# 701- HTML form Tag

The **&lt;form&gt; tag** is used to create an <u>HTML</u> form for user input. It serves as a container for various input elements like text fields, checkboxes, and buttons, enabling the collection of data for submission to a server or processing via client-side scripts.

**Note:** The <small>&lt;form&gt;</small> tag supports the <u>Global Attributes</u> and <u>Event Attributes</u> in HTML.

```
<html>


<body>
 <form action="/submit" method="POST">
  <h2>User Information</h2>
  <label for="fname">First Name</label>
  <input type="text" id="fname" name="fname" placeholder="Enter your first name" required><br><br>


  <label for="lname">Last Name</label>
  <input type="text" id="lname" name="lname" placeholder="Enter your last name" required><br><br>


  <input type="submit" value="Submit">
 </form>
</body>


</html>
```

**Output:**

## User Information

First Name  Enter your first name

Last Name  Enter your last name

Submit

## Common Form Elements

1. &lt;input&gt;
2. &lt;button&gt;
3. &lt;select&gt;
4. &lt;textarea&gt;
5. &lt;label&gt;
6. &lt;fieldset&gt;
7. &lt;legend&gt;

8. \<datalist\>
9. \<output\>

## Attributes

| Attributes | Descriptions |
|---|---|
| name | It provides a name for the form. |
| target | It specifies where to display the response after form submission (like a new window or the current window). |
| Action Attribute | Sends form data to the server upon submission. |
| Enctype Attribute | • **application/x-www-form-urlencoded:** Standard form data submission.<br>• **multipart/form-data:** Used for file uploads like images, documents, etc. |
| Methods | • **Get Method:** Limited URL character length, and used when we have to get data from somewhere like server.<br>• **Post Method:** No size limits, submissions cannot be bookmarked and used when we sends data to server. |
| accept-charset | It specifies the character encodings that is to be used for the form submission |
| autocomplete | It specifies whether a form should have autocomplete on or off |
| novalidate | It specifies that the form should not be validated before submitting |
| rel | It specifies the relationship between a linked resource and the current document. |

## Form with Radio button

In this example we creates a form with radio buttons for gender selection. It's properly structured with opening and closing \<form\> tags, labels, and radio input elements within a centered layout.

```
<html>


<body>
 <h1>Gender Validation</h1>
 <form action="#" method="post">
  <label for="male">Male</label>
  <input type="radio" id="male" name="gender" v alue="male">


  <label for="female">Female</label>
  <input type="radio" id="female" name="gender" value="female">


  <label for="other">Other</label>
  <input type="radio" id="other" name="gender" value="other"><br><br>
  <input type="button" value="Submit">
```

```
    </form>
  </body>

</html>
```

**Output:**

# Gender Validation

Male ○ Female ○ Other ○

Submit

### Form with Checkox

The form will display a list of checkboxes with labels. Users can select one or more options and click the Subscribe button to submit their choices.

```
<html>

<body>
 <h1>Newsletter Subscription</h1>
 <form action="/submit" method="POST">
  <p>Select your interests:</p>

  <input type="checkbox" id="tech" name="interests" value="Technology">
  <label for="tech">Technology</label><br>

  <input type="checkbox" id="sports" name="interests" value="Sports">
  <label for="sports">Sports</label><br>

  <input type="checkbox" id="music" name="interests" value="Music">
  <label for="music">Music</label><br>

  <input type="checkbox" id="movies" name="interests" value="Movies">
  <label for="movies">Movies</label><br><br>

  <input type="submit" value="Subscribe">
 </form>
</body>
```

```
</html>
```

**Output:**

# Newsletter Subscription

Select your interests:

- ☐ Technology
- ☐ Sports
- ☐ Music
- ☐ Movies

Subscribe

# 702 - HTML input Tag

**The \<input\> tag in HTML is used to collect user input in web forms. It supports various input types such as text, password, checkboxes, radio buttons, and more.**

An input field can be of various types depending upon the attribute type. The Input tag is an empty element that only contains attributes. For defining labels for the input element, **\<label\>** can be used.

**Note**: The \<input\> Tag supports the Global Attributes & the Event Attributes in HTML

## Syntax

```
<input type = "value" ... />
```

```
<html>


<body>
 <form>

   <label for="username">Username:</label>

   <input type="text" id="username" name="username">

 </form>
</body>
</html>
```
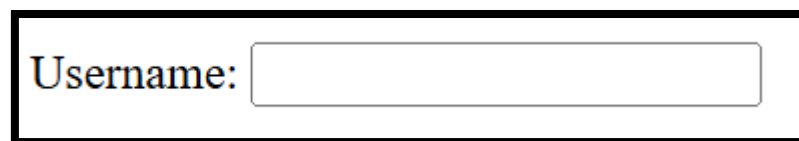
## Output:



- The \<label\> element with the for attribute associates the label "Username:" with the \<input\> field, enhancing accessibility.

- The \<input\> element of type "text" creates a single-line text input field where users can enter their username.

## Common \<input\> Types

| Input Type | Description |
|---|---|
| \<input type="text"\> | Single-line text input |
| \<input type="password"\> | Masked text input for passwords |
| \<input type="checkbox"\> | Toggle for selecting multiple options |
| \<input type="radio"\> | Single selection from multiple options |
| \<input type="submit"\> | Button to submit form data |
| \<input type="button"\> | General-purpose button |

| | |
|---|---|
| <input type="file"> | Input for uploading files |
| <input type="number"> | Input for numerical values |
| <input type="date"> | Input for selecting dates |
| <input type="email"> | Input for email addresses |
| <input type="color"> | Input for selecting colors |
| <input type="range"> | Slider for selecting a numeric value within a range |
| <input type="hidden"> | Hidden input for form data |
| <input type="image"> | Input using an image for form submission |

