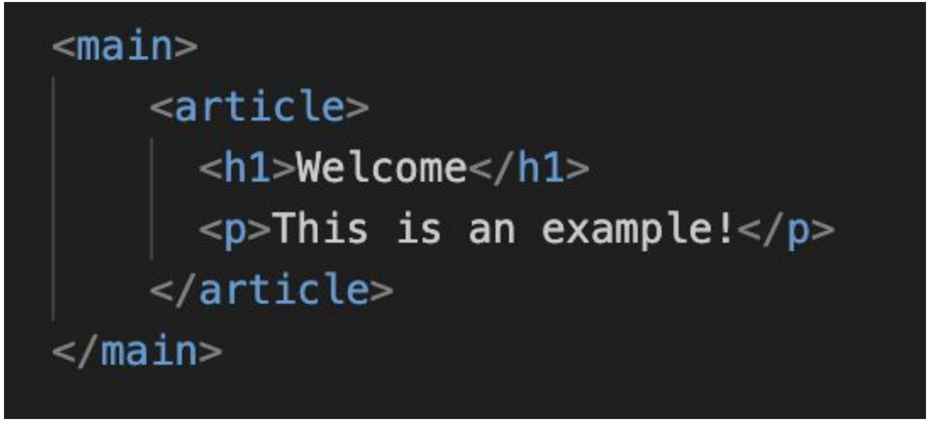
HTML & CSS

6.1040 – Recitation 7

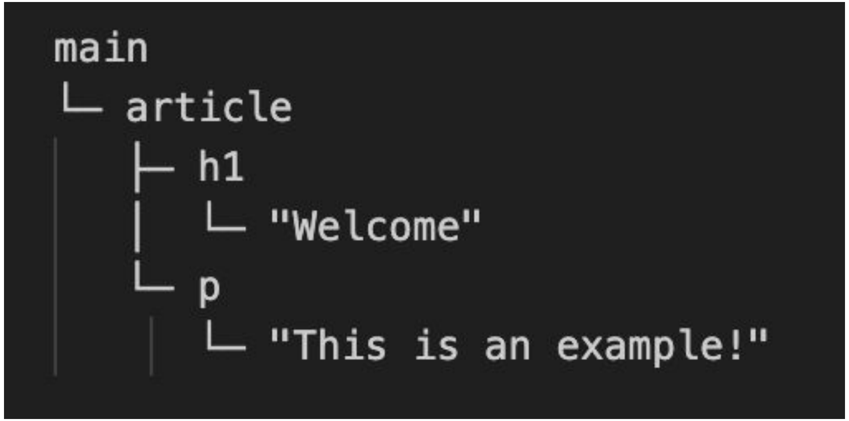
Any issues with the Prep?

HTML

- HTML → HyperText Markup Language
- Tells browser how to display content
- Standard markup language for web pages
- DOM structure (hierarchical) → pages are built from elements *(<header>, <main>, <p>, <a>, etc.)* that are inside other elements



```html
<main>
    <article>
      <h1>Welcome</h1>
      <p>This is an example!</p>
    </article>
</main>
```



```
main
└ article
   ├ h1
   │  └ "Welcome"
   └ p
      └ "This is an example!"
```

Motivation

- Front-end structure that holds web pages together

- Dictates which elements and functionalities will appear on your web page

- In practice, goes together with:
    - CSS (appearance)
    - JavaScript (behavior)

HTML
HTML + CSS
HTML+CSS+JavaScript

Image source: https://medium.com/@rashmipaboda2/basic-concepts-of-html-e9ce7c7e4e82

HTML elements

Syntactically, HTML elements are made up of **tags:**

- Structural elements: `<h1>`, `<p>`, `<section>`, `<article>`, `<header>`, `<footer>`.
- Lists and tables: `<ul>`, `<ol>`, `<table>`, `<tr>`, `<td>`.
- Forms and user input: `<form>`, `<input>`, `<button>`, `<textarea>`.
- Multimedia: `<img>`, `<video>`, `<audio>`.
- Interactive elements: `<details>`, `<summary>`, `<dialog>`.

