Lesson:

HTML form element

PW SKILLS

Topics Covered

- Input Tag
- Input Attributes
- Common Input Attributes
- Validations
- Example for a form with multiple inputs

Input Tag

The <input> HTML element is one of the most important form elements which is used to accept data/input from the user.

A wide variety of types of input data and control widgets are available

A simple example,

```
Unset
<label for="email">Enter Email: </label>
<input type="email" name="email" id="email"
required>
```

Above code defines an email input for entering an email address. It also checks for a valid email address.

Here <label> tags help users to know the purpose of the form control element (input), and for attribute specifies the target control element, which allows users to click on the label to focus on the associated form control.

Input Attributes

- **type:**

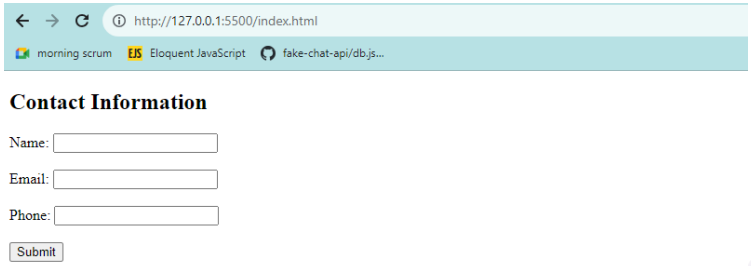
This attribute specifies the type of input control to be created. It is a required attribute and can take various values such as **text, password, email, number, checkbox, radio, submit,** and many more.

- **name:**

It's used to give a name to the information that a user enters into that field. This name is important because it helps developers and servers understand what the user has typed in. When you submit the form, The value of this attribute is sent to the server. The server uses these names to figure out which data corresponds to which input field. It's like labeling different jars in your kitchen so you can easily find what you need.

Let's look at an example to understand the significance of name attribute:

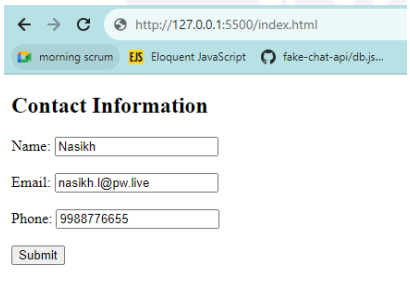
```
Unset
<h2>Contact Information</h2>
<form>
   <label for="name">Name:</label>
   <input type="text" id="name" name="name" /><br /><br />

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

   <label for="phone">Phone:</label>
   <input type="tel" id="phone" name="phone" /><br /><br />

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

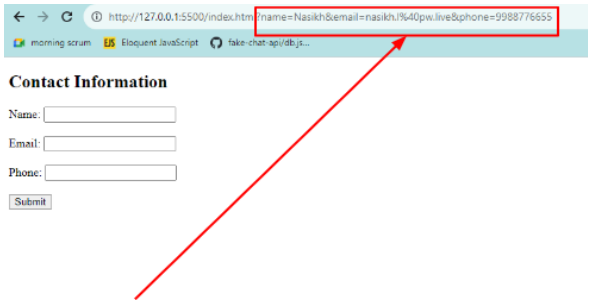


Above example shows a form which takes name, email and phone number as input.

Let's understand the importance of the “name” attribute here.

Below image shows when i filled the form and submitted.



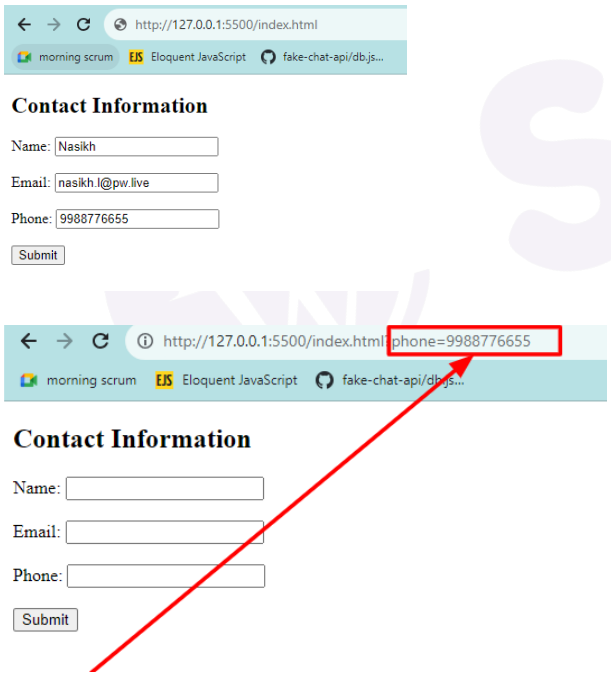


name=Nasikh&email=nasikh.l%40pw.live&phone=9988776655

Here in the URL, you can see that, once we submit the form, It sent a request to the backend/server. And the name which we gave as the attribute to that input tag has been used as the key to that input value. If we forget to add the name attribute to any of the input tags above, it will omit that value while it's sending the data to the backend/server.

Let's look into another example where we are not passing the name for some input tag and try to submit that form.