**Attributes**

| Attributes | Descriptions |
|---|---|
| **type** | It is used to specify the type of the input element. Its default value is text. |
| **value** | It is used to specify the value of the input element. |
| **placeholder** | Placeholder attribute is used to specify hint that describes the expected value of an input field. |
| **name** | It is used to specify the name of the input element. |
| **alt** | It is used to provide alternate text for the user, if they cannot view the image. |
| **autofocus** | It specifies that an element should automatically get focus when the page loads. |
| **checked** | It specifies that an element should be pre-selected (checked) when the page loads. |
| **disabled** | The disabled attribute specifies that the element should be disabled |
| **form** | The form attribute is used to specify one or more forms to which the <input> element belongs to. |
| **max** | It is used to specify the maximum value for an < input > element. |
| **required** | It specifies that an input field must be filled out before submitting the form. |
| **readonly** | It specifies that an input field is read-only. A read-only input field cannot be modified. |
| **accept** | It is used to specifies the types of files that the server accepts. |
| **align** | It is used to specifies the alignment of an image input. |
| **autocomplete** | It s used to specifies whether an <input> element should have autocomplete enabled |
| **dirname** | It is used to specifies that the text direction will be submitted. |
| **formaction** | It is used to specifies the URL of the file that will process the input control when the form is submitted |

| | |
|---|---|
| **formenctype** | It is used to specifies how the form-data should be encoded when submitting it to the server |
| **formmethod** | It is used to defines the HTTP method for sending data to the action URL |
| **formnovalidate** | It is used to defines that form elements should not be validated when submitted |
| **formtarget** | It is used to specifies where to display the response that is received after submitting the form |
| **height** | It is used to specifies the height of an <input> element |
| **list** | It is used to refers to a <datalist> element that contains pre-defined options for an <input> element |
| **maxlength** | It is used to specifies the maximum number of characters allowed in an <input> element |
| **min** | It is used to specifies a minimum value for an <input> element |
| **multiple** | It is used to specifies that a user can enter more than one value in an <input> element |
| **pattern** | It is used to specifies a regular expression that an <input> element's value is checked against |
| **size** | It is used to specifies the width, in characters, of an <input> element |
| **src** | It is used to specifies the URL of the image to use as a submit button |
| **step** | It is used to specifies the legal number intervals for an input field |
| **width** | It is used to specifies the width of an <input> element (only Ifor type="image") |

*More Examples of HTML input tag*

## Styled Password Input Field

```
<html>

<head>
 <style>
  input[type="password"] {
   border: 2px solid #4CAF50;
   border-radius: 4px;
   padding: 10px;
   width: 200px;
  }
 </style>
</head>


<body>
```

```
  <form>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password">
  </form>
</body>

</html>
```

## Output:



**In this example:**

- The <input> element of type "password" creates a field where users can enter sensitive information, with input characters masked for privacy.

- The CSS styles apply a green border, rounded corners, and padding to enhance the input field's appearance.

## Email Input Field with Placeholder

```
<html>

<head>
  <style>
    input[type="email"] {
      border: 1px solid #ccc;
      padding: 8px;
      font-size: 14px;
      width: 250px;
    }
  </style>
</head>

<body>
  <form>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" placeholder="example@example.com">
  </form>
</body>
```
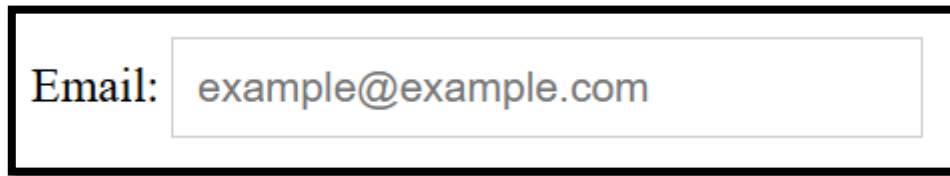
```
</html>
```

## Output:



**In this example:**

- The <input> element of type "email" provides a field for users to enter an email address, with built-in validation for correct email formatting.

- The placeholder attribute offers a hint to the user about the expected input format, and the CSS styles define the input field's border, padding, font size, and width.

## Best Practices for HTML <input> Elements

- **Use Descriptive Labels**: Ensure each input field has a corresponding <label> to enhance accessibility and usability.

- **Implement Client-Side Validation:** Utilize HTML5 validation attributes like required, pattern, and maxlength to provide immediate feedback and reduce invalid submissions.

- **Choose Appropriate Input Types**: Select input types that match the expected data (e.g., email for email addresses, number for numeric input) to improve user experience and leverage built-in validation

# 703 - HTML button Tag

The **\<button\> tag in HTML** is used to **create clickable buttons** on a web page. It can be used to submit forms, trigger JavaScript functions, or perform other interactive tasks.

```
<html>


<body>

 <button type="button" onclick="alert('Welcome to ProdipAlways')">

  Click Here

 </button>

</body>


</html>
```
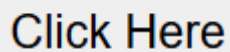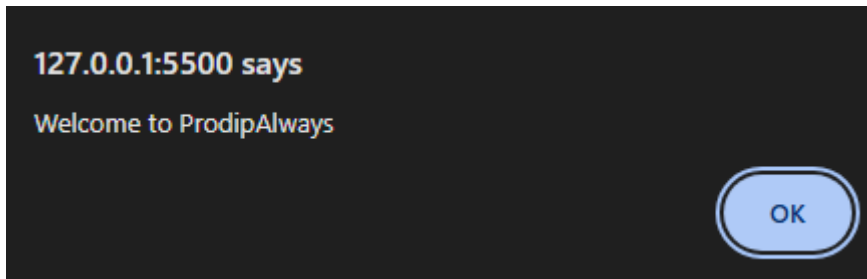
## Output:

Before Click



After Click



## Note:

- It is important to specify the type attribute for a button element to inform the browser of the button's function.

- The HTML \<button\> tag supports the Global Attribute and Event Attribute in HTML.

