

CSS Overview

Web application with HTML and CSS



Lesson Objectives

- CSS Overview
- CSS Syntax
- Cascade and Inheritance
- Selectors
- Common properties

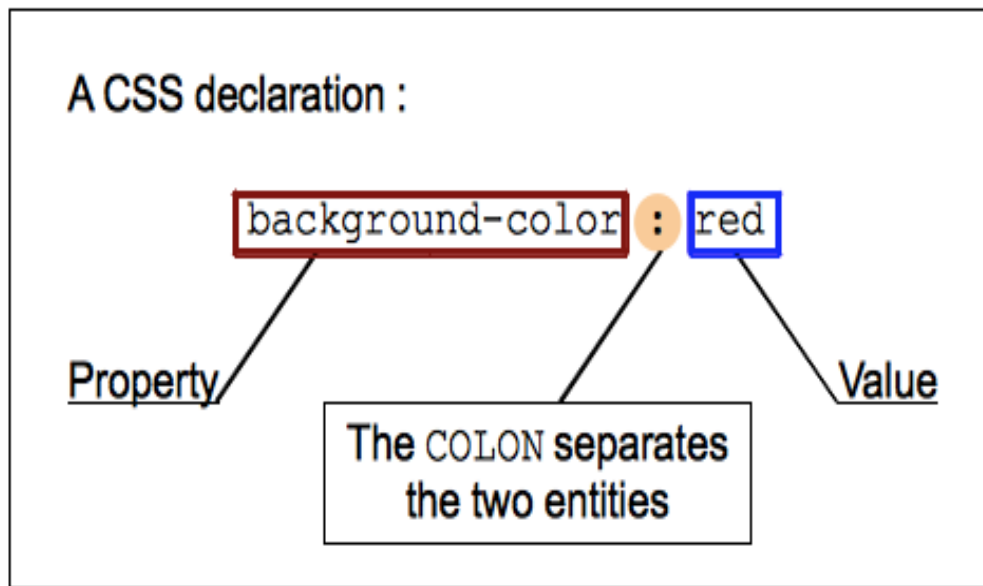
Section 1

CSS OVERVIEW

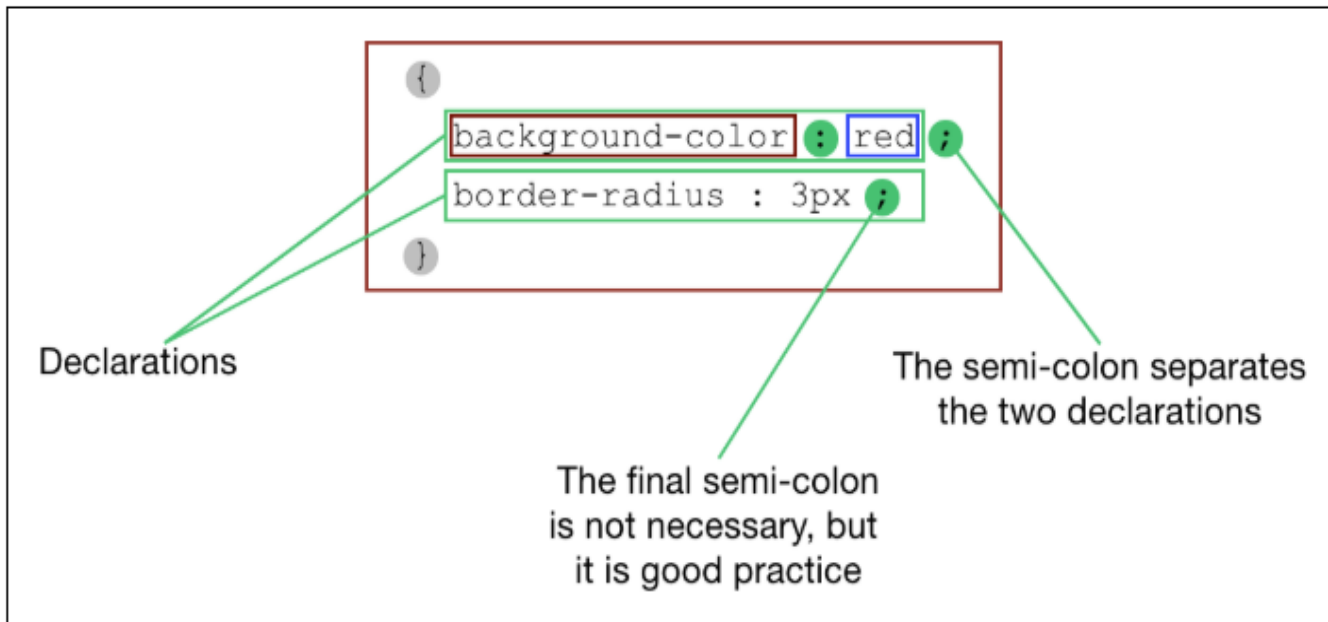
- **CSS** is a language that describes the style of an HTML document
 - CSS stands for **C**ascading **S**tyle **S**heets
 - CSS describes how HTML elements should be rendered on screen, on paper, in speech, or on other media.
 - CSS is one of the core languages of the open Web and is standardized across Web browsers according to the W3C specification

