Lesson:

CSS Positions (Part 1)

PW SKILLS

Topics Covered

- What is CSS Position?
- Why use CSS Position?
- Top, Bottom, Left, and Right properties
- Static Positioning
- Relative Positioning
- Absolute Positioning

What is CSS Position?

In CSS, positioning refers to the process of controlling the layout and placement of elements on a webpage.

There are five main positioning types available in CSS:

3. Static
4. Relative
5. Absolute
6. Fixed
7. Sticky

Why to use CSS Position?

1. **Control over element position:-** It allows us to control the position of an element within the document flow and provides better control over the layout of our webpage.
2. **Positioning relative to other elements:-** By using the position properties, we can easily position an element based on the position of its parent element.
3. **Removing elements from document flow:-** The position property also allows us to remove elements from their normal document flow and position it somewhere else on the webpage.

Example: Modals, pop-ups, etc. on the website can be easily placed on a webpage using the position property.

4. **Overlapping elements:-** The position property allows us to stack elements on each other by using its z-index property. The element having a higher z-index value will appear in front of the other elements having a lower z-index value.

Example: We see some pop-up window that opens on the websites which is often necessary to appear on the top of other content on the page. Like when we try to delete our account from any website a pop-up window asking 'Are you sure you want to delete' is assn example of a pop-up window that uses z-index for being on the top of the page.

5. **Positioning relative to viewport:-** We can position an element with respect to its viewport (screen width and height) so that it will always remain at the same position on the screen.

Example: Chat support icons in the websites are positioned using this property.

6. **Create a scroll effect:-** We can easily fix a website header and footer on the screen so that they are always visible there even when the user is scrolling the webpage.

7. **Accessibility:-** We can fix the logical order of the elements so that it will become more accessible for users who rely on screen readers or keyboard navigation.

Position Properties - Top, Left, Right, and Bottom
CSS provides several position-related properties,
including position, **top, bottom, left, and right,** which allows you to specify the exact position of an element on a webpage.

- **Top:** This property specifies the offset from the top edge of the element from the top of its parent container.
- **Bottom:** This property specifies the offset from the bottom edge of the element from the bottom of its parent container.
- **Left:** This property specifies the offset from the left edge of the element from the left of its parent container.
- **Right:** This property specifies the offset from the right edge of the element from the right of its parent container.

These properties are used in conjunction with the position property to specify the exact offset location of an element on a webpage. We will see their examples in further lessons

Static Positioning

- This is the default position for all HTML elements.
- Elements with static positioning are positioned based on the normal document flow, meaning they are placed one after the other in the order they appear in the HTML code.

Example Code:

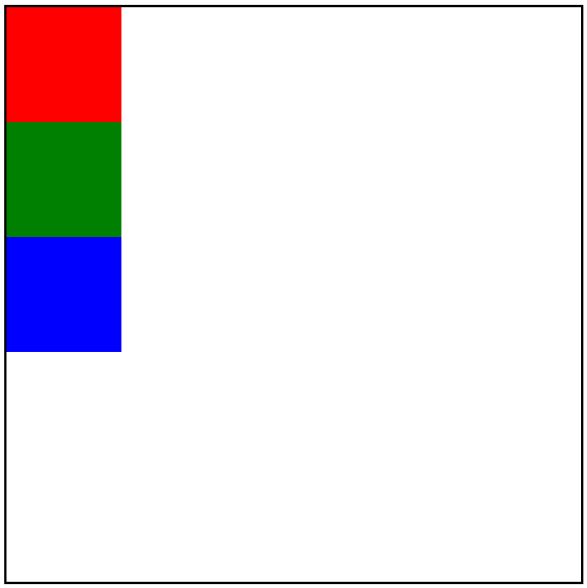
```
Unset
<style>
    /* for better visualization of example */
       .container {
         border: 2px solid black;
         height: 500px;
         width: 500px;
       }
       .box {
         width: 100px;
         height: 100px;
       }
       .box1 {
         background-color: red;
         position: static;
       }
```

```
        .box1 {
          background-color: red;
          position: static;
        }
        .box2 {
          background-color: green;
          position: static;
          top: 50px;
        }
        .box3 {
          background-color: blue;
          position: static;
        }

</style>

<body>
   <div class="container">
    <div class="box box1"></div>
    <div class="box box2"></div>
    <div class="box box3"></div>
   </div>
</body>
```

Browser Output



As we can see in the above output that the default output. After applying the top property on **box 2** which will provide a space from the top of **50px** as per code, but has no effect because we are using the static property which is the default one, and the properties like **top, bottom, left, and right** have no effect in.

Relative Positioning

With relative positioning, an element can be moved up, down, left, or right from its normal position, but it still takes up its original space in the document flow.

Properties of position Relative:-

- It will not break the normal document flow to position the element on the page.
- The properties like top, left, right, bottom, and z-index will have an effect on the element.
- The element will leave the space at its original position & other content will not be adjusted to fit into any gap left by the element.

