Swipe Left

Common CSS Mistakes Made By Developers

Hanzala
@coding_doc

Ariba M.
@frontendcharm

@coding_doc
@frontendcharm

Keep Swiping...

1. Over-Qualifying Selectors

Being overly specific when selecting elements to style is not good practice. The following selector is a perfect example of what I'm talking about:

style.css

```css
#nav ul li a { ... }
```

This HTML structure is perfectly correct, but the CSS selector is really what I'm worried about.

You don't have to put ul and li in the selector syntax because all the a elements inside the #nav are inside list items, so there's no reason for that bit of specificity.

Thus, you can condense that selector as:

style.css

```css
#nav a { ... }
```

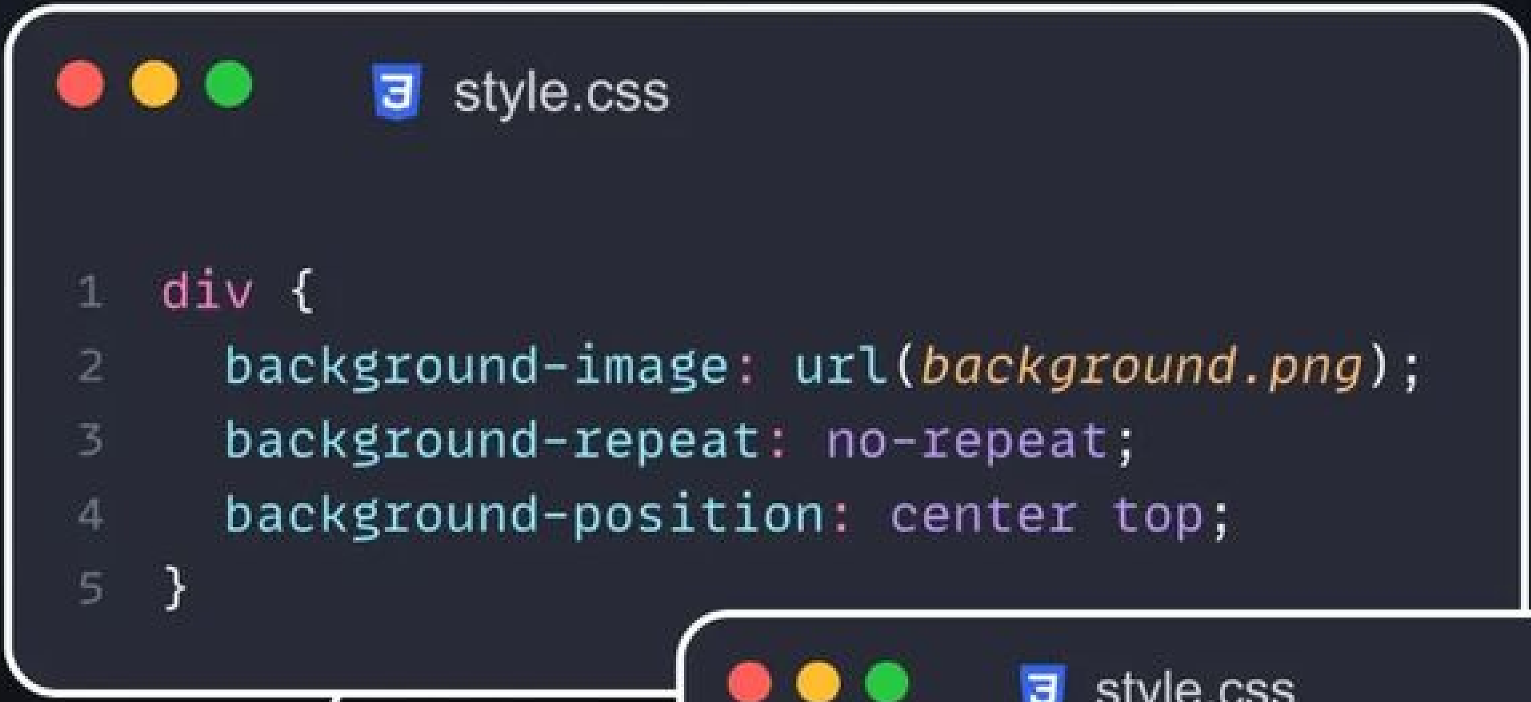
@coding_doc
@frontendcharm

Keep Swiping...

