Css Positions

In CSS, the position property is used to control the positioning of an element on the webpage.
It determines how an element is placed within its containing element and how it interacts with other elements around it. The position property can take one of the following values:

if u use any position property they takes some more properties like:

1. right
2. left
3. up
4. down

1. Static (Default)

Syntax:

```
div {
 position: static; /* Default value */
}
```

2. relative. The element is positioned relative to its normal position in the document flow.

   It will move from its current poistion

Syntax:

```
div {
 position: relative;
 top:20px;
 left:20px;
```

```
}
```

3. absolute. Used when you want to place an element exactly at a certain location within a parent container
   It will move from its parent poistion
   if u apply below code it wii go right downword

Syntax:

```
div {
 position: absolute;
 right:0;
 bottom:0;
}
```

4. Fixed .The element is positioned relative to the viewport, meaning it stays in the same position even when the page is scrolled.

Syntax:

```
div {
 position: fixed;
 right:30px;
 top:30px;
}
```

5. Sticky The element toggles between relative and fixed, depending on scroll position.
   It accept only top.

Syntax:

```
div {
 position: sticky;
 top:0;
```

```
}
```

Example

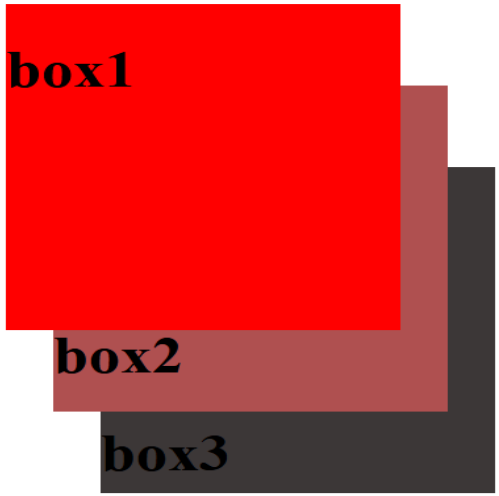
```
in html

<div class="box static-box">Static (default)</div>

<div class="box relative-box">Relative (moved 20px down & right)</div>

<div class="absolute-container">
    <div class="box absolute-box">Absolute (inside relative container)</div>
</div>

<div class="box sticky-box">Sticky (sticks to top while scrolling)</div>

<div class="box fixed-box">Fixed (bottom-right corner, always visible)</div>

in Css
<style>
.box {
    padding: 10px;
    color: white;
    margin-bottom: 20px;
  }

  .static-box {
    background: gray;
    position: static;
  }

  .relative-box {
    background: blue;
    position: relative;
```

```
    top: 20px;
    left: 20px;
  }

  .absolute-container {
    position: relative;
    height: 200px;
    background: #eee;
    margin-bottom: 30px;
  }

  .absolute-box {
    background: green;
    position: absolute;
    bottom: 0px;
    right: 10px;
  }

  .fixed-box {
    background: red;
    position: fixed;
    bottom: 10px;
    right: 10px;
  }

  .sticky-box {
    background: orange;
    position: sticky;
    top: 0;
    left: 80px;
  }
</style>
```

z-index only works on elements that have a position property set to relative, absolute, fixed, or sticky.
It doesn't work with statically positioned elements (the default value).

The z-index accepts integer values (positive, negative, or zero).
Elements with higher z-index values will be stacked in front of elements with lower z-index values.

Suppose you want to achive below example



Code

```
html
<div class="box1">
    <h1>box1>/h1>
</div>
<div class="box2">
    <h1>box2>/h1>
</div>
<div class="box3">
    <h1>box3>/h1>
</div>

in Css
```

```
<style>
.box {
    padding: 10px;
    color: white;
    margin-bottom: 20px;
  }

  .static-box {
    background: gray;
    position: static;
  }

  .relative-box {
    background: blue;
    position: relative;
    top: 20px;
    left: 20px;
  }

  .absolute-container {
    position: relative;
    height: 200px;
    background: #eee;
    margin-bottom: 30px;
  }

  .absolute-box {
    background: green;
    position: absolute;
    bottom: 0px;
    right: 10px;
  }

  .fixed-box {
    background: red;
```

```
  position: fixed;
  bottom: 10px;
  right: 10px;
}

.sticky-box {
  background: orange;
  position: sticky;
  top: 0;
  left: 80px;
}
</style>
```

Pseudo classes

- : :before
- : :after
- :nth-child()
- :nth-first-child(1)
- nth-first-child(odd)
- nth-first-child(even)

:: before , :: after

```
in html
<div>
    <h2>Pseudo class with text decoration wavy</h2>
<div>

in Css
<style>
    div::before{
        border-bottom: 10px solid red;
```

```
        display: block;
        content: '';
        width: 400px;
    }

    div::after{
        border-bottom: 10px solid red;
        display: block;
        content: '';
        width: 400px;
    }

    h2{
        text-decoration: underline wavy purple;
    }
    </style>
```

nth-child

```
in html
<ul>
    <li>tea>/li>
    <li>coffe>/li>
    <li>juice>/li>
    <li>soft drink>/li>
</ul>

in Css
<style>
    ul li:nth-child(odd){
        color: purple;
        list-style-type: upper-alpha;
    }
```

```
    ul li:nth-child(1){
        color: red;
    }

    ul li:nth-last-child(1){
        color: darkorange;
        font-size: 30px;
    }
</style>
```