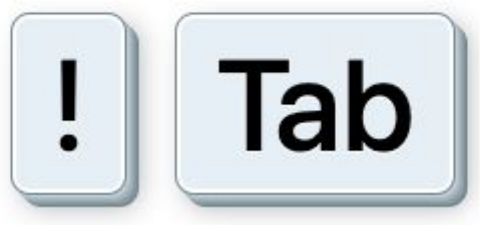
A single HTML element

<h1> Hello world! </h1>

Starting/opening tag **Element content** **Ending/closing tag**

NOTE: Many HTML elements do come with some default styling, for example a heading element will be rendered in bold and a larger font size. However, it is an antipattern to use HTML elements to style content, that’s what CSS is for!

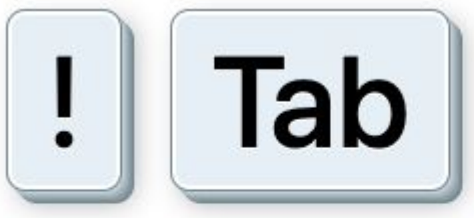
Emmet’s HTML skeleton



Emmet’s HTML skeleton

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>


</body>

</html>
```

Emmet’s HTML skeleton



`<!DOCTYPE html>`	Document type declaration
`<html lang="en">`	
`<head>`	Contains metadata about the page
`    <meta charset="UTF-8">`	
`    <meta name="viewport" content="width=device-width, initial-scale=1.0">`	
`    <title>Document</title>`	Title of the page, which is shown in the browser tab
`<head>`	
`<body>`	Contains the visible content of the page
`</body>`	
`</html>`	

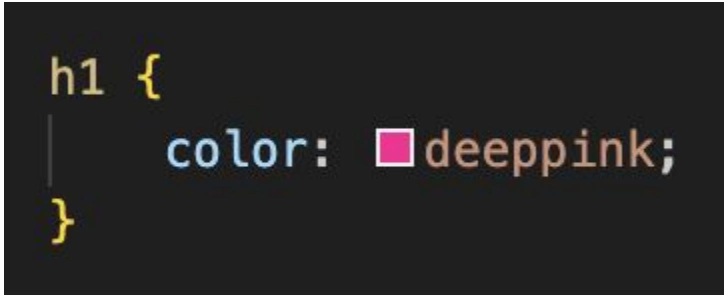
Demo

CSS

- CSS → Cascading Style Sheets
- Used to style web pages by controlling how elements look.
- Can be used to change:
    - Colors
    - Fonts
    - Layouts
    - …

- Made up of rules, containing two parts:
    - **Selector**: tells the browser which elements to style.
    - **Declaration**: specifies what the style should be.

CSS example



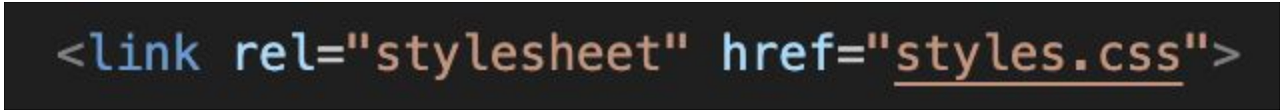
```css
h1 {
    color: deeppink;
}
```

In this example, `h1` is the selector, which means this rule will apply to all `h1` elements on the page. Inside the curly braces `{}` is the declaration: `color: deeppink;`.

This means that all `h1` elements will now appear in deep pink.

Ways to add CSS to a web page I

1. External CSS file
    - Styles are written in a separate *.css* file and linked to the HTML file.
    - Makes the code cleaner and reusable.

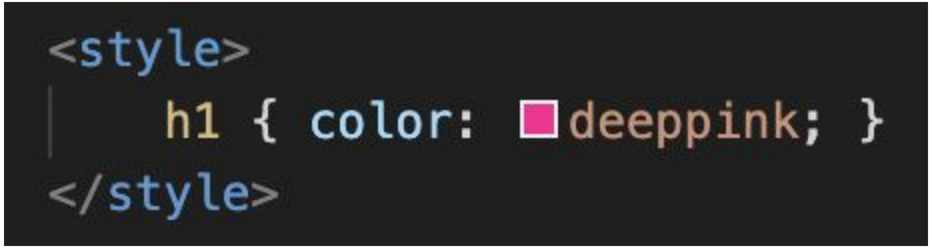


