Lecture 4

CSS

Intro to CSS

Topics

- History
- What is CSS?
  - CSS Cascade
- CSS Rulesets
- CSS Syntax
  - Comments
- CSS Selectors
  - Attributes and Selectors
  - Using Selectors

History

Invention of CSS

- CSS was first proposed by **Håkon Wium Lie** on October 10, 1994. At the time, Lie was working with **Tim Berners-Lee** at **CERN**.

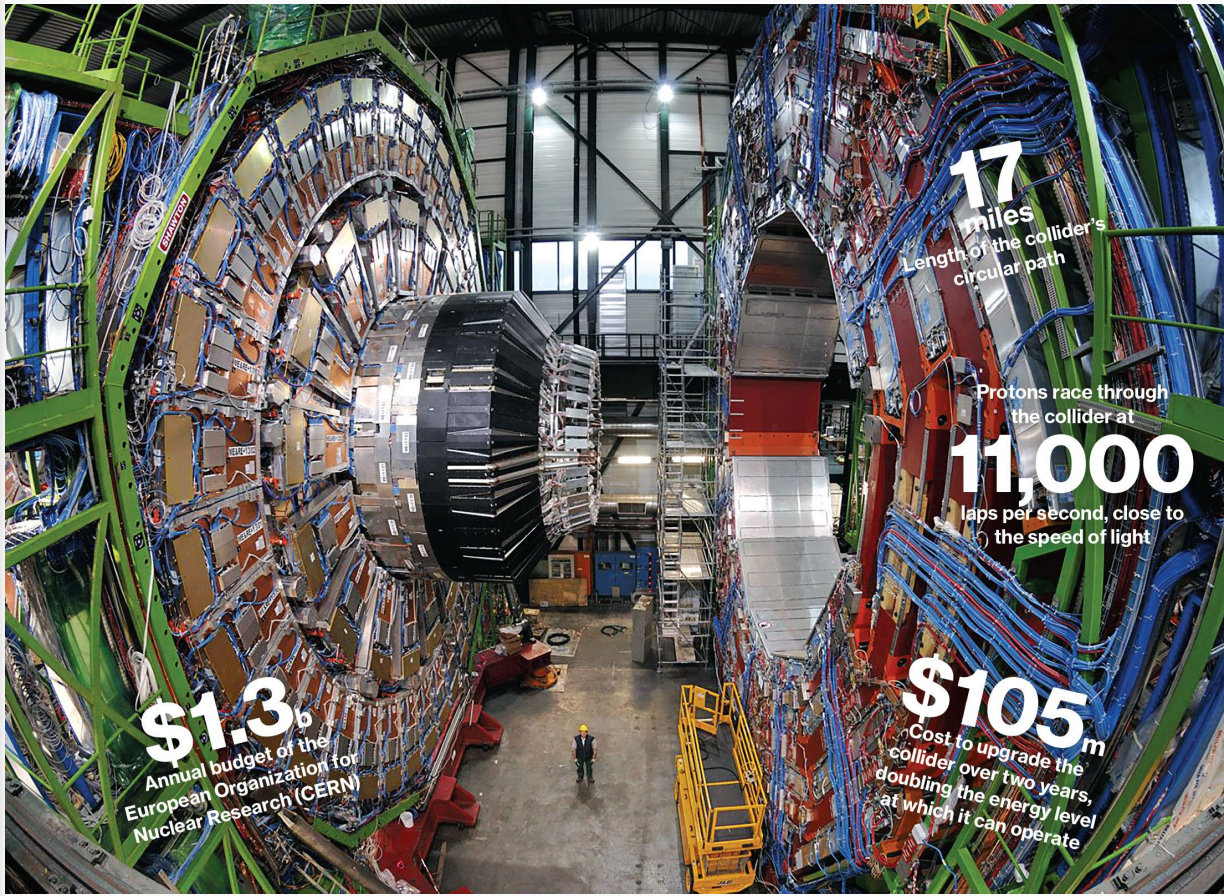
- Recommendations that are used to keep the web experience consistent among different browsers.

- They could see the Web diverge in different directions with browsers supporting different specifications. The battles within the HTML ERB were long and hard, but CSS level 1 finally emerged as a W3C Recommendation in December 1996.

