Introduction to CSS

CSS Text Properties

CSS provides several properties that allows you to define various text styles such as color, alignment, spacing, decoration, transformation, etc. very easily and effectively.

The commonly used text properties are: text-align, text-decoration, text-transform, text-indent, line-height, letter-spacing, word-spacing, and more. These properties give you precise control over the visual appearance of the characters, words, spaces, and so on.

Property	Some Possible Values
color:	#ffccee;
text-align:	start | end| left | right | center | justify
text-decoration:	underline | overline | line-through | none | blink;
text-transform:	uppercase | lowercase | capitalize | none ;
text-indent:	length | %
letter-spacing:	normal | length | %
word-spacing:	normal | length | %
text-outline:	none | color | length
text-shadow:	none | color | length

CSS Text Example

```
<html>
   <head>
   </head>

   <body>
      <p style = "color:red;">
         This text will be written in red.
      </p>
   </body>
</html>
```

It will produce the following result –

This text will be written in red.

```
<html>
   <head>
   </head>

   <body>
      <p style = "letter-spacing:5px;">
         This text is having space between letters.
      </p>
   </body>
</html>
```

It will produce the following result –

This text is having space between letters.

CSS Text Example

```
<html>
   <head>
   </head>
   <body>
      <p style = "text-decoration:underline;">
         This will be underlined
      </p>
      <p style = "text-decoration:line-through;">
         This will be striked through.
      </p>
      <p style = "text-decoration:overline;">
         This will have a over line.
      </p>
      <p style = "text-decoration:blink;">
         This text will have blinking effect
      </p>
   </body>
</html>
```

This will produce following result –

This will be underlined

~~This will be striked through.~~

This will have a over line.

This text will have blinking effect

CSS Text Example

```
<html>
   <head>
   </head>

   <body>
      <p style = "text-shadow:4px 4px 8px blue;">
         If your browser supports the CSS text-shadow
property,
         this text will have a  blue shadow.
      </p>
   </body>
</html>
```

It will produce the following result −

If your browser supports the CSS text-shadow property, this text will have a blue shadow.

CSS list Properties

Lists are very helpful in conveying a set of either numbered or bullet points. This chapter teaches you how to control list type, position, style, etc., using CSS.

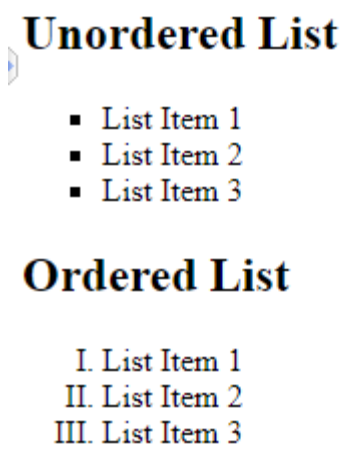
We have the following five CSS properties, which can be used to control lists −

- list-style-type: none | disc | circle | square …. (allows you to control the shape or appearance of the marker)
- list-style-position: inside | outside (specifies whether a long point that wraps to a second line should align with the first line or start underneath the start of the marker)
- list-style-image: none | url (specifies an image for the marker rather than a bullet point or number)
- marker-offset: auto | length (specifies the distance between a marker and the text in the list)

CSS list example

```
<html>
<head>
<title>Setting Numbering or Bullet Point Style of
Lists</title>
<style>
    ul { list-style-type: square;    }
    ol {list-style-type: upper-roman;    }
</style>
</head>
<body>
    <h2>Unordered List</h2>
    <ul>
        <li>List Item 1</li>        <li>List Item 2</li>
        <li>List Item 3</li>
     </ul>
```

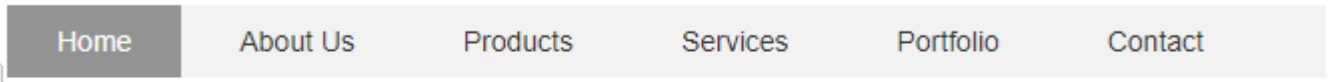
```
<h2>Ordered List</h2>
    <ol>
        <li>List Item 1</li>
        <li>List Item 2</li>
        <li>List Item 3</li>
    </ol>
</body>
</html>
```



