

HTML Form Elements

HTML form elements are used to store user input. There are different types of form elements such as the text box, check box, drop-down, submit button, etc. For example,

```
<form>
  <label for="name">Name:</label>
  <input type="text" name="name"><br><br>
  <label for="sex">Sex:</label>
  <input type="radio" name="sex" id="male" value="male">
  <label for="male">Male</label>
  <input type="radio" name="sex" id="female" value="female">
  <label for="female">Female</label> <br><br>
  <input type="submit" value="Submit">
</form>
```

Browser Output

A screenshot of a web browser window. The browser has a light orange header bar with a globe icon on the left and three colored window control buttons (red, yellow, green) on the right. Below the header, the rendered HTML form is displayed. It consists of a text input field preceded by the label "Name:", followed by two radio buttons labeled "Male" and "Female" preceded by the label "Sex:". At the bottom of the form is a rectangular button with the text "Submit".

Name:

Sex: ☐ Male ☐ Female

Here, `<input>` and `<label>` are all HTML Form Elements.

HTML Form Elements

There are various HTML form elements for various input types. The form elements available in HTML5 are as follows:

- HTML `<input>` tag
- HTML `<label>` tag
- HTML `<button>` tag
- HTML `<select>`, `<option>` and `<optgroup>` tags
- HTML `<textarea>` tag
- HTML `<fieldset>` tag
- HTML `<legend>` tag
- HTML `<datalist>` tag
- HTML `<output>` tag

My feedback form

- Name:
- Email:
- Password:
- Please check all the emotions that apply to you:
 - Angry ☒
 - Sad ☐
 - Happy ☐
 - Ambivalent ☐
- How satisfied were you with our service?
 - Very satisfied ☒
 - Satisfied ☐
 - Didn't care ☐
 - Dissatisfied ☐
 - Very dissatisfied ☐
- Further comments:
- Bio photo:
- Location visited:
-

HTML Input Tag

The HTML `<input>` tag defines the field where the user can enter data. The type attribute determines what type of user input is taken.

```
<input type="text" name="firstname">
```

Browser Output



Here,

- *type* - determines the type of input the `<input>` tag takes
 - *name* - specifies the name of the input which is what the data is named as when submitting the data to a server.
-

Different Input Types

The various types of input tags available in HTML5 are:

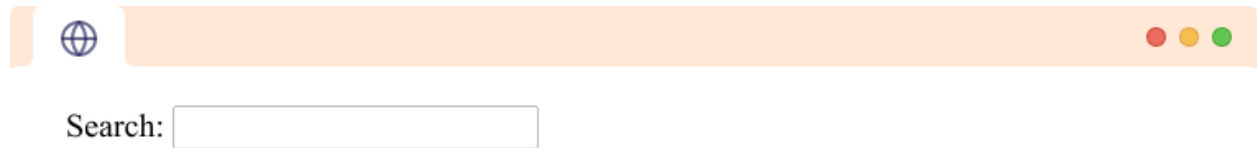
1. [text](#) - creates a single-line text field (default)
 2. [button](#) - creates a button with no default functionality
 3. [checkbox](#) - creates a checkbox
 4. [color](#) - creates a color picker
 5. [date](#) - creates a date picker
 6. [datetime-local](#) - creates a date and time picker
 7. [email](#) - creates an input field that allows the user to input a valid email address
 8. [file](#) - creates an input field that lets the user upload a file or multiple files
 9. [image](#) - creates a button using an image
 10. [month](#) - creates an input field that lets the user enter month and year
 11. [password](#) - creates an input field that lets the user enter information securely
 12. [radio](#) - creates a radio button
 13. [range](#) - creates a range picker from which the user can select the value
 14. [reset](#) - creates the button which clears all the form values to their default value
 15. [search](#) - allows user to enter their search queries in the text fields
 16. [submit](#) - allows user to submit form to the server
 17. [time](#) - creates an input field that accepts time value
 18. [url](#) - lets the user enter and edit a URL
-

1. Input Type text

The input type `text` is used to create single-line text fields. It is the default input type.

```
<label for="name">Search: </label>  
<input type="text" id="name">
```

Browser Output



The input type `text` can also contain *minlength*, *maxlength*, and *size* attributes. For example,

```
<label for="name">Name</label>
<input type="text" id="name" minlength="4" maxlength="8">
```

Browser Output



In the above example, values are only allowed between the length of **4** to **8** characters.

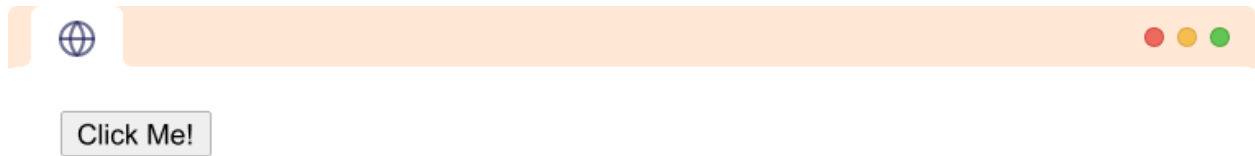
Note: If the *type* attribute is not provided, the tag becomes the type of text.

2. Input Type button

The input type `button` is used to create a button with no default functionality. For example,

```
<input type="button" value="Click Me!">
```

Browser Output



The text inside the value attribute is displayed in the button.

Note: Generally, javascript is used to add functionality to such buttons.

3. Input Type checkbox

The input type `checkbox` is used to create a checkbox input. For example,

```
<input type="checkbox" id="subscribe" value="subscribe">  
<label for="subscribe">Subscribe to newsletter!</label><br>
```

Browser Output (checkbox unselected)



The checkbox can be toggled between selected and not selected.