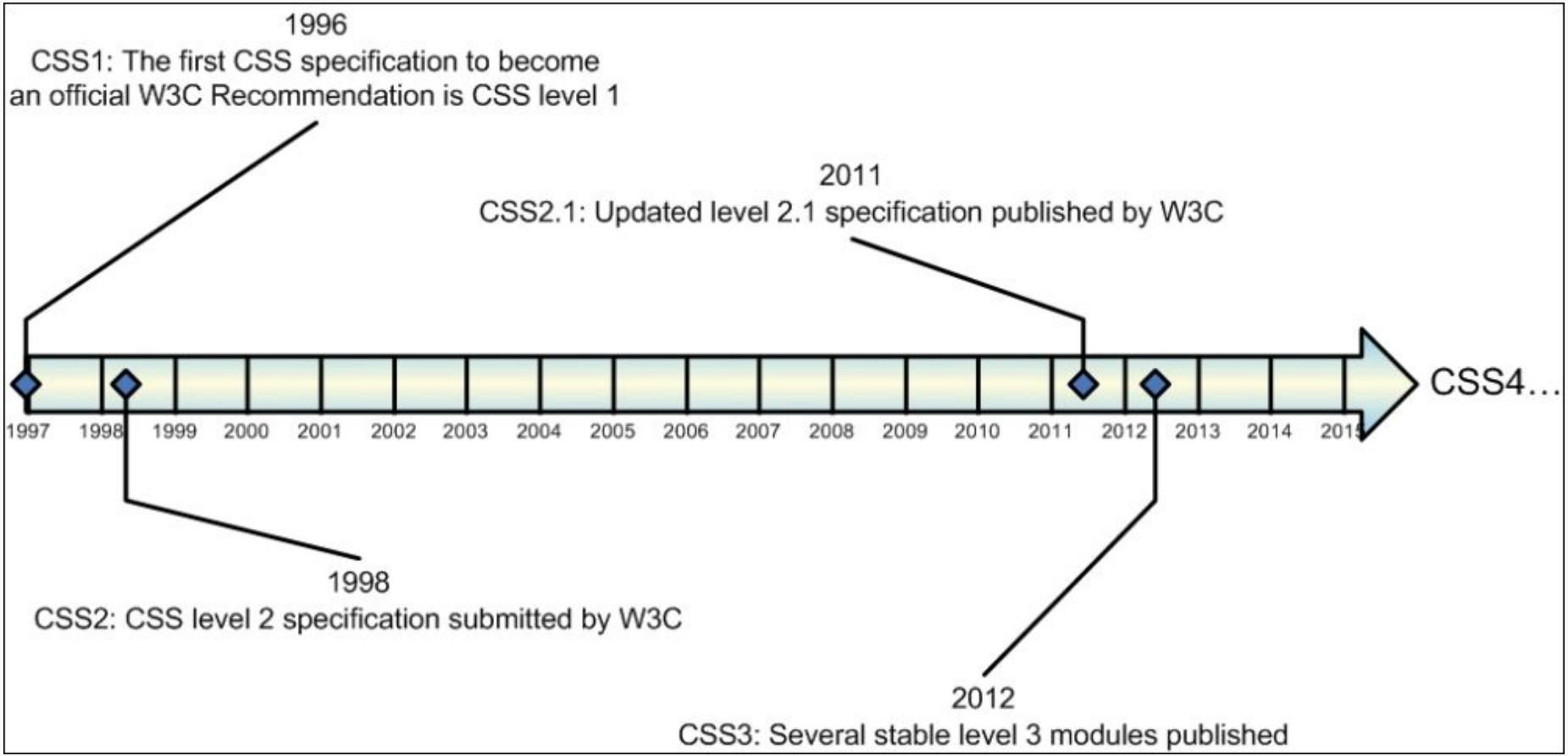
- https://www.w3.org/

What is CERN?

Remember the Atom Smasher (Large Haron Collider)?



CSS Timeline



What is CSS?

What is CSS?

- Like HTML, CSS is not a programming language. It's not a markup language either. **CSS is a style sheet language.** CSS is what you use to selectively style HTML elements. For example, this CSS selects paragraph text, setting the color to red:

```
p {
    color: red;
}
```

- 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 web pages all at once
- External stylesheets are stored in CSS files

Why use CSS?

- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Example

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

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

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 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>TENET is playing theatres</h1>
<p>Good, this is my first time out!</p>
```

- 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 Cascade

What does Cascading mean?

Different levels of style sheets can be applied to HTML

Default Browser style (User Agent)

- most general: The default if there is no specific style sheet

Author style

- Web page author creates a style sheet to give HTML a particular look
- Takes precedence over the Default Browser style
- Most common

User style

- The end user creates a style sheet to override Author and
- Default Browser styles
- One reason might be to accomodate impaired vision (larger font)
- Least common

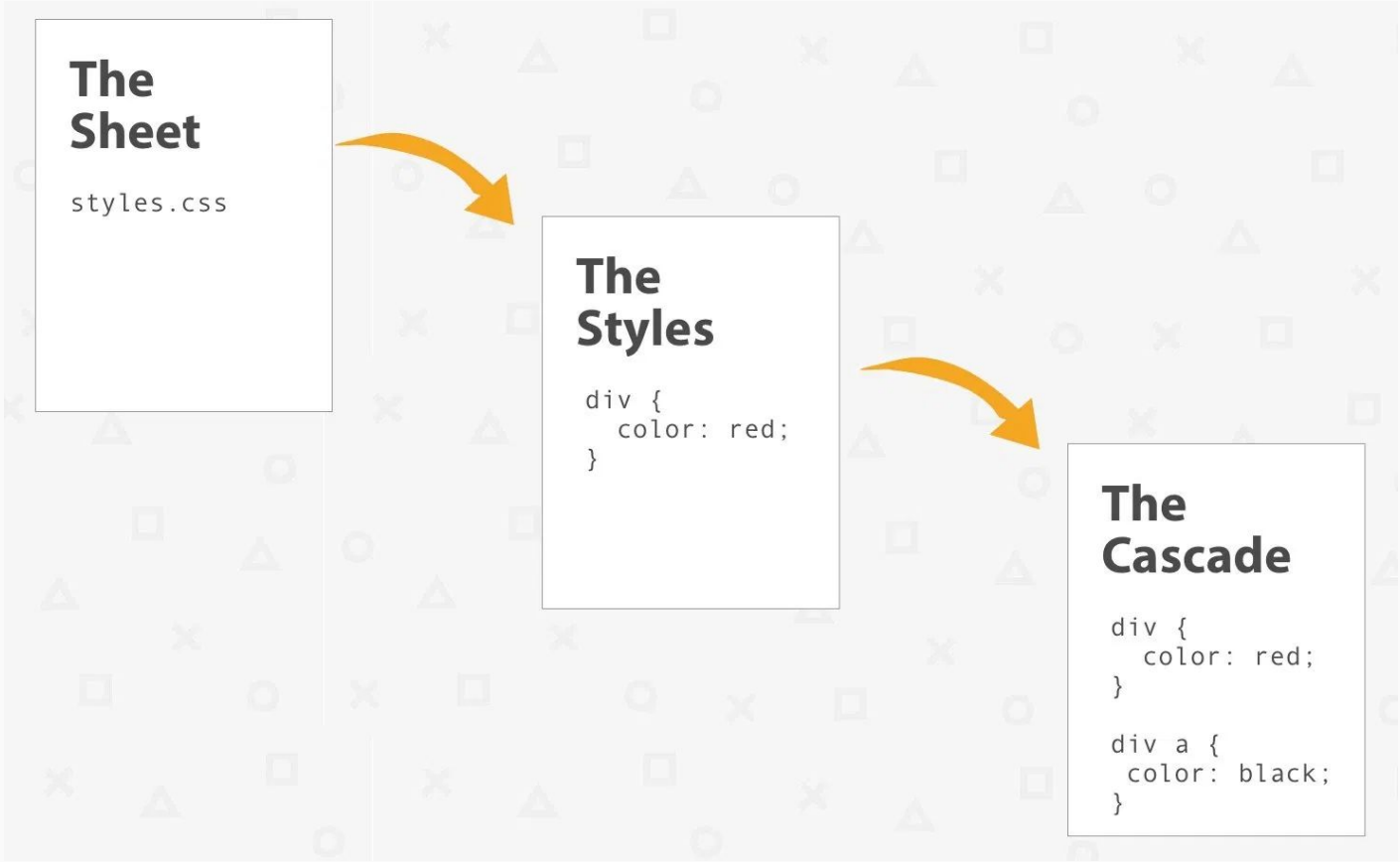
How does the CSS Cascade work?

- The **CSS Cascade** is the algorithm by which the browser decides which **CSS** styles to apply to an element — a lot of people like to think of this as the style that “wins”. ... If a **CSS** rule wins at a higher-priority level, that's the rule that gets wins.