- The basic goal of the **CSS** language is to allow a browser engine to paint elements of the page with specific features, like colors, positioning, or decorations.
- **CSS** Syntax includes:
 - The **property** which is an identifier, that is a human-readable name, that defines which feature is considered.
 - The **value** which describe how the feature must be handled by the engine. Each property has a set of valid values, defined by a formal grammar, as well as a semantic meaning, implemented by the browser engine.

- **CSS Declarations:** A property and value pair is called a **declaration**



➤ **CSS Declarations block:** Declarations are grouped in **blocks**



- CSS can be added to HTML elements in 3 ways:
 - **Inline** - by using the style attribute in HTML elements
 - **Internal** - by using a <style> element in the <head> section
 - **External** - by using an external CSS file

- **Inline style:** An inline CSS is used to apply a unique style to a single HTML element

❖ *Syntax*

```
<element style="style_definitions">
```

Value	Description
<i>style_definitions</i>	One or more CSS properties and values separated by semicolons (e.g. style="color:blue;text-align:center")

- ❖ **Note:** The style attribute will override any style set globally, e.g. styles specified in the <style> tag or in an external style sheet.

➤ Internal style:

- An internal CSS is used to define a style for a single HTML page.
- It is defined in the **<head>** section of an HTML page, within a **<style>** element

```
<head>
  <meta charset="utf-8">
  <title>My CSS experiment</title>
  <style>
    h1 {
      color: blue;
      background-color: yellow;
      border: 1px solid black;
    }

    p {
      color: red;
    }
  </style>
</head>
```

➤ External style:

- An external style sheet is used to define the style for many HTML pages.
- With an external style sheet, you can change the look of an entire web site, by changing one file.
- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

mystyle.css

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

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

- **Cascading order:** All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:
1. Inline style.
 2. External and internal style sheets.
 3. Browser default

Section 2

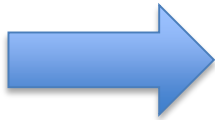
CASCADE AND INHERITANCE

- Is an algorithm that defines how to combine property values originating from different sources.

```
<h1>This is my heading.</h1>
```

+

```
h1 {  
  color: red;  
}  
h1 {  
  color: blue;  
}
```



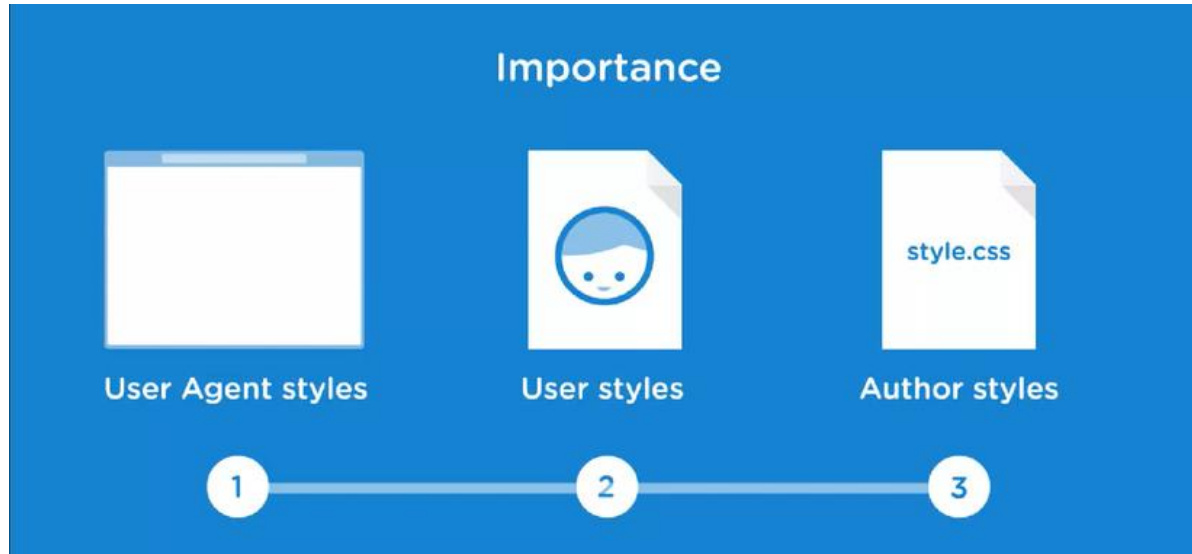
This is my heading.

