

HTML Forms & Inputs

Part 3: Các thành phần nâng cao và HTML5 Validation



Mục tiêu học tập

01. Form & Attributes

Hiểu rõ cấu trúc thẻ form và các thuộc tính quan trọng như action, method.

02. HTML5 Input Types

Sử dụng thành thạo email, date, number, range để tối ưu trải nghiệm trên mobile.

03. Form Controls

Áp dụng fieldset, legend, textarea và select box vào các trường hợp thực tế.

04. Semantic & Accessible

Thực hành gắn nhãn label đúng cách và validation để form thân thiện với mọi người dùng.

Form element — Container cho controls

- Thẻ `<form>` đóng vai trò là **container** bao bọc tất cả các phần tử tương tác (controls).
- Nó chịu trách nhiệm xác định **nơi gửi dữ liệu** (action) và **cách gửi** (method) khi người dùng submit.
- Chứa các thành phần như: input, label, button, textarea, v.v.

```
<!-- Cấu trúc cơ bản -->
<form action="/submit"
method="POST">
    <!-- Form controls go here
-->
</form>
```

Form attribute: action (URL xử lý)

- **Mục đích:** Chỉ định URL sẽ nhận và xử lý dữ liệu form khi người dùng nhấn Submit.
- **Các loại giá trị:**

Absolute URL

Đường dẫn đầy đủ

`https://site.com/api`

Relative URL

Đường dẫn tương đối

`/api/login`

Empty Value

Gửi về trang hiện tại

`action=""`

```
<form action="/process-data"> ... </form>
```

Form attribute: method (GET vs POST)

GET Method

- **Cơ chế:** Dữ liệu đính kèm vào URL (query string).
- **Đặc điểm:** Nhanh, có thể bookmark, cache được. Giới hạn độ dài.
- **Use Case:** Tìm kiếm, lọc dữ liệu.
- **Lưu ý:** KHÔNG dùng cho mật khẩu hay dữ liệu nhạy cảm.

POST Method

- **Cơ chế:** Dữ liệu gửi trong HTTP body.
- **Đặc điểm:** Bảo mật hơn, không giới hạn kích thước, không cache.
- **Use Case:** Đăng ký, đăng nhập, gửi bài viết, upload file.
- **Lưu ý:** Dùng cho hành động thay đổi dữ liệu server.

Form attributes nâng cao

enctype

Bắt buộc khi upload file.

```
enctype="multipart/form-data"
```

novalidate

Tắt kiểm tra dữ liệu mặc định của trình duyệt (HTML5 validation).

```
<form novalidate>
```

autocomplete

Bật/tắt gợi ý điền tự động của trình duyệt.

```
autocomplete="off"
```

```
<form action="/upload" method="POST" enctype="multipart/form-data">  
  <!-- File input here -->  
</form>
```

Submit form — Cách thực hiện

- **Nút Submit:** Click vào nút có `type="submit"` bên trong form.
- **Phím Enter:** Nhấn Enter khi đang focus vào bất kỳ trường input nào (trong form có nút submit).
- Hành vi mặc định: Trình duyệt sẽ thu thập dữ liệu và gửi request đến URL trong action.

```
<form>
  <input type="text"
name="query" />
  <!-- Cách 1: Button submit -->
  <button
type="submit">Gửi</button>
</form>
```

💡 *Mẹo: Luôn bao gồm nút submit để đảm bảo trải nghiệm người dùng rõ ràng, ngay cả khi hỗ trợ phím Enter.*

Input element cơ bản

- **type**: Loại dữ liệu (text, email, password...).
- **name**: Tên định danh gửi lên server. (Quan trọng!)
- **id**: Định danh duy nhất để liên kết với label hoặc CSS/JS.
- **value**: Giá trị mặc định ban đầu.
- **placeholder**: Văn bản gợi ý mờ (biến mất khi nhập).

```
<input  
  type="text"  
  name="username"  
  id="user_id"  
  placeholder="Nhập tên..."  
>
```


Text input (type="text")

- **maxlength / minlength:** Giới hạn số lượng ký tự nhập vào.
- **required:** Bắt buộc phải nhập, trình duyệt sẽ chặn submit nếu để trống.
- **readonly:** Chỉ đọc, không sửa được, nhưng **VẪN gửi** dữ liệu khi submit.
- **disabled:** Vô hiệu hóa hoàn toàn, **KHÔNG gửi** dữ liệu khi submit.

```
<!-- Yêu cầu nhập, tối đa 20  
ký tự -->
```

```
<input type="text"  
requiredmaxlength="20" />
```

```
<!-- Input bị vô hiệu hóa -->  
<input type="text" disabled  
/>
```

