# Introduction to HTML and CSS

Week 2

# Agenda

1. Homework Review
2. Introduction to HTML
3. Introduction to CSS
4. “The perfect setup”

# Homework Review

- **Read**

  **Only the following sections!!: ‘What is HTML’, ‘Anatomy of an HTML element’, ‘Attributes’ and ‘Anatomy of an HTML document’**

  https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started

- **Watch**

  https://www.youtube.com/watch?v=PlxWf493en4

- **Do**

  Try to follow the video from above and bring to the lecture next week your very first website. By this, we mean create your first HTML document with at least a title, a paragraph, an image and an anchor link

# Introduction to HTML

HTML (HyperText Markup Language) is a markup language that tells web browsers how to structure the web pages you visit. HTML consists of a series of elements, which you use to enclose, wrap, or mark up different parts of content to make it appear or act in a certain way. These elements are built by using tags, which can make content into a hyperlink to connect to another page, italicize words, and more.

# Introduction to HTML

HTML (HyperText Markup Language) is a markup language that tells web browsers how to structure the web pages you visit. HTML consists of a series of elements, which you use to enclose, wrap, or mark up different parts of content to make it appear or act in a certain way. These elements are built by using tags, which can make content into a hyperlink to connect to another page, italicize words, and more.

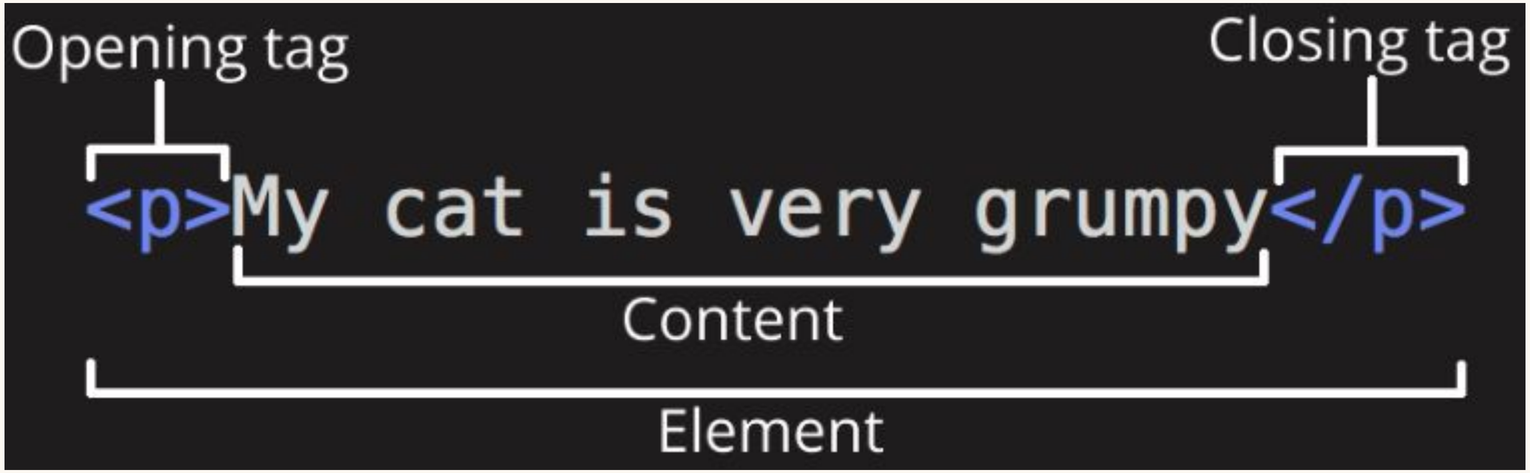
# Introduction to HTML

HTML (HyperText Markup Language) is a markup language that tells web browsers how to structure the web pages you visit. HTML consists of a series of elements, which you use to enclose, wrap, or mark up different parts of content to make it appear or act in a certain way. These elements are built by using tags, which can make content into a hyperlink to connect to another page, italicize words, and more.

# Introduction to HTML

HTML (HyperText Markup Language) is a markup language that tells web browsers how to structure the web pages you visit. HTML consists of a series of elements, which you use to enclose, wrap, or mark up different parts of content to make it appear or act in a certain way. These **elements are built by using tags**, which can make content into a hyperlink to connect to another page, italicize words, and more.

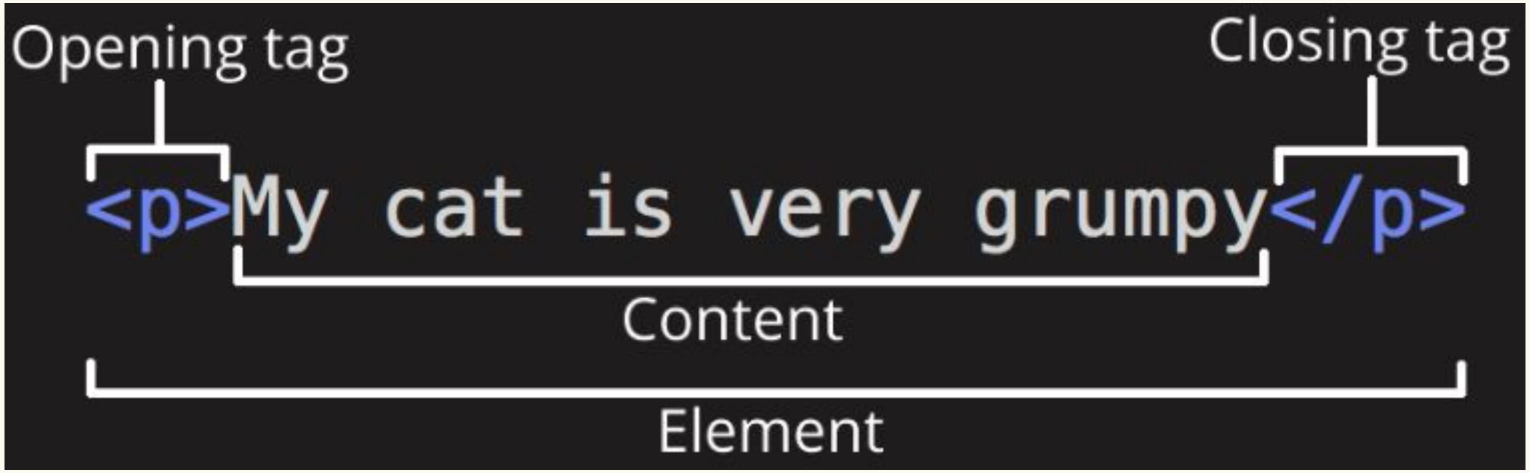
# HTML Elements



## Parts of an HTML element

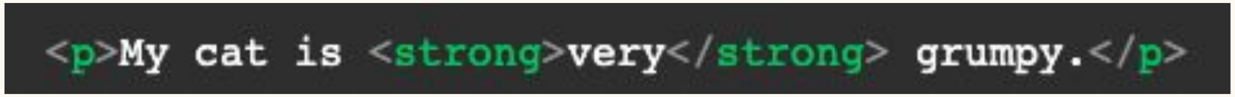
- The opening tag
- The content
- The closing tag

# HTML Elements



```html
<p>My cat is very grumpy</p>
```

## Nested elements



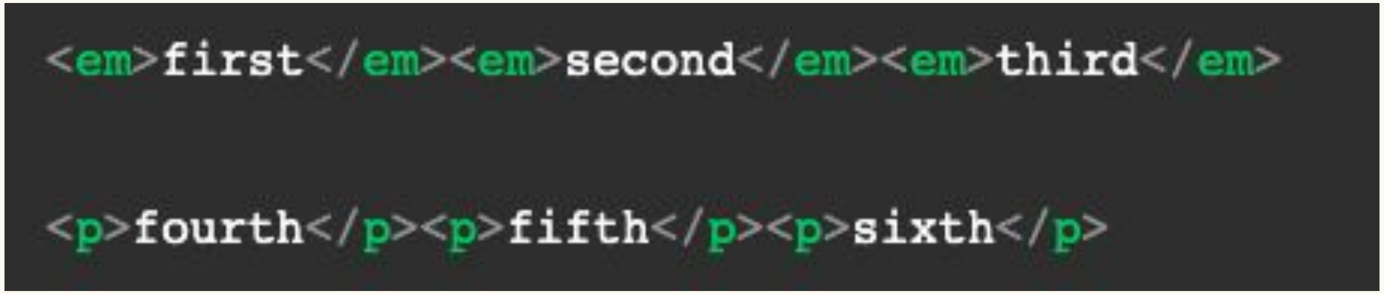
```html
<p>My cat is <strong>very</strong> grumpy.</p>
```