## Attributes:

The various attributes that can be used with the "button" tag are listed below:

| Attributes | Descriptions |
|------------|--------------|
| autofocus | Sspecifies whether the button should automatically get focus or not when the page loads |

| | |
|---|---|
| <u>disabled</u> | indicate whether the element is disabled or not. If this attribute is set, the element is disabled. |
| <u>form</u> | Create a form for user input. There are many elements that> are used within the >form tag. |
| <u>formaction</u> | Sspecifies where to send the data of the form. |
| <u>formnovalidate</u> | Sspecifies that the Input Element should not be validated when submitting the form. |
| <u>formenctype</u> | Sspecifies that the form data should be encoded when submitted to the server. |
| <u>formmethod</u> | Sspecifies the HTTP method used to send data while submitting the form. |
| <u>formtarget</u> | Sspecifies the name or a keyword that indicates where to display the response after submitting the form. |
| <u>type</u> | Sspecifies the type of button for button elements. It is also used in <input> element to Sspecifies the type of input to display. |
| <u>value</u> | Sspecifies the value of the element with which it is used. It has different meanings for different HTML elements. |

## Button with custom CSS

```
<html>

<head>

  <style>

    button {

      padding: 10px 20px;

      font-size: 16px;

      margin-top: 40px;

      border: none;

      cursor: pointer;

      border-radius: 4px;

      transition: background-color 0.3s ease;

      color: #fff;

    }

  </style>

</head>

<body>

  <button style="background-color: #3daaf3; ">Primary</button>

  <button style="background-color: #3bda7d; ">Secondary</button>
```
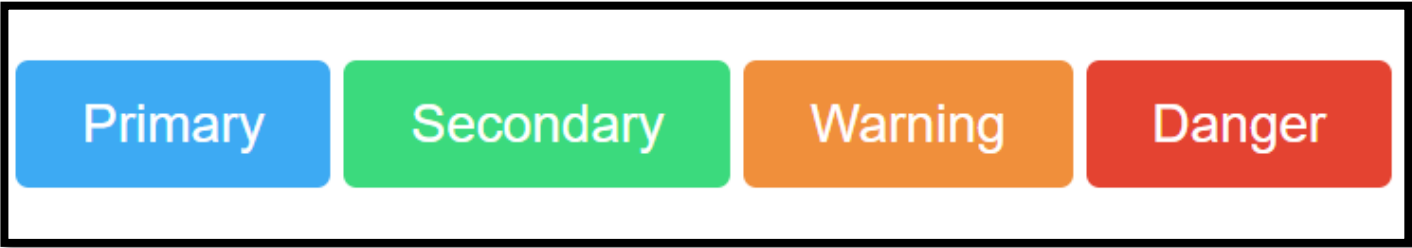
```
    <button style="background-color: #f08f3b; ">Warning</button>

    <button style="background-color: #e44331; ">Danger</button>

</body>

</html>
```

**Output:**

# 704 - button tag vs input type="button" attribute

The **<input type="button">** and **<button>** tag are two different approach to use Button in HTML document. The difference between them is that the buttons created with the <button> tag offer many contributing possibilities than <input type="button"> attribute. The <button> can have content inside it. The <button> tag permits phrasing content inside button element contents like text or images etc, along work with type functionality defined. But the input type="button" attribute does not permit content. For example, a button tag that contains an image functions and may resemble an input tag whose type is set to "image", but the button element type allows content. Both examples below very clearly shows the difference.

**Note:** The <button> tag uses opening and closing bracket but <input type="button"> attribute uses only single tag.

**Example 1:** This example uses <input type="button"> attribute to create button in HTML document.

```
<!DOCTYPE html>

<html>


<head>
 <script>
   function msg() {

     alert("values submitted");

   }
 </script>
</head>


<body style="text-align:center;">


 <h1 style="color:green;">

   Prodip Always

 </h1>


 <h2>Using Input type</h2>


 <form>
  <label for="submit">

    Enter value

  </label>


  <input type="text">


  <input type="button" id="submit" onclick="msg()" value="button">
```
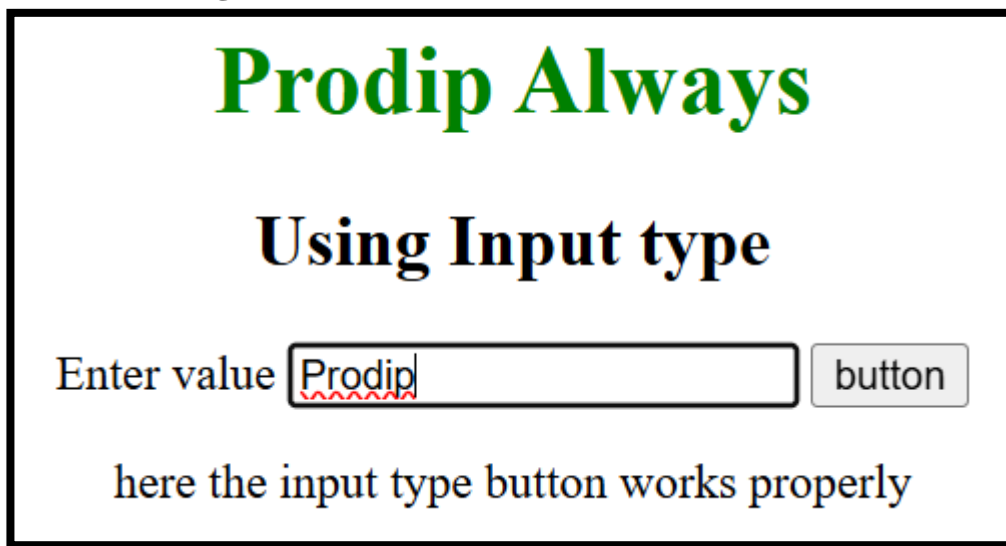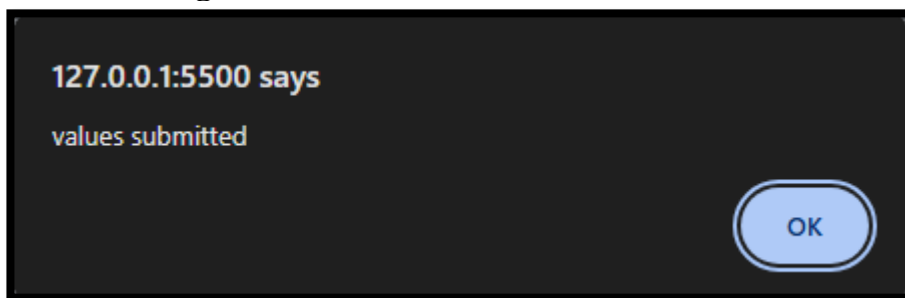
```
  </form>
  <p>
    here the input type button
    works properly
  </p>
</body>

</html>
```

