HTML4, CSS3

Unit - 1 : Content

HTML All Tags with attributes and sub-elements of List, Tables, Images, Form, div

Cascading Style Sheets (CSS) 3

- Types of CSS: inline, internal, and external
- Selectors:
  1. Basic
  2. Combinatory
  3. Attribute
  4. Pseudo-class
  5. Pseudo-element
- Properties: font, background

Introduction to HTML

HTML stands for Hypertext Markup Language, and it is the most widely used language to create Web Pages or electronic data or hypertext

Sample Structure of a web page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
   initial-scale=1.0">
    <title>My Web Page  </title>
</head>
<body>
   This is our web page
</body>
</html>
```

Html Element/Tag

```
    <tagname>Content</tagname>
```

Introduction to HTML

More about the element/tag:

```
<tagname attribute="value">Content</tagname>
```

Here attribute defines the additional properties of tag like style,href,alt,src,class et.,

Every tag in html comes in any one of these types of tags:

- **Paired Tags:**

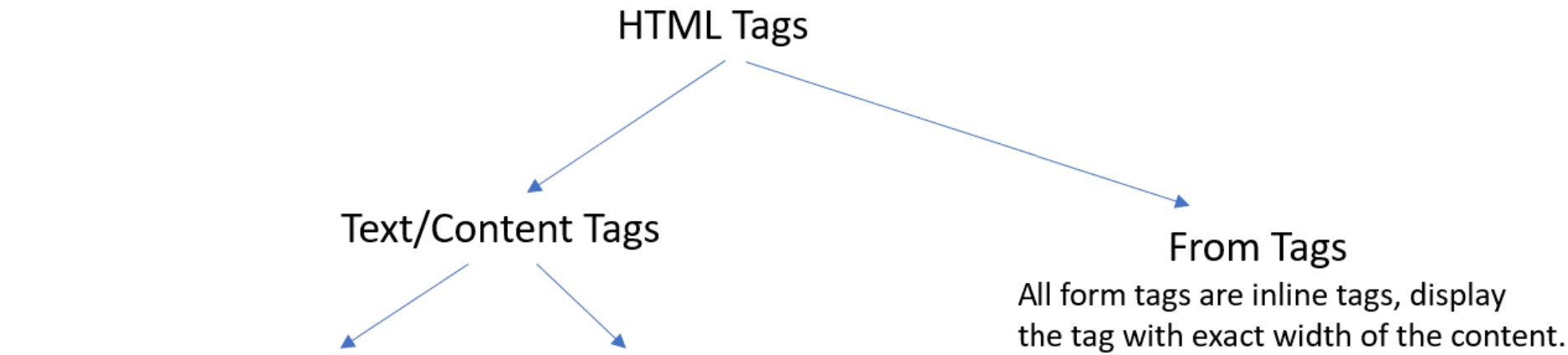
```
<tagname attribute="value">Content</tagname>
Ex:
<h1>Content</h1>
```

- **Unpaired Tags:**

```
<tagname attribute="value"/>
Ex:
<img src=""/>
```

HTML Tags

HTML tags



HTML Tags

- **Heading Tags:**

```
<h1>Most Important Heading</h1>
to
<h6>Least Important Heading</h1>
```

- **div Tag:**

```
<div>This is my page division</div>
```

- **p Tag:**

```
<p>This is my paragrapgh</p>
```

- **img Tag:**

```
<img src='flowers.png' />
```

- **Formatting Tags:**

```
<b>This is bold text</b>
<i>This is italic text</i>
<u>This is underline text</u>
```

Introduction to HTML4 Lists

- HTML lists are used to present information in a structured format.
- Lists are used to organize and display data in a clear and meaningful way.
- HTML4 provides three types of lists: **ordered lists**, **unordered lists**, and **definition lists**.

Attributes of Lists

- **type**: Specifies the type of marker used in ordered lists. Values can be `1`, `A`, `a`, `I`, `i`, or `disc`, `circle`, `square` for unordered lists.
- **start**: Specifies the starting value for the first item in an ordered list.
- **compact**: Deprecated attribute used to reduce the spacing between list items.
- **reversed**: Specifies that the numbering of an ordered list should be reversed.

Ordered Lists (<ol>)

- Ordered lists are used to present items in a sequence.
- Each item in an ordered list is prefixed with a number or another sequential marker.
- Example:

```
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

1. First item
2. Second item
3. Third item

Unordered Lists (<ul>)

- Unordered lists are used to present items in no particular order.
- Each item in an unordered list is prefixed with a bullet point or another marker.
- Example:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

- Item 1
- Item 2
- Item 3

Definition Lists (<dl>)

- Definition lists are used to present terms and their definitions.
- Each term in a definition list is defined using the <dt> tag, and its definition is provided using the <dd> tag.
- Example:

```
<dl>
  <dt>Term 1</dt>
  <dd>Definition 1</dd>
  <dt>Term 2</dt>
  <dd>Definition 2</dd>
</dl>
```

Term 1
    Definition 1
Term 2
    Definition 2

Introduction to HTML4 Table

- HTML tables are used to display data in tabular form.
- Tables consist of rows <tr> and columns made up of table data <td> and table headers <th>.
- They provide a way to organize content in a structured manner.

Basic Table Structure

```
<table>
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>Data 1</td>
    <td>Data 2</td>
  </tr>
</table>
```

HTML4 Table Tags

Tag	Description
<table>	Defines a table in HTML.
<caption>	Defines a caption (title) for the table.
<tr>	Defines a row in the table.
<th>	Defines a header cell in the table.
<td>	Defines a standard data cell in the table.
<thead>	Group header content of the table.
<tbody>,	body content of the table.
<tfoot>	footer content of the table.
<colgroup>	Define groups of columns
<col>	Define individual columns in a table

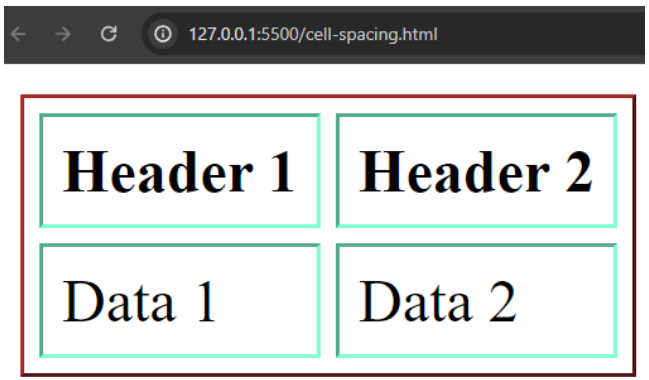
Attributes of the Table Tag

- **border**: Specifies the width of the border around the table. Default value is 0.
- **cellpadding**: Specifies the space between the cell content and cell borders. Default value is 1.
- **cellspacing**: Specifies the space between cells. Default value is 1.
- **frame**: Specifies which parts of the outside borders that should be visible. Values can be "void", "above", "below", "hsides", "vsides", "lhs", "rhs", "box", or "border".
- **rules**: Specifies which parts of the inside borders that should be visible. Values can be "none", "groups", "rows", "cols", or "all".
- **summary**: Provides a summary of the table's content for screen readers.
- **width**: Specifies the width of the table. It can be specified in pixels or as a percentage of the available space.

Table Example with Attributes

```
<table border="1" cellspacing="0" cellpadding="5">
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>Data 1</td>
    <td>Data 2</td>
  </tr>
</table>
```

Output:



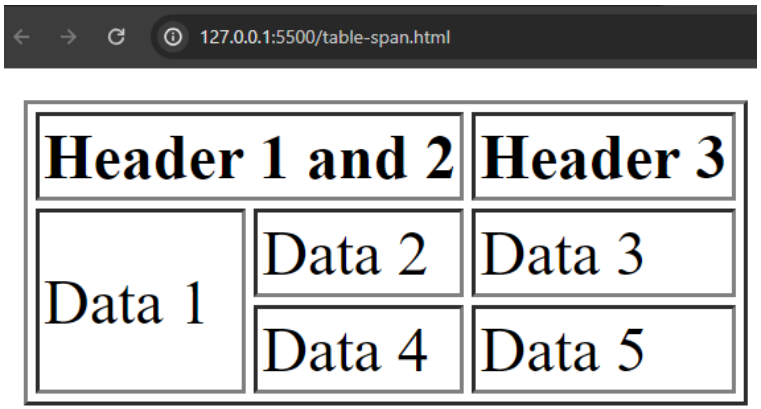
Spanning Rows and Columns I

- `colspan`: Specifies the number of columns a cell should span.
- `rowspan`: Specifies the number of rows a cell should span.

```
<table border="1">
  <tr>
    <th colspan="2">Header 1 and 2</th>
    <th>Header 3</th>
  </tr>
  <tr>
    <td rowspan="2">Data 1</td>
    <td>Data 2</td>
    <td>Data 3</td>
  </tr>
  <tr>
    <td>Data 4</td>
    <td>Data 5</td>
  </tr>
</table>
```

Spanning Rows and Columns II

Output:



Introduction to HTML4 Form

```
<form action="url_to_submit_data" method="post">
 <label for="input_id">Label:</label>
 <input type="text" id="input_id" name="input_name" >
 <input type="submit" value="Submit">
</form>
```

- The <form> tag defines the start of the form.
- Various input fields such as text input, checkboxes, radio buttons, etc., can be placed inside the form.

- HTML form attributes provide additional information and functionality to HTML forms.
- They control various aspects of form behavior and appearance.
- Understanding form attributes is crucial for building user-friendly and accessible web forms.

Common Form Attributes

- **action:** Specifies where to send the form data when submitted.
- **method:** Specifies the HTTP method used to submit the form data (e.g., GET or POST).
- **name:** Assigns a name to the form, which can be used for referencing it in JavaScript or CSS.
- **id:** Assigns a unique identifier to the form for scripting or styling purposes.
- **target:** Specifies where to display the response received after submitting the form (e.g., _self, _blank, _parent, _top, frame_name).
- **autocomplete:** Specifies whether the browser should autocomplete form values.
- **enctype:** Specifies how form data should be encoded before sending it to the server (e.g., multipart/form-data for file uploads).
- **novalidate:** Specifies that the form should not be validated when submitted.

Form Attribute: method

- The ‘method‘ attribute specifies the HTTP method used to submit form data.
- It can have two values: `GET` or `POST`.
- **GET**: Appends form data to the URL in the form of query parameters. Suitable for submitting small amounts of data and retrieving data.
- **POST**: Sends form data in the body of the HTTP request. Suitable for submitting sensitive or large amounts of data.

Form Attribute: target

- The ‘target‘ attribute specifies where to display the response received after submitting the form.
- It can take the following values:
  - **_self**: The response replaces the current page.
  - **_blank**: The response opens in a new tab or window.
  - **_parent**: The response replaces the parent frame.
  - **_top**: The response replaces the current window, including all frames.
  - **frame_name**: The response replaces a specific frame.

List of HTML4 Form Elements

1. <input>: Defines an input control.
   - Types: text, password, checkbox, radio, submit, reset, file, hidden, image, button.
2. <select>: Defines a drop-down list.
   - Options are defined using <option> tags.
3. <textarea>: Defines a multi-line text input control.
4. <button>: Defines a clickable button.
   - Attributes: type, name, value.
5. <label>: Defines a label for an <input>, <select>, <textarea>, or <button> element.
6. <fieldset>: Groups related form elements together.
7. <legend>: Defines a caption for a <fieldset> element.
8. <optgroup>: Groups related options in a <select> dropdown list.
9. <option>: Defines an option in a <select> dropdown list.

