**HTML 5 Assignments**

**HTML5 Lists**

**1: Basic Unordered List (`<ul>`)**

**Create a simple unordered list.**

**Task:**

**1. Create an HTML file.**

**2. Inside the `<body>` tag, create an unordered list (`<ul>`).**

**3. Include at least 5 list items (`<li>`) within the `<ul>`.**

**4. Each list item should contain a different country’s name.**

**5. Save the file and open it in a web browser to verify the list displays correctly.**

**2: Ordered List (`<ol>`)**

**Practice creating an ordered list.**

**Task:**

**1. Modify the existing HTML file from Assignment 1.**

**2. Replace the `<ul>` with an ordered list (`<ol>`).**

**3. Add at least 7 list items (`<li>`) within the `<ol>`.**

**4. List the steps to create a simple HTML webpage in sequential order.**

**5. Ensure each list item is numbered automatically by the browser.**

**6. Save the file and check the ordered list in a web browser.**

**3: Nested Lists**

**Explore nesting lists for hierarchical data.**

**Task:**

**1. Extend the HTML file from Assignment 2.**

**2. Within one of the list items in the ordered list (`<ol>`), create a nested unordered list (`<ul>`).**

**3. Include at least 3 sub-items (`<li>`) under the parent list item.**

**4. Each sub-item should represent a feature or component of HTML5.**

**5. Save and view the nested list structure in the browser to ensure proper indentation and hierarchy.**

**4: Styled Lists with CSS**

**Apply CSS to enhance list presentation.**

**Task:**

**1. Enhance the HTML file from Assignment 3.**

**2. Add a `<style>` section within the `<head>` of your HTML file.**

**3. Apply CSS rules to style the `<ul>` and `<ol>` lists differently.**

**- Change the bullet points for `<ul>` to a custom image or style.**

**- Change the numbering style for `<ol>` to use uppercase Roman numerals.**

**4. Adjust padding, margins, or font styles to improve readability and aesthetics.**

**5. Save and preview the styled lists in a browser to see the CSS effects.**

**5: Practical Application**

**Use lists creatively in a real-world scenario.**

**Task:**

**1. Create a new HTML file.**

**2. Design a webpage that includes both `<ul>` and `<ol>` lists.**

**3. Use lists to showcase a comparison of features between two products, services, or technologies.**

**4. Include at least 2 `<ul>` lists and 1 `<ol>` list on the page.**

**5. Provide descriptive `<li>` items with detailed content under each list.**

**6. Use appropriate CSS to style the lists for clarity and visual appeal.**

**7. Validate the HTML file for correctness and check the presentation in different web browsers.**

**HTML5 Image**

**1: Basic Image Embedding**

**Embedding images using the `<img>` tag.**

**Task:**

**1. Create an HTML file.**

**2. Download at least three different images (e.g., JPEG or PNG format) to your computer.**

**3. Embed each image using the `<img>` tag within the `<body>` section of the HTML file.**

**4. Use the `src` attribute to specify the path to each image file.**

**5. Provide alternative text using the `alt` attribute for each image.**

**6. Save the file and open it in a web browser to ensure images are displayed correctly.**

**2: Image Attributes and Styling**

**Explore attributes and CSS styling for images.**

**Task:**

**1. Extend the HTML file from Assignment 1.**

**2. Add attributes to the `<img>` tags to control image size (`width` and `height`), alignment (`align`), and borders (`border`).**

**3. Experiment with different values for these attributes to understand their impact on image display.**

**4. Create a `<style>` section within the `<head>` of your HTML file.**

**5. Apply CSS rules to style the images further, such as adding margins, borders, or changing the background color around images.**