**Output:**

- **Before clicking the button:**

# Prodip Always

## Using Input type

Enter value Prodip _____ button

here the input type button works properly

- **After clicking the button:**

127.0.0.1:5500 says

values submitted

OK

**Example 2:** This example uses <button> tag to create button in HTML document.

```
<!DOCTYPE html>
<html>

<body style="text-align:center;">

  <h1 style="color:green;">
    Prodip Always
  </h1>
```

```html
<h2>The Button Element</h2>

<form method="post">
  <label for="uname"><b>Username</b></label>

  <input type="text" placeholder="Enter Username" name="uname" required>

  <br>

  <label for="psw"><b>Password</b></label>

  <input type="password" placeholder="Enter Password" name="psw" required>

  <br>

  <button type="submit"><img src="login-button-png-4.jpg" height="40" width="100">
  </button><br>

  <button type="button" class="cancelbtn">Cancel</button>

  <span class="psw">Forgot <a href="#">password?</a></span>
 </form>
</body>

</html>
```

**Output:**

- **Before clicking the button:**

- **After clicking the button:**

  **Prodip Always**

  **The Button Element**

  **Username** Enter Username

  **Password**

  ⚠ Please fill out this field.

  Cancel Forgot [password?](password)

# 705 - HTML select Tag

The HTML <select> tag is used to create a drop-down list for user input, containing <option> tags to display the available choices. It provides functionality for selecting one or multiple options from a list.

**Note:** The <select> tag is used in a form to receive user responses.

**Syntax**

```
<select>
   <option>
   </option>
   ...
</select>
```
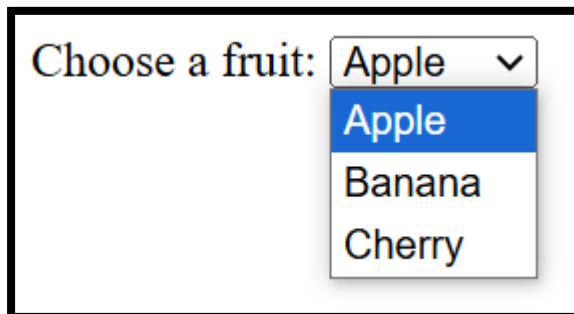
```
<html>


<body>
 <form>
   <label for="fruits">Choose a fruit:</label>
   <select id="fruits" name="fruits">
     <option value="apple">Apple</option>
     <option value="banana">Banana</option>
     <option value="cherry">Cherry</option>
   </select>
 </form>
</body>


</html>
```

## Output:

Choose a fruit: Apple ⌄
Apple
Banana
Cherry

**In this example:**

- The <label> element with the for attribute associates the label with the <select> element, enhancing accessibility.

- The <select> element with the id and name attributes defines the drop-down list, containing multiple <option> elements representing the available choices.

**Attributes**

The table below shows the required attributes and their respective description:

| Attribute | Description |
|-----------|-------------|
| **autofocus** | Automatically focuses the dropdown when the page loads. |
| **disabled** | Disables the dropdown, making it un-clickable and unusable. |
| **form** | Specifies one or more forms that the select element belongs to. |
| **multiple** | Allows the user to select multiple values from the dropdown. |
| **name** | Defines a name for the dropdown, used to reference form data or in JavaScript. |
| **required** | Ensures the user selects a value before submitting the form. |
| **size** | Defines the number of visible options in the dropdown list. |

## Basic Dropdown Menu

```html
<html>
<head>
</head>
<body>
  <form>
    <label for="colors">Choose a color:</label>
    <select id="colors" name="colors">
      <option value="red">Red</option>
      <option value="green">Green</option>
      <option value="blue">Blue</option>
    </select>
  </form>
</body>
</html>
```
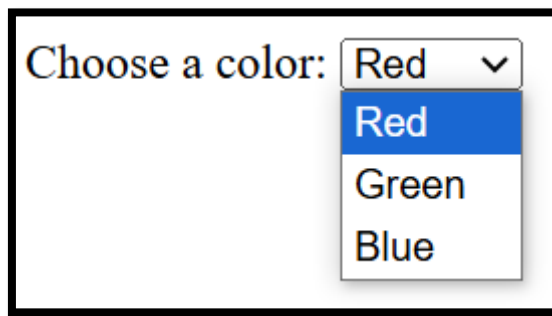
**Output:**

**In this example:**

- The **\<select\>** element with the id and name attributes defines a dropdown menu labeled "Choose a color:".

- The **\<option\>** elements provide the selectable options: Red, Green, and Blue.

## Styled Dropdown Menu

```html
<html>

<head>

  <style>

    label {

      font-family: Arial, sans-serif;

      font-size: 14px;

    }

    select {

      width: 150px;

      padding: 5px;

      border: 1px solid #ccc;

      border-radius: 4px;

      font-family: Arial, sans-serif;

      font-size: 14px;

    }

  </style>

</head>

<body>

  <form>

    <label for="cars">Select a car brand:</label>

    <select id="cars" name="cars">

      <option value="volvo">Volvo</option>
```

```html
        <option value="saab">Saab</option>

        <option value="mercedes">Mercedes</option>

        <option value="audi">Audi</option>

      </select>

   </form>

</body>

</html>
```