Password input (type="password")

- **Che giấu ký tự:** Hiển thị dấu chấm tròn hoặc sao (*) thay vì ký tự thực.
- **Validation:** Thường kết hợp với `minlength` để đảm bảo độ mạnh của mật khẩu.
- **Lưu ý bảo mật:** Trang chứa form mật khẩu **BẮT BUỘC** phải dùng HTTPS để mã hóa đường truyền.

```
<!-- Mật khẩu tối thiểu 8 ký  
tự -->  
<input type="password"  
      requiredminlength="8" />
```

Email input (type="email")

Tại sao nên dùng type="email"?

- **Validation tự động:** Trình duyệt tự kiểm tra định dạng (phải có @, domain...).
- **Trải nghiệm Mobile:** Bàn phím ảo trên điện thoại sẽ tự động hiển thị phím @ và .com để nhập liệu nhanh hơn.

```
<label for="mail">Email:</label>  
<input type="email" id="mail"  
name="user_email" />
```

Date input (type="date")

- **Native Date Picker:** Trình duyệt tự động hiển thị lịch chọn ngày.
- **Định dạng chuẩn:** Giá trị gửi về server LUÔN là **YYYY-MM-DD** (bất kể hiển thị theo vùng miền nào).

```
<!-- Chọn ngày từ hôm nay trở đi -->
<input type="date" min="2023-01-01" />

<!-- Giới hạn max -->
<input type="date" max="2024-12-31" />
```

Các input thời gian khác

type="time"

Chọn giờ phút (HH:mm).

```
<input type="time" />
```

type="datetime-local"

Chọn ngày và giờ (không có múi giờ).

```
<input type="datetime-local" />
```

type="month"

Chọn tháng và năm (YYYY-MM).

```
<input type="month" />
```

type="week"

Chọn tuần và năm (YYYY-Www).

```
<input type="week" />
```

Number input (type="number")

- **Mục đích:** Nhập số lượng, tuổi, giá tiền...
- **Thuộc tính min / max:** Giới hạn giá trị tối thiểu và tối đa.
- **Thuộc tính step:** Quy định bước nhảy (ví dụ 0.5, 10...).
- **Mobile UX:** Tự động hiển thị bàn phím số giúp nhập liệu nhanh.

```
<!-- Nhập tuổi từ 18 đến 99 ->
<input type="number"
  min="18"max="99" />

<!-- Bước nhảy 0.5 -->
<input type="number"
  step="0.5" />
```

Checkbox (type="checkbox")

- **Mục đích:** Cho phép người dùng chọn **NHIỀU** tùy chọn cùng lúc.
- **Cơ chế gửi:** Chỉ những checkbox **được tích** mới gửi giá trị về server.
- **Thuộc tính:** checked (chọn sẵn).

```
<!-- Chọn sở thích -->
<label>
  <input type="checkbox"
name="hobbies" value="music" />
  Nghe nhạc
</label>
<label>
  <input type="checkbox"
name="hobbies" value="code" checked
/>
  Lập trình
</label>
```

Radio button (type="radio")

- **Mục đích:** Cho phép người dùng chọn **DUY NHẤT** một tùy chọn trong nhóm.
- **Nhóm Radio:** Các radio button phải có **CÙNG name** để hoạt động loại trừ lẫn nhau.
- **Mặc định:** Dùng thuộc tính checked để chọn sẵn một giá trị.

```
<!-- Chọn giới tính (Name giống nhau)
-->
<label>
  <input type="radio" name="gender"
value="male" />
  Nam
</label>
<label>
  <input type="radio" name="gender"
value="female" checked />
  Nữ
</label>
```


File input (type="file")

- **accept:** Giới hạn loại file được chọn (ví dụ: chỉ ảnh, chỉ PDF).
- **multiple:** Cho phép giữ phím Ctrl/Shift để chọn nhiều file cùng lúc.

```
<!-- Chọn nhiều ảnh (png, jpeg) -->  
<input type="file"  
      accept="image/png, image/jpeg"  
      multiple  
/>
```

Range input (type="range")

