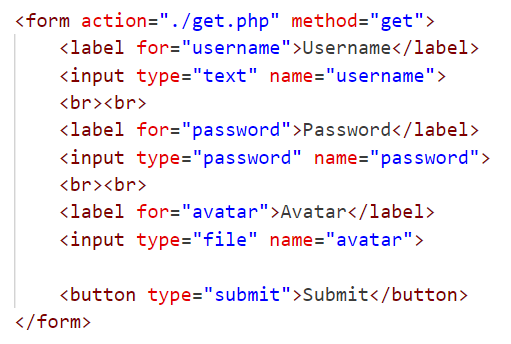
**1. Khởi tạo Form**

Một form (biểu mẫu) trong HTML được sử dụng để thu thập dữ liệu từ người dùng và gửi dữ liệu đó đến máy chủ để xử lý. Form có thể chứa nhiều loại trường đầu vào (input fields) khác nhau, như hộp văn bản, nút radio, hộp kiểm tra (checkbox), và nút submit.



<form>: Thẻ mở đầu của biểu mẫu.

* action: Xác định URL của trang sẽ xử lý dữ liệu form. Ví dụ: "./get.php".
* method: Xác định phương thức gửi dữ liệu. Giá trị có thể là "GET" hoặc "POST".

<label>: Thẻ nhãn để mô tả các trường nhập liệu.

<input>: Thẻ để nhận dữ liệu đầu vào từ người dùng.

* type: Xác định loại đầu vào (text, password, submit, v.v.).
* name: Xác định tên của trường đầu vào, được gửi đến máy chủ để xử lý.

<button type="submit">: Tạo nút submit để gửi dữ liệu từ form đến máy chủ.

1. **Hiểu về GET và POST**

GET và POST là hai phương thức được sử dụng để gửi dữ liệu từ form HTML đến máy chủ web để xử lý. Mỗi phương thức có cách thức hoạt động và mục đích sử dụng khác nhau.

**GET Method:**

*Cách hoạt động:*

* Dữ liệu được gửi dưới dạng chuỗi truy vấn (query string) và được đính kèm vào URL của trang đích. Chuỗi truy vấn bắt đầu bằng dấu ? và bao gồm các cặp key=value.
* Ví dụ: Khi sử dụng phương thức GET, URL có thể trông như sau: http://example.com/xuly.php**?username=john&password=123**.

*Đặc điểm:*

* Dữ liệu gửi qua GET được hiển thị trên thanh địa chỉ của trình duyệt.
* Hạn chế về kích thước dữ liệu (khoảng 2000 ký tự, tùy trình duyệt).
* Thường được sử dụng cho các yêu cầu mà không yêu cầu bảo mật cao, như tìm kiếm hoặc lọc dữ liệu.

**POST Method:**

*Cách hoạt động:*

* Dữ liệu được gửi trong thân (body) của yêu cầu HTTP, không hiển thị trên URL.

*Đặc điểm:*

* Bảo mật hơn so với GET vì dữ liệu không được hiển thị trên URL.
* Không giới hạn kích thước dữ liệu gửi đi.
* Thường được sử dụng cho các thao tác nhạy cảm như gửi thông tin đăng nhập, đăng ký tài khoản, hoặc tải lên tệp.

1. **Phân biệt GET và POST**

GET và POST là hai phương thức chính được sử dụng để gửi dữ liệu từ client (trình duyệt) đến server (máy chủ) trong các ứng dụng web. Cả hai đều là thành phần của giao thức HTTP, nhưng chúng có cách thức hoạt động và ứng dụng khác nhau.

|  |  |  |
| --- | --- | --- |
| Tiêu chí | GET | POST |
| Mục đích | GET được sử dụng để yêu cầu dữ liệu từ máy chủ. | POST được sử dụng để gửi dữ liệu đến máy chủ để xử lý. |
| Cách gửi dữ liệu | Dữ liệu được gửi qua URL dưới dạng chuỗi truy vấn (query string). | Dữ liệu được gửi qua phần thân (body) của yêu cầu HTTP. |
| Kích thước dữ liệu | Giới hạn bởi độ dài URL (thường tối đa 2048 ký tự). | Không giới hạn chính thức về kích thước dữ liệu. Thích hợp để gửi dữ liệu lớn. |
| Bảo mật | Không an toàn, vì dữ liệu được hiển thị trong URL và lưu lại trong lịch sử trình duyệt hoặc nhật ký máy chủ. | An toàn hơn GET, vì dữ liệu không hiển thị trong URL, nhưng vẫn có thể bị chặn nếu không sử dụng HTTPS. |
| Hiển thị dữ liệu | Dữ liệu được hiển thị trực tiếp trong URL (có thể nhìn thấy). | Dữ liệu không được hiển thị trong URL (ẩn đối với người dùng). |
| Tính idempotent | GET là idempotent, tức là gọi nhiều lần một yêu cầu GET sẽ luôn trả về cùng một kết quả, không thay đổi trạng thái của server. | POST không phải là idempotent, vì mỗi yêu cầu POST có thể thay đổi trạng thái của server (ví dụ: gửi đơn hàng, tạo tài khoản). |
| Ứng dụng | Phù hợp cho việc lấy thông tin, như tìm kiếm, lấy tài liệu, hoặc các tác vụ không làm thay đổi dữ liệu trên server. | Phù hợp cho các hành động yêu cầu thay đổi hoặc lưu trữ dữ liệu trên server, như đăng nhập, gửi form liên hệ, đăng ký tài khoản. |
| Lưu trữ bộ nhớ cache | GET có thể được lưu trữ trong bộ nhớ cache của trình duyệt hoặc máy chủ proxy. | POST không được lưu trữ trong bộ nhớ cache bởi trình duyệt hoặc proxy. |
| Đánh dấu trang (Bookmark) | Có thể đánh dấu trang (Bookmark) một URL với GET, vì URL chứa toàn bộ thông tin. | Không thể đánh dấu trang một yêu cầu POST, vì dữ liệu nằm trong thân yêu cầu không được đưa vào URL. |

**Khi nào nên sử dụng GET và POST?**

*Sử dụng GET khi:*

* Bạn muốn truy vấn dữ liệu từ máy chủ mà không làm thay đổi dữ liệu.
* Dữ liệu yêu cầu nhỏ và có thể được hiển thị trong URL.
* Bạn muốn cho phép người dùng đánh dấu trang hoặc chia sẻ URL của yêu cầu.

*Sử dụng POST khi:*

* Bạn cần gửi dữ liệu lớn hoặc nhạy cảm (như mật khẩu, thông tin cá nhân).
* Hành động sẽ làm thay đổi dữ liệu trên máy chủ (như thêm mới, cập nhật dữ liệu).
* Bạn muốn giữ dữ liệu được gửi ẩn khỏi URL.