

Web Technologies 1

Introduction to CSS



Lec6

Mohammed Sultan

What is CSS?

- CSS stands for Cascading Style Sheets.
- CSS ("Cascading Style Sheets") determines how the elements in our HTML documents are displayed and formatted.
- By using CSS, we separate the content of a web page from the presentation (format and styling) of that content.
- CSS enables us to make all pages of our website look similar and consistent.
- The power of CSS is that it allows us to make site-wide formatting changes by making edits to a single file.

Three Ways to Use CSS

- We can add CSS code in any combination of three different ways:
 - **Inline Style**
 - CSS code is placed directly into an XHTML element within the `<body>` section of a web page.
 - **Internal Style Sheet**
 - CSS code is placed into a separate, dedicated area within the `<head>` section of a web page.
 - **External Style Sheet**
 - CSS code is placed into a **separate computer file** and then linked to a web page.

Inline Style

- An inline style declaration is highly specific and formats just one element on the page. No other elements, including other `<h2>` elements on the page, will be affected by this CSS style.

```
<p style="background: blue; color: white;">
```

Since inline styles have limited scope and do not separate content from presentation, their use is generally discouraged

Internal Style Sheet

- Styles declared in the internal style sheet affect all matching elements on the page.

```
<head>
```

```
<style type="text/css">
```

```
p {color: white; }
```

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

```
</style>
```

```
</head>
```

- Since formatting declarations are entirely in the <head> section, away from the actual page content, internal CSS style sheets do a much better job than inline styles at separating content from presentation.

External Style Sheet

- It keeps your website design and content separate.

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

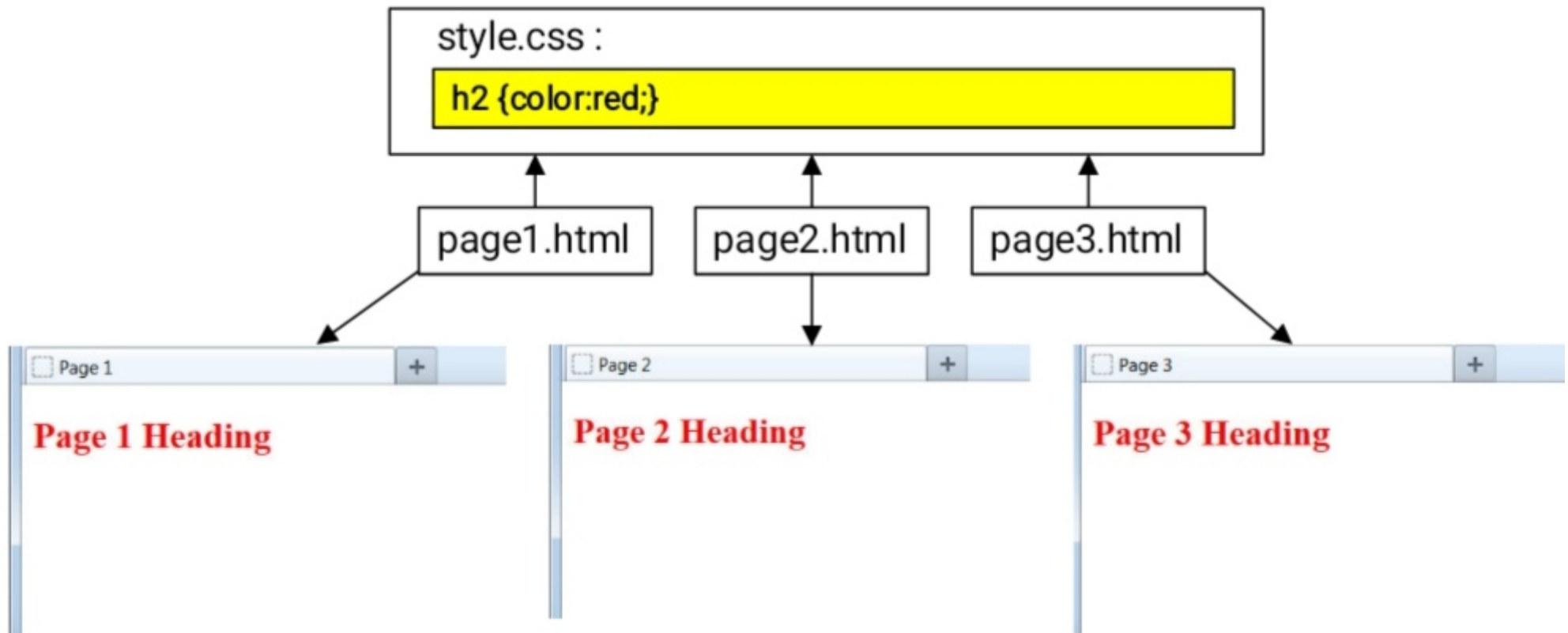
- The **<link>** element instructs the browser to load the external file specified by the href attribute and to apply the CSS style declarations contained there.

