind another.
- The HTML elements are positioned using the top, bottom, left, and right properties.
- But before these properties the position property and positioning method should be set for that element, otherwise positioning using top, bottom, left, and right properties will not work properly.

- There are four different positioning methods as explained below :

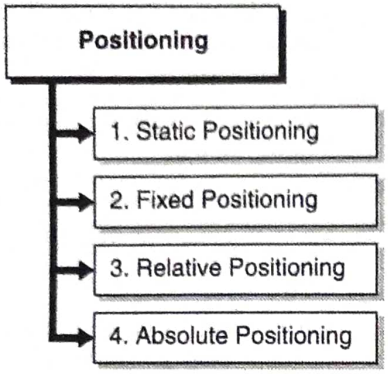


Fig. 1.20.1 : Positioning methods

1. Static Positioning

- By default the HTML elements are positioned static.
- The HTML elements positioned in static have the flow like the normal flow of the page i.e. displaying the contents of the element in sequence as given in the HTML document.
- The top, bottom, left, and right properties can't affect the static positioned HTML elements.

2. Fixed Positioning

- An element with fixed position gets a fixed place on the webpage regardless of page scrolling. The positioning of HTML element is relative to the browser window.
- This method of positioning can overlaps other HTML elements.

3. Relative Positioning

- Here the positioning of an HTML element is relative to its normal position the webpage
- This method of positioning can overlap other HTML elements.

4. Absolute Positioning

- An HTML element positioned using absolute method is appear on the webpage at specified position relative to the first parent element which has a position other than static position.
- Otherwise the positioning of an HTML element is relative to the left top corner of the webpage screen. In other way we can say that the positioning is relative to the <html> element as parent element to all the HTML elements.
- This method of positioning can overlaps other HTML elements.

Overlapping Elements

- Basically we can say that the HTML element can overlap other HTML elements if they are positioned outside the normal flow of the webpage.
- For overlapping the elements, z-index property is also used. It specifies how the elements should be placed i.e. in front of all stacked elements, or behind the elements.
- The value of z-index property is a positive or negative number. The element with greatest z-index value will appear at the front of all stacked elements.
- If the z-index property is not used while overlapping the elements then the HTML element which is positioned at last in the HTML code will appear in the front of other overlapped elements.

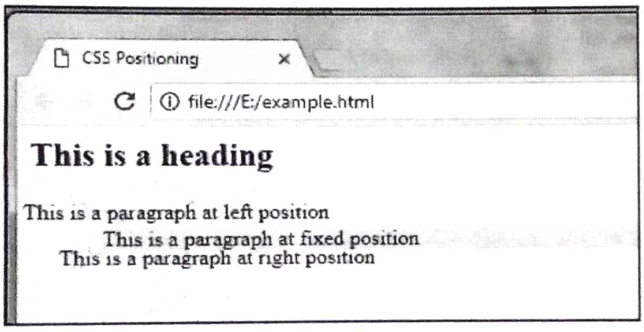
Following program shows how to position the HTML elements using different positioning methods

Program 1.20.1 : Write a code to demonstrate use of position property of CSS.

```
<html>
<title>CSS Margin</title>
<head>
<style>
h2.pos_static
{
position:static;
}
p.pos_fixed
{
position:relative;bottom:48px; left:50px; background-color:yellow;
}
p.pos_left
{
position:relative; left:-5px;
}
p.pos_right
{
position:relative; left:20px;
}
img
{
```

```
  position:absolute;
  left:35px;
  top:35px;
  z-index:-1;
 }
 </style>
</head>
<body>
 <h2 class="pos_static">
 This is a heading
 </h2>
 <p class="pos_left">This is a paragraph at left position</p>
 <p class="pos_right">This is a paragraph at right position</p>
 <p class="pos_fixed">This is a paragraph at fixed position</p>
 <img src="bg.gif">
 </body>
</html>
```

Output



1.21 CSS DISPLAY

- CSS display is the most important property of CSS which is used to control the layout of the element. It specifies how the element is displayed.
- Every element has a default display value according to its nature. Every element on the webpage is a rectangular box and the CSS property defines the behaviour of that rectangular box.

Syntax

```
display: value;
```

Property values

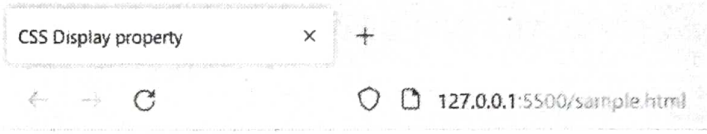
Value	Description
inline	It is used to displays an element as an inline element.
block	It is used to displays an element as a block element
contents	It is used to disappear the container.
flex	It is used to display an element as a block-level flex container.
grid	It is used to display an element as a block-level grid container.
inline-block	It is used to display an element as an inline-level block container.
inline-flex	It is used to display an element as an inline-level flex container.
inline-grid	It is used to display an element as an inline-level grid container.
inline-table	It is used to display an inline-level table
list-item	It is used to display all the elements in <li> element.
run-in	It is used to display an element inline or block level, depending on the context.
table	It is used to set the behavior as <table> for all elements.
table-caption	It is used to set the behavior as <caption> for all elements.
table-column-group	It is used to set the behavior as <column> for all elements.
table-header-group	It is used to set the behavior as <header> for all elements.
table-footer-group	It is used to set the behavior as <footer> for all elements.
table-row-group	It is used to set the behavior as <row> for all elements.
table-cell	It is used to set the behavior as <td> for all elements.
table-column	It is used to set the behavior as <col> for all elements.
table-row	It is used to set the behavior as <tr> for all elements.
none	It is used to remove the element.
initial	It is used to set the default value.
inherit	It is used to inherit the property from it's parents' elements.

1. **Block :** This property is used as the default property of div. This property places the div one after another vertically. The height and width of the div can be changed using the block property if the width is not mentioned, then the div under block property will take up the width of the container.

Example

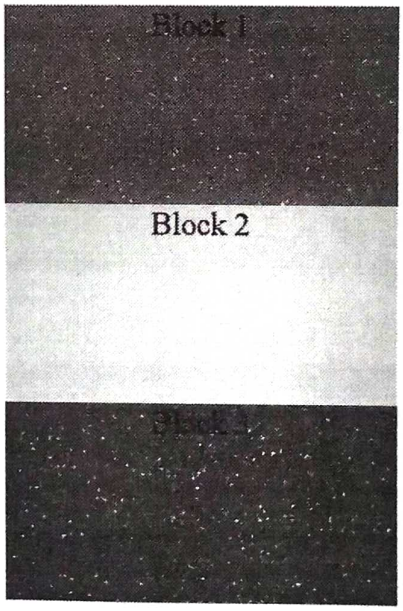
```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS Display property</title>
    <style>
      #sample1{
        height: 100px;
        width: 200px;
        background: teal;
        display: block;
      }
      #sample2{
        height: 100px;
        width: 200px;
        background: cyan;
        display: block;
      }
      #sample3{
        height: 100px;
        width: 200px;
        background: green;
        display: block;
      }
      .demo1 {
        margin-left:20px;
        font-size:42px;
        font-weight:bold;
        color:#009900;
      }
      .demo2 {
        font-size:25px;
        margin-left:30px;
      }
      .main {
        margin:50px;
        text-align:center;
      }
    </style>
  </head>

  <body>
    <div class = "demo1">Saidham</div>
    <div class = "demo2">display: block;
property</div>
    <div class = "main">
      <div id="sample1">Block 1 </div>
      <div id="sample2">Block 2</div>
      <div id="sample3">Block 3</div>
    </div>
  </body>
</html>
```

Output



Saidham

display: block; property



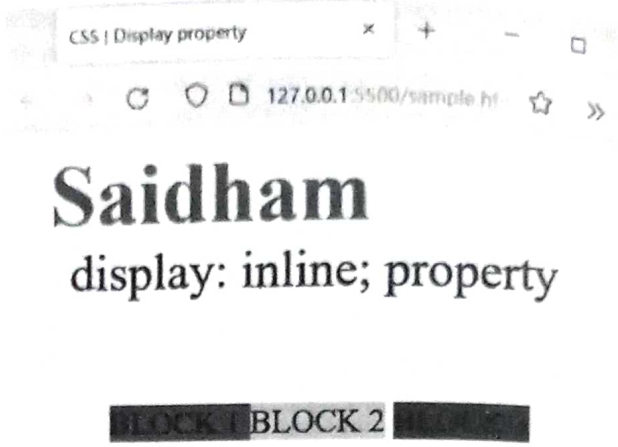
2. **Inline:** This property is the default property of anchor tags. This is used to place the div inline i.e. in a horizontal manner. The inline display property ignores the height and the width set by the user.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS | Display property</title>
    <style>
```

```
        #main{
            height: 200px;
            width: 200px;
            background: teal;
            display: inline;

        }
        #main1{
            height: 200px;
            width: 200px;
            background: cyan;
            display: inline;

        }
        #main2{
            height: 200px;
            width: 200px;
            background: green;
            display: inline;
        }
        .demo1 {
            margin-left:20px;
            font-size:42px;
            font-weight:bold;
            color:#009900;
        }
        .demo2 {
            font-size:25px;
            margin-left:30px;
        }
        .main {
            margin:50px;
        }
    </style>
  </head>
  <body>
    <div class = "demo1">Saidham</div>
    <div class = "demo2">display: inline;
property</div>
    <div class = "main">
        <div id="main"> BLOCK 1 </div>
        <div id="main1"> BLOCK 2</div>
        <div id="main2">BLOCK 3 </div>
    </div>
  </body>
</html>
```

Output



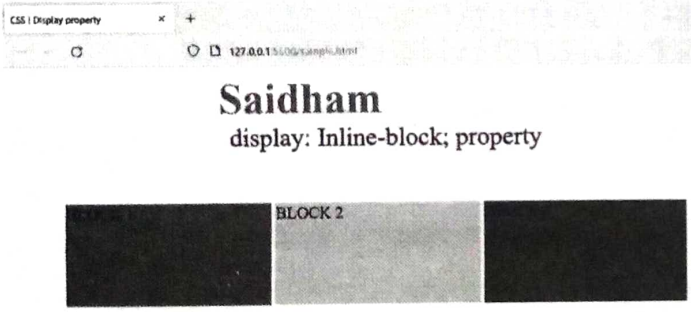
3. **Inline-block :** This features uses the both prope mentioned above, block and inline. So, this pro aligns the div inline but the difference is it can edi height and the width of block. Basically, this will the div both in block and inline fashion.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS | Display property</title>
    <style>
    #main{
        height: 100px;
        width: 200px;
        background: teal;
        display: inline-block;

    }
    #main1{
        height: 100px;
        width: 200px;
        background: cyan;
        display: inline-block;

    }
    #main2{
        height: 100px;
        width: 200px;
        background: green;
        display: inline-block;
    }
    .demo1 {
        margin-left:200px;
```

```
            font-size:42px;
            font-weight:bold;
            color:#009900;
        }
        .demo2 {
            font-size:25px;
            margin-left:210px;
        }
        .main {
            margin:50px;
        }
    </style>
  </head>
  <body>
    <div class = "demo1">Saidham</div>
    <div class = "demo2">display: Inline-block;
property</div>
    <div class = "main">
        <div id="main"> BLOCK 1 </div>
        <div id="main1"> BLOCK 2</div>
        <div id="main2">BLOCK 3 </div>
    </div>
  </body>
</html>
```

Output



1.22 CSS FLEX PROPERTY

- The flex CSS shorthand property is the combination of flex-grow, flex-shrink, and flex-basis property. It is used to set the length of flexible items.
- The flex property is much responsive and mobile-friendly. It is easy to position child elements and the main container.
- The margin doesn't collapse with the content margins. The order of any element can be easily changed without editing the HTML section.

Syntax

```
flex: flex-grow flex-shrink flex-basis | auto | initial | inherit;
```

Property Values

- **flex-grow Property :** A number that specifies, how much items will grow relative to the rest of the flexible items.
- **flex-shrink Property :** A number that specifies, how much items will shrink relative to the rest of the flexible items.
- **flex-basis Property :** It sets the length of items. Legal values of flex-basis are: auto, inherit, or a number followed by %, em, px, or any other length unit.
- **flex-wrap Property :** The CSS flex-wrap property is used to specify whether flex items are forced into a single line or wrapped onto multiple lines.

The flex property can be specified with the help of 1, 2 or 3 values:

- **One-value syntax :** the value should contain one of following:
  - **number :** If it is represented as flex: <number> 1 0; then the value of flex-shrink, flex-basis will supposed to be 1 & 0 respectively.
  - It can be specified by one of the keyword as auto, none or initial.
- **Two-value syntax :** It must contains the following values:

  The first value should be the number that will represent the flex-grow.

  The second value should contain one of the following:
  - **Number :** If it is number then it will represented as flex-shrink.
  - a width with the valid value will represents the flex-basis.
- **Three-value syntax :** The values should be in the same order:
  - first number represents the flex-grow.
  - second number represents the flex-shrink.
  - a width with the valid value will represents the flex-basis.

Example 1

This example describes the flex property with the help of the single value to represent the flex.

```
<!DOCTYPE html>
<html>

<head>
    <title> CSS flex Property </title>
    <style>
    #sample {
        width: 300px;
```

```
        height: 200px;
        border: 1px solid black;
        display: flex;
    }
    #sample div {
        flex: 1;
    }
    .GFG1 {
        background-color: green;
    }
    .GFG2 {
        background-color: lightgreen;
    }
    .GFG3 {
        background-color: darkgreen;
    }
    </style>
</head>
<body>
    <h2>CSS flex Property</h2>
    <div id="sample">
        <div class="GFG1"> ITSA </div>
        <div class="GFG2"> SPCA </div>
        <div class="GFG3"> ASSET </div>
    </div>
</body>
</html>
```

Output

Example 2

This example describes the flex property with the help of the 3 values that represents the flex-grow, flex-shrink & flex-basis properties.

Program

```
<!DOCTYPE html>
<html>

<head>
    <title> CSS flex Property </title>
    <style>
    #sample {
        width: 300px;
        height: 200px;
        border: 1px solid black;
        display: flex;
    }

    #sample div {
        flex: 1 0 auto;
    }

    .GFG1 {
        background-color: green;
    }

    .GFG2 {
        background-color: lightgreen;
    }

    .GFG3 {
        background-color: darkgreen;
    }
    </style>
</head>

<body>
    <h2>CSS flex Property</h2>
    <div id="sample">
        <div class="GFG1"> ITSA </div>
        <div class="GFG2"> SPCA </div>
        <div class="GFG3"> ASSET </div>
    </div>
</body>

</html>
```

Output

1.23 CSS GRID PROPERTY

It is a CSS property that offers a grid-based layout system, with rows and columns, making it easier to design web pages without floats and positioning.

row1-colum1		
row2-column2	row2-column2	row2-column2
	Footer	

Syntax

grid: none | grid-template-rows / grid-template-columns | grid-template-areas | grid-template-rows / [grid-auto-flow] grid-auto-columns | [grid-auto-flow] grid-auto-rows / grid-template-columns | initial | inherit;

Property values

Value	Description
none	It is default value no specific size mentioned for row and column.
grid-template-rows / grid-template-columns	It is used to sepcifie the size of rows and columns.
grid-template-areas	It is used to specifie the grid layout using named items.
grid-template-rows / grid-auto-columns	It is used to specifie the auto size(height) and sets the auto size columns.
grid-auto-rows / grid-template-columns	It is used to specifie the auto size and sets the auto grid size columns.
grid-template-rows / grid-auto-flow grid-auto-columns	It is used to specifie how to place items and auto size row and columns.
grid-auto-flow grid-auto-rows / grid-template-columns	It is used to specifie how to place items and auto size row and grid-template columns.

Example 1 : Grid with 2-rows and 4-column.

```
<!DOCTYPE html>
<html>

<head>
    <title>
        CSS | grid Property
    </title>
    <style>
        .main {
            display: grid;
            grid: auto auto / auto auto auto auto;
            grid-gap: 10px;
            background-color: green;
            padding: 10px;
        }

        .sample1 {
            background-color: lightgrey;
            text-align: center;
            padding: 25px 0;
            font-size: 30px;
        }
    </style>
</head>

<body>

    <h1>Welcome to CSS Grid </h1>
    <h3> CSS grid property</h3>

    <div class="main">
        <div class="sample1">1</div>
        <div class="sample1">2</div>
        <div class="sample1">3</div>
        <div class="sample1">4</div>
        <div class="sample1">5</div>
```

```
    <div class="sample1">6</div>
    <div class="sample1">7</div>
    <div class="sample1">8</div>
  </div>

</body>
</html>
```

Output

Welcome to CSS Grid

CSS grid property

1	2	3	4
5	6	7	8

1.23.1 CSS3 Transformations

The transform property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skew, etc., elements.

CSS 2D Transforms

CSS transforms allow you to move, rotate, scale, and skew elements.

With the CSS transform property you can use the following 2D transformation methods:

- translate()
- rotate()
- scaleX()
- scaleY()
- scale()
- skewX()
- skewY()
- skew()
- matrix()

1. The translate() Method

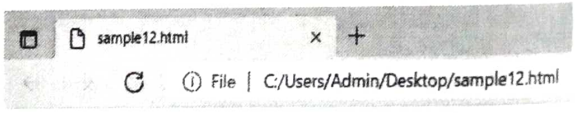
The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).

