CSS (Cascading Style Sheets)

Introduction

What is CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External stylesheets are stored in **CSS files**

CSS Solved a Big Problem

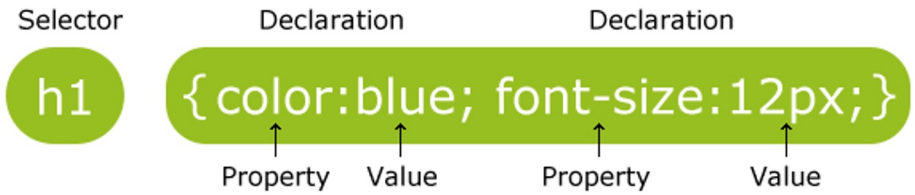
- HTML was NEVER intended to contain tags for formatting a web page!
- HTML was created to **describe the content** of a web page, like:
  - <h1>This is a heading</h1>
  - This is a paragraph.
- When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, 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.
- CSS removed the style formatting from the HTML page!

CSS Saves a Lot of Work!

- The style definitions are normally saved in external .css files.
- With an external stylesheet file, you can change the look of an entire website by changing just one file!

CSS Syntax

- 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 CSS property name and a value, separated by a colon.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

Example

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

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

CSS Selectors

- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, 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 (in this case, all <p> elements will be center-aligned, with a red text color):

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

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>
```

The id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element should be 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.
- The style rule below will be applied to the HTML element with id="para1":

Example

```
<!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>
```

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 red and center-aligned:

Example

```
<!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>
```

Referring to more than one class

- HTML elements can also refer to more than one class.
- In the following example, the <p> element will be styled according to class="center" and to class="large":

Example

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

		p.large {
		    font-size: 300%;
		}
	</style>
</head>
<body>
	<h1 class="center">This heading will not be affected</h1>
	<p class="center">This paragraph will be red and center-aligned.</p>
	<p class="center large">This paragraph will be red, center-aligned, and in a large font-size.</p>

</body>
</html>
```

Grouping Selectors

- If you have elements with the same style definitions, it will be better to group the selectors, to minimize the code.
- To group selectors, separate each selector with a comma.

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

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

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

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

Universal Selector

- There's also a very special selector you can use to apply CSS styling to every element on the page: the * selector.
- For example, if you type

```
* {
    border: 2px solid black;
}
```

You'll create a two-pixel wide solid black border around *every* element on the HTML page.

CSS Comments

- Comments are used to explain the code, and may help 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:

```
p {
    color: red;
    /* This is a single-line comment */
    text-align: center;
}

/* This is
a multi-line
comment */
```

Three Ways to Insert CSS

- 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>
```

External Style Sheet

- 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.
- Here is how the "myStyle.css" looks:

```
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    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:

```
<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.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.
- The example below shows how to change the color and the left margin of a <h1> element:

```
<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 (element) in different style sheets, the value from the last read style sheet will be used.

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
    color: orange;
}
</style>
</head>
```

```
<head>
<style>
h1 {
    color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

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 one has the highest priority:
  1. Inline style (inside an HTML element)
  2. External and internal style sheets (in the head section)
  3. Browser default
- So, an inline style (inside a specific 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 a browser default value.

CSS Colors

- Colors in CSS are most often specified by:
  - a valid color name - like "red"
  - an RGB value - like "rgb(255, 0, 0)"
  - a HEX value - like "#ff0000"