External Style Sheet

- An external style sheet can be written in any text editor, and must be saved with a .css extension.
- The external .css file should not contain any HTML tags.

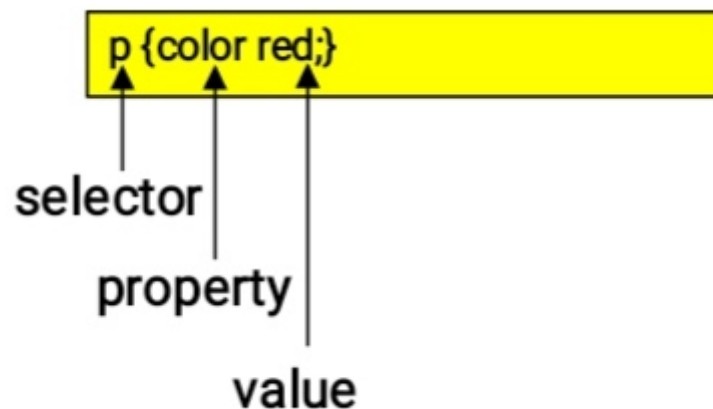
Benefit of External Style Sheet

- The real power of using an external style sheet is that multiple web pages on our site can link to the same style sheet:



CSS Syntax

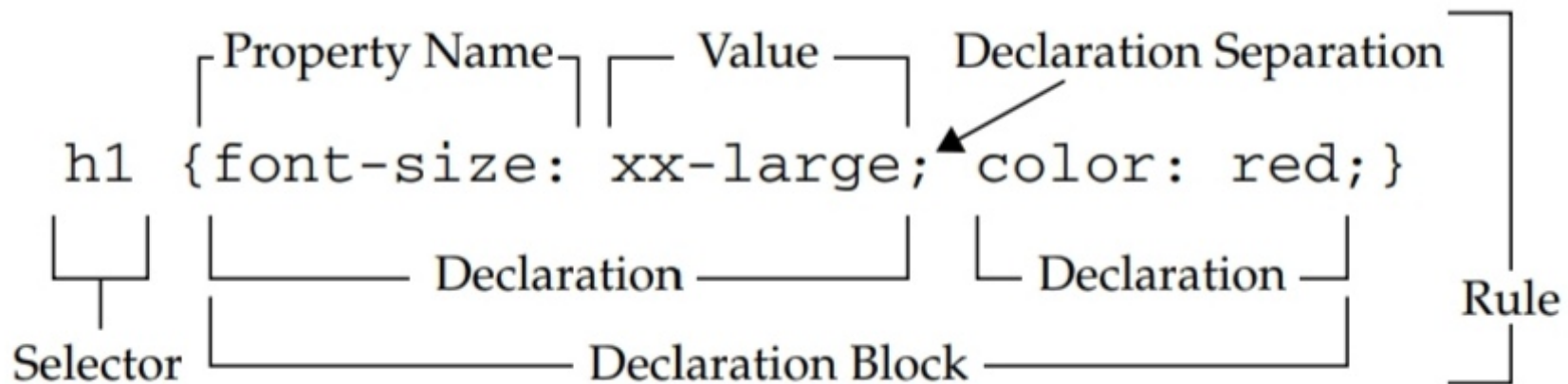
- The correct syntax of a CSS declaration is:
selector {property:value;}



Internal and external style sheets use this identical CSS syntax. Internal style sheets must use the opening and closing `<style>` tags to surround the CSS code, A semicolon must be placed after each CSS declaration.

CSS Syntax

- CSS rules are defined as a property name followed by a colon and then a property value. Individual rules are terminated by semicolons, with the final rule having an optional semicolon:
 - property-name1 : value1; ... property-nameN : valueN;



CSS Syntax

- CSS **property names** are separated by dashes when they are multiple words—for example, font-face, font-size, line-height, and so on.
- Allowed **values** come in many forms; from simple keywords like xx-small, strings like "Arial", plain numbers like 0, numbers with units like 100px or 2cm, and special delimited values such as URLs, url(../styles/fancy.css).

Setting Multiple Properties

- We can define as many properties as we wish for a selector:
 - `p {color:red;font-style:italic;text-align:center;}`

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

This format is readable
than the first format

How Browsers Process CSS

- A web browser will process all CSS code it encounters, even if it is from all three methods.
- For example, an external style sheet could define the font of a heading, an internal style sheet could specify the font size of the heading, and an inline style could italicize the heading. All three would be applied.
- Sometimes a browser will receive conflicting instructions from the CSS code. For example, what if each of the above CSS sources specified a different color for the heading text?

