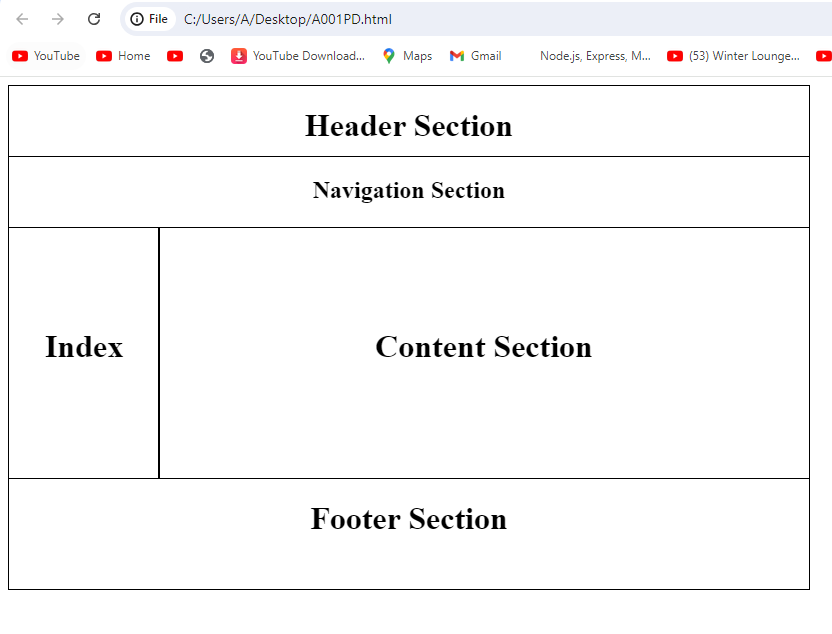
# HTML, CSS3 & Javascript\_1/300 Projects

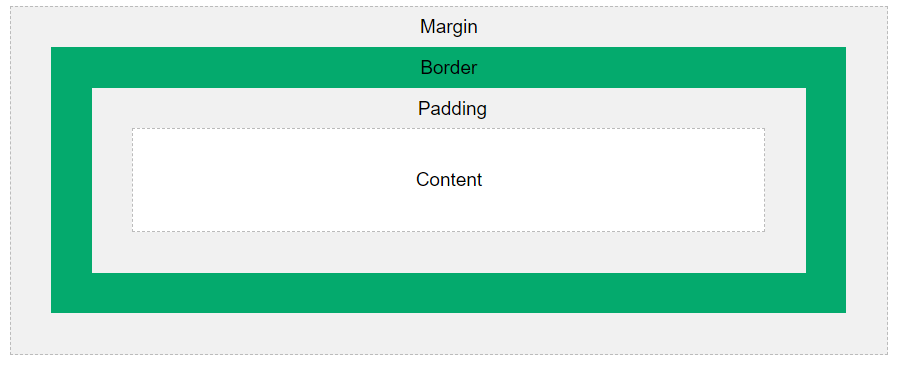


## Exercises

The CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

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



**Explanation of the different parts:**

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

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content

Margin - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

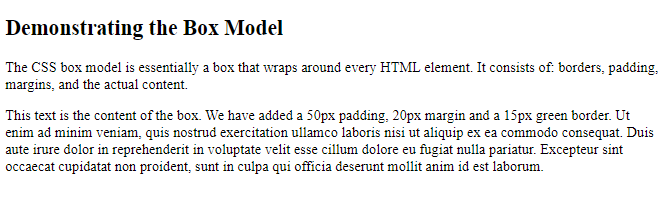
# Example 1

1. <!DOCTYPE html>
2. <html>
3. </html>

# Example 2

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>Demonstrating the Box Model</h2>
5. <p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
6. <div>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.</div>
7. </body>
8. </html>

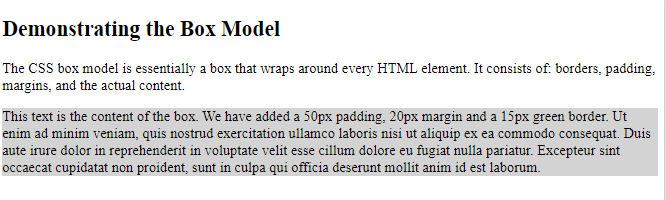
# Out Put



# Example 3

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div {
6. background-color: lightgrey;
7. }
8. </style>
9. </head>
10. <body>
11. <h2>Demonstrating the Box Model</h2>
12. <p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
13. <div>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.</div>
14. </body>
15. </html>

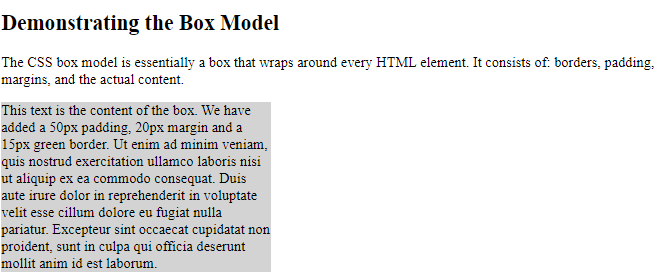
# Output



# Example 4

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div {
6. background-color: lightgrey;
7. width: 300px;
8. }
9. </style>
10. </head>
11. <body>
12. <h2>Demonstrating the Box Model</h2>
13. <p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
14. <div>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.</div>
15. </body>
16. </html>

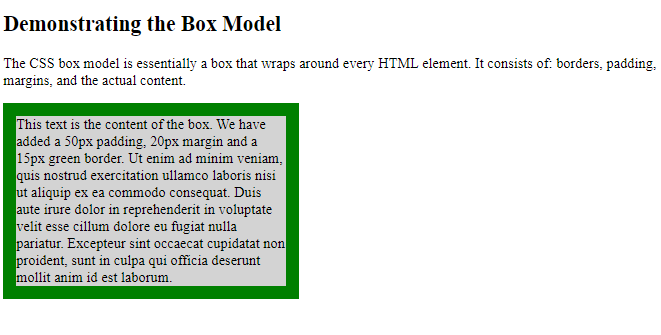
# Output



# Example 5

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div {
6. background-color: lightgrey;
7. width: 300px;
8. border: 15px solid green;
9. }
10. </style>
11. </head>
12. <body>
13. <h2>Demonstrating the Box Model</h2>
14. <p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
15. <div>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.</div>
16. </body>
17. </html>

