

# CSS Positioning

## Working With Floats

- **Definition:** Floats are used to remove an element from its normal flow on the page and position it either on the left or right side of its container. When this happens, the text will wrap around that floated content.

Example Code

```
float: left;
```

```
float: right;
```

- **Clearing Floats:** The clear property is used to determine if an element needs to be moved below the floated content. When you have multiple floated elements stacked next to each other, there could be issues with overlap and collapsing in the layouts. So a clearfix hack was created to fix this issue.

Example Code

```
.clearfix::after {  
  content: "";  
  display: block;  
  clear: both;  
}
```

## Static, Relative and Absolute Positioning

- **Static Positioning:** This is the normal flow for the document. Elements are positioned from top to bottom and left to right one after another.
- **Relative Positioning:** This allows you to use the top, left, right and bottom properties to position the element within the normal document flow. You can also use relative positioning to make elements overlap with other elements on the page.

Example Code

```
.relative {  
  position: relative;
```

```
top: 30px;  
left: 30px;  
}
```

- **Absolute Positioning:** This allows you to take an element out of the normal document flow, making it behave independently from other elements.

Example Code

```
.positioned {  
  position: absolute;  
  top: 30px;  
  left: 30px;  
  background-color: coral;  
}
```

### Fixed and Sticky Positioning

- **Fixed Positioning:** When an element is positioned with `position: fixed`, it is removed from the normal document flow and placed relative to the viewport, meaning it stays in the same position even when the user scrolls. This is often used for elements like headers or navigation bars that need to remain visible at all times.

Example Code

```
.navbar {  
  position: fixed;  
  top: 0;  
  width: 100%;  
}
```

- **Sticky Positioning:** This type of positioning will act as a relative positioned element as you scroll down the page. If you specify a `top`, `left`, `right` or `bottom` property, then the element will stop acting like a relatively positioned element and start behaving like a fixed position element.

Example Code

```
.positioned {  
  position: sticky;  
  top: 30px;  
  left: 30px;  
}
```

### Working With the z-index Property

- **Definition:** The z-index property in CSS is used to control the vertical stacking order of positioned elements that overlap on the page.

#### Example Code

```
.container {  
  position: relative;  
  width: 300px;  
  height: 300px;  
  border: 1px solid black;  
}
```

```
.box1 {  
  position: absolute;  
  z-index: 1;  
  background: lightcoral;  
  top: 20px;  
  left: 20px;  
  width: 100px;  
  height: 100px;  
}
```

### Assignment

Review the CSS Positioning topics and concepts.

Please complete the assignment

Submit

Ask for Help

Navigated to CSS Positioning Review