- **Chức năng:** Tạo thanh trượt (slider) cho phép chọn giá trị trong một khoảng.
- **Use Case:** Điều chỉnh âm lượng, độ sáng, giá tiền...
- **Lưu ý:** Giá trị không chính xác tuyệt đối như nhập số, phù hợp cho ước lượng.

```
<!-- Volume Slider -->  
<label  
  for="vol">Volume</label>  
<input type="range"  
  id="vol"min="0"max="100" />
```



Color input (type="color")

- **Giao diện:** Hiển thị bảng chọn màu (color picker) của hệ điều hành.
- **Giá trị:** Luôn trả về mã Hex 6 ký tự.
- #RRGGBB (Ví dụ: #ff0000)

```
<label>Chọn màu chủ  
đạo:</label>  
<input type="color"  
name="favcolor"  
value="#ff0000" />
```



Search input (type="search")

- **Về cơ bản:** Giống hệt type="text" nhưng mang ý nghĩa ngữ nghĩa (semantic) là ô tìm kiếm.
- **Tiện ích:** Một số trình duyệt tự động thêm nút "X" để xóa nhanh nội dung khi nhập liệu.

```
<form action="/search">
  <input type="search" placeholder="Tìm
kiếm..." />
</form>
```

Tìm kiếm... 

URL input (type="url")

- **Validation:** Trình duyệt tự động kiểm tra xem nội dung nhập vào có phải là URL hợp lệ hay không (phải bắt đầu bằng http://...).
- **Mobile Keyboard:** Bàn phím ảo sẽ hiển thị thêm các phím **.com**, **/** để nhập địa chỉ web dễ dàng.

```
<!-- Nhập website cá nhân -->  
<input type="url"  
placeholder="https://example.  
com" />
```

Tel input (type="tel")

- **Mobile UX:** Kích hoạt bàn phím số (giống khi gọi điện) trên điện thoại, rất tiện lợi.
- **Validation:** KHÔNG có validation tự động vì định dạng số điện thoại trên thế giới rất đa dạng.
- **Giải pháp:** Dùng thuộc tính **pattern** với RegEx để kiểm tra.

```
<!-- Số điện thoại 10 chữ số  
-->  
<input type="tel"  
pattern="[0-9]{10}" />
```

Hidden input (type="hidden")

- **Khái niệm:** Input không hiển thị trên giao diện nhưng giá trị của nó VẪN được gửi đi khi submit form.

Use Cases phổ biến:

- Gửi ID của bản ghi cần sửa/xóa.
- Gửi CSRF Token (bảo mật).
- Lưu trạng thái (state) tracking.

```
<input type="hidden"  
name="user_id" value="12345" />
```

Textarea (nhiều dòng)

- **Mục đích:** Nhập văn bản dài (ghi chú, mô tả, tin nhắn...).
- **Cú pháp:** Là thẻ đóng/mở `<textarea>...</textarea>` (không phải self-closing như input).
- **Thuộc tính:** rows (số dòng), cols (số cột), maxlength (giới hạn ký tự).

```
<!-- Khung nhập liệu 4 dòng -  
->  
<textarea  
  name="message" rows="4" cols="5  
  0">  
  Nội dung mặc định...  
</textarea>
```

Nội dung mặc định...

Select dropdown

- **Cấu trúc:** Thẻ cha `<select>` chứa các thẻ con `<option>`.
- **Thuộc tính multiple:** Cho phép chọn nhiều option (thường hiển thị dạng list box thay vì dropdown).
- **Selected:** Đặt thuộc tính `selected` vào option muốn chọn mặc định.

```
<label>Quốc gia:</label>
<select name="country">
  <option value="vn">Việt
  Nam</option>
  <option value="us">Hoa
  Kỳ</option>
  <option value="jp"
  selected>Nhật Bản</option>
</select>
```

Optgroup — Nhóm options

- **Chức năng:** Phân nhóm các option có liên quan, giúp người dùng dễ dàng tìm kiếm trong danh sách dài.

```
<select>
  <optgroup label="Miền Bắc">
    <option>Hà Nội</option>
    <option>Hải Phòng</option>
  </optgroup>
  <optgroup label="Miền Nam">
    <option>TP.HCM</option>
  </optgroup>
</select>
```

Button types (submit, reset, button)

type="submit"

Gửi dữ liệu form đi.

(Đây là type mặc định của button trong form)

```
<button type="submit">
```

type="reset"

Xóa sạch dữ liệu về mặc định.