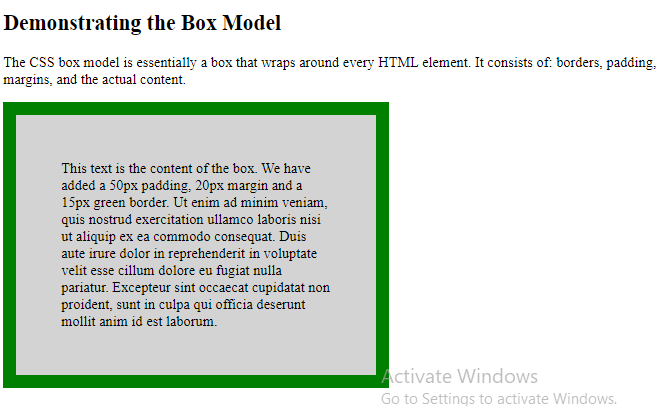
# Output



# Example 6

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div {
6. background-color: lightgrey;
7. width: 300px;
8. border: 15px solid green;
9. padding: 50px;
10. }
11. </style>
12. </head>
13. <body>
14. <h2>Demonstrating the Box Model</h2>
15. <p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
16. <div>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.</div>
17. </body>
18. </html>

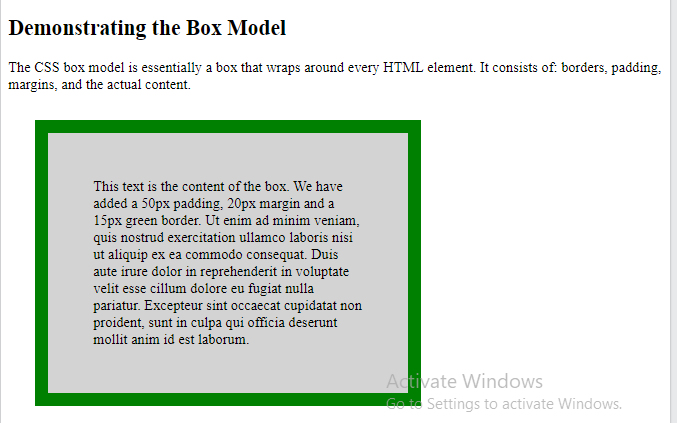
# Output

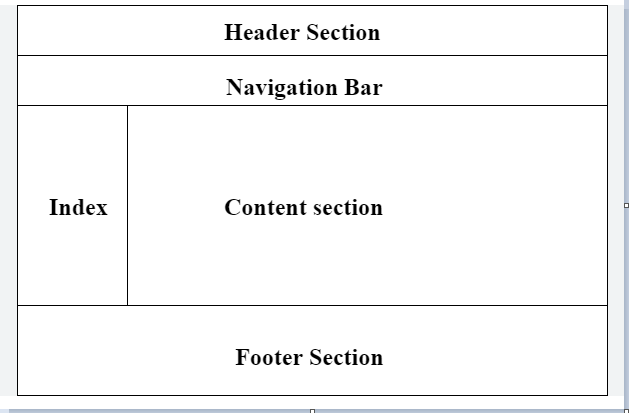


# Example 7

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div {
6. background-color: lightgrey;
7. width: 300px;
8. border: 15px solid green;
9. padding: 50px;
10. margin: 30px;
11. }
12. </style>
13. </head>
14. <body>
15. <h2>Demonstrating the Box Model</h2>
16. <p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>
17. <div>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.</div>
18. </body>
19. </html>

# Output





# A001PA.html

<!DOCTYPE html>

<head>

    <style>

        .header{

            width: 800px;

            height: 70px;

            border: 1px solid black;

        }

    </style>

</head>

<body>

        <div class="header">

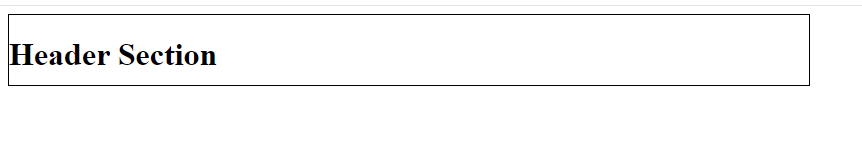
            <h1>Header Section</h1>

        </div>

</body>

</html>

# Output



# A001PB.html

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Project0001</title>

    <style>

        .header{

            width: 800px;

            height: 70px;

            border: 1px solid black;

        }

        .nav{

            width: 800px;

            height: 70px;

            border: 1px solid black;

        }

    </style>

</head>

<body>

    <!-- header section -->

    <header>

        <div class="header">

            <h1>Header Section</h1>

        </div>

    </header>

    <!-- navbar section -->

    <nav>

        <div class="nav">

            <h1>Navigation Section</h1>

        </div>

    </nav>

</body>

5</html>

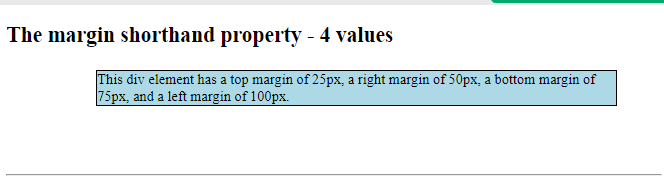
# Output



# Example 8

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div {
6. border: 1px solid black;
7. margin: 25px 50px 75px 100px;
8. background-color: lightblue;
9. }
10. </style>
11. </head>
12. <body>
13. <h2>The margin shorthand property - 4 values</h2>
14. <div>This div element has a top margin of 25px, a right margin of 50px, a bottom margin of 75px, and a left margin of 100px.</div>
15. <hr>
16. </body>
17. </html>

# OUTPUT



# Example 9

# position: fixed;

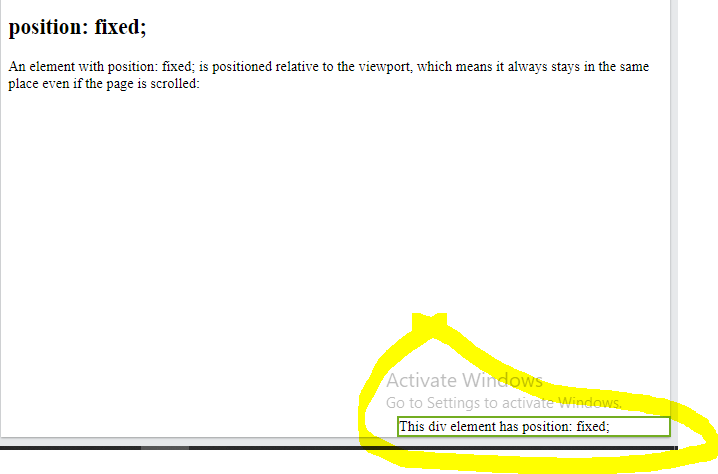
An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div.fixed {
6. position: fixed;
7. bottom: 0;
8. right: 0;
9. width: 300px;
10. border: 3px solid #73AD21;
11. }
12. </style>
13. </head>
14. <body>
15. <h2>position: fixed;</h2>
16. <p>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:</p>
17. <div class="fixed">
18. This div element has position: fixed;
19. </div>
20. </body>
21. </html>

# OUTPUT



For instance a website with a header and navigation that always stays visible when you are scrolling is a good example of fixed positioning. Another examples of fixed positioning would be advertisements that pop up on your screen. Some of these will always stay visible even if you start scrolling the page.