HTML Form Input Tags

1. **Text Input**
   - **Description:** Allows users to enter single-line text.
   - **Example:**
     ```
     <input type="text" name="username" id="username">
     ```
2. **Password Input**
   - **Description:** Allows users to enter passwords securely.
   - **Example:**
     ```
     <input type="password" name="password" id="password">
     ```
3. **Checkbox**
   - **Description:** Allows users to select one or more options from a list.
   - **Example:**
     ```
     <input type="checkbox" name="interest" value="music">
     Music
     ```
4. **Radio Button**
   - **Description:** Allows users to select one option from a list.
   - **Example:**
     ```
     <input type="radio" name="gender" value="male"> Male
     ```

HTML Form Input Tags: continued

5. **Submit Button**
   - **Description:** Submits the form data to the server.
   - **Example:**

```
<input type="submit" value="Submit">
```

6. **Reset Button**
   - **Description:** Resets the form's controls to their initial values.
   - **Example:**

```
<input type="reset" value="Reset">
```

7. **File Input**
   - **Description:** Allows users to select files for upload.
   - **Example:**

```
<input type="file" name="file" id="file">
```

HTML Form Input Tags: continued

8. **Hidden Input**
   - **Description:** Provides a way to store data without displaying it in the form.
   - **Example:**
     ```
     <input type="hidden" name="user_id" value="123">
     ```
9. **Image Input**
   - **Description:** Displays an image that represents a submit button.
   - **Example:**
     ```
     <input type="image" src="submit.png" alt="Submit">
     ```
10. **Button**
    - **Description:** Defines a clickable button.
    - **Example:**
      ```
      <button type="button">Click Me</button>
      ```

HTML Form Input Types : Example Program I

```
<!DOCTYPE html>
<html lang="en">
<head> <title>HTML Form Inputs</title> </head>
<body>
    <form action="/submit" method="post">
        <label for="fname">First Name:</label><br>
        <input type="text" id="fname"><br><br>

        <label for="lname">Last Name:</label><br>
        <input type="text" id="lname"><br><br>

        <label for="email">Email:</label><br>
        <input type="email" id="email"><br><br>

        <label for="password">Password:</label><br>
        <input type="password" id="password"><br><br>

        <label for="dob">Date of Birth:</label><br>
```

HTML Form Input Types : Example Program II

```
        <input type="date" id="dob" name="dob"><br><br>

        <label for="gender">Gender:</label><br>
        <input type="radio" id="male" name="gender" value="
    male">
        <label for="male">Male</label>
        <input type="radio" id="female" name="gender" value=
    "female">
        <label for="female">Female</label><br><br>
        <label for="terms">Agree to Terms:</label>
        <input type="checkbox" id="terms" name="terms"
    checked><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

HTML Form Input Types: Output

Button Tag in HTML

- The <button> tag in HTML is used to define a clickable button.
- It can be used to submit form data, trigger JavaScript functions, or perform any other action defined by the web developer.

Attributes:

- type: Specifies the type of button. Common values are "submit", "reset", and "button". The default type is "submit".
- name: Specifies the name of the button, which can be used to reference the button in JavaScript or when submitting a form.
- value: Specifies the value associated with the button, which is submitted along with the form data when the button is clicked.
- Other common attributes include id, class, disabled, autofocus, etc.

Example:

```
<button type="submit" name="submitBtn"
    value="Submit">Submit</button>
```

Differences Between Input Submit and Button Submit

Context	Input Submit	Button Submit
Default Behavior	Submits the form when clicked.	similar to Input Submit
Content	Does not have any content. Its value attribute defines the text displayed on the button.	Can contain text or other HTML elements as its content.
Accessibility	Some screen readers may not interpret the value attribute correctly.	More accessible to screen readers, as it provides better context for users with disabilities.
Styling	More limited in terms of styling.	More flexible, as it can contain additional HTML elements and CSS.
Compatibility	It is supported by all browsers and versions of HTML.	Older versions of Internet Explorer have issues with CSS.

Select Tag in HTML I

- The `<select>` tag in HTML is used to create a drop-down list of options for the user to choose from.
- It allows users to select one or more options from the list.

Nested Tags:

- `<option>`: Each `<option>` tag represents an option in the drop-down list. The text content of the `<option>` tag is displayed to the user, and the `value` attribute specifies the value associated with the option.
- The `<optgroup>` tag in HTML is used to group related options within a dropdown list created using the `<select>` tag. It allows for better organization and categorization of options, enhancing the user experience.

Select Tag in HTML II

Attributes:

- `name`: Specifies the name of the select element, which is submitted along with the form data.
- `id`: Specifies a unique identifier for the select element, which can be used for scripting or styling purposes.
- `multiple`: If present, allows the user to select multiple options from the list.
- `size`: Specifies the number of visible options in a dropdown list. Example: `<select name="countries" size="5">`
- `disabled`: Disables the dropdown list, making it unselectable. Example: `<select name="colors" disabled>`
- `autofocus`:Automatically focuses on the dropdown list when the page loads. Example: `<select name="fruit" autofocus>`

Explanation of Textarea and Fieldset Tags

- **Textarea Tag**: Creates a multiline text input field.