```html
<link rel="stylesheet" href="styles.css">
```

Here, the `link` tag connects the HTML file to `styles.css`, where all our CSS rules are stored.

Ways to add CSS to a web page II

2. Using the <*style*> tag inside the HTML file.
    - Useful for small projects or quick testing.
    - Generally not recommended for larger websites.



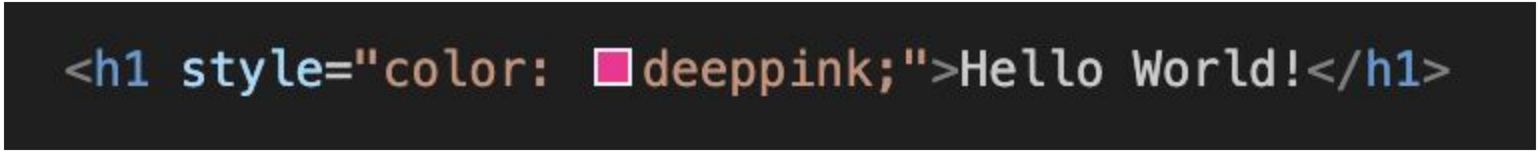
```
<style>
    h1 { color: deeppink; }
</style>
```

Here, the `style` tag within the HTML file will make all `h1` elements in the file appear in deep pink.

Ways to add CSS to a web page III

3. Inline styling

    - CSS is applied directly to an element using the style attribute.
    - It should usually be avoided because it makes the code harder to maintain.
    - (It should be used only for special cases like dynamic styles controlled by JavaScript).



```html
<h1 style="color: deeppink;">Hello World!</h1>
```

CSS Box Model

- Everything in a web page is a box.
  - Every HTML element (paragraph, image, button, …) is represented as a rectangular box.
- With CSS, we can control the size, spacing, and appearance of these boxes using the Box Model.
- The Box Model consists of four main parts:
  - Content – Text or image inside the box.
  - Padding – Space around the content, inside the border.
  - Border – The edge of the box, which can be styled.
  - Margin – Space outside the border, separating this box from others.
- Demo: https://css.land/box-model/

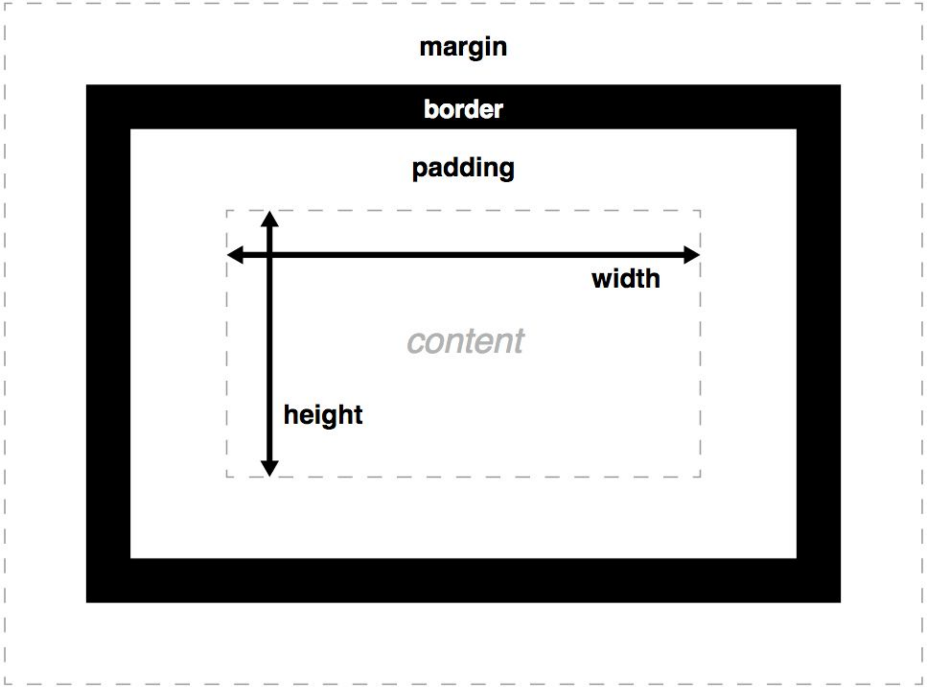
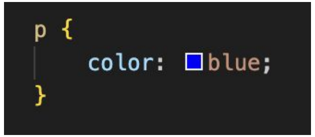


Image source: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_box_model

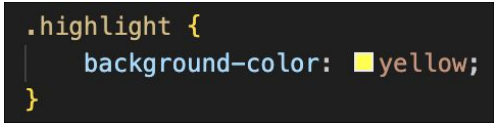
CSS Selectors

- Selectors are used to choose which elements we want to style.
  - Type Selector: targets all elements of a certain type.



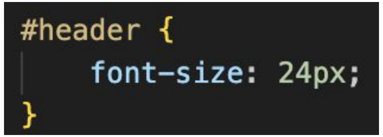
```css
p {
    color: blue;
}
```

  - Class Selector: targets elements with a specific class.



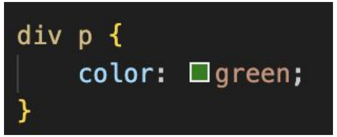
```css
.highlight {
    background-color: yellow;
}
```

  - ID Selector: targets an element with a specific ID.