# Block vs Inline elements

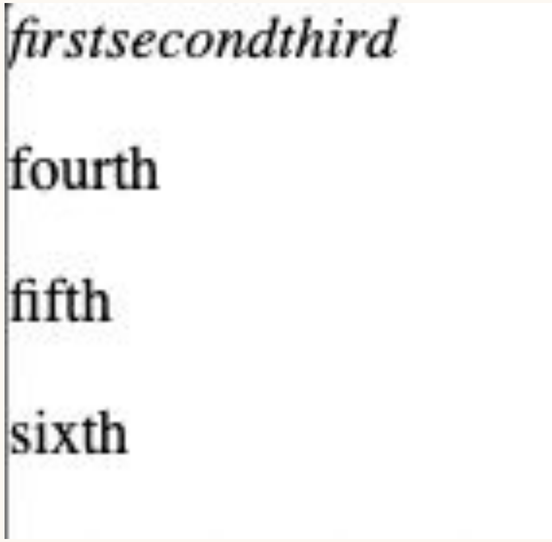
- Block-level elements: A block-level element appears on a new line following the content that precedes it. Block-level elements are usually structural elements on the page. For example, a block-level element might represent headings, paragraphs, lists, navigation menus, or footers.

- Inline elements are contained within block-level elements, and surround only small parts of the document's content (not entire paragraphs or groupings of content). An inline element will not cause a new line to appear in the document. It is typically used with text, for example an <a> element creates a hyperlink, and elements such as <em> or <strong> create emphasis.

# Block vs Inline elements



```html
<em>first</em><em>second</em><em>third</em>

<p>fourth</p><p>fifth</p><p>sixth</p>
```



*firstsecondthird*

fourth

fifth

sixth

# Block vs Inline elements

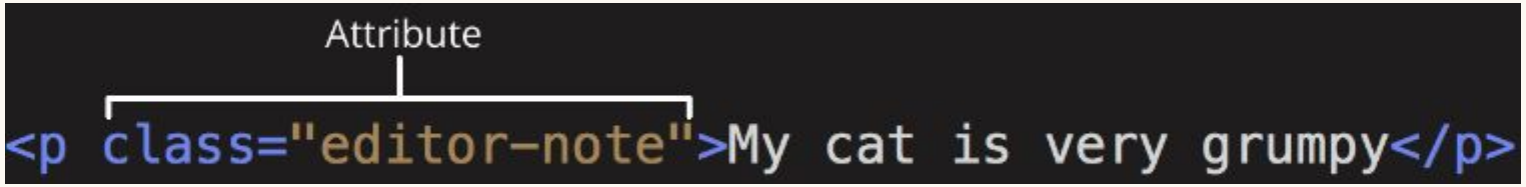
- <address>
- <article>
- <div>
- <figcaption>
- <figure>
- <footer>
- <form>
- <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
- <header>
- <main>
- <nav>
- <ol>
- <p>
- <section>
- <ul>

- <a>
- <audio> (if it has visible controls)
- <b>
- <br>
- <button>
- <canvas>
- <em>
- <embed>
- <i>
- <iframe>
- <img>
- <input>
- <label>
- <picture>
- <span>
- <strong>
- <svg>
- <template>
- <textarea>
- <u>
- <video>

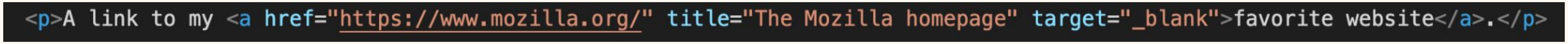
# Attributes

Attributes contain extra information about the element that won't appear in the content. They should have:

- A space between it and the element name. (For an element with more than one attribute, the attributes should be separated by spaces too.)
- The attribute name, followed by an equal sign.
- An attribute value, wrapped with opening and closing quote marks.



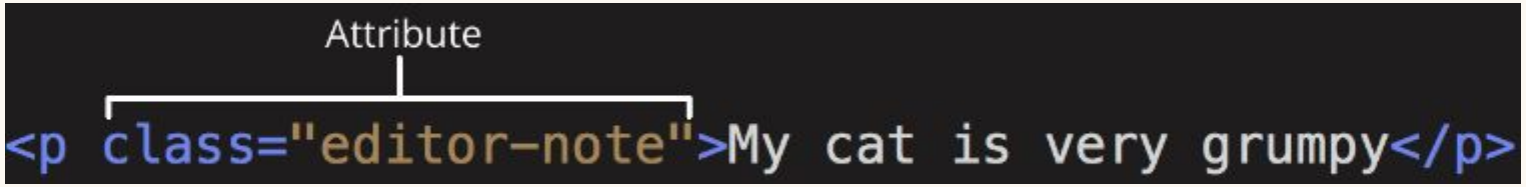
```html
<p class="editor-note">My cat is very grumpy</p>
```



```html
<p>A link to my <a href="https://www.mozilla.org/" title="The Mozilla homepage" target="_blank">favorite website</a>.</p>
```

# HTML class attribute

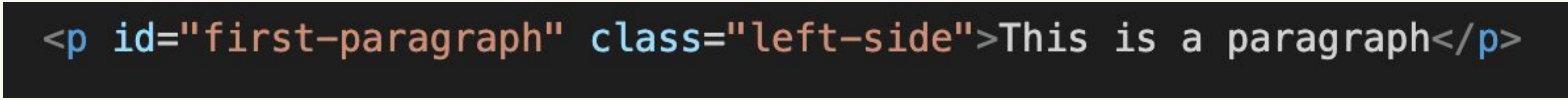
- The class attribute can be used on any HTML element.
- All HTML class attribute specifies one or more class names for an element.
- Different HTML elements can point to the same class name.
- Classes are used by CSS and JavaScript to select and access specific elements.
- The class name is case sensitive.
- You can access an element with a specific class in CSS using '.'



```
<p class="editor-note">My cat is very grumpy</p>
```