The following example moves the element 50 pixels to the right, and 100 pixels down from its current position:

Program 1.23.1

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;
  -ms-transform: translate(50px,100px); /* IE 9 */
  transform: translate(50px,100px); /* Standard syntax */
}
</style>
</head>
<body>
<h1>The translate() Method</h1>
<p>The translate() method moves an element from its current position:</p>
<div>
This div element is moved 50 pixels to the right, and 100 pixels down from its current position.
</div>
</body>
</html>
```

Output



The translate() Method

The translate() method moves an element from its current position:

This div element is moved 50 pixels to the right, and 100 pixels down from its current position.

2. The rotate() Method

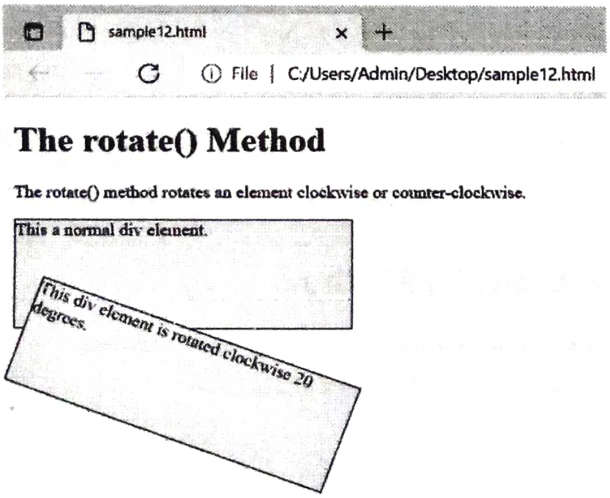
The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.

The following example rotates the element clockwise with 20 degrees:

Program 1.23.2

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 300px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;
}
div#myDiv {
 -ms-transform: rotate(20deg); /* IE 9 */
 transform: rotate(20deg); /* Standard syntax */}
</style>
</head>
<body>
<h1>The rotate() Method</h1>
<p>The rotate() method rotates an element clockwise or
counter-clockwise.</p>
<div>
This a normal div element.
</div>
<div id="myDiv">
This div element is rotated clockwise 20 degrees.
</div>
</body>
</html>
```

Output

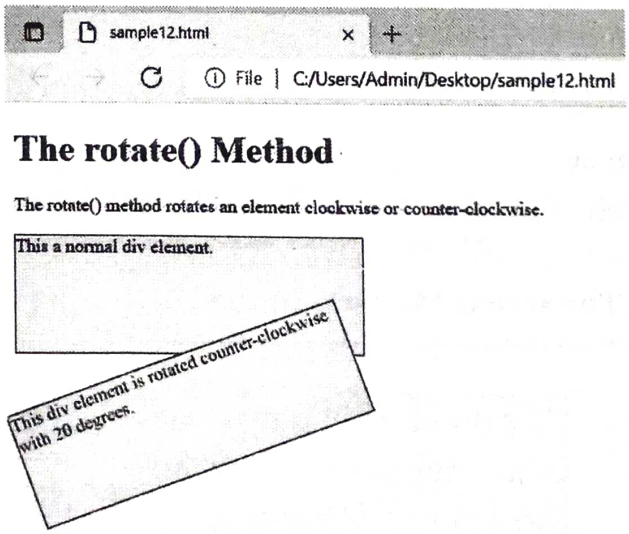


Using negative values will rotate the element counter-clockwise. The following example rotates the element counter-clockwise with 20 degrees:

Program 1.23.3

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 300px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;
}
div#myDiv {
 -ms-transform: rotate(-20deg); /* IE 9 */
 transform: rotate(-20deg); /* Standard syntax */
}
</style>
</head>
<body>
<h1>The rotate() Method</h1>
<p>The rotate() method rotates an element clockwise or
counter-clockwise.</p>
<div>
This a normal div element.
</div>
<div id="myDiv">
This div element is rotated counter-clockwise with 20 degrees.
</div>
</body>
</html>
```

Output



3. The scale() Method

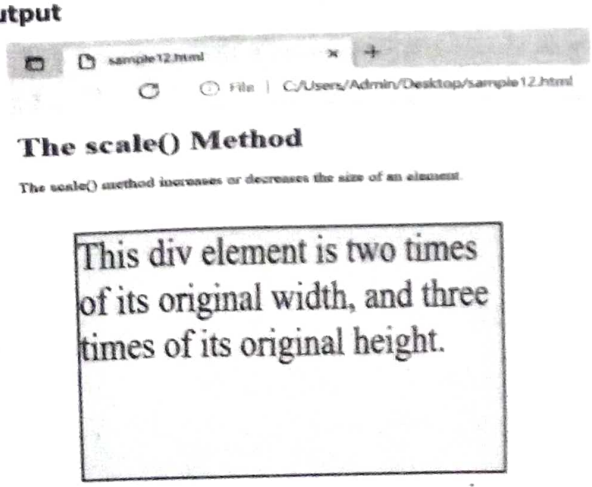
The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).

The following example increases the element to be two times of its original width, and three times of its original height:

Program 1.23.4

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 margin: 150px;
 width: 200px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;
 -ms-transform: scale(2,3); /* IE 9 */
 transform: scale(2,3); /* Standard syntax */
}
</style>
</head>
<body>
<h1>The scale() Method</h1>
<p>The scale() method increases or decreases the size of an element.</p>
<div>
This div element is two times of its original width, and three times of its original height.
</div>
</body>
</html>
```

Output



4. The scaleX() Method

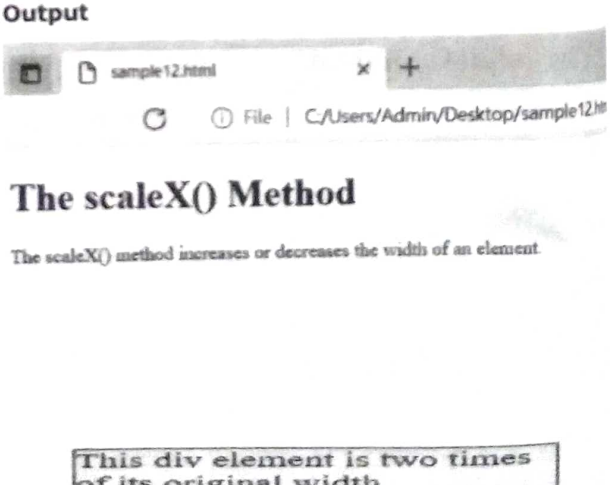
The scaleX() method increases or decreases the wi of an element.

The following example increases the element be two times of its original width:

Program 1.23.5

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 margin: 150px;
 width: 200px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;
 -ms-transform: scaleX(2); /* IE 9 */
 transform: scaleX(2); /* Standard syntax */
}
</style>
</head>
<body>
<h1>The scaleX() Method</h1>
<p>The scaleX() method increases or decreases the width
an element.</p>
<div>
This div element is two times of its original width.
</div>
</body>
</html>
```

Output



5. The scaleY() Method

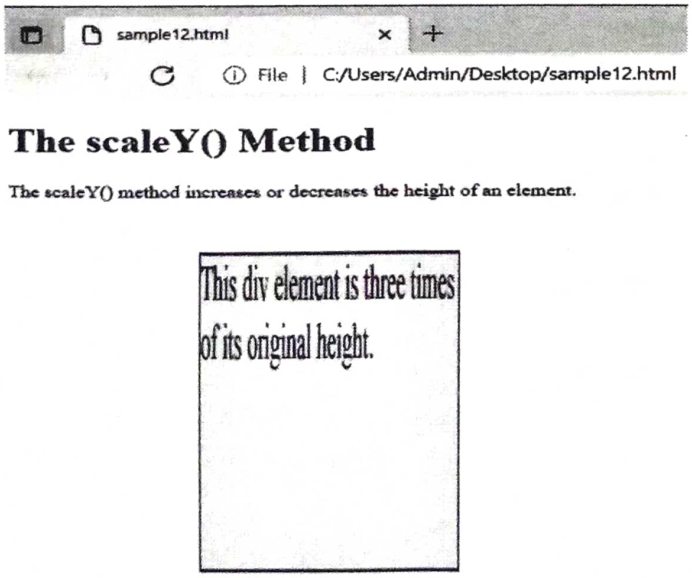
The scaleY() method increases or decreases the height of an element.

The following example increases the element to be three times of its original height.

Program 1.23.6

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  margin: 150px;
  width: 200px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;
  -ms-transform: scaleY(3); /* IE 9 */
  transform: scaleY(3); /* Standard syntax */
}
</style>
</head>
<body>
<h1>The scaleY() Method</h1>
<p>The scaleY() method increases or decreases the height of an element.</p>
<div>
This div element is three times of its original height.
</div>
</body>
</html>
```

Output



6. The skew() Method

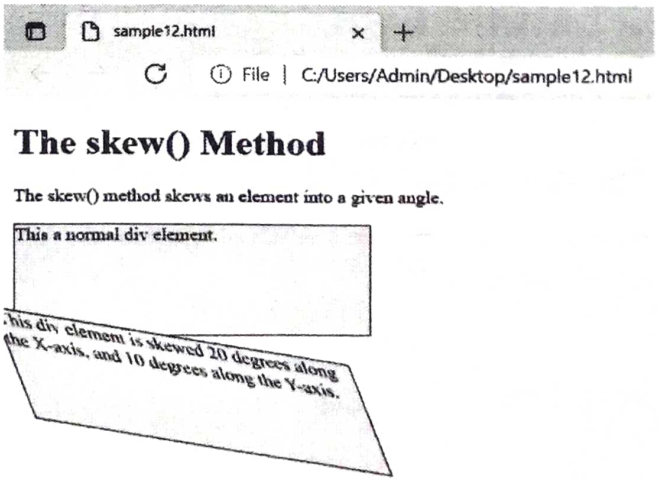
The skew() method skews an element along the X and Y-axis by the given angles.

The following example skews the element 20 degrees along the X-axis, and 10 degrees along the Y-axis:

Program 1.23.7

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;}
div#myDiv {
  -ms-transform: skew(20deg,10deg); /* IE 9 */
  transform: skew(20deg,10deg); /* Standard syntax */}
</style>
</head>
<body>
<h1>The skew() Method</h1>
<p>The skew() method skews an element into a given angle.</p>
<div>
This a normal div element.
</div>
<div id="myDiv">
This div element is skewed 20 degrees along the X-axis, and 10 degrees along the Y-axis.
</div>
</body>
</html>
```

Output



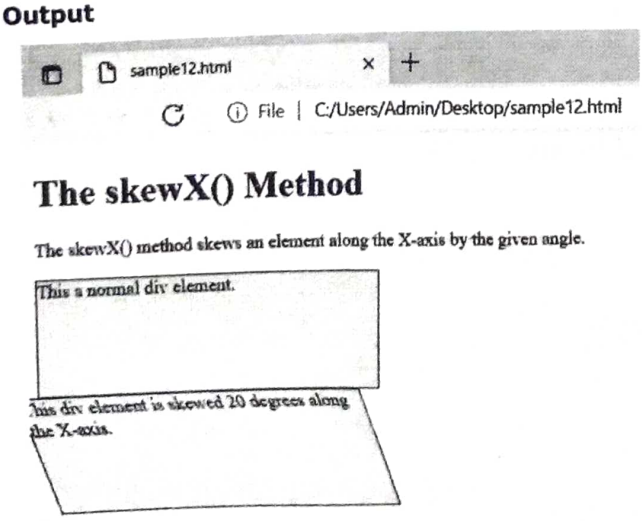
7. The skewX() Method

The skewX() method skews an element along the X-axis by the given angle. The following example skews the element 20 degrees along the X-axis:

Program 1.23.8

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 300px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;
}
div#myDiv {
 -ms-transform: skewX(20deg); /* IE 9 */
 transform: skewX(20deg); /* Standard syntax */
}
</style>
</head>
<body>
<h1>The skewX() Method</h1>
<p>The skewX() method skews an element along the X-axis by the given angle.</p>
<div>
This a normal div element.
</div>
<div id="myDiv">
This div element is skewed 20 degrees along the X-axis.
</div>
</body>
</html>
```

Output



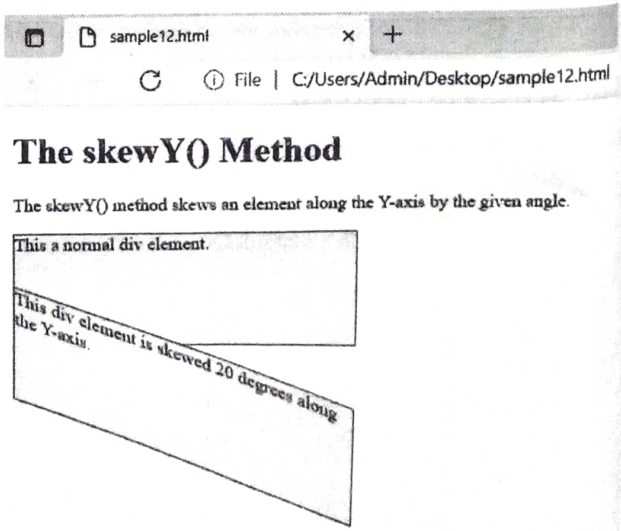
8. The skewY() Method

The skewY() method skews an element along the Y-axis by the given angle. The following example skews the element 20 degrees along the Y-axis:

Program 1.23.9

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 300px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;}
div#myDiv {
 -ms-transform: skewY(20deg); /* IE 9 */
 transform: skewY(20deg); /* Standard syntax */}
</style>
</head>
<body>
<h1>The skewY() Method</h1>
<p>The skewY() method skews an element along the Y-axis by the given angle.</p>
<div>
This a normal div element.
</div>
<div id="myDiv">
This div element is skewed 20 degrees along the Y-axis.
</div>
</body>
</html>
```

Output



9. The matrix() Method

The matrix() method combines all the 2D transform methods into one.

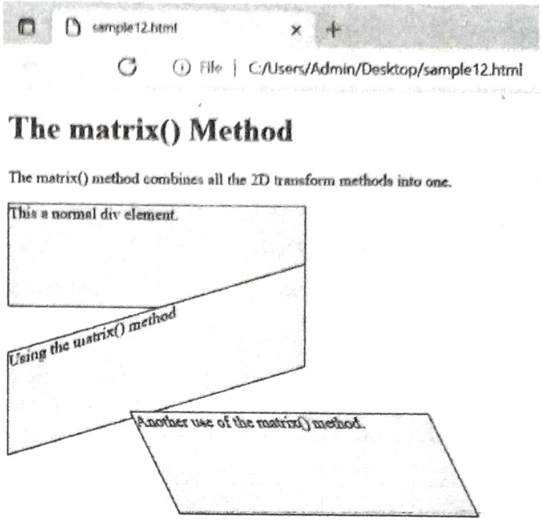
The matrix() method take six parameters, containing mathematic functions, which allows you to rotate, scale, move (translate), and skew elements.

The parameters are as follow: matrix(scaleX(), skewY(),skewX(),scaleY(),translateX(),translateY())

Program 1.23.10

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 300px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;
}
div#myDiv1 {
 -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */
 transform: matrix(1, -0.3, 0, 1, 0, 0); /* Standard syntax */
}
div#myDiv2 {
 -ms-transform: matrix(1, 0, 0.5, 1, 150, 0); /* IE 9 */
 transform: matrix(1, 0, 0.5, 1, 150, 0); /* Standard syntax */
}
</style>
</head>
<body>

<h1>The matrix() Method</h1>
<p>The matrix() method combines all the 2D transform
methods into one.</p>

<div>
This a normal div element.
</div>

<div id="myDiv1">
Using the matrix() method:
</div>
<div id="myDiv2">
Another use of the matrix() method.
</div>

</body>
</html>
```

Output



☞ Transform Properties

The following table lists all the 2D transform properties.

Property	Description
transform	Applies a 2D or 3D transformation to an element.
transform-origin	Allows you to change the position on transformed elements.

☞ CSS 3D Transforms

With the CSS transform property you can use the following 3D transformation methods :

1. rotateX()
2. rotateY()
3. rotateZ()

▶ 1. The rotateX() Method

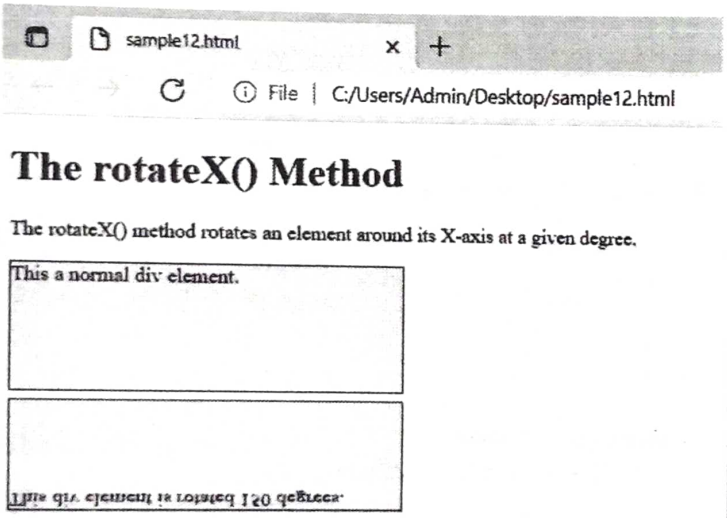
The rotateX() method rotates an element around its X-axis at a given degree:

Program 1.23.11

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 300px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;
}
```

```
#myDiv {
 transform: rotateX(150deg);
}
</style>
</head>
<body>

<h1>The rotateX() Method</h1>
<p>The rotateX() method rotates an element around its X-
axis at a given degree.</p>

<div>
This a normal div element.
</div>

<div id="myDiv">
This div element is rotated 150 degrees.
</div>
</body>
</html>
```

Output



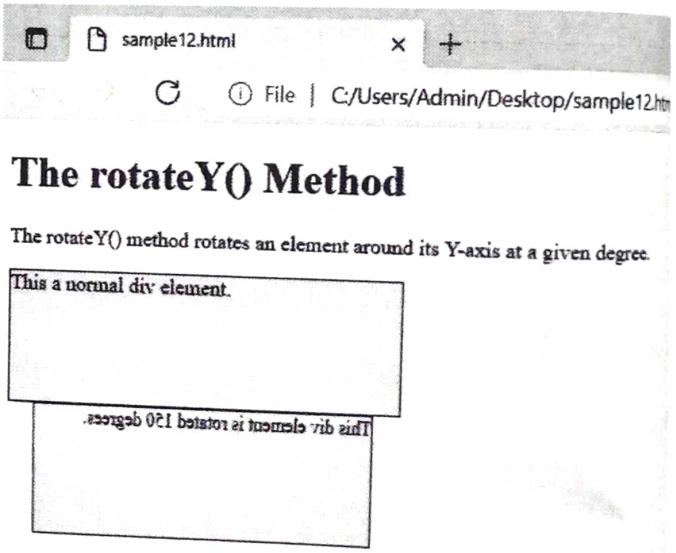
2. The rotateY() Method

The rotateY() method rotates an element around its Y-axis at a given degree.

Program 1.23.12

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 300px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;
}

#myDiv {
 transform: rotateY(150deg);
}
</style>
</head>
<body>

<h1>The rotateY() Method</h1>
<p>The rotateY() method rotates an element around its Y-
axis at a given degree.</p>

<div>
This a normal div element.
</div>

<div id="myDiv">
This div element is rotated 150 degrees.
</div>

</body>
</html>
```

Output



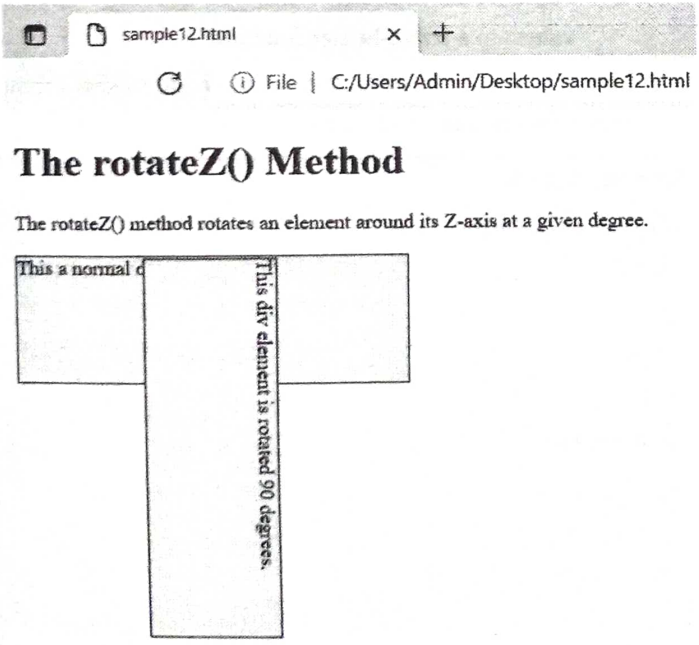
3. The rotateZ() Method

The rotateZ() method rotates an element around its Z-axis at a given degree.