**6. Save and view the styled images in a browser to observe the effects of your CSS styling.**

**3: Responsive Images**

**Implement responsive images using HTML5 attributes and CSS.**

**Task:**

**1. Create a new HTML file.**

**2. Include a large image using the `<img>` tag.**

**3. Implement responsive image techniques by using the `srcset` and `sizes` attributes.**

**- Provide multiple image sources (`srcset`) for different screen resolutions.**

**- Use the `sizes` attribute to specify different image sizes based on viewport width (`(max-width: 600px) 400px, 800px`).**

**4. Add CSS to ensure images scale appropriately on different devices (`max-width: 100%; height: auto;`).**

**5. Test the responsiveness of the images by resizing the browser window and viewing the webpage on different devices.**

**4: Image Gallery**

**Create an image gallery using HTML5 and CSS.**

**Task:**

**1. Modify the HTML file from Assignment 3.**

**2. Create a grid layout using `<div>` containers to display multiple images.**

**3. Embed at least 6 images within the grid using `<img>` tags.**

**4. Apply CSS to style the grid layout, such as defining column widths, spacing between images, and borders around each image container.**

**5. Implement hover effects using CSS (e.g., changing opacity or adding a border) to enhance user interaction with images.**

**6. Ensure each image has appropriate alternative text (`alt` attribute) for accessibility purposes.**

**7. Validate the HTML and CSS code for correctness and consistency.**

**HTML5 Table**

**1: Basic Table Structure**

**Create a simple table structure using HTML5.**

**Task:**

**1. Create an HTML file.**

**2. Inside the `<body>` tag, create a `<table>` element.**

**3. Define a table header (`<thead>`) with at least 3 columns using `<th>` elements.**

**4. Populate the table body (`<tbody>`) with 5 rows (`<tr>`) and fill each row with data using `<td>` elements.**

**5. Include relevant data such as student names, ages, and grades.**

**6. Save the file and open it in a web browser to ensure the table displays correctly.**

**2: Styling Tables with CSS**

**Apply CSS to enhance the appearance of tables.**

**Task:**

**1. Extend the HTML file from Assignment 1.**

**2. Create a `<style>` section within the `<head>` of your HTML file.**

**3. Use CSS to style the table (`<table>`), table headers (`<th>`), and table data (`<td>`).**

**- Set table width, cell padding, and cell borders.**

**- Apply background colors, font styles, and alignment to headers and data cells.**

**4. Add alternating row colors (`nth-child` selector) to improve readability.**

**5. Save and view the styled table in a browser to observe the effects of your CSS styling.**

**3: Complex Table Structure**

**Explore nested tables and table captions.**

**Task:**

**1. Create a new HTML file.**

**2. Create a `<table>` element with a caption (`<caption>`) describing the content of the table.**

**3. Use `<thead>`, `<tbody>`, and `<tfoot>` sections to organize data logically.**

**4. Include at least one nested table (`<table>` within a `<td>`) to display additional details for a specific row.**

**5. Populate the tables with relevant data, such as product details, sales figures, or sports statistics.**

**6. Apply appropriate CSS to style nested tables, captions, and data for clarity and aesthetics.**

**7. Validate the HTML and CSS code to ensure proper structure and formatting.**

**HTML5 Select Assignments**

**1: Basic Dropdown Menu**

**Create a simple dropdown menu with a list of fruits.**

**2: Pre-selected Option**

**Create a dropdown menu where "Banana" is pre-selected by default.**

**Hint: - Selected attribute**

**3: Grouped Options**

**Create a dropdown menu with grouped options. Group the options into "Citrus Fruits" and "Berries".**

**Hint: - optgroup**

**4: Disabled Options**

**Create a dropdown menu with some options disabled.**

**Hint: - disabled**

**5: Multiple Selection**

**Create a dropdown menu that allows multiple selections.**

**Hint: - multiple**

**HTML5 data list Assignments**

**Do all select assignments with data list and use data as cities of India instead of fruits. Group them based on states.**