What Does "Cascading" Mean?

- We use the term "**cascading**" because there is an established order of priority to resolve formatting conflicts:
 1. Inline style (highest priority)
 2. External and internal style sheets (in the head section)
 3. Web browser default (only if not defined elsewhere)

If multiple, conflicting styles are defined in the same style sheet, only the final one will be applied. Be careful, as this is another common mistake committed by beginners.

CSS Selectors

- CSS selectors are used to "find" (or select) the HTML elements you want to style.
- The selector is normally the HTML element you want to style.
- Each declaration consists of a property and a value.
- The property is the style attribute you want to change. Each property has a value.

CSS Selectors

- CSS selectors divide into five categories:
 - **Simple selectors** (select elements based on name, id, class)
 - **Combinator selectors** (select elements based on a specific relationship between them)
 - **Pseudo-class selectors** (select elements based on a certain state)
 - Pseudo-elements selectors (select and style a part of an element)
 - Attribute selectors (select elements based on an attribute or attribute value)

Element

- The simplest rules can be applied to all occurrences of a particular tag, such as `<p>`. These selectors are called element selectors and are simply used as follows:
 - `element-name { /* properties */ }`
 - As an example, to set the line spacing for all paragraphs, use a rule such as the following:
 - `p {line-height: 150%;}`

The CSS id Selector

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

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

An id name cannot start with a number!

Class

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.

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

Multiple Selectors

```
h1 {  
  color: blue;  
  text-align: center;  
}  
.summary {  
  color: blue;  
}  
#header {  
  color: blue;  
  text-align: center;  
}  
a:link {  
  color: blue;  
}
```

=

```
h1, .summary, #header,  
a:link {  
  color: blue;  
}  
h1, #header {  
  text-align: center;  
}
```

It's better to group the selectors, to minimize the code.

Separate each selector with a comma.

Universal selector

- The universal selector (*) selects all HTML elements on the page.

```
* {  
  text-align: center;  
  color: blue;  
}
```

CSS Combinators

- A combinator is something that explains the relationship between the selectors.
- There are four different combinators in CSS:
 - descendant selector (space)
 - child selector (>)
 - adjacent sibling selector (+)
 - general sibling selector (~)

Descendant Selector

- The descendant selector matches all elements that are descendants of a specified element.

```
div p {  
  background-color: yellow;  
}
```

```
<div>  
  <p>Paragraph 1 in the div.</p>  
  <p>Paragraph 2 in the div.</p>  
  <section><p>Paragraph 3 in the div.  
</p></section>  
</div>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

```
<p>Paragraph 4. Not in a div.</p>  
<p>Paragraph 5. Not in a div.</p>
```


Child Selector (>)

- The child selector selects all elements that are the children of a specified element.

```
div > p {  
  background-color: yellow;  
}
```

```
<div>  
  <p>Paragraph 1 in the div.</p>  
  <p>Paragraph 2 in the div.</p>  
  <section>  
    <!-- not Child but Descendant -->  
    <p>Paragraph 3 in the div (inside  
a section element).</p>  
  </section>  
  <p>Paragraph 4 in the div.</p>  
</div>
```

```
<p>Paragraph 5. Not in a div.</p>  
<p>Paragraph 6. Not in a div.</p>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div (inside a section element).

Paragraph 4 in the div.

Adjacent Sibling Selector (+)

- The adjacent sibling selector is used to select an element that is directly after another specific element.
- Sibling elements must have the same parent element, and "adjacent" means "immediately following".

```
div + p {  
  background-color: yellow;  
}
```

```
<div>
```

```
<p>Paragraph 1 in the div.</p>
```

```
<p>Paragraph 2 in the div.</p>
```

```
</div>
```

```
<p>Paragraph 3. After a div.</p>
```

```
<p>Paragraph 4. After a div.</p>
```

```
<div>
```

```
<p>Paragraph 5 in the div.</p>
```

```
<p>Paragraph 6 in the div.</p>
```

```
</div>
```

```
<p>Paragraph 7. After a div.</p>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. After a div.

Paragraph 4. After a div.

Paragraph 5 in the div.

Paragraph 6 in the div.

Paragraph 7. After a div.

General Sibling Selector (~)

- The general sibling selector selects all elements that are next siblings of a specified element.

```
div ~ p {  
  background-color: yellow;  
}
```

```
<p>Paragraph 1.</p>
```

Paragraph 1.

```
<div>  
  <p>Paragraph 2.</p>  
</div>
```

Paragraph 2.

Paragraph 3.

