

**ALWAYS CHECK OUT BOOTSTRAP BEFORE DOING YOUR OWN CSS!!!**

**NO NEED TO REINVENT THE WHEELS!!!**

### SASS vs LESS vs SCSS

<https://www.ionos.com/digitalguide/websites/web-development/sass/>

<https://stackoverflow.com/questions/52003198/differences-between-scss-and-less>

#### Key Notes:

- What are they? CSS Preprocessors. A CSS preprocessor is basically a scripting language that extends CSS and then compiles it into regular CSS.
- Allow users to define variables, add functions, math equations etc.

### CSS

#### Section 1: Display and Alignment

#### **Align Horizontally v Vertically:**

- 1) Set display to flex,
- 2) align-items: center; /\* Vertical alignment \*/
- 3) justify-content: center; /\* Horizontal alignment \*/

For horizontal alignment, can also use **text-align** property

For example: text-align: center;

---

#### *display* properties:

#### **Resources:**

[https://www.w3schools.com/css/css\\_inline-block.asp](https://www.w3schools.com/css/css_inline-block.asp)

[https://www.w3schools.com/cssref/pr\\_class\\_display.php](https://www.w3schools.com/cssref/pr_class_display.php) (List of All Display Properties)

#### Key Notes:

##### **a. Block vs Inline**

- 1) block: **add line breaks after each element (by default of <div> and <p>)**
- 2) inline: **width/height not respected; margin/padding not respected; (by default of <span>)**

- b. **Grid vs Flex** (<https://blog.logrocket.com/css-flexbox-vs-css-grid/>)
- 

## **Section 2: Selectors**

### **!important property:**

[https://www.w3schools.com/css/css\\_important.asp](https://www.w3schools.com/css/css_important.asp)

The **!important** rule in CSS is used to add more importance to a property/value than normal.

In fact, if you use the **!important** rule, it will override ALL previous styling rules for that specific property on that element!

---

### **CSS Attribute Selector**

[https://www.w3schools.com/css/css\\_attribute\\_selectors.asp](https://www.w3schools.com/css/css_attribute_selectors.asp)

```
[attribute to be specified] {  
    ...  
}
```

---

### **::before**

<https://developer.mozilla.org/en-US/docs/Web/CSS/::before>

In CSS, **::before** creates a [pseudo-element](#) that is the first child of the selected element. It is often used to add cosmetic content to an element with the [content](#) property. It is inline by default.

---

### **Multiple Classes and ID selector:**

<https://css-tricks.com/multiple-class-id-selectors/>

Key Takeaway:

```
#header.callout {
```

- Select all elements that have an id of “header” and class of “callout”

```
#header .callout { }
```

- Select all elements that have the class name of “callout” that are descendants of elements with id “header”

---

### Apply the Same Styling to Multiple Selectors:

Use comma (,) to separate the selectors.

For example,

```
.class1 .class2, .class3 .class4 {  
    Styling Goes Here...  
}
```

**Apply styling to all elements with class = “class2” that are descendants of “class1”, AND all elements with class = “class4” that are descendants of “class3”.**

---

**+, >, ~ Selectors**

<https://techbrij.com/css-selector-adjacent-child-sibling>

### Key Notes:

>: It is a child selector, which selects **DIRECT** child elements of a specified parent element.

+: It is **Adjacent sibling** combinator. It combines two sequences of simple selectors having the same parent and the second one must come **IMMEDIATELY** after the first.

~: It is **general sibling** combinator and similar to Adjacent sibling combinator. The difference is that the second selector does **NOT** have to immediately follow the first one means It will select all elements that is preceded by the former selector.

## Section 3: Animations

---

### Transform

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform>

The **transform** [CSS](#) property lets you rotate, scale, skew, or translate an element. It modifies the coordinate space of the CSS [visual formatting model](#).

---

**@keyframe**

<https://developer.mozilla.org/en-US/docs/Web/CSS/@keyframes>