

# Welcome to CS1315 Intro to Media Computation

“Our greatest weakness lies in giving up.  
The most certain way to succeed  
is always to try just one more time.”

– Thomas Edison

Day 33

# Cascading Style Sheet

## What is CSS?

- CSS stands for Cascading Style Sheets
- 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 webpages all at once
- External stylesheets are stored in .css files

# Cascading Style Sheet

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>
```

```
<p>This is a paragraph.</p>
```

When tags like `<font>` and color attributes were added, the development of large websites, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

# Cascading Style Sheet

CSS removed the style formatting from the HTML page.

- **CSS saves a lot of work.** The style definitions are usually saved in a `.css` external file. This file can be used on multiple HTML pages.
- **Pages load faster.** With a CSS you do not need to write HTML tag attributes every time. Write one rule for a tag and it applies to all the occurrences of that tag.
- **Easy maintenance.** To make a global change, just change the style, and all the elements in all the pages will be updated automatically.
- **Multiple Device Compatibility.** Style sheets allow different versions of a website to be presented and optimized for various devices.

# Cascading Style Sheet

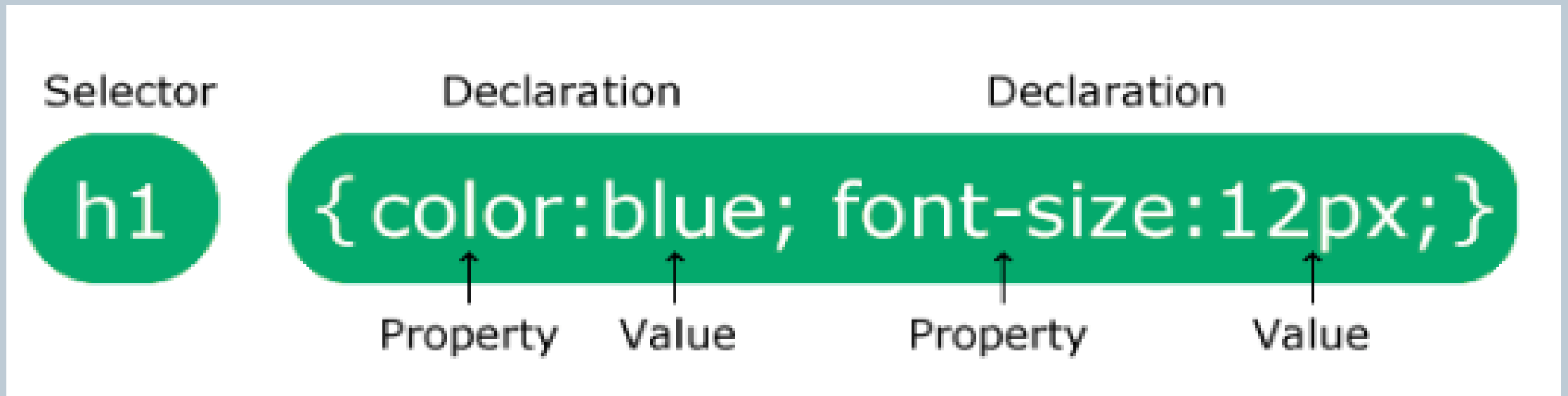
CSS handles the look and feel part of a web page:

- Color of the text
- Style of the fonts
- Spacing between paragraphs
- What background images or colors are used
- The design, layout and variations in display for different devices and screen sizes.

# Cascading Style Sheet

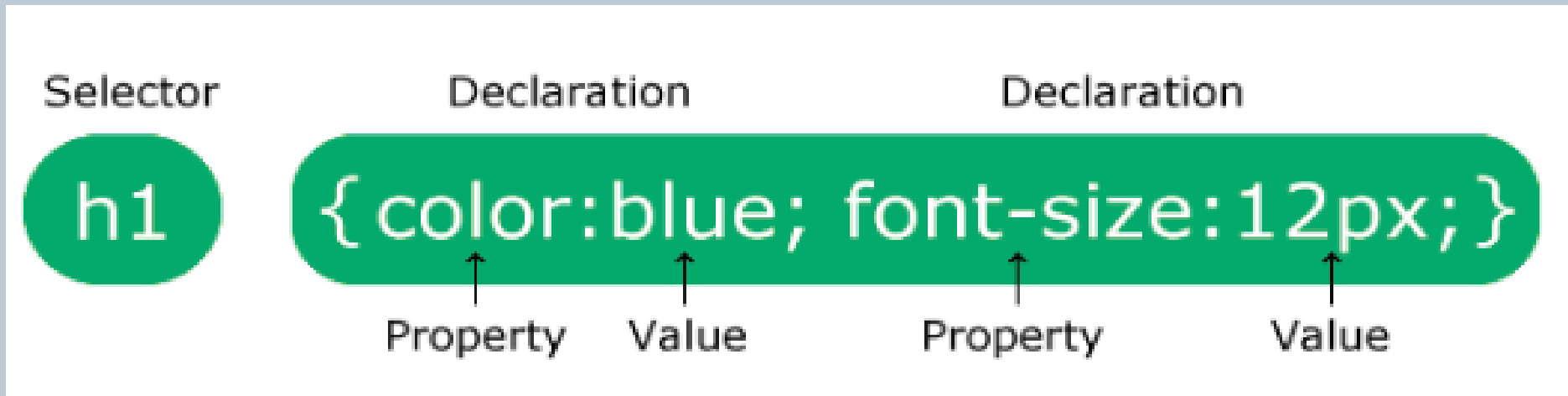
CSS syntax: a style rule is interpreted by a browser and then applied to the elements in an HTML document

A CSS rule consists of a selector and a declaration block.



# Cascading Style Sheet

- The **selector** points to the HTML element to style.
- The **declaration block** contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.



# CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style. We can divide CSS selectors into five categories:

- **Simple selectors** (select elements based on element, 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)



## CSS Selector – element

In this example all <p> elements will be center-aligned, with a red text color.

- `p` is a selector in CSS, pointing to the HTML element to be styled
- `color` is a property, and `red` is the property value
- `text-align` is a property, and `center` is the property value

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

<p>Hello World!</p>
<p>These paragraphs are styled with
CSS.</p>

</body>
</html>
```

