**Instructions**

Close

**Lab instructions**

0 of 1 completed

Styling with Pseudo-Classes and Pseudo-Elements Lab

​

**Introduction:**

Pseudo-classes and pseudo-elements allow developers to apply styles dynamically based on user interaction or structural content. In this lab, you will use the **:hover** pseudo-class to create interactive effects and the **::first-line** pseudo-element to style the first line of paragraphs distinctly.

**Goal:**

The goal of this lab is to create visually dynamic and accessible web pages using **:hover** and **::first-line** for styling paragraphs, improving interactivity and readability.

**Objectives:**

* Understand the purpose and usage of pseudo-classes and pseudo-elements in CSS.
* Apply **:hover** to change the appearance of paragraphs when the mouse hovers over them.
* Use **::first-line** to style the first line of each paragraph to make it stand out.

**Learner Instructions:**

**Part 1: Understand and View the Initial Webpage:**

**Steps:**

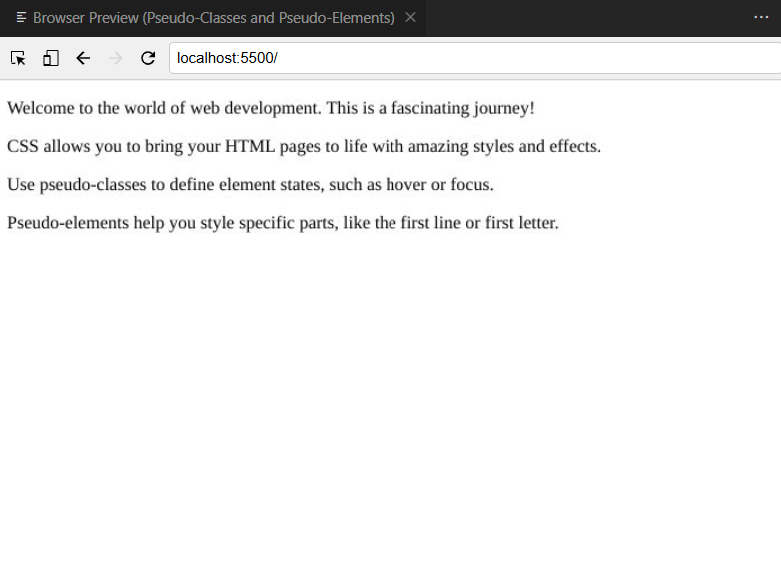
**Step 1:** Open the **index.html** file present under the **PROJECT** folder.

**Step 2:** Review the structure of the **index.html** file:

* It includes multiple div elements, each containing one or more paragraphs (**<p>** tags).

**Step 3:** Start the **live server** and view the webpage layout:

* At the bottom-right of the editor, click on the **Go Live** button.
* Once the server is up and running, you will see an **exposed port** number (**e.g., 5500**). This means your server is now live.
* At the middle-left of the editor, click on the **Browser Preview** button toopen a new Browser Preview tab.
* In the browser, enter the following URL format (replacing **<exposed port>** with the actual port number shown in the editor): http://localhost:<exposed port>.
* Once the URL is entered correctly, the webpage will load, and you should see the **Pseudo-Elements and Pseudo-Classes** webpage based on the default layout defined in the **index.html** file.



**Part 2: Style the page with Pseudo-Elements and Pseudo-Classes:**

**Task 1:** Add a Hover Effect

**Steps:**

**Step 1:** Open the **styles.css** file present under the **PROJECT** folder. This is where you’ll add the CSS pseudo-classes and pseudo-elements to make the web page interactive and visually appealing.

**Step 4:** In the **styles.css** file, write a CSS code using the **:hover** pseudo-class to apply a background color to paragraphs when hovered.

* **Example:** Change the background to light gray when hovering over any paragraph.

1

2

3

p:hover {

    background-color: lightgray;

}

**Task 2:** Style the First Line

**Steps:**

**Step 1:** Write a CSS code using the **::first-line** pseudo-element to change the font size and weight of the first line of each paragraph.

* **Example:** Increase the font size and make the first line bold.

