Getting Started with CSS Syntax

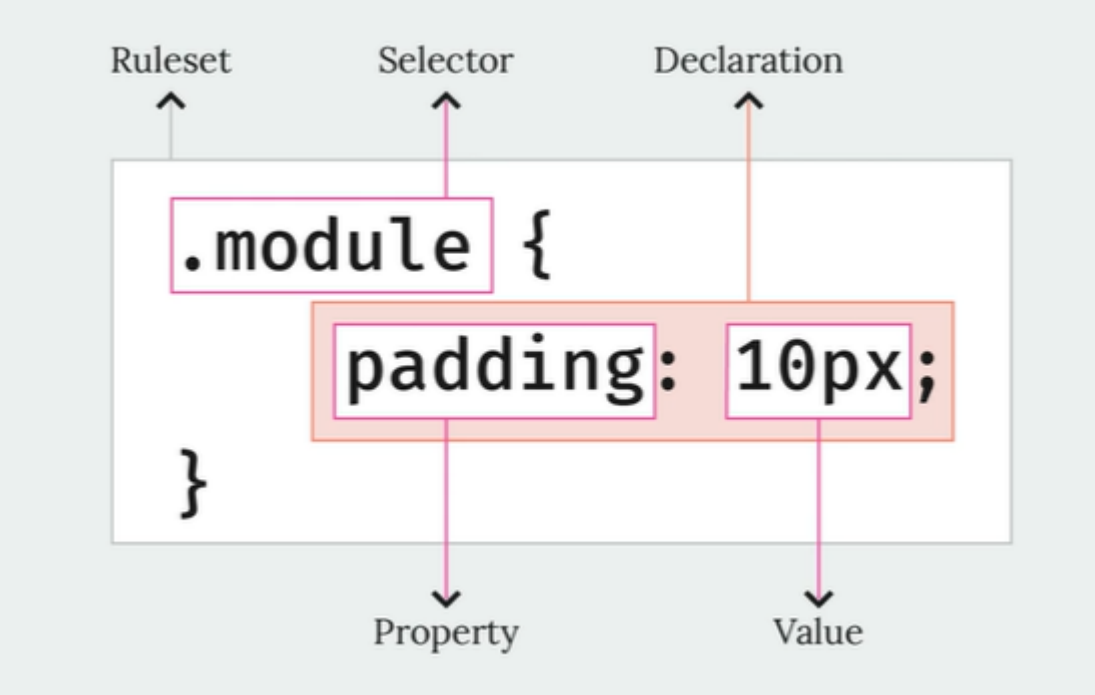
Introduction

Welcome to the first CSS lesson inside the Academy. At the beginning, we need to get familiar with the correct CSS syntax and that will be the focus of this lesson.

When you're starting to play around with CSS at the very beginning, like any other language, you have to get used to the syntax. Like any other syntax, there are a bunch of little things you need to know. Some characters and the placement of them is very important and required for the CSS to work correctly. And some characters are more about clean-looking code and generally followed standards but don't matter for the CSS to work.

CSS Ruleset

First, so we have the terminology down. A CSS ruleset consists of a selector and declaration wrapped in curly braces. A single ruleset can have multiple selectors and multiple declarations.



In this example, an element with a class module is the selector (our target element) and we have only one CSS declaration for this target. A declaration is a pair of property and value, and here, we have padding - a CSS property - and 10px - the CSS value.

Now, let's look at a few different symbols and I'll explain if they are important or not.

Braces

First, braces. All CSS rulesets must have opening and closing curly braces. You would also see additional braces (e.g. rounded brackets) in media queries, in animations.

```
ul li {padding: 10px; color: #342434; margin: 10px;}

div.main {border: 2px #919191 solid; background-image: url('main-bg.jpg');}
header.title {
    padding: 10px;
}
```

If you miss the opening brace, CSS will keep reading as if the next bit of text is still part of the selector. Then it's likely to find a character like a semicolon, which is invalid as part of a selector - and the code will break. Breaking likely means it will mess itself and the next ruleset, and recover after that.

Missing a closing bracket is a bit worse in that it's likely to mess up the rest of the entire CSS file unless it somehow finds a double closing brace and can resolve that first missing one. Overall point: braces are very important!

