**1 Các câu lệnh để submit 1 file README.MD lên github là gì? Giải thích tác dụng của từng câu lệnh**

Git init: khởi tạo local repository

Git clone: sao chép 1 remote repository về local

Git pull = git fetch + git merge

Git fetch: dùng để cập nhật sự thay đổi trên remote repository về local repository

Git merge: merge repository chỉ định vào local repository

Git status: kiểm tra xem thử mình đã thay đổi file nào so với local repository

**2 "Nêu các thẻ để tạo danh sách?**

=> Thẻ ul, ol và li

Sự khác nhau giữa <ul> và <ol>?"

=> ul: unorder list, không có đánh số thứ tự

ol: order list, có đánh thứ tự

**3 Các thẻ tạo nên 1 table? Phân biệt th và td?**

=> th, td, tr, table

=> th: dùng cho tiêu đề, có thuộc tính css in đậm và canh giữa. td: dùng cho dữ liệu, tr: tạo dòng

**4 Phân biệt innerHTML và innerText?**

innerHTML: in ra màn hình theo cấu trúc HTML

innerText: in ra tất cả các ký tự

Ví dụ: truyền vào chuỗi sau: “<b>In hoa</b>”

innerHTML: **In hoa**

innerText: <b>In hoa</b>

**5 Phân biệt get và post.**

|  |  |
| --- | --- |
| Get | Post |
| Hiển thị thông tin trên url | Không hiển thị thông tin lên url |
| Không bảo mật | Có mã hóa và bảo mật thông tin |
| Hạn chế số lượng ký tự trên url (2048 ký tự) | Không hạn chế số lượng ký tự |
| Không thể truyền dữ liệu dạng binary | Có thể truyền dữ liệu dạng binary (ảnh, phim, file) |
| Có thể access trực tiếp trên thanh address của browser | Không thể access trực tiếp trên thanh address browser mà phải thông qua form hoặc ajax |

**6 Mô tả thuật toán tìm kiếm 1 phần tử trong mảng bằng mã giả?**

**7 Mô tả thuật toán sắp xếp mảng số nguyên bằng mã giả?**

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

let: dùng khai báo biến trong JS, khai báo bằng let thì không thể khai báo lại, có phạm vi sử dụng là block scope (nằm giữa 2 dấu ngoặc nhọn nơi mà nó được khai báo)

var: dùng khai báo biến trong JS, khai báo bằng var thì có thể khai báo lại, được hoisted (được ưu tiên chạy trước) và phạm vi sử dụng là function scope

const: dùng để khai báo hằng trong JS, khai báo hằng phải được gán giá trị vào lúc khai báo và không được gán lại giá trị cho nó, phạm vi sử dụng là block scope

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

String, number, Boolean, null, underfined, object

Dùng type of để xác định biến có kiểu dữ liệu gì

**10 Các cách ép sang kiểu String trong Javascript?**

Cộng thêm chuỗi rỗng, ví dụ: 3 + “” => “3”

**11 Các cách tạo chuỗi chứa dấu nháy?**

Dùng dấu \ trước dấu nháy. Ví dụ: “\”Hello\””

**12 NaN là gì? NaN === NaN có đúng không? Vì sao?**

NaN => Not a number

NaN === NaN => false

**13 Toán tử ba ngôi là gì? Cú pháp?**

Toán tử ba ngôi là loại toán tử có 3 đối số gồm biểu thức điều kiện và 2 giá trị trả về

Cú pháp: điều kiện ? giá trị trả về nếu đúng : giá trị trả về nếu sai

**14 Toán tử == và === khác và giống nhau như thế nào**

Giống nhau: dùng để so sánh dữ liệu

Khác nhau: “==” chỉ so sánh giá trị còn “===” thì so sánh kiểu dữ liệu và giá trị

**15 Phân biệt giữa 2 toán tử && và || trong JavaScript?**

&&: chỉ trả về true nếu tất cả biểu thức đều true

||: trả về true khi 1 trong các biểu thức có giá trị true

**16 Các hàm dùng để tạo thông báo trong Javascript?**

Alert: thông báo ra màn hình

Prompt: nhập dữ liệu vào từ bàn phím (lưu ý dữ liệu nhập vào là chuỗi)

Confirm: xác nhận, hiển thị lựa chọn yes/no

**17 Casting data type trong JS**

Chuyển đổi kiểu dữ liệu.

**18 Có bao nhiêu Statement control trong Java Script?**

If else, switch case.

**19 So sánh sự khác nhau giữa if và switch case?**

**20 Switch case so sánh == hay ===. Đặt ra trường hợp là so sánh bằng thì khi nào sử dụng if bậc thang? Khi nào sử dụng switch case**

Switch case so sánh ===

**21 Các biểu thức và luồng thực thi của for. Nếu thiếu 1 hoặc tất cả các biểu thức thì vòng for sẽ chạy như thế nào?**

3 biểu thức: giá trị khởi tạo, điều kiện tiếp tục vòng lặp, biểu thức tăng giảm.