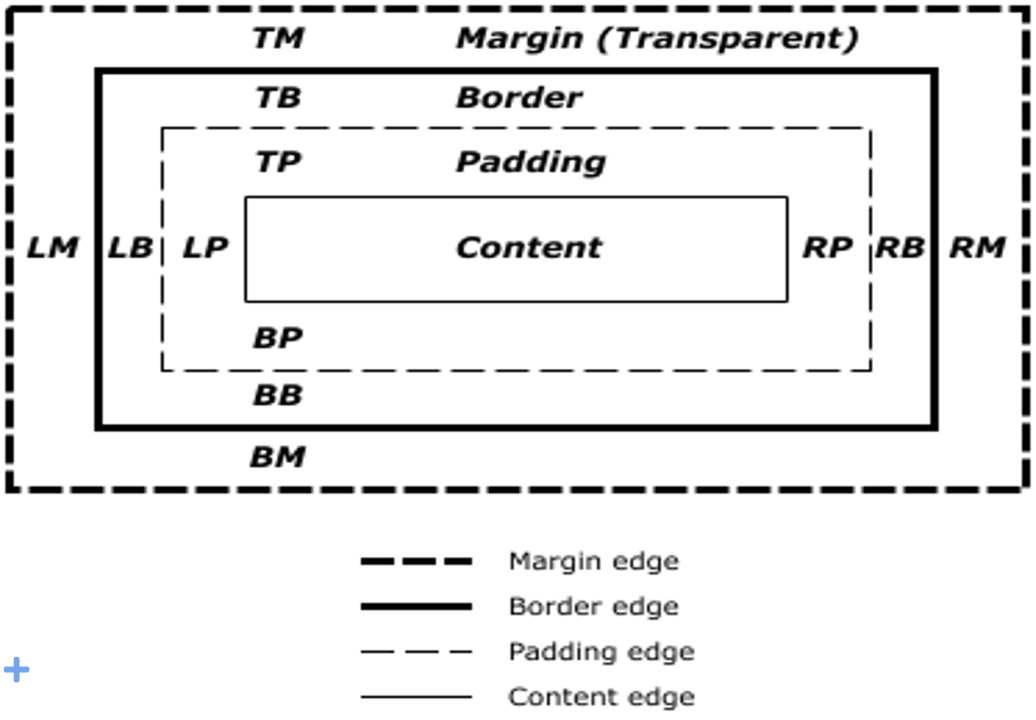
CSS Backgrounds

- The CSS background properties are used to define the background effects for elements.
- CSS background properties:
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position

Example

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

The Box Model



Padding and Margin both are same structure :

Padding { padding: 10px 15px 10px 15px;}

Top Right Bottom Left

Padding { padding: 10px 15px 10px;}

Top Right + Left Bottom

Padding { padding: 10px 15px;}

Top + Bottom Left + Right

Padding{ padding: 10px;}

Top + Bottom + Left + Right

CSS Borders

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

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
```

border: 1px dashed black;

A dotted border.

A dashed border.

A solid border.

A double border.

CSS Margins

- The CSS margin properties are used to generate space around elements.
- The margin properties set the size of the white space OUTSIDE the border.
- CSS has properties for specifying the margin for each side of an element:
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
}
```

```
margin: 1px 2px 3px 4px;
```

CSS Padding

- The CSS padding properties are used to generate space around content.
- The padding properties set the size of the white space between the element content and the element border.
  - The padding clears an area around the content (inside the border) of an element.
  - Example:

```
p {
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
```

CSS Fonts

- Some properties:
  - font-family (backup fonts)
  - font-style

```
p {
    font-family: "Times New Roman", Times, serif;
}
```

```
p.normal {
    font-style: normal;
}

p.italic {
    font-style: italic;
}
```

```
h1 {
    font-size: 40px;
}
```

CSS Links

- The four links states are:
  - a:link - a normal, unvisited link
  - a:visited - a link the user has visited
  - a:hover - a link when the user mouses over it
  - a:active - a link the moment it is clicked

- Example:

```
/* unvisited link */
a:link {
    color: red;
}

/* visited link */
a:visited {
    color: green;
}

/* mouse over link */
a:hover {
    color: hotpink;
}

/* selected link */
a:active {
    color: blue;
}
```

CSS Lists

- In HTML, there are two main types of lists:
  - unordered lists (<ul>) - the list items are marked with bullets
  - ordered lists (<ol>) - the list items are marked with numbers or letters
- The CSS list properties allow you to:
  - Set different list item markers for ordered lists
  - Set different list item markers for unordered lists
  - Set an image as the list item marker
  - Add background colors to lists and list items

Example

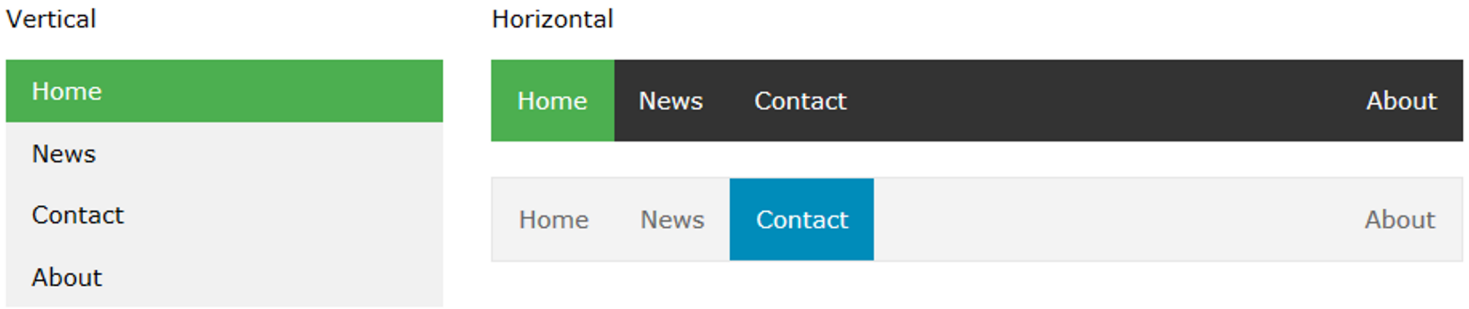
```
ul.a {
    list-style-type: circle;
}

ul.b {
    list-style-type: square;
}

ol.c {
    list-style-type: upper-roman;
}

ol.d {
    list-style-type: lower-alpha;
}
```

CSS Navigation Bars



CSS Website Outline