

# CSS position Property

Here's a quick overview of CSS positioning:

## 1. Static

- **Default Positioning:** Elements are positioned according to the normal flow of the document.
- **No offset properties** (top, right, bottom, left) apply.

## 2. Relative

- **Positioned Relative to Itself:** The element is positioned relative to its normal position.
- **Offsets** (top, right, bottom, left) can be applied, shifting the element without affecting others.

## 3. Absolute

- **Positioned Relative to the Nearest Positioned Ancestor:** If no positioned ancestor exists, it uses the initial containing block (usually the viewport).
- **Removes the element from the normal document flow**, meaning it can overlap other elements.

## 4. Fixed

- **Positioned Relative to the Viewport:** The element stays in the same place even when the page is scrolled.
- Also removed from the normal flow, and offsets apply.

## 5. Sticky

- **Hybrid Positioning:** The element toggles between relative and fixed, depending on the scroll position.
- Initially positioned relative until a defined scroll point is reached, after which it behaves like fixed.

## Key Points

- **Z-index:** Controls the stacking order of positioned elements. Works with relative, absolute, fixed, and sticky.
- **Overflow:** Pay attention to the `overflow` property, especially with absolute and fixed elements.

## Css Examples:-

```
/* Static */
```

```
.static-element {  
    position: static; /* Default behavior */  
}
```

```
/* Relative */
```

```
.relative-element {  
    position: relative;  
    top: 10px; /* Moves down 10px from its original position */  
}
```

```
/* Absolute */
```

```
.absolute-element {  
    position: absolute;  
    top: 50px; /* 50px from the top of the nearest positioned ancestor */  
}
```

```
/* Fixed */
```

```
.fixed-element {  
    position: fixed;  
    bottom: 0; /* Sticks to the bottom of the viewport */  
}
```

```
}
```

```
/* Sticky */
```

```
.sticky-element {
```

```
  position: sticky;
```

```
  top: 0; /* Becomes fixed when scrolled to the top */
```

```
}
```

Here are detailed examples of each CSS positioning type, along with explanations of how they work and their typical use cases.

## 1. Static

- **Description:** The default positioning. Elements are placed according to the normal flow of the document.
- **Example:**

```
html
```

```
Copy code
```

```
<div class="container">
```

```
  <div class="static-element">Static Element</div>
```

```
  <div class="static-element">Another Static Element</div>
```

```
</div>
```

```
css
```

```
Copy code
```

```
.static-element {
```

```
  position: static; /* Default */
```

```
  background: lightblue;
```

```
  margin: 10px;
```

```
}
```

## 2. Relative

- **Description:** Positioned relative to its original position in the document flow. Offsets can shift its position without affecting surrounding elements.
- **Example:**

html code

```
<div class="container">
  <div class="relative-element">Relative Element</div>
  <div class="static-element">Another Static Element</div>
</div>
```

css code

```
.relative-element {
  position: relative;
  top: 20px; /* Moves down 20px from its original position */
  left: 10px; /* Moves right 10px from its original position */
  background: lightgreen;
}
```

### 3. Absolute

- **Description:** Positioned relative to the nearest positioned ancestor (an element with `position: relative, absolute, or fixed`). If there is none, it will be relative to the viewport. It removes the element from the normal flow.
- **Example:**

Html code

```
<div class="container" style="position: relative;">
  <div class="absolute-element">Absolute Element</div>
</div>
```

Css code

```
.absolute-element {
  position: absolute;
  top: 30px; /* 30px from the top of the nearest positioned ancestor */
  right: 20px; /* 20px from the right */
  background: lightcoral;
}
```

### 4. Fixed

- **Description:** Positioned relative to the viewport. It remains in the same place even when the page is scrolled. Also removes the element from the normal flow.
- **Example:**

htmlcode

```
<div class="fixed-element">Fixed Element</div>
```

css code

```
.fixed-element {
  position: fixed;
  bottom: 10px; /* Sticks 10px from the bottom of the viewport */
  right: 10px; /* Sticks 10px from the right of the viewport */
  background: lightgoldenrodyellow;
}
```

