### 1. ****Xử lý lỗi trong PHP****

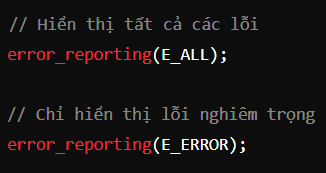
Trong PHP, lỗi là những vấn đề xảy ra trong quá trình thực thi của chương trình, có thể làm gián đoạn hoặc ngừng chương trình. PHP cung cấp nhiều cách để xử lý lỗi, từ đơn giản đến phức tạp.

#### ****Phân loại lỗi trong PHP:****

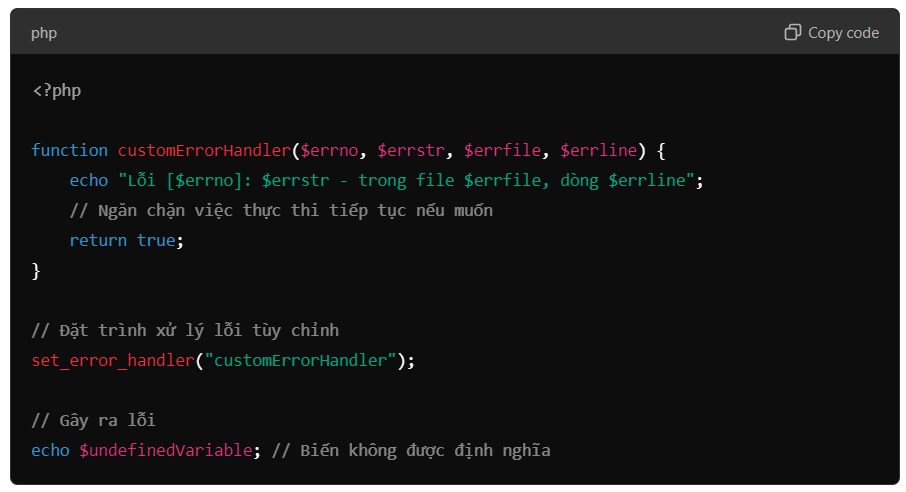
* **Parse Error (Lỗi phân tích cú pháp):** Xảy ra khi PHP gặp lỗi cú pháp, chẳng hạn như thiếu dấu chấm phẩy hoặc ngoặc. Những lỗi này thường xảy ra trước khi script được thực thi.
* **Fatal Error (Lỗi nghiêm trọng):** Xảy ra khi PHP gặp lỗi không thể khắc phục, chẳng hạn như gọi một hàm hoặc lớp không tồn tại.
* **Warning (Cảnh báo):** PHP gặp vấn đề nhưng vẫn có thể tiếp tục thực thi, chẳng hạn như khi cố gắng bao gồm một file không tồn tại bằng include.
* **Notice (Thông báo):** Lỗi không nghiêm trọng và thường chỉ là cảnh báo cho những vấn đề tiềm ẩn, chẳng hạn như sử dụng biến chưa được khởi tạo.

#### ****Xử lý lỗi với**** error\_reporting() ****và**** set\_error\_handler()****:****

**error\_reporting():** Hàm này được sử dụng để đặt mức độ báo cáo lỗi mà PHP sẽ hiển thị. Bạn có thể chọn mức độ lỗi bạn muốn hiển thị bằng cách sử dụng các hằng số như E\_ALL, E\_WARNING, E\_NOTICE.

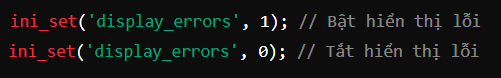


**set\_error\_handler():** Hàm này cho phép bạn xác định một hàm tùy chỉnh để xử lý lỗi. Hàm này sẽ được gọi mỗi khi có lỗi xảy ra.

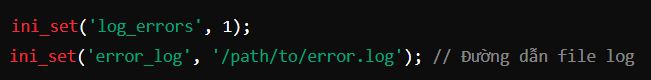


#### ****Xử lý lỗi mặc định của PHP:****

**display\_errors:** Đặt trong file php.ini hoặc trực tiếp trong mã PHP. Khi display\_errors bật (On), PHP sẽ hiển thị lỗi trên trình duyệt. Khi tắt (Off), lỗi sẽ không hiển thị nhưng có thể được ghi lại nếu log\_errors được bật.