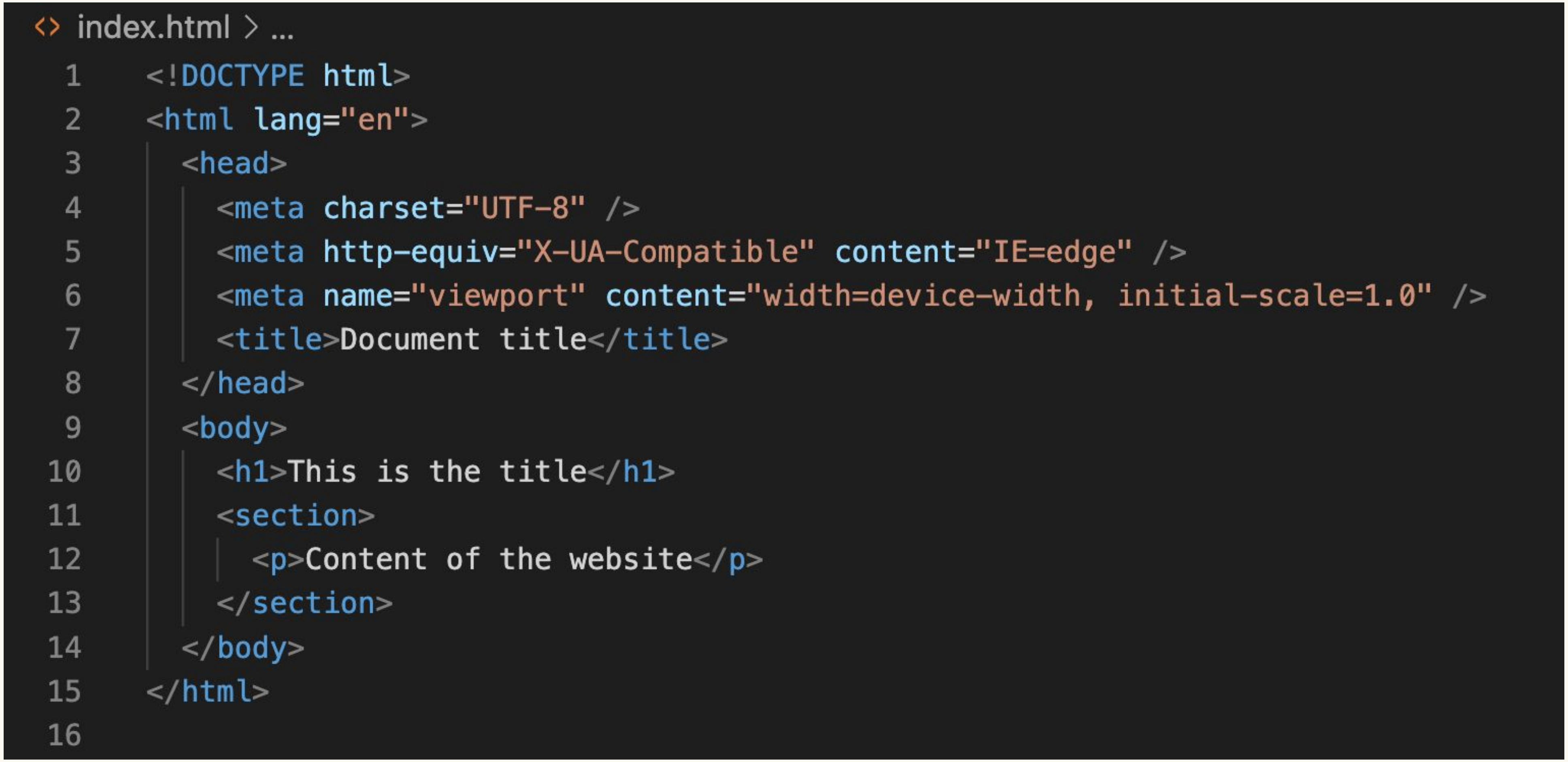
# HTML id attribute

- The id attribute is used to specify a unique id for an HTML element.
- The value of id attribute must be unique within the HTML document.
- The id attribute is used by CSS and JavaScript to select and access specific elements.
- The id is case sensitive.
- You can access an element with a specific id in CSS using #



```
<p id="first-paragraph" class="left-side">This is a paragraph</p>
```

