

## **Day 2**

### **Instructions for Students:**

Watch all the tutorial videos provided on the given websites. These will give you a solid understanding of the CSS concepts needed to complete these exercises.

Complete at least 12 exercises from the below list. Choose at least 3 exercises from each topic area (CSS Basics, CSS Text Styling, CSS Layout, and CSS Animations).

For each task:

Write the CSS code required.

Add HTML markup to demonstrate the application of the CSS.

### **CSS Basics Exercises:**

**Create a webpage using all three types of CSS: inline, internal, and external.**

- Use inline CSS to style a single `<p>` tag.
- Add internal CSS within the `<style>` tag to style the `<h1>` tag.
- Link an external CSS file to style the background color of the body.

**Create a simple webpage and use CSS selectors:**

- Style a class selector to change the background color of paragraphs with a specific class.
- Style an ID selector to change the font size of a header with a specific ID.
- Apply a universal selector to set the font family of the entire page.

**Design a page that demonstrates the CSS box model (margin, border, padding, content):**

- Create a `<div>` with padding, a border, and margin. Set its content size and background color.
- Use different values for margin, padding, and border to see how they affect the layout.

**Style a webpage with different text properties:**

- Change the font size, font family, and color of the text in the `<h1>`, `<p>`, and `<a>` tags using CSS.
- Add text-align properties to center-align a paragraph and left-align a header.

**Design a webpage with background properties:**

- Set a background color for the page.
- Add a background image to a `<div>` element with background-size and background-repeat properties.
- Set the background of a specific `<p>` tag using a gradient.

**Style borders in a webpage:**

- Create a box with a solid border. Use CSS to change the border's color, thickness, and style (e.g., dotted, dashed, solid).
- Experiment with border-radius to round the corners of the box.

**Create a webpage that demonstrates the use of CSS pseudo-classes:**

- Apply :hover to change the color of links when the user hovers over them.
- Use :nth-child to style the even or odd list items in a <ul> list.
- Add :focus to style an input field when it gains focus.

**Use CSS positioning:**

- Position a <div> element relatively with some margin from the top of the page.
- Position an image absolutely within a container <div> to place it at a specific location.
- Use position: fixed to create a fixed navigation bar that remains at the top of the page when scrolling.

**Create a simple layout using Flexbox:**

- Create a page with a header, content section, and footer. Use Flexbox to position the elements.
- Ensure that the content section adjusts its size dynamically based on the viewport size.

**Create a page with both an absolute and relative positioning context:**

- Position one element absolutely within a container <div> that has position: relative.
- Style the absolute element to appear at the top right corner of the container.

**Create a webpage with different font styles:**

- Apply font-weight, font-style, and text-transform properties to different elements.
- Use letter-spacing and line-height to adjust the spacing of text.

**Create a webpage with a fixed header and a scrollable body:**

- Use position: fixed for the header and position: relative for the body section.
- Ensure the body content scrolls independently from the header.

**CSS Text Formatting and Styling Exercises:**

**Create a webpage with a list of items styled using CSS:**

- Style an ordered list with different font sizes, colors, and list markers (e.g., disc, circle).
- Style an unordered list with different padding and margin to ensure proper alignment.

**Create a webpage with multiple boxes that demonstrate the border, padding, and margin properties:**

- Create at least 3 <div> elements with different background colors, padding, borders, and margins to visually demonstrate how they affect layout.

**Create a webpage with styled text and a background:**

- Style the background of a page with a color or image.
- Use text-shadow to add depth to text in a heading.

**Create a form with styled input fields:**

- Style text input and textarea fields with padding, borders, and a background color.
- Apply a hover effect on buttons using the :hover pseudo-class.

**Create a webpage with a container that has an image and text side by side using CSS Flexbox:**

- Use Flexbox to arrange the image and text in a row. Adjust the size of the image and the text to fit properly within the container.

**CSS Layout and Positioning Exercises:**

**Create a responsive webpage layout:**

- Use Flexbox or CSS Grid to create a layout with a header, navigation menu, content area, and footer. Ensure the layout adjusts based on screen size.

**Use media queries to create a mobile-friendly layout:**

- Create a basic layout with a sidebar and main content. Use media queries to adjust the layout for smaller screen sizes (e.g., stacking the sidebar and content)

**Create a grid layout for a photo gallery:**

- Use CSS Grid to arrange multiple images in a responsive grid layout. Adjust the number of columns based on the viewport width.

**CSS Animation Exercises:**

**Create a webpage with a simple animation:**

- Use @keyframes to create an animation that moves a box from left to right.
- Apply animation properties to control the duration, timing, and direction of the animation.

**Create a button that animates on hover:**

- Use CSS :hover to animate a button's size or color when the user hovers over it.