## Output:
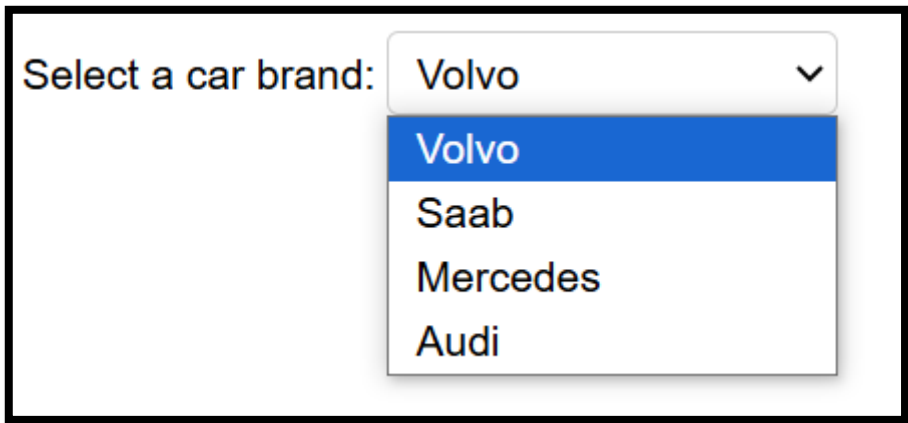


## In this example:

- The **<select>** element is styled with a fixed width, padding, border, and border-radius to enhance its appearance.

- The <u>font family</u> and size are set for both the <label> and <select> elements to ensure consistent typography.

## Dropdown Menu with Placeholder

```html
<html>

<head>

   <style>

     select {

        width: 200px;

        padding: 6px;

        border: 1px solid #888;

        border-radius: 5px;

        background-color: #f9f9f9;

        font-family: Verdana, sans-serif;
```

```
        font-size: 13px;

      }

      option[disabled] {

        color: #999;

      }

    </style>

</head>

<body>

  <form>

      <label for="fruits">Choose a fruit:</label>

      <select id="fruits" name="fruits">

          <option value="" disabled selected>Select a fruit</option>

          <option value="apple">Apple</option>

          <option value="banana">Banana</option>

          <option value="cherry">Cherry</option>

      </select>

  </form>

</body>

</html>
```

## Output:

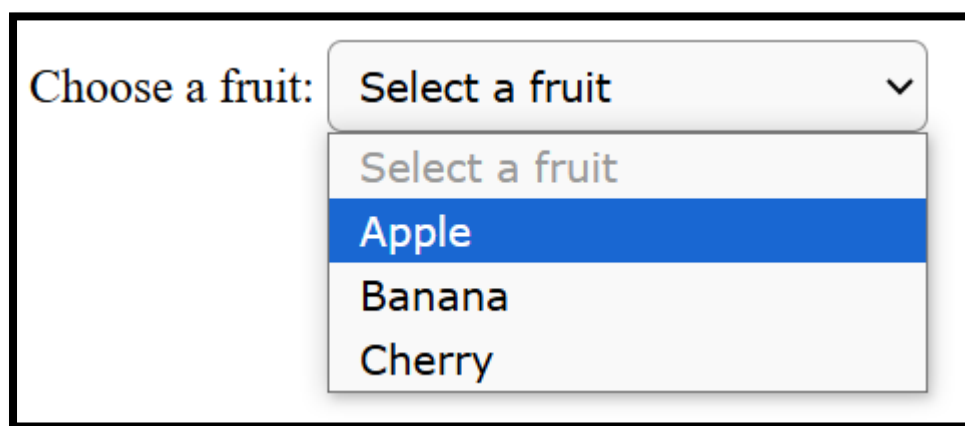Choose a fruit: [ Select a fruit ⌄ ]
- Select a fruit
- **Apple**
- Banana
- Cherry

**In this example:**

- The first <option> serves as a placeholder with the text "Select a fruit," is disabled, and is selected by default.

- The <select> element is styled with specific dimensions, padding, border properties, background color, and font settings to improve usability and aesthetics.

Best Practices for Using the HTML <select> Tag

- **Associate Labels Clearly:** Use the <label> element with the for attribute matching the <select> element's id to enhance accessibility and usability.

- **Provide a Default Option:** Include a default <option> with a value of an empty string and the selected attribute to prompt users to make a selection.

- **Group Related Options:** Utilize the <optgroup> element to categorize options within the dropdown, improving navigation for users.

# 706 - HTML <textarea> name Attribute

The **name** attribute for the **<textarea> element** is essential for several reasons:

- **Form Data Submission:** The **name** attribute specifies the name of the **<textarea>** control, which becomes the key in the key-value pair data submitted with the form.

- **Server-side Processing:** On the server, this key is used to access the corresponding value entered by the user, which is crucial for handling form data.

- **Client-side Accessibility:** In client-side scripting, such as JavaScript, the **name** attribute can be used to manipulate the text area's value dynamically.

**Syntax:**

**<textarea** name="text**">**

**Attribute Values:** It contains the value i.e name which specifies the name for the <Textarea> element.

**Example:**

Here's a simple HTML code example that demonstrates the use of the name attribute in a <textarea> element.

```
<!DOCTYPE html>
<html>

<head>
  <title>Textarea Name Attribute Example</title>
</head>

<body>
  <h3>Textarea Example</h3>
  <p>Please enter your comments:</p>
  <textarea name="userComments" rows="4" cols="50">
    Enter your text here...
    </textarea>
  <br>
</body>

</html>
```

**Output:**

**Textarea Example**

Please enter your comments:

Enter your text here...

**In this example:**

- **name="userComments"** – Here, the **name** attribute is a key component of the **<textarea>** element. It assigns a unique identifier to the textarea's data, making it accessible when the form data is processed on the server side. For example, when the form is submitted, the server can reference "**userComments**" to retrieve the value entered by the user.

- **rows="4″ and cols="50″**: These attributes specify the size of the textarea. rows="4″ sets the number of text lines visible at a time in the textarea to 4, and cols="50″ sets the width of the textarea to accommodate 50 characters per line.

- **Text within the <textarea>:** The text **"Enter your text here…"** could serve as a placeholder to inform users where to type their comments. However, it's actually part of the content and will need to be erased by the user before entering their comments, unlike a placeholder attribute which disappears when typing starts.