 **Ít dùng vì dễ bấm nhầm.**

```
<button type="reset">
```

type="button"

Không làm gì cả (Button vô tri).
Dùng để gắn sự kiện JavaScript (click).

```
<button type="button">
```

Input submit vs Button submit

<input type="submit">

- Kiểu cũ (Legacy).
- Chỉ chứa được văn bản thuần (thông qua thuộc tính value).
- Khó tùy biến giao diện phức tạp.

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

<button type="submit">

- Hiện đại & Linh hoạt.
- Có thể chứa HTML (ảnh, icon, thẻ span, bold...).
- Dễ CSS (pseudo-elements).

```
<button>  Gửi  
</button>
```

Label element — Liên kết với input

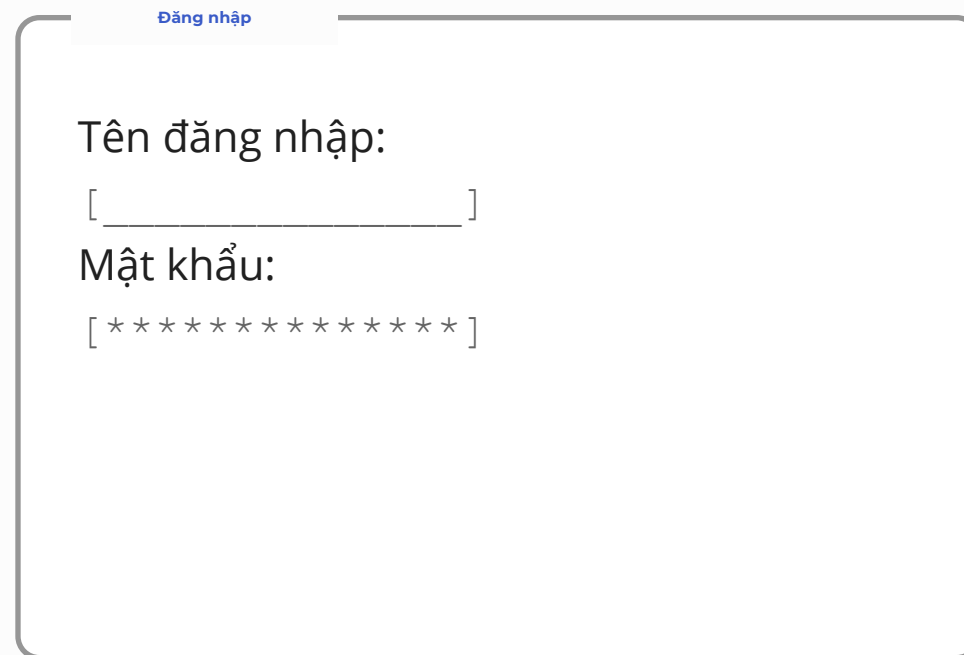
- **Lợi ích UX:** Click vào nhãn (label) sẽ tự động focus vào ô input. Tăng vùng click cho checkbox/radio.
- **Accessibility:** Giúp trình đọc màn hình đọc tên trường cho người khiếm thị.

```
<!-- Cách 1: Dùng for + id -->
<label
  for="user">Tên:</label>
<input id="user" type="text"
/>
```

```
<!-- Cách 2: Lồng nhau -->
<label>
  Tên:
  <input type="text" />
</label>
```

Fieldset & Legend — Nhóm controls

- **<fieldset>**: Gom nhóm các trường input có liên quan lại với nhau (ví dụ: thông tin cá nhân, địa chỉ giao hàng).
- **<legend>**: Đặt tiêu đề (caption) cho nhóm fieldset đó.
- **Lợi ích**: Giúp form dài trở nên dễ nhìn hơn và hỗ trợ tốt cho trình đọc màn hình.



Đăng nhập

Tên đăng nhập:
[_____]

Mật khẩu:
[*****]

```
<fieldset>  
  <legend>Đăng nhập</legend>  
  ...inputs...  
</fieldset>
```

HTML5 validation: required

- **Text Input:** Ngăn người dùng submit nếu để trống.
- **Checkbox:** Bắt buộc phải tích chọn (ví dụ: "Tôi đồng ý điều khoản").
- **Radio:** Bắt buộc phải chọn một trong các tùy chọn trong nhóm.

```
<input type="text" required  
>  
<input type="checkbox"  
required />
```

 Please fill out this field.