Program 1.23.13

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 300px;
 height: 100px;
 background-color: yellow;
 border: 1px solid black;
}

#myDiv {
 transform: rotateZ(90deg);
}
</style>
</head>
<body>
<h1>The rotateZ() Method</h1>
<p>The rotateZ() method rotates an element around its Z-axis at a given degree.</p>
<div>
This a normal div element.
</div>
<div id="myDiv">
This div element is rotated 90 degrees.
</div>
</body>
</html>
```

Output



CSS Transform Properties

The following table lists all the 3D transform properties :

Property	Description
transform	Applies a 2D or 3D transformation to an element
transform-origin	Allows you to change the position on transformed elements
transform-style	Specifies how nested elements are rendered in 3D space
perspective	Specifies the perspective on how 3D elements are viewed
perspective-origin	Specifies the bottom position of 3D elements
backface-visibility	Defines whether or not an element should be visible when not facing the screen

1.24 CSS TRANSITIONS

- CSS transitions allows you to change property values smoothly, over a given duration.
- To create a transition effect, you must specify two things:
  - The CSS property you want to add an effect to
  - The duration of the effect
- The following example shows a 100px * 100px red element. The element has also specified a transition effect for the width property, with a duration of 2 seconds.

Program 1.24.1

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background: red;
 transition: width 2s;
}
div:hover {
 width: 300px;
}
</style>
</head>
<body>
```

```
<h1>The transition Property</h1>
<p>Hover over the div element below, to see the transition effect:</p>
<div></div>
</body>
</html>
```

Output

The transition Property

Hover over the div element below, to see the transition effect:

In the above example, when the cursor mouses out of the element, it will gradually change back to its original style.

Change Several Property Values

The following example adds a transition effect for both the width and height property, with a duration of 2 seconds for the width and 4 seconds for the height:

Program 1.24.2

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s, height 4s;
}
div:hover {
  width: 300px;
  height: 300px;
}
</style>
</head>
<body>
<h1>The transition Property</h1>
<p>Hover over the div element below, to see the transition effect:</p>
<div></div>
</body>
</html>
```

Output

The transition Property

Hover over the div element below, to see the transition effect:

Specify the Speed Curve of the Transition

- The transition-timing-function property specifies the speed curve of the transition effect.
- The transition-timing-function property can have the following values:
  - **ease** - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
  - **linear** - specifies a transition effect with the same speed from start to end
  - **ease-in** - specifies a transition effect with a slow start
  - **ease-out** - specifies a transition effect with a slow end
  - **ease-in-out** - specifies a transition effect with a slow start and end
  - **cubic-bezier(n,n,n,n)** - lets you define your own values in a cubic-bezier function
- The following example shows some of the different speed curves that can be used.

Program 1.24.3

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;
}
```

```
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
div:hover {
 width: 300px;
}
</style>
</head>
<body>
<h1>The transition-timing-function Property</h1>
<p>Hover over the div elements below, to see the different
speed curves:</p>
<div id="div1">linear</div><br>
<div id="div2">ease</div><br>
<div id="div3">ease-in</div><br>
<div id="div4">ease-out</div><br>
<div id="div5">ease-in-out</div><br>
</body>
</html>
```

Output

The transition-timing-function Property

Hover over the div elements below, to see the different speed curves:

Delay the Transition Effect

The transition-delay property specifies a delay (in seconds) for the transition effect.

The following example has a 1 second delay before starting:

Program 1.24.4

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background: red;
 transition: width 3s;
 transition-delay: 1s;
}
div:hover {
 width: 300px;
}
</style>
</head>
<body>
<h1>The transition-delay Property</h1>
<p>Hover over the div element below, to see the transition
effect:</p>
<div></div>
<p>The transition effect has a 1 second delay before
starting.</p>
</body>
</html>
```

Output

The transition-delay Property

Hover over the div element below, to see the transition effect:

The transition effect has a 1 second delay before starting.

CSS Transition Properties

The following table lists all the CSS transition properties:

Property	Description
transition	A shorthand property for setting the four transition properties into a single property
transition-delay	Specifies a delay (in seconds) for the transition effect
transition-duration	Specifies how many seconds or milliseconds a transition effect takes to complete
transition-property	Specifies the name of the CSS property the transition effect is for
transition-timing-function	Specifies the speed curve of the transition effect

Program 1.24.5

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background: red;
 transition-property: width;
 transition-duration: 2s;
 transition-timing-function: linear;
 transition-delay: 1s;
}

div:hover {
 width: 300px;
}
</style>
</head>
<body>
<h1>The transition Properties Specified One by One</h1>

<p>Hover over the div element below, to see the transition effect:</p>
<div></div>
<p> The transition effect has a 1 second delay before starting.</p>
</body>
</html>
```

Output

The transition Properties Specified One by One

Hover over the div element below, to see the transition effect:

The transition effect has a 1 second delay before starting.

1.25 CSS ANIMATIONS

- CSS allows animation of HTML elements without using JavaScript or Flash.
- An animation lets an element gradually change from one style to another.
- To use CSS animation, you must first specify some keyframes for the animation.
- Keyframes hold what styles the element will have at certain times.

CSS Animation Properties

The following table lists the @keyframes rule and all the CSS animation properties:

Property	Description
@keyframes	Specifies the animation code
animation	A shorthand property for setting all the animation properties
animation-delay	Specifies a delay for the start of an animation
animation-direction	Specifies whether an animation should be played forwards, backwards or in alternate cycles
animation-duration	Specifies how long time an animation should take to complete one cycle
animation-fill-mode	Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
animation-iteration-count	Specifies the number of times an animation should be played
animation-name	Specifies the name of the @keyframes animation
animation-play-state	Specifies whether the animation is running or paused
animation-timing-function	Specifies the speed curve of the animation

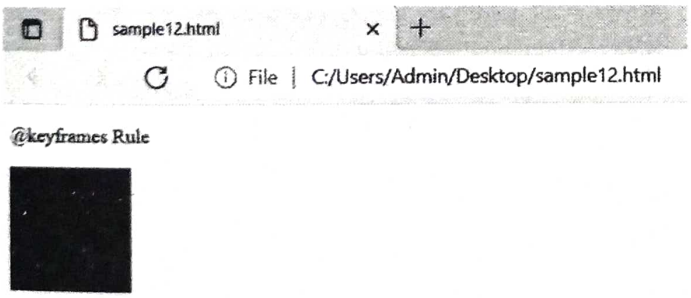
The @keyframes Rule

- When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.
- The following example binds the "example" animation to the element. The animation will last for 4 seconds, and it will gradually change the background-color of the element from "red" to "yellow".

Program 1.25.1

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background-color: red;
 animation-name: example;
 animation-duration: 4s;
}
@keyframes example {
 from {background-color: red;}
 to {background-color: yellow;}
}
</style>
</head>
<body>
<p>@keyframes Rule</p>
<div></div>
</body>
</html>
```

Output



- The animation-duration property defines how long an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds).
- In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).

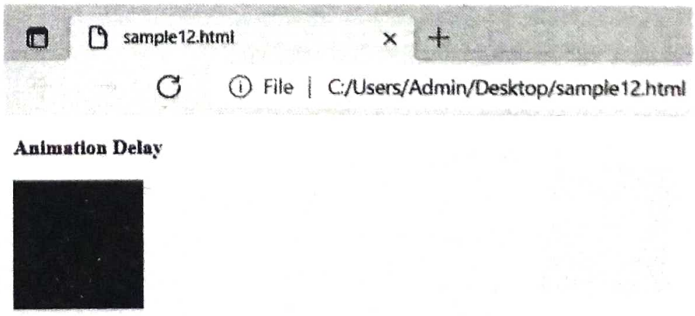
Delay an Animation

- The animation-delay property specifies a delay for the start of an animation.
- The following example has a 2 seconds delay before starting the animation:

Program 1.25.2

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background-color: red;
 position: relative;
 animation-name: example;
 animation-duration: 4s;
 animation-delay: 2s;
}
@keyframes example {
 0%  {background-color:red; left:0px; top:0px;}
 25% {background-color:yellow; left:200px; top:0px;}
 50% {background-color:blue; left:200px; top:200px;}
 75% {background-color:green; left:0px; top:200px;}
 100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<p><b>Animation Delay</b></p>
<div></div>
</body>
</html>
```

Output



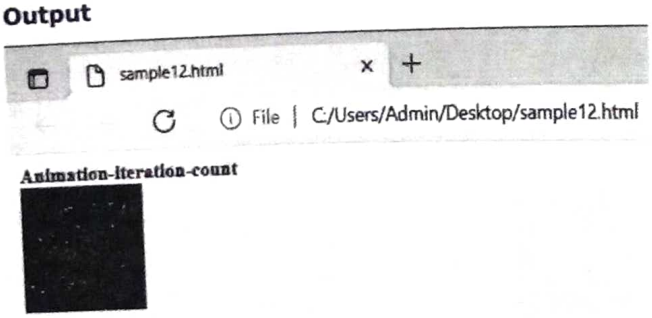
The animation-iteration-count

- The animation-iteration-count property specifies the number of times an animation should run.
- The following example will run the animation 3 times before it stops:

Program 1.25.3

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background-color: red;
 position: relative;
 animation-name: example;
 animation-duration: 4s;
 animation-iteration-count: 3;
}
@keyframes example {
 0%  {background-color:red; left:0px; top:0px;}
 25% {background-color:yellow; left:200px; top:0px;}
 50% {background-color:blue; left:200px; top:200px;}
 75% {background-color:green; left:0px; top:200px;}
 100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<b>Animation-iteration-count</b>
<div></div>
</body>
</html>
```

Output



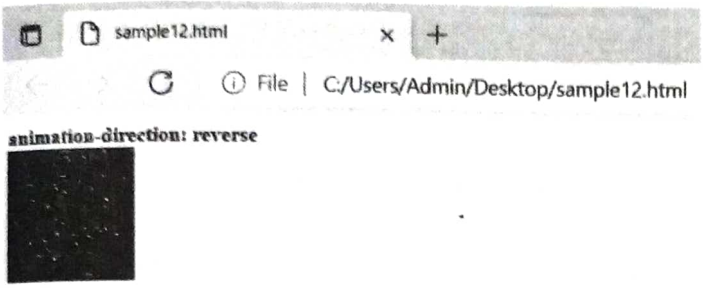
Animation direction property

- The animation-direction property specifies whether an animation should be played forwards, backwards or in alternate cycles.
- The animation-direction property can have the following values:
  - normal - The animation is played as normal (forwards). This is default
  - reverse - The animation is played in reverse direction (backwards)
  - alternate - The animation is played forwards first, then backwards
  - alternate-reverse - The animation is played backwards first, then forwards
- The following example will run the animation in reverse direction (backwards):

Program 1.25.4

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background-color: red;
 position: relative;
 animation-name: example;
 animation-duration: 4s;
 animation-direction: reverse;
}
@keyframes example {
 0%  {background-color:red; left:0px; top:0px;}
 25% {background-color:yellow; left:200px; top:0px;}
 50% {background-color:blue; left:200px; top:200px;}
 75% {background-color:green; left:0px; top:200px;}
 100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<b>animation-direction: reverse</b>
<div></div>
</body>
</html>
```

Output



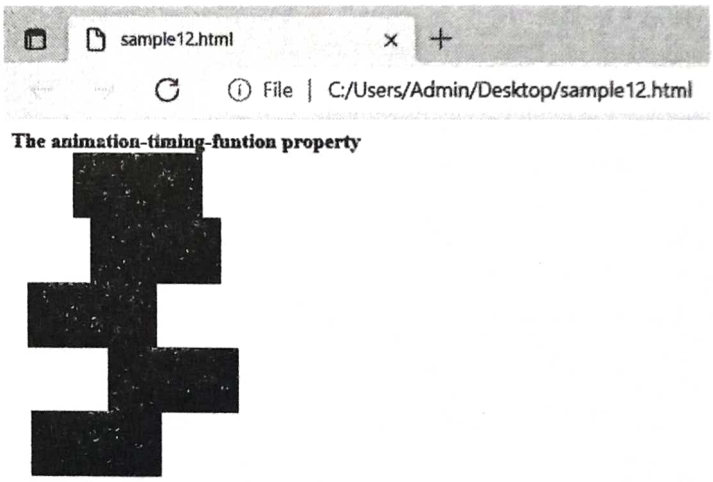
Animation-timing-function property

- The animation-timing-function property specifies the speed curve of the animation.
- The animation-timing-function property can have the following values:
  - **ease** - Specifies an animation with a slow start, then fast, then end slowly (this is default)
  - **linear** - Specifies an animation with the same speed from start to end
  - **ease-in** - Specifies an animation with a slow start
  - **ease-out** - Specifies an animation with a slow end
  - **ease-in-out** - Specifies an animation with a slow start and end
  - **cubic-bezier(n,n,n,n)** - Lets you define your own values in a cubic-bezier function
- The following example shows some of the different speed curves that can be used:

Program 1.25.5

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 50px;
 background-color: red;
 font-weight: bold;
 position: relative;
 animation: mymove 5s infinite;
}
#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}
@keyframes mymove {
 from {left: 0px;}
 to {left: 300px;}
}
</style>
</head>
<body>
<b>The animation-timing-funtion property </b>
<div id="div1">linear</div>
<div id="div2">ease</div>
<div id="div3">ease-in</div>
<div id="div4">ease-out</div>
<div id="div5">ease-in-out</div>
</body>
</html>
```

Output



Animation-fill-mode property

- The animation-fill-mode property specifies a style for the target element when the animation is not playing (before it starts, after it ends, or both).
- The animation-fill-mode property can have the following values:
  - **none** - Default value. Animation will not apply any styles to the element before or after it is executing.
  - **forwards** - The element will retain the style values that is set by the last keyframe (depends on animation-direction and animation-iteration-count).
  - **backwards** - The element will get the style values that is set by the first keyframe (depends on animation-direction), and retain this during the animation-delay period.

- **both** - The animation will follow the rules for both forwards and backwards, extending the animation properties in both directions.

- The following example lets the element retain the style values from the last keyframe when the animation ends:

Program 1.25.6

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background: red;
 position: relative;
 animation-name: example;
 animation-duration: 3s;
 animation-fill-mode: forwards;
}
@keyframes example {
 from {top: 0px;}
 to {top: 200px; background-color: blue;}
}
</style>
</head>
<body>
<p>Let the div element retain the style values from the last keyframe when the animation ends:</p>
<div></div>
<p>The animation-fill-mode property</p>
</body>
</html>
```

Output

Let the div element retain the style values from the last keyframe when the animation ends:

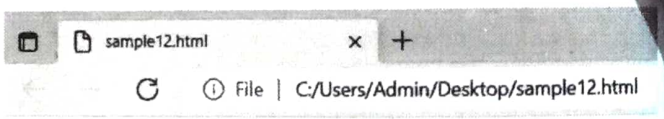
The animation-fill-mode property

Animation Shorthand Property

Program 1.25.7

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width: 100px;
 height: 100px;
 background-color: red;
 position: relative;
 animation: myfirst 5s linear 2s infinite alternate;
}

@keyframes myfirst {
 0%  {background-color:red; left:0px; top:0px;}
 25% {background-color:yellow; left:200px; top:0px;}
 50% {background-color:blue; left:200px; top:200px;}
 75% {background-color:green; left:0px; top:200px;}
 100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<p>Shorthand animation property</p>
<div></div>
</body>
</html>
```

Output



Shorthand animation property

1.26 BOOTSTRAP

GQ. What is Bootstrap ?

- Bootstrap is a framework for developing responsive and mobile friendly websites.
- It is free and easy to download.

- Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open source product in August 2011 on GitHub.
- Bootstrap is a front-end framework that helps easier and faster web development.
- It includes various HTML and CSS based design templates for buttons, tables, forms, panels and many more.
- It solves many problems which we had once, one of which is the cross-browser compatibility issue.
- It can also use JavaScript plug-ins.
- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
- It facilitates you to create responsive designs.

1.26.1 Uses of Bootstrap

Following are the main advantage of Bootstrap:

1. It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
2. It facilitates users to develop a responsive website.
3. It designes the responsive web pages for mobile devices too.
4. It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

Bootstrap package contains

- **Scaffolding :** Bootstrap provides a basic structure with Grid System, link styles, and background.
- **CSS :** Bootstrap comes with the feature of global CSS settings, fundamental HTML elements style and an advanced grid system.
- **Components :** Bootstrap contains a lot of reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.
- **JavaScript Plugins :** Bootstrap also contains a lot of custom jQuery plugins. You can easily include them all, or one by one.
- **Customize :** Bootstrap components are customizable and you can customize Bootstrap's components, LESS variables, and jQuery plugins to get your own style.

1.27 CSS VS. BOOTSTRAP

The main difference between Bootstrap and CSS is that the Bootstrap is a front-end framework while CSS is a style sheet language.

Cascading Style Sheet(CSS)	Bootstrap
CSS is developed by Hakon Wium Lie, Bert Bos, World Wide Web 17 December 1996.	Bootstrap is developed by Mark Otto, Jacob Thornton 19 August 2011.
CSS represent the style and the appearance of content like font, color, margin, padding, etc.	Bootstrap is a free and open-source CSS Framework that is used for developing responsive website.
CSS does not have a grid system.	Bootstrap is based on-grid system.
It currently working on CSS3 which is the latest version of CSS.	Bootstrap currently working on Bootstrap 5 which is the latest version of Bootstrap.
CSS does not provide responsive pages or website.	In Bootstrap we can design a responsive website or webpages.
CSS is more complex than Bootstrap because there is no pre-defined class and design.	Bootstrap is easy to understand and it has much pre-design class.
In CSS, we have to write code from scratch.	In Bootstrap, we can add pre-defined class into the code without writing code.

1.28 HOW TO USE BOOTSTRAP IN WEBPAGE ?

There are two ways to include Bootstrap in the website.

Include Bootstrap from the CDN link.

Bootstrap CSS Library

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T"crossorigin="anonymous">
```

Bootstrap CSS Library

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T"crossorigin="anonymous">
```

jQuery Library

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1
Pi6jizo"
crossorigin="anonymous">
</script>
```

JS Library

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.j
s"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF8
6dIHNDz0W1"
crossorigin="anonymous">
</script>
```

Latest compiled JavaScript Library

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B0
7jRM"
crossorigin="anonymous">
</script>
```

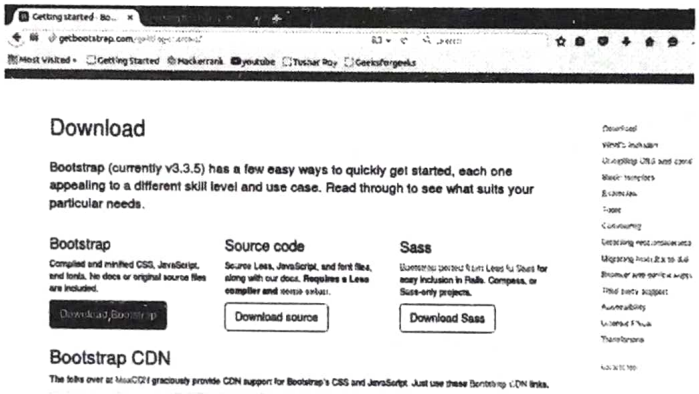
Copy the links and paste them into the head section of the HTML code.