CSS Navigation Menus example

```
<html>
<head>
<title>Building Navigation Bar with HTML List
and CSS</title>
<style>
    body{font-size: 14px;
    font-family: Arial,sans-serif;   }
    ul { padding: 0; list-style: none;
    background: #f2f2f2;  }
    ul li {display: inline-block;  }
    ul li a { display: block;
        padding: 10px 25px;
        color: #333;
        text-decoration: none;  }
    ul li a:hover { color: #fff;
        background: #939393;
    }
</style>
</head>
```

```
<body>
    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About Us</a></li>
            <li><a href="#">Products</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Portfolio</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </nav>
            <p><strong>Note:</strong> Place mouse
pointer over the menu link to see the hover effect.</p>
</body>
```



Note: Place mouse pointer over the menu link to see the hover effect.

CSS Pseudo-classes

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

Syntax

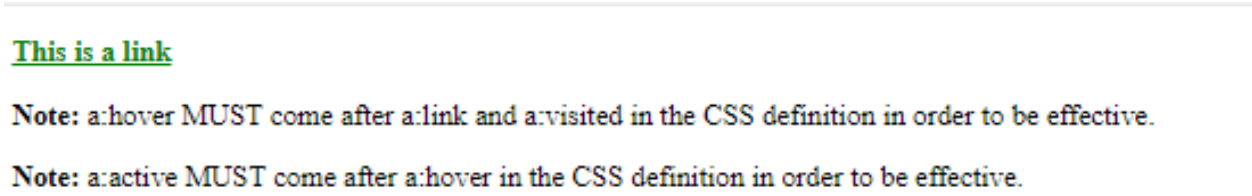
The syntax of pseudo-classes:

```
selector:pseudo-class {
  property: value;
}
```

CSS Pseudo-classes Example

```
<html>
<head>
<style>
/* unvisited link */
a:link { color: red;}

/* visited link */
a:visited {  color: green; }

/* mouse over link */
a:hover {  color: hotpink; }

/* selected link */
a:active {  color: blue; }
</style>
</head>
```

```
<body>

<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>

</body>
</html>
```



Hover on

```
<html>
<head>
<style>
div {
  background-color: green;
  color: white;
  padding: 25px;
  text-align: center;
}

div:hover {  background-color: blue;}
</style>
</head>
```

```
<body>

<p>Mouse over the div element below to change its background color:</p>

<div>Mouse Over Me</div>

</body>
</html>
```

CSS position Properties

CSS helps you to position your HTML element. You can put any HTML element at whatever location you like. You can specify whether you want the element positioned relative to its natural position in the page or absolute based on its parent element.

Property:Some Possible Values

position: relative | absolute | fixed | static
bottom: auto | % | length
left: auto | % | length
right: auto | % | length
top: auto | % | length
z-index: auto | number (The z-axis position of each layer is expressed as an integer representing the stacking order for rendering. An element with a larger z-index overlaps an element with a lower one.)

CSS static position

A static positioned element is always positioned according to the normal flow of the page. HTML elements are positioned static by default. Static positioned elements are not affected by the top, bottom, left, right, and z-index properties.

