ADVANCED CSS POSITIONING

STATIC POSITIONING

▸ This is the normal flow of the document, the **default**
▸ Elements render in order, as they appear in the document flow.

```css
.my-class {
  position: static;
}
```

RELATIVE POSITIONING

▸ Relative positioning moves an element *relative to where it would have been in normal flow*.

▸ For example, "left: 20px" adds 20px to an element's **left** position

▸ **Creates a coordinate system for child elements.**

```css
.my-class {
  position: relative;
  top: 20px;
  left: 30%;
}
```

ABSOLUTE POSITIONING

▸ When the *position* property is given a value of *absolute*, an element is taken out of the normal flow of the document.

▸ This element no longer affects the position of other elements on the page (they act like it's not there).

▸ You can add the *right, top, left* and *bottom* properties to specify where the element should appear relative to its first positioned (not static) ancestor element

```css
.my-class {
  position: absolute;
  top: 0;
  left: 500px;
}
```

FIXED POSITIONING

‣ When the *position* property is given a value of *fixed*, the element is positioned in relation to *the browser window*

‣ When the user scrolls down the page, it stays in the same place.

‣ You can add the *right, top, left* and *bottom* properties to specify where the element should appear in relation to the browser window.

```
.my-class {
  position: fixed;
  top: 0;
  left: 500px;

}
```

OVERLAPPING ELEMENTS — Z-INDEX

‣ When using relative, fixed or absolute positioning, elements can overlap.

‣ When elements overlap, the elements that appear later in the HTML code sit on top of those that appear earlier in the page.

‣ If you want to control which elements are layered on top of each other, you can use the z-index property.

‣ This property takes a number — the higher the number the closer that element is to the front.

‣ Similar to 'bring to front' and 'send to back' in programs like *Adobe Illustrator*.

```css
.my-class {
    z-index: 10;
}
```

WANT TO LEARN MORE?

Resources for more info/examples:

- ‣ A List Apart: [CSS Positioning 101]