Css basics

<!DOCTYPE html>

<html>

<head>

<link type="text/css" rel="stylesheet" href="stylesheet.css"/>

Linking css inside of html

<head>

<link type="text/css" rel="stylesheet" href="stylesheet.css"/>

<title>Result</title>

</head>

CHANGING THE COLOR OF THE TEXT IN CSS

p {

color: green;

}

Changing the color text and font

p {

font-family: Arial;

color: blue;

font-size: 24px;

}

Writing comments in css

*/\*I'm a comment!\*/*

<!DOCTYPE html>

There are two ways to put CSS in one place. This first is to put your CSS between <style></style> tags, right in the same file as your HTML. These <style> tags go inside the <head></head> of your webpage; check out the example in the editor to the right

<html>

<head>

<style>

p {

color: purple;

}

</style>

<title>Result</title>

</head>

<body>

<p>Check it out! I'm purple!</p>

</body>

</html>

LINKING CSS INSIDE OF HTML MAKING THEM BOTH TOGETHER

<!DOCTYPE html>

<html>

<head>

<link type="text/css" rel="stylesheet" href="stylesheet.css"/>

<title>Result</title>

</head>

<body>

<p>I want to be SIZE 44 font!</p>

</body>

</html>

Syntax for the wintax

CSS syntax is different from the HTML you're used to, but don't worry: it's easy to pick up! The general format looks like this:

selector {

property: value;

}

A selector can be any HTML element, such as <p>, <img>, or <table>. You just take off the <>s! To make a paragraph's text red with CSS, you'd type:

p {

color: red;

}

A property is an aspect of a selector. For instance, you can change the font-family, color, and font-size of the text on your web pages (in addition to many more).

A value is a possible setting for a property. color can be red, blue, black, or almost any color; font-family can be a whole bunch of different fonts; and so on.

You need to end each property-value with a semi-colon (;). That's how CSS knows you're done with one pair and ready for the next.

The CSS background properties are used to define the background effects for elements.

CSS background properties:

* background-color
* background-image
* background-repeat
* background-attachment
* background-position

each selector has to have its own {}

/\* adding comments to css \*/

Hexadecimal counting is **base-16**. Each digit can be the numbers 0 through 9**or the letters a through f**! Crazy, right? Check it out:

You don't have to jump straight to a default value like cursive or sans-serif: you can tell CSS to try several, going from one to the next if the one you want isn't available.

One selector to rule them all

There's also a very special selector you can use to apply CSS styling to every element on the page: the \* selector. For example, if you type

\* {

border: 2px solid black;

}

To target all paragraph text, you can use the p selector like this:

p {

/\* enter your styles here \*/

}

To target all paragraph text inside div tags, you can use the div p selector like this:

div p {

/\* enter your styles here \*/

}

To target every HTML element, you can use the universal selector like this:

\* {

/\* enter your styles here \*/

}

Because saying "<p>s inside <div>s" is more specific than "all <p>s," the paragraphs outside divs will turn teal and those inside divs will turn deep red. This behavior is called cascading, and we'll learn about it in the next section!

Classes are assigned to HTML elements with the word class and an equals sign, like so:

<div class="square"></div>

<img class="square"/>

<td class="square"></td>

Classes are identified in CSS with a dot (.), like so:

.square {

IDs are assigned to HTML elements with the word id and an equals sign:

<div id="first"></div>

<div id="second"></div>

<p id="intro"></p>

IDs are identified in CSS with a pound sign (#):

IDs are assigned to HTML elements with the word id and an equals sign:

<div id="first"></div>

<div id="second"></div>

<p id="intro"></p>

IDs are identified in CSS with a pound sign (#):

First child

Another useful pseudo-class selector is first-child. It's used to apply styling to only the elements that are the first children of their parents. For instance:

p:first-child {

color: red;

}

Nth child

Well done! You can actually select any child of an element after the first child with the pseudo-class selector nth-child; you just add the child's number in parentheses after the pseudo-class selector. For example,

p:nth-child(2) {

color: red;

}

Would turn every paragraph that is the second child of its parent element red.

The element that is the child goes before :nth-child; its parent element is the element that contains it.



You can also set an element's margins all at once: you just start from the top margin and go around clockwise (going from top to right to bottom to left). For instance,

margin: 1px 2px 3px 4px;

id #

class .

If you read the first exercise's CSS carefully, you'll see we've snuck in a little trick for you: the z-index property. You can think of the z-index as a measure of importance: the higher an element's z-index, the higher it will be "stacked" on the page. Giving your header a z-index of 1 while not giving any of the other elements a z-index ensures that your header will sit "on top of" your other elements and won't get stuck behind them.