

## Cascading Style Sheet

### ❖ What is CSS?

- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements
- Styles are normally stored in Style Sheets
- Styles were added to HTML 4.0 to solve a problem
- External Style Sheets can save you a lot of work
- External Style Sheets are stored in CSS files
- Multiple style definitions will cascade into one

### ❖ Styles Solve a Common Problem

- HTML tags were originally designed to define the content of a document. They were supposed to say "This is a header", "This is a paragraph", "This is a table", by using tags like `<h1>`, `<p>`, `<table>`, and so on. The layout of the document was supposed to be taken care of by the browser, without using any formatting tags.
- As the two major browsers - Netscape and Internet Explorer - continued to add new HTML tags and attributes (like the `<font>` tag and the color attribute) to the original HTML specification, it became more and more difficult to create Web sites where the content of HTML documents was clearly separated from the document's presentation layout.
- To solve this problem, the World Wide Web Consortium (W3C) - the non profit, standard setting consortium, responsible for standardizing HTML - created STYLES in addition to HTML 4.0.
- All major browsers support Cascading Style Sheets.

### ❖ Style Sheets Can Save a Lot of Work

- Styles sheets define HOW HTML elements are to be displayed, just like the font tag and the color attribute in HTML 3.2. Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in your Web, just by editing one single CSS document!
- CSS is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once. As a Web developer you can define a style for each HTML element and apply it to as many Web pages as you want. To make a global change, simply change the style, and all elements in the Web are updated automatically.

### ❖ Multiple Styles Will Cascade Into One

- Style sheets allow style information to be specified in many ways. Styles can be specified inside a single HTML element, inside the `<head>` element of an HTML page, or in an external CSS file. Even multiple external style sheets can be referenced inside a single HTML document.

## Cascading Style Sheet

### ❖ 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 four has the highest priority:

1. Browser default
2. External style sheet
3. Internal style sheet (inside the <head> tag)
4. Inline style (inside an HTML element)

### ❖ Syntax

```
<STYLE type="text/css" >
    selector {property: value ; property: value ; property: value ; ....}
    selector {property: value ; property: value ; property: value ; ....}
    selector {property: value ; property: value ; property: value ; ....}
    .....
    .....
</STYLE>
```

- The style assign a process accomplished with the <STYLE>...</STYLE>tags.
- The <STYLE></STYLE> HTML tags are written within the <HEAD>...</HEAD> tags.
- In the <STYLE> Tag, the expression "TYPE=text/css" indicate that the style sheet conform to CSS syntax.
- The CSS syntax is made up of three parts: a selector, a property and a value.
- The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change, and each property can take a value. The property and value are separated by a colon and surrounded by curly braces:

```
body {color: black}
```

- If the value is multiple words, put quotes around the value:

```
p {font-family: "sans serif"}
```
- If you wish to specify more than one property, you must separate each property with a semi-colon. The example below shows how to define a center aligned paragraph, with a red text color:

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

- To make the style definitions more readable, you can describe one property on each line, like this:

```
P {    text-align: center;
      color: black;
      font-family: arial
    }
```

## Cascading Style Sheet

- You can group selectors. Separate each selector with a comma. In the example below we have grouped all the header elements. All header elements will be green:

```
h1,h2,h3,h4,h5,h6
{
    color: green
}
```

- **The class Selector :** With the class selector you can define different styles for the same type of HTML element. Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph. Here is how you can do it with styles:

```
p.right {text-align: right}
p.center {text-align: center}
```

You have to use the class attribute in your HTML document:

```
<p class="right">
    This paragraph will be right-aligned.
</p>
<p class="center">
    This paragraph will be center-aligned.
</p>
```

- But Only one class attribute can be specified per HTML element! The example below is wrong:

```
<p class="right" class="center">
    This is a paragraph.
</p>
```

- You can also omit the tag name in the selector to define a style that will be used by all HTML elements that have a certain class. In the example below, all HTML elements with class="center" will be center-aligned:

```
.center {text-align: center}
```

In the code below both the h1 element and the p element have class="center". This means that both elements will follow the rules in the ".center" selector:

```
<h1 class="center">
    This heading will be center-aligned
</h1>
<p class="center">
    This paragraph will also be center-aligned.
</p>
```

- **The id Selector :** With the id selector you can define the same style for different HTML elements.
  - The style rule below will match any element that has an id attribute with a value of "green":

```
#green {color: green}
```