# CSS Selector – element

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>
<p>Hello World!</p>
<p>These paragraphs are styled with
CSS.</p>
</body>
</html>
```

Hello World!  
These paragraphs are styled with CSS.

In this example all HTML  
<p> elements on the  
page will be red text and  
center-aligned.

## CSS Selector - id

Uses the id attribute of an HTML element to select a specific element.

- `id` is unique, so use the id selector to select one unique element.
- To select an element with a specific id, write a hash (#) character, followed by the id of the element (cannot start with a number).

```
<!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>
```

# CSS Selector - id

```
<!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>
```

Hello World!

This paragraph is not affected by the style.

In this example  
the HTML element with  
id="para1" will be red and  
center-aligned.

## CSS Selector - 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 (cannot start with a number).

```
<!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>
```

## CSS Selector - class

```
<!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>
```

Red and center-aligned heading  
Red and center-aligned paragraph.

In this example all  
HTML elements  
with class="center" will be  
red and center-aligned.

## CSS Selector – element with class

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red; }
</style>
</head>
<body>
<h1 class='center'>This heading
will not be affected</h1>
<p class='center'>This paragraph
will be red, center-aligned.</p>
</body>
</html>
```

This heading will not be affected  
This paragraph will be red and center-aligned.

In this example only <p>  
elements with  
class="center" will be red  
and center-aligned.

## CSS Selector – universal selector (\*)

```
<!DOCTYPE html>
<html>
<head>
<style>
*{ text-align: center;
   color: blue; }
</style>
</head>
<body>
<h1> Hello World!</h1>
<p>Every element will be affected
by the style.</p>
<p id='para1'>Me too!</p>
<p>And me!</p>
</body>
</html>
```

Hello world!

Every element will be affected by the style.

Me too!

And me!

In this example every  
HTML element on the page  
will be affected .



# CSS Selector – grouping selector

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p{ text-align: center;
          color: green;
        }
</style>
</head>
<body>
<h1> Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>

</body>
</html>
```

Hello World!  
Smaller heading!  
This is a paragraph.

In this example every HTML element listed in the style together will have the same style definition applied.

# CSS All Simple Selectors

Selector	Example	Example description
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>element.class</u>	p.intro	Selects only <p> elements with class="intro"
<u>*</u>	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element,element,...</u>	div, p	Selects all <div> elements and all <p> elements

# Cascading Style Sheet

## How to add CSS for an HTML document?

- When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.
- There are three ways of inserting a style sheet:
  1. External CSS: separate file with a `.css` extension
  2. Internal CSS: same HTML file, in the `<style>` element
  3. Inline CSS: same HTML file, within a single element

## CSS – adding external

- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

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

## CSS – adding external

- An external style sheet can be written in any text editor, and must be saved with a `.css` extension, in the same folder as the HTML file it will be styling
- The external `.css` file should not contain any HTML tags.

```
/* mystyle.css */
body {
    background-color: lightblue;
}
h1 {
    color: navy;
    margin-left: 20px;
}
```

## CSS – adding external

- Documenting your CSS file.
- A CSS comment is placed inside the `<style>` element, and starts with `/*` and ends with `*/`

```
<style>
/* mystyle.css - single line*/
p {
    color: red; /* set text color red - inline*/
}

/* other styles can follow
- multi line*/
</style>
```

## CSS – adding external

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet"
      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;
}
```

**This is a heading**

This is a paragraph.

## CSS – adding internal

```
<!DOCTYPE html>
<html>
<head>
<style>
body { background-color: linen;}
h1 { color: maroon;
     margin-left: 40px;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

### **This is a heading**

This is a paragraph.

Internal styles are defined within the <style> element, inside the head section.



## CSS – adding inline

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

**This is a heading**

This is a paragraph.

An inline style may be used to apply a unique style for a single element.

- add the style attribute to the element. The style attribute can be any CSS property.

## CSS – type priority

### Which CSS type has priority?

- All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number 1. has the highest priority:
  1. Inline style (inside an HTML element)
  2. External and internal style sheets (in the head section)
  3. Browser default

## CSS – type priority

- An inline style has the highest priority, and will override external and internal styles and browser defaults.
- If the internal style is defined after the link to the external style sheet, the <h1> elements will be "orange", otherwise it will be blue.

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

# CSS – type priority

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet"
type="text/css" href="mystyle.css">
<style>
h1 { color: orange; }
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This document has an external
stylesheet, and internal style</p>
</body>
</html>
```

```
/* mystyle.css */
body {
    background-
color: lightblue; }
h1 {
    color: navy;
margin-left: 20px; }
```

**This is a heading**

This document has an external stylesheet, and internal style

# Reality Checks

Time toProgram

from CANVAS Files-Class Notes-day31\_HTML Select and download the files:

**day33\_css.pdf, mystyle.css**

To YOUR folder: **day31\_html**

•After Class:

- Complete your Reality Check(s) tonight before (11:59 PM)
- Complete your Homework (Due Thursday, 4/17/25)
- Complete your Coding Miniquiz (Due Sunday, 4/20/25 @11:59 PM)
- Read the textbook
- Course Extra Credit: Codingbat ([Extra Credit Opportunity Announcement](#))
- Practice Python: W3schools, Real Python, Codecademy