## Best Practices for Using the name Attribute

- Ensure that each name attribute in your form is unique so that each data field can be individually identified.

- Use meaningful names that relate to their function, making it easier to handle data on the server side.

# 707 - HTML &lt;label&gt; Tag

The **&lt;label&gt; HTML** element represents a caption for a form element in a user interface. It improves accessibility by linking text to form elements. When a user clicks on the **label**, it automatically focuses on or activates the **associated input**, such as text fields, checkboxes, or radio buttons. This helps make forms more **user-friendly** and easier to navigate.

The &lt;label&gt; tag can be used in two ways:

- **Using the for attribute**: The label is connected to an input field by using the for attribute, which matches the id of the input.

- **Wrapping the input inside the label:** The input element can also be placed directly inside the label, where no for or id attributes are needed.

HTML Inputs and Labels

In HTML, the **&lt;label&gt; tag** works hand-in-hand with input elements. It allows users to click on the label, which then selects or focuses on the associated input field. This connection between inputs and labels enhances both usability and accessibility for forms, making it easier to interact with them.

Supported Tags

The &lt;label&gt; tag can be defined with the following Tags:

- &lt;button&gt;

- &lt;input&gt;

- &lt;meter&gt;

- &lt;output&gt;

- &lt;progress&gt;

- &lt;select&gt;

- &lt;textarea&gt;

**Syntax**

&lt;label&gt; form content... &lt;/label&gt;

Attribute Value

| Attribute Value | Descriptions |
| --- | --- |
| **for** | It refers to the input control that this label is for. Its value must be the same as the value of the input control's "id" attribute. |
| **form** | It refers to the form to which the label belongs to. |

HTML &lt;label&gt; Tag Examples

Here are some examples of how **labeling** can be used:

# Example 1: Using the <label> Tag with the For Attribute

This example illustrates the basic usage of the <label> tag in HTML. Here, we will use the <input> tag outside of the <label> tag.

```
<!DOCTYPE html>
<html>

<head>
 <title>
   HTML label Tag
 </title>
</head>

<body>

 <strong>HTML label Tag</strong>
 <br><br>
 <form>

   <!-- Starts label tags from here -->
   <label for="student">
    Student
   </label>
   <input type="radio" name="Occupation" id="student" value="student"><br>

   <label for="business">
    Business
   </label>
   <input type="radio" name="Occupation" id="business" value="business"><br>

   <label for="other">
    Other
   </label>
   <!-- Ends label tags here -->

   <input type="radio" name="Occupation" id="other" value="other">
 </form>
</body>

</html>
```

**Output:**



Example 2: Using the <input> Tag Inside the <label> Tag

In this example, we will demonstrate how to use the <input> tag inside the <label> tag.

```
<!DOCTYPE html>
<html>

<body>
 <strong> HTML label Tag</strong>
 <br><br>
 <form>

  <!-- label tag starts from here -->
  <label>
   Male
   <input type="radio" name="gender" id="male" value="male" />
  </label><br />

  <label>
   Female
   <input type="radio" name="gender" id="female" value="female" />
  </label><br />

  <label>
   Other
   <input type="radio" name="gender" id="other" value="other" />
  </label>

  <!-- label tag ends from here -->
 </form>
</body>
```

```
</html>
```

**Output:**

**HTML label Tag**

Male ○
Female ○
Other ○

# 708 - HTML fieldset form Attribute

The **form** attribute in the <fieldset> tag specifies which **<form>** element the <fieldset> belongs to.

However, when the <u>fieldset</u> is not inside the <form> (i.e., outside of the <form> element), the **form** attribute is used to explicitly link the <fieldset> to a specific form.

**Syntax**

**<fieldset** form="form_id"**>**

## Attribute Values:

| Value | Description |
|-------|-------------|
| form_id | The form attribute contains the form_id, which specifies the form that the <fieldset> element belongs to. The value should be the id of a <form> element. |

**Example:**

```
<!DOCTYPE html>
<html>


<head>
 <title>
  HTML fieldset form Attribute
 </title>
 <style>
  h1,
  h2,
  .title {
   text-align: center;
  }


  fieldset {
   width: 50%;
   margin-left: 22%;
  }


  h1 {
   color: green;
  }
 </style>
</head>
```

```
<body>
 <h1>ProdipAlways</h1>
 <h2>
   HTML &lt;fieldset&gt;
   form Attribute
 </h2>
 <form id="myprodip">
  <div class="title">
    Suggest article for video:
  </div>
 </form>


 <fieldset form="myprodip">
  <legend>JAVA:</legend>
  Title:
  <input type="text">
  <br> Link:
  <input type="text">
  <br> User ID:
  <input type="text">
 </fieldset>
 <br>
 <fieldset form="myprodip">
  <legend>PHP:</legend>
  Title:
  <input type="text">
  <br> Link:
  <input type="text">
  <br> User ID:
  <input type="text">
 </fieldset>

</body>

</html>
```

**Output:**

## HTML <fieldset> form Attribute

Suggest article for video:

JAVA:
Title: [                    ]
Link: [                    ]
User ID: [                    ]

PHP:
Title: [                    ]
Link: [                    ]
User ID: [                    ]

**In this example:**

- **<form id="mygeeks">**: A form element identified by id="mygeeks". It contains no form controls directly under it but serves as a logical container for the inputs defined in the subsequent fieldsets.

  - **<div class="title">**: A division formatted with centered text, serving as a section title within the form.

- **Two <fieldset> Elements**:

  - Both fieldsets use the form="mygeeks" attribute, explicitly associating them with the form whose id is "mygeeks". This connection allows inputs within these fieldsets to be part of the "mygeeks" form data, even though they are not nested within the <form> tag in the HTML structure.

  - **First <fieldset>**

    - **<legend>**: Defines a caption for the fieldset, "JAVA:", indicating that the inputs within this fieldset are related to Java topics.

