



CSS



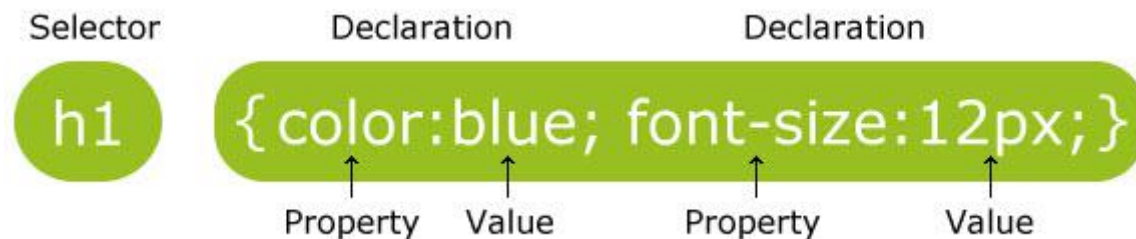
CSS is used to control the style of a web document in a simple and easy way.

CSS stands for **C**ascading **S**tyle **S**heets

CSS handles the look and feel part of a web page.

Styles are normally saved in external .css files.


A CSS rule set 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 property name and a value, separated by a colon.



A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly braces:

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

CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date.

Comments are ignored by browsers.

A CSS comment starts with `/*` and ends with `*/`.

Comments can also span multiple lines:



CSS Selectors

CSS selectors allow you to select and manipulate HTML element(s).

CSS selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more.

Mainly we use 3 types of selector:


- Element selector

- Id selector

- Class selector

The element Selector

The element selector selects elements based on the element name.



You can select all `<p>` elements on a page like this: (all `<p>` elements will be center-aligned, with a red text color)

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

Result:

Every paragraph will be affected by the style.

Me too!

And me!



The id Selector

The id selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the id selector when you want to find a single, unique element.

To find an element with a specific id, write a hash # character, followed by the id of the element.

Example

The style rule below will be applied to the HTML element with id="para1":



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#para1 {
```

```
    text-align: center;
```

```
    color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p id="para1">Hello World!</p>
```

```
<p>This paragraph is not affected by the  
style.</p>
```

```
</body>
```

```
</html>
```

Result:

Hello World!

This paragraph is not affected by the style.



The class Selector

The class selector finds elements with the specific class.

The class selector uses the HTML class attribute.

To find elements with a specific class, write " . "symbol, followed by the name of the class:

In the example below,
all HTML elements with class="center" will be center-aligned:



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.center {
```

```
    text-align: center;
```

```
    color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1 class="center">Red and center-aligned heading</h1>
```

```
<p class="center">Red and center-aligned paragraph.</p>
```

```
</body>
```

```
</html>
```

Result:

Red and center-aligned heading

Red and center-aligned paragraph.



Grouping Selectors


In style sheets there are often elements with the same style:

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

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

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

To minimize the code, you can group selectors.



To group selectors, separate each selector with a comma.
In the example below we have grouped the selectors from the code above:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

Result

Hello World
Smaller heading!
This is a paragraph.



CSS are classified into three;

External stylesheet

Internal stylesheet

Inline stylesheet

External Style Sheet

An external style sheet is ideal when the style is applied to many pages.

With an external style sheet, you can change the look of an entire Web site by changing just one file.

Each page must include a link to the style sheet with the `<link>` tag.

The `<link>` tag goes inside the head section:



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

An external style sheet can be written in any text editor.
The file should not contain any html tags.
The style sheet file must be saved with a .css extension.
An example of a style sheet file called "myStyle.css", is shown below:

```
body {  
    background-color: lightblue;  
}  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, inside the <style> tag, like this:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: linen;
      }
      h1 {
        color: maroon;
        margin-left: 40px;
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

Result

This is a heading

This is a paragraph.



Inline Styles

An inline style loses many of the advantages of a style sheet (by mixing content with presentation).

To use inline styles, add the style attribute to the relevant tag.

The style attribute can contain any CSS property.

The example shows how to change the color and the left margin of a h1 element:

Example

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

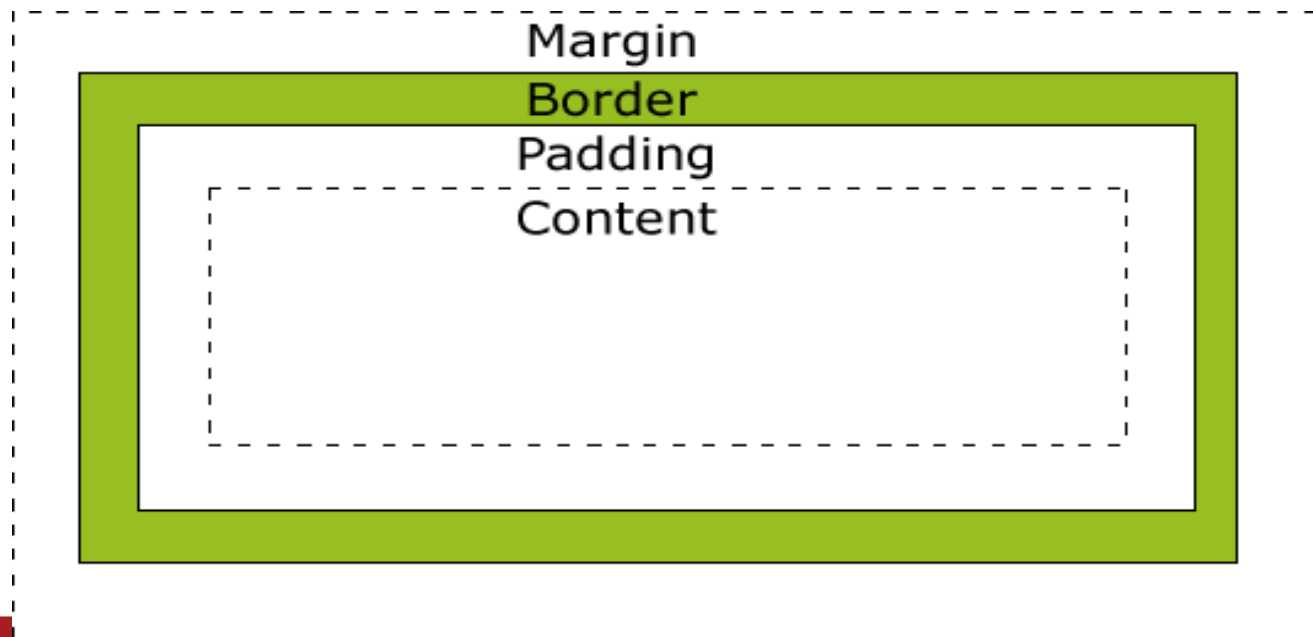
This is a heading.

This is a paragraph.

An inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet.

The CSS Box Model

All HTML elements can be considered as boxes.





The CSS box model is essentially a box that wraps around every HTML element.

It consists of: margins, borders, padding, and the actual content.

Explanation of the different parts:

Content - The content of the box, where text and images appear

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content.

Margin - Clears an area outside the border. The margin is transparent



Margin

The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent.

The top, right, bottom, and left margin can be changed independently using separate properties.

A shorthand margin property can also be used, to change all margins at once.



The margin property can have from one to four values.

margin: 25px 50px 75px 100px;

top margin is 25px

right margin is 50px

bottom margin is 75px

left margin is 100px

margin: 25px 50px 75px;

top margin is 25px

right and left margins are 50px

bottom margin is 75px

margin: 25px 50px;

top and bottom margins are 25px

right and left margins are 50px

margin: 25px;

all four margins are 25px

Margin - Individual sides

In CSS, it is possible to specify different margins for different sides of an element:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  background-color: yellow;
}
```

```
p.ex {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 50px;
}
</style>
</head>
<body>
```

```
<p>This is a paragraph with no specified margins.</p>
<p class="ex">This is a paragraph with specified margins.</p>
```

```
</body>
</html>
```

Result

This is a paragraph with no specified margins.

This is a paragraph with specified margins.

Margin - Shorthand property

To shorten the code, it is possible to specify all the margin properties in one property. This is called a shorthand property.

The shorthand property for all the margin properties is "**margin**":

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  background-color: yellow;
}
```

Result

This is a paragraph with no specified margins.

```
p.ex {
  margin: 100px 50px;
}
</style>
</head>
<body>
<p>This is a paragraph with no specified margins.</p>
<p class="ex">This is a paragraph with specified margins.</p>
</body>
</html>
```

This is a paragraph with specified margins.



Padding

The padding clears an area around the content (inside the border) of an element.

The padding is affected by the background color of the element.

The top, right, bottom, and left padding can be changed independently using separate properties.

A shorthand padding property can also be used, to change all paddings at once.



Border

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

Border Style

The border-style property specifies what kind of border to display.

The following values are allowed:

dotted - Defines a dotted border

dashed - Defines a dashed border

solid - Defines a solid border

double - Defines a double border

groove - Defines a 3D grooved border. The effect depends on the border-color value

ridge - Defines a 3D ridged border. The effect depends on the border-color value

inset - Defines a 3D inset border. The effect depends on the border-color value

outset - Defines a 3D outset border. The effect depends on the border-color value

none - Defines no border

hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).



Border Width

The **border-width** property specifies the width of the four borders.

The width can be set as a specific size (in **px**, **pt**, **cm**, **em**, etc) or by using one of the three pre-defined values: **thin**, **medium**, or **thick**.

Border Color

The **border-color** property is used to set the color of the four borders.

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

- border-width
- border-style (required)
- Border-color

example

```
p {  
  border: 5px solid red;  
}
```