Example

```
<!DOCTYPE html>
<html lang="en">

<head>
        <title>Bootstrap Example</title>
        <meta charset="utf-8">
        <meta name="viewport"
content="width=device-width, initial-scale=1">

        <!-- Bootstrap CSS library -->
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/
bootstrap.min.css"
                integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhc
Wr7x9JvoRxT2MZw1T"
                crossorigin="anonymous">

        <!-- jQuery library -->
        <script src="https://code.jquery.com/jquery-
3.3.1.slim.min.js"
                integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRv
H+8abtTE1Pi6jizo"
                crossorigin="anonymous">
                </script>

        <!-- JavaScript library -->
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/u
md/popper.min.js"
                integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZ
wrnQq4sF86dIHNDz0W1"
                crossorigin="anonymous">
                </script>

        <!-- Latest compiled JavaScript library -->
        <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bo
otstrap.min.js"
                integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf
4x0xIM+B07jRM"
                crossorigin="anonymous">
                </script>
</head>

<body>
        <div class="container text-center">
                <h1 class="text-
success">Bootstrap</h1>
                <p>Welcome to Bootstrap</p>
        </div>
</body>

</html>
```

Download Bootstrap from getbootstrap.com and use it

- Go to www.getbootstrap.com and click Getting Started. Click on the Download Bootstrap button.



- A.zip file would get downloaded. Extract the zip file and go in the distribution folder. It contains two folders named as CSS and JS.

```
<link rel="stylesheet" type="text/css"
href="css/bootstrap.min.css">
<script src="js/bootstrap.min.js"> </script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery
.min.js">
</script>
```

- Add the file link to the HTML document and then open the web page using web browsers.

Example

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content=
    "width=device-width, initial-scale=1">

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

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

<body>
  <div class="container text-center">
    <h1 class="text-success"> Bootstrap </h1>
    <p>Welcome to Bootstrap </p>
  </div>
</body>

</html>
```

☞ **First Bootstrap Example (with responsive fixed width container)**

```
<!DOCTYPE html>
<html lang="en">
<head>
 <title>This is a Bootstrap example</title>
   <meta name="viewport" content="width=device-width,
initial-scale=1">
 <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bo
otstrap.min.css">
</head>
<body>
<div class="container">
  <h1> First Bootstrap web page</h1>
  <p>Write your text here..</p>
</div>
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquer
y.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boot
strap.min.js"></script>
</body>
</html>
```

Output

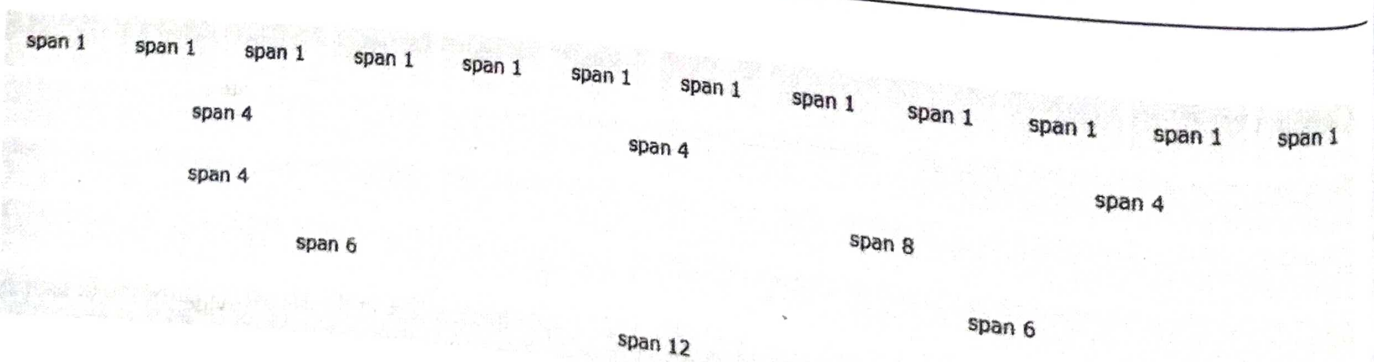
First Bootstrap web page

Write your text here..

1.29 BOOTSTRAP GRID SYSTEM

GQ. What is Bootstrap Grid System?

- Bootstrap grid system provides an easy and powerful way to create responsive layouts of all shapes and sizes.
- It is built with flexbox with mobile-first approach.
- Bootstrap's grid system allows up to 12 columns across the page.
- Bootstrap's grid system is responsive, and the columns will re-arrange depending on the screen size. On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items were stacked on top of each other.



☞ Basic Structure of a Bootstrap Grid

The following is a basic structure of a Bootstrap grid:

```
<div class="row">
 <div class="col-*-*"></div>
 <div class="col-*-*"></div>
</div>
<div class="row">
 <div class="col-*-*"></div>
 <div class="col-*-*"></div>
 <div class="col-*-*"></div>
</div>
<div class="row">
 ...
</div>
```

First; create a row ( ). Then, add the desired number of columns (tags with appropriate .col-*-* classes). Note that numbers in .col-*-* should always add up to 12 for each row.

☞ Bootstrap 4 Grid Classes

There are 5 classes in Bootstrap 4 grid system.

- **.col-** (extra small devices - screen width less than 576px)
- **.col-sm-** (small devices - screen width equal to or greater than 576px)
- **.col-md-** (medium devices - screen width equal to or greater than 768px)
- **.col-lg-** (large devices - screen width equal to or greater than 992px)
- **.col-xl-** (xlarge devices - screen width equal to or greater than 1200px)

☞ Creating Two Column Layouts

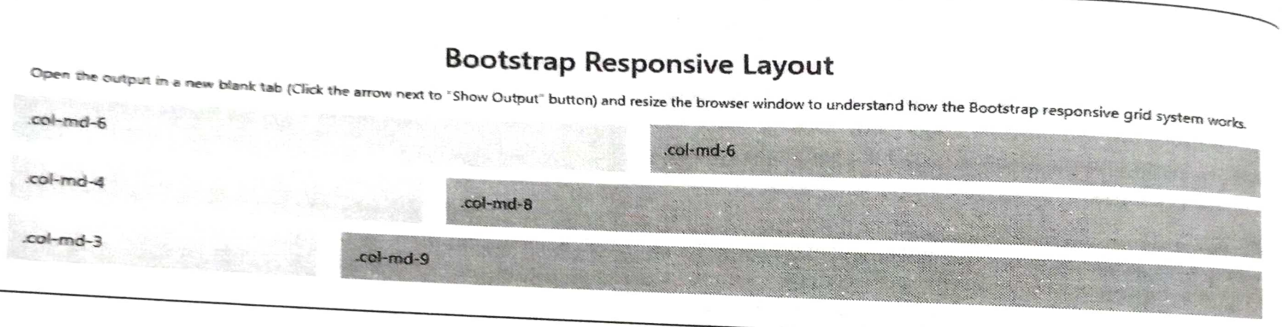
The following example will shows how to create two column layouts for medium, large and extra large devices like tables, laptops and desktops etc. However, on mobile phones (screen width less than 768px), the columns will automatically become horizontal (2 rows, 1 column).

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Two Column Grid Layouts for Tablets and Desktops</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  .demo-content{
    padding: 15px;
    font-size: 18px;
    background: #dbdfe5;
    margin-bottom: 15px;
  }
  .demo-content.bg-alt{
    background: #abb1b8;
  }
</style>
</head>
<body>
  <h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>
   <div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</div>
  <div class="container">
    <!--Row with two equal columns-->
    <div class="row">
      <div class="col-md-6">
        <div class="demo-content">.col-md-6</div>
      </div>
      <div class="col-md-6">
        <div class="demo-content bg-alt">.col-md-6</div>
      </div>
    </div>

    <!--Row with two columns divided in 1:2 ratio-->
    <div class="row">
      <div class="col-md-4">
```

```
        <div class="demo-content">.col-md-4</div>
      </div>
      <div class="col-md-8">
        <div class="demo-content bg-alt">.col-md-
8</div>
      </div>
    </div>

    <!--Row with two columns divided in 1:3 ratio-->
    <div class="row">
      <div class="col-md-3">
        <div class="demo-content">.col-md-3</div>
      </div>
      <div class="col-md-9">
        <div class="demo-content bg-alt">.col-md-
9</div>
      </div>
    </div>
  </div>
</body>
</html>
```

Output



- In a grid layout, content must be placed inside the columns (.col and .col-*) and only columns may be the immediate children of rows (.row). Also, rows should be placed inside a container (either fixed or fluid) for proper padding and alignment.
- Since the Bootstrap grid system is based on 12 columns, therefore to keep the columns in a one line (i.e. side by side), the sum of the grid column numbers within a single row should not be greater than 12.

Creating Three Column Layouts

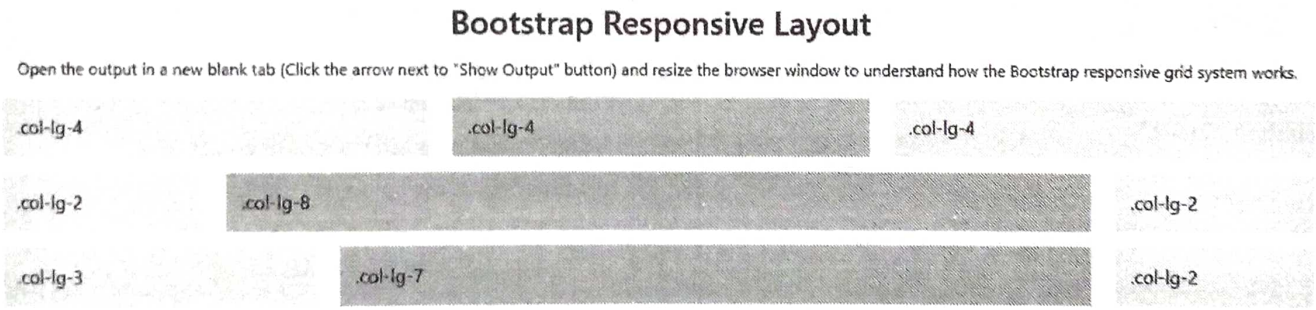
The following example will typically create three column layouts for laptops and desktops screens. It also works in tablets in landscape mode if screen resolution is more than or equal to 992 pixels (e.g. Apple iPad). However, in portrait mode the grid columns will be horizontal as usual.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Three Column Grid Layouts for Tablets
(landscape) and Desktops</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  .demo-content{
    padding: 15px;
    font-size: 18px;
    background: #dbdfe5;
    margin-bottom: 15px;
  }
  .demo-content.bg-alt{
    background: #abb1b8;
  }
</style>
</head>
<body>
  <h2 class="text-center mt-3">Bootstrap Responsive
Layout</h2>
```

```
    <div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</div>
    <div class="container">
      <!--Row with three equal columns-->
      <div class="row">
        <div class="col-lg-4">
          <div class="demo-content">.col-lg-4</div>
        </div>
        <div class="col-lg-4">
          <div class="demo-content bg-alt">.col-lg-4</div>
        </div>
        <div class="col-lg-4">
          <div class="demo-content">.col-lg-4</div>
        </div>
      </div>

      <!--Row with three columns divided in 1:4:1 ratio-->
      <div class="row">
        <div class="col-lg-2">
          <div class="demo-content">.col-lg-2</div>
        </div>
        <div class="col-lg-8">
          <div class="demo-content bg-alt">.col-lg-8</div>
        </div>
        <div class="col-lg-2">
          <div class="demo-content">.col-lg-2</div>
        </div>
      </div>

      <!--Row with three columns divided unevenly-->
      <div class="row">
        <div class="col-lg-3">
          <div class="demo-content">.col-lg-3</div>
        </div>
        <div class="col-lg-7">
          <div class="demo-content bg-alt">.col-lg-7</div>
        </div>
        <div class="col-lg-2">
          <div class="demo-content">.col-lg-2</div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Output



☞ Bootstrap Auto-layout Columns

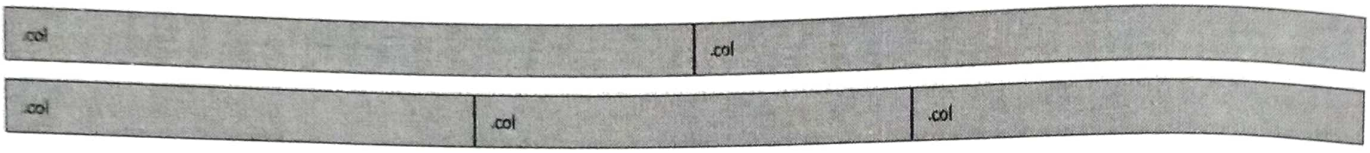
You can also create *equal width columns* for all devices (x-small, small, medium, large, x-large, and xx-large) through simply using the class .col, without specifying any column number.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Auto-layout Columns</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  .row{
    margin-top: 1rem;
```

```
    background: #dbdfe5;   }
  .col{
    padding: 10px 15px;
    background: #abb1b8;
    border: 1px solid rgba(0,0,0,0.5)
  }
</style>
</head>
<body>
<div class="container">
  <!--Row with two equal columns-->
  <div class="row">
    <div class="col">.col</div>
    <div class="col">.col</div>
  </div>
  <!--Row with three equal columns-->
  <div class="row">
    <div class="col">.col</div>
    <div class="col">.col</div>
    <div class="col">.col</div>
  </div>
</div>
</body>
</html>
```

Output



☞ Nesting of Grid Columns

The Bootstrap grid columns are also nestable, that means you can put rows and columns inside an existing column. However, the formula for placing the columns will be the same, i.e. the sum of column numbers should be equal to 12 or less within a single row.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Nested Rows and Columns</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  .main-content{
      min-height: 230px;
    background: #dbdfe5;
  }
  .sidebar-content{
    min-height: 100px;
    margin-bottom: 30px;
    background: #b4bac0;
  }
</style>
</head>
<body>
<div class="container mt-3">
  <div class="row">
    <div class="col-sm-8">
      <div class="main-content"></div>
    </div>
    <div class="col-sm-4">
      <!--Nested rows within a column-->
      <div class="row">
        <div class="col-12">
          <div class="sidebar-content"></div>
        </div>
      </div>
      <div class="row">
        <div class="col-6">
          <div class="sidebar-content"></div>
        </div>
        <div class="col-6">
          <div class="sidebar-content"></div>
        </div>
      </div>
    </div>
  </div>
</div>
</body>
</html>
```

Output

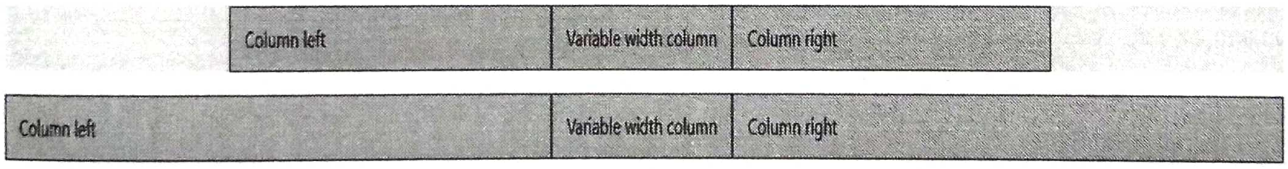
☞ Creating Variable Width Columns

You can use the col-{breakpoint}-auto classes to size columns based on the natural width of their **content.**

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Sizing Columns Based on the Width of their
Content</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
  .row{
    margin-top: 1rem;
    background: #dbdfe5;
  }
  [class^=col]{
    padding: 10px 15px;
    background: #abb1b8;
    border: 1px solid rgba(0,0,0,0.5)
  }
</style>
</head>
<body>
  <div class="container">
    <div class="row justify-content-md-center">
      <div class="col-md-3">Column left</div>
      <div class="col-md-auto">Variable width
column</div>
      <div class="col-md-3">Column right</div>
    </div>
    <div class="row">
      <div class="col">Column left</div>
      <div class="col-auto">Variable width
column</div>
      <div class="col">Column right</div>
    </div>
  </div>
  <p class="text-center mt-3"><strong>Note:</strong>
Enter some more content inside the variable width column to
understand how it works.</p>
</body>
</html>
```

Output



Note: Enter some more content inside the variable width column to understand how it works.

☞ Alignment of Grid Columns

You can use the flexbox alignment utilities to vertically and horizontally align grid columns inside a container.

▶ 1. Vertical Alignment of Grid Columns

You can use the classes .align-items-start, .align-items-center, and .align-items-end to align the grid columns vertically at the top, middle and bottom of a container, respectively.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Vertical Alignments of Grid
Columns</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
 .row{
      min-height: 10rem;
    margin-top: 1rem;
    background: #dbdfe5;   }
 .col{
    padding: 10px 15px;
    background: #abb1b8;
    border: 1px solid rgba(0,0,0,0.5)   }
</style>
</head>
<body>
<div class="container">
  <div class="row align-items-start">
    <div class="col">Column one</div>
    <div class="col">Column two</div>
    <div class="col">Column three</div>
  </div>
  <div class="row align-items-center">
    <div class="col">Column one</div>
    <div class="col">Column two</div>
    <div class="col">Column three</div>
  </div>
  </div>
</body>
</html>
```

