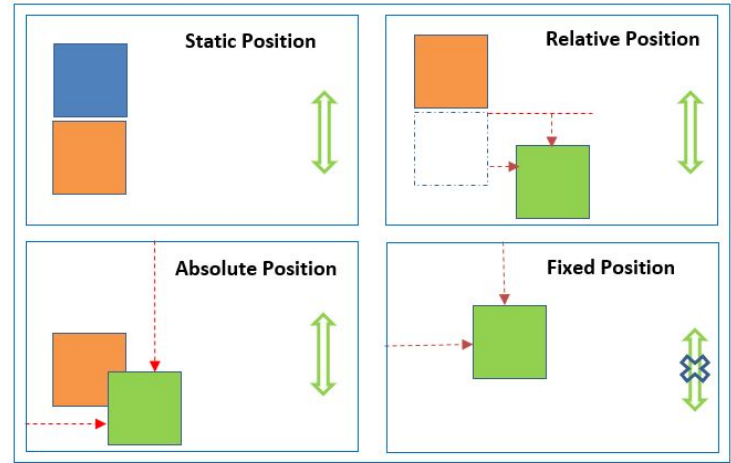


CSS Positioning

By Nitesh Gupta



- Block level elements create a full width of their parent elements, and they prevent other elements from appearing in the same horizontal line.
- Block level elements take up their own line of space and do not overlap with each other.
- The default position of the block level elements is to appear on the left side of the browser.



Blue box

Green box

**Block level
elements**

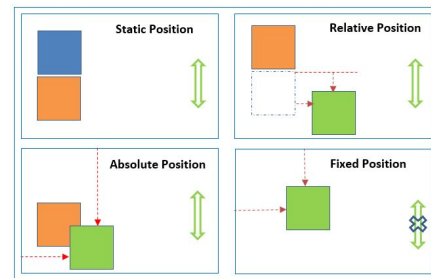
CSS
Position

position: values;

z-index

CSS Position Property

- The CSS **position** property is used to set position for an element.
- The CSS **position** property is also used to place an element behind another and also useful for scripted animation effects.
- The CSS **position** property can take following possible values:
 - static.
 - relative.
 - absolute.
 - fixed.
 - sticky.



Block level
elements

**CSS
Position**

position: values;

z-index

- The default value of the CSS **position** property is **static**.
- HTML elements are positioned **static** by default.
- An element with **position: static;** is not positioned in any special way.
- It is not affected by top, right, left, bottom properties.



- The relative position property is used to set the element relative to its normal position.
- Example:

```
.green-box {  
  background-color: green;  
  position: relative;  
}
```
- The code in the above example instructs the browser to place the .green-box element in relative position.
- But it does not specify where the .green-box element should be positioned. This can be done by accompanying the position declaration with any one of the following offset properties.



- Offset properties:
 - top - moves the element down from the top.
 - bottom - moves the element up from the bottom.
 - left - moves the element away from the left to right side.
 - right - moves the element away from the right to left side.
- The values of the offset properties can be in pixels, ems, percentages,...

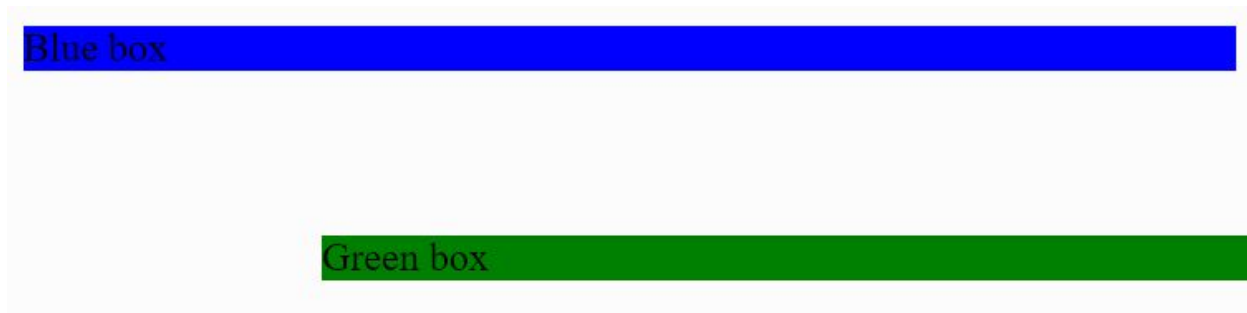
- Example:

```
.green-box {  
  background-color: green;  
  position: relative;  
  top: 50px;  
  left: 120px;  
}
```



Position: Relative;

Before and After applying offset properties:



- When an element's position is set to absolute, all other elements on the page will ignore the element and act like it is not present on the page.
- The element will be positioned relative to its closest positioned parent element, while offset properties can be used to determine the final position from there.
- Example:

```
header {  
  background-color: #466995;  
  border-bottom: 1px solid #466995;  
  position: absolute;  
  width: 100%;  
}
```



- When the element position is set to absolute, the element will scroll when the user scrolls the document.
- We can fix an element to a specific position on the page (regardless of user scrolling) by setting its position to fixed, and accompanying it with the familiar offset properties top, bottom, left, and right.

- Example:

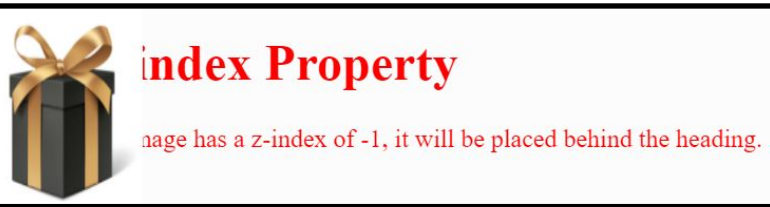
```
header {  
  background-color: #466995;  
  border-bottom: 1px solid #466995;  
  position: fixed;  
  width: 100%;  
}
```



- The sticky value is another position value that keeps an element in the document flow as the user scrolls, but *sticks* to a specified position as the page is scrolled further.
- This is done by using the sticky value along with the familiar offset properties, as well as one new one.
- 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).



- When elements on a web page have combinations of different positions, their contents can overlap, making the content difficult to read.
- The z-index property specifies the stack order of an element. z-index only works on positioned elements.
- The z-index property accepts integer values. Depending on their values, the integers instruct the browser on the order in which elements should be layered on the web page.



Position property in our LinkedIn project

```
.header{  
  padding:5px 150px;  
  position:sticky;  
  top:0;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  background-color: white;  
  z-index:1;  
}
```

```
.nav_icon{  
  position:relative;  
  font-size: var(--para_font_size);  
}
```

```
.info{  
  position:absolute;  
  top:-6;  
  right:-5;  
  background-color: brown;  
  color: white;  
  padding:1px 3px;  
  font-weight: 600;  
  border-radius: 6px;  
}
```

Block level
elements

CSS
Position

position: values;

z-index