Course material  
Web Development

short line

HTML, CSS & JS

# Assignments:

Install VS Code and Setup for HTML, CSS and JS.

Make a simple webpage using at least 12 different HTML tags.

**June 8th 2024:**

Make a webpage including all of these tags:  
A. Style

B. Anchor

C. Bold

D. Table

E. LIst (ordered and unordered))

June 09th

Book a personal free domain with the domain name “Yourname.com.np”

Create a GitHub Account.

**Tables:**

Design a simple table with 3 rows, 3 columns

Design a table with collapsed border, few rows, and columns.

Design a following table:

| Theory -80 Marks | | | Practical - 20 Marks |
| --- | --- | --- | --- |
| A | B | C | P |
| 20 Marks | 50 Marks | 10 marks | 20 marks |

**Assignment T2: Styling Tables with Internal CSS**

**Objective**: Enhance the appearance of the table using internal CSS.

**Instructions:**

1. Create a table similar to Assignment 1 but with the following columns: "Product Name," "Category," "Price," and "Stock Status."
2. Add at least 4 rows of data.
3. Use internal CSS to style the table with the following:
4. Set a fixed width for the table and centre it on the page.
5. Use a different font for the table text.
6. Apply alternating row colours for the table body.
7. Add padding to table cells for better readability.
8. Style the header row with a bold font and a background colour.

**Five assignments that focus on using `<div>`, ID, and class selectors:**

**Assignment 1: Creating a Simple Layout with Divs**

**Objective**: Use `<div>` elements to create a basic webpage layout.

**Instructions**:

1. Create a simple webpage with the following sections: Header, Navigation, Main Content, Sidebar, and Footer.

2. Use `<div>` elements to define each section.

3. Apply IDs to each `<div>` for styling purposes.

4. Use internal CSS to style each section with different background colors and padding.

**Assignment 2: Styling with Class Selectors**

**Objective**: Apply styles using class selectors.

**Instructions**:

1. Create a webpage with multiple paragraphs and images.

2. Apply at least three different classes to style the paragraphs and images.

3. Use internal CSS to:

- Set different font styles, colours, and margins for each class.

- Add borders and padding to images based on their class.

**Assignment 3: Complex Layout with Nested Divs and Class Selectors**

**Objective:** Create a more complex layout using nested `<div>` elements and class selectors.

**Instructions**:

1. Create a webpage for a blog post with the following structure:

- Header (with the blog title)

- Navigation bar

- Main content area (with an article and a sidebar)

- Footer

2. Use nested `<div>` elements to structure the main content area.

3. Apply classes to style the header, navigation bar, article, sidebar, and footer.

4. Use internal CSS to:

- Set different background colors, font sizes, and margins for each section.

- Add specific styles for the sidebar and article using class selectors.

**Assignment 4: Interactive Webpage with ID Selectors**

**Objective**: Use ID selectors to create a simple interactive webpage.

**Instructions**:

1. Create a webpage with a header, a navigation bar, three content sections, and a footer.

2. Use `<div>` elements to define each section.

3. Apply unique IDs to each content section.

4. Use internal CSS to style each section with different colors and fonts.

5. Add links in the navigation bar that scroll to the corresponding content sections using the IDs.

**Assignment 5: Product Listing with Divs, IDs, and Classes**

**Objective**: Create a product listing page using `<div>` elements, IDs, and classes.

**Instructions**:

1. Create a webpage to display a list of products. Each product should have a name, image, description, and price.

2. Use `<div>` elements to structure each product.

3. Apply classes to style common elements (e.g., product names, descriptions, prices).

4. Use IDs for individual product styling if needed.

5. Use internal CSS to:

- Set styles for the product list layout.

- Style individual elements within each product using class selectors.

- Optionally, add hover effects for product images or prices using class selectors.

These assignments will help your student practice using `<div>`, ID, and class selectors while building practical and interactive web pages.

**Assignment 6: Basic Button Styling**

Objective: Create a webpage with multiple buttons and style them using CSS.

Instructions:

Create a webpage with at least three different buttons.

Each button should have different text, such as "Submit," "Cancel," and "Reset."

Style each button with different background colors, font sizes, and border styles using CSS.

**Assignment 7: Form Styling with External CSS**

Objective: Create and style a form using an external CSS file.

