**CÂU HỎI AUDIT MODULE 1**

**BOOTCAMP PREPARATION 2.0**

|  |  |
| --- | --- |
| **STT** | **Câu hỏi** |
| 1 | Các câu lệnh để submit 1 file README.MD lên github là gì? Giải thích tác dụng của từng câu lệnh |
| 2 | Nêu các thẻ để tạo danh sách ? Sự khác nhau giữa <ul> và <ol> ? |
| 3 | Các thẻ tạo nên 1 table? Phân biệt th và td? |
| 4 | Phân biệt innerHTML và innerText? |
| 5 | Phân biệt get và post. |
| 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? |
| 9 | Trong Javascript có bao nhiêu loại kiểu dữ liệu ?. Làm thế nào để xác định được biến có kiểu dữ liệu gì? |
| 10 | Các cách ép sang kiểu String trong Javascript? |
| 11 | Các cách tạo chuỗi chứa dấu nháy ? |
| 12 | NaN là gì? NaN === NaN có đúng không?Vì sao? |
| 13 | Toán tử ba ngôi là gì? Cú pháp? |
| 14 | Toán tử == và === khác và giống nhau như thế nào |
| 15 | Phân biệt giữa 2 toán tử && và || trong JavaScript? |
| 16 | Các hàm dùng để tạo thông báo trong Javascript ? |
| 17 | Casting data type trong JS |
| 18 | Có bao nhiêu Statement control trong Java Script ? |
| 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 |
| 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? |
| 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 |
| 24 | So sánh null và rỗng |
| 25 | Đặc điểm mảng một chiều trong Javascript |
| 26 | Các cách khởi tạo một mảng kiểu String trong JavaScript |
| 27 | 1 số hàm thao tác với mảng?  - Phân biệt push() và unshift()?  - Phân biệt push() và pop()? - Phân biệt shift() và unshift()? |
| 28 | Phân biệt tham trị và tham chiếu trong Javascript   * Tham trị và tham chiếu dùng để truyền tham số và hàm * Tham trị (Primitives values) là các kiểu dữ liệu có sẵn ví dụ: kiểu số nguyên, kiểu số thực, kiểu ký tự, boolean. * Giá trị của biến trước và sau khi gọi hàm không thay đổi   Tham chiếu:   * Áp dụng cho kiểu dữ liệu object, array * Giá trị của biến trước và sau khi gọi hàm có thể thay đổi   Ngoài ra, khi truyền tham chiếu thới hàm, hàm đó không làm thay đổi tham chiếu để trỏ tới một đối tượng khác.  Tất cả các tham số truyền vào hàm trong JavaScript mặc định là truyền bằng giá trí chứ không phải truyền bằng tham chiếu. |
| 29 | Phân biệt giữa Hàm có return và hàm không 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. |
| 30 | So sánh break, continue và return  Break: Dừng vòng lặp  Continue: Bỏ qua vòng lặp hiện tại đến vòng lặp tiếp theo |
| 31 | Lập trình hướng đối tượng là gì  OOP (viết tắt của Object Oriented Programming) – lập trình hướng đối tượng là một phương pháp lập trình dựa trên khái niệm về lớp và đối tượng. OOP tập trung vào các đối tượng thao tác hơn là logic để thao tác chúng, giúp code dễ quản lý, tái sử dụng được và dễ bảo trì.  Nhiệm vụ của OOP là ánh xạ các đối tượng đưa vào ngôn ngữ lập trình |
| 32 | Các đặc điểm trong Lập trình hướng đối tượng. Đưa ra 1 ví dụ 1 trong 4 tính chất.  **Tính đóng gói (Encapsulation)**  Tính đóng gói cho phép che giấu thông tin và những tính chất xử lý bên trong của đối tượng. Các đối tượng khác không thể tác động trực tiếp đến dữ liệu bên trong và làm thay đổi trạng thái của đối tượng mà bắt buộc phải thông qua các phương thức công khai do đối tượng đó cung cấp.  Tính chất này giúp tăng tính bảo mật cho đối tượng và tránh tình trạng dữ liệu bị hư hỏng ngoài ý muốn.  **Tính kế thừa (Inheritance)**   * Tính kế thừa cho phép xây dựng một lớp mới (lớp Con), kế thừa và tái sử dụng các thuộc tính, phương thức dựa trên lớp cũ (lớp Cha) đã có trước đó. * Các lớp Con kế thừa toàn bộ thành phần của lớp Cha và không cần phải định nghĩa lại. Lớp Con có thể mở rộng các thành phần kế thừa hoặc bổ sung những thành phần mới.   **Ví dụ:** Lớp Cha là smartphone, có các thuộc tính: màu sắc, bộ nhớ, hệ điều hành… Các lớp Con là iPhone, Samsung, Oppo cũng có các thuộc tính: màu sắc, bộ nhớ, hệ điều hành…  **Tính đa hình (Polymorphism)**   * Tính đa hình trong lập trình OOP cho phép các đối tượng khác nhau thực thi chức năng giống nhau theo những cách khác nhau.   **Ví dụ:** Chó và mèo cùng nghe mệnh lệnh “kêu đi” từ người chủ. Chó sẽ “gâu gâu” còn mèo lại kêu “meo meo”.  **Tính trừu tượng (Abstraction)**  Tính trừu tượng giúp loại bỏ những thứ phức tạp, không cần thiết của đối tượng và chỉ tập trung vào những gì cốt lõi, quan trọng.  **Ví dụ:** Quản lý nhân viên thì chỉ cần quan tâm đến những thông tin như:   * Họ tên * Ngày sinh * Giới tính |
| 33 | Constructor là gì? Trong 1 class có nhiều hơn 1 contrustor được hay không?  Constructor là từ khóa dùng để khai báo hàm khởi tạo đối tượng, mỗi class trong JS chỉ có một constructor.  Từ khóa “new” dùng để tạo đối tượng mới.  Từ khóa “this” đại diện cho đối tượng hiện tại. |
| 34 | Những phương thức nào cho phép tương tác với chuỗi |
| 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?  Số học => So sánh => Logic => Gán |