Output

Column one	Column two	Column three

Column one	Column two	Column three

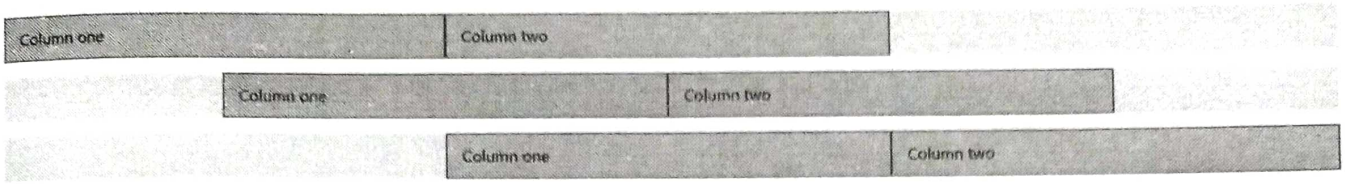
2. Horizontal Alignment of Grid Columns

You can use the classes .justify-content-start, .justify-content-center, and .justify-content-end to align the grid columns horizontally at the left, center and right of a container, respectively.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Horizontal Alignments of Grid
Columns</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
<style>
  /* Some custom styles to beautify this example */
 .row{
    margin-top: 1rem;
    background: #dbdfe5;
  }
  [class ^ =col-]{
    padding: 10px 15px;
    background: #abb1b8;
    border: 1px solid rgba(0,0,0,0.5)   }
</style>
</head>
<body>
<div class="container">
  <div class="row justify-content-start">
    <div class="col-4">Column one</div>
    <div class="col-4">Column two</div>
  </div>
  <div class="row justify-content-center">
    <div class="col-4">Column one</div>
    <div class="col-4">Column two</div>
  </div>
  <div class="row justify-content-end">
    <div class="col-4">Column one</div>
    <div class="col-4">Column two</div>
  </div>
</div>
</body>
</html>
```

Output



1.30 BOOTSTRAP RESPONSIVE LAYOUT

GQ. What is Responsive Web Design.

- Responsive web design is a process of designing and building websites to provide better accessibility and optimal viewing experience to the user by optimizing it for different devices.
- With the growing trend of smart phones and tablets, it has become almost unavoidable to ignore the optimization of sites for mobile devices. Responsive web design is a preferable alternative and an efficient way to target a wide range of devices with much less efforts.
- Responsive layouts automatically adjust and adapts to any device screen size, whether it is a desktop, a laptop, a tablet, or a mobile phone

1.30.1 Creating Responsive Layout with Bootstrap

- With the Bootstrap powerful mobile first flexbox grid system creating the responsive and mobile friendly websites and applications has become much easier.
- Bootstrap is responsive and mobile friendly from the start. Its six tier grid classes provides better control over the layout as well as how it will be rendered on different types of devices like mobile phones, tablets, laptops and desktops, large screen devices, and so on.

Example

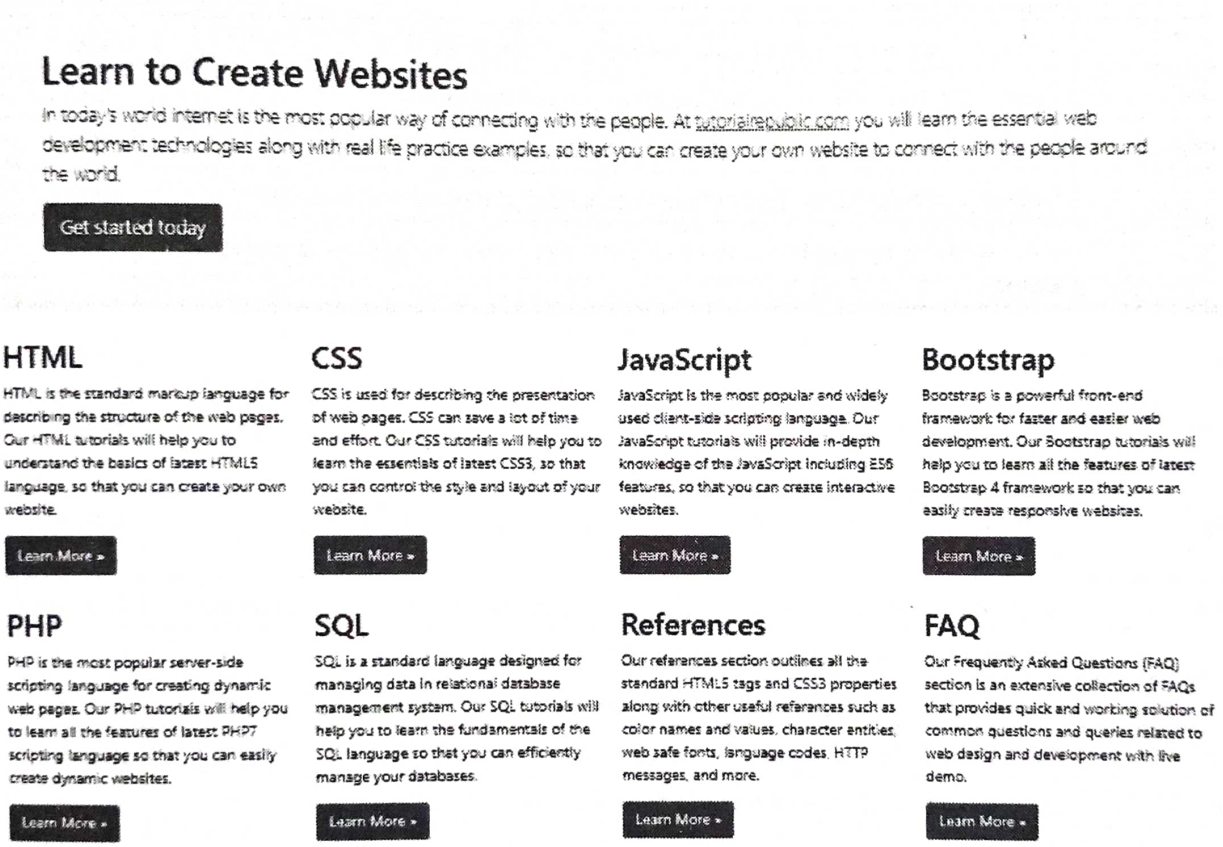
- The following example will create a responsive layout that is rendered as 4 column layout in extra-large devices (viewport ≥ 1200px), and 3 column layout in large devices (992px ≤ viewport < 1200px), whereas 2 column layout in medium devices (768px ≤ viewport < 992px), and 1 column layout in small and extra-small devices (viewport < 768px).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap 4 Responsive Layout</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-
dark">
  <div class="container-fluid">
    <a href="#" class="navbar-brand">Tutorial
Republic</a>
    <button type="button" class="navbar-toggler" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse"
id="navbarCollapse">
      <div class="navbar-nav">
        <a href="#" class="nav-item nav-link
active">Home</a>
        <a href="#" class="nav-item nav-
link">Services</a>
        <a href="#" class="nav-item nav-
link">About</a>
        <a href="#" class="nav-item nav-
link">Contact</a>
      </div>
      <div class="navbar-nav ms-auto">
        <a href="#" class="nav-item nav-
link">Login</a>
        <a href="#" class="nav-item nav-
link">Register</a>
      </div>
```

```
    </div>
  </div>
</nav>
<div class="container">
  <div class="p-5 my-4 bg-light rounded-3">
    <h1>Learn to Create Websites</h1>
    <p class="lead">In today's world internet is the most popular way of connecting with the people. At <a href="https://www.tutorialrepublic.com" class="text-success" target="_blank">tutorialrepublic.com</a> you will learn the essential web development technologies along with real life practice examples, so that you can create your own website to connect with the people around the world.</p>
    <p><a href="https://www.tutorialrepublic.com" target="_blank" class="btn btn-success btn-lg">Get started today</a></p>
  </div>
  <div class="row g-3">
    <div class="col-md-6 col-lg-4 col-xl-3">
      <h2>HTML</h2>
      <p>HTML is the standard markup language for describing the structure of the web pages. Our HTML tutorials will help you to understand the basics of latest HTML5 language, so that you can create your own website.</p>
      <p><a href="https://www.tutorialrepublic.com/html-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
    </div>
    <div class="col-md-6 col-lg-4 col-xl-3">
      <h2>CSS</h2>
      <p>CSS is used for describing the presentation of web pages. CSS can save a lot of time and effort. Our CSS tutorials will help you to learn the essentials of latest CSS3, so that you can control the style and layout of your website.</p>
      <p><a href="https://www.tutorialrepublic.com/css-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
    </div>
    <div class="col-md-6 col-lg-4 col-xl-3">
      <h2>JavaScript</h2>
      <p>JavaScript is the most popular and widely used client-side scripting language. Our JavaScript tutorials will provide in-depth knowledge of the JavaScript including ES6 features, so that you can create interactive websites.</p>
      <p><a href="https://www.tutorialrepublic.com/javascript-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
    </div>
    <div class="col-md-6 col-lg-4 col-xl-3">
      <h2>Bootstrap</h2>
      <p>Bootstrap is a powerful front-end framework for faster and easier web development. Our Bootstrap tutorials will help you to learn all the features of latest Bootstrap 4 framework so that you can easily create responsive websites.</p>
      <p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
    </div>
    <div class="col-md-6 col-lg-4 col-xl-3">
      <h2>PHP</h2>
      <p>PHP is the most popular server-side scripting language for creating dynamic web pages. Our PHP tutorials will help you to learn all the features of latest PHP7 scripting language so that you can easily create dynamic websites.</p>
      <p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
    </div>
    <div class="col-md-6 col-lg-4 col-xl-3">
      <h2>SQL</h2>
      <p>SQL is a standard language designed for managing data in relational database management system. Our SQL tutorials will help you to learn the fundamentals of the SQL language so that you can efficiently manage your databases.</p>
      <p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
    </div>
    <div class="col-md-6 col-lg-4 col-xl-3">
      <h2>References</h2>
      <p>Our references section outlines all the standard HTML5 tags and CSS3 properties along with other useful references such as color names and values, character entities, web safe fonts, language codes, HTTP messages, and more.</p>
      <p><a href="https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
    </div>
    <div class="col-md-6 col-lg-4 col-xl-3">
      <h2>FAQ</h2>
      <p>Our Frequently Asked Questions (FAQ) section is an extensive collection of FAQs that provides quick and working solution of common questions and queries related to web design and development with live demo.</p>
```

```
        <p><a
href="https://www.tutorialrepublic.com/twitter-bootstrap-
tutorial/" target="_blank" class="btn btn-success">Learn
More &raquo;</a></p>
      </div>
    </div>
    <hr>
    <footer>
      <div class="row">
        <div class="col-md-6">
          <p>Copyright © 2021 Tutorial Republic</p>
        </div>
        <div class="col-md-6 text-md-end">
          <a href="#" class="text-dark">Terms of
Use</a>
          <span class="text-muted mx-2">|</span>
          <a href="#" class="text-dark">Privacy
Policy</a>
        </div>
      </div>
    </footer>
  </div>
  </body>
  </html>
```

Output



1.31 BOOTSTRAP CLASSES

Bootstrap CSS classes along with description and category are described in following table.

Class	Description	Category
.active	Adds a grey background color to the table row (<tr> or table cell (<td>) (same color used on hover)	Tables
.active	Adds a gray background color to the active link in a default navbar. Adds a black background and a white color to the current link inside an inverted navbar.	Navbar
.active	Adds a blue background color to the active list item in a list group	List Groups

Class	Description	Category
.active	Adds a blue background color to simulate a "pressed" button	Buttons
.active	Animates a striped progress bar	Progress Bars
.active	Adds a blue background color to the active dropdown item in a dropdown	Dropdowns
.active	Adds a blue background color to the active pagination link (to highlight the current page	Pagination
.affix	The Affix plugin allows an element to become affixed (locked/sticky) to an area on the page. It toggles position:fixed on and off	Affix
.alert	Creates an alert message box	Alerts
.alert-danger	Red alert box. Indicates a dangerous or potentially negative action	Alerts
.alert-dismissible	Together with the .close class, this class is used to close the alert	Alerts
.alert-info	Light-blue alert box. Indicates some information	Alerts
.alert-link	Used on links inside alerts to add matching colored links	Alerts
.alert-success	Green alert box. Indicates a successful or positive action	Alerts
.alert-warning	Yellow alert box. Indicates caution should be taken with this action	Alerts
.badge	Creates a circular badge (grey circle - often used as a numerical indicator)	Badges
.bg-danger	Adds a red background color to an element. Represents danger or a negative action	Helpers
.bg-info	Adds a light-blue background color to an element. Represents some information	Helpers
.bg-primary	Adds a blue background color to an element. Represents something important	Helpers
.bg-success	Adds a green background color to an element. Indicates success or a positive action	Helpers
.bg-warning	Adds a yellow background color to an element. Represents a warning or a negative action	Helpers
.breadcrumb	A pagination. Indicates the current page's location within a navigational hierarchy	Pagination
.btn	Creates a basic button (gray background and rounded corners)	Buttons
.btn-block	Creates a block level button that spans the entire width of the parent element	Buttons
.btn-danger	Red button. Indicates danger or a negative action	Buttons
.btn-default	Default button. White background and grey border	Buttons
.btn-group	Groups buttons together on a single line	Button Groups
.btn-group-justified	Makes a group of buttons span the entire width of the screen	Button Groups
.btn-group-lg	Large button group (makes all buttons in a button group larger - increased font-size and padding)	Button Groups
.btn-group-sm	Small button group (makes all buttons in a button group smaller)	Button Groups
.btn-group-xs	Extra small button group (makes all buttons in a button group extra small)	Button Groups
.btn-group-vertical	Makes a button group appear vertically stacked	Button Groups
.btn-info	Light-blue button. Represents information	Buttons
.btn-link	Makes a button look like a link (get button behavior)	Buttons

Class	Description	Category
.btn-lg	Large button	Buttons
.btn-primary	Blue button.	Buttons
.btn-sm	Small button	Buttons
.btn-success	Green button. Indicates success or a positive action	Buttons
.btn-warning	Yellow button. Represents warning or a negative action	Buttons
.btn-xs	Extra small button	Buttons
.caption	Adds a caption text inside a .thumbnail	Images
.caret	Creates a caret arrow icon , which indicates that the button is a dropdown	Dropdowns
.carousel	Creates a carousel (slideshow)	Carousel
.carousel-caption	Creates a caption text for each slide in the carousel	Carousel
.carousel-control	Container for next and previous links	Carousel
.carousel-indicators	Adds little dots/indicators at the bottom of each slide (which indicates how many slides there is in the carousel, and which slide the user are currently viewing)	Carousel
.carousel-inner	Container for slide items	Carousel
.center-block	Centers any element (Sets an element to display:block with margin-right:auto and margin-left:auto)	Helpers
.checkbox	Container for checkboxes	Inputs
.checkbox-inline	Makes multiple checkboxes appear on the same line	Inputs
.clearfix	Clears floats	Helpers
.close	Indicates a close icon	Helpers
.col-*-*	Responsive grid (span 1-12 column). Extra small devices Phones (< 768px), Small devices Tablets (≥768px), Medium devices Desktops (≥992px), Large devices Desktops (≥1200px). Column values can be 1-12.	Grid
.col-*-offset-*	Move columns to the right. These classes increase the left margin of a column by * columns	Grid
.col-*-pull-*	Changes the order of the grid columns	Grid
.col-*-push-*	Changes the order of the grid columns	Grid
.collapse	Indicates collapsible content - which can be hidden or shown on demand	Collapse
.collapse in	Show the collapsible content by default	Collapse
.container	Fixed width container with widths determined by screen sites. Equal margin on the left and right.	Containers
.container-fluid	A container that spans the full width of the screen	Containers
.control-label	Allows a label to be used for form validation	Forms
.danger	Adds a red background to the table row (<tr> or table cell (<td>). Indicates a dangerous or potentially negative action	Tables
.disabled	Disables a button (adds opacity and a "no-parking-sign" icon on hover)	Buttons
.disabled	Disables a dropdown item (adds a grey text color and a "no-parking-sign" icon on hover)	Dropdowns

Class	Description	Category
.disabled	Disables a pagination link (cannot be clicked - adds a grey text color and a "no-parking-sign" icon on hover)	Pagination
.disabled	Disables a list item in a list group (cannot be clicked - adds a grey background color and a "no-parking-sign" icon on hover)	List Groups
.divider	Used to separate links in the dropdown menu with a thin horizontal border	Dropdowns
.dl-horizontal	Lines up the terms <dt> and descriptions <dd> in <dl> elements side-by-side. Starts off like default <dl>s, but when the browser window expands, it will line up side-by-side	Typography
.dropdown	Creates a toggleable menu that allows the user to choose one value from a predefined list	Dropdowns
.dropdown-header	Used to add headers inside the dropdown menu	Dropdowns
.dropdown-menu	Adds the default styles for the dropdown menu container	Dropdowns
.dropdown-menu-right	Right-aligns a dropdown menu	Dropdowns
.dropdown-toggle	Used on the button that should hide and show (toggle) the dropdown menu	Dropdowns
.dropup	Indicates a dropup menu (upwards instead of downwards)	Dropdowns
.embed-responsive	Container for embedded content. Makes videos or slideshows scale properly on any device	Images
.embed-responsive-16by9	Container for embedded content. Creates an 16:9 aspect ratio embedded content	Images
.embed-responsive-4by3	Container for embedded content. Creates an 4:3 aspect ratio embedded content	Images
.embed-responsive-item	Used inside .embed-responsive. Scales the video nicely to the parent element	Images
.fade	Adds a fading effect when closing an alert box	Alerts
.form-control	Used on input, textarea, and select elements to span the entire width of the page and make them responsive	Forms
.form-control-feedback	Form validation class	Inputs 2
.form-control-static	Adds plain text next to a form label within a horizontal form	Inputs 2
.form-group	Container for form input and label	Forms
.form-inline	Makes a <form> left-aligned with inline-block controls (This only applies to forms within viewports that are at least 768px wide)	Forms
.form-horizontal	Aligns labels and groups of form controls in a horizontal layout	Forms
.glyphicon	Creates an icon. Bootstrap provides 260 free glyphicons from the Glyphicons Halflings set	Glyphicons
.has-danger	Adds a red color to the label and a red border to the input, as well as an error icon inside the input (used together with .has-feedback)	Forms
.has-feedback	Adds feedback icons for inputs (checkmark, warning and error signs)	Forms

Class	Description	Category
.has-success	Adds a green color to the label and a green border to the input, as well as a checkmark icon inside the input (used together with .has-feedback)	Forms
.has-warning	Adds a yellow/orange color to the label and a yellow/orange border to the input, as well as a checkmark icon inside the input (used together with .has-feedback)	Forms
.help-block	A block of help text that breaks onto a new line and may extend beyond one line.	Input Sizing
.hidden	Forces an element to be hidden (display:none)	Helpers
.hidden-*	Hides content depending on screen size	Helpers
.hide	Deprecated. Use .hidden instead	Helpers
.h1 - .h6	Makes an element look like a heading of the chosen class (h1-h6)	Typography
.icon-bar	Used in the navbar to create a hamburger menu (three horizontal bars)	Navbar
.icon-next	Unicode icon (arrow pointing right), used in carousels. This is often replaced with a glyphicon	Carousel
.icon-prev	Unicode icon (arrow pointing left), used in carousels. This is often replaced with a glyphicon	Carousel
.img-circle	Shapes an image to a circle (not supported in IE8 and earlier)	Images
.img-responsive	Makes an image responsive	Images
.img-rounded	Adds rounded corners to an image	Images
.img-thumbnail	Shapes an image to a thumbnail (borders)	Images
.in	Fades in tabs	Tabs
.info	Adds a light-blue background to the table row (<tr> or table cell (<td>). Indicates a neutral informative change or action	Tables
.initialism	Displays the text inside an <abbr> element in a slightly smaller font size	Typography
.input-group	Container to enhance an input by adding an icon, text or a button in front or behind it as a "help text"	Inputs
.input-group-lg	Large input group	Inputs
.input-group-sm	Small input group	Inputs
.input-group-addon	Together with the .input-group class, this class makes it possible to add an icon or help text next to the input field	Inputs
.input-group-btn	Together with the .input-group class, this class attaches a button next to an input. Often used as a search bar	Inputs
.input-lg	Large input field	Input Sizing
.input-sm	Small input field	Input Sizing
.invisible	Makes an element invisible (visibility:hidden). Note: Even though the element is invisible, it will take up space on the page	Helpers
.item	Class added to each carousel item. May be text or images	Carousel
.jumbotron	Creates a padded grey box with rounded corners that enlarges the font sizes of the text inside it. Creates a big box for calling extra attention to some special content or information	Jumbotron