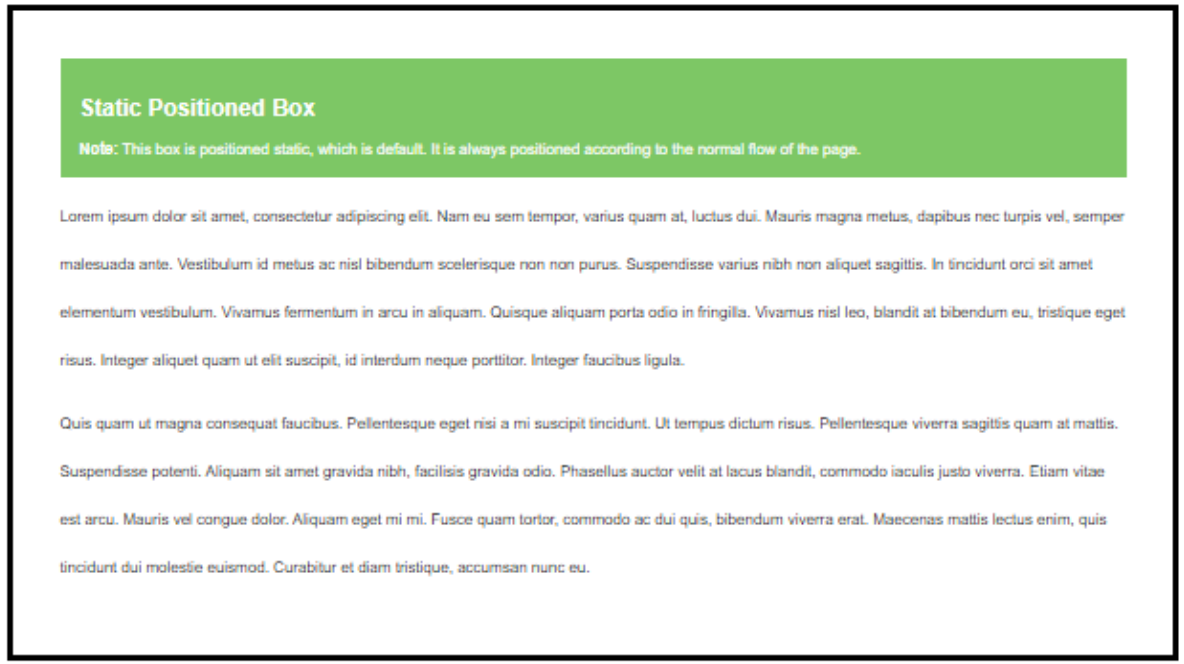
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of CSS Static Positioning</title>
<style>
    .box{
        color: #fff;
        background: #7dc765;
        padding: 20px;
    }
```

```
.container{
        padding: 50px;
        margin: 50px;
        position: relative;
        border: 5px solid black;
        font-family: Arial, sans-serif;
    }
    .container p{
        line-height: 50px;
    }
</style>
</head>
```

```
<body>
    <div class="container">
        <div class="box">
            <h2>Static Positioned Box</h2>

<div><strong>Note:</strong> This box is positioned static, which is default. It is always positioned according to the normal flow of the page.</div>
        </div>
```

CSS static position

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus,.</p>
        <p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit euismod. Curabitur et diam tristique, accumsan nunc eu.</p>
    </div>
</body>
</html>
```