    - **Inputs**: Contains three input fields for "Title", "Link", and "User ID". Each label is followed by an input field, separated by line breaks (<br>).

  - **Second <fieldset>:**

    - Similar to the first, with a <legend> of "PHP:", and the same structure of inputs for "Title", "Link", and "User ID".

**Purpose of the form Attribute:**

The **form** *attribute of the* **<fieldset> tag** *links the fieldset to a form elsewhere in the HTML document. This is particularly useful in complex web pages where the form elements are not physically contained within the* **<form> tag** *due to styling or layout requirements. By using the form attribute, developers can maintain a logical association between the* **<fieldset>** *and its corresponding <form> without altering the structure of the HTML document.*

# 709 - HTML legend Tag

The HTML <legend> tag is used to provide a title or caption for its parent <fieldset> element. It must be the first child inside the <ins>fieldset</ins> to describe the grouped content clearly.

The <legend> tag also supports the <ins>Global Attributes</ins> and <ins>Event Attributes in HTML</ins>.

## Syntax

<legend> Text </legend>

## Attributes

| Attribute | Description | Example Value |
|---|---|---|
| **align** | Specifies the alignment of the legend text. | left, center, right |

**Example 1:** In this example, we are using the <legend> tag inside a <fieldset> to group and label form elements related to student information.

```
<!DOCTYPE html>
<html>

<body>
 <h1>ProdipAlways</h1>
 <strong>HTML Legend Tag</strong>
 <form>
  <fieldset>
   <!-- Legend tag using -->
   <legend>STUDENT::</legend>
   <label>Name:</label>
   <input type="text">
   <br><br>
   <label>Email:</label>
   <input type="text">
   <br><br>
   <label>Date of birth:</label>
   <input type="text">
   <br><br>
   <label>Address:</label>
   <input type="text">
   <br><br>
   <label>Enroll No:</label>
   <input type="text">
  </fieldset>
```
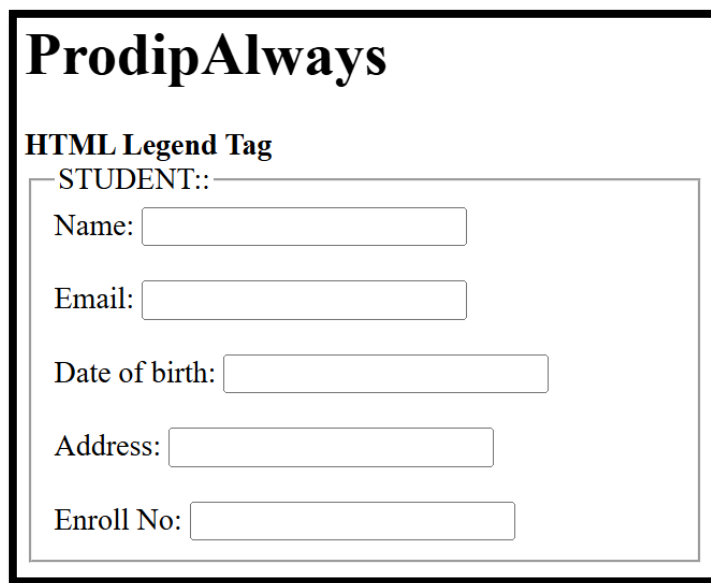
```
    </form>
  </body>

</html>
```

## Output

**ProdipAlways**

**HTML Legend Tag**

```
┌─STUDENT::──────────────────────────────┐
│                                         │
│  Name: [_____]               │
│                                         │
│  Email: [_____]              │
│                                         │
│  Date of birth: [_____]       │
│                                         │
│  Address: [_____]              │
│                                         │
│  Enroll No: [_____]          │
│                                         │
└─────────────────────────────────────────┘
```

**Example 2:** In this example, the <legend> tag is used to create a titled label for the grouped form elements inside a <fieldset>, with customized styling.

```
<!DOCTYPE html>
<html>

<head>
 <style>
   form {
     width: 50%;
   }

   legend {
     display: block;
     padding-left: 10px;
     padding-right: 10px;
     border: 3px solid green;
     background-color: tomato;
     color: white;
     ;
   }

   label {
```

```
      display: inline-block;
      float: left;
      clear: left;
      width: 90px;
      margin: 5px;
      text-align: left;
    }

    input[type="text"] {
      width: 250px;
      margin: 5px 0px;
    }

    .gfg {
      font-size: 40px;
      color: green;
      font-weight: bold;
    }
  </style>
</head>

<body>
  <div class="gfg">ProdipAlways</div>
  <h2>HTML Legend Tag</h2>
  <form>
    <fieldset>
      <!-- Legend tag using -->
      <legend>STUDENT:</legend>
      <label>Name:</label>
      <input type="text">
      <br>
      <label>Email:</label>
      <input type="text">
      <br>
      <label>Date of birth:</label>
      <input type="text">
      <br>
      <label>Address:</label>
      <input type="text">
      <br>
```
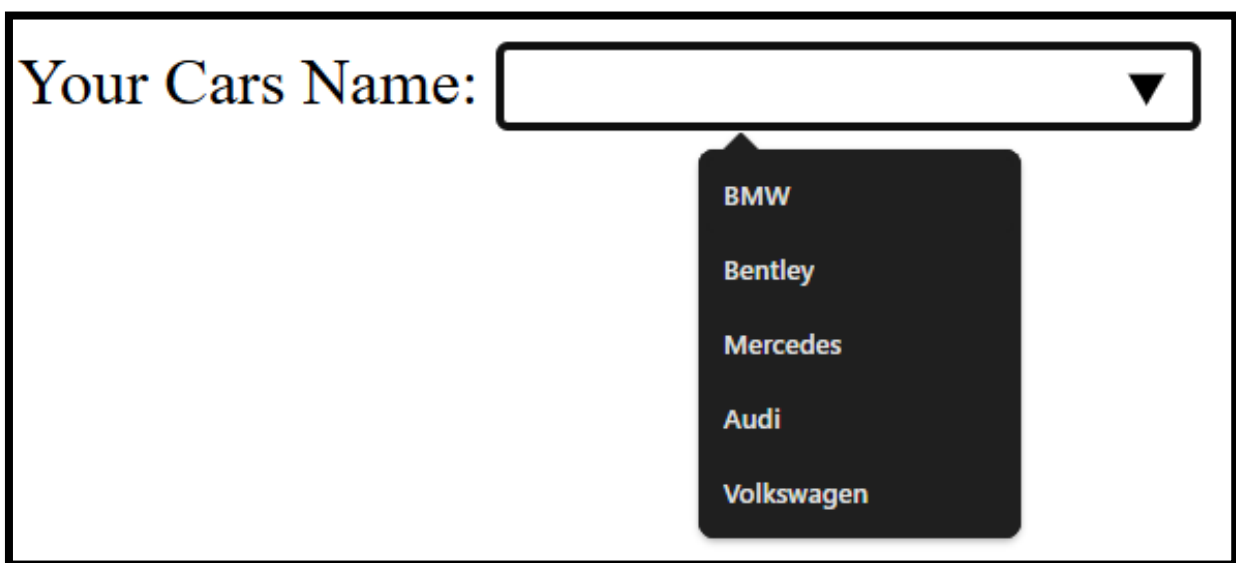
```
      <label>Enroll No:</label>

      <input type="text">

    </fieldset>

  </form>

</body>


</html>
```