Browser Output (checkbox selected)



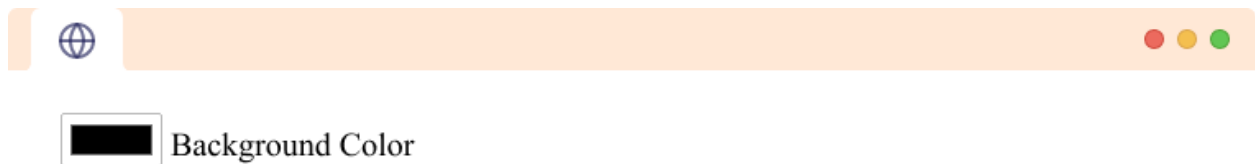
The value of the checkbox is included in the form data only if the checkbox is selected and is omitted if the checkbox is not selected.

4. Input Type Color

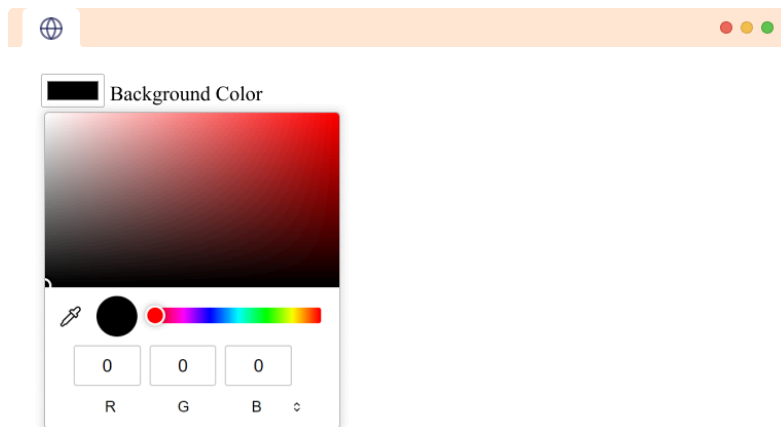
The input type `color` is used to create an input field that lets the user pick a color. For example,

```
<input type="color" id="background">  
<label for="background">Background Color</label>
```

Browser Output (before expanding)



Browser Output (after expanding)



The color picker is inbuilt into the browser. Users can also manually enter the hex code of the color instead. The UI for the color picker differs from browser to browser.

5. Input Type date

The input type `date` is used to create an input field that lets the user input a date using the date picker interface from the browser. For example,

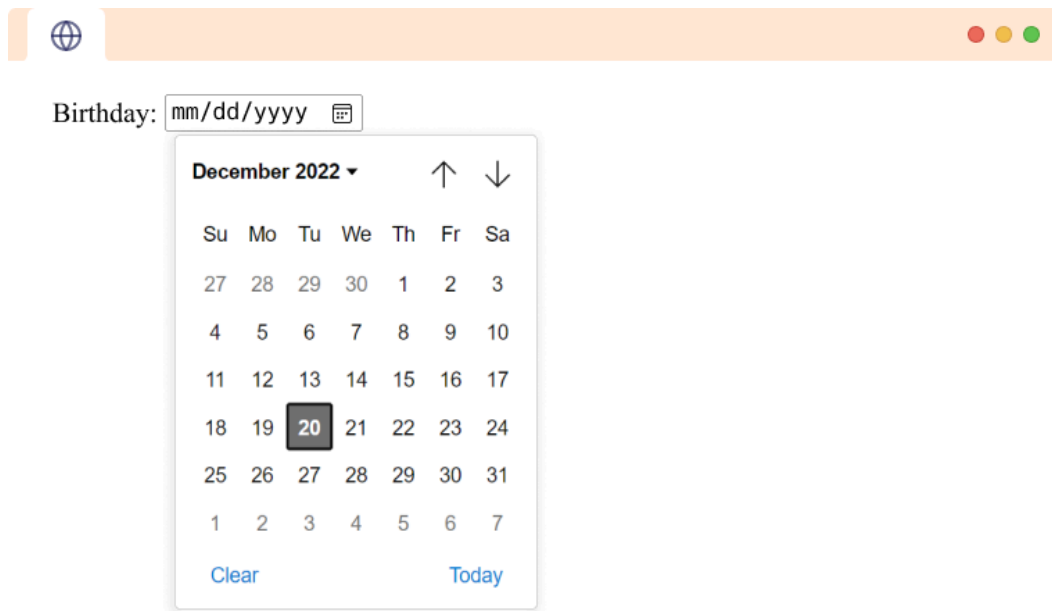
```
<label for="birthday">Birthday:</label>  
<input type="date" id="birthday">
```

Browser Output (before expanding)



A screenshot of a web browser window. The address bar is empty. Below it, the text "Birthday:" is followed by a date input field. The input field contains the placeholder text "mm/dd/yyyy" and a small calendar icon on the right. The browser window has a light orange header bar with a globe icon on the left and three colored window control buttons (red, yellow, green) on the right.

Browser Output (after expanding)



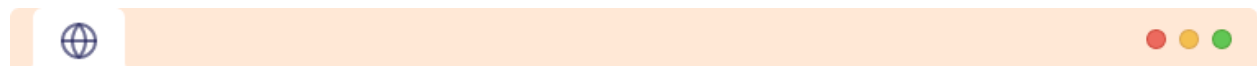
A screenshot of a web browser window, similar to the one above, but with the date input field expanded into a calendar picker. The input field still shows "mm/dd/yyyy" and the calendar icon. The calendar is displayed below the input field. It has a title "December 2022" with a dropdown arrow. Below the title are two arrows, one pointing up and one pointing down. The calendar grid shows the days of the week (Su, Mo, Tu, We, Th, Fr, Sa) and the dates. The date "20" is highlighted with a dark grey background. At the bottom of the calendar, there are two links: "Clear" and "Today".

