



WEB SYSTEM & TECHNOLOGY CSS - LAB MANUAL

By: DR. ZEESHAN BHATTI

For Video Lecture Visit

<https://www.youtube.com/zeeshanacademy>

For Full Android App Development Course “Z-Tech Tutorials”

<https://www.youtube.com/c/ZTechTutorials/>

Cascading Style Sheets (CSS) properties

Attribute	Property	Possible Values
background-attachment	value	fixed, default
background-color:	color	a color value
background-image:	location	none, url(location)
background-position	value	
background-repeat	value	
border:	width, style, color	value for width dotted, dashed, inset a color value
border-bottom:	width, style, color	value for width dotted, dashed, inset a color value
border-left:	width, style, color	value for width dotted, dashed, inset a color value
border-right:	width, style, color	value for width dotted, dashed, inset a color value
border-top:	width, style, color	value for width dotted, dashed, inset a color value
clear:	value	none, left, right, both
color:	color	a color value
cursor:	value	auto, crosshair, default, hand, move, text, wait, help, n-resize, ne-resize, nw-resize, s-resize, se-resize, sw-resize, e-resize, w-resize
display	value	block, inline
font-family:	familyName	Times, Helvetica, serif, sans-serif, Courier, Western, cursive
font-size:	size	xx-small, small, medium, large, x-large, xx-large value in cm, mm, in, pt, px, pc percentage
font-style:	style	normal, italic, oblique
font-variant:	variant	normal, small-caps
font-weight:	weight	normal, bold, bolder, lighter, 100, 200, 300,... 900

height:	value	value in cm, mm, in, pt, px, pc percentage
letter-spacing:	value	normal value in cm, mm, in, pt, px, pc
line-height:	value	normal value in cm, mm, in, pt, px, pc percentage
list-style-image:	URL	url(location)
list-style-type:	type	disc, circle, square, decimal, none lower-roman, upper-roman, lower-alpha, upper-alpha
margin:	value	value in cm, mm, in, pt, px, pc percentage
margin-left:	value	value in cm, mm, in, pt, px, pc percentage
margin-right:	value	value in cm, mm, in, pt, px, pc percentage
margin-top:	value	value in cm, mm, in, pt, px, pc percentage
padding:	value	value in cm, mm, in, pt, px, pc percentage
text-align:	alignment	left, right, center, justify
text-decoration:	decoration	none, underline, overline, line-through, blink
text-indent:	indentation	value in cm, mm, in, pt, px, pc percentage
text-transform:	transformation	none, capitalize, uppercase, lowercase
vertical-align:	alignment	baseline, sub, super, top, middle, bottom, text-top, text-bottom
visibility:	visibility	inherit, visible, hidden
white-space	value	value in cm, mm, in, pt, px, pc percentage
width:	value	value in cm, mm, in, pt, px, pc percentage
word-spacing	value	value in cm, mm, in, pt, px, pc percentage

Pseudo-Classes

:active

:hover

:link

:visited

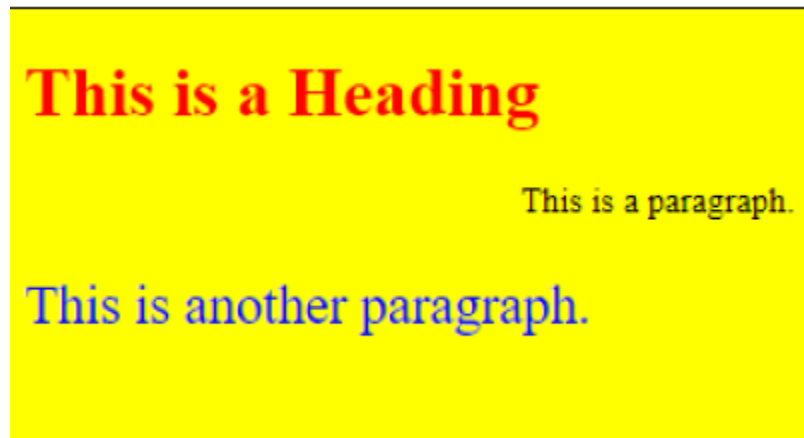
Pseudo-Elements
:first-letter
:first-line