```
Unset
<h2>Contact Information</h2>
<form>
   <label for="name">Name:</label>
   <input type="text" id="name" /><br /><br />

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

   <label for="phone">Phone:</label>
   <input type="tel" id="phone" name="phone"  /><br /><br />

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



phone=9988776655

Now here you can see that the name and email value has been omitted since we didn't include the name attribute for that input element. So the name attribute has a very important role in an input element

- **value:**

This attribute specifies the default value of the input control. It is used to pre-fill the input field with a value that the user can modify.

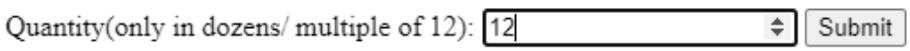
- **placeholder:**

This attribute specifies a short hint or example text that is displayed in the input field to provide guidance to the user on what should be entered.

- **step:**

The step attribute is used with the <input> element of type number or datetime-local to specify the increment value for the input field. It defines how much the value of the input should increase or decrease when the user interacts with the associated input control, such as clicking the up/down arrows or using keyboard input. Step attribute works with number, range, date, datetime-local, month, time and week <input> types

```
Unset
<form>
      <label for="quantity">Quantity(only in dozens/ multiple of 12):</label>
      <input type="number" id="quantity" name="quantity" step="12" />

      <input type="submit" />
</form>
```

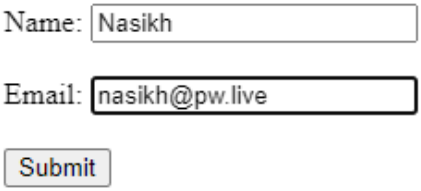


The above example shows that, suppose you are creating an online-store selling some items online and it comes in a pack of 12, so then you can use this step attribute and add the step value as 12 so that the user can enter only multiple of 12 as the input.

- **autocomplete:**

The autocomplete attribute is used in HTML forms to control whether a browser should automatically complete input fields. The browser should display options to fill in the field, based on earlier typed values. This attribute can be applied to <input> elements of type text or number, <textarea> elements, <select> elements, and <form> elements. By default the value is on. We can turn the auto-complete suggestions on a form by keeping the value as off (autocomplete="off")

```
Unset
<form autocomplete="on">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name"><br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" autocomplete="off"><br><br>
  <input type="submit">
</form>
```



First fill the form and submit the form.

Now let's try to fill the form again

Now you can see that it's giving suggestions in the name field but since we added autocomplete="off" in the email input, it won't give any suggestions there.

Boolean attributes
A boolean attribute means that the presence of that attribute on an element represents the "true" value, and the absence of the attribute represents the "false" value.

These boolean attributes are an essential part of HTML and are used to control the behavior and properties of various elements without requiring additional attribute values. Their presence or absence determines whether a particular behavior or state is enabled or disabled, providing flexibility and simplicity in HTML markup. Kindly observe in attributes below how they do not need any value.

This design principle of boolean attributes serves two primary purposes

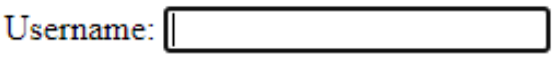
Simplicity: Boolean attributes make the HTML markup more concise and readable. They eliminate the need for explicit values, reducing clutter in the code and making it easier for developers to understand and maintain.

Clarity: Boolean attributes enhance the clarity of HTML documents by clearly indicating the state of an element or its associated behavior. Their presence provides a straightforward visual cue that a particular feature is active, while their absence communicates the opposite

- **autofocus:**

The autofocus attribute is an HTML attribute that can be applied to specify that the input field should receive focus automatically when the web page loads. This means that as soon as the page is loaded, the cursor will be placed inside the input field, allowing the user to start typing or interacting with it without needing to click on it first.

```
Unset
<form>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" autofocus>
</form>
```

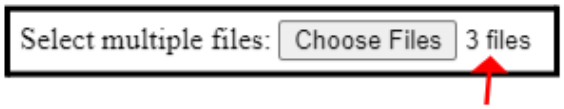


As you can see in the above example, the input element getting focused once the page is loaded and we can see that the cursor is blinking which indicates that we can start typing

- **multiple:**

The multiple attribute is used with the <input> element when creating file upload fields “<input type="file">” to allow users to select multiple files for upload in a single input field. This attribute is particularly useful when you want to enable users to upload multiple files simultaneously.
Multiple also works with <input> elements of type email to allow users to enter multiple email addresses in a single input field. When the multiple attribute is applied to an email input field, it transforms the field into a "multiple email" input, enabling users to enter a list of email addresses separated by commas or semicolons.

```
Unset
<label for="fileInput">Select multiple files:</label>
<input type="file" id="fileInput" name="fileInput" multiple>
```