## 5. Sticky

- **Description:** A hybrid between relative and fixed positioning. The element is treated as relative until a specified scroll position is reached, after which it becomes fixed.
- **Example:**

htmlcode

```
<div class="sticky-container">
  <div class="sticky-element">Sticky Element</div>
  <p>Scroll down to see the sticky effect.</p>
  <div style="height: 1000px;"></div> <!-- Filler content -->
</div>
```

css code

```
.sticky-element {
  position: sticky;
  top: 0; /* Becomes fixed at the top of its container when scrolled
*/
  background: lightpink;
}
```

## Summary

- **Static:** Default flow, no offsets.
- **Relative:** Moves based on original position.
- **Absolute:** Positioned from nearest ancestor or viewport, overlaps other elements.
- **Fixed:** Stays in place on scroll, relative to viewport.
- **Sticky:** Behaves like relative until a scroll position is reached, then behaves like fixed.

The CSS `position` property is used to define the position of an element on a webpage.

The location of the positioned element is set with the four properties: `top`, `left`, `right`, and `bottom`. These properties only work when the `position` property is set and have different positioning behaviors.

The `position` property has the following five values:

- static (default value)
- relative
- absolute
- fixed
- sticky

We will look into each of them.

## CSS static Position

The `static` value of the `position` property allows elements to be positioned accordingly to the normal flow of the document. The `static` position is not affected by the `top`, `right`, `bottom`, and `left` values.

Let's see an example,

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

  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
    <title>CSS position</title>
  </head>

  <style>
    p {
      border: 1px solid black;
      padding: 4px;
    }

    p.main {
      position: static;
      top: 50px; /* doesn't work */
      right: 50px; /* doesn't work */
      bottom: 50px; /* doesn't work */
      left: 50%; /* doesn't work */
      background-color: greenyellow;
    }
  </style>

  <body>
    <p>This is a normal paragraph.</p>
    <p class="main">This paragraph is positioned using the static value.</p>
    <p>This is a normal paragraph.</p>
  </body>
</html>
```

```
</html>
```

OutPut:-

This is a normal paragraph.

This paragraph is positioned using the static value.

This is a normal paragraph.

## CSS Relative Position

The `relative` value positions the element relative to the original position in the document. The element is positioned with the `top`, `right`, `bottom`, and `left` values.

Let's see an example,

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<link rel="stylesheet" href="style.css" />
```

```
<title>CSS position</title>
```

```
<style>
```

```
p {
```

```
border: 1px solid black;
padding: 4px;
}
```

```
p.main {
  position: relative;
  /* positions 90px from the top */
  top: 90px;

  /* positions 40px from the left */
  left: 40px;
  background-color: greenyellow;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is a normal paragraph.</p>
```

```
<p class="main">
```

```
  This paragraph is positioned with a relative value of top 50px and
```

```
  left 40px.
```

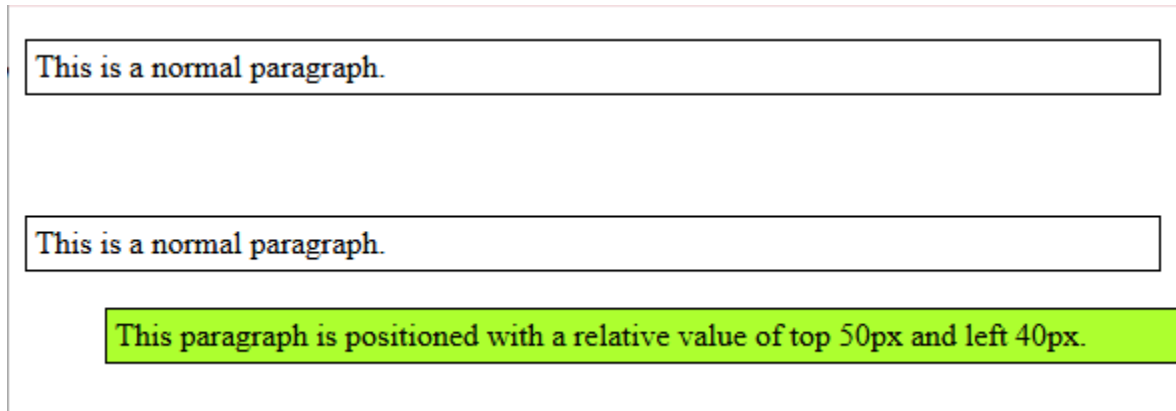
```
</p>
```

```
<p>This is a normal paragraph.</p>
```

```
</body>
```