Exercises

Exercise 1: USING INLINE STYLE

In this task you will create an HTML page, and apply CSS style inline to different elements on the page using the element's style attribute. The output should look as shown in the figure below:

Output for Task 1



Solution

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body style="background-color: yellow">
    <h1 style="color: red">This is a Heading</h1>
    <p style="text-align:right">This is a paragraph.</p>
    <p style="color:blue;font-size: 25px">This is another paragraph.</p>
  </body>
</html>
```

Exercise 2: USING INTERNAL STYLE SHEET

In this task you will create an HTML page, and apply CSS style within the <style> element, inside the <head> section of an HTML page. The output should look as shown in the figure below:

Output for Task 2

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {background-color: pink;}
      h1 {color: red; text-align:center;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

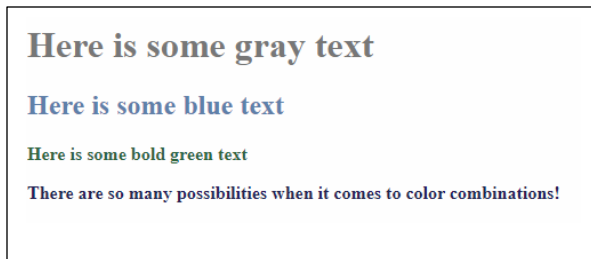
Exercise 3: Using CSS set the background color of h1, H2, p, and body tags.

Exercise 4: Using CSS Set a background image for a page

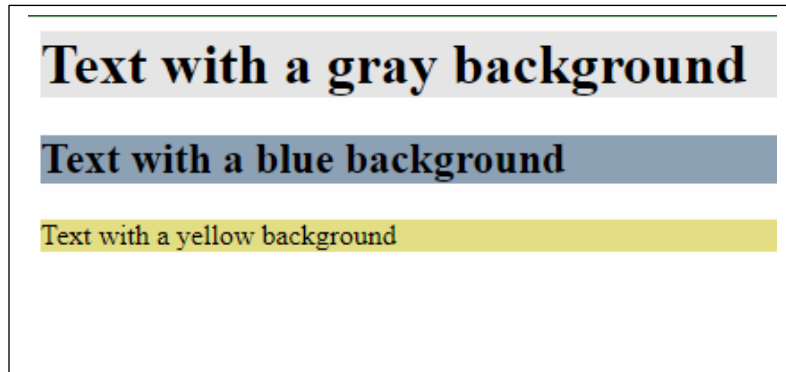
Exercise 5: Set a background image that repeats for the whole page

Exercise 6: Set a background image that is center positioned

Exercise 7: Set different text decorations as shown in figure



Exercise 8: Setting text background color



Exercise 9: Setting text decoration

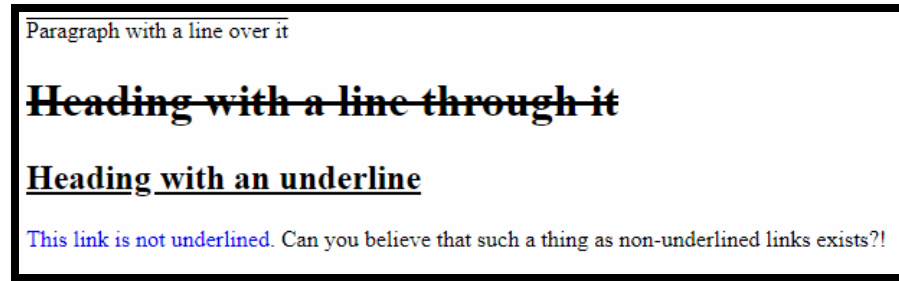
```
<html>
  <head>
    <style type="text/css">
      h1 {text-decoration: line-through;}
      h2 {text-decoration: underline;}
      p {text-decoration: overline;}
      a {text-decoration: none;}
    </style>
  </head>
<body>
  <p>
    Paragraph with a line over it
  </p>

