

CSS Basics

is a style sheet language used to decorate elements. A separate styling language allows front-end developers and designers to unleash their creativity and build unique design solutions.

CSS follows its own syntax, which is different from HTML, even though it uses its elements and attributes. To understand how CSS works, you need to know what it's comprised of.

We use CSS to write rules that make everything look like we imagine. However, creating CSS rules isn't possible without knowing the anatomy and purpose of selectors, declarations, [properties](#), and their values.

Definition of CSS



All languages have a basic vocabulary that is used by native speakers and even tourists who want to order a meal or ask for directions.

Advanced learners or native speakers possess a broader and richer vocabulary that makes their speech more exquisite and elaborate.

Likewise, HTML is the plain part of any website, while CSS is its beautiful companion.

CSS — Cascading Style Sheets — is a language that web developers use to decorate HTML elements displayed on a screen. Styles are usually saved in separate external CSS files.

SS syntax rules

```
h1 {  
    font-size: 32px;  
}
```

The CSS syntax consists of a set of rules. A rule, in turn, is comprised of 2 parts: the selector and the declaration block. The selector points to the element (or elements) you want to style, and declarations define what style you would like to apply.

For example, here's the rule to style all <h1> headings with the 32px

```
h1 {
```

```
font-size: 32px;
```

```
}
```

Each declaration ends with a semicolon (;), and the declaration block must be surrounded by curly braces ({ }).

CSS selector

```
h1 {  
    margin-left: 20px;  
}
```

A selector is a part of CSS syntax that points to the HTML element you want to style. For example, to style `<h1>` headings on the we'd need to use the selector `h1`. A rule for this selector will affect all `<h1>` elements on the page because we use the element type selector.

There are various selector types in CSS that allow us to target elements more precisely: by their ids, classes, attributes, or based on their location.

CSS declaration block

```
p {  
  color: Lavender;  
  font-size: 12px;  
}
```

The declaration block defines one or more properties of the selector element, and the number of declarations depends on how many properties you want to define. Each declaration includes the CSS property name and a value — the same as when using inline CSS.

For example, the following CSS rule has two declarations: one that sets the text of paragraphs <p> to lavender, and the other that sets its to 12px:

```
p {
```

```
color: Lavender;
```

```
font-size: 12px;
```

}

CSS properties

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

Each declaration consists of two parts — a property and a value — separated by a colon. Property refers to the characteristic you want to

define. For example, to define font size, we use the property `font-size`.

Properties allow you to style pretty much anything on the page: text, alignment,, etc. You can find alphabetical lists of all CSS properties on different resources.

CSS property value

```
body {  
  text-align: left;  
}
```

Each declaration has a property and its value. The value chooses one out of many possible appearances for a given property.

Every property used in CSS has a set of values it can take. Most values are relatively simple keywords or numeric values. Some properties have few allowed values. For example, the `text-align` property can take 6: `left`, `right`, `center`, and a couple of more specific ones.[\[3\]](#)

Other properties like `color` have a much larger number of possible values but also allow multiple value types. The color can be specified through and RGB values and color keywords.

Applying the same CSS rules to multiple elements

```
h1, h2, p {  
    color: red;  
}
```

In the style sheets, there are often elements using the same CSS rules. Let's say you want to apply the color red to all <h1>, <h2>, and <p> elements. To minimize the code, you can group selectors by typing them, separated with a comma:

```
h1, h2, p {
```

```
color: red;
```

}

Pro Tip! Some consider this syntax to be a combinator selector "or."

Curly braces in CSS

```
h1 {  
    margin-left: 20px;  
}
```

All languages have rules — not because people love rules, but because that's what makes communication possible. For example, the two written phrases "let's eat, grandma!" and "let's eat grandma!" carry very different meanings. While listeners can parse the meaning from the context, browsers can't (not just yet, at least).

So what happens if you use the wrong brackets or forget to use any brackets at all? Nothing — literally. The browser won't see that your code is a declaration and won't apply style to the elements. Make sure curly braces are always there, hugging the declaration from both sides.

CSS property-value order

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

Sometimes the order of things matters, and sometimes it doesn't.

When it comes to the CSS declaration syntax, the property-value order is crucial. The first part of the declaration has to be the property, and the second — its value. Typing them in the wrong order will result in an — browsers simply won't understand the code. Also, remember that there must be a colon (:) between the property and its value.

Using semicolons to close the declaration in CSS

```
body {  
    font-size: 16px;  
    margin: 20px;  
}
```

In the earlier versions of CSS, the semicolons (;) were used to separate declarations within the declaration block. However, this would cause problems as people often forget to add the symbol when adding more declarations to the block.

Nowadays, the purpose of semicolons isn't to separate declarations but to close them. Without a semicolon, the declaration won't render correctly or at all.