# Anatomy of an HTML document



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document title</title>
  </head>
  <body>
    <h1>This is the title</h1>
    <section>
      <p>Content of the website</p>
    </section>
  </body>
</html>

```

# Anatomy of an HTML document

- <html></html>: The <html> element. This element wraps all the content on the page. It is sometimes known as the root element.

- <head></head>: The <head> element. This element acts as a container for everything you want to include on the HTML page, that isn't the content the page will show to viewers. This includes keywords and a page description that would appear in search results, CSS to style content and more.

- <meta charset="utf-8">: The <meta> element. This element represents metadata that cannot be represented by other HTML meta-related elements, like <base>, <link>, <script>, <style> or <title>.

- <title></title>: The <title> element. This sets the title of the page, which is the title that appears in the browser tab the page is loaded in. The page title is also used to describe the page when it is bookmarked.

- <body></body>: The <body> element. This contains all the content that displays on the page, including text, images, videos, games, playable audio tracks, or whatever else.

# Exercise

**< 10 minutes >**

1. Install the VS Code extensions needed (check next slide)
2. Create a website where you display a list with 3 of your favourite things/activities/topics.

Each topic has to have title, short description and image.

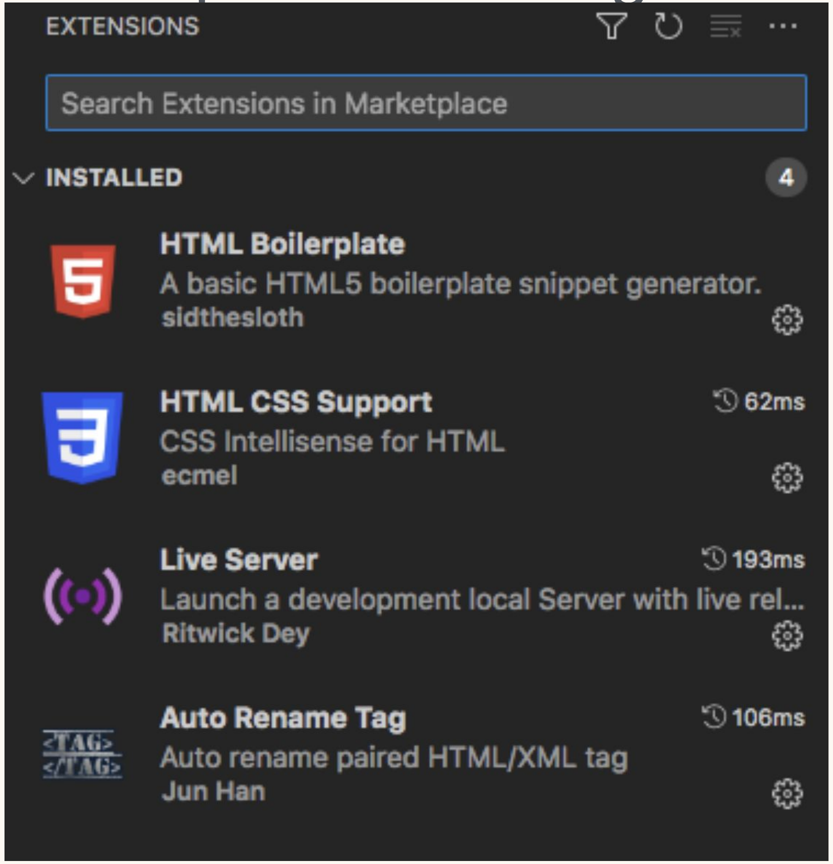
Hint: google - unordered list html

More on tags

https://www.w3schools.com/tags/

or type in google:

HTML tags


EXTENSIONS

Search Extensions in Marketplace

INSTALLED 4

**HTML Boilerplate**
A basic HTML5 boilerplate snippet generator.
sidthesloth

**HTML CSS Support** 62ms
CSS Intellisense for HTML
ecmel

**Live Server** 193ms
Launch a development local Server with live rel...
Ritwick Dey

**Auto Rename Tag** 106ms
Auto rename paired HTML/XML tag
Jun Han

# Break

# What is CSS

- CSS stands for Cascading Style Sheets.
- CSS is the language we use to style a Web page.
- 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 Zen garden

# How to add CSS to your HTML ?

There are 3 ways of inserting stylesheet.

- External CSS
- Internal CSS
- Inline CSS

# Add external CSS

There are mainly 3 steps you need to follow to ensure your CSS is linked to your HTML

1. Create a file in the same folder as your HTML document and save it as styles.css. The .css extension shows that this is a CSS file.

1. Link the stylesheet styles.css to index.html by adding the following line somewhere inside the <head> of the HTML document:

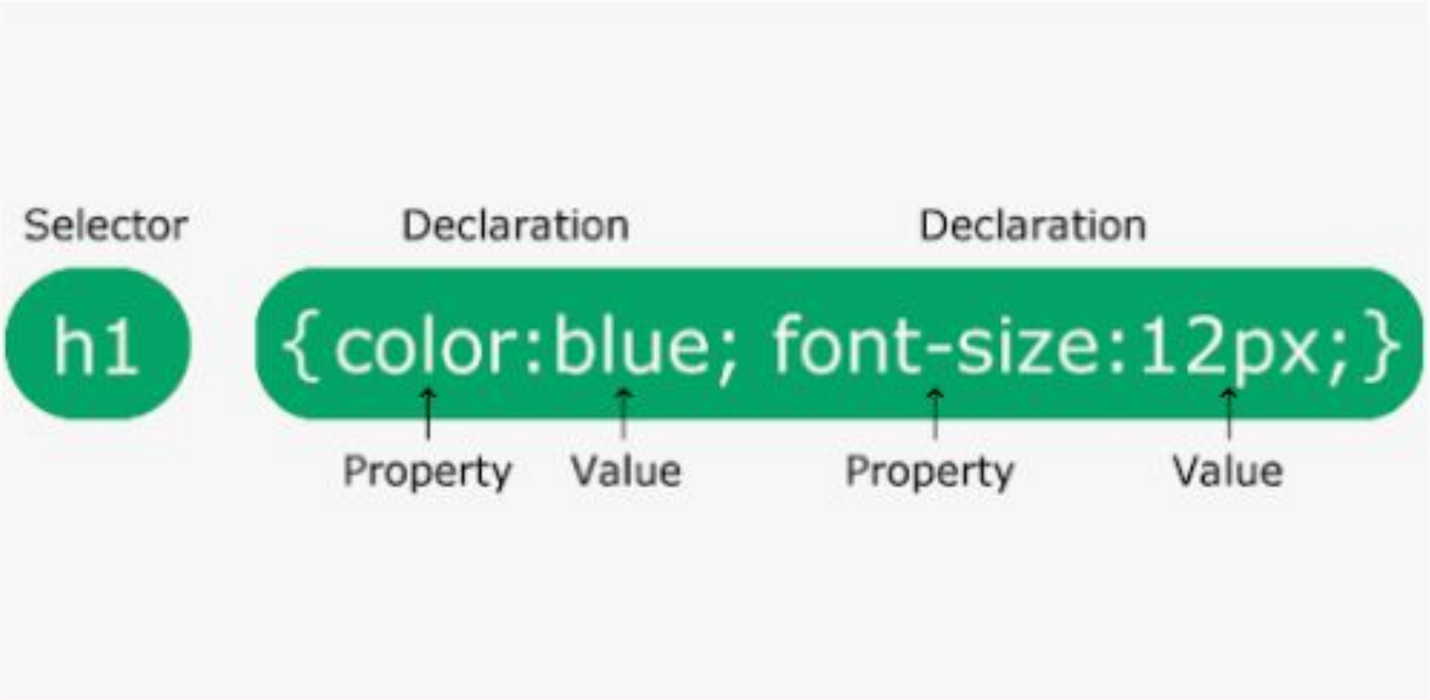
```
<link rel="stylesheet" href="styles.css">
```

This <link> element tells the browser that we have a stylesheet, using the rel attribute, and the location of that stylesheet as the value of the href attribute.

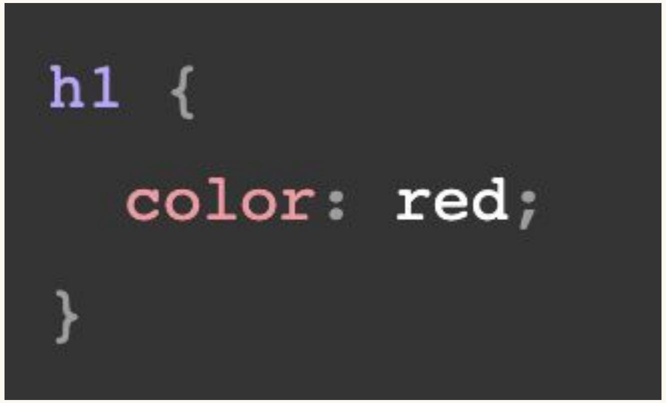
3. Test your setup by adding a style to any HTML element

# CSS Syntax

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



# Styling HTML elements



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

This will target all h1 in the document

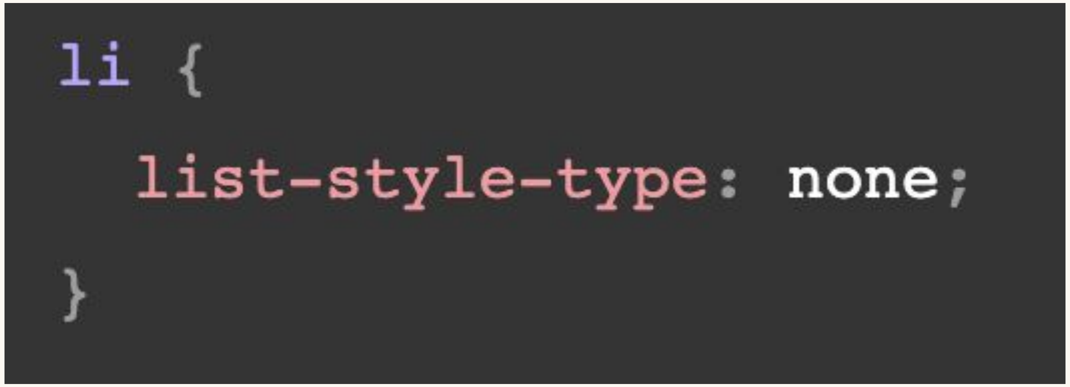
# Change default behaviour of elements

Add this to your index.html without any style

```
<h1>I am a level one heading</h1>

<p>This is a paragraph of text. In the text is a <span>span
element</span>
and also a <a href="http://example.com">link</a>.</p>

<p>This is the second paragraph. It contains an <em>emphasized</em>
element.</p>

<ul>
    <li>Item one</li>
    <li>Item two</li>
    <li>Item <em>three</em></li>
</ul>
```

# Change default behaviour of elements

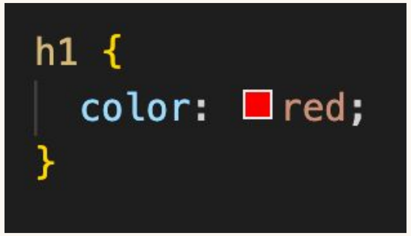
The browser is making the HTML readable by adding some default styling. Headings are large and bold and our list has bullets. This happens because browsers have internal style-sheets containing default styles, which they apply to all pages by default;



```
li {
  list-style-type: none;
}
```

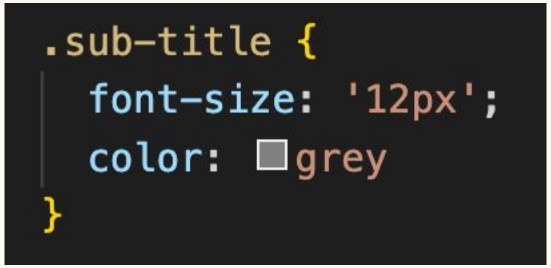
# CSS Selectors: Type, Class and ID

Type

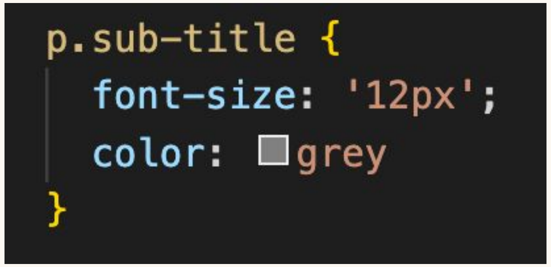