Class	Description	Category
.label	Adds a grey rounded box to an element. Provides additional information about something (e.g. "New")	Labels
.label-danger	Red label	Labels
.label-info	Light-blue label	Labels
.label-success	Green label	Labels
.label-warning	Yellow label	Labels
.lead	Increase the font size and line height of a paragraph	Typography
.left	Used to identify the left carousel control	Carousel
.list-group	Creates a bordered list group for <li> elements	List Group
.list-group-item	Added to each <li> element in the list group	List Group
.list-group-item-heading	Creates a list group heading (used on other elements besides <li>)	List Group
.list-group-item-text	Used for item text inside the list group (used on other elements besides <li>)	List Group
.list-group-item-danger	Red background color for a list item in a list group	List Group
.list-group-item-info	Light-blue background color for a list item in a list group	List Group
.list-group-item-success	Green background color for a list item in a list group	List Group
.list-group-item-warning	Yellow background color for a list item in a list group	List Group
.list-inline	Places all list items on a single line (horizontal menu)	Tabs
.list-unstyled	Removes all default list-style (bullets, left margin, etc.) styling from a <ul> or <ol> list	Typography
.mark	Highlights text: Highlighted text	Typography
.media	Aligns media objects (like images or videos - often used for comments in a blog post etc)	Media Objects
.media-body	Text that should appear next to a media object	Media Objects
.media-heading	Creates a heading inside the media object	Media Objects
.media-list	Nested media lists	Media Objects
.media-object	Indicates a media object (image or video)	Media Objects
.modal	Identifies the content as a modal and brings focus to it	Modals
.modal-body	Defines the style for the body of the modal. Add any HTML markup here (p, img, etc)	Modals
.modal-content	Styles the modal (border, background-color, etc). Inside this, add the modal's header, body and footer, if needed	Modals
.modal-dialog	Sets the proper width and margin of the modal	Modals
.modal-footer	The footer of the modal (often contains an action button and a close button)	Modals
.modal-header	The header of the modal (often contains a title and a close button)	Modals

Class	Description	Category
.modal-lg	Large modal (wider than default)	Modals
.modal-open	Used on the <body> element to prevent page scrolling (overflow:hidden)	Modals
.modal-sm	Small modal (less width)	Modals
.modal-title	The title of the modal	Modals
.nav nav-tabs	Indicates a tabbed menu	Tabs
.nav nav-pills	Indicates a pill menu	Tabs
.nav .navbar-nav	Used on a <ul> container that contains the list items with links inside a navigation bar	Navbar
.nav-justified	Centers tabs/pills. Note that on screens smaller than 768px the items are stacked (content will remain centered)	Tabs
.nav-stacked	Vertically stack tabs or pills	Tabs
.nav-tabs	Creates a tabbed menu	Tabs
.navbar	Creates a navigation bar	Navbar
.navbar-brand	Added to a link or a header element inside the navbar to represent a logo or a header	Navbar
.navbar-btn	Vertically aligns a button inside a navbar	Navbar
.navbar-collapse	Collapses the navbar (hidden and replaced with a menu/hamburger icon on mobile phones and small tablets)	Navbar
.navbar-default	Creates a default navigation bar (light-grey background color)	Navbar
.navbar-fixed-bottom	Makes the navbar stay at the bottom of the screen (sticky/fixed)	Navbar
.navbar-fixed-top	Makes the navbar stay at the top of the screen (sticky/fixed)	Navbar
.navbar-form	Added to form elements inside the navbar to vertically center them (proper padding)	Navbar
.navbar-header	Added to a container element that contains the link/element that represent a logo or a header	Navbar
.navbar-inverse	Creates a black navigation bar (instead of light-grey)	Navbar
.navbar-left	Aligns nav links, forms, buttons, or text, in the navbar to the left	Navbar
.navbar-link	Styles an element to look like a link inside the navbar (anchors get proper padding and an underline on hover, while other elements like p or span gets a default hover effect - white color in an inversed navbar and a black color in a default navbar)	Navbar
.navbar-nav	Used on a <ul> container that contains the list items with links inside a navigation bar	Navbar
.navbar-right	Aligns nav links, forms, buttons, or text in the navbar to the right.	Navbar
.navbar-static-top	Removes left, top and right borders (rounded corners) from the navbar (default navbar has a gray border and a 4px border-radius by default)	Navbar
.navbar-text	Vertical align any elements inside the navbar that are not links (ensures proper padding)	Navbar

Class	Description	Category
.navbar-toggle	Styles the button that should open the navbar on small screens. Often used together with three .icon-bar classes to indicate a toggleable menu icon (hamburger/bars)	Navbar
.next	Used in the carousel control to identity the next control	Carousel
.next	Used to align pager buttons to the right side of the page (next button)	Pager
.page-header	Adds a horizontal line under the heading (+ adds some extra space around the element)	Page Header
.pager	Creates previous/next buttons (used on <ul> elements)	Pager
.pagination	Creates a pagination (Useful when you have a web site with lots of pages. Used on <ul> elements)	Pagination
.pagination-lg	Large pagination (each pagination link gets a font-size of 18px. Default is 14px)	Pagination
.pagination-sm	Small pagination (each pagination link gets a font-size of 12px. Default is 14px)	Pagination
.panel	Creates a bordered box with some padding around its content	Panels
.panel-body	Container for content inside the panel	Panels
.panel-collapse	Collapsible panel (toggle between hiding and showing panel(s))	Collapse
.panel-danger	Red panel. Indicates danger	Panels
.panel-info	Light-blue panel. Indicates information	Panels
.panel-success	Green panel. Indicates success	Panels
.panel-warning	Yellow panel. Indicates warning	Panels
.panel-footer	Creates a panel footer (light background color)	Panels
.panel-group	Used to group many panels together. This removes the bottom margin below each panel	Panels
.panel-heading	Creates a panel header (light background color)	Panels
.panel-title	Used inside a .panel-heading to adjust the styling of the text (removes margins and adds a font-size of 16px)	Panels
.popover	Popup-box that appears when the user clicks on an element	Popover
.pre-scrollable	Makes a <pre> element scrollable (max-height of 350px and provide a y-axis scrollbar)	Helpers
.prev	Used in carousels to indicate a "previous" link	Carousel
.previous	Used to align pager buttons to the left side of the page (previous button)	Pager
.progress	Container for progress bars	Progress Bars
.progress-bar	Creates a progress bar	Progress Bars
.progress-bar-danger	Red progress bar. Indicates danger	Progress Bars
.progress-bar-info	Light-blue progress bar. Indicates information	Progress Bars
.progress-bar-striped	Creates a striped progress bar	Progress Bars
.progress-bar-success	Green progress bar. Indicates success	Progress Bars
.progress-bar-warning	Yellow progress bar. Indicates warning	Progress Bars

Class	Description	Category
.pull-left	Float an element to the left	Helpers
.pull-right	Float an element to the right	Helpers
.right	Used to identify the right carousel control	Carousel
.row	Container for responsive columns	Grid
.row-no-gutters	Removes the gutters from a row and its columns	Grid
.show	Shows an element (display:block)	Helpers
.small	Creates a lighter, secondary text in any heading	Typography
.sr-only	Hides an element on all devices except for screen readers	Helpers
.sr-only-focusable	Hides an element on all devices except for screen readers	Helpers
.success	Adds a green background color to a table row (<tr> or table cell (<td>). Indicates success or a positive action	Tables
.tab-content	Used together with .tab-pane to creates toggleable/dynamic tabs/pills	Tabs
.tab-pane	Used together with .tab-content to creates toggleable/dynamic tabs/pills	Tabs
.table	Adds basic styling to a table (padding, bottom borders, etc)	Tables
.table-bordered	Adds borders on all sides of the table and cells	Tables
.table-condensed	Makes a table more compact by cutting cell padding in half	Tables
.table-hover	Creates a hoverable table (adds a grey background color on table rows on hover)	Tables
.table-responsive	Makes a table responsive (adds a horizontal scrollbar when needed)	Tables
.text-capitalize	Indicates capitalized text	Typography
.text-center	Center-aligns text	Typography
.text-danger	Red text color. Indicates danger	Typography
.text-hide	Hides text (helps replace an element's text content with a background image)	Typography
.text-info	Light-blue text color. Indicates information	Typography
.text-justify	Indicates justified text	Typography
.text-left	Aligns the text to the left	Typography
.text-lowercase	Changes text to lowercase	Typography
.text-muted	Grey text color	Typography
.text-nowrap	Prevents the text from wrapping	Typography
.text-primary	Blue text color	Typography
.text-right	Aligns text to the right	Typography
.text-success	Green text color. Indicates success	Typography
.text-uppercase	Makes text uppercase	Typography
.text-warning	Yellow/orange text color. Indicates warning	Typography
.thumbnail	Adds a border around an element (often images or videos) to make it look like a thumbnail	Images
.tooltip	Popup-box that appears when the user moves the mouse pointer over an element	Tooltip

Class	Description	Category
.visible-*	Deprecated as of v3.2.0. Used to show and/or hide content by device. Note: Use .hidden-* instead	Helpers
.visible-print-block	Displays the element (display:block) in print (pre)view	Helpers
.visible-print-inline	Displays the element (display:inline) in print (pre)view	Helpers
.visible-print-inline-block	Displays the element (display:inline-block) in print (pre)view	Helpers
.hidden-print	Hides the element (display:none) in print (pre)view	Helpers
.warning	Adds a yellow background color to the table row (<tr> or table cell (<td>). Indicates a warning	Tables
.well	Adds a rounded border around an element with a gray background color and some padding	Wells
.well-lg	Large well (more padding)	Wells
.well-sm	Small well (less padding)	Wells

1.32 BOOTSTRAP COMPONENTS

1.32.1 Bootstrap Buttons

Creating Buttons with Bootstrap

Buttons are the integral part of a website and application. They are used for various purposes like, submit or reset an HTML form, performing interactive actions such as showing or hiding something on a web page on click of the button, redirecting user to another page, and so on. Bootstrap provides a 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 or semantic. Button styles can be applied to any element. However, it is applied normally to the <a>, <input>, and <button> elements for the best rendering.

The following example will show you how to create different styles of buttons in Bootstrap:

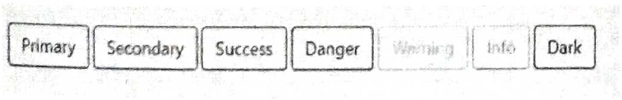
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Button Styles</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <button type="button" class="btn btn-primary">Primary</button>
  <button type="button" class="btn btn-secondary">Secondary</button>
  <button type="button" class="btn btn-success">Success</button>
  <button type="button" class="btn btn-danger">Danger</button>
  <button type="button" class="btn btn-warning">Warning</button>
  <button type="button" class="btn btn-info">Info</button>
  <button type="button" class="btn btn-dark">Dark</button>
  <button type="button" class="btn btn-light">Light</button>
  <button type="button" class="btn btn-link">Link</button>
</div>
</body>
</html>
```

Output

☞ Bootstrap Outline Buttons

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Outline Button Styles</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <button type="button" class="btn btn-outline-primary">Primary</button>
  <button type="button" class="btn btn-outline-secondary">Secondary</button>
  <button type="button" class="btn btn-outline-success">Success</button>
  <button type="button" class="btn btn-outline-danger">Danger</button>
  <button type="button" class="btn btn-outline-warning">Warning</button>
  <button type="button" class="btn btn-outline-info">Info</button>
  <button type="button" class="btn btn-outline-dark">Dark</button>
  <button type="button" class="btn btn-outline-light">Light</button>
</div>
</body>
</html>
```

Output

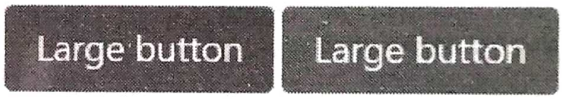


☞ Changing the Sizes of Buttons

Bootstrap gives option further to scaling a button up or down. To make buttons larger add an extra class .btn-lg to the buttons.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Large Buttons</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <button type="button" class="btn btn-primary btn-lg">Large button</button>
  <button type="button" class="btn btn-secondary btn-lg">Large button</button>
</div>
</body>
</html>
```

Output

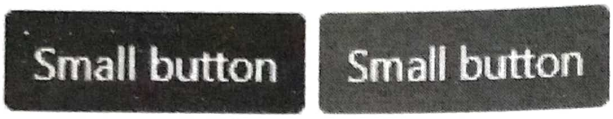


Similarly, to make buttons smaller add an extra class .btn-sm to the buttons.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Small Buttons</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <button type="button" class="btn btn-primary btn-sm">Small button</button>
  <button type="button" class="btn btn-secondary btn-sm">Small button</button>
```

```
</div>
</body>
</html>
```

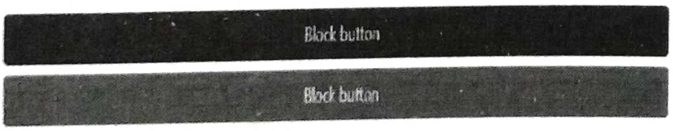
Output



You can also create full-width or block buttons (buttons that covers the full width of the parent elements) through using the Bootstrap's display and gap utility classes. These utilities offers much greater control over spacing, alignment, and responsive behaviours.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Block Buttons</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <div class="d-grid gap-2">
    <button type="button" class="btn btn-primary">Block
button</button>
    <button type="button" class="btn btn-
secondary">Block button</button>
  </div>
</div>
</body>
</html>
```

Output



☞ Bootstrap Disabled Buttons

Sometimes we need to disable a button for certain reasons, for example, a user in case is not eligible to perform this particular action, or we want to ensure that user should performed all other required actions before proceed to this particular action. Let's see how to do that.

☞ Creating Disabled Buttons Using Button and Input Element

Buttons created through <button> or <input> tag can be disabled by adding the disabled attribute to the respective element, as shown in the following example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Disabled Buttons</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <button type="button" class="btn btn-primary"
disabled>Primary button</button>
  <button type="button" class="btn btn-secondary"
disabled>Secondary button</button>
</div>
</body>
</html>
```

Output

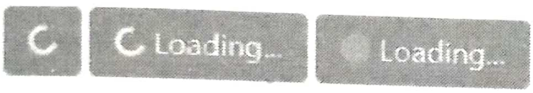
☞ Creating Spinner Buttons

With Bootstrap you can easily include spinner icon in a button to indicate the loading state in your application.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Spinner Buttons</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <button type="button" class="btn btn-primary" disabled>
    <span class="spinner-border spinner-border-
sm"></span>
  </button>
  <button type="button" class="btn btn-primary" disabled>
    <span class="spinner-border spinner-border-
sm"></span> Loading...
  </button>
  <button type="button" class="btn btn-primary" disabled>
    <span class="spinner-grow spinner-grow-
sm"></span> Loading...
  </button>
</div>
</body>
</html>
```

Output



1.32.2 Bootstrap Tables

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.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Simple Table</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <table class="table">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>
```

Output

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

Creating Accented Tables

Bootstrap even provides a handful of contextual classes such as. table-primary, .table-secondary, .table-success, .table-danger, .table-warning, .table-info, .table-light and .table-dark to color tables, table rows or individual cells.