# A001PC.html

<!DOCTYPE html>

<head>

<title>Project0001</title>

<style>

.header{

width: 800px;

height: 70px;

border: 1px solid black;

position: fixed;

}

.nav{

width: 800px;

height: 70px;

border: 1px solid black;

text-align: center;

position: fixed;

margin-top: 71px;

}

.index{

width: 150px;

height: 250px;

border: 1px solid black;

text-align: center;

position: fixed;

margin-top: 142px;

}

.content{

width: 649px;

height: 250px;

border: 1px solid black;

text-align: center;

position: fixed;

margin-top: 142px;

margin-left: 151px;

}

</style>

</head>

<body>

<!-- header section -->

<header>

<div class="header">

<h1>Header Section</h1>

</div>

</header>

<!-- navbar section -->

<nav>

<div class="nav">

<h1>Navigation Section</h1>

</div>

</nav>

<!-- main section -->

<main>

<div class="index">

<h1> Index</h1>

</div>

<div class="content">

<h1> Content Section</h1>

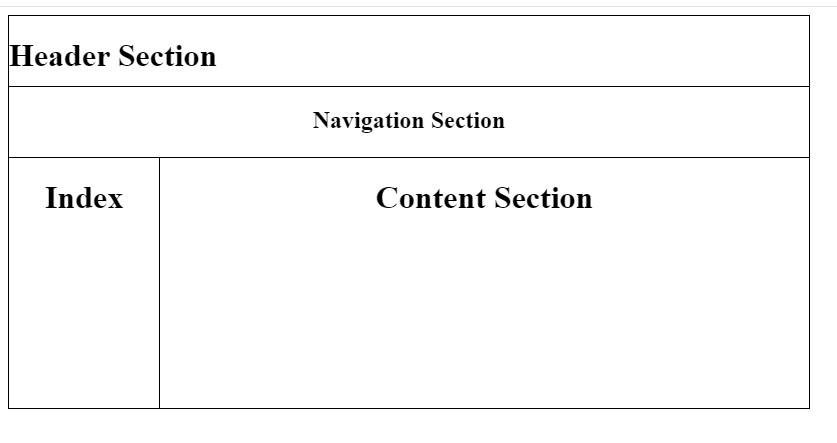
</div>

</main>

</body>

</html>

## Output



# Example 00010.html

<html>

<body>

<header>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</header>

<main>

<p>Page content 1</p>

<p>Page content 2</p>

<p>Page content 3</p>

<p>Page content 4</p>

<p>Page content 5</p>

<p>Page content 6</p>

<p>Page content 7</p>

<p>Page content 8</p>

<p>Page content 9</p>

<p>Page content 10</p>

<p>Page content 11</p>

<p>Page content 12</p>

<p>Page content 13</p>

<p>Page content 14</p>

<p>Page content 15</p>

<p>Page content 16</p>

<p>Page content 17</p>

<p>Page content 18</p>

</main>

<footer>

<p>Copyright &copy; 2024 | All rights reserved</p>

</footer>

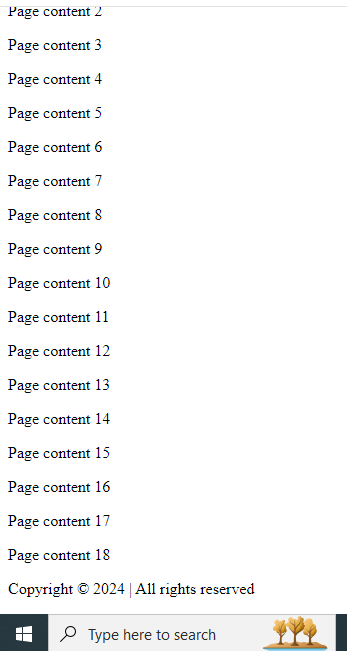
</body>

</html>

# Output



# Output part 2



# Example 00010B

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. span.a {
6. display: inline; /\* the default for span \*/
7. width: 100px;
8. height: 100px;
9. padding: 5px;
10. border: 1px solid blue;
11. background-color: yellow;
12. }
13. span.b {
14. display: inline-block;
15. width: 100px;
16. height: 100px;
17. padding: 5px;
18. border: 1px solid blue;
19. background-color: yellow;
20. }
21. span.c {
22. display: block;
23. width: 100px;
24. height: 100px;
25. padding: 5px;
26. border: 1px solid blue;
27. background-color: yellow;
28. }
29. </style>
30. </head>
31. <body>
32. <h1>The display Property</h1>
33. <h2>display: inline</h2>
34. <div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. <span class="a">Aliquam</span> <span class="a">venenatis</span> gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet. </div>
35. <h2>display: inline-block</h2>
36. <div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. <span class="b">Aliquam</span> <span class="b">venenatis</span> gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet. </div>
37. <h2>display: block</h2>
38. <div>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. <span class="c">Aliquam</span> <span class="c">venenatis</span> gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet. </div>
39. </body>
40. </html>

# Output



**The display: inline-block Value**

Compared to display: inline, the major difference is that display: inline-block allows to set a width and height on the element.

Also, with display: inline-block, the top and bottom margins/paddings are respected, but with display: inline they are not.

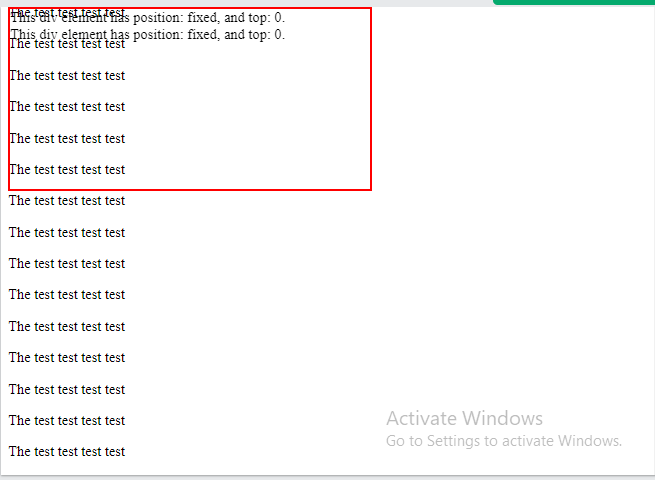
Compared to display: block, the major difference is that display: inline-block does not add a line-break after the element, so the element can sit next to other elements.