Instructions:

* Create an HTML file named index.html.
* Create an external CSS file named styles.css.
* Link the styles.css file to index.html.
* In the HTML file, create a contact form with fields for name, email, and message, and a submit button.
* In the CSS file, style the form fields to have a border, padding, and margin. Style the submit button with a background colour and hover effects.

**Assignment 8: Navigation Bar Styling**

Objective: Create and style a navigation bar using an external CSS file.

Instructions:

Create an HTML file named index.html.

Create an external CSS file named styles.css.

Link the styles.css file to index.html.

In the HTML file, create a navigation bar with four links: Home, About, Services, and Contact.

In the CSS file, style the navigation bar to have a background color, remove the default list style, and style the links with padding and color changes on hover.

**Combined Assignments**

### **Assignment 1: Interactive Quiz**

**Objective:** Create an interactive quiz with HTML, CSS, and JavaScript.

**Instructions:**

1. **HTML:**
   * Create a webpage with a quiz containing three multiple-choice questions.
   * Each question should have four possible answers, with radio buttons for selecting the answer.
   * Include a "Submit" button at the end of the quiz.
2. **CSS:**
   * Style the quiz with basic CSS to make it visually appealing.
   * Use external CSS to style the div, label, input, and button elements.
   * Center the quiz on the page and add some padding and margin to elements.
3. **JavaScript:**
   * When the "Submit" button is clicked, use JavaScript to check the selected answers.
   * Display the result (number of correct answers) below the quiz.

## **Complete the following assignments to gain proper insight into CSS properties**

### **Assignment 1: Typography and Links**

**Objective:** Practice using different font properties and styling links.

**Instructions:**

1. Create a webpage with a heading and a few paragraphs.
2. Use the font-family property to apply three different fonts to the heading and paragraphs, including at least one web-safe font.
3. Apply different font-style properties (normal, italic, oblique) to some text.
4. Use the font shorthand property to set multiple font-related properties in one declaration.
5. Add a list of links (e.g., home, about, contact) and use CSS to style the links with different colors for their normal, hover, active, and visited states. Remove the underline for the normal state.

### **Assignment 2: Lists, Tables, and Display**

**Objective:** Style HTML lists, tables, and use different display properties.

**Instructions:**

1. Create a webpage with an unordered list and an ordered list.
2. Use CSS to change the list style type (e.g., disc, circle, square for unordered; decimal, roman, alpha for ordered) and style the list items with custom colors and fonts.
3. Create a table with some data (e.g., a schedule or a list of items) and use CSS to style the table borders, header, rows, and cells. Apply different background colors to alternating rows.
4. Add several div elements and use CSS to change their display properties to block, inline, and inline-block. Observe how the elements behave differently with each display property.

### **Assignment 3: Responsive Design and Max-Width**

**Objective:** Use the max-width property and practice responsive design.

**Instructions:**

1. Create a webpage with an image and a paragraph of text.
2. Use CSS to set a max-width for the image so it doesn't exceed a certain width, even if the screen size is large.
3. Set a max-width for the paragraph to ensure it remains readable on larger screens.
4. Use media queries to create a responsive design that adjusts the layout and font sizes for different screen sizes.

### **Assignment 4: Layout with Positioning and Floating**

**Objective:** Practice using positioning and floating properties.

**Instructions:**

1. Create a webpage with a header, a sidebar, and a main content area.
2. Use position: relative, absolute, and fixed to position these elements. Experiment with top, right, bottom, and left properties to place the elements as desired.
3. Create a layout with floated elements (e.g., images or text boxes) and ensure the container properly wraps around them. Use clear property to manage the layout.
4. Add a scrollable container with content that overflows its bounds and use the overflow property to manage it (hidden, scroll, auto).

### **Assignment 5: Advanced Styling and Alignment**

**Objective:** Use advanced CSS properties like z-index, overflow, float, inline-block, and alignment.

**Instructions:**

1. Create a webpage with overlapping elements and use z-index to control their stacking order.
2. Add several div elements styled as inline-block and align them horizontally using text-align on the parent container.
3. Experiment with vertical alignment using vertical-align on the inline-block elements.
4. Create a complex layout that includes both fixed and floating elements, and use position properties to achieve the desired layout.

short dash