

## CSS Tutorial: Position absolute, relative, fixed and sticky in CSS | Web Developm...



Course: Web Dev Tutorials For Beginners (+)

[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

## CSS Tutorial: Position absolute, relative, fixed and sticky in CSS | Web Development Tutorials #25

This tutorial begins where the #24 tutorial left off. In this tutorial, we will discuss the position property of CSS. With the help of CSS, you can put your HTML elements at the position of your choice. After going through this tutorial, you will master the differences in the positioning of the elements. So, without wasting any time, let's start discussing the CSS positioning-related properties.

### Types Of Position Property :

There are five types of position property :

- static
- relative
- absolute
- fixed

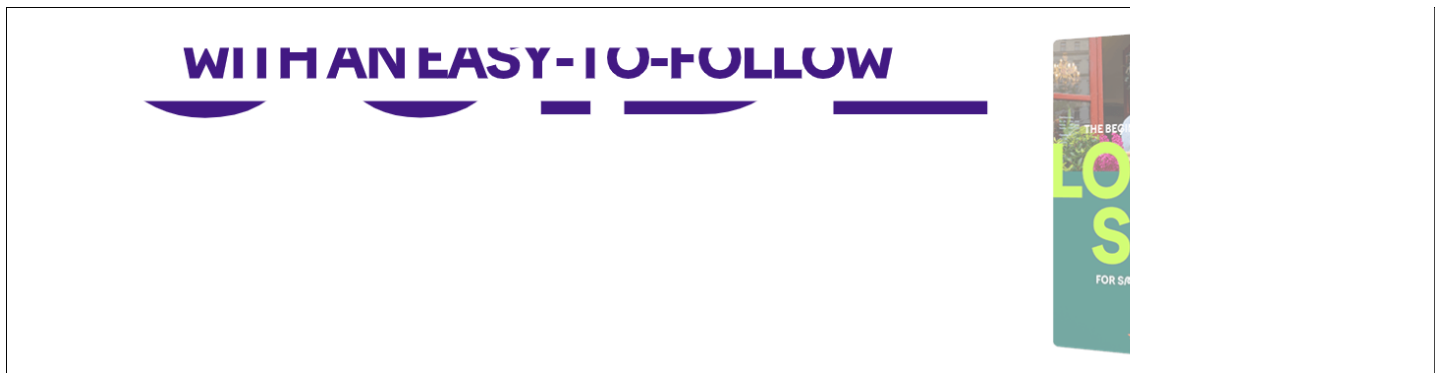
- sticky

## position: static;

- It is the default position of HTML elements.

## position: relative;

- It is used when we need to position an HTML element relative to its normal position.
- We can set the top, right, bottom, and left properties that will cause the element to adjust away from the normal position.



Example: We have used the below CSS to design four boxes as shown in the given image :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Position Tutorial</title>
  <style>
    .box{
      border: 2px solid red;
      display: inline-block;
      width: 150px;
      height: 150px;
      margin: 2px;
    }
  </style>
```

```
</head>
<body>
  <div class="box" id="box1"></div>
  <div class="box" id="box2"></div>
  <div class="box" id="box3"></div>
  <div class="box" id="box4"></div>
</body>

</html>
```

## Output:

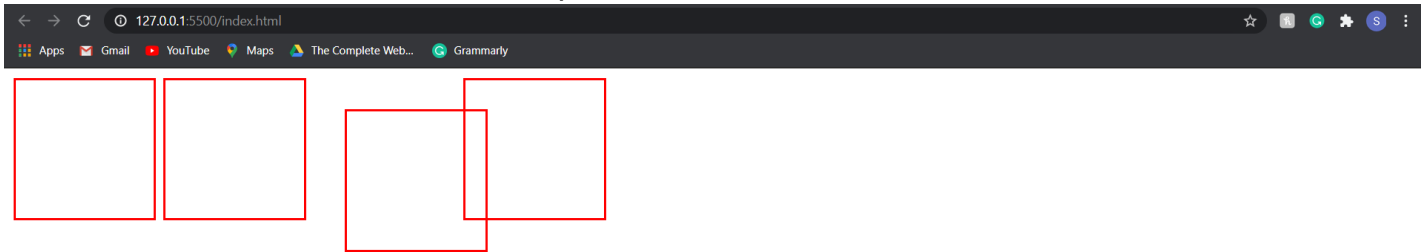


The default position of all the boxes in the above image is static. Now, we will change the position from static to relative of box 3. Here is the CSS used :

```
<style>
  .box {
    border: 2px solid red;
    display: inline-block;
    width: 150px;
    height: 150px;
    margin: 2px;
  }
  #box3 {
    position: relative;
    top: 34px;
  }
```