```
<p>Paragraph 3.</p>  
<code>Some code.</code>  
<p>Paragraph 4.</p>
```

Some code.

Paragraph 4.

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

CSS comments

- Comments are used to explain the code, and may help when you edit the source code at a later date.
- Everything between the `/* and */` will be disregarded, even if the comments span multiple lines.
- CSS comments are often used when experimenting with styles or troubleshooting problems. Rather than deleting style declarations and then retyping them again, we can just "comment them out" temporarily, evaluate the effect, and then restore the style by removing the `/* and */` characters.
 - `/* mystyle.css - a sample style sheet */`
 - `h1 {color: red; text-align: center;}`
 - `p {line-height: 150%;}`

STYLING TEXT

Styling Text

- CSS has a lot of properties for formatting text.
- CSS text properties :
 - font-family(serif)
 - font-size(32)
 - *font-style (italic)*
 - **font-weight (bold)**
 - color
 - background-color
 - text-decoration
 - Text-align (center)

Styling Text

Property Some Possible Values

text-align: center, left, right, justify

text-decoration: underline, line-through, blink

color: blue, green, yellow, red, white, etc.

font-family: Arial, Verdana, "Times New Roman"

font-size: large, 120%, 20px (pixels)

font-weight: bold, normal

font-style: italic, normal

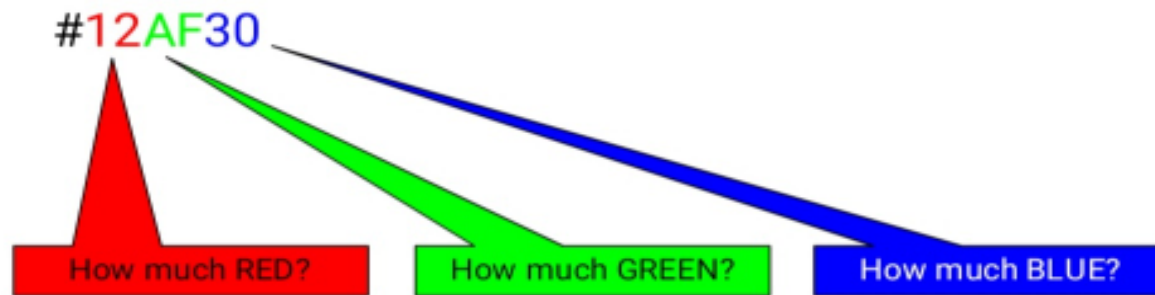
Font

- Choosing the right font for your website is important!
- Using a font that is easy to read is important.
- The font adds value to your text.
- If the font name is more than one word, it must be in quotation marks, like: "Times New Roman".

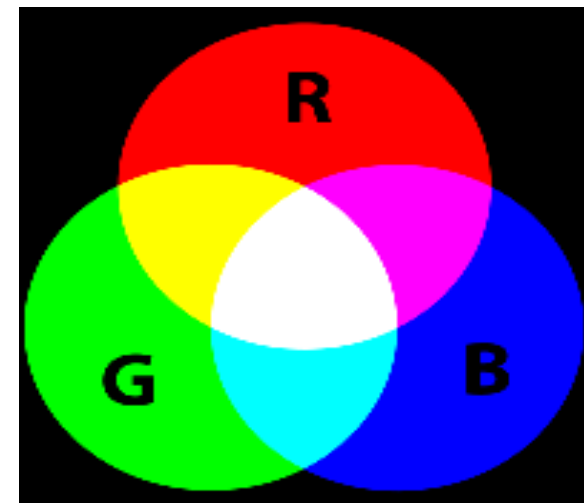
```
.p1 {  
  font-family: "Times New Roman", Times, serif;  
}
```

Color

- The color property is used to set the color of the text. The color is specified by:
 - a color name - like "red"
 - a HEX value - like "#12AF30"



- an RGB value - like "rgb(255,0,0)"



Styling Text Links

- Like other text on the page, link text (between the opening `<a>` and closing `` tags) can be styled using CSS. Some of the properties that can be set are:
 - font-family(serif)
 - font-size(32)
 - *font-style (italic)*
 - **font-weight**
 - **color**
 - **background-color**
 - text-decoration

Link States

- The <a> element is unusual in that it can be in one of four different conditions - or **states** - and we can define separate styles for each of these **states**. The four states are:
 - a:link = unvisited link (default state)
 - a:visited = previously visited link
 - a:hover = link being moused over
 - a:active = link being clicked

Example CSS Link Styling

```
<head>
<style type="text/css">
  a:link {
    color: blue;
  }
  a:visited {
    color: green;
  }
  a:hover {
    color: orange;
  }
  a:active {
    color: red;
  }
</style>
</head>
...
<a href="http://www.google.com">
  Link to Google
</a>
```

Link to Google

Link to Google

Link to Google

Link to Google

As we mouse over the link but before clicking on it.

As we are actually clicking on the link.

Do not put a space after a colon in the selector section. Doing so will result in the style declaration being ignored.

Any Questions?