6. Input Type datetime-local

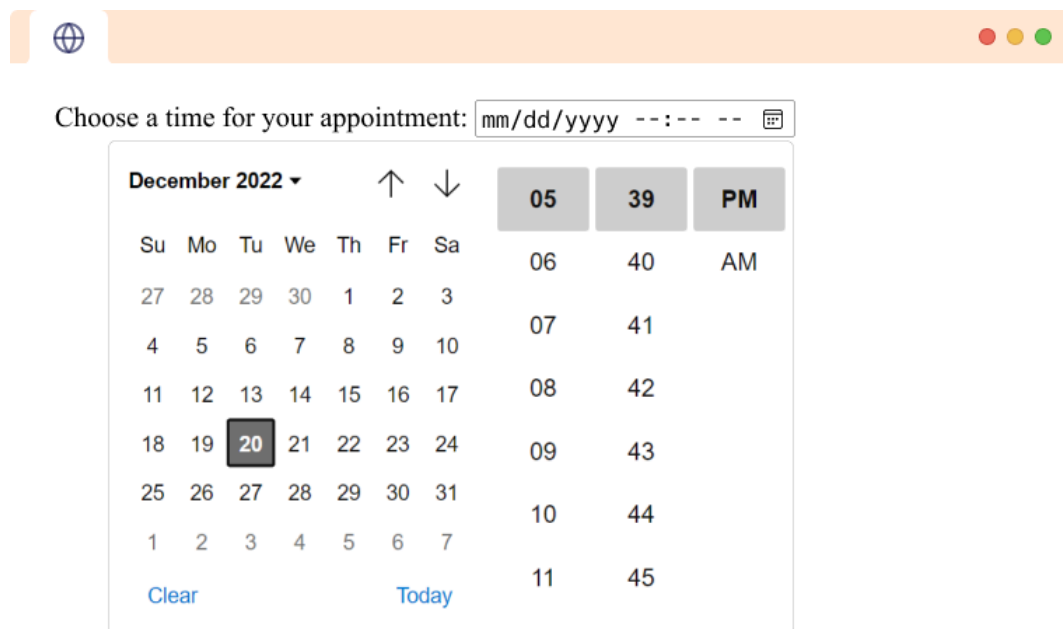
The input type `datetime-local` is used to create an input field that lets the user pick a date and time using a date-time-picker from the browser. The time selected from the input does not include information about the timezone. For example,

```
<label for="meeting-time">Choose a time for your appointment:</label>
<input type="datetime-local" id="meeting-time" >
```

Browser Output (before expanding)



Browser Output (after expanding)

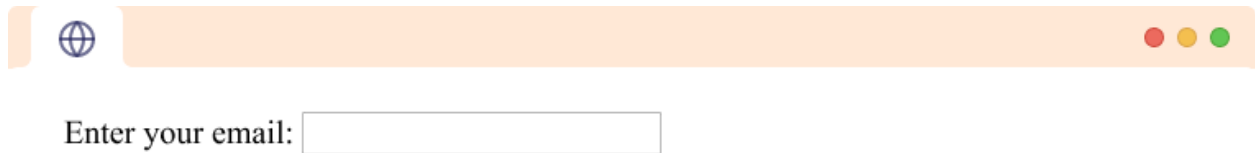


7. Input Type email

The input type `email` is used to create an input field that allows the user to input a valid email address.

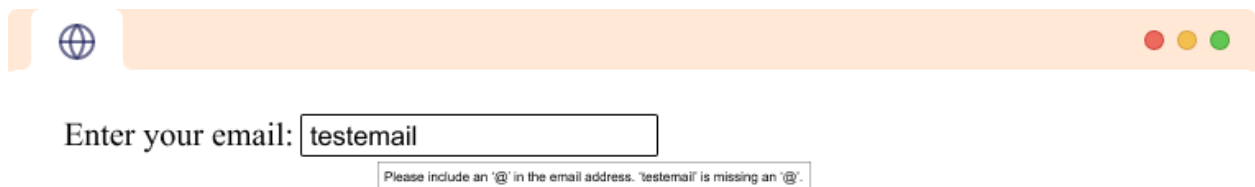
```
<label for="email">Enter your email:</label>  
<input type="email" id="email">
```

Browser Output

A browser window mockup with an orange header bar containing a globe icon on the left and three colored window control buttons (red, yellow, green) on the right. Below the header, the text "Enter your email:" is followed by an empty text input field with a light gray border.

This input field throws an error if the value provided is not a valid email. For example,

Browser Output

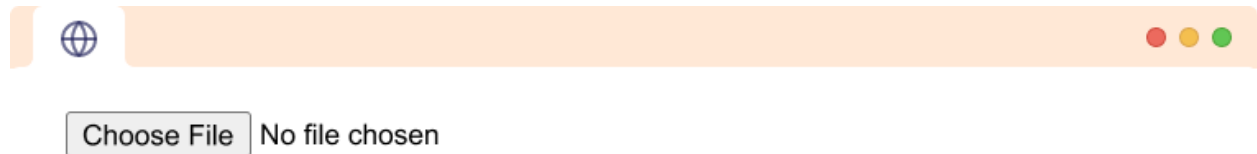
A browser window mockup with an orange header bar containing a globe icon on the left and three colored window control buttons (red, yellow, green) on the right. Below the header, the text "Enter your email:" is followed by a text input field containing the value "testemail". Below the input field, a small gray error message box is displayed with the text: "Please include an '@' in the email address. 'testemail' is missing an '@'."