**HTML5 Custom attributes / data attributes**

**1: Create a Table with Custom Data Attributes**

**Create a table where each row has custom data attributes for `id`, `name`, and `age`.**

**2: Create a Contact List with Custom Data Attributes**

**Create a contact list where each contact has custom data attributes for `id`, `phone`, and `email`.**

**HTML5 Input Types Assignments**

**1: Create a Registration Form**

**Create a registration form using different HTML5 input types. The form should include:**

**1. Full Name : `text`**

**2. Email : `email`**

**3. Password : `password`**

**4. Phone Number : `tel`**

**5. Date of Birth : `date`**

**6. Gender : `radio`**

**7. Hobbies : `checkbox`**

**8. Profile Picture : `file`**

**9. Website : `url`**

**10. Age : `number`**

**11. Submit Button : `submit`**

**2: Create a Feedback Form**

**Create a feedback form using different HTML5 input types. The form should include:**

**1. Name : `text`**

**2. Email : `email`**

**3. Feedback : `textarea`**

**4. Rating : `range`**

**5. Favorite Color : `color`**

**6. Feedback Date : `datetime-local`**

**7. Submit Button : `submit`**

**3: Create a Survey Form**

**Create a survey form using different HTML5 input types. The form should include:**

**1. Name : `text`**

**2. Email : `email`**

**3. Age : `number`**

**4. Date of Visit : `date`**

**5. Time of Visit : `time`**

**6. Satisfaction Level : `range`**

**7. Comments : `textarea`**

**8. Submit Button : `submit`**

**Form Validations**

**1: Create a User Registration Form with Validation**

**Create a registration form that includes the following fields with appropriate HTML5 validation:**

**1. Full Name : Required, text**

**2. Email : Required, email**

**3. Password : Required, password, at least 8 characters**

**4. Confirm Password : Required, password, must match the Password field**

**5. Phone Number : Required, pattern for phone number**

**6. Date of Birth : Required, date**

**7. Website : Optional, URL**

**8. Age : Required, number, min 18, max 100**

**2: Create a Feedback Form with Validation**

**Create a feedback form that includes the following fields with appropriate HTML5 validation:**

**1. Name : Required, text**

**2. Email : Required, email**

**3. Feedback : Required, textarea, minimum 10 characters**

**4. Rating : Required, range, min 1, max 5**

**5. Favorite Color : Required, color**

**6. Submit Button : Submit**

**3: Create a Survey Form with Validation**

**Create a survey form that includes the following fields with appropriate HTML5 validation:**

**1. Name : Required, text**

**2. Email : Required, email**

**3. Age : Required, number, min 18, max 100**

**4. Date of Visit : Required, date**

**5. Time of Visit : Required, time**

**6. Satisfaction Level : Required, range, min 1, max 10**

**7. Comments : Optional, textarea**

**8. Submit Button : Submit**

**HTML5 Form Attributes Assignments**

**1: Create a Login Form with `autocomplete` and `novalidate`**

**Create a login form that includes the following attributes:**

**1. autocomplete : Set to "on" or "off" for username and password fields.**

**2. novalidate : Use this attribute to disable HTML5 validation.**

**Form -novalidate**

**Input type for username and password – autocomplete**

**2: Create a Registration Form with `required` and `pattern`**

**Create a registration form that includes the following attributes:**

**1. required : Make fields mandatory.**

**2. pattern : Use a regular expression to validate input.**

**Input type – name, email, password – required**

**Input type – phone number – required and pattern**

**3: Create a Feedback Form with `maxlength`, `minlength`, `readonly`, and `disabled`**

**Create a feedback form that includes the following attributes:**

**1. maxlength : Set a maximum length for the textarea.**

**2. minlength : Set a minimum length for the textarea.**

**3. readonly : Make a field read-only.**

**4. disabled : Disable a field.**

**Input type – id, email, feedback, submit time – required.**

**Feedback – minlength, maxlength**

**Email – readonly (initially provide some email)**

**Submit time – text – hard coded to 12:00 pm – disabled.**

**Do validations with javascript checkValidity function and show appropriate error messages.**