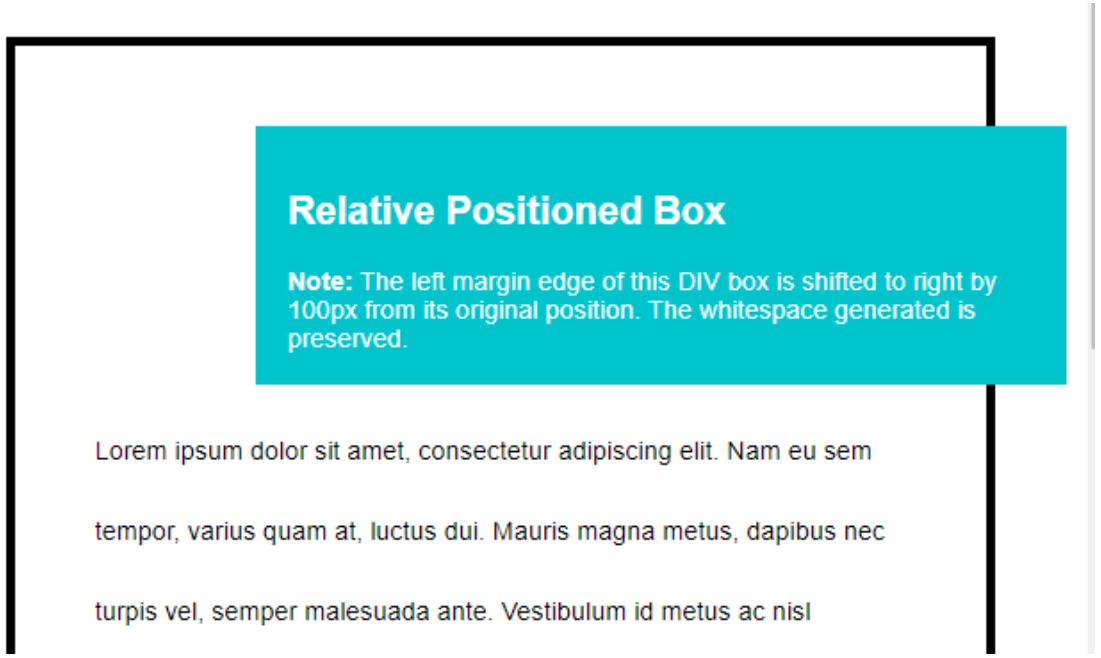


CSS relative position

A relative positioned element is positioned relative to its normal position.

In the relative positioning scheme the element's box position is calculated according to the normal flow. Then the box is shifted from this normal position according to the properties — top or bottom and/or left or right.

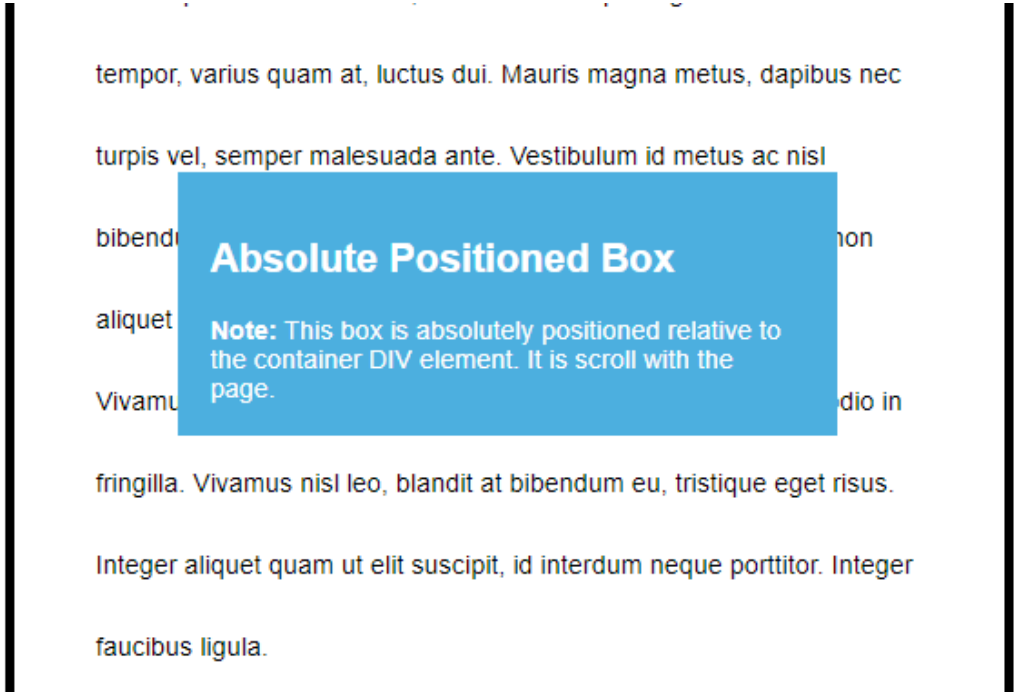
```
<style>
    .box{
        position: relative; left: 100px; color: #fff;
        background: #00c4cc; padding: 20px; }
    .container{
        padding: 50px; margin: 50px;
        border: 5px solid black; font-family: Arial,
sans-serif;
    }
    .container p{line-height: 50px; }
</style>
```



CSS absolute position

An absolutely positioned element is positioned relative to the first parent element that has a position other than static. If no such element is found, it will be positioned on a page relative to the 'top-left' corner of the browser window. The box's offsets further can be specified using one or more of the properties top, right, bottom, and left.

```
<style>
    .box{
        position: absolute; top: 200px; left: 100px;
        color: #fff;   width: 60%; background: #4cafdf;
        padding: 20px;
    }
    .container{
        padding: 50px; margin: 50px;   position: relative;
        border: 5px solid black;  font-family: Arial, sans-serif;
    }
    .container p{ line-height: 50px;  }
</style>
```