1

2

3

4

p::first-line {

    font-size: 1.2em;

    font-weight: bold;

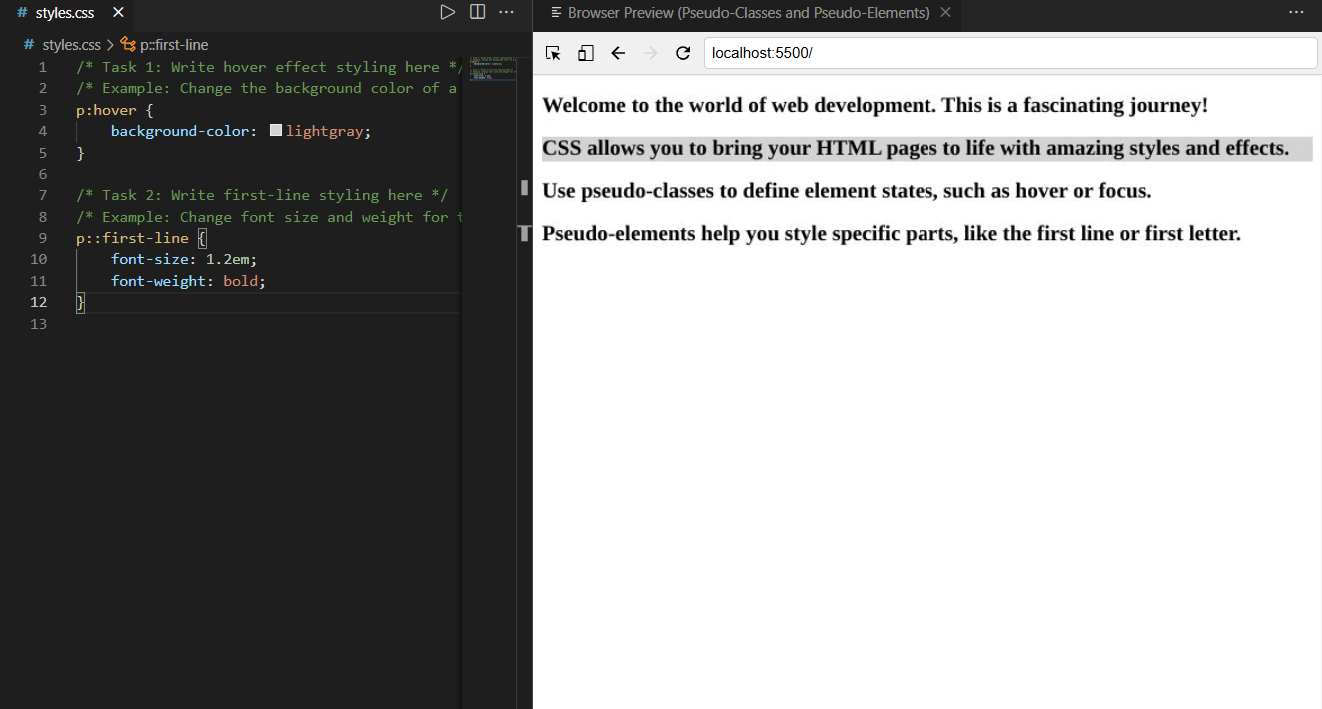
}

**Step 2:** After successfully modifying the **styles.css** file, navigate to **File** > **Save** to save changes in the file.

**Part 3: Verify Style Changes and Close the Live Server:**

**Steps:**

**Step 1:** In the **Browser Preview** panel, click the refresh icon, check, and verify that the webpage displays the updated styles.



**Step 2:** Close the server after completing the lab. Once you’re done with the lab, make sure to **close the server** to free up the port:

* Click on the exposed port number (**e.g., 5500**,**5501**, and so forth) at the bottom-right of VSCode.
* You should see a notification confirming that the server is now offline (stopped).

**Key Takeaways:**

* Pseudo-classes like :hover enable dynamic interaction styles based on user actions.
* Pseudo-elements like ::first-line allow targeted styling of specific portions of text within elements.
* Combining pseudo-classes and pseudo-elements enhances design possibilities without modifying HTML structure.

**Final Step: Mark as Completed**

* Click the **Mark as Completed** button present below to mark the lab as **Completed**.