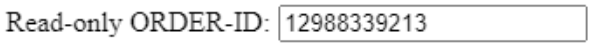


You can see in the above example that, when we added the “multiple” attribute, we were able to select multiple Files from the choose file window which pops up once we click on the choose files button. For the above example, i have chosen 3 files, that's why on its right side, its showing 3 files

- **readonly:**

The readonly attribute is used with form elements in HTML to make an input field or textarea non-editable by the user. When you apply the readonly attribute to an input element or textarea, it prevents the user from modifying the content of that field. However, the user can still view the content, highlight it, and copy the text from it and interact with other elements on the page.

```
Unset
 <label for="readonlyInput">Read-only ORDER-ID:</label>
   <input type="text" id="readonlyInput" name="readonlyInput"
value="12988339213" readonly />
```

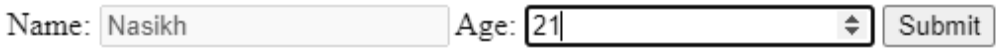


In the above example, the value in the input field cannot be edited by the user

- **disabled:**

The disabled attribute in HTML used to disable user interaction with an HTML element, such as an <input> field, a <button>, or a <select> element, etc... . When the disabled attribute is applied to an element, it prevents the user from interacting with that element. If you add the disabled attribute to an <input> field within an HTML form, the field will not be included in the form submission. In other words, the data entered into a disabled input field will not be sent to the backend/server when the form is submitted.

```
Unset
<form>
  <label for="name">Name:</label>
  <input type="text" id="name" name="user_name" value="Nasikh" disabled />
  <label for="age">Age:</label>
  <input type="number" id="age" name="age" />

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



In this example, the "Name" input field is disabled. If the user tries to submit the form, the value "Nasikh" from the disabled input field will not be included in the data sent to the server.

- **novalidate:**

The novalidate attribute is used in HTML to disable the built-in form validation provided by web browsers. When this attribute is added to a <form> element, it instructs the browser not to perform its default client-side validation checks when the form is submitted.

For Example, if you have an input field for email and you have added the type email. If you type some random text on that input field and try to submit, generally what happens is that it will throw us a warning and won't submit that form to the backend/server. But when you add the novalidate attribute on to the form, it won't validate anything, rather it just sends/submit the data to the backend.

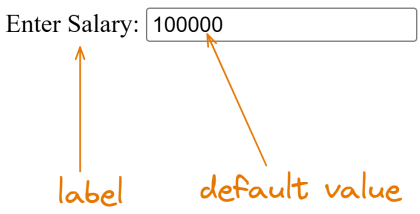
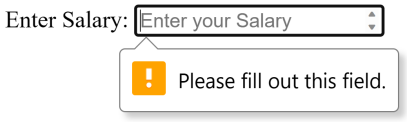
```
Unset
<form method="post" novalidate >
   <label for="user_email">Email:</label>
   <input type="email" id="user_email" name="user_email" />

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

In the above example, the validation we kept by adding the type email will be ignored when we submit the form because we have added novalidate attribute on the form. We will be understanding different types of attributes and some more validations on the form.

Example of an input attribute

```
Unset
<form>
      <label for="email">Enter Email: </label>


      <input
        type="number"
        name="salary"
        placeholder="Enter your Salary"
        value="100000"
        required
      />
</form>
```

Browser Output:-

Initially Input will look like below	When the input is empty, we can see placeholder value.	When a user tries to submit, without filing a value.
		

Common Input Types

1. Text

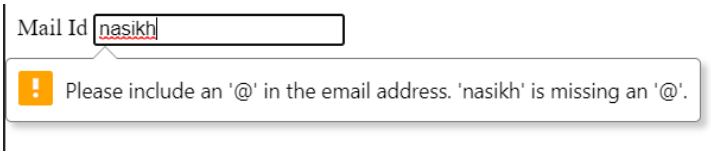
It accepts a single line string. Let's define a text input,

```
Unset
<label for="mailid">Mail Id</label>
<input type="text" placeholder="Text Input"
id="mailid" name="mailid" required/>
```

Text Input Write Here

2. Email
An input element with type="email" is used to create an email input field in HTML

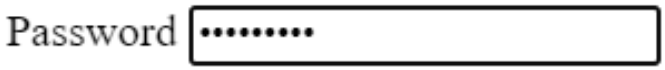
```
Unset
<label for="mailid">Mail Id</label>
<input type="email" id="mailid" name="mailid"/>
```



When the form is submitted, the value of the email input field will be included in the form data with the key specified by the **name** attribute. If the **email address entered in the input field is not a valid** email address, the **browser will throw a warning** message to the user before allowing the form to be submitted.

