**Tiêu chuẩn code là tập hợp các quy tắc để viết mã nguồn một cách nhất quán và dễ hiểu.**

Việc tuân thủ tiêu chuẩn code mang lại nhiều lợi ích, bao gồm:

1. **Tăng tính nhất quán**: Giúp mã nguồn đồng nhất, dễ đọc và dễ hiểu.
2. **Dễ bảo trì**: Mã dễ bảo trì và sửa lỗi, đặc biệt trong các dự án lớn.
3. **Giảm lỗi**: Giúp giảm thiểu các lỗi phát sinh.
4. **Cải thiện hợp tác**: Hỗ trợ làm việc nhóm hiệu quả hơn.
5. **Hỗ trợ code review**: Quá trình kiểm tra mã nhanh chóng và hiệu quả hơn.
6. **Dễ tích hợp công cụ**: Dễ dàng tích hợp với các công cụ kiểm tra tự động.
7. **Bảo vệ chất lượng**: Đảm bảo mã nguồn chất lượng và dễ mở rộng.
8. **Đào tạo nhanh chóng**: Hướng dẫn lập trình viên mới dễ dàng.
9. **Thể hiện tính chuyên nghiệp**: Tăng tính chuyên nghiệp trong phát triển phần mềm.
10. **Giảm phụ thuộc vào cá nhân**: Dễ thay thế nhân sự mà không ảnh hưởng đến dự án.

**1. Tên Biến**

* Sử dụng kiểu camelCase: Tên biến nên được viết theo kiểu camelCase, bắt đầu bằng một chữ cái viết thường và các từ sau đó bắt đầu bằng chữ cái viết hoa.

Ví dụ: $userName, $totalAmount, $orderListForHome.

* Đặt tên có ý nghĩa: Tên biến phải mô tả rõ ràng vai trò và chức năng của biến.

Ví dụ: Sử dụng $userAge thay vì $ua, $productList thay vì $pl.

*Tránh sử dụng ký tự đặc biệt: Tránh sử dụng ký tự đặc biệt trong tên biến ngoại trừ dấu gạch dưới \_ trong một số trường hợp đặc biệt.*

**2. Tên Hàm**

* Sử dụng kiểu camelCase: Giống như tên biến, tên hàm cũng nên sử dụng kiểu camelCase, bắt đầu bằng chữ cái viết thường.

Ví dụ: getUserInfo(), calculateTotalPrice(), sendEmailNotification().

* Đặt tên theo hành động: Tên hàm nên bắt đầu bằng động từ để chỉ rõ chức năng của hàm.

Ví dụ: fetchData(), saveRecord(), deleteUser().

* Đặt tên có ý nghĩa: Tên hàm nên mô tả chính xác hành động mà hàm thực hiện.

Ví dụ: createUserSession() thay vì doTask().

**3. Tên File**

* Sử dụng kiểu PascalCase hoặc snake\_case:
  + PascalCase: Các từ trong tên file được viết hoa chữ cái đầu tiên của mỗi từ, không có khoảng trắng hoặc dấu gạch dưới.

Ví dụ: UserController.php, ProductModel.php.

* + snake\_case: Các từ trong tên file được viết thường và phân tách bằng dấu gạch dưới \_.

Ví dụ: user\_list.php, product\_show.php.

* Đặt tên theo chức năng: Tên file nên phản ánh chức năng hoặc vai trò của nó trong ứng dụng.

Ví dụ: DatabaseConnection.php cho file kết nối cơ sở dữ liệu, UserRepository.php cho file chứa logic xử lý liên quan đến người dùng.

**4. Tên Hằng Số (Constants)**

* Sử dụng kiểu UPPERCASE\_SNAKE\_CASE: Tên hằng số nên viết hoa toàn bộ và sử dụng dấu gạch dưới \_ để phân tách các từ.

Ví dụ: MAX\_UPLOAD\_SIZE, DEFAULT\_LANGUAGE, API\_KEY.

* Đặt tên miêu tả rõ ràng: Tên hằng số nên miêu tả chính xác giá trị mà nó đại diện.

Ví dụ: MIN\_PASSWORD\_LENGTH thay vì MIN\_LEN.