Nếu thiếu 1 hoặc tất cả thì vòng for vẫn chạy bình thường. Nhưng nếu thiếu biểu thức điều kiện thì vòng lặp for sẽ chạy vô hạn nếu không có lệnh break

**22 Đặt ra 1 bài toán. Xác định bài toán cần sử dụng vòng lặp nào?**

**23 So sánh giống và khác nhau giữa for, while và do...while**

- For

Có 3 biểu thức trong vòng lặp for: biểu thức khởi tạo, biểu thức điều kiện, biểu thức tăng giảm

Nên dùng vòng lặp for khi biết trước số lần lặp của nó

- While và Do... while

Gồm biểu thức điều kiện và khối lệnh lặp

Nên dùng khi chưa biết trước số lần lặp của nó

Khác nhau giữa while và do while:

+ while: đánh giá biểu biểu thức điều kiện sau đó mới lặp => có thể không thực hiện khối lệnh 1 lần nào cả

+ do while: thực hiện rồi mới đánh giá biểu thức điều kiện lặp => ít nhất thực hiện khối lệnh 1 lần

**24 So sánh null và rỗng**

- Null: chỉ ra rằng biến này không chứa bất kỳ giá trị nào

- Underfined: một biến chưa được gán giá trị có giá trị underfined

**25 Đặc điểm mảng một chiều trong Javascript**

+ Mảng là 1 loại biến đặc biệt, có thể lưu nhiều giá trị thay vì chỉ một giá trị như biến thông thường

+ Mỗi giá trị trong mảng thì sẽ được gọi là 1 phần tử

+ Các phần tử trong mảng được lưu trữ ở các vị trí kế tiếp nhau trong bộ nhớ

**26 Các cách khởi tạo một mảng kiểu String trong JavaScript**

Gán bằng 1 mảng đã tồn tại (dùng dấu ngoặc vuông)

Dùng toán tử new để khởi tạo mảng, sau đó dùng vòng for để gán giá trị cho từng phần tử trong mảng

**27 - 1 số hàm thao tác với mảng?**

- Phân biệt push() và unshift()?

+ Hàm push() sẽ thêm một phần tử mới vào phần cuối của mảng.

+ Hàm unshift() sẽ thêm một phần tử mới và phần đầu của mảng.

- Phân biệt push() và pop()?

+ Hàm push() sẽ thêm một phần tử mới vào phần cuối của mảng.

+ Hàm pop() sẽ lấy đi phần tử cuối cùng của mảng.

- Phân biệt shift() và unshift()?

+ Hàm shift() sẽ lấy đi phần tử đầu tiên của mảng.

+ Hàm unshift() sẽ thêm một phần tử mới vào phần đầu của mảng.

**28 Phân biệt tham trị và tham chiếu trong Javascript**

- pass by value (tham trị): sao chép dữ liệu vào hàm => đối số không bị thay đổi khi đi qua hàm (trường dữ liệu truyền vào là kiểu nguyên thủy, ví dụ số ...)

- pass by reference (tham chiếu): sao chép địa chỉ tham chiếu vào hàm => đối số có thể bị thay đổi khi đi qua hàm (trường hợp dữ liệu là object vd: mảng, object)

**29 Phân biệt giữa Hàm có return và hàm không có return**

- 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

- Một hàm không có return thì nó sẽ trả về undefined

**30 So sánh break, continue và return**

**31 Lập trình hướng đối tượng là gì**

Là kỹ thuật hướng các đối tượng trong thực tế vào trong ngôn ngữ lập trình => tất cả mọi thứ trong oop là "đối tượng" 1 chương trình phần mềm được coi như là thế giới bao gồm các đối tượng tương tác lẫn nhau

**Đối tượng bao gồm:**

Thuộc tính: là các dữ liệu, tính chất của đối tượng

Hành vi: là các khả năng, hành động mà đối tượng có thể thực hiện

- Đối tượng là đối tượng có những thuộc tính và những hành vi riêng nhằm phân biệt đối tượng này với các đối tượng khác

- Đối tượng là 1 thể hiện cụ thể của lớp

- Lớp: là kn dùng để mô tả 1 nhóm các đối tượng có những thuộc tính và hành vi tương tự nhau

- Lớp là khuôn mẫu của 1 nhóm các đối tượng

- Thuộc tính là những đối tượng của 1 đối tượng, thể hiện thông qua những giá trị cụ thể

- Hành vi là những cách thức mà qua đó đối tượng thể hiện sự hoạt động hay chức năng của chúng

**32 Các đặc điểm trong Lập trình hướng đối tượng. Đưa ra 1 ví dụ 1 trong 4 tính chất.**

- Tính bao đóng (encapsulation): khả năng truy suất vào các thành phần của một đối tượng trong khi vẫn đảm bảo tính che giấu các đặc tính riêng tư bên trong đối tượng.

- Kế thừa (inhertitance): cho phép các đối tượng có thể chia sẻ hay mở rộng các thuộc tính hoặc phương thức mà không cần phải tiến hành định nghĩa lại.

- Trừu tượng (abstraction): Loại bỏ những thuộc tính và hành vi không quan trọng của đối tượng, chỉ giữ lại những thuộc tính và hành vi liên quan tới vấn đề đang giải quyết.

- Đa hình (polymorphism): thể hiện khi với cùng một phương thức nhưng có thể có cách ứng xử khác nhau ở những lớp cùng giao diện

**33 Constructor là gì? Trong 1 class có nhiều hơn 1 contrustor được hay không?**

Constructor là một phương thức đặc biệt, nó được dùng để khởi tạo và trả về đối tượng của lớp mà nó được định nghĩa

**34 Những phương thức nào cho phép tương tác với chuỗi**

Replace, charAt, substring, concat, length...

**35 Các câu lệnh để vẽ 1 hình tròn. Giải thích từng câu lệnh?**

**36 Trình bày ý tưởng của 1 thuật toán bất kì?**

**37 Cho 1 bài toán liên quan đến ++ trước và ++ sau. Yêu cầu dự đoán kết quả, và tại sao lại có kết quả như vậy?**

**38 Các loại toán tử trong js? Cho biết độ ưu tiên của các toán tử trong một biểu thức?**

X = 1

Y = x++ + ++x + ++x + x-- - x++

x = x++ - --x + x-- - ++x + x++