```
left: 34px;  
}  
</style>
```

You can see in the image given below that box3 has shifted 34px away from the top and left side relative to its normal position.



## position: absolute;

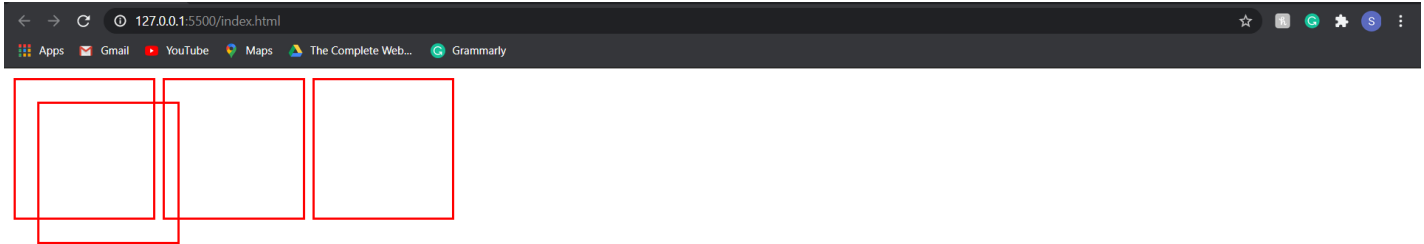
- An element with the absolute position will move according to the position of its parent element.

- In the absence of any parent element, the HTML element will be placed relative to the page.

Now, we have changed the position of box3 from relative to absolute. Here is the CSS used :

```
<style>
    .box {
        border: 2px solid red;
        display: inline-block;
        width: 150px;
        height: 150px;
        margin: 2px;
    }
    #box3 {
        position: absolute;
        top: 34px;
        left: 34px;
    }
    .container{
        border: 2px solid black;
        background-color: khaki;
        height: 3444px;
    }
</style>
```

You can see in the image given below that the box3 has moved to the left side of the page.



---

## position: fixed;

- An element with `position:fixed;` will remain stuck to a specific position even after the page is scrolled.
- This position property is used when we want to keep an HTML element at a fixed spot no matter where on the page the user is.

Notice the box fixed at the top right corner of the page in the image given below. Here is the CSS used :

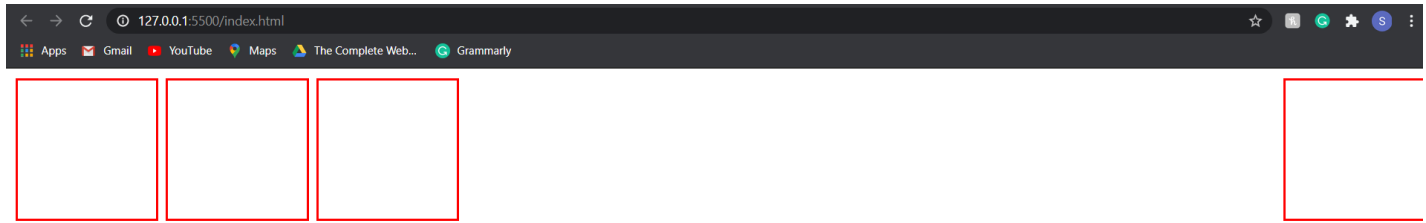
```
<!DOCTYPE html>

<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Fixed Position In CSS</title>
  <style>
    .box{
      border: 2px solid red;
      display: inline-block;
      width: 150px;
```

```
        height: 150px;
        margin: 2px;
    }
    #box3{
        position: fixed;
        right: 4px;
    }
</style>
</head>
<body>
    <div class="box" id="box1"></div>
    <div class="box" id="box2"></div>
    <div class="box" id="box3"></div>
    <div class="box" id="box4"></div>
</body>
</html>
```

## Output:



## position: sticky;

- It is a hybrid of relative and fixed position.
- An HTML element with `position: sticky;` will just sit there until a given position offset is met.

Use the CSS given below to get a better understanding of the sticky element.

```
#box3 {  
    position: sticky;  
    top: 3px;  
}
```

We have learnt various properties of CSS, and in the next tutorial, we will design a gym website using the HTML and CSS we have learnt so far. This tutorial ends here, and I will see you in the next tutorial.

## Code as described/written in the video

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



```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
<style>
    .container{
        border: 2px solid black;
        background-color: khaki;
        height: 3444px;
    }
    /* CSS Position: static (default), relative, absolute, fixed, sticky */
    .box{
        display: inline-block;
        border: 2px solid red;
        width: 150px;
        height: 150px;
        margin: 2px;
    }
    #box3{
        /* relative: Positions the element relative to its normal position and width
        /* position: relative; */

        /* absolute: Positions the element relative to the position of its first ancestor
        /* position: absolute; */

        /* top: 34px;
        left: 134px; */

        /* fixed: Positions the element relative to the browser window; */
        /* position: fixed;
        right: 4px;
        bottom: 2px */

        /* sticky: Positions the element relative to the user's scroll position
        position: sticky;
        top: 3px;

    }
</style>
</head>
```

```
<body>
  <div class="container">
    <div class="box" id="box1">1</div>
    <div class="box" id="box2">2</div>
    <div class="box" id="box3">Chat with us</div>
    <div class="box" id="box4">4</div>
  </div>
</body>
</html>
```

[Previous](#)[Next](#)**CodeWithHarry**

Copyright © 2022 CodeWithHarry.com