For example, you can create a dark version of the table (i.e. table with light text on dark backgrounds) by adding the contextual class .table-dark to the .table base class.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Dark Table</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <table class="table table-dark">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>
```

Output

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

Similarly, you can use other contextual classes. For instance, the following example uses the class .table-success on the .table to create green colored variant of a table.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Accented Tables</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <table class="table table-primary">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
```

```
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>

<table class="table table-secondary">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>

<table class="table table-success">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>

<table class="table table-danger">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
```

```
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-warning">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-info">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-light">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
```

```
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>
```

Output

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

☞ Creating Tables with Striped Rows

You can also add zebra-striping to the table rows within the <tbody> by simply adding an additional class .table-striped to the .table base class, as shown below:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Zebra-Striped Tables</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <table class="table table-striped">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-striped table-dark">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-striped table-primary">
    <thead>
```

```
        <tr>
          <th>#</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Clark</td>
          <td>Kent</td>
          <td>clarkkent@mail.com</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Peter</td>
          <td>Parker</td>
          <td>peterparker@mail.com</td>
        </tr>
        <tr>
          <td>3</td>
          <td>John</td>
          <td>Carter</td>
          <td>johncarter@mail.com</td>
        </tr>
      </tbody>
    </table>

    <table class="table table-striped table-secondary">
      <thead>
        <tr>
          <th>#</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Clark</td>
          <td>Kent</td>
          <td>clarkkent@mail.com</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Peter</td>
          <td>Parker</td>
          <td>peterparker@mail.com</td>
        </tr>
        <tr>
          <td>3</td>
          <td>John</td>
          <td>Carter</td>
          <td>johncarter@mail.com</td>
        </tr>
      </tbody>
    </table>

    <table class="table table-striped table-success">
      <thead>
        <tr>
          <th>#</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Clark</td>
          <td>Kent</td>
          <td>clarkkent@mail.com</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Peter</td>
          <td>Parker</td>
          <td>peterparker@mail.com</td>
        </tr>
        <tr>
          <td>3</td>
          <td>John</td>
          <td>Carter</td>
          <td>johncarter@mail.com</td>
        </tr>
      </tbody>
    </table>

    <table class="table table-striped table-danger">
      <thead>
        <tr>
          <th>#</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
```

```
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>

<table class="table table-striped table-warning">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>

<table class="table table-striped table-info">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>

<table class="table table-striped table-light">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
```

```
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>
```

Output

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

☞ Creating Bordered Tables

You can add borders on all sides of the table and cells by adding the modifier class .table-bordered to the .table base class, as shown in the following example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Bordered Tables</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <table class="table table-bordered">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-bordered table-dark">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
```

```
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-bordered table-primary">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-bordered table-secondary">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-bordered table-success">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
```

```
        <tr>
          <td>3</td>
          <td>John</td>
          <td>Carter</td>
          <td>johncarter@mail.com</td>
        </tr>
      </tbody>
    </table>

    <table class="table table-bordered table-danger">
      <thead>
        <tr>
          <th>#</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Clark</td>
          <td>Kent</td>
          <td>clarkkent@mail.com</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Peter</td>
          <td>Parker</td>
          <td>peterparker@mail.com</td>
        </tr>
        <tr>
          <td>3</td>
          <td>John</td>
          <td>Carter</td>
          <td>johncarter@mail.com</td>
        </tr>
      </tbody>
    </table>

    <table class="table table-bordered table-warning">
      <thead>
        <tr>
          <th>#</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Clark</td>
          <td>Kent</td>
          <td>clarkkent@mail.com</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Peter</td>
          <td>Parker</td>
          <td>peterparker@mail.com</td>
        </tr>
        <tr>
          <td>3</td>
          <td>John</td>
          <td>Carter</td>
          <td>johncarter@mail.com</td>
        </tr>
      </tbody>
    </table>

    <table class="table table-bordered table-info">
      <thead>
        <tr>
          <th>#</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Clark</td>
          <td>Kent</td>
          <td>clarkkent@mail.com</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Peter</td>
          <td>Parker</td>
          <td>peterparker@mail.com</td>
        </tr>
        <tr>
          <td>3</td>
          <td>John</td>
          <td>Carter</td>
          <td>johncarter@mail.com</td>
```

```
      </tr>
    </tbody>
  </table>

  <table class="table table-bordered table-light">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>
```

Output

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

☞ Creating Borderless Tables

You can also create borderless tables using the class .table-borderless on the .table element.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Borderless Tables</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <table class="table table-borderless">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-borderless table-dark">
    <thead>
      <tr>
```

```
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-borderless table-primary">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-borderless table-secondary">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>

  <table class="table table-borderless table-success">
    <thead>
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
```

```
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>

<table class="table table-borderless table-danger">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>

<table class="table table-borderless table-warning">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
      <td>clarkkent@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Peter</td>
      <td>Parker</td>
      <td>peterparker@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>John</td>
      <td>Carter</td>
      <td>johncarter@mail.com</td>
    </tr>
  </tbody>
</table>
<table class="table table-borderless table-info">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Clark</td>
      <td>Kent</td>
```

```
            <td>clarkkent@mail.com</td>
          </tr>
          <tr>
            <td>2</td>
            <td>Peter</td>
            <td>Parker</td>
            <td>peterparker@mail.com</td>
          </tr>
          <tr>
            <td>3</td>
            <td>John</td>
            <td>Carter</td>
            <td>johncarter@mail.com</td>
          </tr>
        </tbody>
      </table>
      <table class="table table-borderless table-light">
        <thead>
          <tr>
            <th>#</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>1</td>
            <td>Clark</td>
            <td>Kent</td>
            <td>clarkkent@mail.com</td>
          </tr>
          <tr>
            <td>2</td>
            <td>Peter</td>
            <td>Parker</td>
            <td>peterparker@mail.com</td>
          </tr>
          <tr>
            <td>3</td>
            <td>John</td>
            <td>Carter</td>
            <td>johncarter@mail.com</td>
          </tr>
        </tbody>
      </table>
    </div>
    </body>
    </html>
```

Output

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

☞ Setting Table Head Colors

Similar to light and dark tables, you can use the modifier classes .table-light or .table-dark on the <thead> element to make it appear in light or dark gray.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Table Head Light</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <table class="table">
    <thead class="table-light">
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
```

```
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>
```

Output

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

The following example will create a table with dark gray background head.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Table Head Dark</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <table class="table">
    <thead class="table-dark">
      <tr>
        <th>#</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Clark</td>
        <td>Kent</td>
        <td>clarkkent@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Peter</td>
        <td>Parker</td>
        <td>peterparker@mail.com</td>
      </tr>
      <tr>
        <td>3</td>
        <td>John</td>
        <td>Carter</td>
        <td>johncarter@mail.com</td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>
```

Output

#	First Name	Last Name	Email
1	Clark	Kent	clarkkent@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Carter	johncarter@mail.com

☞ Creating Responsive Tables with Bootstrap

You can also create responsive tables to enable horizontal scrolling on small devices.

To make any table responsive just place it inside a element and apply the .table-responsive class on it. You can also specify when the table should have a scrollbar, based on the viewport width (i.e. breakpoints), using the classes .table-responsive{-sml-mdl-lgl-xl}.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Responsive Table</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
<style>
  /* Custom style to make this example easy to understand */
  table * {
    white-space: nowrap;
  }
</style>
</head>
<body>
<div class="m-4">
  <div class="table-responsive">
    <table class="table">
      <thead>
        <tr>
          <th>#</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
          <th>Biography</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Clark</td>
          <td>Kent</td>
          <td>clarkkent@mail.com</td>
          <td>Lorem ipsum dolor sit amet, consectetur
adipiscing elit.</td>
        </tr>
        <tr>
          <td>2</td>
          <td>John</td>
          <td>Carter</td>
          <td>johncarter@mail.com</td>
          <td>Vestibulum consectetur scelerisque
bibendum scelerisque purus.</td>
        </tr>
        <tr>
          <td>3</td>
          <td>Peter</td>
          <td>Parker</td>
          <td>peterparker@mail.com</td>
          <td>Integer pulvinar leo id risus interdum vel
metus dignissim.</td>
        </tr>
      </tbody>
    </table>
  </div>
  <p class="mt-4"><strong>Note:</strong> Change the
editor layout/orientation to see how responsive table
works.</p>
</div>
</body>
</html>
```

Output

#	First Name	Last Name	Email	Biography
1	Clark	Kent	clarkkent@mail.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	John	Carter	johncarter@mail.com	Vestibulum consectetur scelerisque bibendum scelerisque purus.
3	Peter	Parker	peterparker@mail.com	Integer pulvinar leo id risus interdum vel metus dignissim.

Note: Change the editor layout/orientation to see how responsive table works.

1.32.3 Bootstrap Lists

Creating Lists With Bootstrap

You can create three different types of HTML lists:

- Unordered lists — A list of items in which the order does not explicitly matter. The list items in unordered lists are marked with bullets, e.g. ▯, •, etc.
- Ordered lists — A list of items in which the order does explicitly matter. The list items in ordered lists are marked with numbers, e.g. 1, , etc.
- Definition list — A list of terms with their associated descriptions.

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.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Unstyled List</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <ul class="list-unstyled">
    <li>Home</li>
    <li>Products
      <ul>
        <li>Gadgets</li>
        <li>Accessories</li>
      </ul>
    </li>
    <li>About Us</li>
    <li>Contact</li>
  </ul>
</div>
</body>
</html>
```

Output

- Home
- Products
  - Gadgets
  - Accessories
- About Us
- Contact

☞ Placing Ordered and Unordered List Items Inline

If you want to create a horizontal menu using the ordered or unordered list you need to place all list items in a single line (i.e. side by side). You can do this simply by adding the class .list-inline to the <ul> or <ol>, and the class .list-inline-item to the child <li> elements.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Inline List</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <ul class="list-inline">
    <li class="list-inline-item">Home</li>
    <li class="list-inline-item">Products</li>
    <li class="list-inline-item">About Us</li>
    <li class="list-inline-item">Contact</li>
  </ul>
</div>
</body>
</html>
```

Output

Home Products About Us Contact

☞ Creating Horizontal Definition Lists

The terms and descriptions in a definition list can also be aligned horizontally side-by-side using the Bootstrap grid system's predefined classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Horizontal Definitoin List</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <dl class="row">
    <dt class="col-sm-3">User Agent</dt>
    <dd class="col-sm-9">An HTML user agent is any
device that interprets HTML documents.</dd>
```

```
    <dt class="col-sm-3 text-truncate">Client-side Scripting</dt>
    <dd class="col-sm-9">Client-side scripting generally refers to the category of computer programs on the web that are executed by the user's web browser.</dd>
    <dt class="col-sm-3">Document Tree</dt>
    <dd class="col-sm-9">The tree of elements encoded in the source document.</dd>
  </dl>
</div>
</body>
</html>
```

Output

User Agent	An HTML user agent is any device that interprets HTML documents.
Client-side Scripting	Client-side scripting generally refers to the category of computer programs on the web that are executed by the user's web browser.
Document Tree	The tree of elements encoded in the source document.

1.32.4 Bootstrap Navbar

Creating Navbar with Bootstrap

- You can use the Bootstrap navbar component to create responsive navigation header for your website or application. These responsive navbar will be collapsed on devices having small viewports like mobile phones but expand when user click the toggle button. However, it will be horizontal as normal on the medium and large devices such as laptop or desktop.
- You can also create different variations of the navbar such as navbars with dropdown menus and search boxes as well as fixed positioned navbar with much less effort. The following example will show you how to create a simple static navbar with navigation links.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Static Navbar</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link active">Home</a>
          <a href="#" class="nav-item nav-link">Profile</a>
          <a href="#" class="nav-item nav-link">Messages</a>
          <a href="#" class="nav-item nav-link disabled" tabindex="-1">Reports</a>
        </div>
        <div class="navbar-nav ms-auto">
          <a href="#" class="nav-item nav-link">Login</a>
        </div>
      </div>
    </div>
  </nav>
</div>
</body>
</html>
```

Output

Adding Logo Images to Navbars

You can also place your logo image inside the navbar, instead of plain text. However, you need to set the logo height manually to fit it properly inside the navbar, as shown here:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Navbar with Logo Image</title>
```

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <nav class="navbar navbar-expand-lg navbar-light bg-
light">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">
        <img src="/examples/images/logo.svg"
height="28" alt="CoolBrand">
      </a>
      <button type="button" class="navbar-toggler" data-
bs-toggle="collapse" data-bs-target="#navbarCollapse">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse"
id="navbarCollapse">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link
active">Home</a>
          <a href="#" class="nav-item nav-
link">Profile</a>
          <a href="#" class="nav-item nav-
link">Messages</a>
          <a href="#" class="nav-item nav-link
disabled" tabindex="-1">Reports</a>
        </div>
        <div class="navbar-nav ms-auto">
          <a href="#" class="nav-item nav-
link">Login</a>
        </div>
      </div>
    </div>
  </nav>
</div>
</body>
</html>
```

Output

☞ Adding Dropdowns to Navbar

You can also include dropdowns and search box within the navbars, as shown here:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Navbar with Dropdown</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <nav class="navbar navbar-expand-lg navbar-light bg-
light">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-
bs-toggle="collapse" data-bs-target="#navbarCollapse">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse"
id="navbarCollapse">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link
active">Home</a>
          <a href="#" class="nav-item nav-
link">Profile</a>
          <div class="nav-item dropdown">
            <a href="#" class="nav-link dropdown-
toggle" data-bs-toggle="dropdown">Messages</a>
            <div class="dropdown-menu">
              <a href="#" class="dropdown-
item">Inbox</a>
              <a href="#" class="dropdown-
item">Sent</a>
              <a href="#" class="dropdown-
item">Drafts</a>
            </div>
          </div>
          <a href="#" class="nav-item nav-link
disabled" tabindex="-1">Reports</a>
        </div>
        <div class="navbar-nav ms-auto">
```

```
            <a href="#" class="nav-item nav-
link">Login</a>
            </div>
        </div>
    </div>
  </nav>
</div>
</body>
</html>
```

Output

☞ Placing Search Form inside Navbar

Search form is a very common component of the navbars and you have seen it on various website quite often. You can place various form controls within a navbar using the class .d-flex on the <form> element, as demonstrated in the following example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Navbar with Dropdown</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <nav class="navbar navbar-expand-lg navbar-light bg-
light">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-
bs-toggle="collapse" data-bs-target="#navbarCollapse">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse justify-content-
between" id="navbarCollapse">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link
active">Home</a>
          <a href="#" class="nav-item nav-
link">Profile</a>
          <div class="nav-item dropdown">
            <a href="#" class="nav-link dropdown-
toggle" data-bs-toggle="dropdown">Messages</a>
            <div class="dropdown-menu">
              <a href="#" class="dropdown-
item">Inbox</a>
              <a href="#" class="dropdown-
item">Sent</a>
              <a href="#" class="dropdown-
item">Drafts</a>
            </div>
          </div>
        </div>
        <form class="d-flex">
          <div class="input-group">
            <input type="text" class="form-control"
placeholder="Search">
            <button type="button" class="btn btn-
secondary"><i class="bi-search"></i></button>
          </div>
        </form>
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-
link">Login</a>
        </div>
      </div>
    </div>
  </nav>
</div>
</body>
</html>
```

Output

☞ Changing the Color Scheme of Navbars

- You can also change the color scheme of the navbar by using the .navbar-light for the light background colors, or .navbar-dark for the dark background colors. Then, customize it with the background color utility classes, such as .bg-dark, .bg-primary, and so on.
- Alternatively, you can also apply the CSS background-color property on the .navbar element yourself to

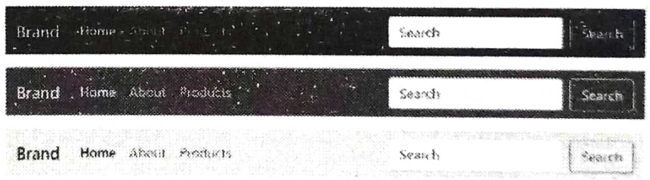
customize the navbar theme, as shown in the following example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Navbar Color Schemes</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
<style>
  /* Custom style */
 .navbar{
    margin-bottom: 1rem;
 }
</style>
</head>
<body>
<div class="m-4">
  <nav class="navbar navbar-expand-lg navbar-dark bg-
dark">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-
bs-toggle="collapse" data-bs-target="#navbarCollapse1">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse"
id="navbarCollapse1">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link
active">Home</a>
          <a href="#" class="nav-item nav-
link">About</a>
          <a href="#" class="nav-item nav-
link">Products</a>
        </div>
        <form class="d-flex ms-auto">
          <input type="text" class="form-control me-sm-
2" placeholder="Search">
          <button type="submit" class="btn btn-outline-
light">Search</button>
        </form>
      </div>
    </div>
  </nav>

  <nav class="navbar navbar-expand-lg navbar-dark bg-
primary">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-
bs-toggle="collapse" data-bs-target="#navbarCollapse2">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse"
id="navbarCollapse2">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link
active">Home</a>
          <a href="#" class="nav-item nav-
link">About</a>
          <a href="#" class="nav-item nav-
link">Products</a>
        </div>
        <form class="d-flex ms-auto">
          <input type="text" class="form-control me-sm-
2" placeholder="Search">
          <button type="submit" class="btn btn-outline-
light">Search</button>
        </form>
      </div>
    </div>
  </nav>

  <nav class="navbar navbar-expand-lg navbar-dark bg-
secondary">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-
bs-toggle="collapse" data-bs-target="#navbarCollapse3">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse"
id="navbarCollapse3">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link
active">Home</a>
          <a href="#" class="nav-item nav-
link">About</a>
          <a href="#" class="nav-item nav-
link">Products</a>
        </div>
```

```
            <form class="d-flex ms-auto">
                <input type="text" class="form-control me-sm-2" placeholder="Search">
                <button type="submit" class="btn btn-outline-light">Search</button>
            </form>
        </div>
    </div>
  </nav>

  <nav class="navbar navbar-expand-lg navbar-dark bg-success">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse4">
          <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse4">
          <div class="navbar-nav">
              <a href="#" class="nav-item nav-link active">Home</a>
              <a href="#" class="nav-item nav-link">About</a>
              <a href="#" class="nav-item nav-link">Products</a>
          </div>
          <form class="d-flex ms-auto">
              <input type="text" class="form-control me-sm-2" placeholder="Search">
              <button type="submit" class="btn btn-outline-light">Search</button>
          </form>
      </div>
    </div>
  </nav>

  <nav class="navbar navbar-expand-lg navbar-dark bg-info">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse5">
          <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse5">
          <div class="navbar-nav">
              <a href="#" class="nav-item nav-link active">Home</a>
              <a href="#" class="nav-item nav-link">About</a>
              <a href="#" class="nav-item nav-link">Products</a>
          </div>
          <form class="d-flex ms-auto">
              <input type="text" class="form-control me-sm-2" placeholder="Search">
              <button type="submit" class="btn btn-outline-light">Search</button>
          </form>
      </div>
    </div>
  </nav>

  <nav class="navbar navbar-expand-lg navbar-dark bg-danger">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse6">
          <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse6">
          <div class="navbar-nav">
              <a href="#" class="nav-item nav-link active">Home</a>
              <a href="#" class="nav-item nav-link">About</a>
              <a href="#" class="nav-item nav-link">Products</a>
          </div>
          <form class="d-flex ms-auto">
              <input type="text" class="form-control me-sm-2" placeholder="Search">
              <button type="submit" class="btn btn-outline-light">Search</button>
          </form>
      </div>
    </div>
  </nav>

  <nav class="navbar navbar-expand-lg navbar-dark bg-warning">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse7">
          <span class="navbar-toggler-icon"></span>
      </button>
```

```
      <div class="collapse navbar-collapse"
id="navbarCollapse7">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link
active">Home</a>
          <a href="#" class="nav-item nav-
link">About</a>
          <a href="#" class="nav-item nav-
link">Products</a>
        </div>
        <form class="d-flex ms-auto">
          <input type="text" class="form-control me-sm-
2" placeholder="Search">
          <button type="submit" class="btn btn-outline-
light">Search</button>
        </form>
      </div>
    </div>
  </nav>

  <nav class="navbar navbar-expand-lg navbar-light bg-
light">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-
bs-toggle="collapse" data-bs-target="#navbarCollapse8">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse"
id="navbarCollapse8">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link
active">Home</a>
          <a href="#" class="nav-item nav-
link">About</a>
          <a href="#" class="nav-item nav-
link">Products</a>
        </div>
        <form class="d-flex ms-auto">
          <input type="text" class="form-control me-sm-
2" placeholder="Search">
          <button type="submit" class="btn btn-outline-
secondary">Search</button>
        </form>
      </div>
    </div>
  </nav>

  <nav class="navbar navbar-expand-lg navbar-light"
style="background-color: #e3f2fd;">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">Brand</a>
      <button type="button" class="navbar-toggler" data-
bs-toggle="collapse" data-bs-target="#navbarCollapse9">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse"
id="navbarCollapse9">
        <div class="navbar-nav">
          <a href="#" class="nav-item nav-link
active">Home</a>
          <a href="#" class="nav-item nav-
link">About</a>
          <a href="#" class="nav-item nav-
link">Products</a>
        </div>
        <form class="d-flex ms-auto">
          <input type="text" class="form-control me-sm-
2" placeholder="Search">
          <button type="submit" class="btn btn-outline-
primary">Search</button>
        </form>
      </div>
    </div>
  </nav>
</div>
</body>
</html>
```

Output



☞ Bootstrap Fixed Navbars

Bootstrap also provides mechanism to create navbar that is fixed to the top, fixed to the bottom, or stickied to the top (i.e. scrolls with the page until it reaches the top, then stays there).

1. Navbar Fixed to the Top

Apply the position utility class .fixed-top to the .navbar element to fix the navbar at the top of the viewport, so that it won't scroll with the page

```
<nav class="navbar fixed-top navbar-expand-lg navbar-dark
bg-dark"> <!-- Navbar content --> </nav>
```

2. Navbar Stickied to the Top

You can also create sticky top navbar that scrolls with the page until it reaches the top, then stays there, by simply using the .sticky-top class on the .navbar element, like this:

```
<nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark"> <!-- Navbar content --> </nav>
```

3. Navbar Fixed to the Bottom

Similarly, add the class .fixed-bottom to the .navbar element to fix the navbar at the bottom of the viewport. It also won't scroll with the page.