</html>

**Output:-**



## CSS Absolute Position

The `absolute` value removes the element completely from its normal flow in the document.

The element is positioned relative to their closest positioned parent element (an ancestor element with a position value other than `static`).

If there is no positioned ancestor, they are positioned relative to the document itself.

Let's see an example,

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<link rel="stylesheet" href="style.css" />

<title>CSS position</title>

    <style>

        p {

border: 1px solid black;

padding: 4px;

}

p.main {

    position: absolute;

    top: 70px;

    left: 60px;

    background-color: greenyellow;

}

        </style>

</head>

<body>

    <p>This is a normal paragraph.</p>

    <p class="main">

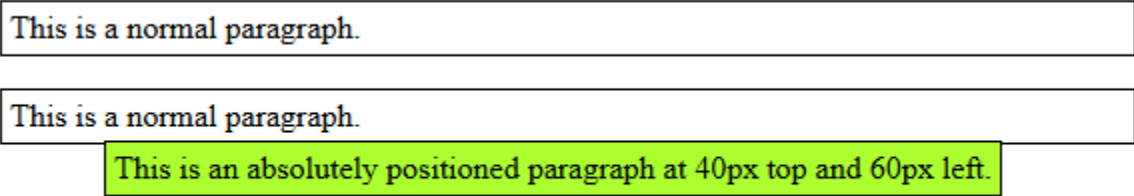
        This is an absolutely positioned paragraph at 40px top and 60px left.

    </p>
```

```
<p>This is a normal paragraph.</p>
```

```
</body>
```

```
</html>
```



This is a normal paragraph.

This is a normal paragraph.

This is an absolutely positioned paragraph at 40px top and 60px left.

In the above example, the second paragraph doesn't have a positioned parent element, so it is positioned relative to the viewport (visible area of the browser).

Let's see an example to understand how the `absolute` position works with positioned parent element:

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<link rel="stylesheet" href="style.css" />
```

```
<title>CSS position</title>
```

```
        <style>
        p {
border: 1px solid black;
padding: 4px;
}

p.main {
position: absolute;
top: 70px;
left: 60px;
background-color: greenyellow;
}

div.parent {
position: relative;
background-color: orange;
border: 4px solid red;
}

        </style>
```

```
</head>
```

```
<body>
```

```
<p>This is a normal paragraph.</p>
```

```
<p class="main">
```

```
    This is an absolutely positioned paragraph at 40px top and 60px
```

left.

</p>

<p>This is a normal paragraph.</p>

<!-- Creating an container -->

<div class="parent">

<p>This paragraph is inside the div element.</p>

<p class="main">

This is an absolutely positioned paragraph inside the div  
element at 40px top and 60px left.