2. Not Using Shorthand Properties

Most people know about and use some shorthand, but many don’t make full use of these space saving properties.

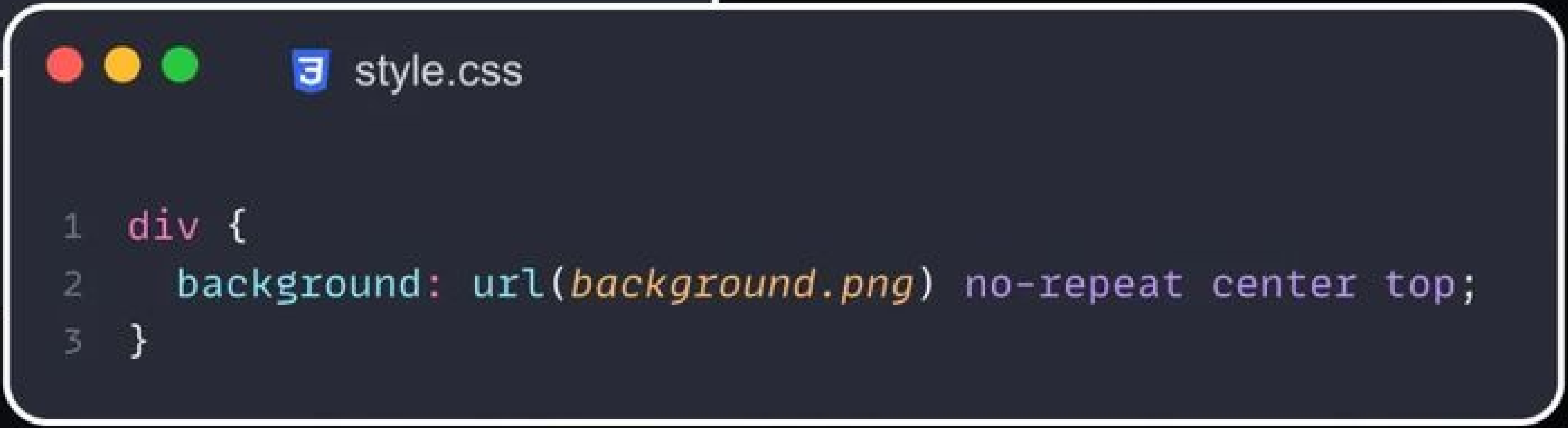
Shorthand properties can be used to set several properties at once, in a single declaration, instead of using a separate declaration for each individual property. This can save a lot of space in your CSS file.



style.css

```css
div {
  background-image: url(background.png);
  background-repeat: no-repeat;
  background-position: center top;
}
```

For example this background properties can be written in shorthand CSS as such:



style.css

```css
div {
  background: url(background.png) no-repeat center top;
}
```