  <h1>
    Heading with a line through it
  </h1>

  <h2>
    Heading with an underline
  </h2>
```

Tip: use the following code for link text with no underline.

`This link is not underlined`. Can you believe that such a thing as non-underlined links exists?!



Exercise 10: Use CSS to change the font type of text differently for each of the heading tags from h1 to h6. All heading tags must have different font

Exercise 11: Use CSS to create Unordered lists with different bullets

```
<html>
  <head>
    <style type="text/css">
      ol.inside {list-style-position: inside;}
      ol.outside {list-style-position: outside;}
    </style>
  </head>
  <body>
    <ol class="inside">
      <li>HTML</li>
      <li>XHTML</li>
      <li>CSS</li>
      <li>Javascript</li>
    </ol>

    <ol class="outside">
      <li>HTML</li>
      <li>XHTML</li>
      <li>CSS</li>
      <li>Javascript</li>
    </ol>
  </body>
</html>
```

Exercise 12: Use CSS to create Ordered lists with different bullets

Exercise 13: Use CSS to create a list Using an image for bullets in a list

Exercise 14: USE CSS to set styles of the left, Right, Top and bottom boarder

```
<style type="text/css">
  p.dotted {border-left-style: dotted}
  p.dashed {border-left-style: dashed}
  p.solid {border-left-style: solid}
  p.double {border-left-style: double}
  h1.groove {border-left-style: groove}
  h2.ridge {border-left-style: ridge}
  h3.inset {border-left-style: inset}
  h4.outset {border-left-style: outset}
```

```
</style>
```

Exercise 15 : Changet the Indentation of the Left , Right , TOP Margins using CSS.

```
<style type="text/css">
  p.margOne {margin-left: 14px;}
  p.margTwo {margin-left: 30px;}
  p.margThree {margin-left: 40%;}

  p.relative {left: 50px; position: relative;}
  h1 {top: -10px; position: relative;}
</style>
```

Exercise 16 : What will be the output of the following code

```
<style type="text/css">
  img.default {height: auto;}
  img.bigger {height: 140px;}
  img.smaller {height: 20%;}
</style>
<body>
  
  
  
</body>
```

Exercise 17 : Set different margins for all four sides of a <p> element as shown in code below and apply it on a page.

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}

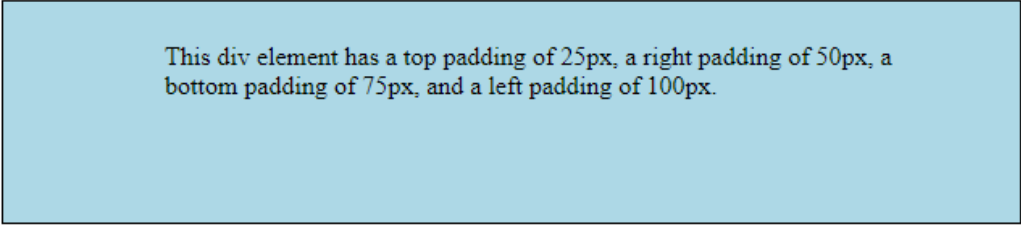
p.shortmargin {
  margin: 25px 50px 75px 100px;
}
```

Exercise 18: Set different padding for all four sides of a <div> element:

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
```

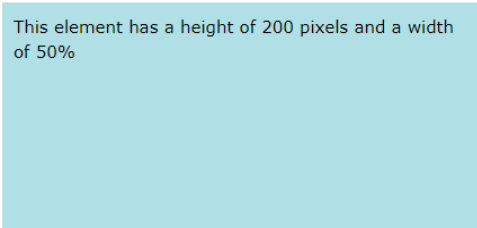
Exercise 19: Write the CSS for creating a padding of DIV tag with boarder and background color as shown below

By: Dr. Zeeshan Bhatti



This div element has a top padding of 25px, a right padding of 50px, a bottom padding of 75px, and a left padding of 100px.

Exercise 20: Create a DIV element with height of 200pixels, and with of 50% as shown below,

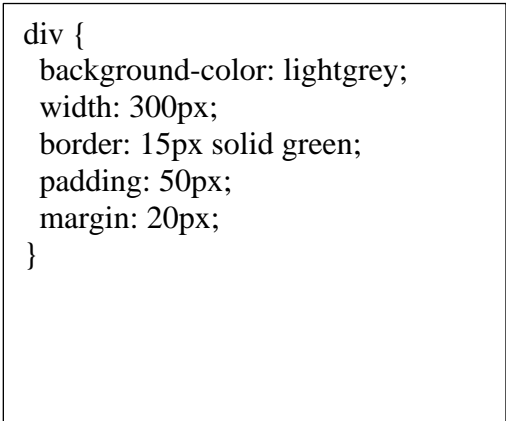


This element has a height of 200 pixels and a width of 50%

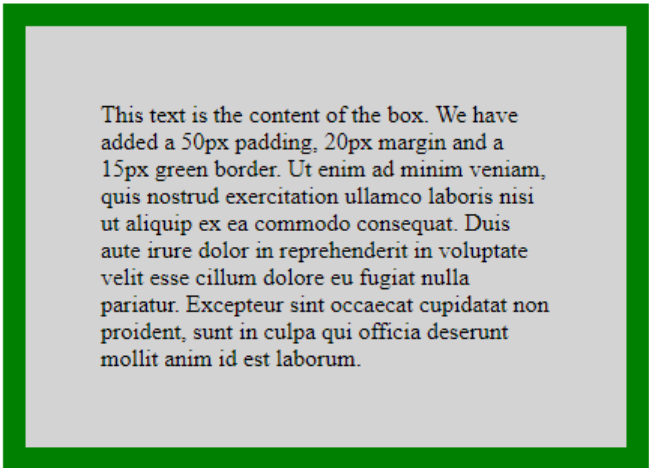
```
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
```

Exercise 21: The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

Create a box with following specifications

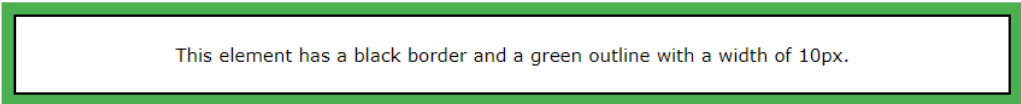


```
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```



This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Exercise 22: An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out".



This element has a black border and a green outline with a width of 10px.

Use the following Outline styles on paragraph tag

```
p.dotted {outline-style: dotted;}
p.dashed {outline-style: dashed;}
p.solid {outline-style: solid;}
p.double {outline-style: double;}
p.groove {outline-style: groove;}
p.ridge {outline-style: ridge;}
p.inset {outline-style: inset;}
p.outset {outline-style: outset;}
p.ex1 {
    border: 1px solid black;
    outline-style: solid;
    outline-color: red;
}
```

Exercise 23: Add various different Icons in your Webpage using Bootstrap Icons with following code.

To use the Bootstrap glyphs, add the following line inside the `<head>` section of your HTML page:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

Note: No downloading or installation is required!

```
<head>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.c
ss">
</head>
<body>

<i class="glyphicon glyphicon-cloud"></i>
<i class="glyphicon glyphicon-remove"></i>
<i class="glyphicon glyphicon-user"></i>
<i class="glyphicon glyphicon-envelope"></i>
<i class="glyphicon glyphicon-thumbs-up"></i>

</body>
```

To use the Google icons, add the following line inside the `<head>` section of your HTML page:

```
<link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
```

Note: No downloading or installation is required!

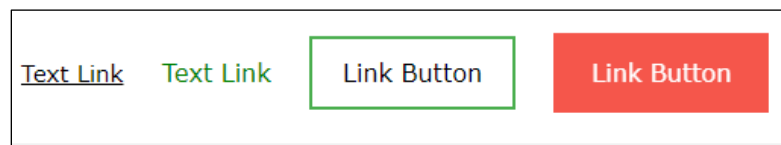
```
<head>
    <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
</head>
<body>
    <i class="material-icons">cloud</i>
    <i class="material-icons">favorite</i>
    <i class="material-icons">attachment</i>
    <i class="material-icons">computer</i>
    <i class="material-icons">traffic</i>
</body>
```

Exercise 24: Use the following link states on your custom link

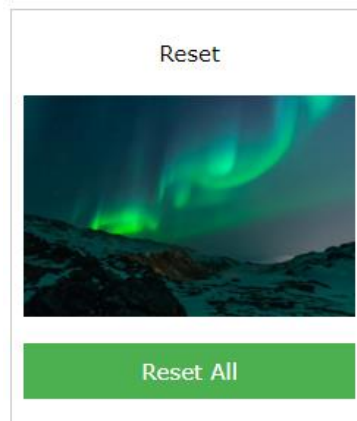
The four links states are:

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it
- `a:active` - a link the moment it is clicked

Use the Link state properties to create following type of links, with every link should change color when mouse cursor hovers on top of it.



