

CSS Mistakes to Avoid



Art by
[@johnycaballeroart](#)

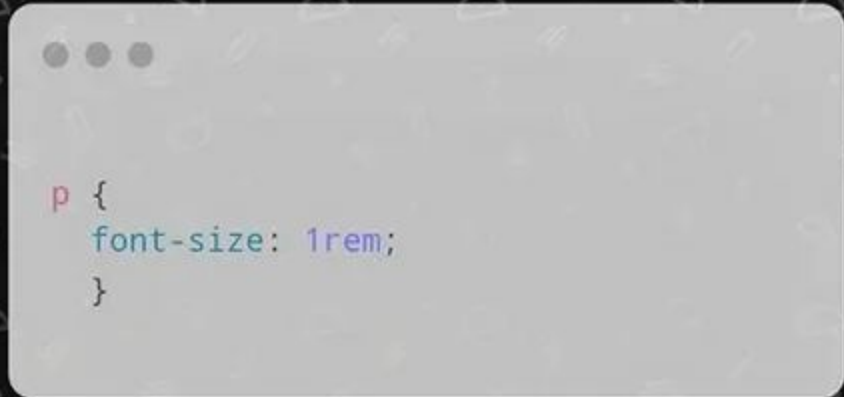
[@codebysid](#)
.....

Using px



```
p {  
  font-size: 16px;  
}
```

px locks your design to a fixed size for all screen sizes and makes your design hard to scale to different devices.



```
p {  
  font-size: 1rem;  
}
```

We prefer using relative measurements such as em, %, rem, and others whenever possible, they ensures responsiveness according to the screen size and zoom level

Using inline CSS

```
<h2 style="color:blue;font-size:3rem;">Python</h2>
<h2 style="color:blue;font-size:3rem;">CSS</h2>
<h2 style="color:blue;font-size:3rem;">Java</h2>
<h2 style="color:blue;font-size:3rem;">JavaScript</h2>
```

You are using same code again and again on different elements but the goal should be to minimise the line of code by following reusability. Here we are breaking the DRY rule.

Not using consistent Naming

```
.redBox {  
  color: red;  
}  
  
.box-head {  
  font-size: 3rem;  
}
```



Make sure to use same naming style while defining class, id, or any other attribute because it would be hard to read and understand by others or any other team member.

.redBox and .box-head are different, you should use like .redBox and .boxHead or .red-box and .box-head

Misusing !important



```
p {  
  background-color: red;  
}  
  
p{  
  background-color:blue;!important  
}  
  
p{  
  background-color:green;!important  
}
```

Using !important normally means you're trying to override some property with a higher specificity.

The problem arise when you have to override an !important rule because only an !important rule can override another !important rule.

Using Single File

Using single file in a small project is totally fine but when it comes to a large scale project using single CSS file would be a headache. Consider a CSS files with hundreds and thousands line of code.

Divide your CSS code in multiple files by grouping similar styling.