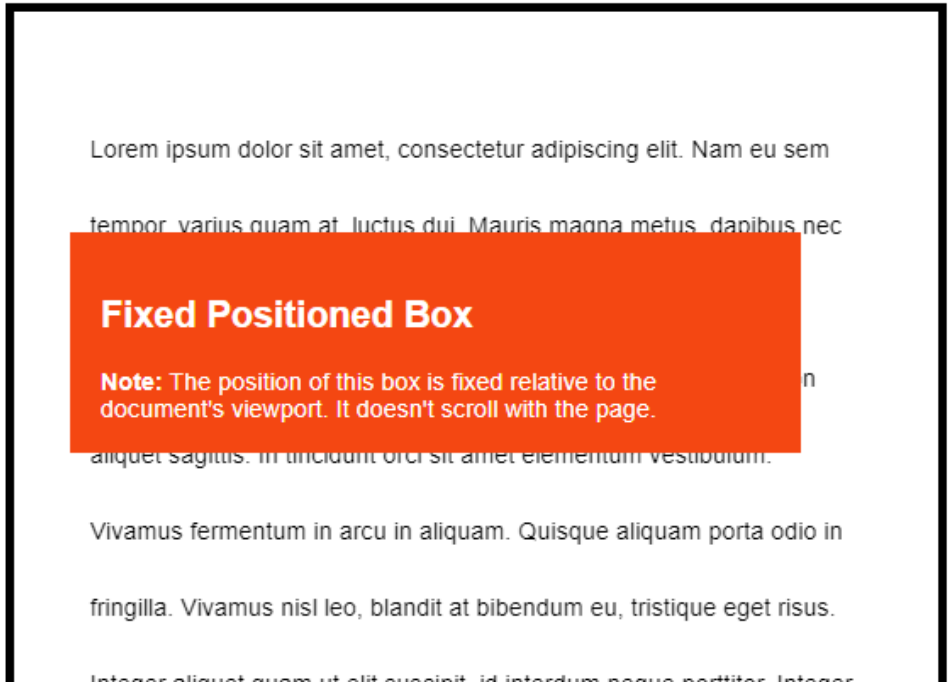


CSS fixed position

Fixed positioning is a subcategory of absolute positioning.

The only difference is, a fixed positioned element is fixed with respect to the browser's viewport and does not move when scrolled.

```
<style>
    .box{
        position: fixed;  top: 200px;  left: 100px;
        color: #fff; width: 60%;
        background: #f44712;  padding: 20px;  }
    .container{
        padding: 50px;  margin: 50px;
        position: relative;  border: 5px solid black;
        font-family: Arial, sans-serif;  }
    .container p{   line-height: 50px;   }
</style>
```



