CSS Basics

Not Completed

Table Of Contents

Last updated : April, 2nd 2021

What You Will Learn

- Starting CSS from scratch
- The concept of Colors and Backgrounds in CSS.
- Fonts and Text
- Widths and Heights in CSS
- Margin, Padding
- Floating and Display.
- Links and Lists.
- Classes IDs

Useful Resources

- Basic Introduction to CSS

- 35+ HTML & CSS Resources for Beginners
- Learn Layout

CSS Basics

1. What Is CSS

CSS stands for **C**ascading **S**tyle **S**heets. While HTML defines the structure of a web page. CSS describes how elements are displayed on the screen. HTML represents content, and CSS represents the appearance of the content.

---

2. CSS Syntax

A CSS rule-set consists of a selector and some declarations.

The selector points to the HTML element you want to style. It can be a *html tag*, *class* or *id*.

Declarations are defined in a declaration block, surrounded by curly brackets `{}`, each declaration includes a CSS property name and value in this format `name: value`, and ends with a semicolon `;`.

You can also add comments, surrounded by `/*` and `*/`.

For example:

```
h1 {                            /* Selector */
    color: blue;                    /* Declaration 1 */
    font-size: 12px;                /* Declaration 2 */
}
```

---

3. CSS Stylesheets

We can write CSS in three different places.

First, it can be written in any HTML element,

For example :

```
<h1 style="color:blue; font-size:12px;">Hello World</h1>
```

A more readable way is to write CSS in a **stylesheet**; a **stylesheet** is a text file with the `.css` extension. The browser will read this text file, and the webpage will apply the style.

You need to specify the path of your **stylesheet** in your HTML code if you want it to be executed:

```
<html>
    <head>
        <link rel="stylesheet" href="/path/to/stylesheet.css"></link>
    </head>
    <body>
        <h1>Hello world</h1>
    </body>
</html>
```

CSS can also be written directly on the HTML page, but that method is deprecated.

---

4. CSS Simple Selectors

CSS selectors are used to “find” (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)

- Combinator selectors : Select elements based on a specific relationship between them
- Pseudo-class selectors : Select elements based on a certain state
- Pseudo-elements selectors : Select and style a part of an element
- Attribute selectors: Select elements based on an attribute or attribute value)

Element Selector

Element selector selects HTML elements based on the element name (tag).

For example, this style will be applied on every `<p>` tag:

```
p {
    color: red;
}
```

HTML Id And Class

HTML elements can have a `class` attribute, used to define similar styles for elements with the same class name. Elements can also have a unique `id` attribute.

Id Selector

The **id selector** uses the `id` attribute of an HTML element. To use it, write a hash `#` character, followed by the element's id.

For example, to style this paragraph `<p id="welcome">Hello World</p>`, use:

```
#welcome{
    color: red;
}
```

Class Selector

While the **class selector** uses the `class` attribute of the elements, to use it, write a period `.` followed by the class of the element.

For example, to style this image `<img class="netfliximg" src="netflix-pic"/>`, use:

```
.netfliximg{
    width: 200px;
    height: 200px;
}
```

Universal Selector

Universal selector is also available in CSS (and in a lot of computer languages), `*` means "Everything."

This style will be applied on every HTML elements of the page:

```
* {
    color: blue;
}
```

Grouping Selector

You can style more than one element at the time; just separate each selector with a comma `,`:

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

---

5. CSS Combinators Selectors

Descendant Selector

The descendant selector matches **all descendants** of a specified element (meaning they are nested in it).

The following example selects all `<p>` elements inside `<div>` elements:

```
div p {
  background-color: yellow;
}
```

Child Selector

The child selector selects all elements that are the **direct descendant** of a specified element.

The following example selects all `<p>` elements that are children of a `<div>` element:

```
div > p {
  background-color: yellow;
}
```

Sibling Selector

The general **sibling selector** selects all elements that are siblings of a specified element.

Sibling means “that have the same parent”.

The following example selects all `<p>` elements that are siblings of `<div>` elements:

```
div ~ p {
  background-color: yellow;
}
```

Adjacent Sibling Selector

The **adjacent sibling selector** selects all elements that are the adjacent siblings of a specified element.

“Adjacent” means “immediately following”.

The following example selects all `<p>` elements that are placed immediately after `<div>` elements:

```
div + p {
  background-color: yellow;
}
```

Exercise 1

Create a stuctured HTML file

Copy this paragraph and style it with Inline CSS first and then with a CSS stylesheet.

Hello Everyone, Congratulation on your first day of coding at DevelopersInstitute

What happens if you style this paragraph both with Inline CSS and with a CSS stylesheet ?

6. CSS Properties

Colors

In CSS, colors can be Hexadecimal Color Codes (`#ffffff`), rgb (`rgb(255,255,255)`), or color names (`black`).

Backgrounds

You can use CSS to add a background to your sections; it can be a `background-color` or a `background-image`.

`background-image` needs a URL to work.

```
body{
    background-image: url("my_img.png")
}
```

Borders

The CSS `border` properties allow you to specify the `border-style`, `border-width`, and `border-color` of an element's border.

```
p {
  border-style: solid;
  border-width: 5px;
  border-color: green;
}
```

Margins

The CSS `margin` properties are used to create space **around elements**, **outside** of any defined borders.