8. Input Type file

The input type `file` is used to create an input field that lets the user upload a file or multiple files from their device. For example,

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

Browser Output

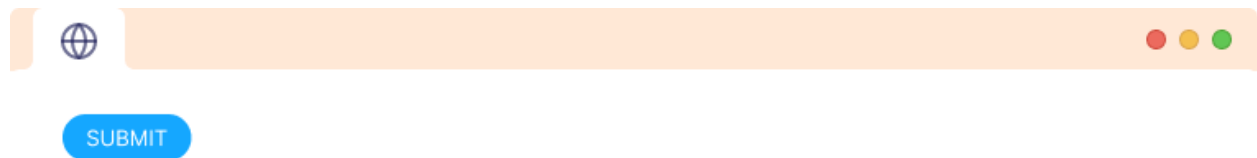


9. Input Type image

The input type `image` is used to create a button using an image.

```
<input type="image" src="/submit.png" alt="submit" >
```

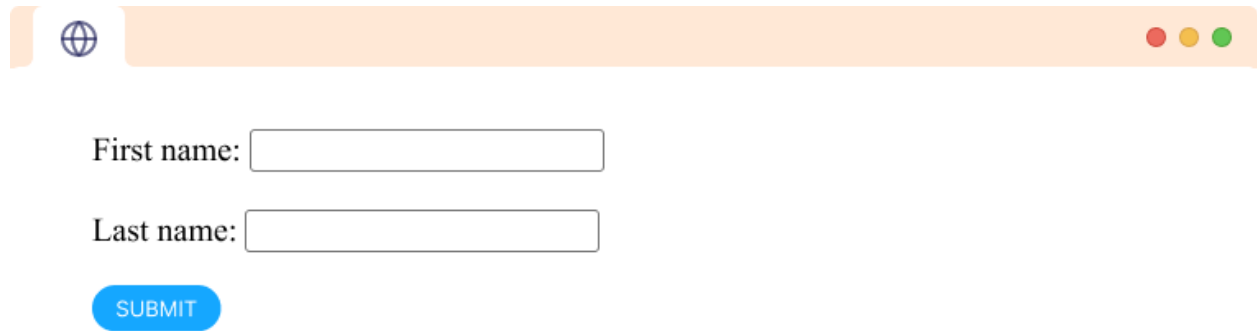
Browser Output



Let's see an example of how we can use it in a form.

```
<form>
  <label for="firstname">First name: </label>
  <input type="text" id="firstname" name="firstname"><br><br>
  <label for="lastname">Last name: </label>
  <input type="text" id="lastname" name="lastname"><br><br>
  <input type="image" src="/submit.png" alt="submit" >
</form>
```

Browser Output



First name:


Last name:

10. Input Type month

The input type `month` is used to create an input field that lets the user enter month and year using a visual interface from the browser. For example,

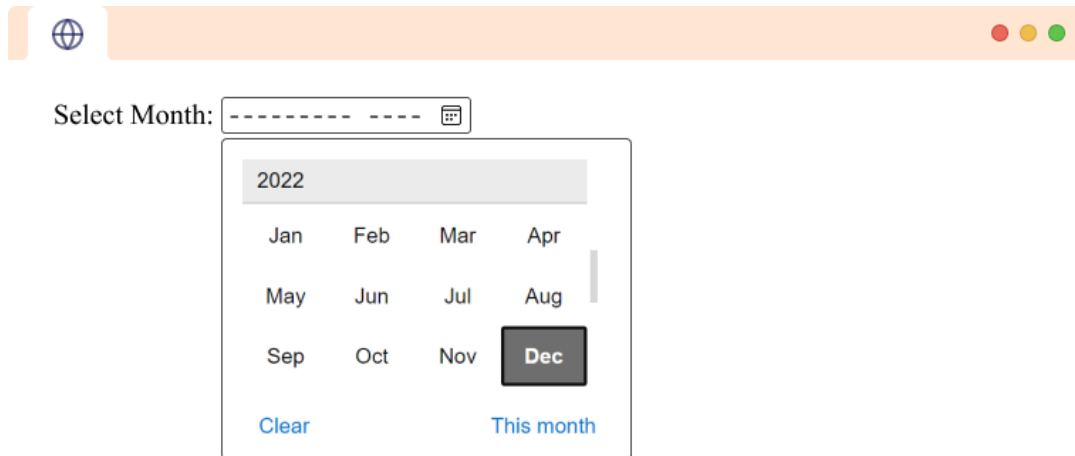
```
<label for="start">Select Month:</label>  
<input type="month" id="start" >
```

Browser Output (before expanding)



Select Month:

Browser Output (after expanding)



Select Month:

2022

Jan	Feb	Mar	Apr
May	Jun	Jul	Aug
Sep	Oct	Nov	Dec

Clear

This month

11. Input Type password

The input type `password` is used to create an input field that lets the user enter information securely. For example,

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

Browser Output



Password:

The browser displays all the characters the user types using an asterisk (*).

12. Input Type radio

The input type `radio` is used to define a radio button. Radio buttons are defined in a group. Radio buttons let users pick one option out of a group of options.

```
<form>
  <input type="radio" id="cat" name="animal" value="cat">
  <label for="cat">cat</label>
  <input type="radio" id="dog" name="animal" value="dog">
  <label for="dog">dog</label>
</form>
```

Browser Output



From the above example, we can see that all the radio buttons share the same `name` attribute. It allows the user to select exactly one option from the group of radio buttons.