3. Password
It allows users to enter passwords secretly. Let's define password input,

```
Unset
<label for="pass">Password</label>
<input type="password" id="pass" name="pass"/>
```



4. Number
Only numbers allowed. Let's define number input,

```
Unset
<label for="salary">Salary</label>
<input type="number" id="salary"
name="salary"/>
```

5. Checkbox

Checkboxes are a type of input field that **allows the user** to **select one or more options** from a list of predefined options. The **type** attribute of the **<input>** tag should be set to **"checkbox"** to create a checkbox. The **name** attribute is used to group related checkboxes together, and the **value** attribute specifies the value of the checkbox when it is selected. When the user selects a checkbox, the value associated with that checkbox is submitted with the form data. If you want to **pre-select a checkbox by default,** you can add the **checked** attribute to the <input> tag.

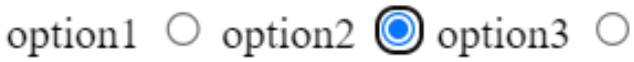
```
Unset
<label for="male">Male</label>
<input type="checkbox" id="male" name="male"/>

<label for="female">Female</label>
<input type="checkbox" id="female" name="female"/>

<label for="other">other</label>
<input type="checkbox" id="other" name="other"/>
```

Male ☐ Female ☐ other ☐

6. Radio

Radio buttons are a type of input field that allows the user to **select one option** from a list of predefined options.
The **“name”** attribute in the radio button **(<input type="radio">)** tag is **used to group related radio buttons together** so that **only one radio button can be selected** at a time **within a given group**. By giving all the radio buttons in a group the same name attribute, you ensure that only one value is submitted for that group of radio buttons. This is important when you have a list of options where only one option can be selected, such as in a multiple-choice question or a preference selection.

```
Unset
<label for="option1">option1</label>
<input type="radio" id="option1"
name="selected" value="option1" />

<label for="option2">option2</label>
<input type="radio" id="option2"
name="selected" value="option2"/>

<label for="option3">option3</label>
<input type="radio" id="option3"
name="selected" value="option3"/>
```



Note: Don't forget to give a value in the case of checkboxes and radio buttons.

7. Button

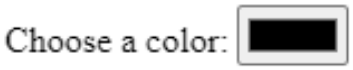
The "button" type is used to create clickable buttons within a form or on a web page. This works similarly to the <Button> element.

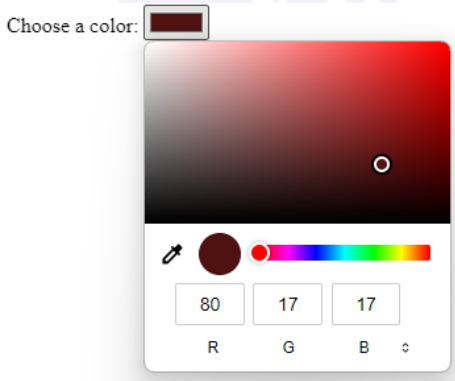
```
Unset
<input type="button" value="Click me">
```

8. Color

The "color" type is used to create an input field that allows users to select a color. When this input field is rendered in a web browser, it typically displays a color picker dialog that lets users choose a color visually.

```
Unset
<label for="colorPicker">Choose a color:</label>
<input type="color" id="colorPicker" name="color" />
```



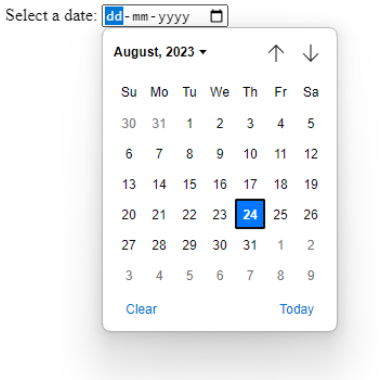


When users interact with the color picker input, they can click on it to open the color picker dialog, where they can select a color using various methods.

9. Date

The "date" input type is used to create an input field that allows users to select a date. When you use this input type, it typically displays a date picker or calendar control. This allows users to easily choose a date without having to manually enter it.

```
Unset
<label for="datePicker">Select a date:</label>
<input type="date" id="datePicker" name="selectedDate">
```



Additionally, you can use attributes like min and max to specify a range of acceptable dates and the value attribute to set an initial date value.

