CSS – Positioning

CSS positioning properties are used how to position the HTML elements on the webpage. The position property is used for this and it can have following value:

- static
- relative
- absolute
- sticky
- fixed

Now left, right, top, bottom properties are used to set the position of element on the webpage after setting the value of position (except for static).

**position: static**

The element will be positioned according to the normal flow of the webpage. This is default value of position; it means that if the position value for any element is not set, then it will be static. The left, right, top, bottom properties have no effect in this position.

```css
div{
  position: static;
  height:100px;
  width:200px;
}
```

**Note:** Above both declarations of CSS properties for div tag are same. Position is set to static by default.
**Example - Static**

```html
<html>
<head>
    <style>
    #box1{
        background-color: beige;
        height:100px;
    }
    #box2{
        background-color: cyan;
        height:200px;
        position: static;
    }
</style>
</head>
<body>
<h1> Static positioned elements are positioned according to the normal flow of the web page. </h1>
<div id="box1"> I am division 1, I am static positioned.</div>
<div id="box2">I am division 2, I am also static positioned.</div>
</body>
</html>
```

**Output - Static**

![Static positioning example](static-positioning-example.png)

---

**position: relative**

The element will be positioned relative to its normal position in the webpage. The left, right, top, bottom properties are used then to set it location on the webpage. The gap (space) created due to this adjustment (setting value of left, right, top, bottom) will not be fit or fill by any other element.

```html
div{
    position: relative;
    height:100px;
    width:200px;
    top:30px;
    left:50px;
}
```
**Example - Relative**

```html
<html>
<head>
  <style>
    #box1{
      background-color:beige;
      position: relative;
      left:40px;
      top:50px;
      width:200px;
      height:200px
    }
    #box2{
      background-color:azure;
      position: relative;
      left:250px;
      bottom:150px;
      width:200px;
      height:200px;
    }
  </style>
</head>
<body>
  <h1>Relative positioned elements are positioned relative to its normal position.</h1>
  <div id="box1">I am division 1, I am relatively positioned (i.e. 40px left and 50px top from my normal position.)</div>
  <div id="box2">I am division 2, I am relatively positioned (i.e. 250px left and 150px bottom from my normal position.)</div>
</body>
</html>
```

**Output - Relative**

![Relative positioned elements are positioned relative to its normal position.](relative-position.html)

**position: absolute**

The element will be positioned relative to its closest positioned ancestor (parent). The left, right, top, bottom properties are used to set the location of the element relative to the closest positioned ancestor. No space (gap) is created due to this adjustment (setting value of left, right, top, and bottom).
Example - Absolute

```html
<html>
<head>
<style>
#box1 {
  background-color: beige;
  position: relative;
  left: 40px;
  top: 20px;
  width: 500px;
  height: 300px
}
img {
  position: absolute;
  height: 200px;
  width: 200px;
  right: 10px;
  bottom: 10px;
}
</style>
</head>
<body>
<h1>Absolute positioned elements are positioned relative to its closest positioned ancestor.</h1>
<div id="box1">I am division 1, I am relatively positioned <br>(i.e. 40px left and 50px top from my normal position.<br> This image is positioned absolute (i.e. 10px from bottom, 10px from right relative to my (div) position because I am closest ancestor of the image).<img src="quote.jpg" />
</div>
</body>
</html>
```

Output - Absolute

Absolute positioned elements are positioned relative to its closest positioned ancestor.
position: fixed

The element will be positioned relative to viewport of the screen. The left, right, top, bottom properties are used to set the location of the element according to viewport of the screen. No space (gap) is created due to this adjustment (setting value of left, right, top, and bottom). Even if the page scrolls, fixed element always stays at same place.

```html
<html>
  <head>
    <style>
      div{
        background-color: aqua;
        position: fixed;
        height:60px;
        width:400px;
        right:10px;
        bottom:10px;
      }
    </style>
  </head>
  <body>
    <br><br>
    <h1> Fixed positioned elements are positioned relative to viewport of the screen.</h1>
    <br> 
    See at bottom right corner of the screen. The fixed positioned div adjusted according to viewport of the screen (i.e. 10px right and 10px bottom from viewport of the screen.
  </body>
</html>
```
position: sticky

This position value works like as a mixture of relative and fixed. The element will be positioned relative to viewport of the screen. It behaves like relative position before the page scrolls and then sticks on the page based on the value of top, left, right and bottom.

```html
<html>
<head>
<style>
div{
    background-color:aqua;
    position:sticky;
    height:60px;
    width:400px;
    left:0px;
    top:0px;
}
</style>
</head>
<body>
<h1>Stay Home. Stay Safe.</h1>
</body>
</html>
```

**Example - Sticky**

Firstly, it behaves like relative position,
<table>
<thead>
<tr>
<th>Exercise:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. What are CSS positioning properties?</td>
</tr>
<tr>
<td>2. Explain about sticky position with an example.</td>
</tr>
</tbody>
</table>

```html
When page scrolls it behaves fixed position:
```