- The CSS cascade takes a few of these attributes and assigns each of them a weight. If a CSS rule wins at a higher-priority level, that’s the rule that gets wins.

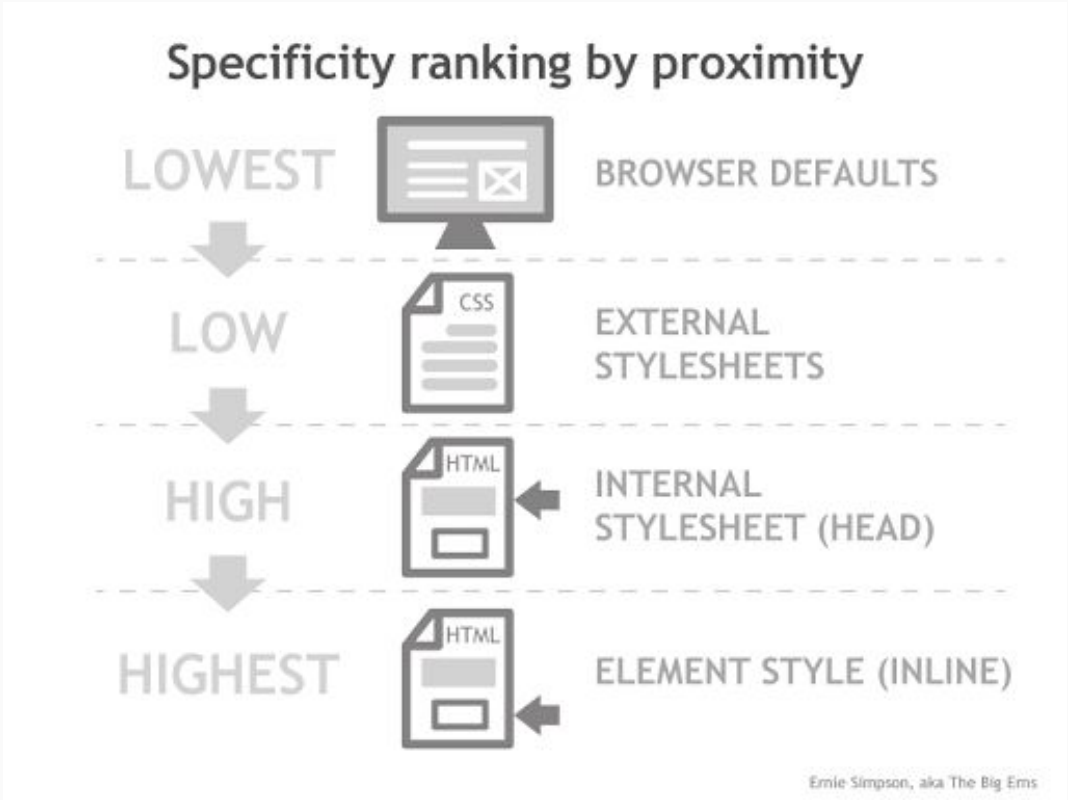
- However, if there are two rules still in conflict at a given weight, the algorithm will continue to “cascade down” and check the lower-priority attributes until it finds one that wins.

CSS Cascade


The Sheet
styles.css
The Styles
div {
color: red;
}
The Cascade
div {
color: red;
}
div a {
color: black;
}

CSS Cascade cont.

- The power of CSS is found in the "cascade" which is the combination of the browser's default styles, external style sheets, embedded, inline and even user-defined styles.

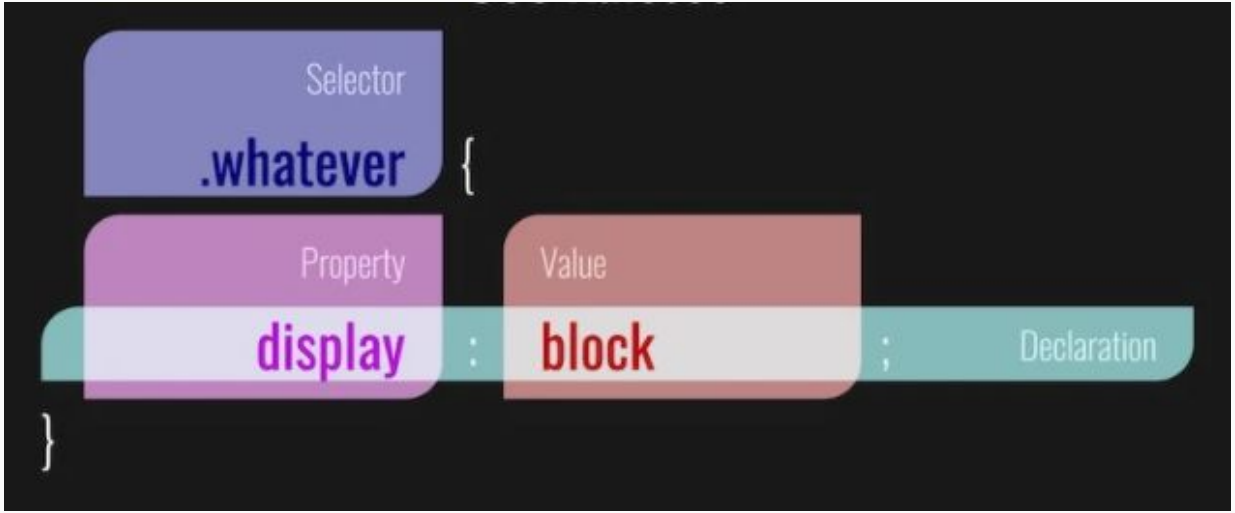


CSS Ruleset

CSS Rulesets

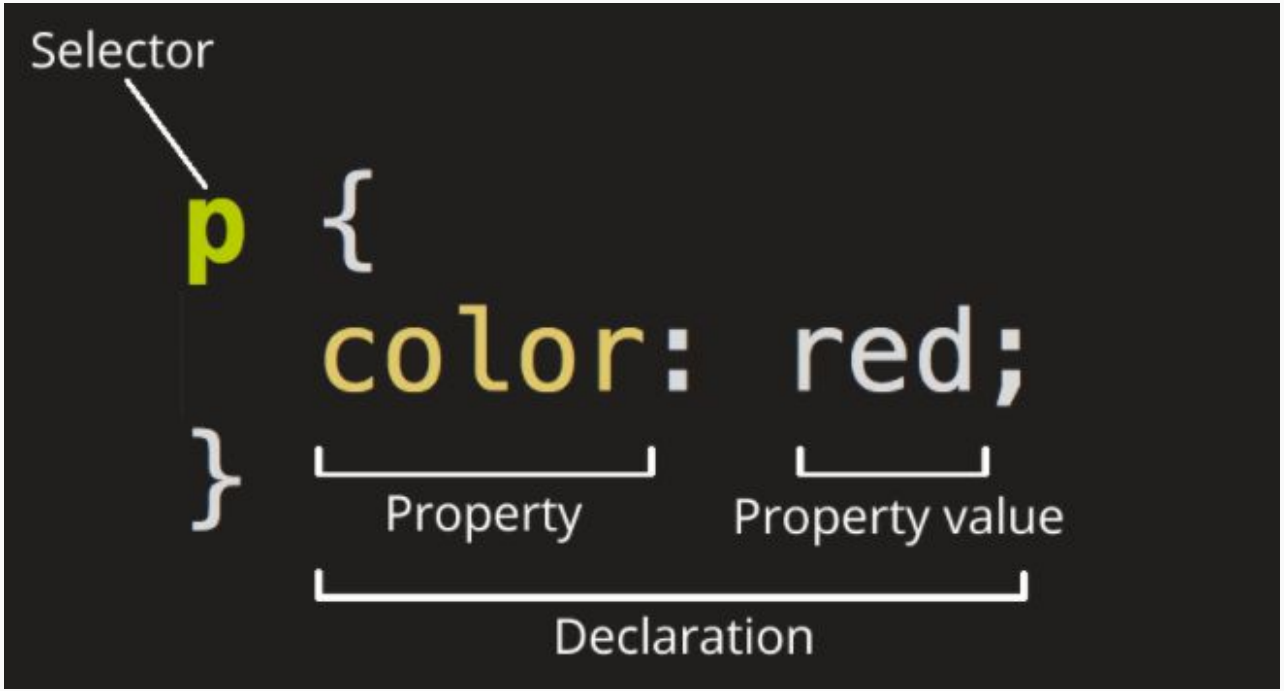
- A CSS rule-set consists of a selector and a declaration block:

- The foundation of writing scalable CSS for our web application is a ruleset. A ruleset contains one or more **selector**(s) and a declaration block with a collection of declarations. Declarations contain properties and values in a **key/value pair** syntax.



CSS Rulesets cont.

- The whole structure is called a **ruleset.** (The term *ruleset* is often referred to as just *rule.)*



CSS Syntax

CSS Syntax

Selector

- This is the HTML element name at the start of the ruleset. It defines the element(s) to be styled (in this example, `<p>` elements). To style a different element, change the selector.

Declaration

- This is a single rule like `color: red;`. It specifies which of the element's **properties** you want to style.

Properties

- These are ways in which you can style an HTML element. (In this example, `color` is a property of the `<p>` elements.) In CSS, you choose which properties you want to affect in the rule.

Property value

- To the right of the property—after the colon—there is the **property value**. This chooses one out of many possible appearances for a given property. (For example, there are many `color` values in addition to `red`.)

CSS Syntax cont...

Note the other important parts of the syntax:

- Apart from the **selector,** each ruleset must be wrapped in curly braces. (`{ }`)
- Within each **declaration,** you must use a colon (`:`) to separate the property from its value or values.
- Within each **ruleset**, you must use a semicolon (`;`) to separate each declaration from the next one.

To modify multiple property values in one ruleset, write them separated by semicolons, like this:

```
p {
    color: red;
    width: 500px;
    border: 1px solid black;
}
```

Selecting Multiple Elements

- Selectors are the interface that CSS uses to match our rulesets to elements in our HTML document. Multiple selectors and rulesets might apply to the same element, and this is where the 'cascade' part of CSS comes into play.

- You can also select multiple elements and apply a single ruleset to all of them. Separate multiple selectors by commas.

CSS Comments

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 is placed inside the `<style>` element, and starts with `/*` and ends with `*/`:

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


  p {
    color: red;  /* You can add comments to side */
  }
```

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

p {
    color: red;
  }
```

- We use a combination of HTML and CSS comments:

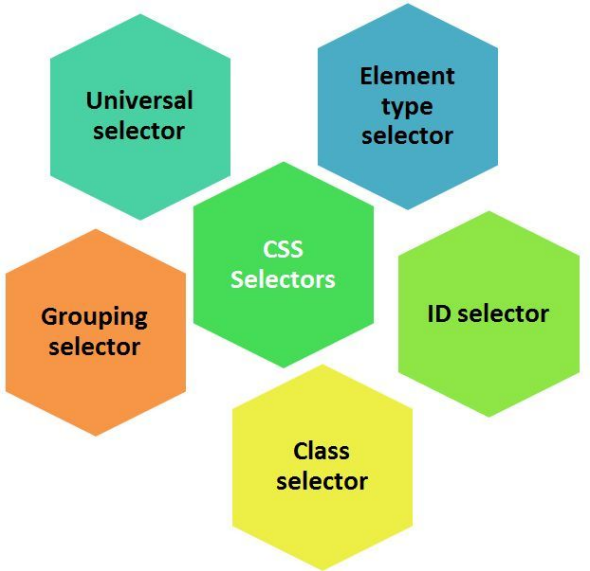
```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red; /* Set text color to red */
}
</style>
</head>
<body>

<!-- These paragraphs will be red -->
<p>Hello World!</p>

</body>
</html>
```

CSS Selectors

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 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)

CSS Element Selector

- The **element selector** selects HTML elements based on the element name.
- CSS Element Selector is also known as a **Tag** or **Type** selector. Element Selector in CSS tries to match the HTML elements having the same name.
- Here, all **<p>** elements on the page will be center-aligned, with a **red text color**:

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

```html
<p>Hello World.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
```

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.
- The CSS rule below will be applied to the HTML element with **id="para1":**

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

```html
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
```

CSS **Class** Selector

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

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

```
<h1 class="red-text">This text</h1>
<p class="red-text">is red.</p>
```

CSS **Class** Selector cont..

- Note: A class name cannot start with a number!
- You can also specify that only specific HTML elements should be affected by a class.
- HTML elements can also refer to more than one class.

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

```html
<p class="center large">
  This paragraph refers to two classes
</p>
```

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

<h1>Hello world!</h1>

<p>Every element.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

CSS **Grouping** Selector

- The grouping selector selects all the HTML elements with the same style definitions.

- The following CSS code (the h1, h2, and p elements have the same style definitions):

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

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

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

- It will be better to group the selectors, to minimize the code.
- To group selectors, separate each selector with a comma.

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

All CSS Simple Selectors

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