```
<textarea rows="4" cols="50">
Enter your message here...
</textarea>
```

- **Fieldset Tag**: Groups related form elements together.

```
<fieldset>
  <legend>Contact Information</legend>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name"><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email"><br>
</fieldset>
```

HTML form Tags : Example Program I

```
<!DOCTYPE html>
<html>
<head>
    <title>Complete HTML Program : select, option, optgroup,
     textarea, fieldset, button tags</title>
</head>
<body>
    <h2>Select your favorite fruit:</h2>
    <select id="fruit" name="fruit">
        <optgroup label="Summer Fruits">
            <option value="watermelon">Watermelon</option>
            <option value="strawberry">Strawberry</option>
        </optgroup>
        <optgroup label="Winter Fruits">
            <option value="apple">Apple</option>
            <option value="orange">Orange</option>
        </optgroup>
    </select><br>
```

HTML form Tags : Example Program II

```
    <textarea rows="4" cols="50" placeholder="Enter your
    feedback"></textarea>
    <fieldset>
        <legend>Contact Information</legend>
        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email"><br>
    </fieldset><br>

    <button type="submit">Submit</button>
    <button type="reset">Reset</button>
    <button type="button" onclick="alert('Custom action')">
    Custom Button</button>
</body>
</html>
```

HTML Form tags: Output

Select your favorite fruit:

Watermelon

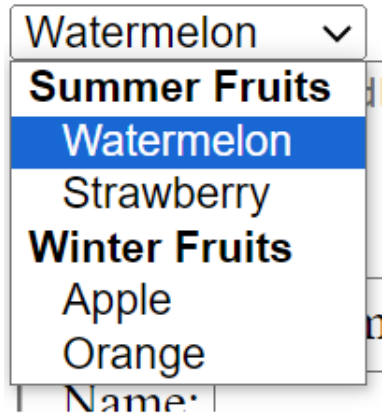
Enter your feedback

Contact Information

Name:

Email:

Submit Reset Custom Button



HTML <img> Tag

The HTML <img> tag is used to embed images into a webpage. The <img> tag supports several attributes:

- **src**: Specifies the URL of the image.
- **alt**: Specifies an alternate text for the image, used when the image cannot be displayed.
- **width**: Specifies the width of the image.
- **height**: Specifies the height of the image.
- **title**: Specifies a title for the image, often displayed as a tooltip.
- **align**: Specifies the alignment of the image within the surrounding text (deprecated in HTML5).

```
<img src="image.jpg" alt="Description of the image" width="
    300" height="200">
```

HTML Tag I

The tag is a block-level element used to group other HTML elements and apply styling to them as a group.

- It creates a division or section in an HTML document.
- Commonly used to structure a web page into sections like headers, navigation bars, content areas, and footers.
- Can be styled using CSS (Cascading Style Sheets) to change its appearance and layout.
- It does not provide any visual formatting itself, but it can be used with attributes for styling and identification purposes.

The tag can be customized using various attributes:

- **id**: Specifies a unique identifier for the element, used for styling or JavaScript manipulation.

HTML Tag II

- **class**: Assigns one or more class names to the element, allowing multiple elements to be styled together.
- **style**: Defines inline CSS styles for the element, such as color, font size, padding, etc.

Example:

```
<div id="header" class="container" style="background-color:
    #333; color: #fff;">
    <h1>Welcome to my Website</h1>
    <p>This is the header section.</p>
</div>
```

HTML Tag I

The tag is an inline-level element used to apply styles to a specific part of the text within a block-level element.

- It does not create a new line or a new block.
- Typically used to apply styles or manipulate small pieces of text within a larger block of content.
- Similar to the tag, but for smaller, inline elements.
- Useful when you want to style or manipulate a specific portion of text within a paragraph, heading, or other block-level element.

The tag supports similar attributes as the tag:

- **id**
- **class**
- **style**

HTML Tag II

These attributes allow for customization and styling of the element. Example:

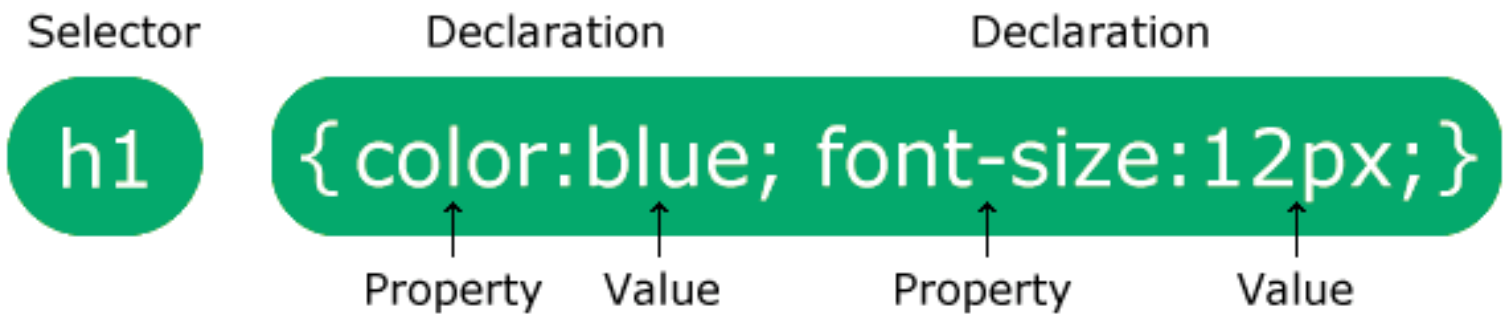
```
<p>
    Welcome to my <span style="font-weight: bold;">website</
    span>.
    Feel free to <span style="color: blue;">explore</span>
    the content.
</p>
```

CSS Introduction

CSS (Cascading Style Sheets) is used to style the appearance of HTML elements on a webpage. There are three main types of CSS:

- Inline CSS
- Internal CSS
- External CSS

CSS Structure:



Inline CSS

Inline CSS is applied directly to an HTML element using the ‘style‘ attribute.

- Example:

```
<p style="color: blue; font-size: 16px;">Hello</p>
```

- Inline CSS affects only the specific element it is applied to and overrides any external or internal CSS.
- It is useful for applying unique styles to individual elements.

Internal CSS

Internal CSS is defined within the <style> element in the <head> section of an HTML document.

- Example:

```
<style>
    p {
        color: red;
        font-size: 18px;
    }
</style>
```

- Internal CSS affects all elements with the specified selectors within the same HTML document.
- It is useful for applying styles consistently across multiple elements within a single document.

External CSS

External CSS is defined in a separate CSS file and linked to an HTML document using the `<link>` element.

- Example:

```
<link rel="stylesheet" href="styles.css">
```

- External CSS affects all elements with the specified selectors across multiple HTML documents.
- It allows for better organization and maintenance of styles, especially for large websites with multiple pages.

HTML Example with CSS I

Below is an HTML example that demonstrates the use of different CSS selectors and properties with inline, internal, and external CSS:

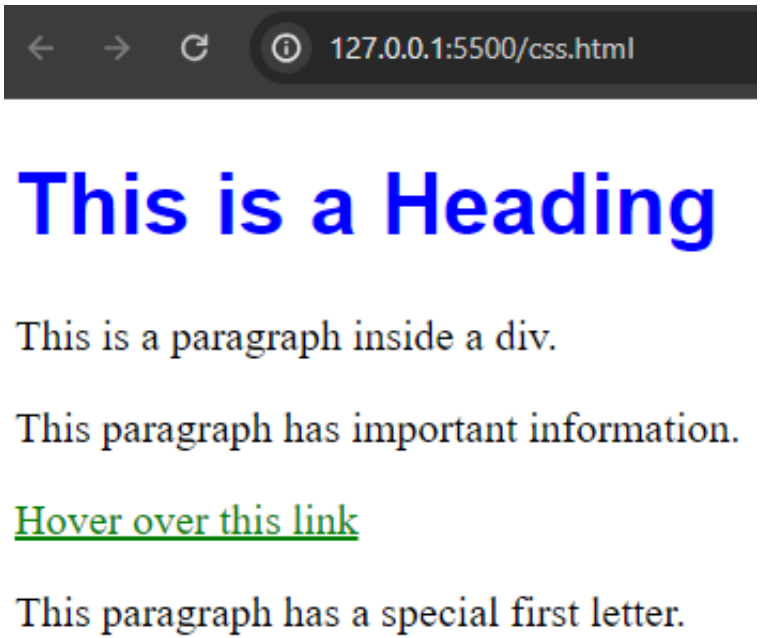
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>CSS Example</title>
    <!-- External CSS -->
    <link rel="stylesheet" href="styles.css">
    <style>
        /* Internal CSS */
        h1 {
            color: blue;
        }
    </style>
</head>
```

HTML Example with CSS II

```
<body>
    <!-- Inline CSS -->
    <h1 style="font-family: Arial;">This is a Heading</h1>
    <div>
        <p>This is a paragraph inside a div.</p>
    </div>
    <div>
        <p data-info="important">This paragraph has
    important information.</p>
    </div>
    <a href="#" style="color: green;">Hover over this link</
    a>
    <p>This paragraph has a special first letter.</p>
</body>
</html>
```

HTML Example with CSS III

Output:



CSS Selectors

- CSS selectors are used to target HTML elements.
- CSS properties are used to define the visual style of targeted elements.

Different types of CSS 3 selectors

- Basic selectors
- Combinatory selectors
- Attribute selectors
- Pseudo-class selectors
- Pseudo-element selectors

CSS Selectors: Basic

- Type Selector: Selects all elements of a specified type.
  - Syntax: `element`
  - Example: `p { color: blue; }`
- Universal Selector: Selects all elements on the page.
  - Syntax: `*`
  - Example: `* { margin: 0; padding: 0; }`
- Class Selector: Selects elements with a specific class attribute.
  - Syntax: `.classname`
  - Example: `.highlight { background-color: yellow; }`
- ID Selector: Selects a single element with a specific ID attribute.
  - Syntax: `#idname`
  - Example: `#header { font-size: 24px; }`
- Attribute Selector: Selects elements with a specific attribute.
  - Syntax: `[attribute]`
  - Example: `[type="text"] { border: 1px solid black; }`

CSS Selectors: Combinatory

- Combinatory selectors allow you to select elements based on their relationship to other elements.
- Common combinators:
  - Descendant (<space>): Selects all elements that are descendants of a specified element.
  - Child (>) : Selects all direct child elements specified by a parent element.
  - Adjacent sibling (+): Selects the next sibling element that follows a specified element.
  - General sibling (~): Selects all sibling elements that follow a specified element.

CSS Selectors: Attribute

- Attribute selectors allow you to select elements based on their attributes and attribute values.
- Common attribute selectors:
  - `[attribute]`: Selects all elements with the specified attribute.
  - `[attribute=value]`: Selects all elements with the specified attribute and value.
  - `[attribute =value]`: Selects all elements with an attribute value containing a specified word.
  - `[attribute|=value]`: Selects all elements with an attribute value starting with a specified value.

CSS Selectors: Pseudo-class and Pseudo-element

- Pseudo-classes are used to define the special state of an element.
  - `:hover` - Selects an element when the mouse pointer is over it.
  - `:active` - Selects an element when it is being activated by the user.
  - `:focus` - Selects an element when it has focus.
  - `:nth-child(n)` - Selects every nth child element.

