

Cascading Style Sheets

[3.0 Introduction](#)

[3.1 Selectors & Declarations](#)

[3.2 Font Properties](#)

[3.3 Color Properties](#)

[3.4 Lists](#)

[3.5 Alignment of Text](#)

[3.6 Background Images](#)

[3.7 The Box Model](#)

[3.7.1 Borders](#)

[3.7.2 Margins and Padding](#)

3.0 Introduction

WHY

CSS stands for Cascading Style Sheets. While HTML is used to add meaning and structure to a web page's content, CSS describes *how* that content should be displayed. This "how" is commonly referred to as the page's *style*.

Although styling and layout can be done using HTML (tables for layout, font elements for style, etc.), using Cascading Style Sheets is a far better option, because it separates code to add meaning (HTML) from code for presentation (CSS).

HOW

There are three ways to use CSS:

- **Inline**
 - Inline style sheets apply to the content of a single element. They appear within the opening and closing of a tag and apply only to the content of that particular tag.
 - Inline style specifications appear as values of the *style* attribute of a tag like this:

```
<p style="color: red;">Welcome to CSS!</p>
```

Example

```
<!DOCTYPE html>
<html>

<head>
<title>My page</title>
</head>

<body>
<p style="color:red">text</p>
</body>

</html>
```

- This is usually a bad way to apply styles, because it mixes up CSS with HTML and can be hard to maintain.

- **Document level**

- Document level style sheets apply to the whole of a document.
- Add a style element in the <head> of a document like this:

```
<style>
p {
color: red;
}
</style>
```

Example

```
<!DOCTYPE html>
<html>

<head>
<title>My page</title>

<style type="text/css">
p {
    color: red;
}
</style>
</head>

<body>
<p>text</p>
</body>

</html>
```

- This separates CSS from HTML code and keeps all CSS rules in one place — which is good! However, with this method CSS code is still included in HTML files and can't be shared between documents.

- **External**

- External style sheets can apply to any number of documents.
- The <link> tag is used to specify external style sheets. Add a link tag to the <head>:

```
<head>
...
<link rel="stylesheet" href="/css/global.css">
...
</head>
```

- You can link to more than one style sheet:

```
<head>
...
<link rel="stylesheet" href="/css/global.css">
<link rel="stylesheet" href="css/local.css">
...
</head>
```

Example

```
<!DOCTYPE html>
<html>

<head>
<title>My page</title>
<link rel="stylesheet" href="my_styles.css">
</head>

<body>
<p>text</p>
</body>

</html>
```

- Using CSS from an external file means that CSS code is kept separate from HTML. That makes CSS easier to understand and maintain and enables you to share styles between documents — and enables caching by the browser, which only needs to load each CSS file once.

TIP: Every browser has a default style sheet: for example, to specify heading sizes, and the default margin before and after paragraph elements. You can find a complete list of browser style sheets [here](#).

GLOSSARY

Cascading Style Sheets are best used to define an order of precedence when conflicting style elements may exist.

3.1 Selectors & Declarations

HOW

Each CSS rule has 3 parts:

1. Selector
2. Properties
3. Values

The selector indicates which elements are affected by the rule.

A property ('color') and a value ('red') make up a declaration.

For example, you can make the text of all `p` elements red with the following CSS rule:

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

Here, `p` is the selector.

`color` is the property. The property is always followed by a colon (:)

`red` is the value. The value is always followed by a semicolon (;)

You can add multiple declarations to each rule:

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

Put each declaration on a separate line to make them easier to read:

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

TIP: We recommend adding a semicolon to every declaration, even if there's only one, to avoid problems if you add more later.

HOW

Selectors can have a variety of forms:

TRY IT

1. Element name

The property value applies to all occurrences of the named elements.

```
h1 {color: red;}
```

You can apply the same values to multiple elements:

```
h2, h3 {color: green;}
```

2. Class Selector

Select elements with class attributes like this:

HTML:

```
<p class="highlight">Uh oh!</p>
```

CSS:

```
p.highlight {  
  color: red;  
  background-color: yellow;  
}
```

Leave out the element name to apply values to multiple elements with the same class:

HTML:

```
<p class="highlight">Uh oh!</p>
```

...

```
<blockquote class="highlight">Some crazy  
quote.</blockquote>
```

CSS:

```
.highlight {  
  color: red;  
  background-color: yellow;  
}
```

3. ID Selector

Apply a style to a single element.

```
#intro {  
  font-size: 18pt;  
}
```

```
<p id="intro">This will be in 18pt font size.</p>
```

4. Universal Selectors

The universal selector applies to all elements in the document and is denoted by an asterisk (*).

```
* {  
    color: yellow;  
}
```

WARNING: the universal selector is not recommended. It can reduce performance and make CSS difficult to debug since the rule is applied to every element.