The following example shows the different behavior of display: inline, display: inline-block and display: block:

# Example 00010C

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div.a {
6. position: fixed;
7. top: 0;
8. width: 400px;
9. height: 200px;
10. border: 3px solid red;
11. }
12. </style>
13. </head>
14. <body>
15. <h1>The top Property</h1>
16. <p>The test test test test</p>
17. <p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p><p>The test test test test</p>
18. <div class="a">
19. <div >This div element has position: fixed, and top: 0.</div>
20. <div >This div element has position: fixed, and top: 0.</div>
21. </body>
22. </html>





# Example00011.html

<!DOCTYPE html>

<html>

<head>

<style>

body{

/\*This makes the box go all the way to the edge of the browser\*/

margin: 0px;

}

header{

/\*These next two lines make the header fixed to the top of the page \*/

position: fixed;

top: 0px;

text-align: center;

background: #eeeeee;

width:1200px;

padding: 10px;

}

header li{

display: inline-block;

}

header ul{

margin: 0px;

text-align: center;

padding: 10px;

}

p{

padding: 10px;

}

multiple5 {

background: #dddddd;

}

footer{

background: #dddddd;

text-align: center;

width:1200px;

}

</style>

</head>

<body>

<header>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</header>

<main>

<p>Page content 1</p>

<p>Page content 2</p>

<p>Page content 3</p>

<p>Page content 4</p>

<p>Page content 5</p>

<p>Page content 6</p>

<p>Page content 7</p>

<p>Page content 8</p>

<p>Page content 9</p>

<p>Page content 10</p>

<p>Page content 11</p>

<p>Page content 12</p>

<p>Page content 13</p>

<p>Page content 14</p>

<p>Page content 15</p>

<p>Page content 16</p>

<p>Page content 17</p>

<p>Page content 18</p>

</main>

<footer>

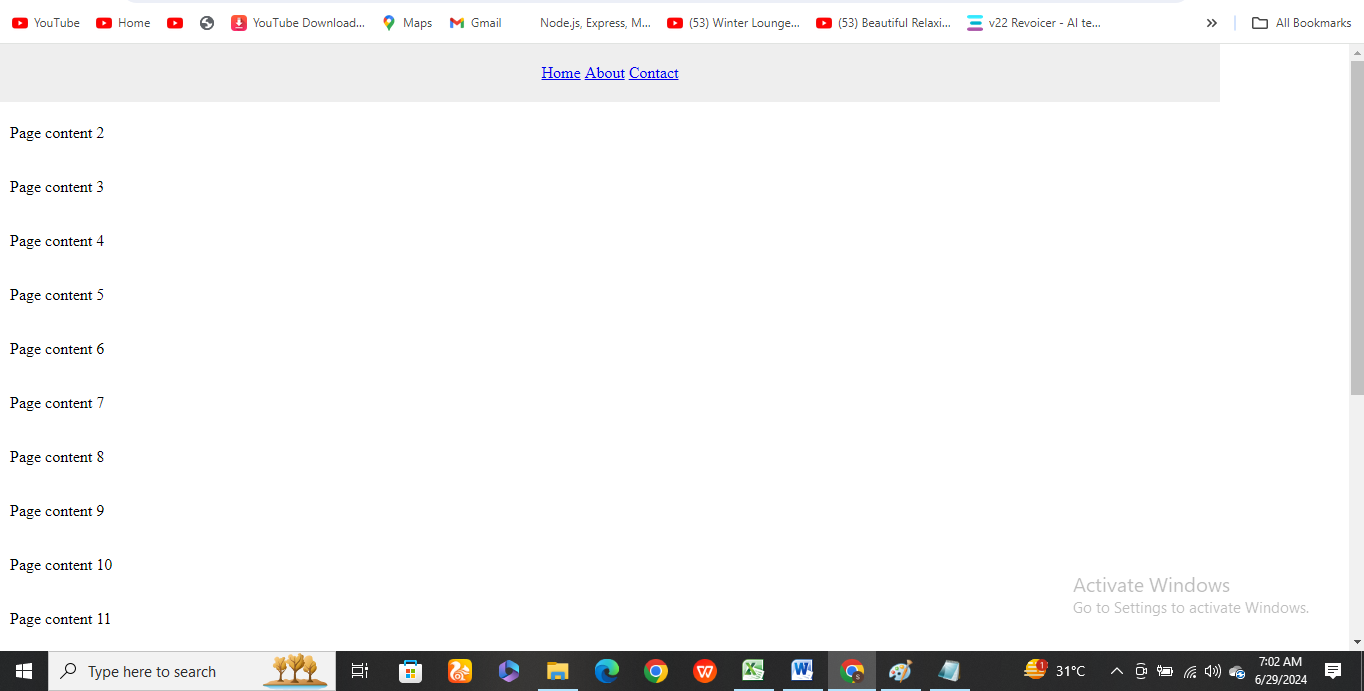
<p>Copyright &copy; 1st-July-2024 | All rights reserved</p>

