# Getting started with CSS

## What is CSS?

Like HTML, CSS is not a programming language. It's not a markup language either. **CSS is a style sheet language.** CSS is what you use to selectively style HTML elements. For example, this CSS selects paragraph text, setting the color to red:

p {

color: red;

}

Anatomy of a CSS ruleset

Let's dissect the CSS code for red paragraph text to understand how it works :



The whole structure is called a **ruleset.**(The term *ruleset* is often referred to as just *rule.)* Note the names of the individual parts:

**Selector**

This is the HTML element name at the start of the ruleset. It defines the element(s) to be styled (in this example, [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p) elements). To style a different element, change the selector.

**Declaration**

This is a single rule like color: red;. It specifies which of the element's **properties**you want to style.

**Properties**

These are ways in which you can style an HTML element. (In this example, color is a property of the [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p) elements.) In CSS, you choose which properties you want to affect in the rule.

**Property value**

To the right of the property—after the colon—there is the **property value**. This chooses one out of many possible appearances for a given property. (For example, there are many color values in addition to red.)

Selecting multiple elements

p, li, h1 {

color: red;

}

Different types of selectors

There are many different types of selectors. The examples above use **element selectors**, which select all elements of a given type. But we can make more specific selections as well. Here are some of the more common types of selectors:

| **Selector name** | **What does it select** | **Example** |
| --- | --- | --- |
| Element selector (sometimes called a tag or type selector) | All HTML elements of the specified type. | p selects <p> |
| ID selector | The element on the page with the specified ID. On a given HTML page, each id value should be unique. | #my-id selects <p id="my-id"> or <a id="my-id"> |
| Class selector | The element(s) on the page with the specified class. Multiple instances of the same class can appear on a page. | .my-class selects <p class="my-class"> and <a class="my-class"> |
| Attribute selector | The element(s) on the page with the specified attribute. | img[src] selects <img src="myimage.png"> but not <img> |
| Pseudo-class selector | The specified element(s), but only when in the specified state. (For example, when a cursor hovers over a link.) | a:hover selects <a>, but only when the mouse pointer is hovering over the link. |

There are many more selectors to discover. To learn more, see the MDN [Selectors guide](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started/Selectors).

## Fonts and text

1. First, find the [output from Google Fonts](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/What_should_your_web_site_be_like#Font) that you previously saved from [What will your website look like?](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/What_should_your_web_site_be_like). Add the [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link) element somewhere inside your index.html's head (anywhere between the [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head) and </head> tags). It looks something like this:

<link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">

This code links your page to a style sheet that loads the Open Sans font family with your webpage.

1. Add the following, all elements inside it inherit the same font-size and font-family.

html {

font-size: 10px; /\* px means "pixels": the base font size is now 10 pixels high \*/

font-family: "Open Sans", sans-serif; /\* this should be the rest of the output you got from Google fonts \*/

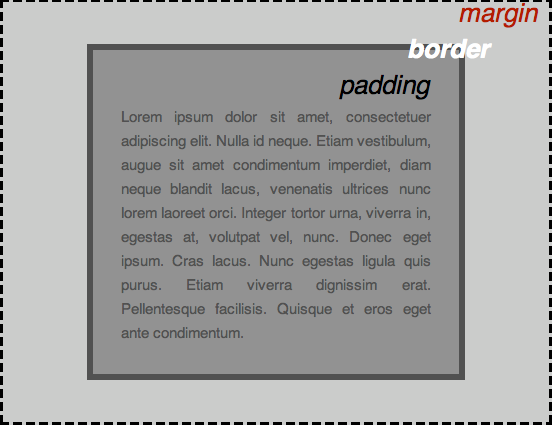
}

## CSS: all about boxes

Something you'll notice about writing CSS: a lot of it is about boxes. This includes setting size, color, and position. Most HTML elements on your page can be thought of as boxes sitting on top of other boxes.

CSS layout is mostly based on the *box model.*Each box taking up space on your page has properties like:

* padding, the space around the content. In the example below, it is the space around the paragraph text.
* border, the solid line that is just outside the padding.
* margin, the space around the outside of the border.



In this section we also use:

* width (of an element).
* background-color, the color behind an element's content and padding.
* color, the color of an element's content (usually text).
* text-shadow sets a drop shadow on the text inside an element.
* display sets the display mode of an element. (keep reading to learn more)

To continue, let's add more CSS. Keep adding these new rules at the bottom of style.css.

Styling the body

body {

width: 600px;

margin: 0 auto;

background-color: #FF9500;

padding: 0 20px 20px 20px;

border: 5px solid black;

}

Positioning and styling the main page title

h1 {

margin: 0;

padding: 20px 0;

color: #00539F;

text-shadow: 3px 3px 1px black;

}

text-shadow applies a shadow to the text content of the element. Its four values are:

* The first pixel value sets the **horizontal offset** of the shadow from the text: how far it moves across.
* The second pixel value sets the **vertical offset** of the shadow from the text: how far it moves down.
* The third pixel value sets the **blur radius** of the shadow. A larger value produces a more fuzzy-looking shadow.
* The fourth value sets the base color of the shadow.

Centering the image

img {

display: block;

margin: 0 auto;

}