

6/11.C35, 6/11.C85

Interactive Data Visualization & Society

Programming Labs

Lab check-off

- Meaning:
 - Completed the lab, ready for grading
 - No issues - everything works as expected
 - Fills all the criteria in the (new) lab rubric
- How to do:
 - Synchronous checkoffs:
 - Regular OHs: questions + checkoffs
 - If your lab is incomplete, please add yourself in the 'questions' queue
 - Check-off only, Zoom:
 - Only for fully completed work
 - Asynchronous checkoffs (Google Forms):
 - “One and done” deal. One and only submission. Cannot make changes afterwards.
 - For incorrect or incomplete labs, you will NOT receive credit.
 - Cannot attend synchronous checkoffs.
- No partial grades are given for the lab. Either complete (1) or incomplete (0).

Lab 2: Styling with CSS

02/13/2025

Introduction to CSS Styling

- HTML provides the structure of a webpage, but without CSS, everything would look very plain.
- **CSS** is 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.

Introduction to CSS Styling

```
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 Webpage

1. External CSS file

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

```
<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 Webpage

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 Webpage

3. Inline styling

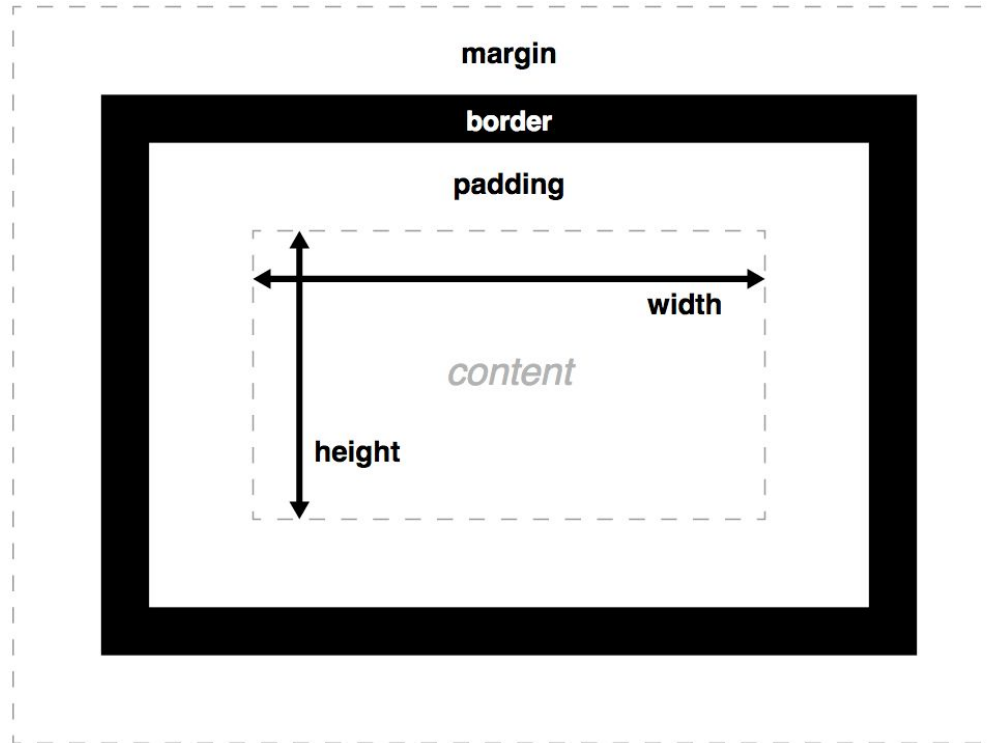
- CSS is applied directly to an element using the style attribute.
- It should usually be avoided because it makes the code harder to maintain.
- (It should be used only for special cases like dynamic styles controlled by JavaScript).

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


CSS Box Model

- Everything in a webpage 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.

CSS Box Model



CSS Box Model

<https://css.land/box-model/>

Block vs. Inline Elements

- Block Elements: take up the full width of the page and start on a new line.
 - `<div>`, `<p>`, `<h1>`, `<section>`
- Inline Elements: take up as much space as necessary and do not start on a new line.
 - ``, `<a>`, ``, ``

<https://css.land/box-model/>

CSS Selectors

- Selectors are used to choose which elements we want to style.

- Type Selector: targets all elements of a certain type.

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

- Class Selector: targets elements with a specific class.

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

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

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

- Descendant Selector: targets elements inside another element.

```
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.

```
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.

```
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.

```
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.

Part 2: hands-on work!