Semicolons

Next: semicolons. Each declaration in a ruleset (a property and value pair) ends in a semicolon.

```
ul li {padding: 10px; color: #342434; margin: 10px;}
```

That semicolon is required. Otherwise, CSS will keep reading the next property as if it's part of the value of the previous declaration. You can leave off the semicolon on the last declaration in a ruleset. I'd warn that doing so manually will require more effort than it's worth, and it's best to leave that to CSS minification tools.

Spaces

Spaces are sometimes important, and sometimes you can add as many as you like.

```
ul li {        padding:        10px;color: #342434; margin: 10px;}
```

You can add the spaces in most of the CSS code and it will only make the CSS code heavier, because CSS minification basically means removing empty spaces. That is why you sometimes see CSS as a bunch of code all squished together. But there are a few places where spaces are very important in CSS.

First, you can't put spaces in properties, function names, basically anywhere you name things. Adding spaces in those situations effectively changes the names, breaking the code.

Another important place is in selectors. One space in a selector means you're selecting descendants - children elements of the previous part of the selector, but we will have a closer look at the CSS selectors in a separate lesson.

Avoid Stray Characters

It's also very important to avoid stray characters. And this is important in any programming language. Every character in code matters, so random unnecessary symbols will almost certainly break things.

Line Breaks

And not that important are line breaks. A line break is treated like any other white space in CSS, so feel free to use them as needed. You can add as many line breaks in your code as you'd like. That is as long as it doesn't break any other rules I just explained earlier.

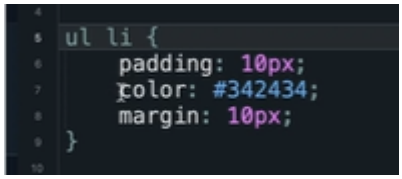
CSS Formatting

That is basically all you need to know about the CSS syntax at the beginning. But there is one more thing I'd like to mention, and that is CSS formatting. And I don't mean formatting your website with CSS but formatting the CSS code itself, the inside of your custom CSS file.

I think it is important to think about formatting and to keep your CSS code nice and organized. Because you are in and out of that file all the time, especially during development, you are scrolling through it, searching for things you want to edit or troubleshoot and it is important that the file is easy to read, that the content is accessible.

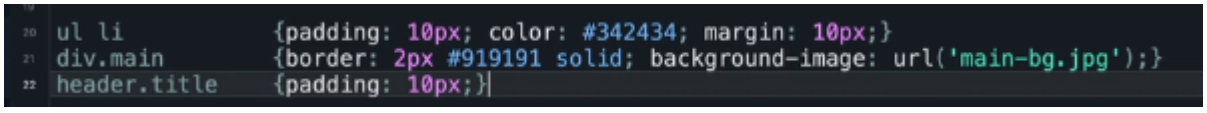
And there is no right or wrong way to format your code. You should do it in a way that makes sense to you. But that being said, there are some common formats you'll often see.

One looks like this: there is a selector on one line and an opening brace, next, you have your declarations, one on each line, and the closing bracket below. The declarations are indented and it makes it easy to browse different CSS properties.



```
ul li {
    padding: 10px;
    color: #342434;
    margin: 10px;
}
```

Another style of formatting is on the single line, which kind of saves space and doesn't require you to scroll that much (if your CSS is very long) but it is a bit harder to read. It makes the selector easy to scan through, though.



```
ul li           {padding: 10px; color: #342434; margin: 10px;}
div.main        {border: 2px #919191 solid; background-image: url('main-bg.jpg');}
header.title    {padding: 10px;}
```

And one important thing is that you should also decide if you want to keep your CSS rulesets connected in the groups for things like header, footer, basically where it applies, or make separate sections for typography, layout, or similar.

```
/* Header */

ul li {
    padding: 10px;
    color: #342434;
    margin: 10px;
}

div.main {
    border: 2px #919191 solid;
    background-image: url('main-bg.jpg');
}

header.title {
    padding: 10px;
}
```

Any approach will be good as long as it will help you find your code faster.

CSS Comments

You can use comments to declare some structure. Forward slash and the star - that would be an opening comment. Anything in between is a comment and then to close it we add a star and forward slash.

```
/* multiple
   lines
   comment
   */
```

Final thoughts

I hope this information will help you pay closer attention to how you write your CSS and also just be able to catch any formatting issues and troubleshoot your code easier now that you are familiar with the correct syntax.

Resources

PRACTICE CSS:

CSS Exercises

CSS Cheatsheet (available to download in the Downloads section of this lesson)

Action Items

- [ ] Download and print the CSS Cheatsheet.