</footer>

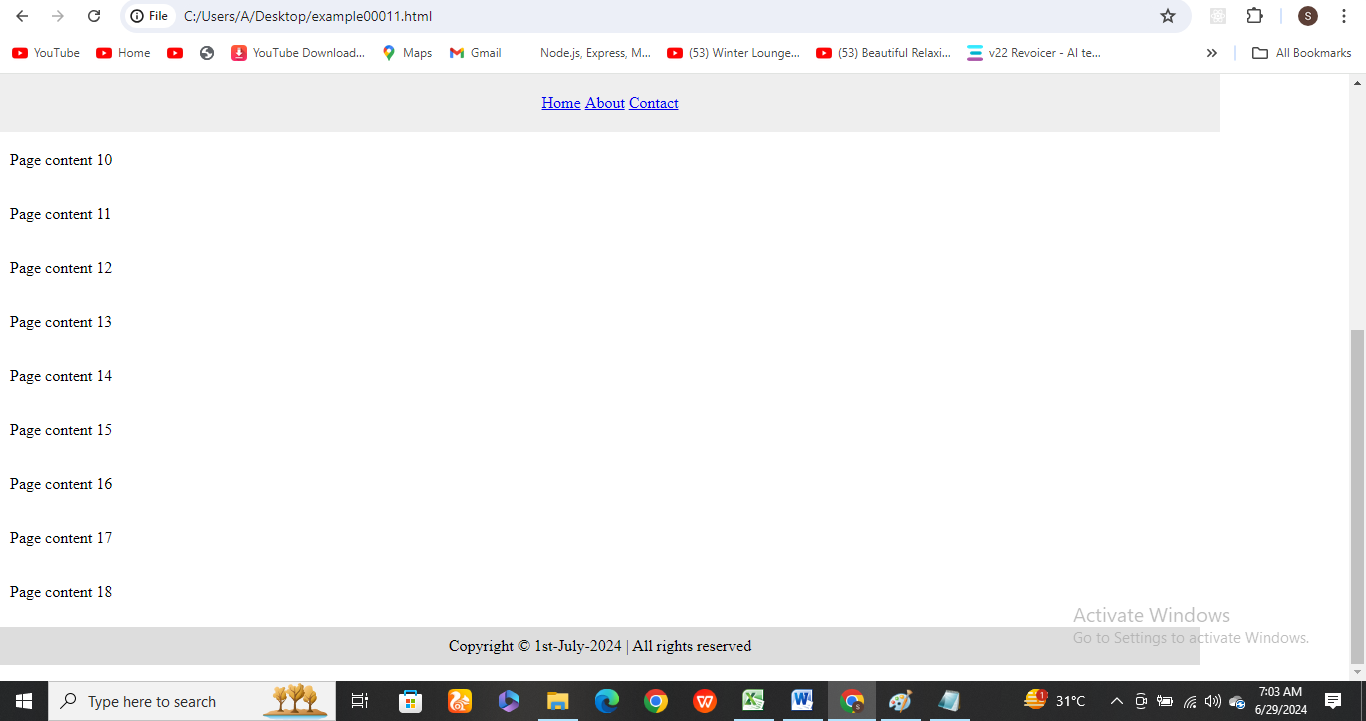
</body>

</html>

# Output 1/2



# Output 2/2



<!DOCTYPE html>

<head>

    <title>Project0001</title>

    <style>

        .header{

            width: 800px;

            height: 70px;

            border: 1px solid black;

            text-align: center;

            position: fixed;

        }

        .nav{

            width: 800px;

            height: 70px;

            border: 1px solid black;

            text-align: center;

            position: fixed;

            margin-top: 71px;

        }

        .index{

            width: 150px;

            height: 250px;

            border: 1px solid black;

            text-align: center;

            position: fixed;

            margin-top: 142px;

        }

        .ind{

            margin-top: 100px;

        }

        .content{

            width: 650px;

            height: 250px;

            border: 1px solid black;

            text-align: center;

            position: fixed;

            margin-top: 142px;

            margin-left: 150px;

        }

        .cntnt{

            margin-top: 100px;

        }

        .footer{

            width: 800px;

            height: 110px;

            border: 1px solid black;

            text-align: center;

            position: fixed;

            margin-top: 393px;

        }

    </style>

</head>

<body>

    <!-- header section -->

    <header>

        <div class="header">

            <h1>Header Section</h1>

        </div>

    </header>

    <!-- navbar section -->

    <nav>

        <div class="nav">

            <h1>Navigation Section</h1>

        </div>

    </nav>

    <!-- main section -->

    <main>

        <div class="index">

            <h1 class="ind">Index</h1>

        </div>

        <div class="content">

            <h1 class="cntnt">Contant Section</h1>

        </div>

    </main>

    <!-- footer section  -->

    <footer>

        <div class="footer">

            <h1>Footer Section</h1>

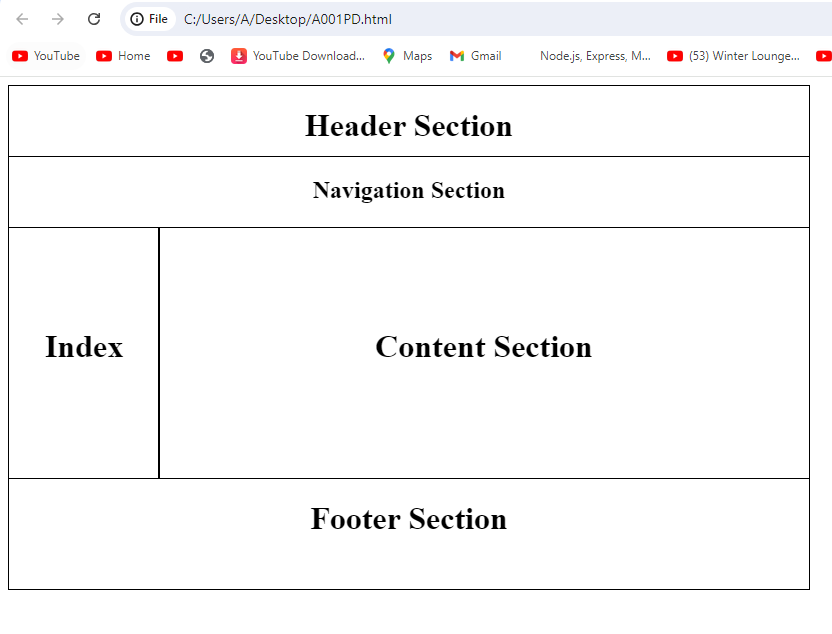
        </div>

    </footer>

</body>

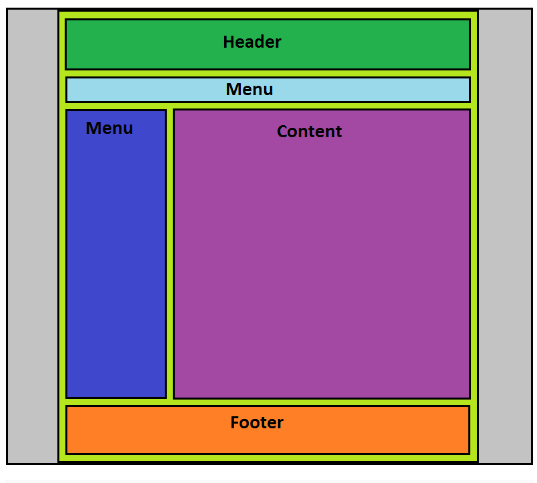
</html>

# Output



# HTML, CSS3 & Javascript\_2/300 Projects

Using Position Relative and Position Absolute



# Static

HTML elements are positioned static by default.

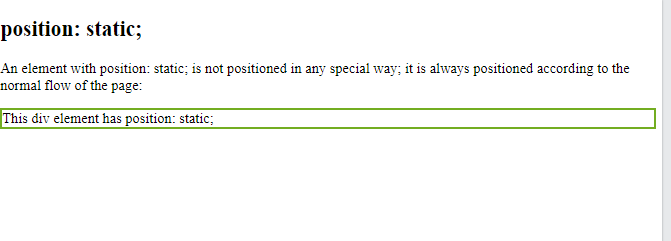
Static positioned elements are not affected by the top, bottom, left, and right properties.

