**Câu 1 : Repository là gì? có bao nhiêu loại repository?**

Thường được gọi ngắn gọn là repo, là nơi chứa toàn bộ mã nguồn, bao gồm toàn bộ các file và lịch sử của các file đó.Repository chứa tất cả các commit.

Có 2 loại repository:

• Local Repository: Ở trên máy của lập trình viên

• Remote Repository: Ở trên một máy chủ chia sẻ (chẳng hạn như GitHub)

**Câu 9 : Phân biệt innerHTML & innerText ?**

* innerHTML : chạy nó dưới dạng nội dung HTML
* innerText : chèn chuỗi vào phần tử, truy xuất và đặt ndung of thẻ thành vbản thuần túy.

**Câu 11 : Các thuộc tính cơ bản của thẻ form ?**

* Action : có giá trị là một URL, quy định nơi dữ liệu được gửi đến.
* Name : qđịnh tên của form, sau đó có thể dùng để truy xuất đến các
  + giá trị trong form.
* Method : xđ cách thức và dữ liệu dc gửi đi gồm có POT và GET.

**Câu 12 ; Phân biệt phương thức get và post trong thẻ form.**

|  |  |
| --- | --- |
| **GET** | **POST** |
| * Dữ liệu được gửi đi sẽ hiển thị trên thanh địa chỉ của trình duyệt. * Không nên sử dụng để gửi các dữ liệu nhạy cảm (chẳng hạn như mật khẩu) * Thường được sử dụng để gửi những dữ liệu nhỏ | * Ko hiển thị lên. * Bảo mật tt * Ko hạn chế |

**Câu 13 : Cách khai báo biến bằng từ khóa let và var khác nhau như thế nào ?. Phạm vi của biến?**

|  |  |
| --- | --- |
| **Let ( hay sd hơn )** | **Var** |
| * Ko thể khai báo lại * Ko có * Có phạm vi truy cập là block scope | * Có thể khai báo lại * Có cơ chế hosting, đưa dòng k.báo lên đầu, có thể sd trước khi kbáo * Có phạm vi truy cập là function scope |

Phạm vị của biến :

* Biến toàn cục : đc khai báo ngoài function
* Biến cục bộ : trong function

**Câu 14 : Trong Javascript có bao nhiêu loại kiểu dữ liệu ?. Làm thế nào để xác định được biến có kiểu dữ liệu gì?**

Có 2 kiểu :

* Kiểu nguyên thủy : number, string, boolean, underfine , null, symbol.
* Kiểu đối tượng (Object) : Array, Date,…

**Câu 19 : Toán tử 3 ngôi là gi ?**

là một toán tử được cấu tạo bởi ba đối số gồm *biểu thức điều kiện*, *kết quả khi điều kiện đúng* và *kết quả khi điều kiện sai*. Kết quả ở đây có thể là một giá trị được trả về, cũng có thể là một xử lý sẽ thực hiện sau đó tùy thuộc vào *điều kiện* chỉ định là đúng hay sai.

CÚ Phap : condition ? true\_value : false\_value;

**Câu 35 : Hàm có return và không có return**

hàm không có return:

không thể thay đổi nội dung thông báo mỗi khi sử dụng.

Hàm có return :

Vậy khi sử dụng lệnh return trong hàm thì khi thực thi chương trình của hàm sẽ dừng lại ngay lệnh return đó, và giá trị khi gán biến cho hàm chính là giá trị của lệnh return.

**Câu 38 : Các tính chất của OOP**

### **Tính đóng gói (Encapsulation):**Khả năng che dấu 1 vài tt của đối tượng

Ví dụ ta thấy một viên thuốc chữa cảm. Chúng ta chỉ biết nó chữa cảm sổ mũi nhức đầu và một số thành phần chính, còn cụ thể bên trong nó có những hoạt chất gì thì hoàn toàn không biết.

### **Tính kế thừa (Inheritance):**Là kỹ thuật cho phép kế thừa lại những tính năng mà một đối tượng khác đã có, giúp tránh việc code lặp dư thừa mà chỉ xử lý công việc tương tự

### **Tính đa hình (Polymorphism ):**Là một đối tượng thuộc các lớp khác nhau có thể hiểu cùng một thông điệp theo cách khác nhau.

Ví dụ đa hình trong thực tế: Mình có 2 con vật: chó, mèo hai con vật này khi nhận được mệnh lệnh  là **"hãy kêu"** thì chó kêu "gâu gâu",  mèo kêu "meo meo".

* **Tính trừu tượng (Abstraction)**Trừu tượng có nghĩ là tổng quát hóa một cái gì đó lên, không cần chú ý chi tiết bên trong. Nó không màng đến chi tiết bên trong là gì và người ta vẫn hiểu nó mỗi khi nghe về nó.