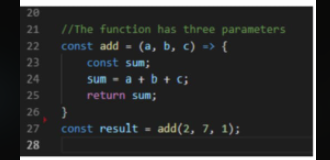
I. Lưu ý:

**1.** **Biến global:** là biến mà ta khai báo bên ngoài và không nằm bên trong 1 hàm cụ thể nào cả.

**2. Side effect:** là những hành động xảy ra khi ta tác động thứ gì đó vào bên trong hàm khiến cho bên ngoài hàm cũng bị tác động vào ngược (trừ khi hàm gọi tới là 1 pure funtion). Ví dụ khi ta thay đổi giá trị hay thuộc tính của 1 hay nhiều biến global, hiển thị dữ liệu ra màn hình.

**3.** **Pure function (hàm thuần khiết):** là hàm luôn trả về cùng một kêt quả nếu được truyền vào các tham số không đổi, nó ko hề phụ thuộc vào bất kì trạng thái hoặc dữ liệu nào, cũng như những thay đổi trong khi chương trình đc chạy.



**II. LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG:**

1. Hướng đối tượng nó là cái mếu gì?

🡪 Hiểu nôm na rằng, hướng dối tướng nó như một cách hành code(viết code) và trong lặp trình có nhiều cách hành code :

+ Lập trình thủ tục (Procedural Programing) (cái đơn giản nhất) : cụ thể nó đc chia thành nhiều hàm, nhiều function con và những các function này người ta gọi là thủ tục. Và mạnh code của toàn bộ chương trình lúc này sẽ là gọi tuần tự các hàm (thủ tục) theo thứ tự từ trên xuống dưới.

+ Lập trình hướng đối tượng (Object-oriented programming, viết tắt: OOP) : Hiểu nôm na nó là lối viết code mà nó dựa trên một khái niệm gọi là Object (đối tượng) . Và code của chúng ta viết nó phải xoay quanh cái khái niệm này. Làm gì thì làm mà nên nhớ luôn hướng về em nó ☺ nhá.

2. Lợi ích của lập trính theo hướng đối tượng:

- Chương trình , cấu trúc nó sẽ dễ đọc hơn, đồng nghĩa với việc dễ bảo trì hơn, dễ dò lỗi và dễ sữa lỗi . Và nó cũng tăng tính sử dụng lại của code nữa..

3. Hai cái khái niệm chính trong lập trình hướng đối tượng : Class & Object

- Class : có thể hiểu là nó định nghĩa cho một tập dữ liệu nào đó , nó là hình mẫu chung chung cho cái khái niệm gì đó.

- Object : thì nó là cái cụ thể cho cái khác niệm chung chung đó.

II. Lập trình hướng đối tượng trong javascript:

Trong javascript nó sẽ 4 tính chất cơ bản:

1. Abstraction (tính trừu tượng): nó như là 1 cái Ojb hay là 1 cái class và mình chỉ khai báo cách sài chứ không chỉ rõ chi tiết triển khai.
2. Encapsulation (tính đóng gói): là một quá trình liên kết dữ liệu với các fucntion hoạt động trên dữ liệu. Tính năng đóng gói cho phép chúng ta kiểm soát và xác nhận dữ liệu. Trong JavaScript, các biến giống như dữ liệu.
3. Inheritance (tính kế thừa): Kế thừa JavaScript là một cơ chế cho phép chúng ta tạo các lớp mới trên cơ sở các lớp đã tồn tại. Nó cung cấp sự linh hoạt cho lớp con để sử dụng lại các phương thức và biến của lớp cha. Từ khóa mở rộng JavaScript được sử dụng để tạo một lớp con trên cơ sở một lớp cha.
4. Polymorphism (tính đa hình) :

+ method overloading : ví dụ thằng cha nó có 1 func gì đó mà thằng cũng có func giống hệt thì thằng con nó sẽ over ride cái thằng cha.

+ method overriding : mình có nhiều các method hay func mà nó giống nhau nhưng các tham số nó khác nhau thì nó sẽ làm các chức năng khác nhau.