Attributes: style

Time to spice up your JSX elements!

In this next section, we'll be learning how to style JSX elements. It's slightly different than normal HTML.

- These are some of the rules to keep in mind:
- Apostrophes (') or quotes (") on strings
- **camelCase** (See next page)
- Commas (,) instead of a semicolon (;)
- Two curly **brackets {{}}** instead of **quotes ("")**

HTML

```
<h1 style= "color:red; font-size:60px">
...
</h1>
```

——→ JSX

```
<h1 style={{ color: "red", fontSize: 60 }}>...</h1>
```

What is camelCase?

You are probably wondering what camelCase is. Well, we have got the answers.

> "It's the practice of writing phrases in a single 'word' with no spaces. We use it to write variable names in JavaScript."

The concept is very basic. There are 2 simple rules to follow:

1. No spaces, no underscores (_) and no dashes (-)
2. If the variable is composed of multiple words, all words begin with a **capital letter** except **the first word.**

Examples

Let's see more styling examples.

We will frequently find that the HTML version is ordered right before the JSX version.

HTML

```
<div style='text-align:center'/>
```

JSX

```
<div style={{textAlign:'center'}}/>
```

HTML

```
<div style="transform:translateX(25px)"/>
```

JSX

```
<div style={{transform:'translateX(25px)'}}/>
```

HTML

```
<div style="box-shadow:0 5px 8px #000"/>
```

JSX

```
<div style={{boxShadow:"0 5px 8px #000"}}/>
```

The units of measurement in JSX:

Some **CSS properties** can take a number, it will have **px (pixels)** as a unit by default. If you need to specify a different unit, you have to use the string type.

HTML

```
<div style="height:10px"/>
```

JSX

```
<div style={{height:'10px'}}/>
```

HTML

```
<div style="height:10px"/>
```

JSX

```
<div style={{height:10}}/>
```

HTML

```
<div style="height:10vw"/>
```

JSX

```
<div style={{height:'10vw'}}/>
```