**Output**

# 710 - HTML datalist Tag

**The &lt;datalist&gt; tag** is used to provide an *in* autocomplete feature in the <u>HTML</u> files. It contains a list of predefined options that the user can select from, making it easier to enter data. The &lt;datalist&gt; tag is often used in conjunction with an <u>&lt;input&gt;</u> element of type "text" or "search".

```
<!DOCTYPE html>
<html>


<body>
 <form action="">
   <label>Your Cars Name: </label>
   <input list="cars">
   <datalist id="cars">
    <option value="BMW" />
    <option value="Bentley" />
    <option value="Mercedes" />
    <option value="Audi" />
    <option value="Volkswagen" />
   </datalist>
 </form>
</body>

</html>
```

## Output:

Your Cars Name:
- BMW
- Bentley
- Mercedes
- Audi
- Volkswagen

## Syntax

```
<input list="datalist-id" name="input-name" id="input-id" placeholder="Enter value...">
```

```
<datalist id="datalist-id">

  <option value="Option 1">

  <option value="Option 2">

  <option value="Option 3">

  ...

</datalist>
```

## Features of <datalist> Tag

- **Autocomplete:** When a user starts typing in the associated <input> field, the browser suggests matching options from the <datalist>.

- **User Input:** Users can either select from the list of options or enter a custom value that is not in the list.

- **Cross-browser Support:** Most modern browsers support the <datalist> element, but it's always good to check for compatibility if you're targeting older browsers.

## More Example

```
<!DOCTYPE html>
<html>


<body>
 <form action="">
  <label>Your Cars Name: </label>
  <input list="cars" id="carsInput" />
  <datalist id="cars">
   <option value="BMW" />
   <option value="Bentley" />
   <option value="Mercedes" />
   <option value="Audi" />
   <option value="Volkswagen" />
  </datalist>


  <button onclick="datalistcall()" type="button">
   Click Here
  </button>
 </form>
 <p id="output"></p>


 <script type="text/javascript">
```

```
    function datalistcall() {

      var o1 = document.getElementById("carsInput").value;

      document.getElementById("output").innerHTML =

        "You select " + o1 + " option";

    }

  </script>

</body>


</html>
```

## Output:

Before Click



After Click

# 711 - HTML Tag

The HTML tag is used to represent the result of a calculation performed by the client-side script such as JavaScript. The tag is a new tag in HTML 5, and it requires a starting and ending tag.

It also supports the Global Attributes and Event Attributes in HTML. The content within the tag can be manipulated dynamically through JavaScript.

## Syntax:

Results...

## Attributes:

| Attribute Value | Description |
|:---:|:---:|
| **for** | This attribute contains an attribute **value** *element_id* which is used to specify the relation between result and calculations. |
| **form** | This attribute contains an attribute **value** *form_id* which is used to specify one or more forms of output elements. |
| **name** | This attribute contains an attribute **value** *name* that is used to specify the name of the output element. |

**Example 1:** In this example, we will see the implementation of output tag with an example.

```
<!DOCTYPE html>
<html lang="en">

<head>
 <title>Document</title>
 <style>
  body {
    text-align: center;
  }
 </style>
</head>

<body>
 <h1 style="color: green;">
   ProdipAlways
 </h1>
 <h2>
   HTML output Tag
 </h2>
```

```
 <form oninput="sumresult.value =
   parseInt(A.value) + parseInt(B.value) +
   parseInt(C.value)">
  <input type="number" name="A" value="50" /> +
  <input type="range" name="B" value="0" /> +
  <input type="number" name="C" value="30" />
  <br>
  <!-- output tag -->
  Result: <output name="sumresult"></output>
 </form>
</body>

</html>
```

**Output:**



**Example 2:** In this example, <output> tag is used with for and form attribute.

```
<!DOCTYPE html>
<html lang="en">

<head>
 <title>Document</title>
 <style>
  body {
    text-align: center;
  }
 </style>
</head>

<body>
```

```html
<h1 style="color: green;">
  ProdipAlways
</h1>
<h2>
  HTML output Tag
</h2>
<form oninput="sumresult.value =
    parseInt(A.value) + parseInt(B.value) +
    parseInt(C.value)">
  <input type="number" name="A" value="50" /> +
  <input type="range" name="B" value="0" /> +
  <input type="number" name="C" value="50" />
  <br /> Submit Result:
  <!-- output tag -->
  <output name="sumresult" for="A B C">
  </output>
  <br>
  <input type="submit">
</form>
</body>

</html>
```

**Output:**

# ProdipAlways

## HTML output Tag

`[50]` + `●━━━━━━━` + `[50]`

Submit Result:

[ Submit ]