Exercise 25: Use the CSS with Div tag to create following image gallery layout with boarder .



Exercise 26: use the Position property to set the position of each div tag. The `position` property specifies the type of positioning method used for an element. There are five different position values:

- `static`
- `relative;` is positioned relative to its normal position.
- `fixed`
- `absolute`
- `sticky`

```
div.static {
  position: static;
  border: 3px solid #73AD21;
}
```

An element with `position: static;` is not positioned in any special way; it is always positioned according to the normal flow of the page:

```
div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
```

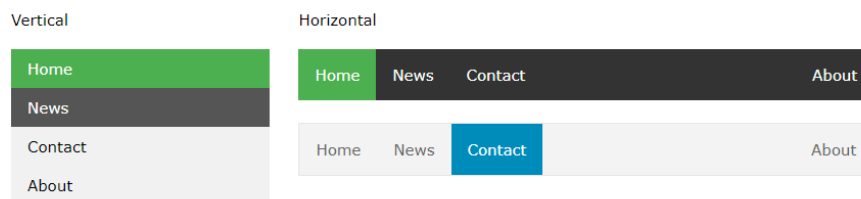
An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

```
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
```

An element with `position: sticky;` is positioned based on the user's scroll position. A sticky element toggles between `relative` and `fixed`, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like `position: fixed`).

```
div.sticky {
  position: -webkit-sticky; /* Safari */
  position: sticky;
  top: 0;
  background-color: green;
  border: 2px solid #4CAF50;
}
```

Exercise 28: Use the CSS to create a Horizontal and Vertical Navigation Bar as shown in figure



Vertical Nav Bar

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}
```

```

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

/* Change the link color on hover */
li a:hover {
  background-color: #555;
  color: white;
}

```

Exercise 29: Create a Navigation Menu that occupy full left side of screed, with full height, sticky navigation menu.



```

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 25%;
  background-color: #f1f1f1;
  height: 100%; /* Full height */
  position: fixed; /* Make it stick, even on scroll */
  overflow: auto; /* Enable scrolling if the sidenav has too much content */
}

```

Exercise 30: Create a Simple Horizontal Navigation menu as shown below

[Home](#) [News](#) [Contact](#) [About](#)

```

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
}

```

```

li {
  float: left;
}

li a {
  display: block;
  padding: 8px;
  background-color: #dddddd;
}

```

Example explained:

- **float: left;** - use float to get block elements to slide next to each other
- **display: block;** - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify padding (and height, width, margins, etc. if you want)
- **padding: 8px;** - Since block elements take up the full width available, they cannot float next to each other. Therefore, specify some padding to make them look good
- **background-color: #dddddd;** - Add a gray background-color to each a element

Now add color and block properties to the above CSS to create following menu style



```

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change the link color to #111 (black) on hover */
li a:hover {
  background-color: #111;
}

```

Add an "active" class to the current link to let the user know which page he/she is on:

```

.active {
  background-color: #4CAF50;
}

```

Add `position: sticky;` to `` to create a sticky navbar.

```
ul {
  position: -webkit-sticky; /* Safari */
  position: sticky;
  top: 0;
}
```

Exercise 31: Create a Dropdown navigation menu

```
<style>
/* Style The Dropdown Button */
.dropbtn {
  background-color: #4CAF50;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
  cursor: pointer;
}

/* The container <div> - needed to position the dropdown content */
.dropdown {
  position: relative;
  display: inline-block;
}