```
<nav class="navbar fixed-bottom navbar-expand-lg navbar-dark bg-dark">
  <!-- Navbar content -->
</nav>
```

1.32.5 Bootstrap Forms

Creating Forms with Bootstrap

HTML forms are an integral part of the web pages and applications, but creating the form layouts or styling the form controls manually one by one using 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:

1. Vertical Form (default form layout)
2. Horizontal Form
3. Inline Form

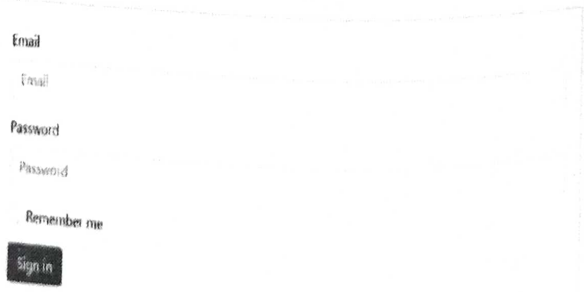
1. Creating Vertical Form Layout

To create vertical form layouts simply use the predefined margin utility classes for grouping the labels, form controls, optional form text, and form validation messages.

Here's an example in which form controls are vertically stacked with labels on the top.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Vertical Form Layout</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <form action="/examples/actions/confirmation.php" method="post">
    <div class="mb-3">
      <label class="form-label" for="inputEmail">Email</label>
      <input type="email" class="form-control" id="inputEmail" placeholder="Email" required>
    </div>
    <div class="mb-3">
      <label class="form-label" for="inputPassword">Password</label>
      <input type="password" class="form-control" id="inputPassword" placeholder="Password" required>
    </div>
    <div class="mb-3">
      <div class="form-check">
        <input class="form-check-input" type="checkbox" id="checkRemember">
        <label class="form-check-label" for="checkRemember">Remember me</label>
      </div>
    </div>
    <button type="submit" class="btn btn-primary">Sign in</button>
  </form>
</div>
</body>
</html>
```

Output

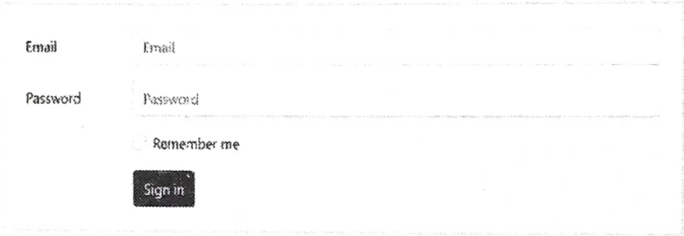


2. Creating Horizontal Form Layout

- You can also create horizontal form layouts where labels and form controls are aligned side-by-side using the Bootstrap grid classes. To create a horizontal form layout add the class .row on form groups and use the .col-*-* grid classes to specify the width of your labels and controls.
- Also, be sure to apply the class .col-form-label on the <label> elements, so that they're vertically centered with their associated form controls. Let's check out an example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Horizontal Form Layout</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <form action="/examples/actions/confirmation.php"
method="post">
    <div class="row mb-3">
      <label for="inputEmail" class="col-sm-2 col-form-
label">Email</label>
      <div class="col-sm-10">
        <input type="email" class="form-control"
id="inputEmail" placeholder="Email" required>
      </div>
    </div>
    <div class="row mb-3">
      <label for="inputPassword" class="col-sm-2 col-
form-label">Password</label>
      <div class="col-sm-10">
        <input type="password" class="form-control"
id="inputPassword" placeholder="Password" required>
      </div>
    </div>
    <div class="row mb-3">
      <div class="col-sm-10 offset-sm-2">
        <div class="form-check">
          <input class="form-check-input"
type="checkbox" id="checkRemember">
          <label class="form-check-label"
for="checkRemember">Remember me</label>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-sm-10 offset-sm-2">
        <button type="submit" class="btn btn-
primary">Sign in</button>
      </div>
    </div>
  </form>
</div>
</body>
</html>
```

Output



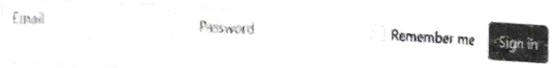
3. Creating Inline Form Layout

Sometimes you may want to display a series of form controls, and buttons in a single horizontal row to compact the layout. You can do this easily by using the Bootstrap's grid classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Inline Form Layout</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <form action="/examples/actions/confirmation.php"
method="post">
```

```
    <div class="row align-items-center g-3">
      <div class="col-auto">
        <label class="visually-hidden"
for="inputEmail">Email</label>
        <input type="email" class="form-control"
id="inputEmail" placeholder="Email" required>
      </div>
      <div class="col-auto">
        <label class="visually-hidden"
for="inputPassword">Password</label>
        <input type="password" class="form-control"
id="inputPassword" placeholder="Password" required>
      </div>
      <div class="col-auto">
        <div class="form-check">
          <input class="form-check-input"
type="checkbox" id="checkRemember">
          <label class="form-check-label"
for="checkRemember">Remember me</label>
        </div>
      </div>
      <div class="col-auto">
        <button type="submit" class="btn btn-
primary">Sign in</button>
      </div>
    </div>
  </form>
</div>
</body>
</html>
```

Output

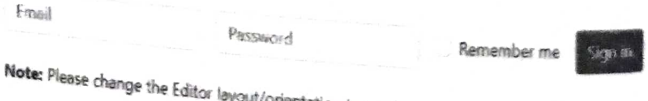


Creating Responsive Form Layout

- You can also make your forms responsive through using the grid classes with specific breakpoints.
- The following example will create a form which laid out inline on medium devices and up (i.e., viewport width ≥768px), but will become vertically stacked on smaller viewports.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Responsive Form Layout</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <form action="/examples/actions/confirmation.php"
method="post">
    <div class="row align-items-center g-3">
      <div class="col-md-auto col-12">
        <label class="form-label d-md-none"
for="inputEmail">Email</label>
        <input type="email" class="form-control"
id="inputEmail" placeholder="Email" required>
      </div>
      <div class="col-md-auto col-12">
        <label class="form-label d-md-none"
for="inputPassword">Password</label>
        <input type="password" class="form-control"
id="inputPassword" placeholder="Password" required>
      </div>
      <div class="col-md-auto col-12">
        <div class="form-check">
          <input class="form-check-input"
type="checkbox" id="checkRemember">
          <label class="form-check-label"
for="checkRemember">Remember me</label>
        </div>
      </div>
      <div class="col-md-auto col-12">
        <button type="submit" class="btn btn-
primary">Sign in</button>
      </div>
    </div>
  </form>
  <p class="mt-4"><strong>Note:</strong> Please
change the Editor layout/orientation to see how responsive
layout actually works.</p>
</div>
</body>
</html>
```

Output



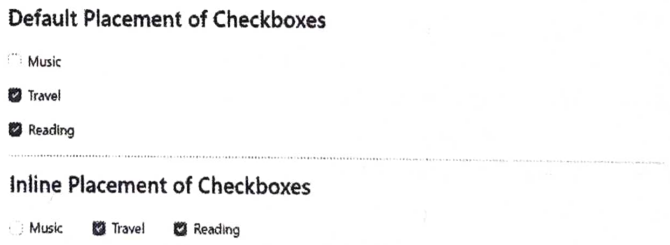
Note: Please change the Editor layout/orientation to see how responsive layout actually works.

Placing Checkboxes and Radio Buttons Inline

- By default, any number of custom checkboxes and radio buttons that are immediate sibling will be vertically stacked and appropriately spaced with .form-check class.
- But, you can also place these custom checkboxes and radio buttons inline (i.e., in the same line) by simply adding the class .form-check-inline to .form-check element, like this:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Placing Checkboxes Inline</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <h3 class="mb-4">Default Placement of
Checkboxes</h3>
  <div class="row">
    <div class="col-12">
      <div class="form-check mb-3">
        <input type="checkbox" class="form-check-input"
name="hobbies" id="checkMusic">
        <label class="form-check-label"
for="checkMusic">Music</label>
      </div>
      <div class="form-check mb-3">
        <input type="checkbox" class="form-check-input"
name="hobbies" id="checkTravel" checked>
        <label class="form-check-label"
for="checkTravel">Travel</label>
      </div>
      <div class="form-check">
        <input type="checkbox" class="form-check-input"
name="hobbies" id="checkReading" checked>
        <label class="form-check-label"
for="checkReading">Reading</label>
      </div>
    </div>
  </div>
  <hr>
  <h3 class="mb-4">Inline Placement of
Checkboxes</h3>
  <div class="row">
    <div class="col-12">
      <div class="form-check form-check-inline">
        <input type="checkbox" class="form-check-input"
name="hobbies" id="checkMusic">
        <label class="form-check-label"
for="checkMusic">Music</label>
      </div>
      <div class="form-check form-check-inline ms-3">
        <input type="checkbox" class="form-check-input"
name="hobbies" id="checkTravel" checked>
        <label class="form-check-label"
for="checkTravel">Travel</label>
      </div>
      <div class="form-check form-check-inline ms-3">
        <input type="checkbox" class="form-check-input"
name="hobbies" id="checkReading" checked>
        <label class="form-check-label"
for="checkReading">Reading</label>
      </div>
    </div>
  </div>
</div>
</body>
</html>
```

Output



Similarly, you can place the radio buttons inline, as shown in the following example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<title>Bootstrap Placing Radio Buttons Inline</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js"></script>
</head>
<body>
<div class="m-4">
  <h3 class="mb-4">Default Placement of Radio
Buttons</h3>
  <div class="row">
    <div class="col-12">
      <div class="form-check mb-3">
        <input type="radio" class="form-check-input"
name="gender" id="radioMale" checked>
        <label class="form-check-label"
for="radioMale">Male</label>
      </div>
      <div class="form-check">
        <input type="radio" class="form-check-input"
name="gender" id="radioFemale">
        <label class="form-check-label"
for="radioFemale">Female</label>
      </div>
    </div>
  </div>
  <hr>
  <h3 class="mb-4">Inline Placement of Radio
Buttons</h3>
  <div class="row">
    <div class="col-12">
      <div class="form-check form-check-inline">
        <input type="radio" class="form-check-input"
name="gender" id="radioMale" checked>
        <label class="form-check-label"
for="radioMale">Male</label>
      </div>
      <div class="form-check form-check-inline ms-3">
        <input type="radio" class="form-check-input"
name="gender" id="radioFemale">
        <label class="form-check-label"
for="radioFemale">Female</label>
      </div>
    </div>
  </div>
</div>
</body>
</html>
```

Output

Default Placement of Radio Buttons

Male

Female

Inline Placement of Radio Buttons

Male Female

Adding Help Text to 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 a form control using the class .form-text. The block help text is typically displayed at the bottom of the control. Here's an example:

```
<label class="form-label"
for="inputPassword">Password</label>
<input type="password" class="form-control"
id="inputPassword">
<div class="form-text">
  Must be 8-20 characters long, contain letters, numbers and
special characters, but must not contain spaces.
</div>
```

Output

Password

Must be 8-20 characters long, contain letters, numbers and special characters, but must not contain spaces.

- Similarly, you can also place inline help text using the <small> element. No need to use .form-text in this case. The following example shows how to implement this:

```
<div class="row align-items-center g-3">
  <div class="col-auto">
    <label class="col-form-label"
for="inputPassword">Password</label>
  </div>
  <div class="col-auto">
    <input type="password" class="form-control"
id="inputPassword">
  </div>
  <div class="col-auto">
    <span class="form-text">Must be 8-20 characters
long.</span>
  </div>
</div>
```

Output

Password Must be 8-20 characters long.

1.33 CROSS-BROWSER COMPATIBILITY

- The Bootstrap team ensures the compatibility of the framework with all modern browsers versions and platforms. Also, the team claims that they don't support proxy browsers and older browsers, but that doesn't affect its display or functionality.
- Bootstrap supports the latest, stable releases of all major browsers and platforms. On Windows, we support Internet Explorer 10-11 / Microsoft Edge. Alternative browsers which use the latest version of WebKit, Blink, or Gecko, whether directly or via the platform's web view API, are not explicitly supported.

Supported browsers

- Bootstrap supports the latest, stable releases of all major browsers and platforms. On Windows, we support Internet Explorer 10-11 / Microsoft Edge.
- Alternative browsers which use the latest version of WebKit, Blink, or Gecko, whether directly or via the platform's web view API, are not explicitly supported. However, Bootstrap should (in most cases) display and function correctly in these browsers as well. More specific support information is provided below.

Mobile devices

Generally speaking, Bootstrap supports the latest versions of each major platform's default browsers. Note that proxy browsers (such as Opera Mini, Opera Mobile's Turbo mode, UC Browser Mini, Amazon Silk) are not supported.

	Chrome	Firefox	Safari	Android Browser & WebView	Microsoft Edge
Android	Supported	Supported	N/A	Android v5.0+ supported	Supported
iOS	Supported	Supported	Supported	N/A	Supported
Windows 10 Mobile	N/A	N/A	N/A	N/A	Supported

Desktop browsers

Similarly, the latest versions of most desktop browsers are supported.

	Chrome	Firefox	Internet Explorer	Microsoft Edge	Opera	Safari
Mac	Supported	Supported	N/A	N/A	Supported	Supported
Windows	Supported	Supported	Supported, IE10+	Supported	Supported	Not supported

1.34 W3C

GQ. What is W3C?

- The World Wide Web Consortium (W3C) is an international organization committed to improving the web.
- It is made up of several hundred member organizations from a variety of related IT industries.
- W3C sets standards for the World Wide Web (WWW) to facilitate interoperability and cooperation among all web stakeholders. It was established in 1994 by the creator of the WWW, Tim Berners-Lee.

History

- World Wide Web Consortium was founded by Tim Berners-Lee in October 1994 at the Massachusetts Institute of Technology, with the support of DARPA (Defense Advanced Research Projects Agency) and CERN. The main vision of W3C was to standardize the technologies as well as the protocols that are used to build the web.
- It also tried to encourage the organizations to adopt the new standards defined by the World Wide Web Consortium. In the next few years, the W3C thus published various recommendations such as the format of PNG images, CSS etc.
- European branch of W3C was first considered to be hosted by CERN but later it did not happen as CERN focused on physics rather than Information Technology. So the French Institute for Research in Computer Science and Automation hosted the European branch of W3C in April 1995. Further various other hosts were decided such as Beihang University being the Chinese host, Keio University being the Asian host, etc.
- The mission of the W3C is to lead the Web to its full potential by developing relevant protocols and

guidelines. This is achieved primarily by creating and publishing Web standards. By adopting the Web standards created by the W3C, hardware manufacturers and software developers can ensure their equipment and programs work with the latest Web technologies.

- For example, most Web browsers incorporate several W3C standards, which allows them to interpret the latest versions of HTML and CSS code. When browsers conform to the W3C standards, it also helps Web pages appear consistent across different browsers.

1.34.1 Characteristics of W3C

- It is responsible for creating and publishing web standards.
- It also ensures the growth and development of web.
- It also develops the standards for web scripting, web applications and other dynamic contents.
- It is an organization which helps in the promotion of interoperability by the promotion and designing of open protocols.
- W3C uses the principles of modularity, simplicity and extensibility while designing web protocols.

1.34.2 Advantages of W3C

1. W3C enables the easier maintenance of the W3C validated websites.
2. It provides a consistent and defined look for all the W3C validated websites.
3. It standardizes the validated websites so that they are accessible to different devices.
4. It enables faster browser interaction.

1.34.3 Disadvantages of W3C

1. W3C validation is a timely process and thus the time for full validation depends on the website code.
2. W3C validation exercises have costs associated with them.
3. Sometimes translation issues arise in W3C validation of websites.

1.34.4 Importance of W3C Standards in Web Technologies

GQ. How W3C standards supports web technologies?

- The World Wide Web Consortium (W3C) is a standards organization for the web. They are best known for their standards for HTML and CSS, and for their accessibility standards that are accepted in many national regulations around the world. It is important because it allows developers to write one set of code and have it work everywhere.
- For accessibility, it means that developers can design accessible sites that can meet the requirements of many countries. Otherwise, they would have to design different sites for each country to meet their regulations.
- W3C has a consensus process that encourages broad public input. Their standards have no charge, and have no charge to implement them. They try hard to act in the best interests of the web. They don't always succeed, but I know from personal experience that they try very hard. When they make a mistake, they try to adjust the process so it won't happen again.
- As a web developer, enforcing these standards ensures that web content is accessible in all browsers to reach all audiences, as well as optimizes the user experience. For example, using W3C-compliant CSS and XML allows every website to function similarly, but also improves SEO.
- Using standards will most likely increase your site's search engine ranking, and you can even easily convert your web pages into other formats or "mash up" the content into other web and desktop applications. Most of all, your site will be easier to maintain, update, and redesign.
- The W3C also provides access to free tools that will assist you in making your websites comply with web standards.
- W3C standards for XML and CSS ensure that every website will function the same on any browser. Using W3C-compliant XML and CSS coding also improves search engine optimization by allowing search engine robots to "crawl" through websites quicker and more efficiently. W3C standards also improve issues of accessibility, privacy, security, and internationalization, while helping to balance speed, fairness, public accountability, and quality on the Internet. W3C-compliant websites are cross-platform compatible. This

means that, in addition to working on any browser, they can also be accessed via mobile phones, screen readers and text browsers, interactive televisions, and other devices. Websites that use XML/CSS markup are cheaper to run because they use less bandwidth. And W3C's royalty-free patent licensing commitments ensure that everyone has access to the technology they need to build their websites with the desired functionality.

- W3C standard technologies are downloadable at no cost, maintained in a predictable fashion, and strengthened through rigorous interoperability testing. Conversely, websites that don't use W3C protocols will be slow to load, which can frustrate visitors. They won't work well with all browsers, which could eliminate many potential viewers. And they won't work as efficiently with search engines, which could lower search rankings and make it harder for people to find the site. W3C standards are the primary reason the web works so well today and will work even better in the future. They are good for web designers and for web users, and should be an integral part of every web design project.

...Chapter Ends

❑❑❑