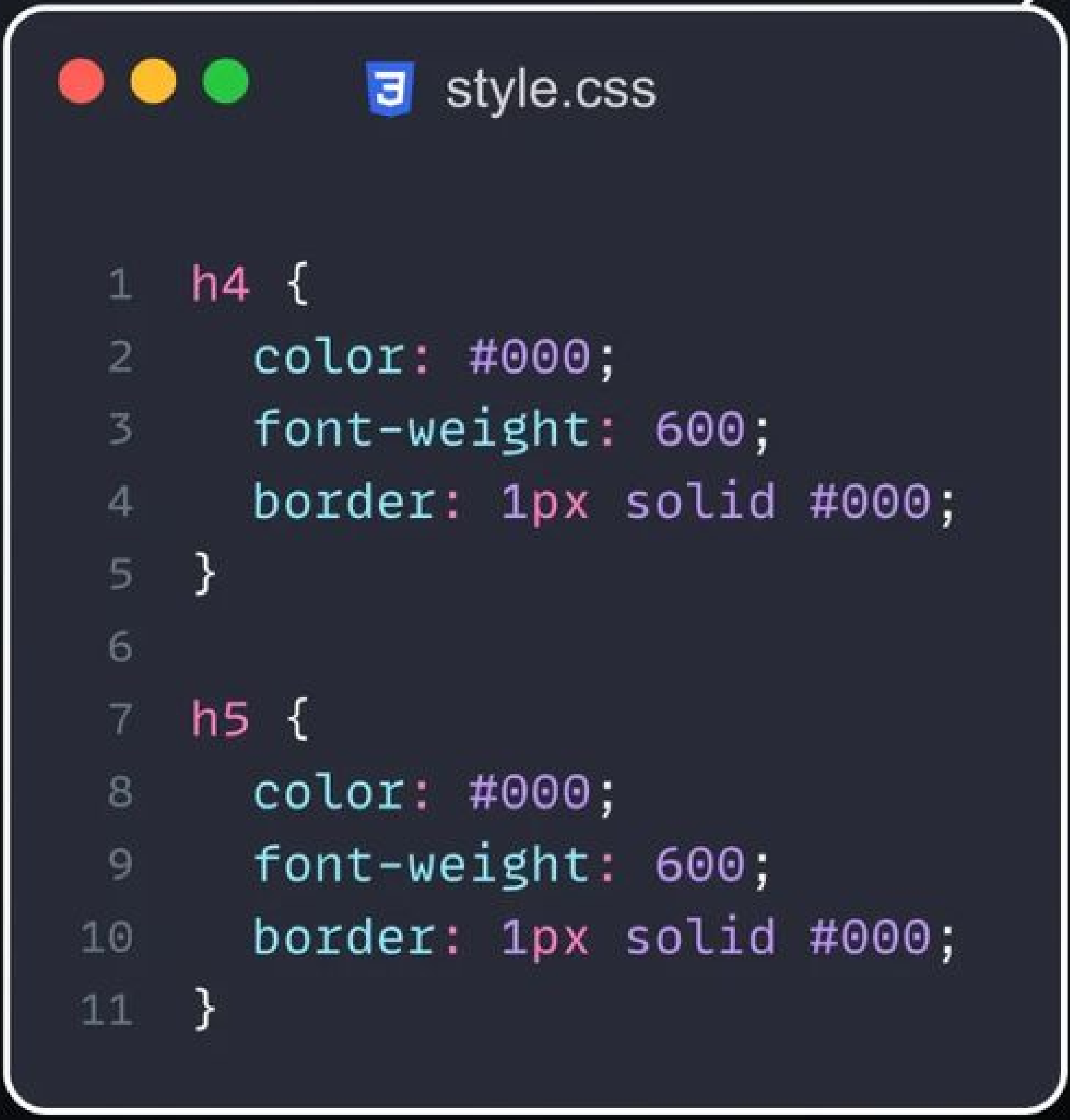
@coding_doc
@frontendcharm

Keep Swiping...