When submitting the form data, the key for the input will be the `name` attribute, and the value will be the radio button selected.

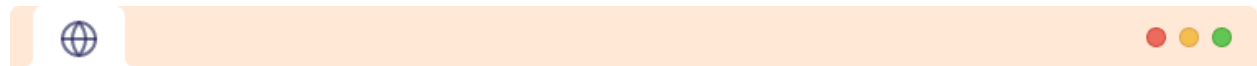
Note: The name attribute is used as the key for the data when submitting the form.

13. Input Type range

The input type `range` is used to create a range picker from which the user can select the value. User can select a value from the range given. It has a default range from **0** to **100**. For example,

```
<label for="range">Select value: </label>
<input type="range" id="range" value="90">
```

Browser Output



Select value:

14. Input Type reset

The input type `reset` defines the button which clears all the form values to their default value. For example,

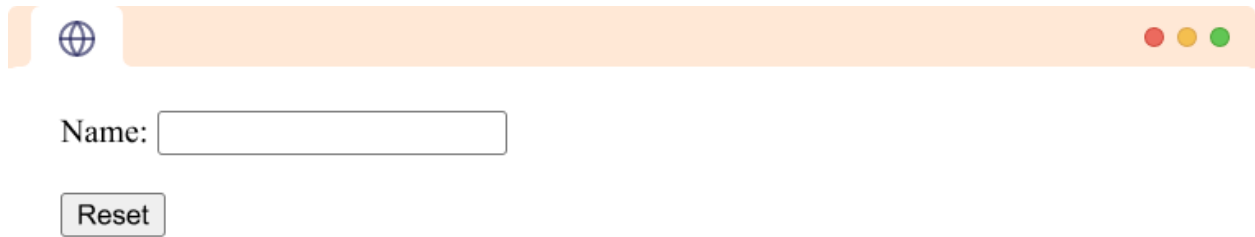
```
<form>
  <input type="radio" id="cat" name="animal" value="cat">
  <label for="cat">cat</label>
  <input type="radio" id="dog" name="animal" value="dog">
  <label for="dog">dog</label>
</form>
```

Browser Output



Name:

Browser Output (after reset)



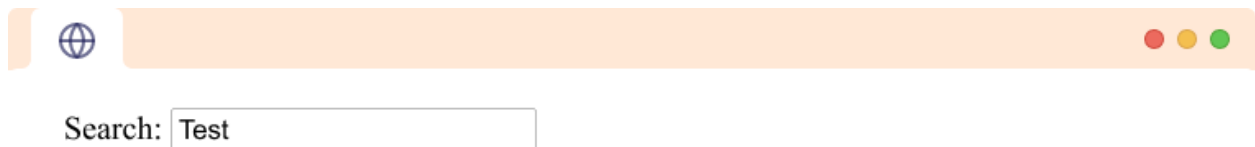
A browser window mockup with an orange header bar containing a globe icon on the left and three colored window control buttons (red, yellow, green) on the right. Below the header, the text "Name:" is followed by an empty text input field. Below the input field is a button labeled "Reset".

15. Input Type search

The input type `search` allows user to enter their search queries in the text fields. It is similar to input type text. For example,

```
<label for="search">Search: </label><input type="search" id="search" >
```

Browser Output



A browser window mockup with an orange header bar containing a globe icon on the left and three colored window control buttons (red, yellow, green) on the right. Below the header, the text "Search:" is followed by a text input field containing the word "Test".

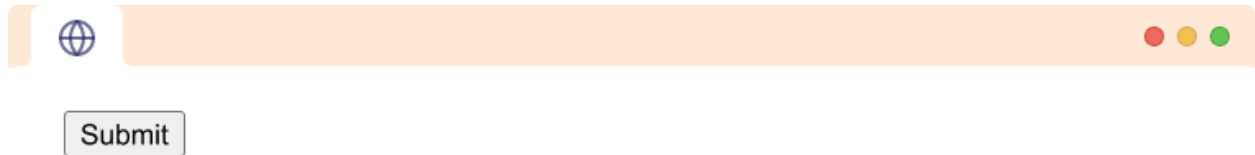
Note: The search input does not work as a search unless we use some JavaScript to do the search calculation.

16. Input Type submit

The input type `submit` is used when the user submits the form to the server. It is rendered as a button. For example,

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

Browser Output



Here, The text provided in the *value* attribute of the input is shown in the button.