➤ There are three main concepts that control the order in which CSS declarations are applied:

1. *Importance*
2. *Specificity*
3. *Source order*

=> **Importance** is the most important. If two declarations have the same importance, the **specificity** of the rules decide which one will apply. If the rules have the same specificity, then **source order** controls the outcome.

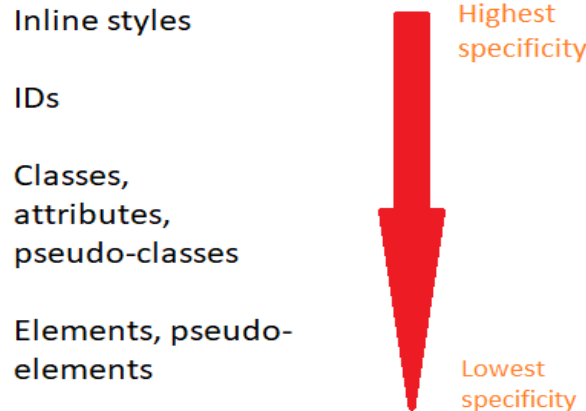
➤ Importance:



➤ Importance:

1. **User-agent stylesheets:** The author of the page defines the styles for the document using one or more stylesheets, which define the look and feel of the website — its theme.
2. **Author stylesheets:** The author of the page defines the styles for the document using one or more stylesheets, which define the look and feel of the website — its theme.
3. **User stylesheets:** The user (or reader) of the web site can choose to override styles in many browsers using a custom **user stylesheet** designed to tailor the experience to the user's wishes.

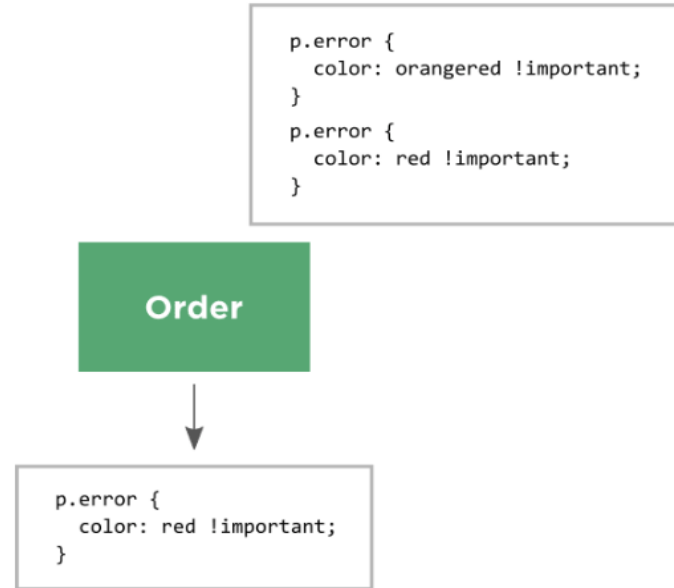
- **Specificity** is how the browser decides which rule applies if multiple rules have different selectors, but could still apply to the same element.



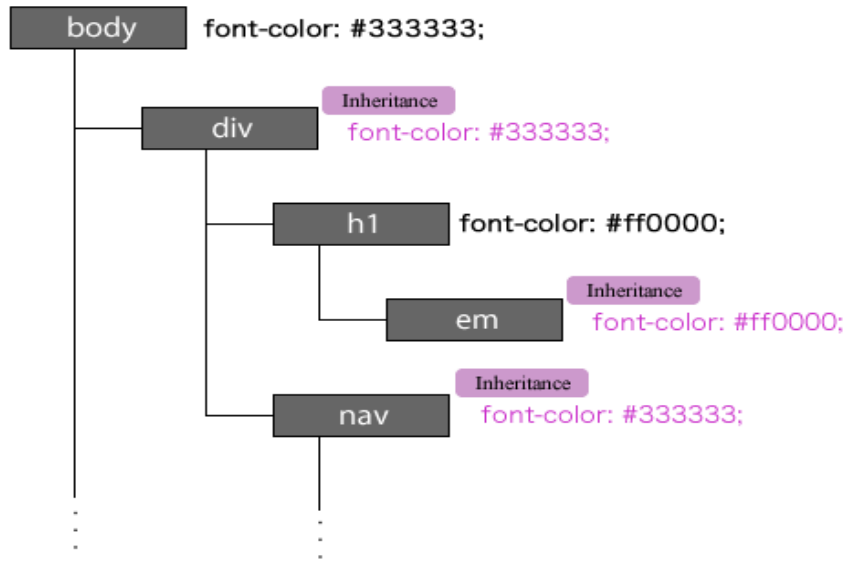
➤ Specificity rules:

- ***Equal specificity: the latest rule counts*** - If the same rule is written twice into the external style sheet, then the lower rule in the style sheet is closer to the element to be styled, and therefore will be applied
- ***ID selectors have a higher specificity than attribute selectors***
- ***Contextual selectors are more specific than a single element selector*** - The embedded style sheet is closer to the element to be styled. So in the following situation
- ***The universal selector and inherited values have a specificity of 0*** - *, body * and similar have a zero specificity. Inherited values also have a specificity of 0.

- **Source order:** If the CSS rules have the same *Importance* and *Specificity*, we will consider the order of their appearance - which CSS rule that is written later will be preferred over CSS that is written first.



- **Inheritance** works on a property by property basis. When you set properties on a selector in CSS, they're either inherited by all the children of that selector or they're not



Section 2

CSS SELECTORS

- **CSS selectors** are used to "find" (or select) the HTML elements you want to style.
- We can divide CSS selectors into four categories:
 1. Simple selector (type, id, class)
 2. Combinator selector
 3. Pseudo classes & pseudo elements
 4. Attribute selectors

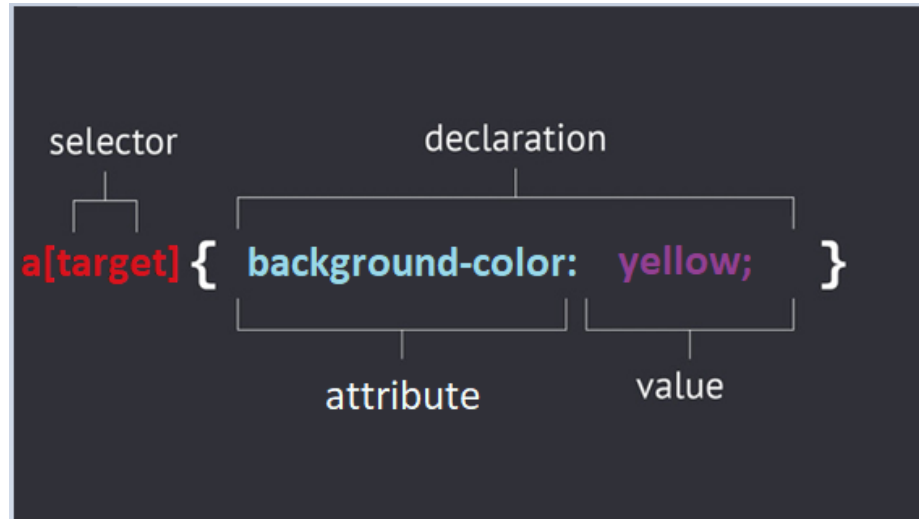
- ***The element selector:*** Selects HTML elements based on the element name
- ***Id selectors:*** Uses the id attribute of an HTML element to select a specific element.
- ***Class selectors:*** Selects HTML elements with a specific class attribute

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

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

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


- The [attribute] selector is used to select elements with a specified attribute.




- **CSS [attribute="value"] Selector:**

Select elements with a specified attribute and value.

```
a[target="_blank"] {  
    background-color: yellow;  
}
```

- **Pseudo class:** Is used to define a special state of an element.

Syntax




```
selector:pseudo-class {  
  property:value;  
}
```

Selector	Example	Example description
<u>:active</u>	a:active	Selects the active link
<u>:checked</u>	input:checked	Selects every checked <input> element
<u>:disabled</u>	input:disabled	Selects every disabled <input> element
<u>:empty</u>	p:empty	Selects every <p> element that has no children
<u>:enabled</u>	input:enabled	Selects every enabled <input> element

- **Pseudo element:** Is used to style specified parts of an element.