3. Using Redundant Properties

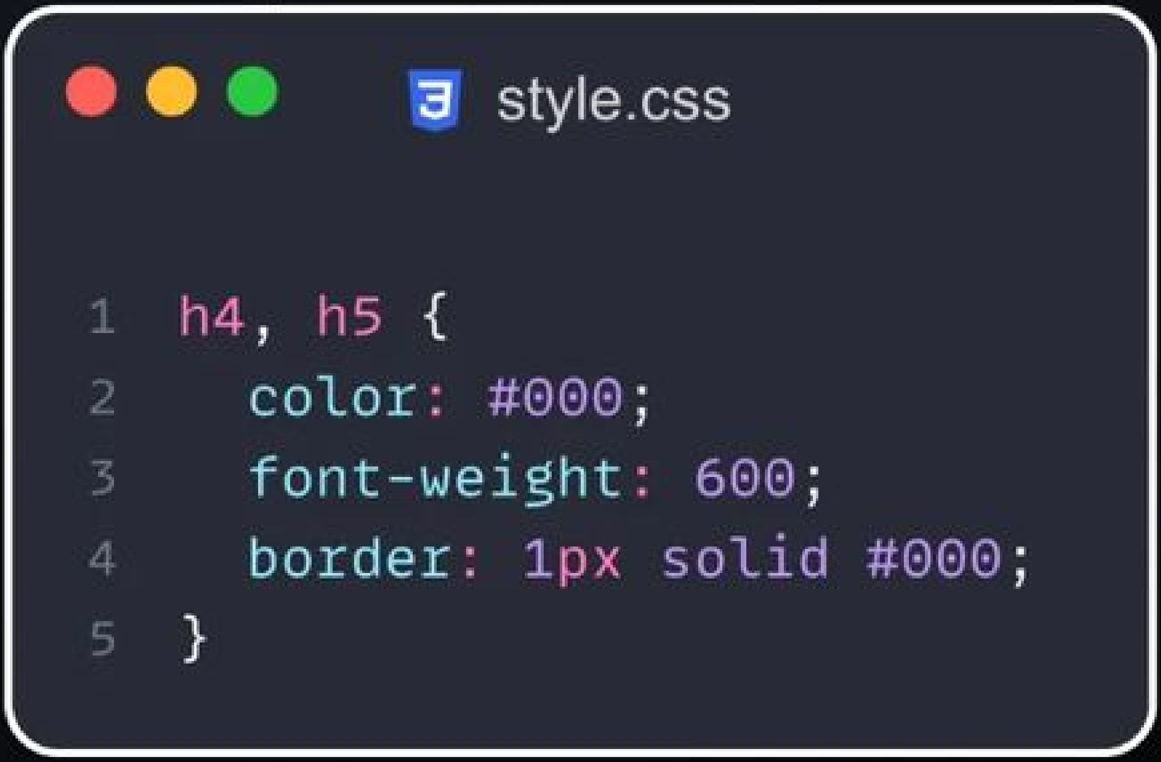
I often find many developers applying the same properties to multiple selectors.

For example styling an <h4> in the header to look exactly like the <h5> in the footer. Instead of writing individual css for both, just combine them, with the selectors separated by a comma (,):


style.css

```css
h4 {
  color: #000;
  font-weight: 600;
  border: 1px solid #000;
}

h5 {
  color: #000;
  font-weight: 600;
  border: 1px solid #000;
}
```


style.css

```css
h4, h5 {
  color: #000;
  font-weight: 600;
  border: 1px solid #000;
}
```

I hope you're seeing the trend here. Try to be as terse and as efficient as possible