CSS overflow

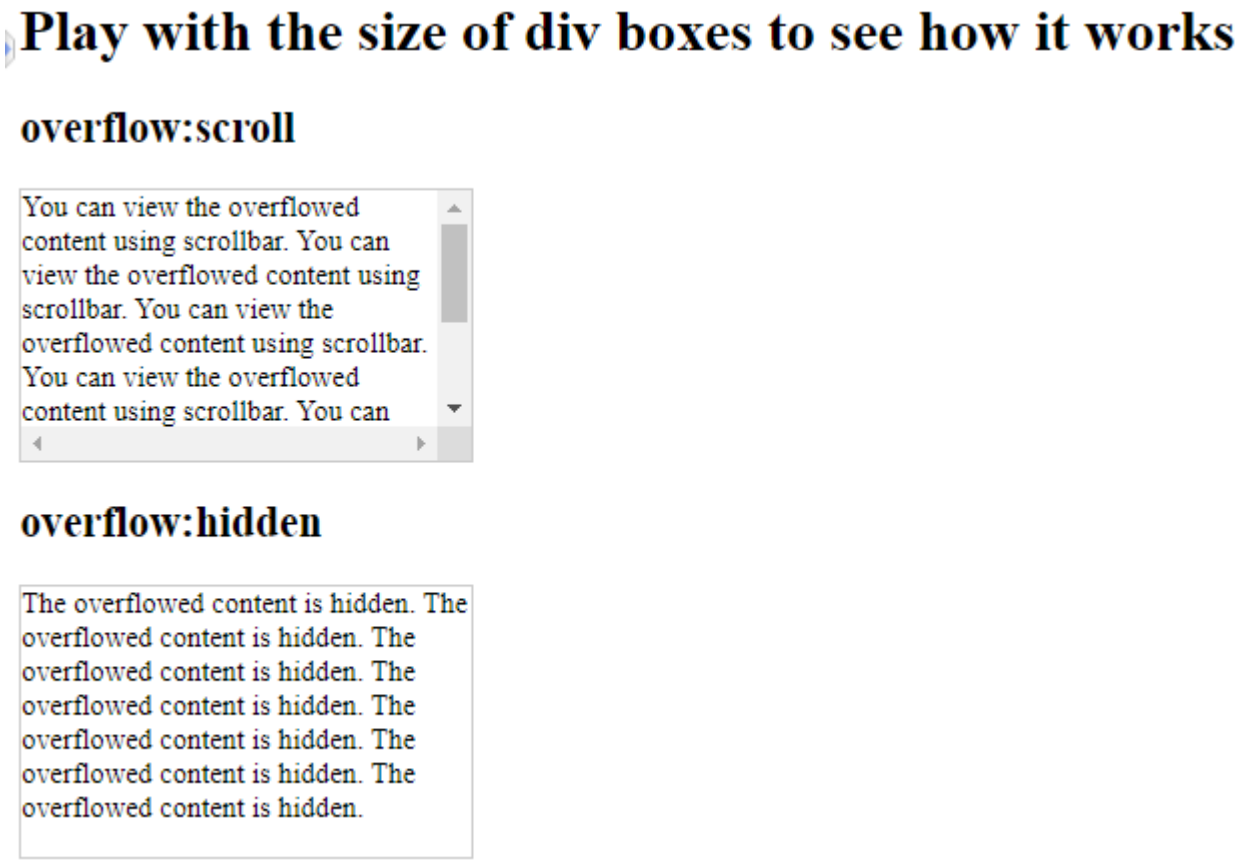
There may be a situation when the content of an element might be larger than the dimensions of the box itself. For example given width and height properties did not allow enough room to accommodate the content of the element.
CSS overflow property allowing you to specify whether to clip content, render scroll bars or display overflow content of a block-level element.

```
<html>
<head>
<title>Example of CSS overflow property</title>
  <style>
      div {
          width: 250px;    height: 150px;
          border: 1px solid #cccccc;
      }
      div.scroll {overflow: scroll;    }
      div.hidden {overflow: hidden; }
  </style>
</head>
```

```
<body>
  <h1>Play with the size of div boxes to see how it
works</h1>
  <h2>overflow:scroll</h2>
  <div class="scroll">
      You can view the overflowed content using
scrollbar.
      You can view the overflowed content using
scrollbar.
</div>
```

CSS overflow

```
<h2>overflow:hidden</h2>
  <div class="hidden">
      The overflowed content is hidden.
      The overflowed content is hidden.
      The overflowed content is hidden.
      The overflowed content is hidden.
      The overflowed content is hidden.
      The overflowed content is hidden.
      The overflowed content is hidden.
  </div>
</body>
</html>
```



CSS float

You can float elements to the left or right, but only applies to the elements that generate boxes that are not absolutely positioned. Any element that follows the floated element will flow around the floated element on the other side.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of Floating Elements</title>
<style>
    img {
        float: left;
        width: 150px;
        height: 150px;
        margin-right: 20px;
    }
</style>
</head>
```