You can specify the width of `margin` for every side, but you can also set the margin for each element's side.

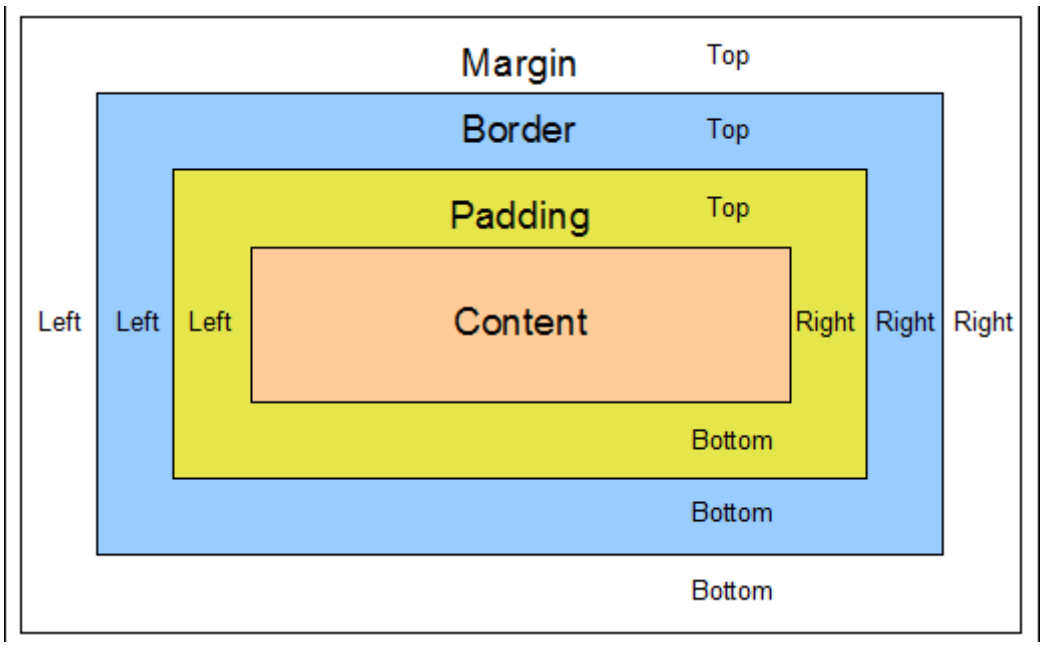
```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

Padding

The CSS `padding` properties are used to generate space **around an element's content**, **inside** of any defined borders.

It works the same way as `margin`.

See the box model to understand `padding` and `margin`



Elements Dimensions

The `height` and `width` properties are used to set an element's height and width.

Text Styling

You can specify the `color` of the text, but also the alignment (with `text-align`), the `text-decoration`, the `text-transformation`, the `text-indent` and much more

```
p {
    color: blue;
    text-align: justify;
    text-indent: 50px;
    text-decoration: underline;
    text-transform: uppercase;
}
```

List Styling

You can modify the icon in front of each list elements, with `list-style-type` to use a built in icon, or with `list-style-image` to add an image of your own.

Exercise 2

On the same HTML file

Recreate the green square of the Exercise 1, using margin, padding and borders

Try different types of borders, and shadows (Check it on Google)

Elements Display

You can modify the way elements are displayed.

The `display` property is controlling the layout of an element.

- Block-level: `display: block;` will display the element as a block, meaning stretch it out to the left and right as far as possible, and the next element will be placed below it.
- Inline: `display: inline;` elements don't start on a new line and only take up as much width as necessary.
- Inline block: `display: inline-block` allows to set a width and height on the element and respect the padding, but doesn't add a line break after the element.

- None: `display: none` won't display the element. **Careful** : It will not hide it (meaning it won't keep a blank place for the element). To hide an element, use `visibility: hidden`

See

This Course on CSS Layout

Example with squares

Example within texts

Exercise 3

On the same HTML file

Recreate this flag with the **display property**

You can look at these links to help you :

- css-display-property
- Understanding CSS Display: None, Block, Inline and Inline-Block

---

Exercise 5

On the same HTML file

Copy this code and do the following steps :

```
<div>
    <ul>
        <li>What is Lorem Ipsum?</li>
        <li>Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, </li>
       <li>when an unknown printer took a galley of type and scrambled it to make a type specimen b
    </ul>
</div>
<div>
    <p>It has survived not only five centuries, but also the leap into electronic typesetting,</p>
    <ol>
        <li> remaining essentially unchanged. It was popularised in the 1960s with the release of L
    </ol>
    <p> containing Lorem Ipsum passages, and more recently with desktop publishing software like Al
</div>
```

1.Change the second li of the 1st div to blue

2.Make all the li of the first div in uppercase and in bold

3.Change the 1st paragraph of the second div so its background color will be pink and its color white

4.Make the 2nd paragraph of the second div with a background color light blue, with padding and margin

Bonus: Try to find a few ways to achieve each of the steps