- Pseudo-elements allow you to style certain parts of an element.
  - `::before` - Inserts content before the content of an element.
  - `::after` - Inserts content after the content of an element.
  - `::first-line` - Selects the first line of text within an element.
  - `::first-letter` - Selects the first letter of text within an element.

HTML Example with CSS Selectors I

Below is an HTML example that demonstrates the use of different CSS selectors and properties:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>CSS Example</title>
    <style>
        /* Basic selector */
        h1 {
            color: blue;
        }

        /* Combinatory selector */
        div p {
            font-size: 18px;
        }
```

HTML Example with CSS Selectors II

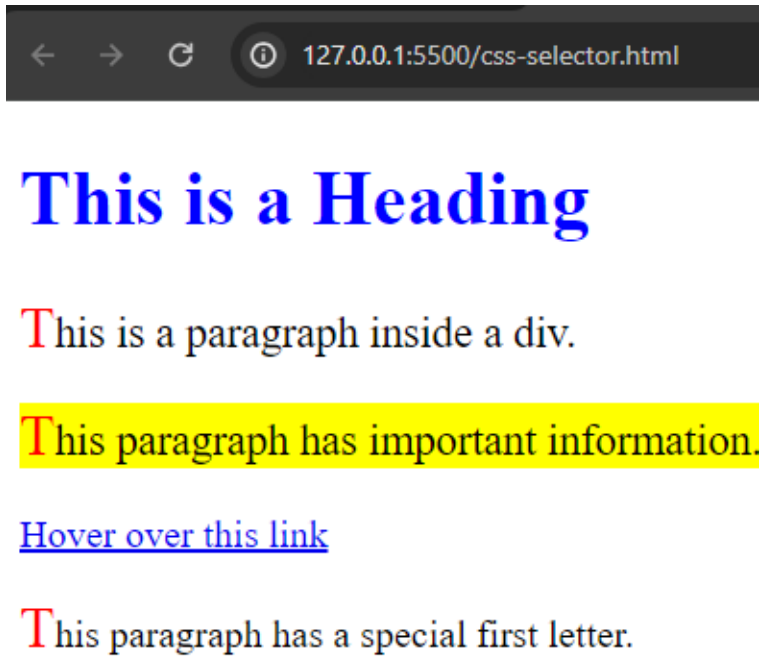
```
        /* Attribute selector */
        [data-info="important"] {
            background-color: yellow;
        }

        /* Pseudo-class selector */
        a:hover {
            text-decoration: underline;
        }

        /* Pseudo-element selector */
        p::first-letter {
            font-size: 24px;
            color: red;
        }
    </style>
</head>
```

HTML Example with CSS Selectors III

```
<body>
    <h1>This is a Heading</h1>
    <div>
        <p>This is a paragraph inside a div.</p>
    </div>
    <div>
        <p data-info="important">This paragraph has
    important information.</p>
    </div>
    <a href="#">Hover over this link</a>
    <p>This paragraph has a special first letter.</p>
</body>
</html>
```

HTML Example with CSS Selectors IV

Output:



Questionnaire

Short Answer Questions

1. How do you structure an ordered list (`<ol>`) with nested list items in HTML?
2. What are the required attributes for the `<form>` tag, and what do they specify?
3. How does CSS enhance the presentation of a web page?
4. How does external CSS contribute to the maintainability of a large website?
5. What is the purpose of the :nth-child pseudo-class, and how is it used?

Questionnaire

Long Answer Questions

1. Describe the structure and use of HTML tables, including attributes and nested elements. How can tables be used for layout, and why is this practice generally discouraged in modern web design?
2. List and explain the different types of lists available in HTML. How can you create nested lists, and what are the attributes that can be applied to customize the appearance and functionality of lists?
3. Discuss the pros and cons of using external CSS in large web applications. How does the use of external stylesheets promote reusability, and what are some strategies for organizing and managing CSS in complex projects?
4. Explain the concept of CSS selectors. Discuss the different types of selectors available, including basic, combinatory, attribute, pseudo-class, and pseudo-element selectors. Provide examples of how each type can be used in practical web development scenarios.

Thank you