

User Input in HTML: Forms and Controls

- [Download PDF](#)
- To access the updated handouts, please click on the following link: <https://yasirbhutta.github.io/html-css/docs/html-forms-controls.html>

Example #1:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
</head>
<body>
  <form method="get" action="">
    <p> Name:
      <input type="text" name="txtName" size="25" maxlength="25" />
    </p>
    <p>
      <input type="submit" value="Submit" />
    </p>
  </form>
</body>
</html>
```

Example #2:

```
<body>
  <form method="post" action="">
    <p> User Name:
      <input type="text" name="txtName" size="25" maxlength="25" />
    </p>
    <p> Password:
      <input type="password" name="txtPwd" id="txtPwd" size="10"
maxlength="10" />
    </p>
    <p>
      <input type="submit" value="Submit" />
    </p>
  </form>
</body>
```

Example #3:

```
<body>
  <form method="post" action="">
    <p> User Name:
      <input type="text" name="txtName" size="25" maxlength="25" />
    </p>
    <p> Password:
      <input type="password" name="txtPwd" id="txtPwd" size="10"
maxlength="10" />
    </p>
    <label for="txtPhone">Phone:
      <input type="text" id="txtPhone" />
    </label>
    <p>
      <input type="submit" value="Submit" />
    </p>
  </form>
</body>
```

Example #4:

```
<body>
  <p>
    Grocery Checklist
  </p>
  <form action="">
    <p>
      <label> <input type="checkbox" name="groceries" value="milk" /> Milk
</label>

      <label> <input type="checkbox" name="groceries" value="bread"
checked="checked" /> Bread </label>

      <label> <input type="checkbox" name="groceries" value="eggs" /> Eggs
</label>
    </p>
    <input type="submit" value="Submit" />
  </form>
</body>
```

Example #5:

```
<form action="" method="get">

  <div>
    <input type="color" id="head" name="head" value="#e66465">
    <label for="head">Head</label>
  </div>
```

```
<div>
  <input type="color" id="body" name="body" value="#f6b73c">
  <label for="body">Body</label>
</div>
<input type="submit" value="Submit">
</form>
```

Example #6:

```
<body>
  <!-- https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/date -->
  <form action="" method="get">
    <label for="start">Start date:</label>
    <input type="date" id="start" name="trip-start" value="2025-03-04"
min="2025-01-01" max="2025-12-31">
  </form>
</body>
```

Example #7:

```
<body>
  <form action="" method="get">
    <label for="email">Enter your globex.com email:</label>

    <input type="email" id="email" name="email" pattern=".+@gmail\.com"
size="30" required>
    <input type="submit" value="submit">
  </form>
</body>
```

In the HTML form you provided, the `pattern` and `required` attributes are used to control the input validation:

1. `pattern=".+@gmail\.com"`

- This is a **regular expression (regex)** that specifies the required format for the email input.
- Breakdown of the pattern:
 - `.+` → Matches **one or more** characters (ensures that something comes before "@").
 - `@gmail\.com` → Requires the email to end exactly with `@gmail.com`.
 - `\.` → The backslash (`\`) escapes the dot (`.`) because a dot in regex means "any character." Escaping ensures it is treated as a literal period.
- **Issue:** The label suggests a `globex.com` email is required, but the pattern only allows `gmail.com`. You may want to update the pattern accordingly.

2. `required`

- This makes the field **mandatory**. The user **must** enter an email before submitting the form.
- If this attribute is missing, the form can be submitted even if the input field is empty.

Potential Fix (for **globex.com** emails)

If you actually want to allow **@globex.com** emails, you should update the pattern:

```
<input type="email" id="email" name="email" pattern=".+@globex\.com" size="30" required>
```

See also:

- [HTML pattern Attribute - w3schools](#)

Example #8:

```
<body>
  <form method="post" enctype="multipart/form-data">
    <div>
      <label for="file">Choose file to upload</label>
      <input type="file" id="file" name="file" accept="image/png, image/jpeg"
multiple>
    </div>
    <div>
      <button>Submit</button>
    </div>
  </form>
</body>
```

Explanation of **multipart/form-data**, **accept** and **multiple** in Your HTML Form:

1. **enctype="multipart/form-data"**

- **Definition:** This attribute specifies how the form data should be encoded when submitting files.
- **Why It's Needed?**
 - When uploading files, form data is more complex than simple text inputs.
 - **multipart/form-data** ensures the file data is properly transmitted to the server.
- **How It Works?**
 - The form data is divided into multiple parts:
 - **File content**
 - **File metadata** (such as name, type)
 - **Other form fields**
 - The browser sends these parts as separate sections in the request body.

2. **accept="image/png, image/jpeg"**

- **Definition:** This attribute restricts the types of files a user can select.

- **How It Works?**
 - It allows only **PNG (.png) and JPEG (.jpg, .jpeg)** files to be chosen.
 - The file picker will **gray out** or **hide** unsupported file types.
- **Example Modifications:**
 - To allow **PDFs and Word documents:**

```
<input type="file" accept=".pdf, .doc, .docx">
```

- To allow **all images:**

```
<input type="file" accept="image/*">
```

3. multiple Attribute

- This allows users to select **more than one file** at once.

Example #9:

```
<body>
  <form action="" method="get">
    <label for="userId">User ID:
    <input type="text" id="userId" name="userId"></label>

    <input id="image" type="image" width="100" height="30" alt="Login"
      src="https://raw.githubusercontent.com/mdn/learning-
      area/master/html/forms/image-type-example/login.png" />

  </form>
</body>
```

Example #10:

```
<body>
  <!-- https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/number -->
  <form action="" method="get">
    <label for="score">Number of score (10-100):</label>

    <input type="number" id="score" name="score" min="0" max="100">
    <!-- placeholder -->
    <br>
    <input type="number" name="mul_score" placeholder="Multiple of 10">
    <input type="submit">
```

```
    </form>
  </body>
```

Example #11:

```
<body>
  <p>
    Age Category
  </p>
  <form action="">
    <p>
      <label><input type="radio" name="age" value="under20" />
        0-19 </label>
      <label><input type="radio" name="age" value="20-35" checked="checked"
/> 20-35 </label>
      <label><input type="radio" name="age" value="36-50" /> 36-50 </label>
      <label><input type="radio" name="age1" value="over50" /> Over 50
    </label>
    </p>
    <input type="submit" value="submit">
  </form>
</body>
```

The **checked** attribute is used in **radio buttons** and **checkboxes** to pre-select an option when the webpage loads.

1. What is the **checked** Attribute?

- It makes a **radio button** or **checkbox** selected by default.
- The user can still change the selection if they want.

2. How It Works with Checkboxes (**type="checkbox"**)

☑ Example: Pre-Selected Checkbox

```
<form>
  <label>
    <input type="checkbox" name="subscribe" checked> Subscribe to Newsletter
  </label>
</form>
```

🔍 Explanation:

- The checkbox is already **checked** when the page loads.
- The user can **uncheck** it if they don't want to subscribe.

3. How It Works with Radio Buttons (**type="radio"**)

🔗 Example: Pre-Selected Radio Button

```
<form>
  <label><input type="radio" name="gender" value="male" checked> Male</label>
  <label><input type="radio" name="gender" value="female"> Female</label>
</form>
```

Explanation:

- The "Male" option is **pre-selected**.
- Since radio buttons work in groups (same `name` value), the user can **only select one option at a time**.
- Clicking "Female" will **deselect** "Male."

4. Important Notes

- ✓ The `checked` attribute **does not require a value** (e.g., `<input checked>` is valid).
- ✓ If **multiple radio buttons** in a group have `checked`, the **last one in the HTML** will be selected.
- ✓ Without `checked`, **no option is selected by default**.

Example #12:

```
<body>
  <!-- https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/range -->
  <form action="" method="get">
    <p>Audio settings:</p>

    <div>
      <input type="range" id="volume" name="volume" min="0" max="11"
value="5">
      <label for="volume">Volume</label>
    </div>

    <div>
      <input type="range" id="cowbell" name="cowbell" min="0" max="100"
value="20" step="10">
      <label for="cowbell">Cowbell</label>
    </div>

    <!-- ++++++ -->
    <input type="range" list="tickmarks" name="rglist" value="80">

    <datalist id="tickmarks">
      <option value="0"></option>
      <option value="10"></option>
      <option value="20"></option>
      <option value="30"></option>
      <option value="40"></option>
      <option value="50"></option>
      <option value="60"></option>
      <option value="70"></option>
```

```
        <option value="80"></option>
        <option value="90"></option>
        <option value="100"></option>
    </datalist>
    <input type="submit" value="submit">
</form>
</body>
```

Example #13:

```
<body>
  <form>
    <div class="controls">

      <label for="id">User ID:</label>
      <input type="text" id="id" name="id" />

      <label for="id">Address</label>
      <input type="text" id="add" name="add" />

      <input type="reset" value="Reset">
      <input type="submit" value="Submit">

    </div>
  </form>
</body>
```

Example #14:

```
<body>
  <form action="" method="get">
    <input type="text" name="" id="" placeholder="enter user name">
    <input type="password" name="" id="" placeholder="please enter your
password">
    <input type="number" name="" id="" placeholder="age">
  </form>
</body>
```

Task: User Registration Form

Design and implement a **User Registration Form** using HTML form controls.

Task Requirements:

Create an online registration form with the following features:

1. Personal Information

- Full Name (Text Input, Required, Max Length: 25)
- Password (Password Input, Required, Max Length: 10)
- Phone Number (Text Input, Required)
- Date of Birth (Date Picker, Required)
- Email Address (Email Input, Must be a valid Gmail address, Required)

2. User Preferences

- Gender (Radio Buttons: Male, Female – Required)
- Hobbies (Checkboxes: Reading, Traveling, Sports – User can select multiple)
- Favorite Color (Color Picker)

3. File Upload

- Profile Picture (File Input, Accepts only JPG/PNG files, Allows multiple uploads)

4. Additional Settings

- Volume Level (Range Input, Min: 0, Max: 10)
- Address (Text Input, Required, Placeholder: "Enter your address")

5. Form Submission

- The form should include **Reset** and **Submit** buttons.
- Ensure the form fields have proper validation (e.g., required fields, number limits, email pattern).

◇ Submission Guidelines:

- Submit the HTML file with your form implementation.
- Ensure all fields are properly labeled and functional.