```
h1 {
  color: 🟥red;
}
```

Class

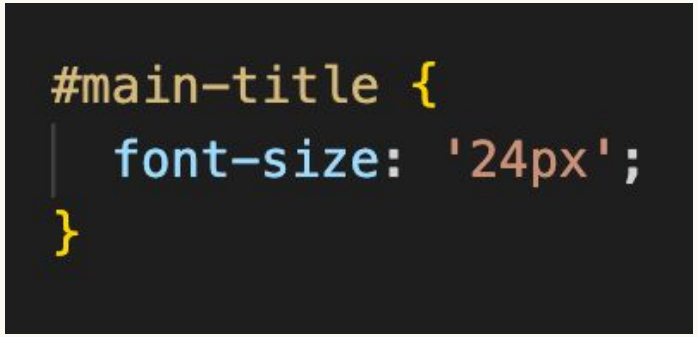


```
.sub-title {
  font-size: '12px';
  color: ◼grey
}
```



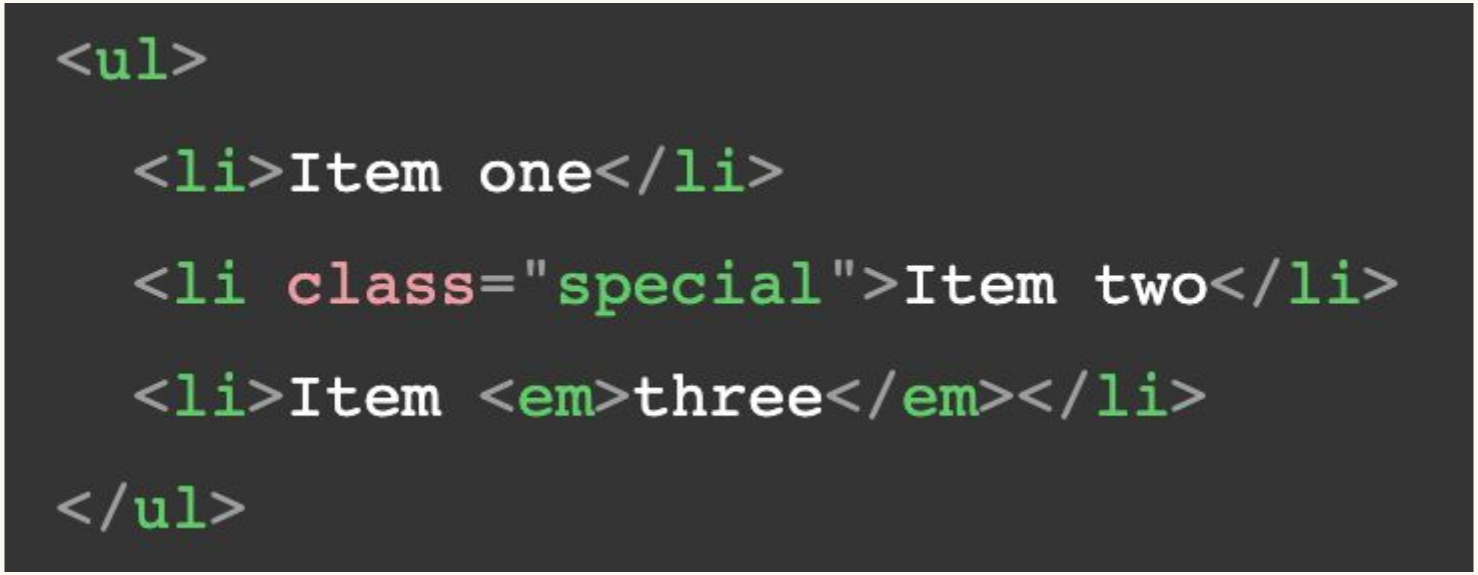
```
p.sub-title {
  font-size: '12px';
  color: ◼grey
}
```

ID

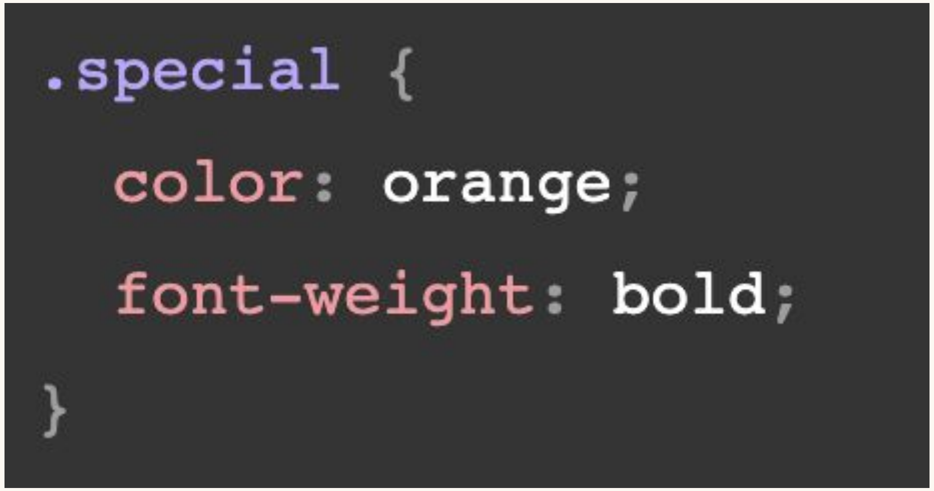


```
#main-title {
  font-size: '24px';
}
```

# Adding a class

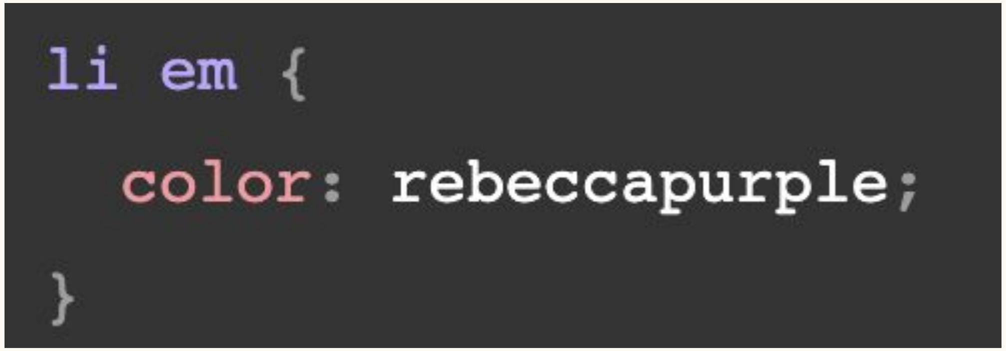


```
<ul>
  <li>Item one</li>
  <li class="special">Item two</li>
  <li>Item <em>three</em></li>
</ul>
```



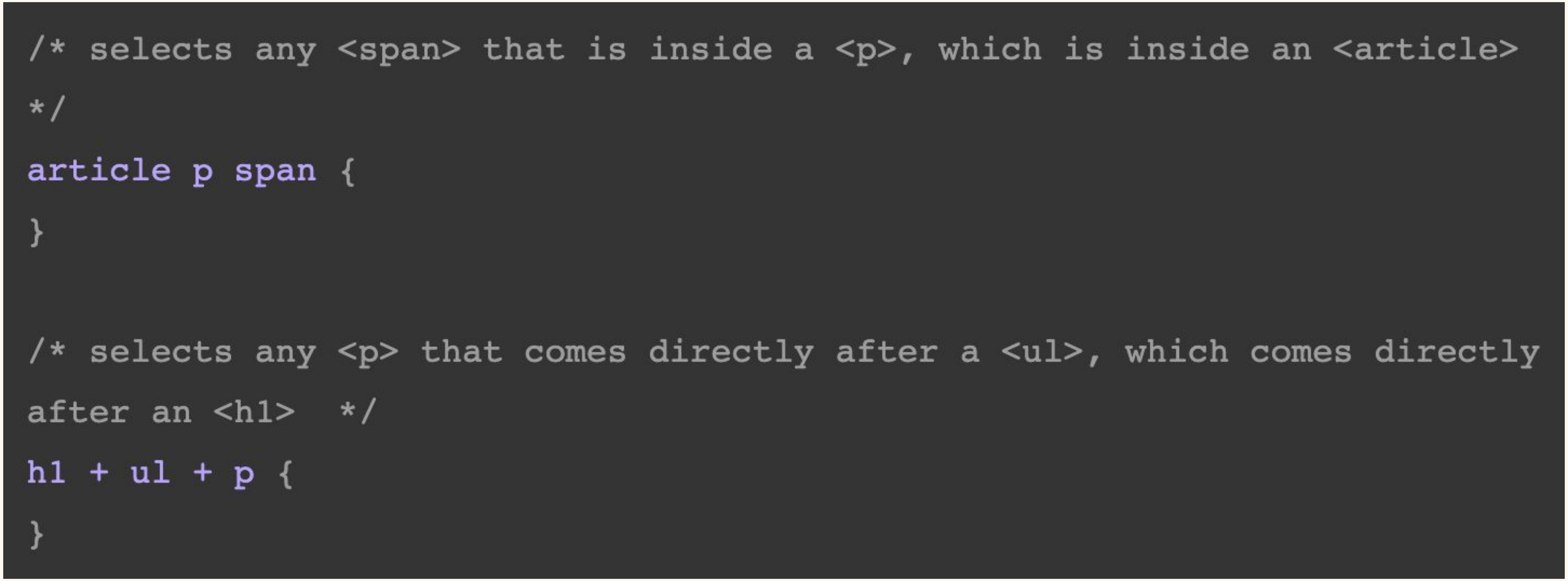
```
.special {
  color: orange;
  font-weight: bold;
}
```

# Styling children or nested elements