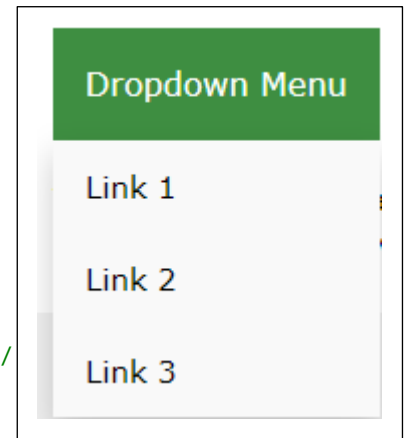
/* Dropdown Content (Hidden by Default) */
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

/* Links inside the dropdown */
.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: #f1f1f1}

/* Show the dropdown menu on hover */
.dropdown:hover .dropdown-content {
  display: block;
}

/* Change the background color of the dropdown button when the dropdown content is shown */
.dropdown:hover .dropbtn {
  background-color: #3e8e41;
}
```



By: Dr. Zeeshan Bhatti

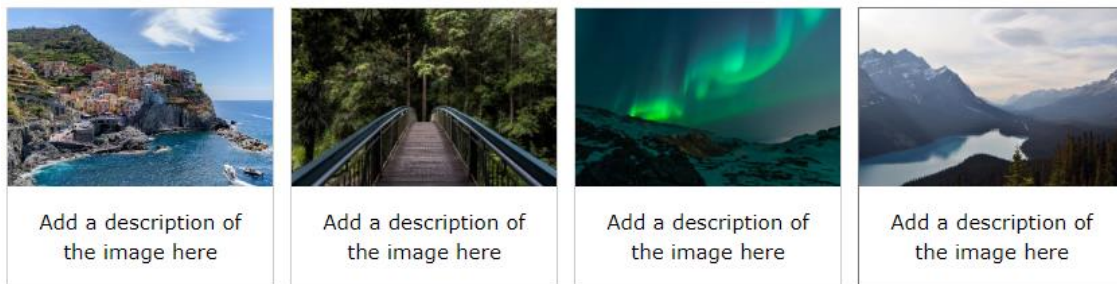
```

}
</style>

<div class="dropdown">
  <button class="dropbtn">Dropdown</button>
  <div class="dropdown-content">
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
  </div>
</div>

```

Exercise 32: Create a Image Gallery with multiple thumbnails as shown



```

<html>
<head>
<style>
div.gallery {
  margin: 5px;
  border: 1px solid #ccc;
  float: left;
  width: 180px;
}

div.gallery:hover {
  border: 1px solid #777;
}

div.gallery img {
  width: 100%;
  height: auto;
}

div.desc {
  padding: 15px;
  text-align: center;
}
</style>
</head>
<body>

<div class="gallery">
  <a target="_blank" href="img_5terre.jpg">

```

By: Dr. Zeeshan Bhatti


```

    
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_forest.jpg">
    
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

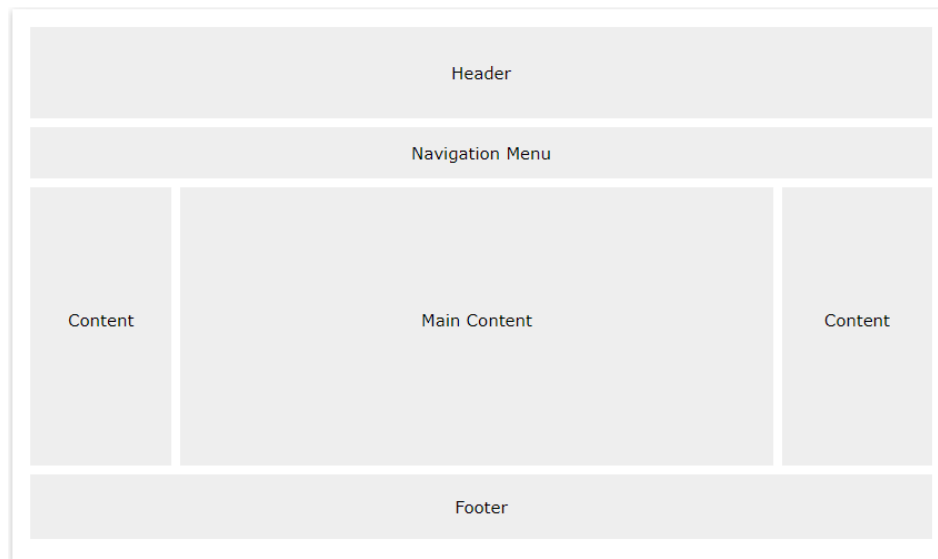
<div class="gallery">
  <a target="_blank" href="img_lights.jpg">
    
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    
  </a>
  <div class="desc">Add a description of the image here</div>
</div>

</body>
</html>

```

Exercise 33: Create a Website Layout using CSS and Div tags as shown in figure below. The Style sheet is also given, add the relevant div tags in html.