**log\_errors:** Đặt trong file php.ini để ghi lại lỗi vào file log.

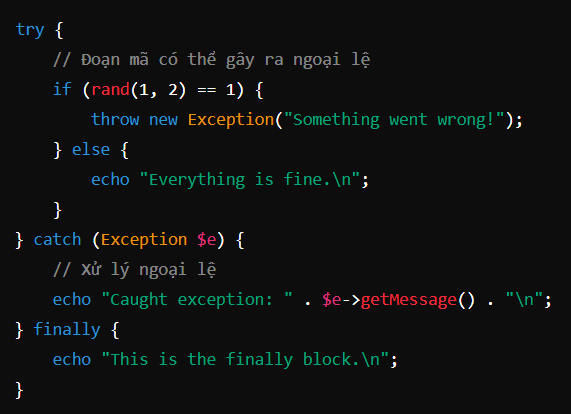


### 2. ****Xử lý Exception trong PHP****

**Exception** là một cơ chế xử lý lỗi tiên tiến hơn trong PHP, cho phép bạn bắt và xử lý các lỗi theo cách kiểm soát được. Exception trong PHP được xây dựng dựa trên lớp cơ sở Exception và sử dụng cấu trúc try...catch...finally để quản lý lỗi.

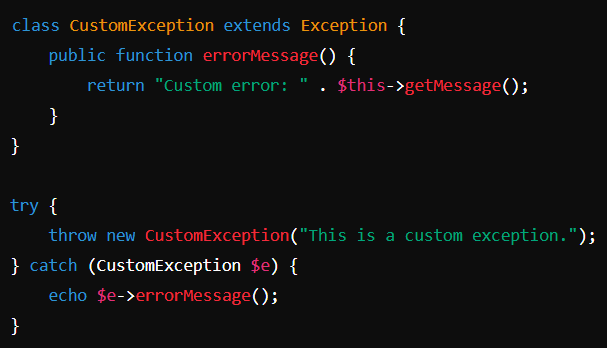
#### ****Cấu trúc**** try...catch...finally****:****

* **try:** Khối mã được bọc trong try là nơi có thể xảy ra lỗi.
* **catch:** Nếu lỗi xảy ra trong khối try, nó sẽ bị bắt trong khối catch. Bạn có thể có nhiều khối catch để xử lý các loại exception khác nhau.
* **finally:** Khối mã finally sẽ được thực thi bất kể có lỗi xảy ra hay không. Khối này thường được sử dụng để giải phóng tài nguyên hoặc thực hiện các bước dọn dẹp.



**Tạo lớp Exception tùy chỉnh:**

Bạn có thể tạo các lớp Exception của riêng mình để xử lý các trường hợp đặc biệt trong ứng dụng.



#### ****Các phương thức phổ biến trong lớp**** Exception****:****

* **getMessage():** Trả về thông báo lỗi.
* **getCode():** Trả về mã lỗi (nếu có).
* **getFile():** Trả về tên file mà ngoại lệ đã được ném ra.
* **getLine():** Trả về dòng mà ngoại lệ đã được ném ra.
* **getTraceAsString():** Trả về thông tin chi tiết về ngăn xếp khi lỗi xảy ra.

### ****Tổng kết****

* **Xử lý lỗi (Error Handling):** Là cách bạn quản lý các vấn đề xảy ra trong quá trình thực thi mã PHP, có thể làm gián đoạn hoặc dừng chương trình. Sử dụng các hàm như error\_reporting() và set\_error\_handler() để quản lý lỗi hiệu quả.
* **Xử lý Exception:** Là một cơ chế xử lý lỗi mạnh mẽ hơn, cho phép bạn bắt và xử lý các lỗi theo cách kiểm soát được thông qua cấu trúc try...catch...finally. Bạn có thể tạo các lớp exception tùy chỉnh để quản lý các trường hợp lỗi cụ thể trong ứng dụng.