```
<body>
            <p><img
src="/examples/images/kites.jpg" alt="Flying
Kites"> Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Nam eu sem tempor, varius quam
at, luctus dui. Mauris magna metus, dapibus nec
turpis vel, semper malesuada ante. Vestibulum id
metus ac nisl bibendum scelerisque non non
purus. Suspendisse varius nibh non aliquet
sagittis. In tincidunt orci sit amet elementum
vestibulum. Vivamus fermentum in arcu in
aliquam. Quisque aliquam porta odio in fringilla.
Vivamus faucibus ligula.</p>
</body>
</html>
```

CSS media query

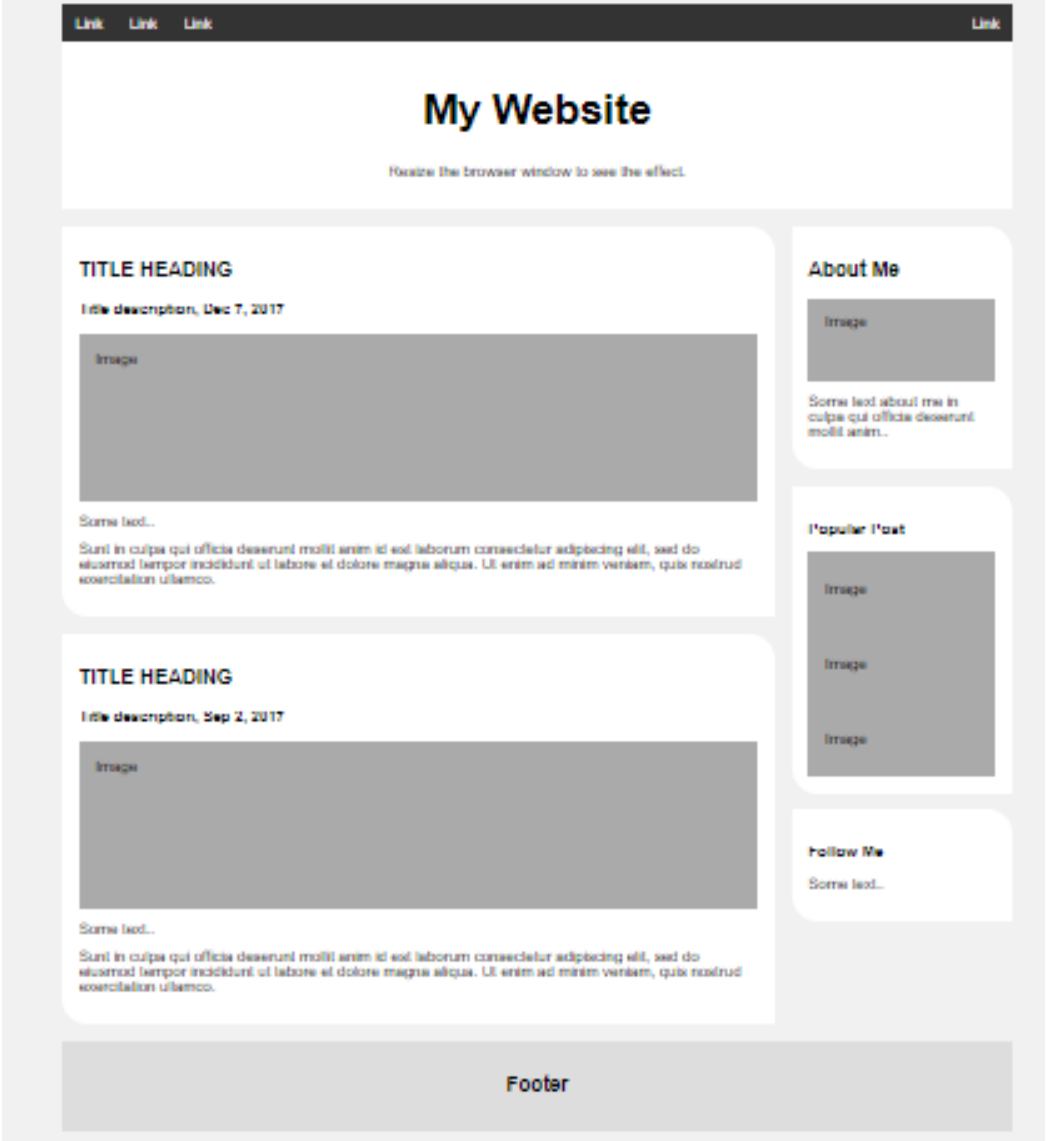
Media queries allow you to customize the presentation of your web pages for a specific range of devices like mobile phones, tablets, desktops, etc. without any change in markups. A media query consists of a media type and zero or more expressions that match the type and conditions of a particular media features such as device width or screen resolution.

Since media query is a logical expression it can be resolve to either true or false. The result of the query will be true if the media type specified in the media query matches the type of device the document is being displayed on, as well as all expressions in the media query are satisfied. When a media query is true, the related style sheet or style rules are applied to the target device.