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

# Static Example

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div.static {
6. position: static;
7. border: 3px solid #73AD21;
8. }
9. </style>
10. </head>
11. <body>
12. <h2>position: static;</h2>
13. <p>An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:</p>
14. <div class="static">
15. This div element has position: static;
16. </div>
17. </body>
18. </html>

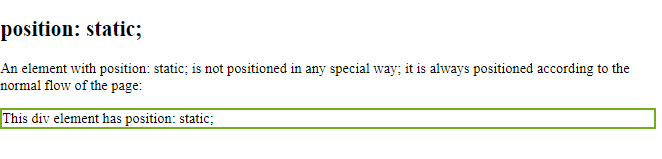
# Output



# More Examples (Static)

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div.static {
6. position: static;
7. border: 3px solid #73AD21;
8. }
9. h2{
10. position: static;
11. }
12. p{
13. position: static;
14. }
15. </style>
16. </head>
17. <body>
18. <h2>position: static;</h2>
19. <p>An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:</p>
20. <div class="static">
21. This div element has position: static;
22. </div>
23. </body>
24. </html>

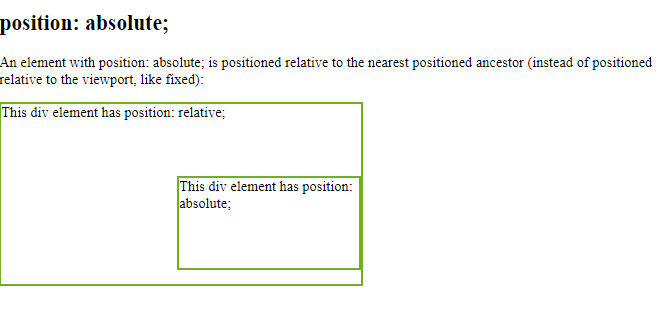
# Output



# position: absolute/relative

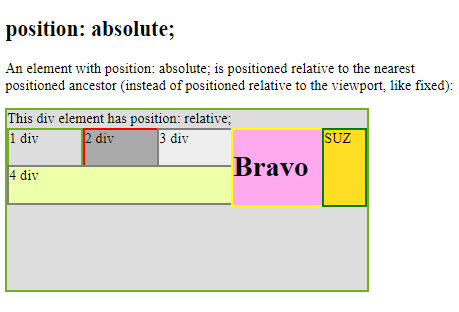
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div.relative {
6. position: relative;
7. width: 400px;
8. height: 200px;
9. border: 3px solid #73AD21;
10. }
11. div.absolute {
12. position: absolute;
13. top: 80px;
14. right: 0;
15. width: 200px;
16. height: 100px;
17. border: 3px solid #73AD21;
18. }
19. </style>
20. </head>
21. <body>
22. <h2>position: absolute;</h2>
23. <p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>
24. <div class="relative">This div element has position: relative;
25. <div class="absolute">This div element has position: absolute;</div>
26. </div>
27. </body>
28. </html>

# Output



1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div.relative {
6. position: relative;
7. width: 400px;
8. height: 200px;
9. border: 3px solid #73AD21;
10. background-color: #ddd;
11. }
12. div.absolute {
13. position: absolute;
14. top: 20px;
15. left: 0px;
16. width: 80px;
17. height: 40px;
18. border: 3px solid #73AD21;
19. }
20. div.absolute2 {
21. position: absolute;
22. top: 20px;
23. left: 84px;
24. width: 80px;
25. height: 40px;
26. border: 3px solid red;
27. background-color: #aaa;
28. }
29. div.absolute3 {
30. position: absolute;
31. top: 20px;
32. left: 167px;
33. width: 80px;
34. height: 40px;
35. border: 3px solid gray;
36. background-color: #eee;
37. }
38. div.absolute4 {
39. position: absolute;
40. top: 61px;
41. left: 0px;
42. width: 246px;
43. height: 40px;
44. border: 3px solid gray;
45. background-color: #efa;
46. }
47. div.absolute5 {
48. position: absolute;
49. top: 20px;
50. left: 249px;
51. width: 100px;
52. height: 83px;
53. border: 3px solid yellow;
54. background-color: #fae;
55. }
56. .anyName {
57. position: absolute;
58. top: 20px;
59. left: 350px;
60. width: 45px;
61. height: 83px;
62. border: 3px solid green;
63. background-color: #fd2;
64. }
65. </style>
66. </head>
67. <body>
68. <h2>position: absolute;</h2>
69. <p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>
70. <div class="relative">This div element has position: relative;
71. <div class="absolute">1 div </div>
72. <div class="absolute2">2 div </div>
73. <div class="absolute3">3 div </div>
74. <div class="absolute4">4 div </div>
75. <div class="absolute5"><h1>Bravo</h1> </div>
76. <div class="anyName">SUZ</div>
77. </div>
78. </body>
79. </html>

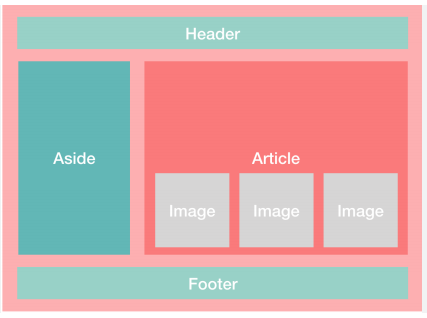
# Output



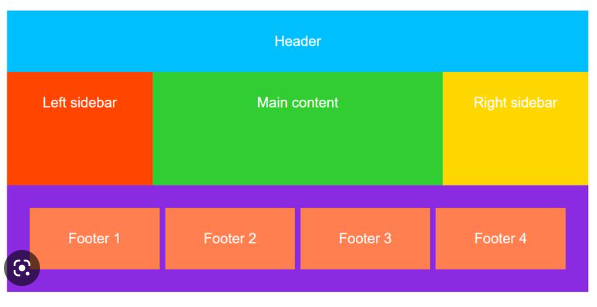
Project 2 (Assignment)



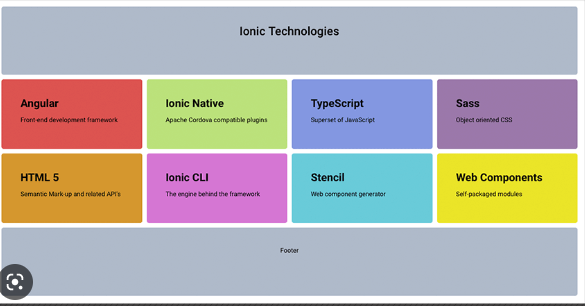
Project 3 (Assignment)



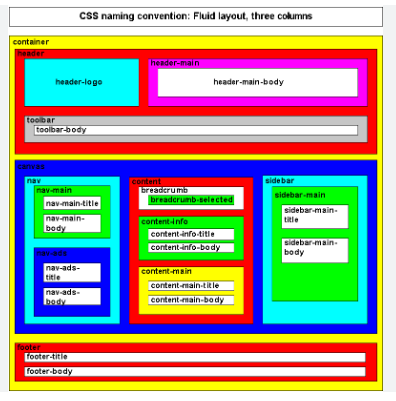
Project 4 (Assignment)