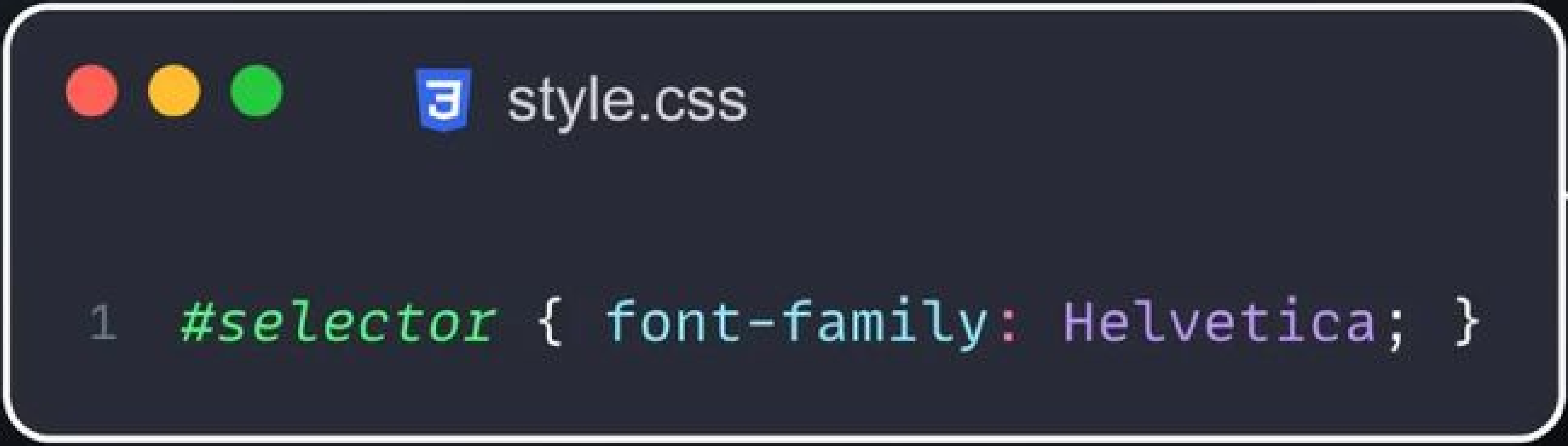
@coding_doc
@frontendcharm

Keep Swiping...

4. Not Providing Fallback Fonts

In a perfect world, every computer would always have every font you would ever want to use installed. Unfortunately, we don't live in a perfect world.

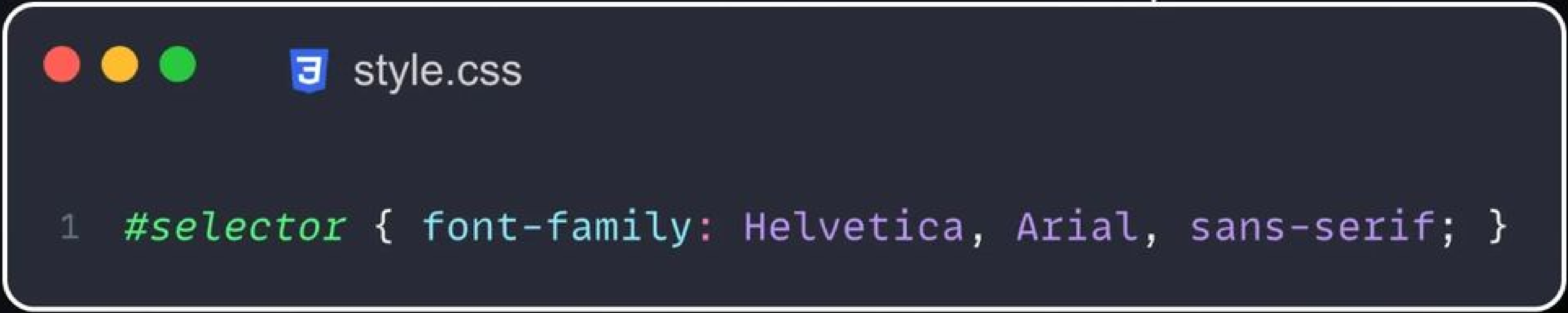
It is very important to always use fallback fonts. This means that you should add a list of similar "backup fonts" in the font-family property. If the first font does not work, the browser will try the next one, and the next one, and so on. For example:



style.css

```css
#selector { font-family: Helvetica; }
```

Can be expanded with fallback fonts as such:



style.css

```css
#selector { font-family: Helvetica, Arial, sans-serif; }
```

@coding_doc
@frontendcharm

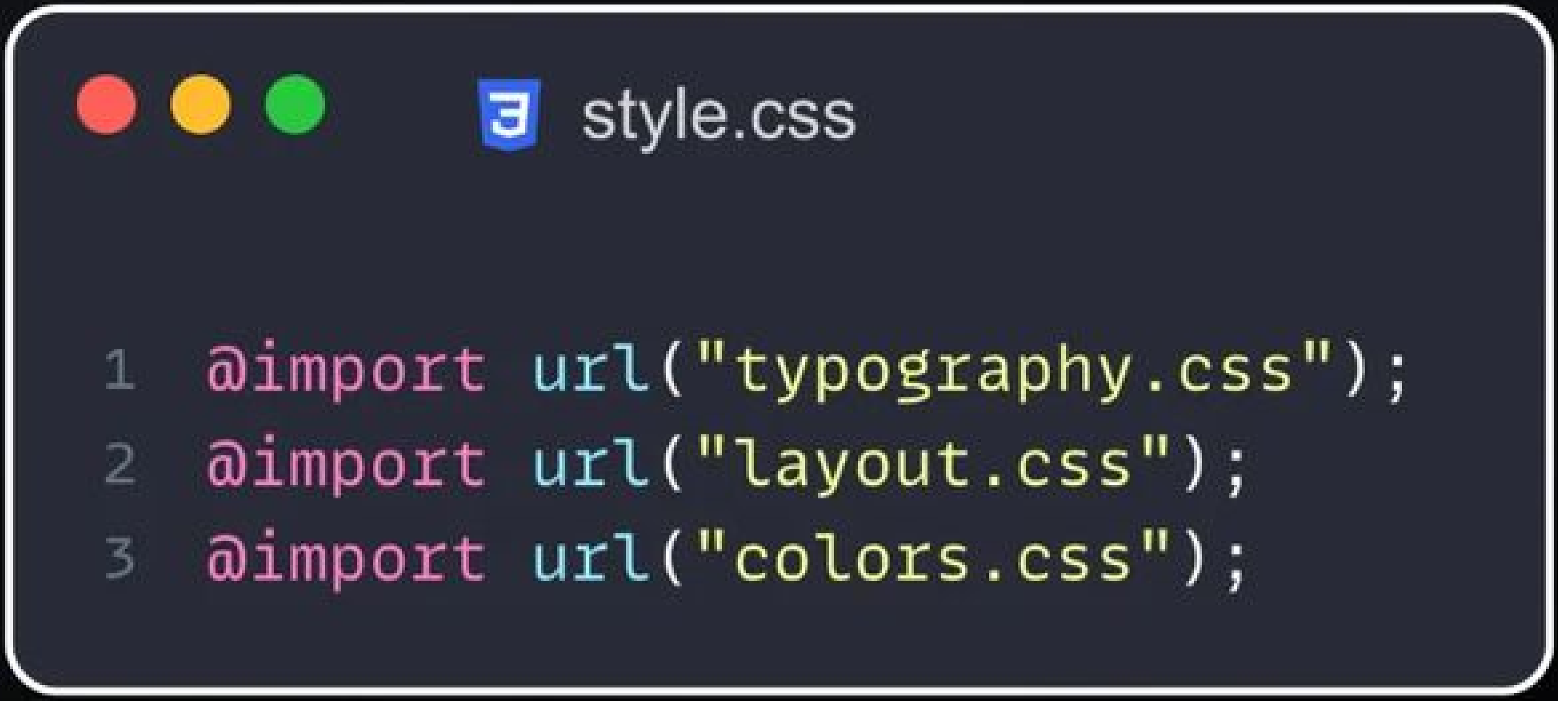
Keep Swiping...

5. Using Only One Stylesheet For Everything

If you're working on a small project, it's fine to use a single style sheet. However, if you're working on a large project, splitting style sheets into distinct ones is strongly suggested because it'll be easier to manage and provide better modularity.

Different CSS files might be used for different fixes. By organizing CSS into disparate stylesheets, you'll know immediately where to find a style you want to change.

You can do this by importing all the stylesheets into a stylesheet like so:



```css
@import url("typography.css");
@import url("layout.css");
@import url("colors.css");
```

@coding_doc
@frontendcharm

Keep Swiping...

6. Over-Using Z-Index Values

I've seen developers over-using the z-index by using high values when trying to put an element in front of the other.

What if another person needs to move another element up ? This person will have to set an even higher z-index value. Isn't it ?

I recommend using it wisely and moderately, increasing the necessary amount to achieve the desired result.

Tip: Using a preprocessor like Sass or Stylus, will help you in handling z-index layers on your application smartly.

Turn ON notification

Did you Find it Useful?

@coding_doc

Hanzala

@frontendcharm

Ariba

Follow For More !