</p>

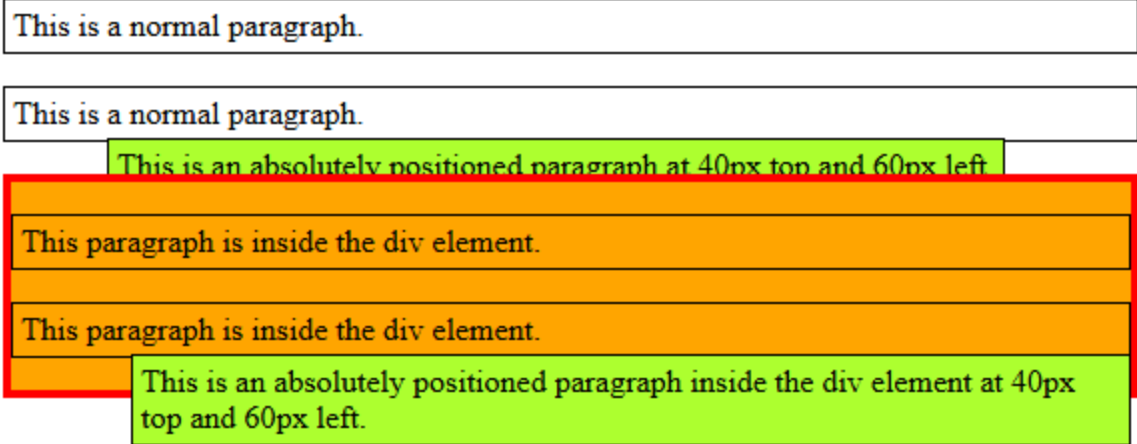
<p>This paragraph is inside the div element.</p>

</div>

</body>

</html>

Output:-



## CSS fixed Position

The `fixed` value positions an element to remain fixed in the same position, even when the page is scrolled. It is similar to the `absolute` value, but it remains relative to the viewport at all times.

Let's see an example,

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

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet" href="style.css" />
  <title>CSS position</title>
  <style>
    p {
```

```
border: 2px solid black;
padding: 4px;
}
```

```
p.main {
  position: fixed;
  top: 10px;
  background-color: greenyellow;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="main">
```

```
  This paragraph has a fixed position value at 10px top.
```

```
</p>
```

```
<p>This is a normal paragraph.</p>
```

```
<p>This is a normal paragraph.</p>
```

```
<p>This is a normal paragraph.</p>
```

```
<p>This is a normal paragraph.</p>
```

```
<p>This is a normal paragraph.</p>
```

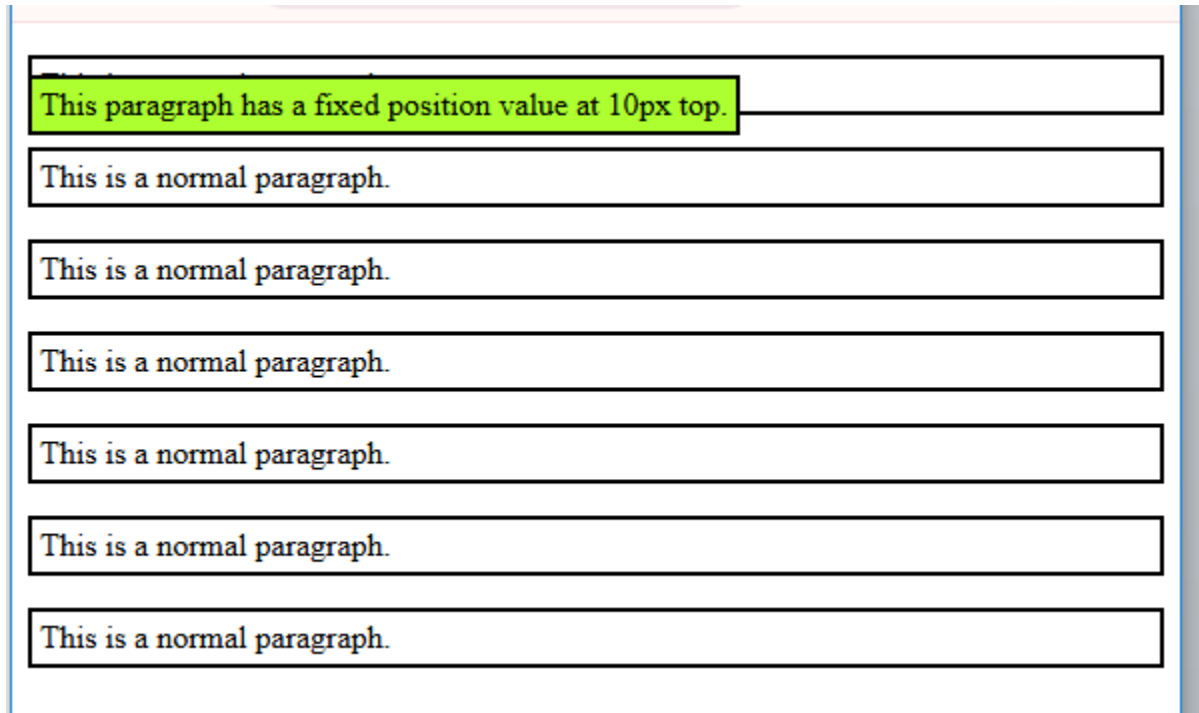
```
<p>This is a normal paragraph.</p>
```