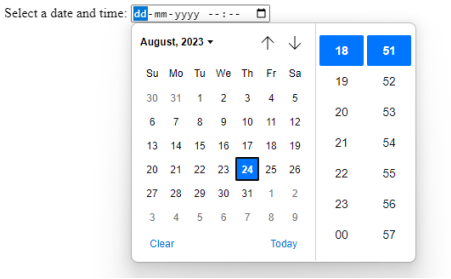
```
Unset
<input type="date" id="datePicker" name="selectedDate"
min="2023-01-01" max="2023-12-31" value="2023-08-24">
```

The date picker would allow users to choose a date between January 1, 2023, and December 31, 2023, with an initial value set to August 24, 2023. The Standard format to set date is YYYY-MM-DD

10. Datetime-local

The “datetime-local” input type in HTML is used to create an input field that allows users to select both a date and a time, including the year, month, day, hour, and minute. It's particularly useful when you need to collect date and time information together, such as scheduling events or appointments.

```
Unset
<label for="dateTimePicker">Select a date and time:</label>
<input type="datetime-local" id="dateTimePicker"
name="selectedDateTime">
```

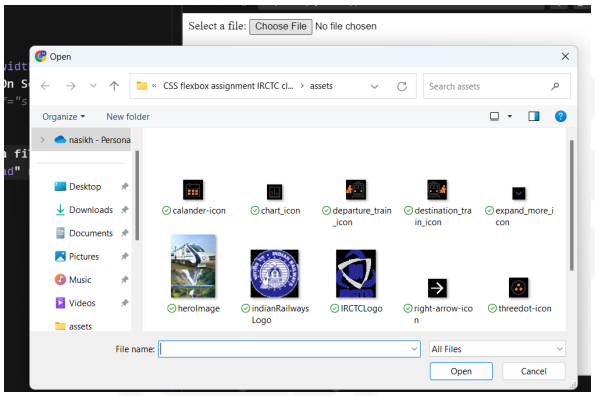


When users interact with the datetime-local input, they can click on it to open a calendar control for selecting the date and a time picker for choosing the time.

11. File

The "file" type is used to create a file input field in HTML forms. This type of input allows users to select and upload files from their local devices to a web server. It is commonly used for tasks such as uploading images, documents, or other files to a website.

```
Unset
<label for="fileUpload">Select a file:</label>
<input type="file" id="fileUpload" name="uploadedFile" />
```



When users interact with the file input, they can click on it to open a file selection dialog provided by their operating system. They can then browse their local files and select the file(s) they want to upload.

12. Hidden

The "hidden" type is used to create a hidden input field in an HTML form. Unlike visible input fields like text boxes or checkboxes, hidden input fields are not visible to users when they view the form on a web page. Instead, they are used to store data like, user ID, Session Id or many other details which the user may not be aware of or you don't want a user to manually type it but this data you still want to send to the backend/ server when the form is submitted.

```
Unset
<input type="hidden" name="hiddenField" value="This is a hidden
value" />
```

Hidden input fields are a useful tool for working with forms and managing data on the server-side without requiring user interaction or displaying the data to users.

13. Image

The "image" type is used to create an image-based submit button in HTML forms. It allows you to use an image as the button instead of traditional text or a styled button. When the user clicks on the image, it functions as a form submit button and sends the form data to the server. The src attribute is used to set the submit button image and alt attribute will act as the label if it's unable to load that image

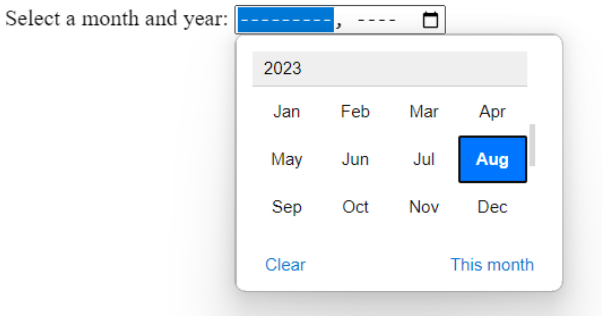
```
Unset
<input
type="image"src="https://t4.ftcdn.net/jpg/00/28/27/95/360_F
_28279558_SqNXoZWsQLfWYVxyKe9hVzZ49dJtKLsc.jpg"
      width="100"
      height="70"
 alt="Submit">
```

The image-based submit button is a visually appealing way to create custom submit buttons in your forms, especially when you want to use images or icons for better user experience.

14. Month

The "month" type is used to create an input field that allows users to select a specific month and year. It provides a dropdown or spinner interface that lets users choose a month and year without having to manually enter the information.

```
Unset
<label for="monthPicker">Select a month and year:</label>
<input type="month" id="monthPicker" name="selectedMonthYear" />
```



When users interact with the month input, they can click on it to open a dropdown control that allows them to select a month and year. You can also set the min and max attributes & values similar to that of we did in the date type

15. Range

The "range" type is used to create an input field that allows users to select a value from a specified range, typically represented as a slider control. It's often used in scenarios where you want users to choose a value within a predefined numeric range, such as setting a volume level, selecting a price range, or adjusting a numerical setting.