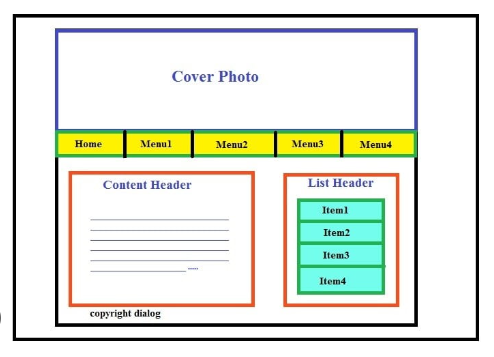
Project 5 (Assignment)



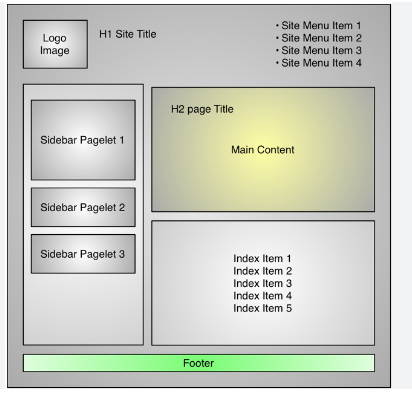
Project 6(Assignment)



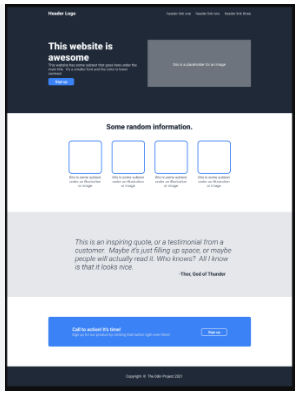
Project 7 (Assignment)



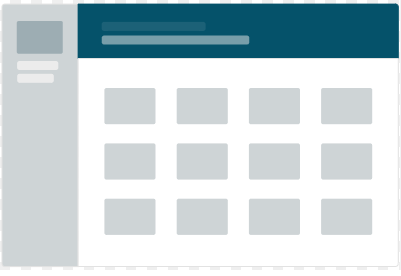
Project 8 (Assignment)



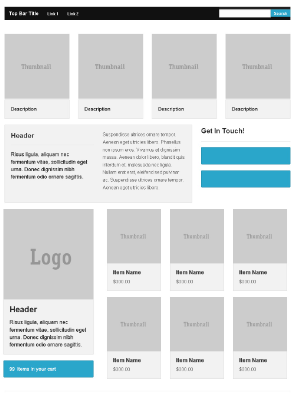
Project 9 (Assignment)



Project 10 (Assignment)



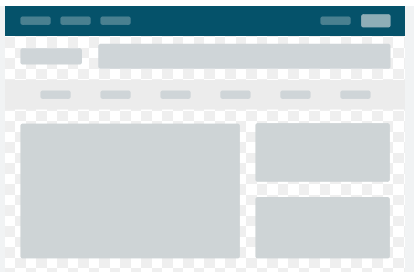
Project 11 (Assignment)



Project 12 (Assignment)



Project 13 (Assignment)

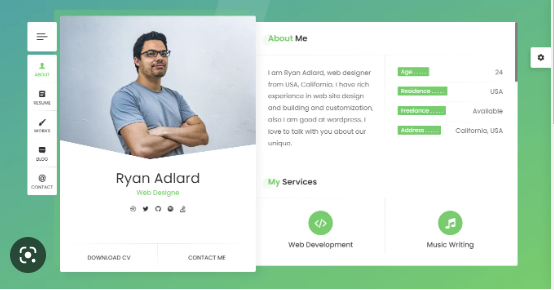


Project 14 (Assignment)

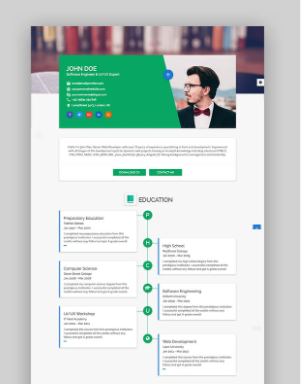


Project 15 (Practical in next Class 2)

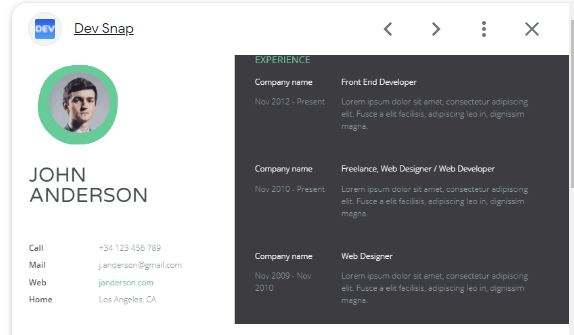
(Next Class 2)……



Project 16



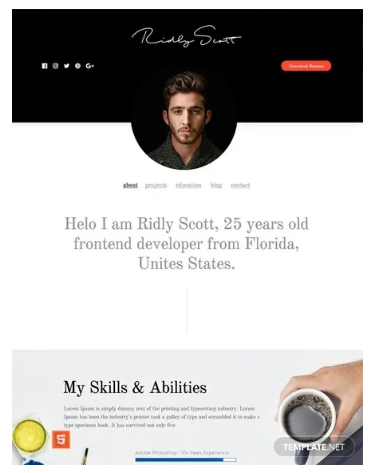
Project 17



Project 18



Project 19

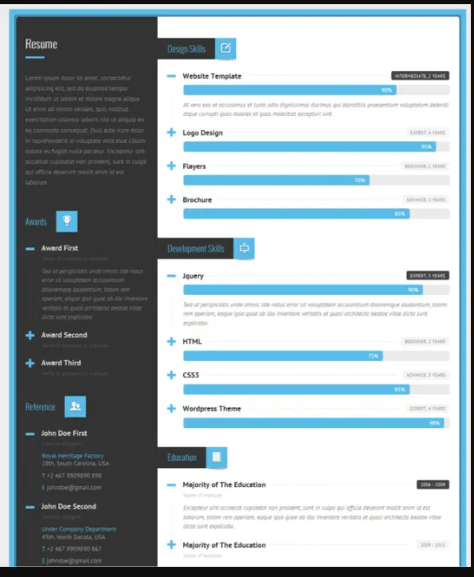


Project 20



Project 21

...



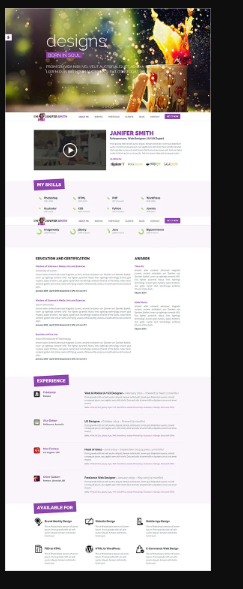
Project 22

...