```
<p>This is a normal paragraph.</p>
```

```
</body>
```

</html>

Output:-

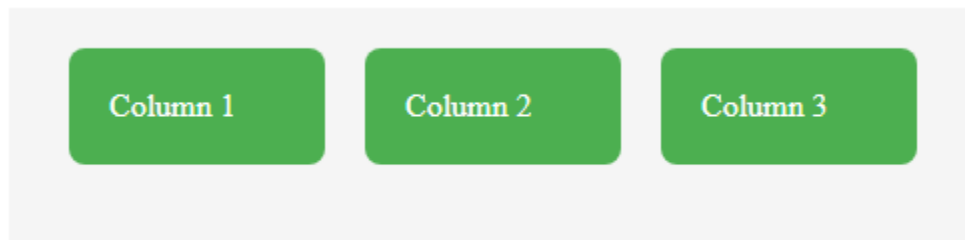


In the above example,

```
position: fixed;  
top: 10px;
```

fixes the paragraph at the `10px` distance from the `top` in the webpage. The paragraph doesn't scroll with the other content in the document.

## Css Relative Examples :-



Code example:-

```
<html>
<head>
<style>
* {
  box-sizing: border-box; /* Ensure padding doesn't affect width */
}

.container {
  display: flex; /* Use flexbox for layout */
  justify-content: space-between; /* Space out columns */
  padding: 20px;
  background-color: #f5f5f5;
}

.column {
  position: relative; /* Relative positioning */
  flex: 1; /* Equal width columns */
}
```

```
margin: 10px; /* Space between columns */
padding: 20px;
background-color: #4caf50; /* Green background */
color: white;
border-radius: 8px; /* Rounded corners */
}

/* Optional: Add a slight upward shift to one of the columns */
.column1 {
  top: -10px; /* Shifts Column 1 slightly up */
}
.column2 {
  top: -10px; /* Shifts Column 2 slightly up */
}
.column3 {
  top: -10px; /* Shifts Column 2 slightly up */
}

</style>
</head>
<body>
<div class="container">
  <div class="column column1">Column 1</div>
  <div class="column column2">Column 2</div>
  <div class="column column3">Column 3</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Css Absolute example :-



```
<html>
```

```
<head>
```

```
<style>
```

```
* {
```

```
  box-sizing: border-box; /* Ensure padding doesn't affect width */
```

```
}
```

```
.container {
```

```
  position: relative; /* Establishes a containing block for absolutely positioned elements */
```

```
  height: 200px; /* Fixed height for demonstration */
```

```
  padding: 20px;
```

```
  background-color: #f5f5f5;
```

```
}
```

```
.column {
```

```
    position: absolute; /* Absolutely position the columns */
```

```
    width: calc(33.33% - 20px); /* Adjust width to account for margins */
```

```
    padding: 20px;
```

```
    background-color: #4caf50; /* Green background */
```

```
    color: white;
```

```
    border-radius: 8px; /* Rounded corners */
```

```
}
```

```
/* Position each column */
```

```
.column1 {
```

```
    top: 20px; /* Positioning from the top */
```

```
    left: 0; /* Aligns to the left */
```

```
}
```

```
.column2 {
```

```
    top: 20px; /* Same vertical alignment */
```

```
    left: calc(33.33% + 10px); /* Position after Column 1 */
```

```
}
```

```
.column3 {
```

```
    top: 20px; /* Same vertical alignment */
```

```
    left: calc(66.66% + 20px); /* Position after Column 2 */
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
  <div class="column column1">Column 1</div>
```

```
  <div class="column column2">Column 2</div>
```