We can define the minimum and maximum and default initial values of the range input element. There is also a step attribute which specifies the increment by which the value can change.

```
Unset
  <label for="volumeControl">Volume Control:</label>
  <input type="range" id="volumeControl" name="volume" min="0" max="100" step="1"
value="50">
```

When users interact with the range input, they can drag the slider handle to select a value within the specified range.

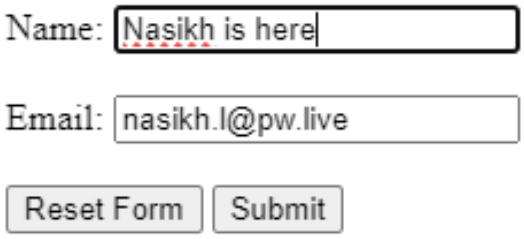
16. Reset

The "reset" is used to create a reset button in HTML forms. When a user clicks this button, it resets all the form controls within the same <form> element back to their initial values or the values they had when the page loaded.

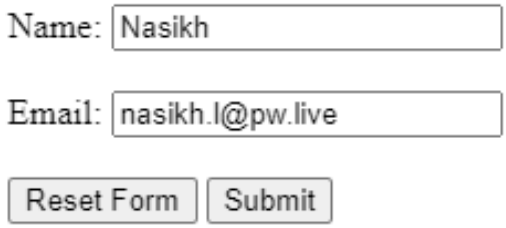
```
Unset
<input type="reset" value="Reset Form">
```

When a user clicks the reset button, all the form controls within the same <form> element will be reset to their initial values or the values they had when the page loaded. This means that any user-entered data will be cleared, and checkboxes and radio buttons will return to their default states.

```
Unset
<form>
   <label for="name">Name:</label>
   <input type="text" id="name" name="name" value="Nasikh" /><br
/><br />

   <label for="email">Email:</label>
   <input
    type="email"
    id="email"
    name="email"
    value="nasikh.l@pw.live"
   /><br /><br />

   <input type="reset" value="Reset Form" />
   <input type="submit" value="Submit" />
</form>
```



Before clicking on reset form button



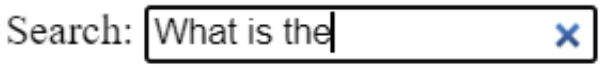
After Clicking on the reset button

In the above code example, you can see that once I click on the reset form button, the form value has been reset to the default value.

17. Search

The "search" type is used to create a search input field in HTML. It is specifically designed for inputting search queries into a web page's search feature. When you use type="search", the browser often provides a search-specific input field with features like a clear button to clear the search query.

```
Unset
<label for="searchBox">Search:</label>
<input type="search" id="searchBox" name="searchQuery"
placeholder="Search...">
```



You can see that a close icon appeared when we started typing in the search input field

18. Submit

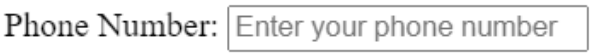
The "submit" type is used to create a submit button in HTML forms. When a user clicks this button, it triggers the submission of the form's data to the server, typically to a URL specified in the form's action attribute.

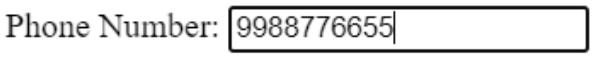
```
Unset
<input type="submit" value="Submit">
```

19. Tel

The "tel" type is used to create an input field for collecting telephone numbers. It is designed for user input of telephone numbers and can provide features like numeric keyboards on mobile devices to make it easier for users to enter phone numbers

```
Unset
<label for="phoneNumber">Phone Number:</label>
  <input
   type="tel"
   id="phoneNumber"
   name="userPhoneNumber"
   placeholder="Enter your phone number"
  />
```



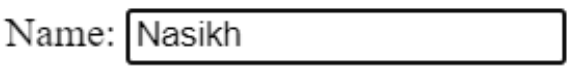


It does not restrict the types of characters that can be input, but browsers will prevent submission if the input value does not match your pattern.

20. Text

The "text" type is one of the most commonly used input types in HTML forms. It is used to create a single-line text input field where users can enter textual information. This input type is versatile and can be used for various purposes, such as collecting names, addresses, email addresses, and more.

```
Unset
<label for="name">Name:</label>
<input type="text" id="name" name="userName" placeholder="Enter your
name">
```

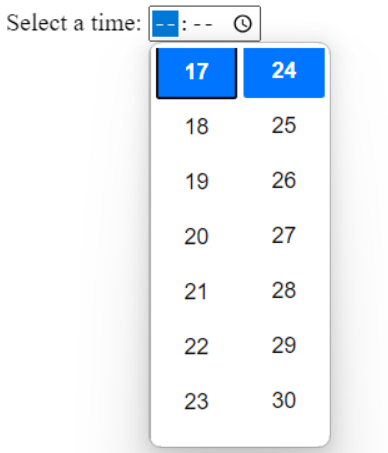


When users interact with the text input, they can type in textual information

21. Time

The "time" type is used to create an input field that allows users to select a specific time of day. It typically displays a time picker interface that lets users choose hours and minutes.

```
Unset
<label for="timePicker">Select a time:</label>
<input type="time" id="timePicker" name="selectedTime">
```



When users interact with the time input, they can click on it to open a time picker dialog that allows them to select hours and minutes

22. Url

The "url" type is used to create an input field that is specifically designed for entering URLs (Uniform Resource Locators) or web addresses. This input type can help ensure that users enter valid URLs by providing validation and user interface enhancements.