The rule above will match both the h1 and the p element:

```
<h1 id="green">Some text</h1>
<p id="green">Some text</p>
```

## Cascading Style Sheet

- The style rule below will match a p element that has an id with a value of "para1":

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

The style rule below will match any p element that has an id attribute with a value of "green":

```
p#green {color: green}
```

The rule above will not match an h1 element:

```
<h1 id="green">Some text</h1>
```

- **CSS Comments** : You can insert comments into CSS to explain your code, which can help you when you edit the source code at a later date. A comment will be ignored by the browser. A CSS comment begins with "/\*", and ends with "\*/", like this:

```
/* This is a comment */
p
{
    text-align: center;
    /* This is another comment */
    color: black;
    font-family: arial
}
```

### ❖ How to Insert a Style Sheet

- When a browser reads a style sheet, it will format the document according to it.
- There are three ways of inserting a style sheet:
  - **External Style Sheet**
  - **Internal Style Sheet**
  - **Inline Styles**
- **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 one file.
  - Each page must link to the style sheet using the <link> tag.
  - The <link> tag goes inside the head section:

```
<head>
    <link rel="stylesheet" type="text/css" href="mystyle.css" >
</head>
```
  - The browser will read the style definitions from the file mystyle.css, and format the document according to it.

## Cascading Style Sheet

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

```
hr {color: sienna}
p { color: black; font-family: arial }
```

- **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 by using the <style> tag, like this:

```
<head>
<style type="text/css">
    hr {color: sienna}
    p { color: black; font-family: arial }
</style>
</head>
```

- The browser will now read the style definitions, and format the document according to it.

- **Inline Styles**

- An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly, such as when a style is to be applied to a single occurrence of an element.
- To use inline styles you use the style attribute in 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 paragraph:

```
<p style="color: sienna; margin-left: 20px">
    This is a paragraph
</p>
```

- ❖ **Multiple Style Sheets**

- If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.
- For example, an external style sheet has these properties for the h3 selector:

```
h3
{ color: red; text-align: left; font-size: 8pt }
```

And an internal style sheet has these properties for the h3 selector:

```
h3
{ text-align: right; font-size: 20pt }
```

If the page with the internal style sheet also links to the external style sheet the properties for h3 will be:

```
color: red; text-align: right; font-size: 20pt
```

The color is inherited from the external style sheet and the text-alignment and the font-size is replaced by the internal style sheet.

## Cascading Style Sheet

### ❖ CSS Font Properties

- The CSS font properties allow you to change the font family, boldness, size, and the style of a text.
  - font-family property
  - font-style property
  - font-weight property
  - font-size property
- **font-family property**
  - The font-family property is a prioritized list of font family names and/or generic family names for an element.
  - The browser will use the first value it recognizes.
  - Separate each value with a comma
- **font-style property**
  - The font-style property sets the style of a font.
- **font-weight property**
  - The font-weight property sets how thick or thin characters in text should be displayed.
- **font-size property**
  - The font-size property sets the size of a font.

### Example

```
p
{
    font-family: arial, "lucida console", sans-serif;
    font-style: italic;
    font-weight: bold;
    font-size: 10px
}

body
{ font-size: x-large }
```

## Cascading Style Sheet

### ❖ CSS Text Properties

- The CSS text properties allow you to control the appearance of text.
- It is possible to change the color of a text, increase or decrease the space between characters in a text, align a text, decorate a text, indent the first line in a text, and more.
  - color property
  - text-align property
  - text-decoration property
  - text-transform property
  - word-spacing property
  - letter-spacing property
- **color property**
  - The color property sets the color of a text.
- **text-align property**
  - The text-align property aligns the text in an element.
- **text-decoration property**
  - The text-decoration property decorates the text.
- **text-transform property**
  - The text-transform property controls the letters in an element.

Value	Description
none	Defines normal text, with lower case letters and capital letters
capitalize	Each word in a text starts with a capital letter
uppercase	Defines only capital letters
lowercase	Defines no capital letters, only lower case letters

- **word-spacing property**
  - The word-spacing property increases or decreases the white space between words.
  - Negative values are allowed.
- **letter-spacing property**
  - The letter-spacing property increases or decreases the white space between characters.
  - Negative values are allowed.

### Examples

```
p
{ color: rgb(255,255,0);
  text-align: center;
  text-decoration: underline;
  text-transform: uppercase;
  word-spacing: 30px;
  letter-spacing: 12px }
```