

Lecture 10

Cascading Style Sheet (CSS)

1. Introduction

Cascading Style Sheet (CSS) is a language that allows the user to change the appearance or presentation of elements on the page: the size, style, and color of text; background colors; border styles; even the position of elements on the page. Presentation refers to the way the document is displayed or delivered to the user, whether on a computer screen, a cell phone display, printed on paper, or read aloud by a screen reader. With style sheets handling the presentation, HTML can handle the business of defining document structure and meaning, as intended.

2. Benefit of CSS

- **CSS saves time:** write CSS once and then reuse the same sheet in multiple HTML pages.
- **Pages load faster:** Using CSS avoids the need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
- **Easy maintenance:** To make a global change, simply change the style, and all the elements in all the web pages will be updated automatically.
- **Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so CSS give a better look to HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device such as PDAs and cellphones using the same HTML document.

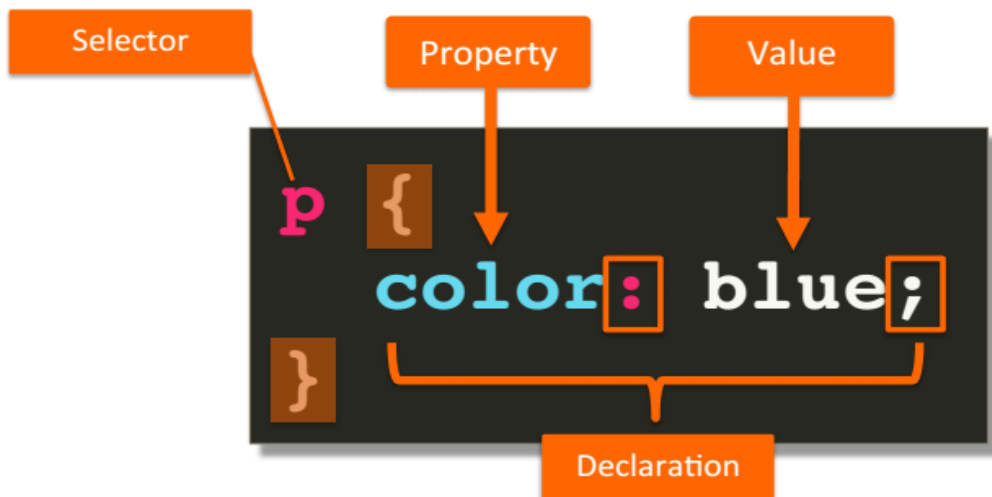
2. How style sheets Work

2.1 Marking up the document

Start with a document that has been marked up in HTML.

2.2 Writing the Rules

A style sheet is made up of one or more style instructions (called **rules** or **rule sets**) that **describe how an element or group of elements should be displayed**. In CSS, the two main sections of a rule are **selectors** and **declarations**.



- **Selectors**

Selector identifies the element or elements to be affected. This is called an element type selector, and it is the most basic type of selector.

For example, if use the h1 and p elements as selectors:

```
h1 { color: green; }
```

```
p { font-size: small; font-family: sans-serif; }
```

The properties defined for each rule will apply to every h1 and p element in the document. This makes all the h1 elements in the document green and specifies the paragraphs in a small, sans-serif font.

Selectors can be used to target elements, including ways to select groups of elements and elements that appear in a particular context.

- **Declarations**

The declaration is made up of a **property** (such as color) and its **value** (green), separated by a colon and a space. If the semicolon is omitted, the declaration and the one following it will be ignored.

There can be more than one declaration in a single rule are placed inside curly brackets. For example, the rule for the **p** element shown earlier in the code example has both the font-size and font-family properties.

```
p {  
  color: blue;  
  font-size: 20px;  
  width: 200px;  
}
```

Zero or More
Declarations
are allowed

Because CSS ignores whitespace and line returns within the declaration block, authors typically write each declaration in the block on its own line, as shown in the following example. This makes it easier to find the properties applied to the selector and to tell when the style rule ends.

```
p {  
  font-size: small;  
  font-family: sans-serif;  
}
```

Note: Sometimes it is helpful to use comments in a style sheet. CSS has its own comment syntax, shown here:

```
/* comment goes here */
```