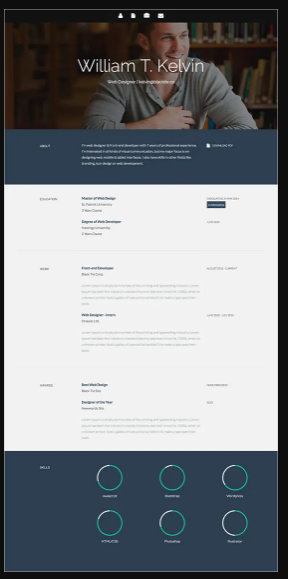
Project 23

...



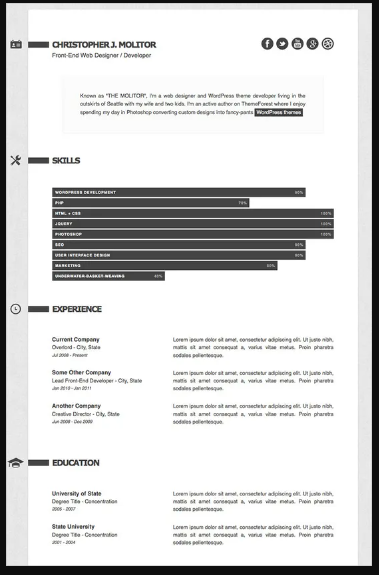
Project 24

...



Project 25

...



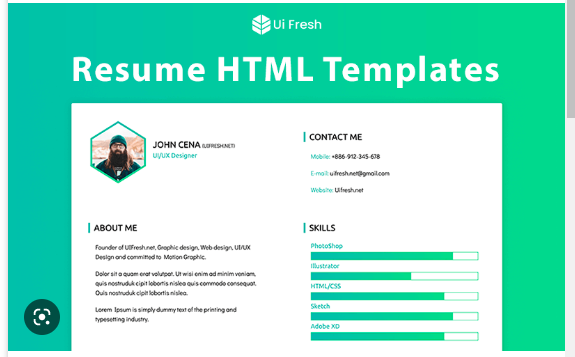
Project 26

...



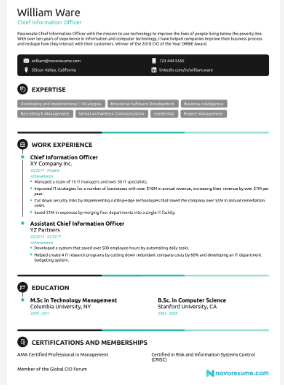
Project 27

...



Project 28

...



Project 29

...



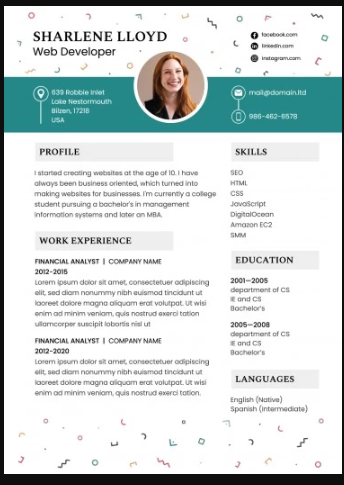
Project 30

...



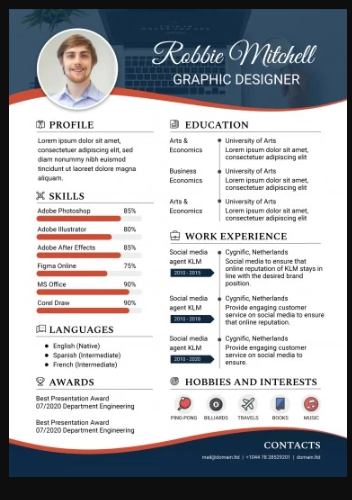
Project 31

...



Project 32

...



Project 33

...



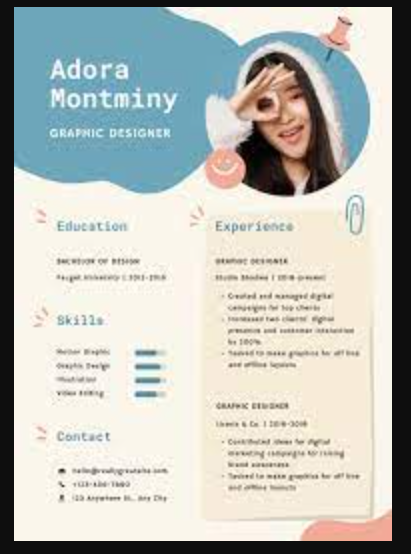
Project 34

...



Project 35

...



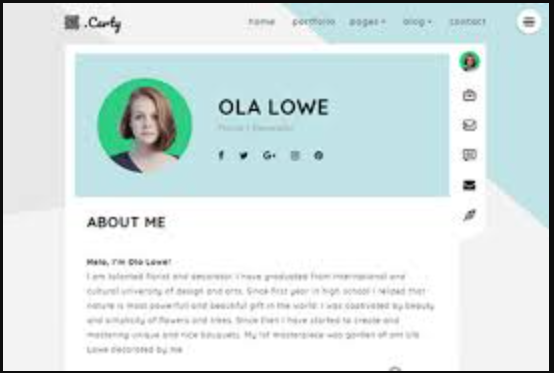
Project 36

...



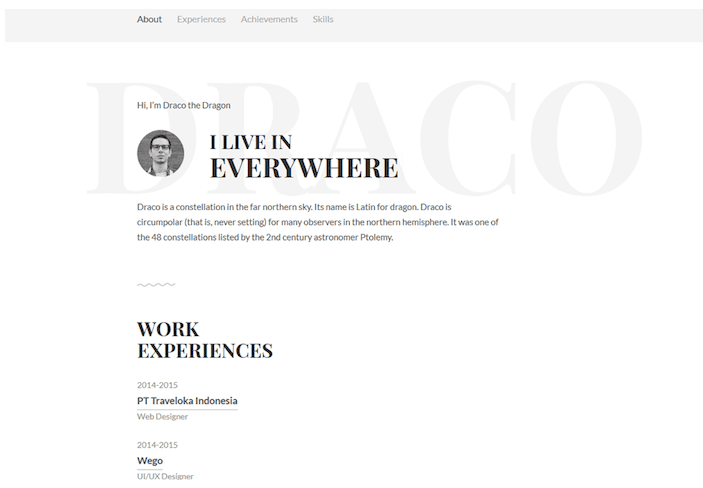
Project 37

...



Project 38

...



Project 39

...



Project 40

...