```css
li em {
  color: rebeccapurple;
}
```

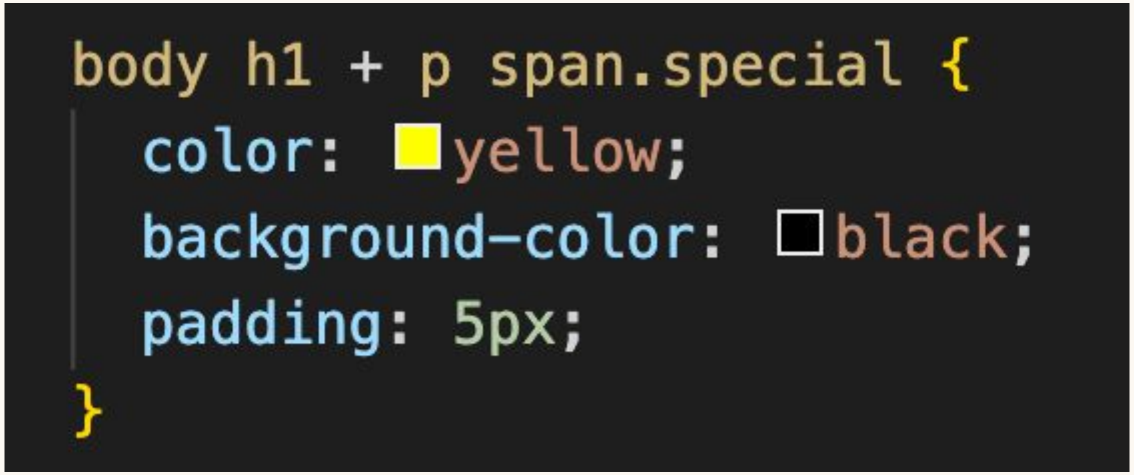
# Combining selectors and combinators



```
/* selects any <span> that is inside a <p>, which is inside an <article>
*/
article p span {
}

/* selects any <p> that comes directly after a <ul>, which comes directly
after an <h1>  */
h1 + ul + p {
}
```

# Combining selectors and combinators

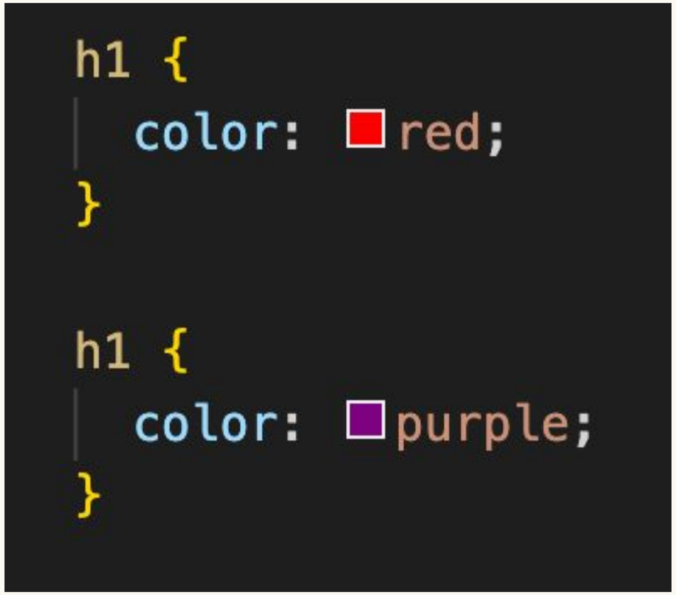
What is this rule applied to?



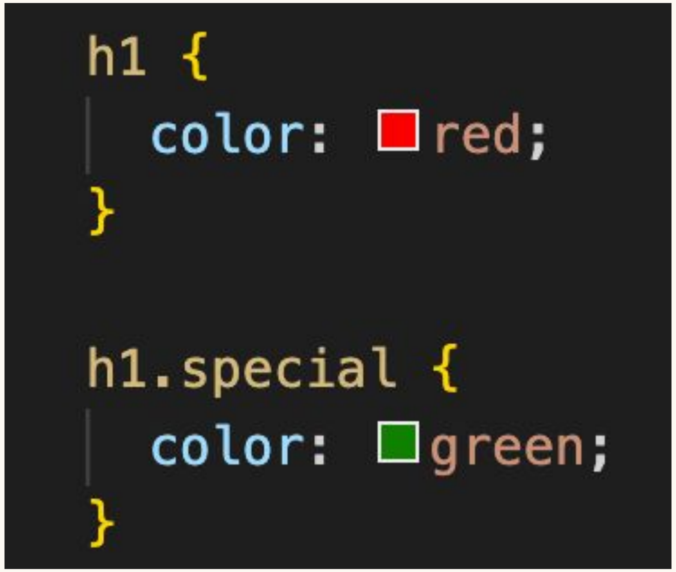
```css
body h1 + p span.special {
  color: yellow;
  background-color: black;
  padding: 5px;
}
```

# Cascade and specificity

Cascade and the closely-related concept of specificity are mechanisms that control which rule applies when there is such a conflict. The rule that's styling your element may not be the one you expect, so you need to understand how these mechanisms work.



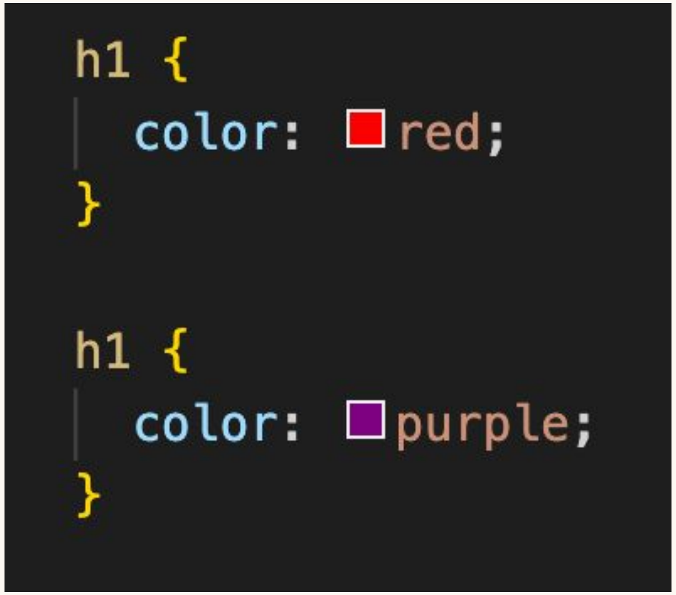
```
h1 {
    color: 🟥red;
}