```css
#header {
    font-size: 24px;
}
```

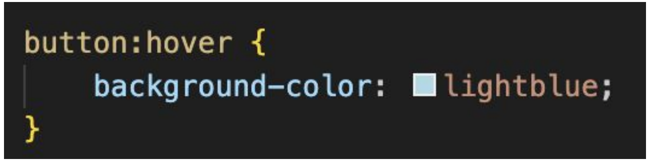
  - Descendant Selector: targets elements inside another element.



```css
div p {
    color: green;
}
```

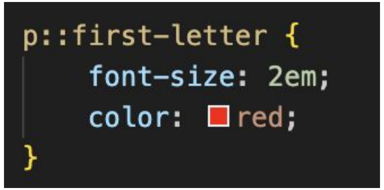
Pseudo-Classes & Pseudo-Elements

- **Pseudo-classes** allow us to style elements based on their state.
  - For example, we can change how a button looks when a user hovers over it.



```css
button:hover {
    background-color: lightblue;
}
```

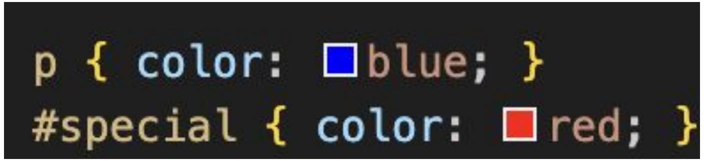
- **Pseudo-elements** allow us to style specific parts of an element
  - For example, we can change the first letter of a paragraph.



```css
p::first-letter {
    font-size: 2em;
    color: red;
}
```

Conflict Resolution

- Sometimes, multiple rules apply to the same element, so CSS needs to determine which rule should take effect. This is called: **Specificity**.
- Priority order:
  - Inline styles.
  - ID selectors.
  - Class selectors.
  - Element selectors.
- If specificity is the same, the last rule in the CSS file wins.



```css
p { color: 🟦blue; }
#special { color: 🟥red; }
```

In this example, if we apply both these rules to the same paragraph, the text will be red because the ID selector has higher specificity.

Exercise

Add a FAQ and Contact Me link to the About Me page of your portfolio!

Requirements:

1. FAQ must begin with a header reading FAQ
2. FAQ questions must be bolded, answers must appear below it italicized.
3. Header and question/answers need to be centered (can we do this in one command?)
4. Wrap contact link in a <footer> element container
5. Contact link should have text: “Questions? E-mail me at <your email>”
    a. You can use a fake address if you do not want to disclose your real email