

CSS IN DEPTH

BY HAJJAT HADIJAH KYAMPEIRE

CSS SYNTAX

CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

A CSS rule consists of a selector and a declaration block

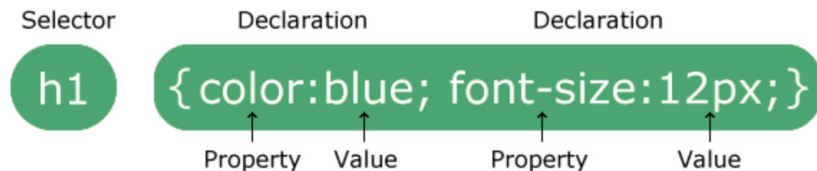
The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon. Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

In this example all `<p>` elements will be center-aligned, with a red text color:

```
p {  
  color: red;  
  text-align: center;  
}
```

CSS Syntax



CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

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)

The element selector selects HTML elements based on the element name.

Example: Here, all <p> elements on the page will be center-aligned, with a red text color:

```
p { text-align: center; color: red; }
```

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element! To select an element with a specific id, write a hash (#) character, followed by the id of the element. **Note: An id name cannot start with a number!**

The CSS class Selector

The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name. You can also specify that only specific HTML elements should be affected by a class.

In this example only `<p>` elements with `class="center"` will be red and center-aligned:

```
p.center { text-align: center; color: red; }
```

HTML elements can also refer to more than one class.

In this example the `<p>` element will be styled according to `class="center"` and to `class="large"`:

```
<p class="center large">This paragraph refers to two classes.</p>
```

The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page. The CSS rule below will affect every HTML element on the page:

```
* {text-align: center;color: blue;}
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1 {ext-align: center; color: red;}
```

```
h2 {text-align: center;color: red;}
```

```
p {text-align: center;color: red;}
```

It will be better to group the selectors, to minimize the code. To group selectors, separate each selector with a comma.

```
h1, h2, p {text-align: center;color: red;}
```

CSS Margins (We have already covered Css colors, backgrounds, and borders. Read more about them.)

Margins are used to create space around elements, outside of any defined borders.

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

All the margin properties can have the following values:

- `auto` - the browser calculates the margin
- *length* - specifies a margin in px, pt, cm, etc.
- `%` - specifies a margin in % of the width of the containing element
- `inherit` - specifies that the margin should be inherited from the parent element

Tip: Negative values are allowed. NB: To shorten the code, it is possible to specify all the margin properties in one property. Top-right-bottom-left. E.g `p { margin: 25px 50px 75px 100px; }`

Margins con't

The auto Value

You can set the margin property to **auto** to horizontally center the element within its container. The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

```
div { width: 300px; margin: auto; border: 1px solid red; }
```

The inherit Value

This example lets the left margin of the `<p class="ex1">` element be inherited from the parent element (`<div>`):

```
div { border: 1px solid red; margin-left: 100px; }
```

```
p.ex1 { margin-left: inherit; }
```

Css Padding

Padding is used to create space around an element's content, inside of any defined borders.

CSS has properties for specifying the padding for each side of an element:

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

All the padding properties can have the following values:

- *length* - specifies a padding in px, pt, cm, etc.
- *%* - specifies a padding in % of the width of the containing element
- *inherit* - specifies that the padding should be inherited from the parent element

Note: Negative values are not allowed.

To shorten the code, it is possible to specify all the padding properties in one property.

```
div {padding: 25px 50px 75px 100px; }
```


Css Height and Width

CSS height and width Values

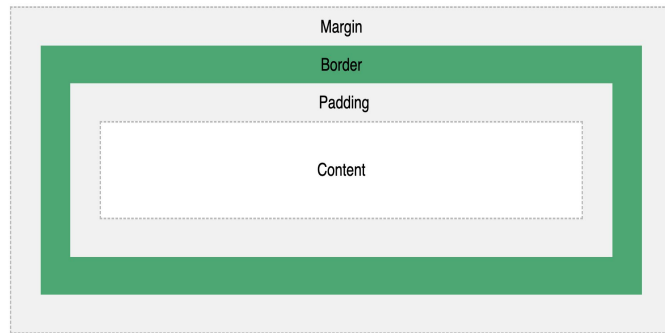
The **height** and **width** properties may have the following values:

- **auto** - This is default. The browser calculates the height and width
- **length** - Defines the height/width in px, cm, etc.
- **%** - Defines the height/width in percent of the containing block
- **initial** - Sets the height/width to its default value
- **inherit** - The height/width will be inherited from its parent value

The CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins. The image below illustrates the box model:



Css outline

CSS Outline Style: The **outline-style** property specifies the style of the outline, and can have one of the following values:

- **dotted** - Defines a dotted outline
- **dashed** - Defines a dashed outline
- **solid** - Defines a solid outline
- **double** - Defines a double outline
- **groove** - Defines a 3D grooved outline
- **ridge** - Defines a 3D ridged outline
- **inset** - Defines a 3D inset outline
- **outset** - Defines a 3D outset outline
- **none** - Defines no outline
- **hidden** - Defines a hidden outline

CSS Outline Width: The **outline-width** property specifies the width of the outline, and can have one of the following values:

- thin (typically 1px)
- medium (typically 3px)
- thick (typically 5px)
- A specific size (in px, pt, cm, em, etc)

CSS Outline Color

The **outline-color** property is used to set the color of the outline. The color can be set by:

- name - specify a color name, like "red"
- HEX - specify a hex value, like "#ff0000"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- HSL - specify a HSL value, like "hsl(0, 100%, 50%)"
- invert - performs a color inversion (which ensures that the outline is visible, regardless of color background)

CSS Outline - Shorthand property

The **outline** property is a shorthand property for setting the following individual outline properties:

- **outline-width**
- **outline-style** (required)
- **outline-color**

The **outline** property is specified as one, two, or three values from the list above. The order of the values does not matter.

p.ex1 {**outline**: dashed;} p.ex2 {**outline**: dotted red;} p.ex3 {**outline**: 5px solid yellow;} p.ex4 {**outline**: thick ridge pink;}

Next in Css

Css text, icons, links, lists, display, position, overflow, displays, Flexbox and grid