Example:-

To understand, we will take the below HTML in which we have three block elements, box1, box2, and box3.

```
Unset
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <title>CSS Positioning</title>
    <link href="style.css" rel="stylesheet"
type="text/css">
  </head>

  <body>
    <div class="container">
      <div class="box box1"></div>
      <div class="box box2"></div>
      <div class="box box3"></div>
    </div>
  </body>


</html>
```

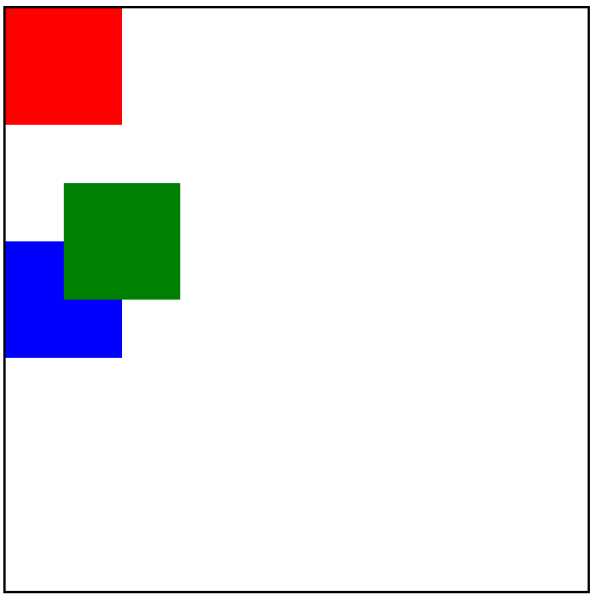
Let's apply CSS relative position to Box2.

```
Unset
/* style.css */

.box2 {
    position: relative;
    top: 50px;
    left: 50px;
    background-color: green;
}
```

After applying the position **relative**, we can see that **box2** has moved from its top by 50px and left position by 50px, leaving space at its original position and also not breaking the document flow.

Browser Output



Absolute Positioning

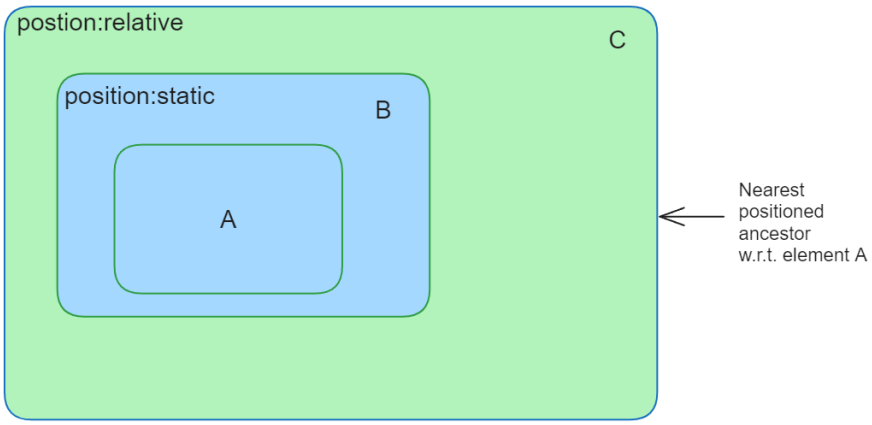
With absolute positioning, an element is positioned relative to its nearest positioned ancestor. If there is no positioned ancestor, then the element is positioned relative to the document body.

Let's understand the term **nearest positioned ancestor,** mentioned in the above definition. We can break it down into two parts: **nearest ancestor & positioned ancestor.**

Nearest ancestor: As the name suggests, It is a parent element, which is nearest to the current element.

Positioned ancestor: A parent element that is not positioned static (eg. relative, absolute, fixed, or sticky.).

So, a parent element that is positioned and nearest to the current element is called the nearest positioned ancestor. Look at the below diagram, element C is called the nearest positioned ancestor for element A.



Properties of position Absolute

- It will break the normal document flow to position the element on the page.
- The properties like top, left, right, bottom, and z-index will have an effect on the element.
- The element will not leave any space in its original position.

This type of positioning is often used to position elements precisely on a web page.

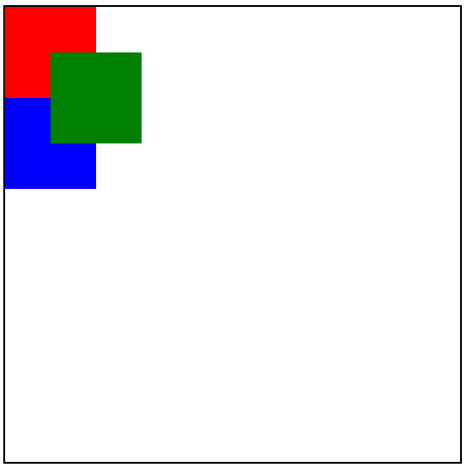
Example: Let's apply CSS absolute position to Box2.

```
Unset
.container {
      border: 2px solid black;
      height: 500px;
      width: 500px;
      position: relative;
}

.box2 {
     position: absolute;
     top: 50px;
     left: 50px;
     background-color: green;
 }
```

Browser Output

This is a wrapper box



As we can see in the output below, the **box2** has come out of the flow of the document and is also positioned 50px left and 50px top with respect to the **container** (positioned ancestor).