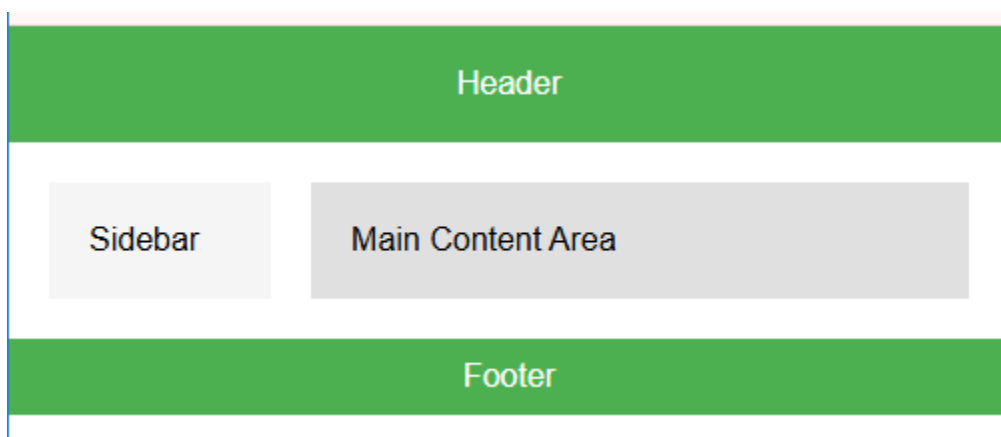
```
  <div class="column column3">Column 3</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Css relative home page layout :-



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Home Page Layout</title>
</style>
* {
  box-sizing: border-box; /* Ensures padding doesn't affect width */
}

body {
  margin: 0; /* Remove default margin */
  font-family: Arial, sans-serif; /* Font styling */
}

.header {
  background-color: #4caf50; /* Green background */
  color: white;
  text-align: center;
  padding: 20px;
}

.container {
  display: flex; /* Use flexbox for layout */
```

```
padding: 20px;  
}
```