Syntax



```
selector::pseudo-element {  
  property:value;  
}
```

Selector	Example	Example description
<u>::after</u>	p::after	Insert something after the content of each <p> element
<u>::before</u>	p::before	Insert something before the content of each <p> element
<u>::first-letter</u>	p::first-letter	Selects the first letter of each <p> element
<u>::first-line</u>	p::first-line	Selects the first line of each <p> element
<u>::selection</u>	p::selection	Selects the portion of an element that is selected by a user

- **Combinator selector** is used to select element base on the relationship.
- There are four different combinators in CSS:
 - *Descendant selector (space)*
 - *Child selector (>)*
 - *Adjacent sibling selector (+)*
 - *General sibling selector (~)*

Combinatory selectors

CSS COMBINATORS		
Descendant Combinators	<code>div h1</code>	heading <code><h1></code> inside <code><div></code>
Child Combinators	<code>div > h1</code>	<code><h1></code> with parent <code><div></code>
Adjacent Sibling Combinators	<code>div + h1</code>	select adjacent <code><h1></code> which is a sibling of <code><div></code>
General Sibling Combinators	<code>div ~ h1</code>	select all <code><h1></code> which are siblings of <code><div></code>

Section 3

COMMON PROPERTIES

- Choosing the right font has a huge impact on how the readers experience a website.
- The right font can create a strong identity for your brand.
- Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

1. **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
2. **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
3. **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
4. **Cursive** fonts imitate human handwriting.
5. **Fantasy** fonts are decorative/playful fonts.

Generic Font Families

Serif



Sans



On computer screens, sans-serif fonts are considered easier to read than serif fonts.

- the font-family property to specify the font of a text.
- The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems.

```
.p1 {  
  font-family: "Times New Roman", Times, serif;  
}  
  
.p2 {  
  font-family: Arial, Helvetica, sans-serif;  
}  
  
.p3 {  
  font-family: "Lucida Console", "Courier New", monospace;  
}
```

- The font-style property is mostly used to specify italic text.
- This property has three values:
 - ✓ normal - The text is shown normally
 - ✓ italic - The text is shown in italics
 - ✓ oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

- The font-size property sets the size of the text.
- Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.
- Always use the proper HTML tags, like `<h1>` - `<h6>` for headings and `<p>` for paragraphs.

Property	Description
<u>font</u>	Sets all the font properties in one declaration
<u>font-family</u>	Specifies the font family for text
<u>font-size</u>	Specifies the font size of text
<u>font-style</u>	Specifies the font style for text
<u>font-variant</u>	Specifies whether or not a text should be displayed in a small-caps font
<u>font-weight</u>	Specifies the weight of a font

- **Color** here refers to text color.
- It takes on a #hex value or a rgb value as with border colors.

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Example

```
<h1 style="color:Tomato;">Hello World</h1>  
<p style="color:DodgerBlue;">Lorem ipsum...</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

- The text-align property is used to set the horizontal alignment of a text.
- A text can be left or right aligned, centered, or justified.

TEXT FORMATTING

This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties. The paragraph is indented, aligned, and the space between characters is specified. The underline is removed from this colored "Try it Yourself" link.

- Text Decoration
 - ✓ text-decoration: used to set or remove decorations from text (often used to remove underlines from links)
- Text Transformation
 - ✓ text-transform: used to specify uppercase and lowercase letters in a text
- Text Indentation:
 - ✓ text-indent: used to specify the indentation of the first line of a text
- Letter Spacing
 - ✓ letter-spacing: used to specify the space between the characters in a text
- Line Height
 - ✓ line-height: used to specify the space between lines
- Word Spacing
 - ✓ word-spacing: used to specify the space between the words in a text
- White Space
 - ✓ white-space: specifies how white-space inside an element is handled

Background color

- The **background-color** property specifies the background color of an element.
- The property accepts any valid color: color name, RGB, HEX,...

```
.box {  
  background-color: #567895;  
}  
  
h2 {  
  background-color: black;  
  color: white;  
}  
  
span {  
  background-color: rgba(255,255,255,.5);  
}
```

```
<div class="box">  
  <h2>Background Colors</h2>  
  <p>Try changing the background <span>colors</span>.</p>  
</div>
```

Background Colors

Try changing the background colors.

Background images

- 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

```
.a {  
  background-image: url(balloons.jpg);  
}  
  
.b {  
  background-image: url(star.png);  
}
```

```
<div class="wrapper">  
  <div class="box a"></div>  
  <div class="box b"></div>  
</div>
```



- Controlling background-repeat: The background-repeat property is used to control the tiling behavior of images. The available values are
 - ✓ no-repeat — stop the background from repeating altogether.
 - ✓ repeat-x — repeat horizontally.
 - ✓ repeat-y — repeat vertically.
 - ✓ repeat — the default; repeat in both directions.

Lesson Summary



Thank you