h1 {
    color: 🟪purple;
}
```



```
h1 {
    color: 🟥red;
}

h1.special {
    color: 🟩green;
}
```

# Cascade

When two rules from the same cascade layer apply and both have equal specificity, the one that is defined last in the stylesheet is the one that will be used.

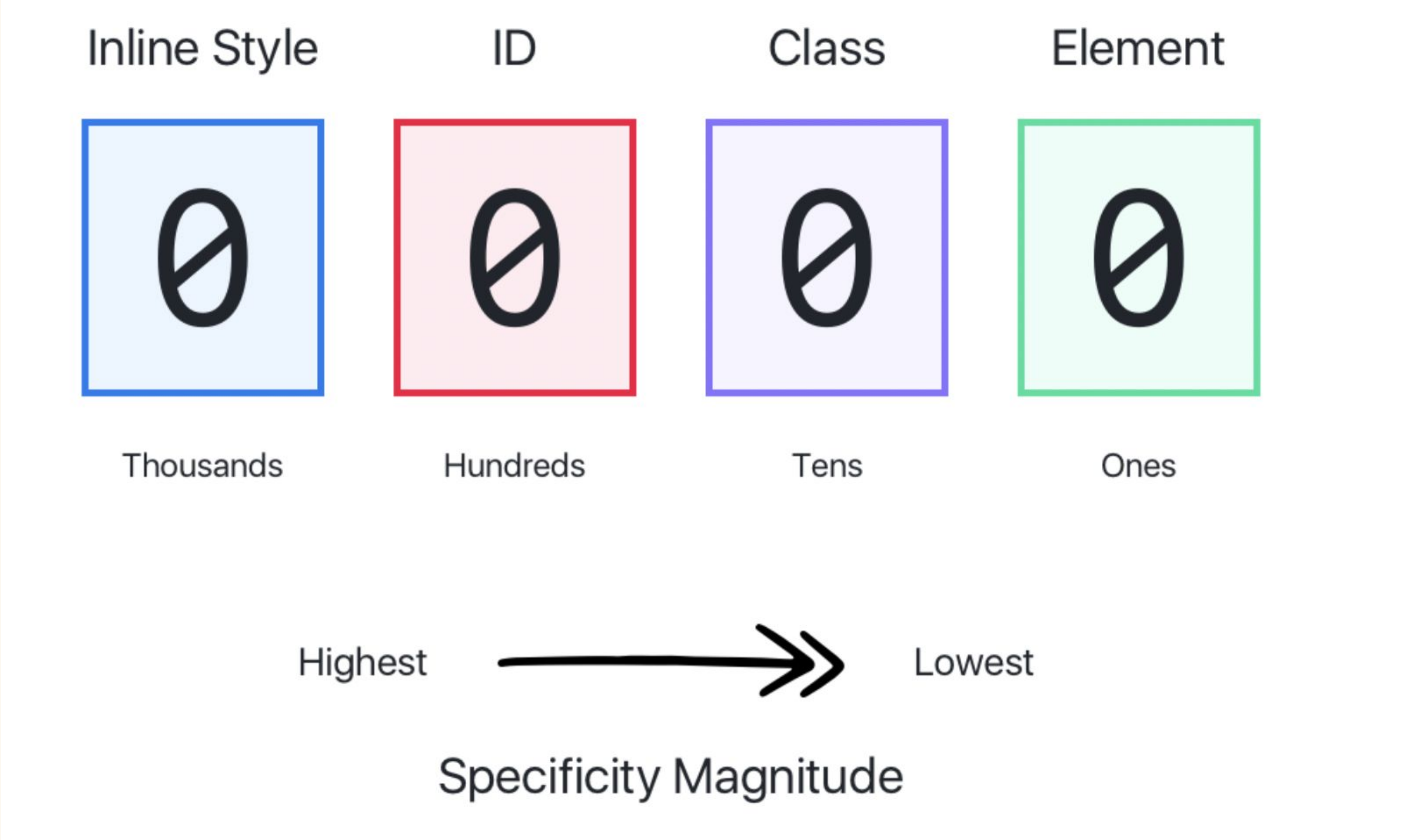


```css
h1 {
  color: red;
}

h1 {
  color: purple;
}
```

# Break

# Specificity

Specificity is the algorithm that the browser uses to decide which property value is applied to an element.



# Exercise

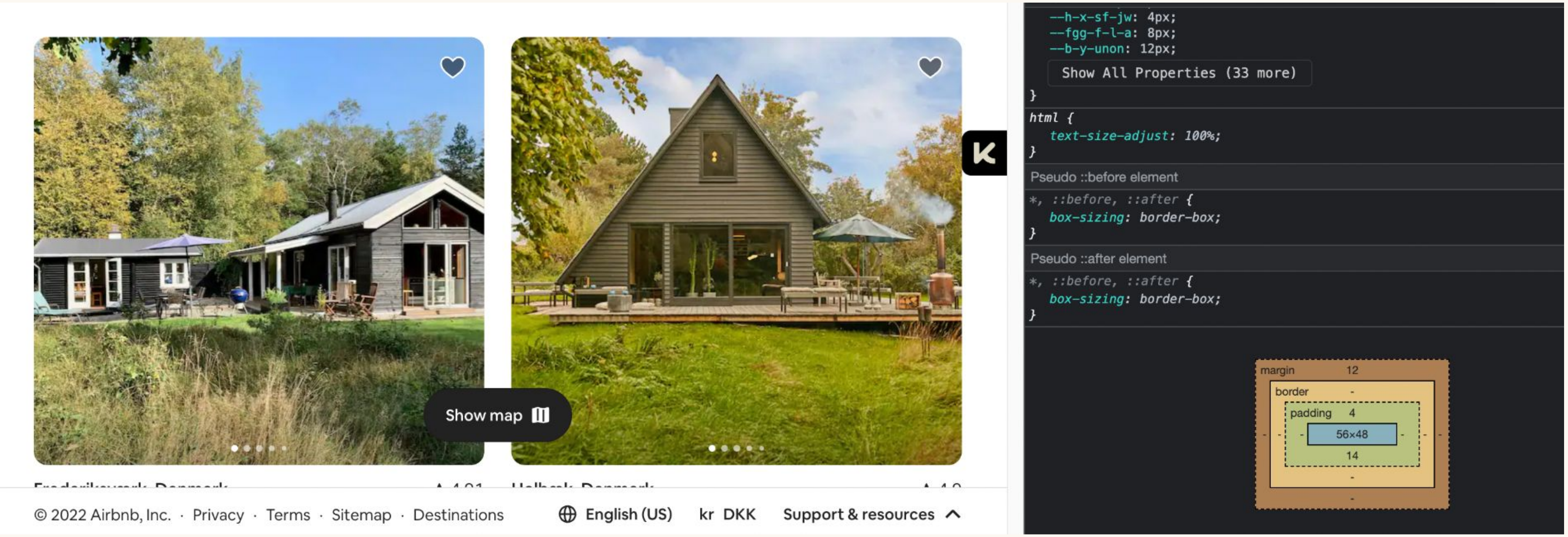
**< 10 minutes >**

Can you find the specificity values for the following CSS selectors?