Content between the `/*` and `*/` will be ignored when the style sheet is parsed, which means the comments can be anywhere in a style sheet, even within a rule.

```
body
```

```
{font-size: small;
```

```
/* font-size:large; */}
```

2.3 Attaching the Styles to the Document

There are three ways to style information that can be applied to an HTML document:

External style sheets. An external style sheet is a separate, text-only document that contains a number of style rules. It must be named with the CSS suffix (file.CSS). The CSS document is then **linked to or imported into one or more HTML documents**. In this way, all the files in a website may share the same style sheet. This is the most powerful and preferred method for attaching style sheets to content.

Embedded style sheets. It is placed in a document using the style element, and its rules apply only to that document. The style element must be placed in the head of the document and it must contain a type attribute that identifies the content of the style element as "**text/css**". This example also includes a comment.

```
<head>
  <title>Required document title here</title>
  <style>
    /* style rules go here */
  </style>
</head>
```

Inline styles. The user can apply properties and values to a **single element** using the **style attribute in the element itself** by using the generic syntax:

```
<element style="...style rules....">
```

For example:

```
<h1 style="color: red">Introduction</h1>
```

To add multiple properties, just separate them with **semicolons**, like this:

```
<h1 style="color: red; margin-top: 2em">Introduction</h1>
```

Because an inline style is applied only to the particular element, it doesn't need a selector. Inline styles should be avoided, unless it is absolutely necessary to override styles from an embedded or external style sheet. An advantage of using an inline style is that it always overrides styles that are defined elsewhere because the inline styles are specific to the element on which the inline style is defined. This specificity can solve isolated problems when a style is applied globally in an external style sheet, but one element needs to be styled differently.

3. Linking CSS to a Web Page

When using an external style sheet, it is need to link the applied styles to the web pages. To do this, a **link element** is added to the head of each web page that will be styled using CSS.

Example:

```
<head>  
<title>Example of external style</title>  
<link href="style1.css" rel="stylesheet" type="text/css"/>  
</head>
```

The **href** attribute tells the web browser where the style sheet file (style1.css) can be found, in the same way that the href attribute is used in an anchor (<a>) to point

to the destination file. The `rel="stylesheet"` and `type="text/css"` parts of the link tag tell the browser what kind of file is being linked to, and how the browser should handle the content. It is important to include these attributes when linking to a `.css` file. The link element is another one of those empty elements, without separate start and end tags.

Example: Create two files, one HTML and the other CSS

First file: save as file.html

```
<html>
<head>
  <title>My document</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
  <h1>My first stylesheet</h1>
</body>
```

Second file: save as style.css

```
body {
  background-color: #FF0000;
}
```

If the web site has three web pages: `index.html`, `about.html`, and `contact.html`, to apply the style sheet, add link element in each of those files.

After save each page, opening each one in the web browser then all paragraphs should now display in same style. If the color is changed in `.css` file from blue to red, then this change reflected across all pages.

4. Grouped and ID Selectors

4.1 Grouped Selectors

There are two selector types: a simple element selector and grouped selectors. To apply the same style property to a number of elements, selectors can be grouped into one rule by separating them with **commas**. One rule has the same effect as the five rules listed. Grouping selectors makes future edits more efficient and results in a smaller file size.

```
h1, h2, p, img { border: 1px solid blue; }
```

4.2 ID Selectors

If one paragraph in the site differs from the others in terms of its purpose, it's distinct from the actual document content, so it might benefit from some alternate styling. It can set a rule for this paragraph only using **div** element with an **id** attribute as:

1. The HTML for that paragraph (for example) will be:

```
<div id="tag">
  <p>Here an example of id to specify the paragraphs!</p>
</div>
```

2. Open the CSS file for editing, and add the following after the first paragraph rule:

```
#tag p {
  font-style: italic;
  font-family: Georgia, Times, serif;}
```

3. Save the files and refresh the page in browser.

This CSS rule means, for any paragraph element that occurs inside an element that has an id of tag, set the text to italics and the font to Georgia, Times. The # notation in the CSS refers to an element with a specific id attribute.