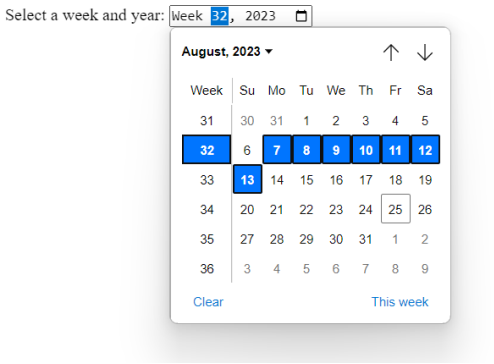
```
Unset
<label for="website">Website URL:</label>


<input type="url" id="website" name="websiteUrl" placeholder="Enter
a URL">
```

When users interact with the URL input, they can enter a web address. Many web browsers provide additional user interface features, such as validation checks and clickable links, to help users enter valid URLs.

23. Week

The "week" type is used to create an input field that allows users to select a specific week and year. It provides a dropdown interface that lets users choose a week and year combination. This input type is particularly useful for tasks that involve selecting a week of the year, such as scheduling or date-related applications.

```
Unset
<label for="weekPicker">Select a week and year:</label>
<input type="week" id="weekPicker" name="selectedWeek">
```



When users interact with the week input, they can click on it to open a dropdown control that allows them to select a week and year.

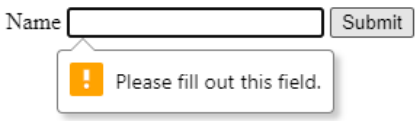
Validations

When creating forms, it's crucial to ensure the data entered by users is accurate and matches the expected format. Adding the appropriate attributes to the <input> tag helps implement various types of validation. This not only enhances the user experience but also ensures the reliability of the collected data.

1. required

The required attribute is used to indicate that an input field must be filled out before submitting the form. It ensures that users cannot leave the field empty. If the field is empty when you submit the form, it will show a warning message.

```
Unset
<form>
    <label for="name">Name</label>
    <input type="text" id="name" name="name" required />
    <input type="submit" value="Submit" />
</form>
```



When I tried to submit the code without typing anything in the input field, it threw me a warning saying that "Please fill out this field" as you can see in the above screenshot

2. type

The type attribute specifies the type of data expected in the input field. It plays a significant role in validation. Use appropriate input types from the "common input types" topic which we discussed previously.

3. accept

The "accept" attribute is used with the <input> element of type "file" to specify the types of files that the file input control can accept. This attribute helps in filtering the files that users can select when they choose to upload a file.

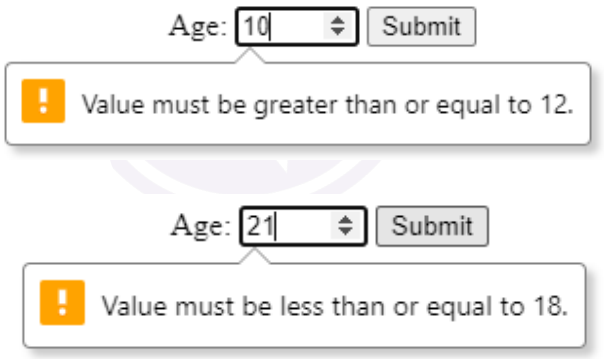
You can specify one or more file extensions to restrict the accepted file types.

accept=".pdf": Accepts only PDF files.
accept=".jpg, .jpeg, .png": Accepts JPEG and PNG image files
accept=".doc,.docx": Accepts Word documents.

4. min and max

For number inputs, you can set min and max attributes to define acceptable value ranges. Min defines the minimum value which can be accepted and max defines the maximum value which can be accepted.

```
Unset
<label for="age">Age:</label>
<input type="number" id="age" name="age" min="12" max="18"
/>
```

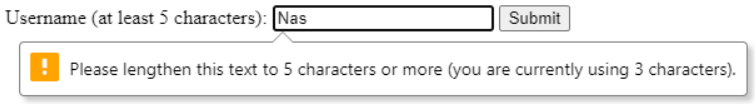


In the above example, you can see that now the input takes only values between 12 and 18 or else, it throws an as shown above and won't submit the form because we set the min value as 12 and max value as 18.