GLOSSARY

Selectors are used to find the HTML elements that you want to style. This could be the element name, id or class name: `h1`, `#intro`, `.warning`, etc.

Declarations have two parts, property and value: `color: red`, for example.

TIP: Don't forget to use comments to increase the readability of your CSS code! Comments are written in this format:

```
/* This is a comment, the format does not matter! */
```

3.2 Font Properties

WHY

In any web document, you will notice a variation in font size, font family (Arial, Helvetica etc), font style (**Bold**, *Italic*, Underline), and other font properties.

TRY IT

1. Specify fonts in order of preference

```
font-family: Arial, Helvetica, Futura, sans-serif
```

2. Specify text size

```
font-size: 14pt
```

3. Italicise text:

```
font-style: italic
```

4. Specify bold text:

```
font-weight: bold
```

Shorthand properties

You can specify more than one value by using the `font` property:

```
font: bold 12pt Arial
```

GLOSSARY

Font-family refers to the typeface (Helvetica, Arial etc).

Font-size refers to the size of the text.

Font-style refers to the style of the text (*italicised*, **bold** or normal).

3.3 Color Properties

WHY

Specify color using one of the 16 named colors: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. Instead of using color names, you can use numerical values to specify the amount of red, green and blue to use. The following three rules all have the same effect. The first uses decimal numbers and the second uses hexadecimal numbers: FF means 255 in hexadecimal (base 16).

```
h1 {color: rgb(255, 0, 0);}
h1 {color: #FF0000;}
h1 {color: red;}
```

You can also set the background color of an element. In the following example, the text will be red and the background color of the H1 element will be yellow.

```
h1 {
  background-color: yellow;
  color: red;
}
```

TRY IT

```
p.bright {color: red; background-color: yellow}
<p> This will be in red with a yellow background. </p>

h1 {
  color: red;
}
<h1>This will be red in color.</h1>
```

GLOSSARY

Color refers to the color of an element. For elements that contain text, this means the foreground text color. The value could be the name of the color (e.g. `black`, `red`) or the hexadecimal or decimal value of the color (e.g. `#ff0000` or `rgb(255, 0, 0)`).

Background-color refers to the background color of an element.

3.4 Alignment of Text & Position

Alignment of Text

WHY

You can align text to the left (the default) or to the right, center text, or use justified text (with a straight edge on both left and right). You can also indent the first line of a block of text. CSS offers a number of units for expressing length which defines how your text looks on output media like your screen or on paper. You can use pixel (px), centimeter (cm) or even inch (in).

HOW

Align text using the `text-align` property; indent text using `text-indent`

TRY IT

```
<style>
    .body {
        text-align: justified;
        text-indent: 10px; }
</style>

<p class="body">
    This paragraph will start 10 pixels away from the margin. The
    next line will start at the margin as usual and all the text in
    the paragraph will be justified, which means it will have a
    straight edge on both left and right.
</p>
```

Position

WHY

The `position` property specifies the type of positioning method used for an element. There are 4 types:

1. Static

The element is positioned according to the normal flow of the page. This is the default.

2. Fixed

The element stays in the same place, even when the page is scrolled.

3. Relative

The element is positioned relative to its normal position.

4. Absolute

The element is positioned relative to the nearest positioned ancestor (element with a specified `position` value). If it has no positioned ancestors, it used the document body and still moves along with page scrolling.

TRY IT


```
<style>
    .box1 {
        position: relative;
        left: 10px; }
    .box2 {
        position: fixed;
        bottom: 20px; }
</style>

<p class="box1">
    This paragraph will start 10 pixels away from the
margin.    </p>
<p class="box2">
    This paragraph will start 20 pixels away from the margin
and will always remain in the same place even if the page is
scrolled.
    </p>
```

GLOSSARY

Text-indent refers to the amount of empty space to the left of the first line of a block of text.

Text-align refers to the alignment of each a whole block of text.

3.5 Background Images

WHY

The `background-image` property is used to set an image as the background of an element.

HOW

Use the `background-image` property with a file path:

```
h1 {background-image: url(/images/gradient.png);}
```

TIP: The path for a background image is relative to the document that contains the rule, so if you use `background-image: url(gradient.png)` in a CSS file at `/css/global.css`, the browser will look for `gradient.png` in the same CSS directory.

TRY IT

Apply a background image to the body element of your web document.

3.6 The Cascade

WHY

Sometimes, more than one stylesheet rule could apply to a particular CSS element. For example, one rule could be to make all elements in a `<p>` tag pink, while another could make it blue. In cases like this, there has to be a way of determining which stylesheet rule applies to the element.

The cascade essentially determines which declarations apply to any element.

HOW

Since one element could potentially have one or more declarations applied to it, 'Cascading' is a method used to determine which one finally is finally applied. In the case conflicting declarations, the last and most specific declaration is used.

