

# I. What is CSS?

- **CSS** stands for **Cascading Style Sheets**
- CSS defines **how HTML elements are to be displayed**
- Styles were added to HTML 4.0 **to solve a problem**
- CSS saves a lot of work
- External Style Sheets are stored in **CSS files**

## CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a document.

HTML was intended to **define the content** of a document, like:

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

When tags like `<font>`, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

In HTML 4.0, all formatting could (and should!) be removed from the HTML document, and stored in a separate CSS file.

## CSS Saves a Lot of Work!

The style definitions are normally saved in external `.css` files.

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

## II. CSS Selectors

CSS selectors allow you to select and manipulate HTML elements.

CSS selectors are used to "find" (or select) HTML elements based on their id, class, type, attribute, and more.

### 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

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

### The id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

An id should be unique within a page, so the id selector is used if you want to select a single, unique element.

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

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

## Example

```
#para1 {  
    text-align: center;  
    color: red;  
}
```



Do **NOT** start an ID name with a number!

## The class Selector

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period character, followed by the name of the class:

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

## Example

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

You can also specify that only specific HTML elements should be affected by a class.

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

## Example

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



Do **NOT** start a class name with a number!

## Grouping Selectors

If you have elements with the same style definitions, like this:

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

you can group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

In the example below we have grouped the selectors from the code above:

## Example

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

When a browser reads a style sheet, it will format the document according to the information in the style sheet.

## III. Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

## External Style Sheet

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the `<link>` element. The `<link>` element 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;  
}
```



Do not add a space between the property value and the unit (such as `margin-left: 20 px;`). The correct way is: `margin-left: 20px;`

## Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the `<style>` element, inside the head section of an HTML page:

### Example

```
<head>  
<style>  
body {  
    background-color: linen;  
}  
  
h1 {  
    color: maroon;  
    margin-left: 40px;  
}  
</style>  
</head>
```

# Inline Styles

An inline style may be used to apply a unique style for a single element.

An inline style loses many of the advantages of a style sheet (by mixing content with presentation). Use this method sparingly!

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

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

# Multiple Style Sheets

If some properties have been defined for the same selector in different style sheets, the value will be inherited from the more specific style sheet.

For example, assume that an external style sheet has the following properties for the <h1> element:

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

then, assume that an internal style sheet also has the following property for the <h1> element:

```
h1 {  
    color: orange;  
}
```

If the page with the internal style sheet also links to the external style sheet the properties for the <h1> element will be:

```
color: orange;  
margin-left: 20px;
```

The left margin is inherited from the external style sheet and the color is replaced by the internal style sheet.

## Multiple Styles Will Cascade into One

Styles can be specified:

- in an external CSS file
- inside the <head> section of an HTML page
- inside an HTML element

### Cascading order

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number three has the highest priority:

1. Browser default
2. External and internal style sheets (in the head section)
3. Inline style (inside an HTML element)

So, 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, or in a browser (a default value).



**Note:** If the link to the external style sheet is placed below the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!



## 4. CSS Background

CSS background properties are used to define the background effects of an element.

CSS properties used for background effects:

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

### Background Color

The `background-color` property specifies the background color of an element.

The background color of a page is set like this:

#### Example

```
body {  
    background-color: #b0c4de;  
}
```

With CSS, a color is most often specified by:

- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"
- a color name - like "red"

In the example below, the `<h1>`, `<p>`, and `<div>` elements have different background colors:

## Example

```
h1 {  
  background-color: #6495ed;  
}  
  
p {  
  background-color: #e0ffff;  
}  
  
div {  
  background-color: #b0c4de;  
}
```

[Try it yourself »](#)

## Background Image

The `background-image` property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

## Example

```
body {  
  background-image: url("paper.gif");  
}
```

## Background Image - Repeat Horizontally or Vertically

By default, the `background-image` property repeats an image both horizontally and vertically.

## Example

```
body {  
  background-image: url("gradient_bg.png");  
  background-repeat: repeat-x;  
}
```



**Note:** To repeat an image vertically set `background-repeat: repeat-y;`

## Background Image - Set position and no-repeat



**Note:** When using a background image, use an image that does not disturb the text.

The position of the image is specified by the `background-position` property:

## Example

```
body {  
  background-image: url("img_tree.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
}
```

## Background - Shorthand property

As you can see from the examples above, there are many properties to consider when dealing with backgrounds.

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

The shorthand property for background is simply "`background`":

## Example

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

When using the shorthand property the order of the property values is:

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`

## 5. CSS Text

### Text Color

The `color` property is used to set the color of the text.

The default color for a page is defined in the body selector.

#### Example

```
body {  
    color: blue;  
}  
  
h1 {  
    color: #00ff00;  
}  
  
h2 {  
    color: rgb(255,0,0);  
}
```

### Text Alignment

The `text-align` property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

#### Example

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

```
p.date {  
    text-align: right;  
}  
  
p.main {  
    text-align: justify;  
}
```

## Text Decoration

The `text-decoration` property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

### Example

```
a {  
    text-decoration: none;  
}
```

It can also be used to decorate text:

### Example

```
h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}
```

# Text Transformation

The `text-transform` property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first

## Example

```
p.uppercase {  
    text-transform: uppercase;  
}  
  
p.lowercase {  
    text-transform: lowercase;  
}  
  
p.capitalize {  
    text-transform: capitalize;  
}
```

# Text Indentation

The `text-indent` property is used to specify the indentation of the first line of a text.

## Example

```
p {  
    text-indent: 50px;  
}
```