5. minlength and maxlength

The minlength and maxlength attributes are used to set the minimum and maximum length of text that can be entered into an <input> element of type "text" or "password" or a <textarea> element.

```
Unset
<label for="username">Username (at least 5
characters):</label>
<input type="text" id="username" name="username"
minlength="5">
```



In the above example, you can see that now the input will only allow us to submit the form when the minimum length of the characters is equal to or more than 5

6. size

The size attribute is used to specify the visible width of an input field and the height of the <select> element. Size attribute works with text, search, tel, url, email, and password input types. It determines how many characters can be displayed horizontally within the input field. The size attribute is particularly useful for controlling the visual layout of text input fields in forms.

```
Unset
<form>
   <label for="first_name">First Name:</label>
   <input type="text" id="first_name" name="first_name"
size="30" />
   <br />
   <br />
   <label for="first_name">Last Name:</label>
   <input type="text" id="first_name" name="first_name"
size="70" />
</form>
```

As you can see in the above example, the size of the first input field is 30 and second input field is 70. You can see the difference in width of the input field in the output screenshot

7. pattern

The pattern attribute lets you specify a regular expression that the input value must match, providing precise control over accepted formats.
A regular expression is like a magic spell that describes a specific text pattern.
We will learn more about regular expressions in upcoming modules.

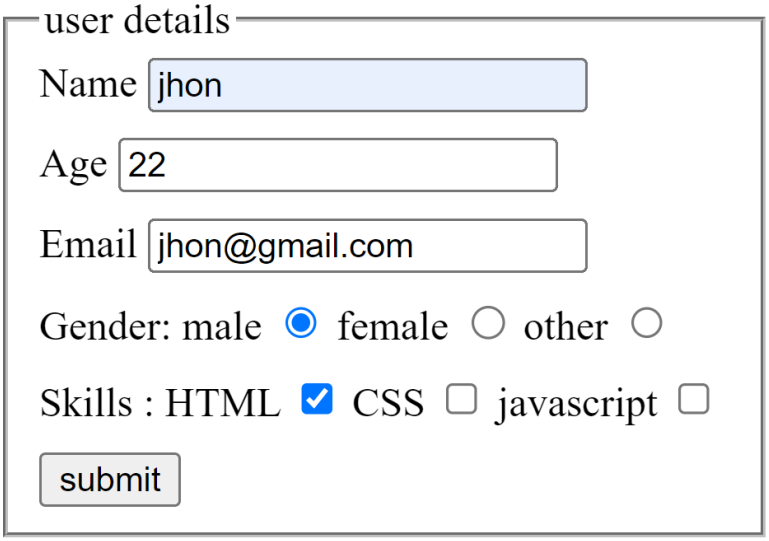
Example for a form with multiple inputs

Unset

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />


    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>input</title>
  </head>
  <body>
    <fieldset>
      <legend>user details</legend>
      <form action="#" method="GET">
        <!-- name (input text) -->
        <div>
          <label for="name"> Name </label>
          <input
            type="text"
            name="Name"
            id="name"
            required
            placeholder="Name"
          />
        </div>
        <!-- age (input number)-->
        <div>
          <label for="age">Age</label>
          <input type="number" name="Age" id="age" required
 placeholder="Age" />
        </div>
        <!-- email (input email)-->
        <div>
          <label for="email">  Email</label>
          <input
            type="email"
            name="Email"
            id="email"
            required
            placeholder="Email"
          />
```

```
        </div>
        <!-- gender (input radio) -->
        <div>
          Gender:
          <label for="male">male</label>
          <input type="radio" id="male" name="Gender"
value="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" />
        </div>
        <!-- skills  (input checkbox)  -->
        <div>
           Skills :
          <label for="HTML">HTML</label>
          <input type="checkbox" id="HTML" name="skill"
 value="HTML" checked />

          <label for="CSS">CSS</label>
          <input type="checkbox" id="CSS" name="skill"
 value="CSS" />

          <label for="javascript">javascript</label>
          <input
            type="checkbox"
            id="javascript"
            name="skill"
            value="javascript"
          />
         </div>
         <!-- submit -->


        <input type="submit">
      </form>
    </fieldset>
  </body>
</html>
```

Browser output



Here we create a simple html form that takes the user's details, like name, age, email, gender, and skills. After we submit the form, the form data will be sent to the server for processing.

Since we have used the "GET" method, you will see the form data in the browser tab in key value pairs (key=value), where key is the **name** that we provided for the inputs, and the **value** will be the same as entered by the user.

Browser url tab:

```
Unset
http://127.0.0.1:5500/index.html?Name=jon&Age=22&Email=jhon%40gmail.com&Gender=male&skill=HTML&skill=CSS#
```