```
#main-link a {
    background-color: red;
}

#nav-links #first-link a {
    background-color: blue;
}

#outer div ul li a {
    color: yellow;
}
```

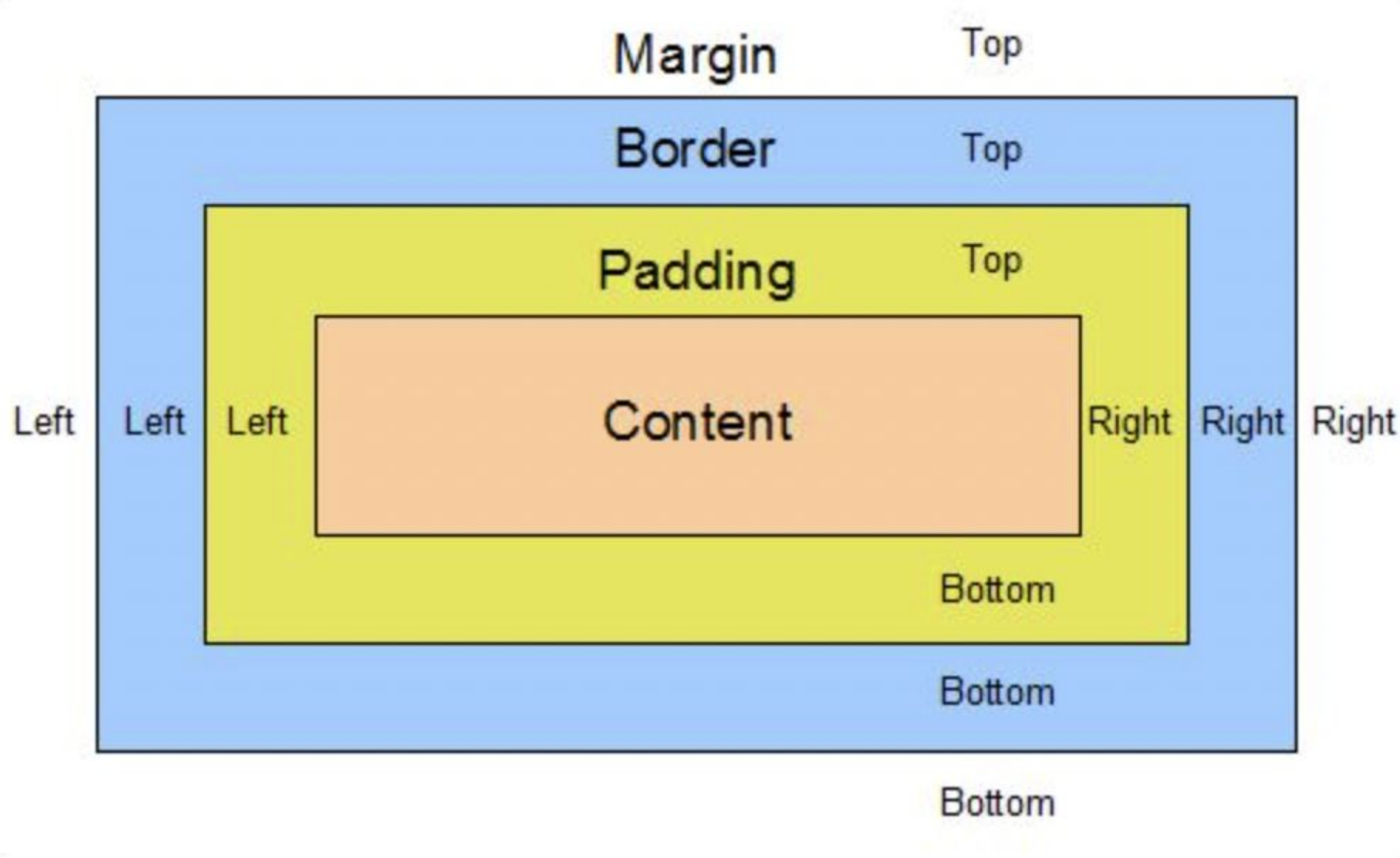
```
#nav div ul li a.special-link {
    color: yellow;
}
```

# Using inspect browsers to see the styling rules

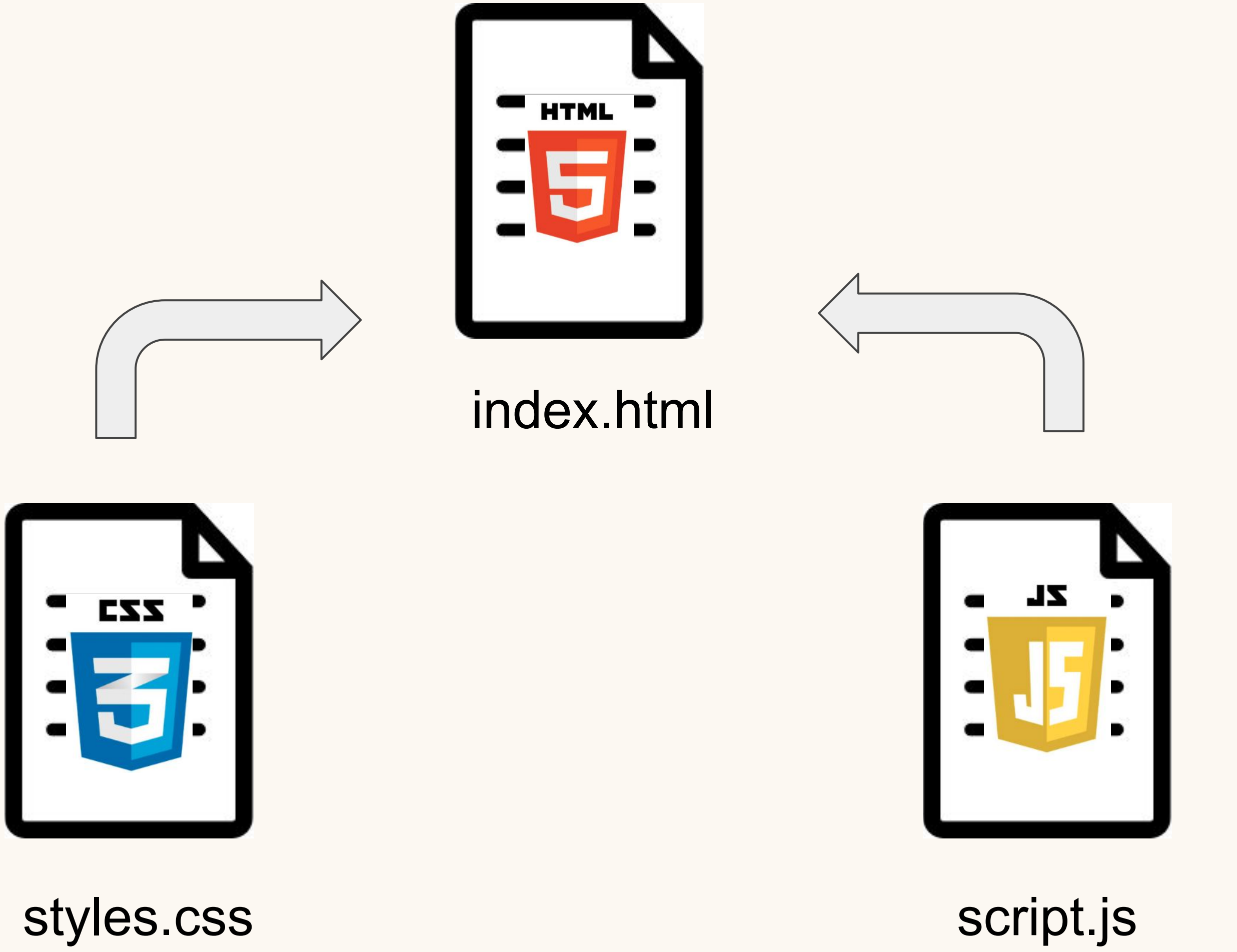


```
   --h-x-sf-jw: 4px;
   --fgg-f-l-a: 8px;
   --b-y-unon: 12px;
   Show All Properties (33 more)
}
html {
   text-size-adjust: 100%;
}
Pseudo ::before element
*, ::before, ::after {
   box-sizing: border-box;
}
Pseudo ::after element
*, ::before, ::after {
   box-sizing: border-box;
}
```

# Margin padding and Borders



# The Perfect Setup



# Homework

https://github.com/ReDISchoolDK/Spring25_Frontend/blob/main/Week-02_HTML-CSS/homework/README.md