17. Input Type time

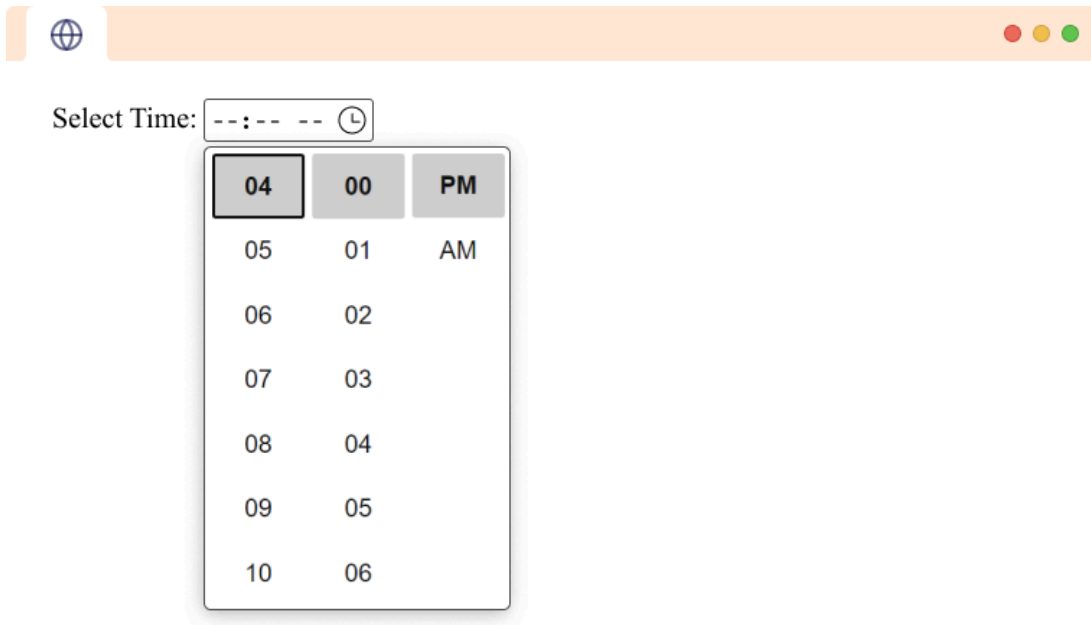
The input type `time` attribute creates an input field that accepts time value. It allows users to add time in hours, minutes, and seconds easily. For example,

```
<label for="time">Select Time:</label>
<input type="time" id="time">
```

Browser Output (before expanding)



Browser Output (after expanding)



Select Time: --:-- -- 🕒

04	00	PM
05	01	AM
06	02	
07	03	
08	04	
09	05	
10	06	


18. Input Type url

The input type `url` is used to let the user enter and edit a URL. For example,

```
<label for="url">URL:</label>
```

```
<input type="url" id="url" placeholder="https://example.com" pattern="https://.*">
```

Browser Output



URL:

Here, `placeholder` is a hint that specifies the expected value of an input field, and the `pattern` defines what type of value is accepted. The above pattern means that only text beginning with `https://` will be valid.

HTML Form Validations

Form validation is the process of checking the data entered by users in form fields against specific criteria before submitting it to the server. This validation process ensures that the input data is both complete and formatted correctly according to predefined rules.

The primary goal of form validation is to ensure that invalid data does not get processed, which can lead to erroneous operations, security risks, and a degraded user experience.

The Role of HTML5 in Form Validation

HTML5 has significantly enhanced the capabilities of form validation with the introduction of new input types and attributes that support different kinds of data input, such as dates, email addresses, numbers, and more.

These built-in features allow developers to implement form validation directly in the HTML, reducing the need for extensive JavaScript code and improving the performance and user experience of web forms.

Importance of Form Validation

1. Data Quality and Accuracy

- **Prevents incorrect data:** It ensures users enter data in the correct format (e.g., email addresses with "@" and ".", phone numbers with digits only).
- **Ensures completeness:** It can require users to fill in mandatory fields, preventing incomplete submissions.
- **Reduces errors:** By catching mistakes early, it minimizes the chance of storing or processing faulty information.

2. Improved User Experience:

- **Provides immediate feedback:** Users know right away if they've made a mistake, allowing them to correct it without waiting for server processing.
- **Reduces frustration:** Clear error messages guide users on how to fix issues, making the form easier to use.
- **Saves time:** Users don't have to resubmit forms due to errors caught on the server-side.

3. Enhanced Security:

- **Protects against malicious input:** It can prevent users from entering harmful code (like JavaScript) that could compromise the website or its users.

- **Reduces spam:** It can filter out bot-generated submissions and prevent spam from reaching your systems.

4. Server Efficiency:

- **Reduces server load:** By validating on the client-side (in the user's browser), it minimizes unnecessary requests to the server.
- **Saves bandwidth:** Less data is transmitted back and forth, improving website performance.

Key HTML5 features that facilitate form validation include:

- **Input Types:** HTML5 introduces specific input types that automatically validate user input to match the expected data format. For example, using **input type="email"** ensures that the user enters a valid email format.

```
<input type="email" name="user-email" placeholder="Enter your email">
```

required

The **required** attribute is a straightforward way to enforce that a form element must be filled out before the form can be submitted. It works with input types like text, email, select, and more.

```
<input type="text" name="fullname" required placeholder="Enter your full name">
```

type

The **type** attribute specifies the kind of data the input field should accept, providing automatic validation for several common data types:

- **email:** Validates that the entry is a properly formatted email address.
- **number:** Ensures the input is a number and allows for further specification with attributes like **min**, **max**, and **step**.
- **date:** Ensures a valid date format, facilitating date pickers in supporting browsers.

```
<input type="email" name="user-email" required placeholder="Enter your email">
```

```
<input type="number" name="age" min="18" placeholder="Enter your age">
```

```
<input type="date" name="birthdate" required>
```

pattern

The **pattern** attribute uses regular expressions to define a custom format that the input data must match. It's particularly useful for formatting entries like phone numbers, zip codes, or custom identifiers.

```
<input type="text" name="zipcode" pattern="\d{5}(-\d{4})?" title="Enter a 5-digit zip code or a 9-digit format with a hyphen">
```

min and max

The **min** and **max** attributes specify the minimum and maximum values for input types such as **number** and **date**, which is essential for setting boundaries on acceptable input.

```
<input type="number" name="donation" min="1" max="1000" step="1" required  
placeholder="Enter an amount between $1 and $1000">
```

maxlength and minlength

These attributes define the maximum and minimum lengths of characters that the input can accept. They are critical for inputs like passwords and personal messages, where length requirements might be a part of security protocols or data handling standards.

```
<input type="password" name="password" minlength="8" maxlength="20" required  
placeholder="Enter a password (8-20 characters)">
```

HTML <label> tag

The HTML label tag is used to create a caption for a form element. The text inside the <label> tag is shown to the user.

```
<label for="firstname">First Name</label>  
<input type="text" name="firstname" id="firstname">
```

Browser Output



The **for** attribute is used to associate a label with the form element. The value for the **for** attribute is set as the **id** of the form element which is **firstname** in the above example.