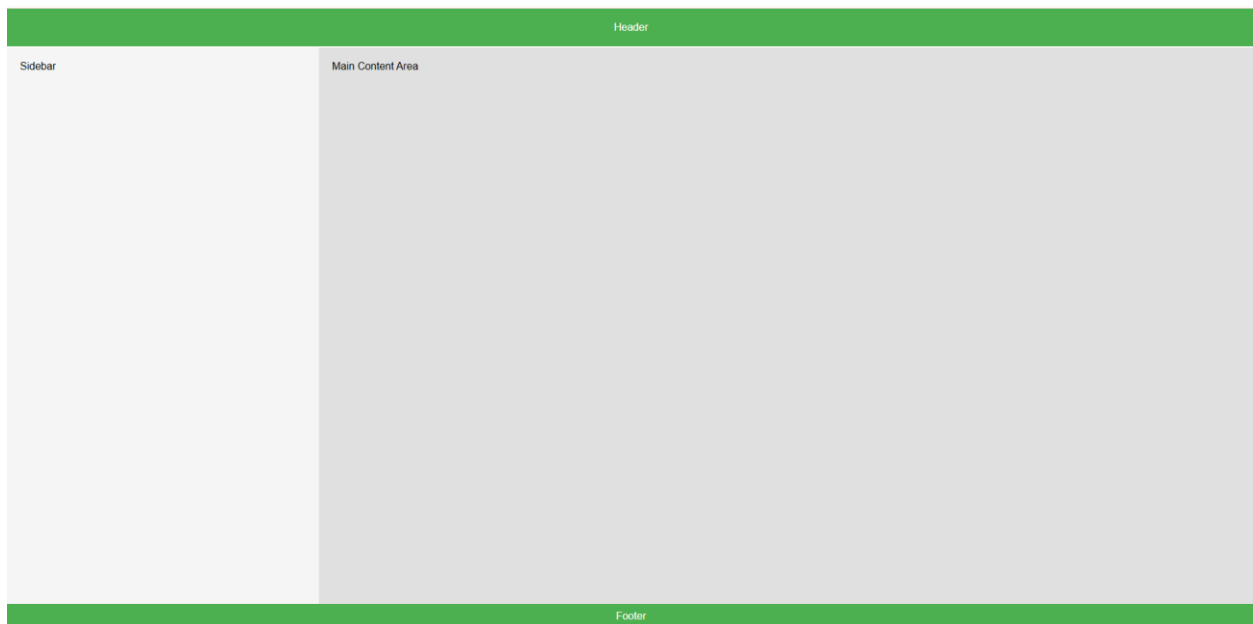
```
.sidebar {  
    position: relative; /* Set relative positioning */  
    width: 25%; /* Sidebar takes 25% of the container */  
    background-color: #f5f5f5; /* Light background for sidebar */  
    padding: 20px;  
    margin-right: 20px; /* Space between sidebar and main content */  
}
```

```
.main-content {  
    position: relative; /* Set relative positioning */  
    width: 75%; /* Main content takes 75% of the container */  
    background-color: #e0e0e0; /* Light grey background for main content */  
    padding: 20px;  
}
```

```
.footer {  
    background-color: #4caf50; /* Green background */  
    color: white;  
    text-align: center;  
    padding: 10px;  
    position: relative; /* Footer relative to its own context */  
}
```

```
</style>
</head>
<body>
  <div class="header">Header</div>
  <div class="container">
    <div class="sidebar">Sidebar</div>
    <div class="main-content">Main Content Area</div>
  </div>
  <div class="footer">Footer</div>
</body>
</html>
```

## Css absolute layout page :-



```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Home Page Layout</title>

<style>

* {

  box-sizing: border-box; /* Ensures padding doesn't affect width */

}

body {

  margin: 0; /* Remove default margin */

  font-family: Arial, sans-serif; /* Font styling */

}

.container {

  position: relative; /* Establishes a containing block for absolutely positioned elements */

  min-height: 100vh; /* Minimum height for the container */

  padding: 0;

}

.header {

  position: absolute; /* Absolutely positioned */
```

```
top: 0; /* Sticks to the top */  
left: 0;  
right: 0;  
background-color: #4caf50; /* Green background */  
color: white;  
text-align: center;  
padding: 20px;  
}
```

```
.sidebar {  
    position: absolute; /* Absolutely positioned */  
    top: 60px; /* Below the header */  
    left: 0;  
    width: 25%; /* Sidebar takes 25% of the container */  
    background-color: #f5f5f5; /* Light background for sidebar */  
    padding: 20px;  
    height: calc(100vh - 60px); /* Full height minus header */  
}
```

```
.main-content {  
    position: absolute; /* Absolutely positioned */  
    top: 60px; /* Below the header */  
    left: 25%; /* Next to the sidebar */  
    right: 0; /* Extends to the right */  
    background-color: #e0e0e0; /* Light grey background for main content */  
}
```

```
padding: 20px;
height: calc(100vh - 60px); /* Full height minus header */
}
```

```
.footer {
    position: absolute; /* Absolutely positioned */
    bottom: 0; /* Sticks to the bottom */
    left: 0;
    right: 0;
    background-color: #4caf50; /* Green background */
    color: white;
    text-align: center;
    padding: 10px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div class="header">Header</div>
```

```
<div class="sidebar">Sidebar</div>
```

```
<div class="main-content">Main Content Area</div>
```