TRY IT

```
<style>
    div {
        color: blue;
        font-style: italic; }
    p {
        color: pink; }
</style>

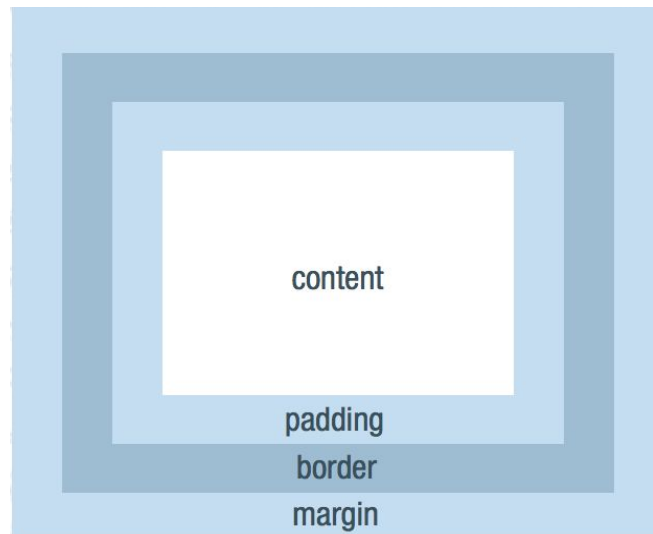
<div>
<p> This will be in Pink color. </p>
</div>
```

3.7 The Box Model

WHAT

In a document, each element is represented using a rectangular box. This is called 'The Box Model'. It essentially is a box that wraps around every HTML element. It consists of 4 edges: margins, borders, padding and content.

The image below illustrates the box model:



Content is the area containing the real content of the HTML element: text or an image, for example.

Padding is the area around the content.

Border is the... border! This goes around the padding. It can have properties like border-style, border-width and border-color.

Margin is the empty space between two neighbouring elements: on the top, right, bottom or left.

3.7.1 Borders

WHY

The property which controls whether the element has a border, as well as the style of the border, is called border properties.

These borders have various features like style, color, width etc.

HOW

Each side of the border (top, bottom, left, right) can be different in terms of style, width or color. Borders have 3 key properties:

- border-style
 - This property determines the style of the border, whether it is *dotted*, *dashed* or *double*.
 - The default value for border-style is *none*.
 - The styles of one of the four sides of an element can be set with `border-top-style`, `border-bottom-style`, `border-left-style` and `border-right-style`.
- border-width
 - This property determines how thick or thin the border is. Its possible values are *thin*, *medium*, *thick* or a *length value in pixels*.
 - The default value for border-width is *medium*.

- The thickness of one of the four sides of an element can be set with `border-top-width`, `border-bottom-width`, `border-left-width` and `border-right-width`.
- **border-color**
 - This property determines the color of the border. It can take named colors or hexadecimal values.
 - The default colour is *black*.
 - The color of one of the four sides of an element can be set with `border-top-color`, `border-bottom-color`, `border-left-color` and `border-right-color`.

TRY IT

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

boxmodel.css:

```
table {
    border-top-width: thin;
    border-bottom-width: thick;
    border-top-color: red;
    border-bottom-color: yellow;
    border-top-style: dotted;
}

p {
    border-width: thick;
    border-color: aqua;
    border-style: dashed;
}
```

Now add a table and a para element, save and execute. You should see the border properties in the output.

GLOSSARY

Border is the area that goes between the padding and the margin.

Border-style refers to the way the border appears (or looks): a dotted or dashed line, for example

Border-width refers to the thickness of the border.

Border-color refers to the color of the border.

3.7.2 Margins and Padding

WHY

The amount of space between the content of an element and its border is called padding. The space between the border of an element and its immediate neighbour (to the top, right, bottom or left) is called margin.

HOW

Margin properties are called *margin* and padding properties are called *padding*. These properties may be applied to the four sides of an image: top, bottom, left and right.

So, the margin properties of an element would be: `margin-top`, `margin-bottom`, `margin-left` and `margin-right`; while the padding properties would be `padding-top`, `padding-bottom`, `padding-left` and `padding-right`.

Values for margin and padding may be given in inches or pixels (or any other size unit).

TRY IT

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

margins.css:

```
p.one {
    margin: 0.1in;
    padding: 0.1in;
    background-color: red;
}

p.two {
    margin: 0.2in;
    padding: 0.2in;
    background-color: blue;
}

p.three {
    margin: 20px;
    padding: 10px;
    background-color: yellow;
}

p.four {
    margin: 40px;
    background-color: red;
}
```

Now add 4 para elements with class names as *one*, *two*, *three* and *four* respectively; and execute. You should see the margin and padding properties in the output.

Please remember to save your work before moving on to the next section.

GLOSSARY

Margin refers to the space between the content of the element and the element's neighbour.

Padding refers to the space between the element and its border.