HTML Label is mainly used for accessibility as screen-readers read out the label associated with the field and it also improves user experience as clicking on the label also focuses on the input field.

This is also greatly helpful in small screens as it makes it easier to perform actions like focusing on input, selecting a checkbox, selecting a radio box, etc.

HTML <select> tags

The HTML <select> tag is used to create a menu of options. Each of the options is represented by the <option> tag. For example,

```
<label for="pets">Pets:</label>
<select id="pets">
  <option value="dog">Dog</option>
  <option value="cat">Cat</option>
</select>
```

Browser Output



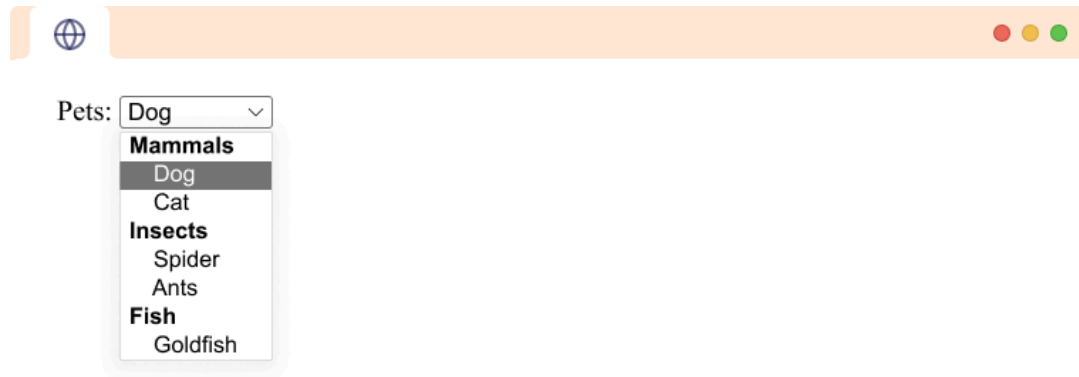
Browser Output



Additionally, we can also group option elements inside the `<optgroup>` tag to create a group of options. For example,

```
<label for="pets">Pets:</label>
<select id="pets">
  <optgroup label="Mammals">
    <option value="dog">Dog</option>
    <option value="cat">Cat</option>
  </optgroup>
  <optgroup label="Insects">
    <option value="spider">Spider</option>
    <option value="ants">Ants</option>
  </optgroup>
  <optgroup label="Fish">
    <option value="goldfish">Goldfish</option>
  </optgroup>
</select>
```

Browser Output



HTML `<textarea>` tag

The HTML `<textarea>` tag is used to define a customizable multiline text input field. For example,

```
<textarea rows="10" cols="30"> Type something...</textarea>
```

Browser Output



Here, the *rows* and *cols* attributes represent the rows and columns of the text field.

HTML `<button>` tag

The HTML `<button>` element is an interactive element that is activated by a user with a mouse, keyboard, finger, voice command, or other assistive technology.

It performs a programmable action, such as submitting a form or opening a dialog when clicked. For example,

```
<button type="button">Click me</button>
```

Browser Output



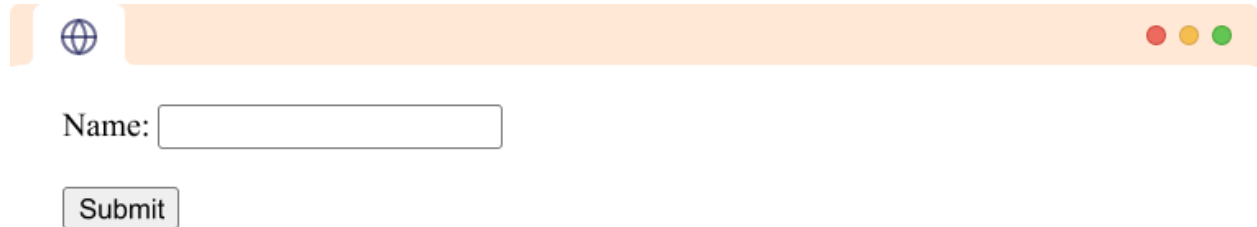
The *type* attribute determines the action performed by clicking the button. It has **3** possible values:

- **submit**

If the value of *type* is `submit`, the button click action submits the form. For example,

```
<form>
  <label for="name">Name:</label>
  <input type="text" name="name"><br><br>
  <button type="submit">Submit</button>
</form>
```

Browser Output



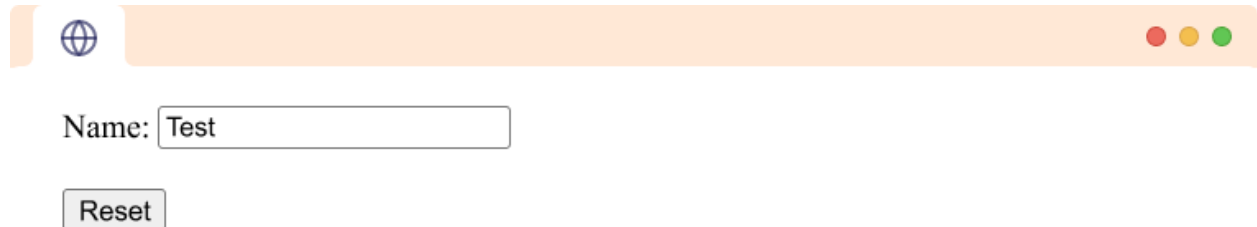
Name:

- **reset**

If the value of *type* is **reset**, the button click action resets the value of all form elements to their initial value. For example,

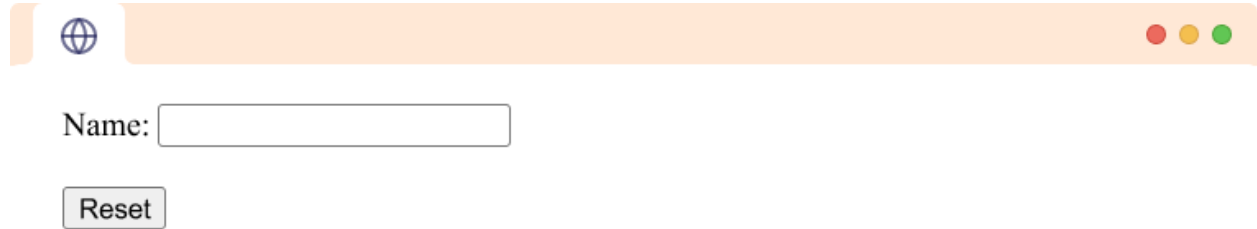
```
<form>
  <label for="name">Name:</label>
  <input type="text" name="name"><br><br>
  <button type="reset">Reset</button>
</form>
```

Browser Output (before reset)



Name:

Browser Output (after reset)



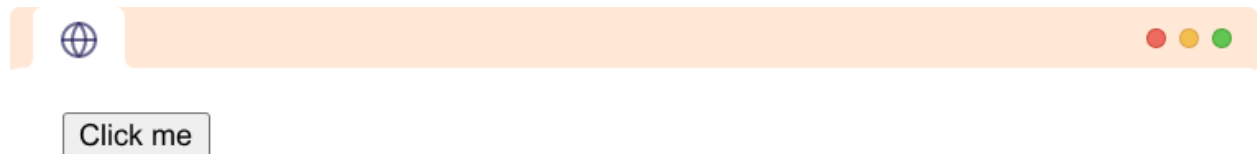
A browser window mockup with an orange header bar containing a globe icon on the left and three colored dots (red, yellow, green) on the right. Below the header, the text "Name:" is followed by an empty text input field. Below the input field is a button labeled "Reset".

- **button**

If the value of *type* is **button**, the button click action does not have a default function. Generally, javascript is used to add functionality to such buttons. For example,

```
<button type="button">Click me</button>
```

Browser Output




A browser window mockup with an orange header bar containing a globe icon on the left and three colored dots (red, yellow, green) on the right. Below the header, there is a button labeled "Click me".

HTML **<fieldset>** tag

The HTML **<fieldset>** tag is used to group similar form elements. It is a presentational tag and does not affect the data in the form. For example,

```
<form >
  <fieldset>
    <label for="firstname">First name:</label><br>
    <input type="text" id="firstname" name="fname"><br>
    <label for="lastname">Last name:</label><br>
    <input type="text" id="lastname" name="lname"><br>
  </fieldset>
</form>
```

Browser Output



Here, the border is from the `<fieldset>` element.

HTML `<legend>` tag

The HTML `<legend>` tag is another presentational tag used to give a caption to a `<fieldset>` element. It acts similarly to an HTML `<label>` tag. For example,

```
<form>
  <fieldset>
    <legend>Name</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname"><br>
  </fieldset>
</form>
```

Browser Output



A screenshot of a web browser window. The browser has a light orange header bar with a globe icon on the left and three colored window control buttons (red, yellow, green) on the right. The main content area is white and contains a form with a title 'Name' in bold. Below the title are two input fields. The first is labeled 'First name:' and the second is labeled 'Last name:'. Both fields are empty and have a light gray border.

HTML `<datalist>` tag

The `<datalist>` tag defines a list of pre-defined options for an `<input>` element. It is used to provide autocomplete options to the form elements that show up as recommended options when the user fills in the form. For example,

```
<label for="country-choice">Choose a country:</label>
<input list="country-options" id="country-choice" name="country-choice">
<datalist id="country-options">
  <option value="Australia">
  <option value="Austria">
  <option value="America">
  <option value="Nepal">
</datalist>
```

Browser Output



A screenshot of a web browser window showing the browser output of the HTML code. The browser has a light orange header bar with a globe icon on the left and three colored window control buttons (red, yellow, green) on the right. The main content area is white and contains a form with a label 'Choose a country:'. To the right of the label is a dropdown menu. The dropdown menu is open, showing the text 'au' in the input field and a list of suggestions: 'Australia' and 'Austria'.

Here, when the user types *au*, the browser suggests options with the letters to the user as a recommendation.

HTML `<option>` tag

The `<option>` HTML element is used to define an item contained in a `<select>`, an `<optgroup>`, or a `<datalist>` element. As such, `<option>` can represent menu items in popups and other lists of items in an HTML document.

The `value` attribute specifies the value to be sent to a server when a form is submitted.

The content between the opening `<option>` and closing `</option>` tags is what the browsers will display in a drop-down list. However, the value of the `value` attribute is what will be sent to the server when a form is submitted.

For example:

```
<label for="pet-select">Choose a pet:</label>
<select id="pet-select">
  <option value="">--Please choose an option--</option>
  <option value="dog">Dog</option>
  <option value="cat">Cat</option>
  <option value="hamster">Hamster</option>
  <option value="parrot">Parrot</option>
  <option value="spider">Spider</option>
  <option value="goldfish">Goldfish</option>
</select>
```

Browser Output