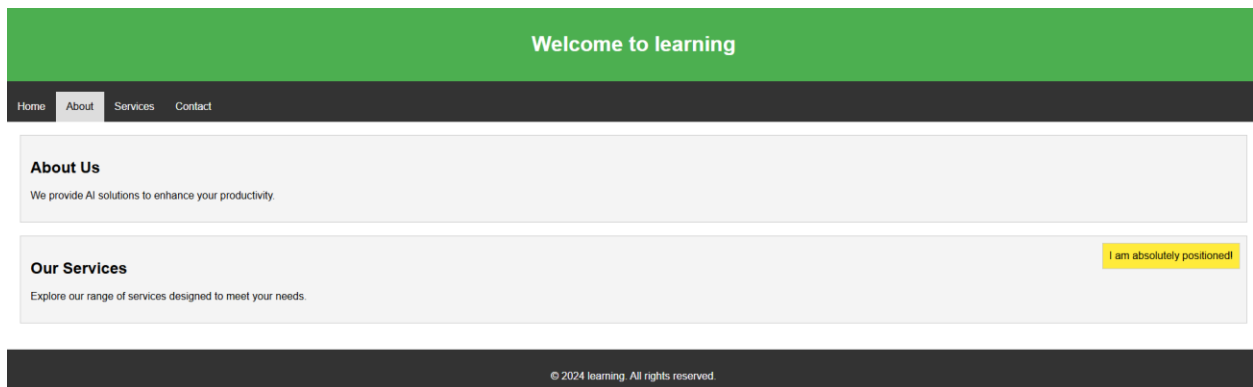
```
<div class="footer">Footer</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

## Css Sticky examples :-



```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>learning Home Page</title>
```

```
  <style>
```

```
    body {
```

```
      margin: 0;
```

```
      font-family: Arial, sans-serif;
```

```
  }
```

```
.header {  
    background-color: #4CAF50;  
    color: white;  
    text-align: center;  
    padding: 1rem;  
}
```

```
.navbar {  
    background-color: #333;  
    overflow: hidden;  
    position: sticky; /* Sticky positioning for navbar */  
    top: 0;  
}
```

```
.navbar ul {  
    list-style-type: none;  
    padding: 0;  
}
```

```
.navbar li {  
    float: left;  
}
```

```
.navbar a {
```

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

```
.navbar a:hover {
background-color: #ddd;
color: black;
}
```

```
.main-content {
padding: 20px;
}
```

```
.section {
margin-bottom: 20px;
padding: 15px;
background-color: #f4f4f4;
border: 1px solid #ccc;
position: relative; /* Relative positioning for sections */
}
```

```
.absolute-box {
```

```
position: absolute; /* Absolutely positioned box */
top: 10px;
right: 10px;
background-color: #ffeb3b;
padding: 10px;
border: 1px solid #ccc;
}

.footer {
background-color: #333;
color: white;
text-align: center;
padding: 1rem;
position: relative; /* Footer with relative positioning */
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<header class="header">
```

```
<h1>Welcome to learning</h1>
```

```
</header>
```

```
<nav class="navbar">
```

```
<ul>
```

```
<li><a href="#">Home</a></li>
<li><a href="#">About</a></li>
<li><a href="#">Services</a></li>
<li><a href="#">Contact</a></li>
```

```
</ul>
```

```
</nav>
```

```
<main class="main-content">
```

```
<section class="section">
```

```
<h2>About Us</h2>
```

```
<p>We provide AI solutions to enhance your productivity.</p>
```

```
</section>
```

```
<section class="section">
```

```
<h2>Our Services</h2>
```

```
<p>Explore our range of services designed to meet your needs.</p>
```

```
<div class="absolute-box">I am absolutely positioned!</div>
```

```
</section>
```

```
</main>
```

```
<footer class="footer">
```

```
<p>© 2024 learning. All rights reserved.</p>
```

</footer>

</body>

</html>