```
/* Smartphones (portrait and landscape) -------
--- */ @media screen and (min-width: 320px) and
(max-width: 480px){ /* styles */ }
```

```
/* Desktops and laptops ---------- */ @media
screen and (min-width: 1224px){ /* styles */ }
/* Large screens ---------- */ @media screen
and (min-width: 1824px){ /* styles */ }
```

Complete Design



Complete Layout Design Example

```
<html>
<head>
<title>Complete Site Layout</title>
<style>
* {
  box-sizing: border-box;
}

#container{
  width:1140px;
  min-height: 800px;
  margin:auto auto;
}

body {
  font-family: Arial;
  padding: 10px;
  background: #f1f1f1;
}
```

```
/* Header/Blog Title */
.header {
  padding: 20px;  text-align: center;
  background: white;}

.header h1 {  font-size: 50px;}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;  display: block;  color: #f2f2f2;
  text-align: center;  padding: 14px 16px;
  text-decoration: none;
}
```

Complete Layout Design Example cont.

```
/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Create two unequal columns that floats next to each
other */
/* Left column */
.leftcolumn {     float: left;  width: 75%;}

/* Right column */
.rightcolumn {
  float: left;  width: 25%;
  background-color: #f1f1f1;
  padding-left: 20px;
}
```

```
/* Fake image */
.fakeimg {  background-color: #aaa;
  width: 100%;  padding: 20px;}

/* Add a card effect for articles */
.card {
  background-color: white;  padding: 20px;
  margin-top: 20px;  border: 1px #ccddee;
  border-radius: 0px 30px 0px 30px;}

/* Clear floats after the columns */
.row:after {  content: "";  display: table;
  clear: both;
}

/* Footer */
.footer {  padding: 20px;  text-align: center;
  background: #ddd;  margin-top: 20px;}
```

Complete Layout Design Example cont..

```
/* Responsive layout - when the screen is less than
800px wide, make the two columns stack on top of each
other instead of next to each other */
@media screen and (max-width: 800px) {
  .leftcolumn, .rightcolumn {
    width: 100%;
    padding: 0;
  }
}
/* Responsive layout - when the screen is less than
400px wide, make the navigation links stack on top of
each other instead of next to each other */
@media screen and (max-width: 400px) {
  .topnav a {
    float: none;
    width: 100%;
  }
}
</style>
```

```
</head>
<body>
  <div id="container">
  <div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#" style="float:right">Link</a>
</div>

<div class="header">
  <h1>My Website</h1>
  <p>Resize the browser window to see the
effect.</p>
</div>
```

Complete Layout Design Example cont…

```
<div class="row">
  <div class="leftcolumn">
    <div class="card">
      <h2>TITLE HEADING</h2>
      <h5>Title description, Dec 7, 2017</h5>
      <div class="fakeimg"
style="height:200px;">Image</div>
      <p>Some text..</p>
      <p>Sunt in culpa qui officia deserunt mollit anim id
est laborum consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation
ullamco.</p>
    </div>
```

```
<div class="card">
      <h2>TITLE HEADING</h2>
      <h5>Title description, Sep 2, 2017</h5>
      <div class="fakeimg"
style="height:200px;">Image</div>
      <p>Some text..</p>
      <p>Sunt in culpa qui officia deserunt mollit
anim id est laborum consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco.</p>
    </div>
  </div>
```

Complete Layout Design Example cont….

```
<div class="rightcolumn">
    <div class="card">
      <h2>About Me</h2>
      <div class="fakeimg"
style="height:100px;">Image</div>
      <p>Some text about me in culpa qui officia deserunt
mollit anim..</p>
    </div>
    <div class="card">
      <h3>Popular Post</h3>
      <div class="fakeimg"><p>Image</p></div>
      <div class="fakeimg"><p>Image</p></div>
      <div class="fakeimg"><p>Image</p></div>
    </div>
```

```
<div class="card">
      <h3>Follow Me</h3>
      <p>Some text..</p>
    </div>
  </div>
</div>

<div class="footer">
  <h2>Footer</h2>
</div>


  </div>


</body>
</html>
```