```

.header {
  background-color: #F1F1F1;

```

By: Dr. Zeeshan Bhatti

```

    text-align: center;
    padding: 20px;
}

/* The navbar container */
.topnav {
    overflow: hidden;
    background-color: #333;
}

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

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

/* Create three EQUAL columns that floats next to each other
    Tip: To create a 2-column layout, change the width to 50%.
    To create a 4-column layout, use 25%, etc. */
    /* .column {
        float: left;
        width: 33.33%;
    }
    */

/* Create three UN-EQUAL columns that floats next to each other
    Tip: To create a 2-column layout, change the width to 50%.
    To create a 4-column layout, use 25%, etc. */
    .column {
        float: left;
    }

    /* Left and right column */
    .column.side {
        width: 25%;
    }

    /* Middle column */
    .column.middle {
        width: 50%;
    }

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

```

```
clear: both;
}

/* Responsive layout - makes the three columns stack on top of each other instead of next to each
other on smaller screens (600px wide or less) */
@media screen and (max-width: 600px) {
  .column {
    width: 100%;
  }
}

.footer {
  background-color: #F1F1F1;
  text-align: center;
  padding: 10px;
}
```

LAB PROJECTS

Project 1:

In previous lab you created a Personal Portfolio website, Now convert your entire Portfolio website into a TABLE or DIV based layout with CSS based styling.

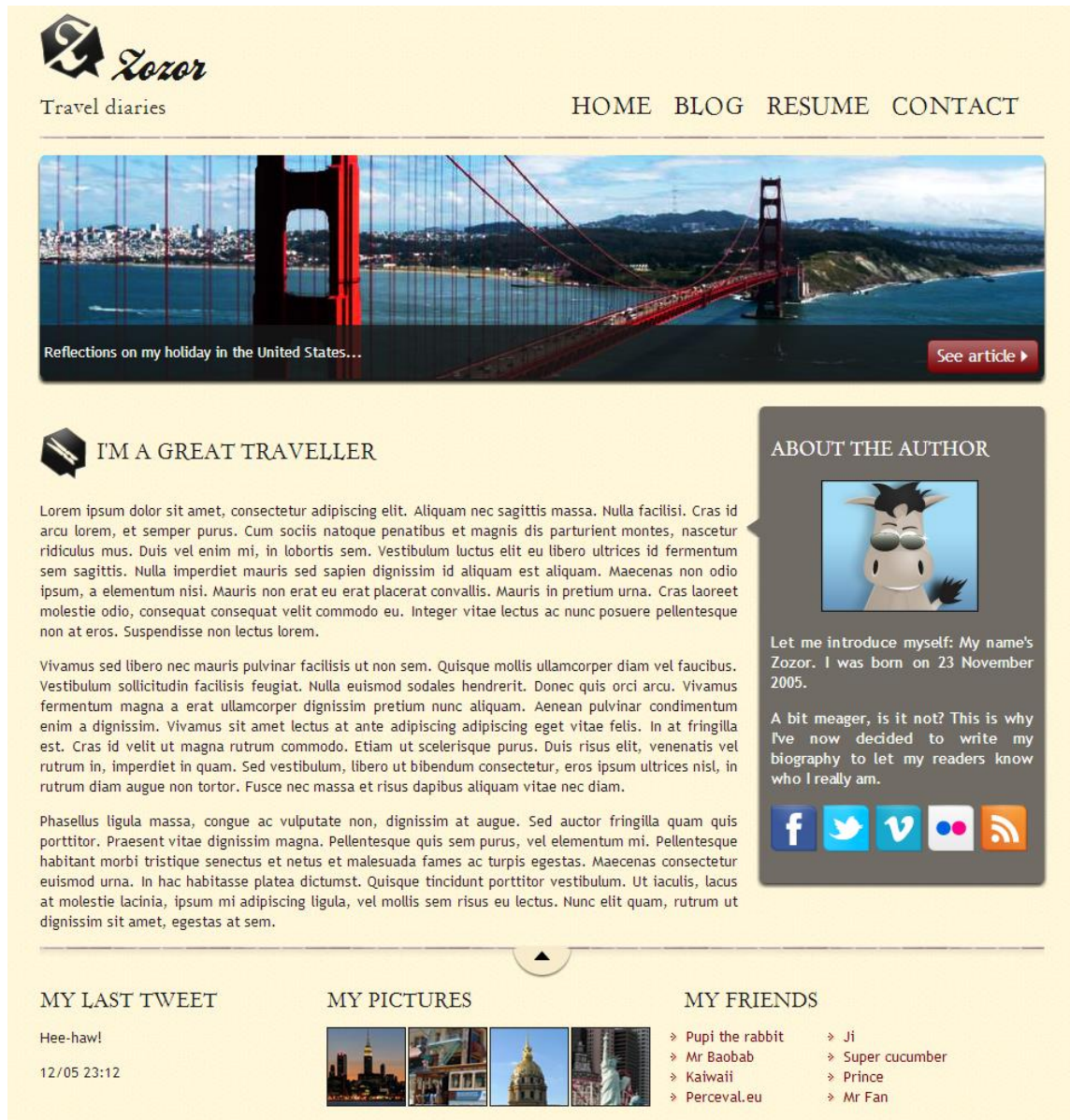
Please Ensure:

- Every page must have same and complete color full design.
- All Links must be present on every page and must be working.
- Use color and images to create appealing design.
- Fill your webpage with as much data as possible, do not leave your page empty with very little data.
- Find a free web hosting web services and upload your website on it.

Project 2:

Create a Website using HTML and CSS based on the image template as shown below. Use HTML and CSS with Div tags to create the website with all working links.

For development steps, see the **“Chapter 3 - part4 Creating Website from Start”** on my website.



Project 3:

Create a Tourist attraction website based on VILLAGE/CITY/TOWN that you live in. The Website must contain all the relevant information and data about your Home Town / Village / City with at least 12 main links. Use external CSS to design and position the web contents. Put all the relevant information such as

- Tourist Places to Visit,
 - Main Monuments, Attractions, Iconic Buildings
 - Market, shops, Mall, shopping centers, etc
 - Famous Restaurants, Food Corners, Popular stops
 - Traveling details, how to get their, Bus and train routes
 - Hotels, Guest houses or other places to stay.
 - Things or activities to be done once you we visit.
 - Mosques, Dargas, etc.
 -
- Famous things about your Village, like, foot, cloths, crockery, places, etc.

Project 4 (OPTIONAL - IDEA)

A Tribute Page for your Favorite Personality/Poet/Writer/Sportsman etc.

The simplest website you can make is a tribute page of someone you admire in your life. It requires only basic knowledge of HTML and CSS. Make a webpage writing about that person adding his/her image. On the top of the webpage, add the image and name of the person and below that give layout for the rest of the details. You can use paragraphs, lists, links, images with CSS to give it a descent look. Add a suitable background color and font style on your webpage. Most of the parts you can make using HTML but to give it a better look using a bit of CSS. Take help from the link given below

Project 5: (OPTIONAL - IDEA) Restaurant Website

Showcase your solid knowledge of HTML and CSS creating a beautiful webpage for a restaurant. Making a layout for a restaurant will be a bit complicated than previous project examples. You will be aligning the different food items and drinks using a CSS layout grid. You will be adding prices, images and you need to give it a beautiful look and feel as well using the proper combination of colors, font-style and images. You can add pictures gallery for different food items, you can also add sliding images for a better look. Add links for redirection to internal pages. Make it responsive setting a viewport, using media queries and grid.

Project 6: (OPTIONAL - IDEA) NOTES / LECTURE Website

Website for students to download notes, lectures or handouts of any discipline or for your department.