**CÂU HỎI AUDIT MODULE 1**

**BOOTCAMP PREPARATION 2.0**

|  |  |
| --- | --- |
| **STT** | **Câu hỏi** |
| 1 | Repository là gì? có bao nhiêu loại repository?  Repository là kho chứa mã nguồn (file, lịch sử thay đổi trong quá trình thực hiện dự án,…)  Có 2 loại Repo:  + Remote Repo: nằm trên máy chủ chia sẻ  + Local Repo: nằm trên máy cá nhân của lập trình viên | |
| 2 | Các cấu lệnh căn bản để làm việc với git?  git add . : đánh dấu để chuẩn bị các nội dung cho commit  git commit: tạo snapshot của các sự thay đổi trong thư mục  git push: cập nhật các thay đổi từ Local Repo lên Remote Repo  git clone: sao chép file, thư mục từ Remote Repo về Local Repo  git pull: cập nhật các thay đổi từ Remote Repo về Local Repo | |
| 3 | Giải thuật là gì? các cách để biểu diễn thuật toán?  Giải thuật (thuật toán): các bước để giải quyết 1 bài toán  2 cách để biểu diễn:  + Mã giả (pseudo code): không phụ thuộc vào cú pháp, gần gũi với ngôn ngữ tự nhiên, không cần biết ngôn ngữ lập trình cụ thể  +Lưu đồ (flowchart): có tính chặt chẽ, quy ước rõ ràng với tất cả mọi người, có thể biễu diễn rõ ràng luồng thực thi của chương trình. | |
| 4 | Mô tả thuật toán tìm kiếm 1 phần tử trong mảng bằng mã giả?  Mã giả:  1. Khởi tạo biến index = -1 để lưu trữ chỉ số của phần tử cần tìm kiếm.  2. Duyệt qua từng phần tử trong mảng arr bắt đầu từ phần tử đầu tiên (index 0):  a. Nếu phần tử hiện tại là phần tử cần tìm kiếm x:  i. Gán index bằng chỉ số của phần tử hiện tại.  ii. Thoát khỏi vòng lặp vì đã tìm thấy phần tử x.  b. Nếu không, tiếp tục duyệt qua phần tử tiếp theo.  3. Kiểm tra giá trị của biến index:  a. Nếu index vẫn bằng -1, tức là không tìm thấy phần tử x trong mảng arr, trả về -1.  b. Ngược lại, trả về index là chỉ số của phần tử x trong mảng arr. | |
| 5 | Mô tả thuật toán sắp xếp mảng số nguyên bằng mã giả?  1. Đặt biến n là độ dài của mảng arr.  2. Bắt đầu vòng lặp ngoài, lặp i từ 0 đến n-1:  a. Bắt đầu vòng lặp trong, lặp j từ 0 đến n-i-1:  i. So sánh arr[j] và arr[j+1].  ii. Nếu arr[j] lớn hơn arr[j+1], hoán đổi giá trị của arr[j] và arr[j+1].  b. Kết thúc vòng lặp trong.  3. Kết thúc vòng lặp ngoài. | |
| 6 | Trình bày ý tưởng của 1 thuật toán bất kì? | |
| 7 | Nêu các thẻ để tạo danh sách ? Sự khác nhau giữa <ul> và <ol> ?  <ul></ul>; <ol></ol>; <li></li>  Khác nhau:  + <ul>: danh sách không thứ tự  + <ol>: danh sách có thứ tự | |
| 8 | Các thẻ tạo nên 1 table? Phân biệt th và td?  <table> </table>: thẻ tạo bảng  <th> </th>: ô tiêu đề (dùng để nhập tiêu đề, in đậm, căn giữa)  <tr> </tr>: dòng  <td> </td> ô dữ liệu (dùng để nhập nội dung) | |
| 9 | Phân biệt innerHTML và innerText?  innerText: truy cập và thay đổi nội dung văn bản của 1 phần tử, không chứa thẻ HTML  innerHTML: truy cập và thay đổi nội dung, cấu trúc của 1 phần tử, có thể chứa các thẻ HTML | |
| 10 | Nêu một số thẻ HTML cơ bản mà bạn biết  <title>; <head>; <body>; <htnl>; <table>; <h1>; <div>; <span>; <tr>; <td>; <ol>; <ul>; <li>;<img src>; <a href>; <br>; <p>; <span>; <form>; <input>;<link>;<style>;<script>; <span>;… | |
| 11 | Các thuộc tính cơ bản của thẻ form  +action: quy định địa chỉ URL dữ liệu được gửi tới  +name: tên của form  +method: cách thức dữ liệu được gửi lên server | |
| 12 | Phân biệt phương thức get và post trong thẻ form.  GET:  Cách hoạt động: Dữ liệu biểu mẫu được gắn vào URL và gửi đến máy chủ. Vì vậy, dữ liệu sẽ hiển thị trực tiếp trong URL.  Sử dụng phù hợp: Phương thức GET thích hợp khi bạn muốn truy xuất dữ liệu từ máy chủ và có thể chia sẻ URL với người dùng. Nó được sử dụng cho các truy vấn không ảnh hưởng đến dữ liệu trên máy chủ, như việc tìm kiếm hoặc lấy dữ liệu từ cơ sở dữ liệu.  POST:  Cách hoạt động: Dữ liệu biểu mẫu được gửi đến máy chủ dưới dạng một gói dữ liệu riêng biệt và không hiển thị trực tiếp trong URL. Thường được sử dụng để gửi dữ liệu nhạy cảm hoặc dữ liệu mà bạn không muốn hiển thị trong URL.  Sử dụng phù hợp: Phương thức POST thích hợp khi bạn muốn thực hiện các thao tác như thêm, cập nhật hoặc xóa dữ liệu trên máy chủ. Nó bảo vệ dữ liệu của bạn khỏi việc hiển thị trực tiếp trong URL và có khả năng gửi nhiều dữ liệu hơn so với GET. | |
| 13 | Cách khai báo biến bằng từ khóa let và var khác nhau như thế nào ?. Phạm vi của biến?  Khai báo biến bằng var:  Phạm vi: Biến được khai báo bằng var có phạm vi toàn cục hoặc phạm vi hàm (function scope). Nghĩa là biến này có thể truy cập từ bất kỳ đâu trong hàm hoặc cả tệp (nếu biến được khai báo ở ngoài hàm).  Hoisting: Biến được khai báo bằng var có hiện tượng "hoisting", nghĩa là biến có thể được truy cập trước khi nó được khai báo trong mã. Tuy nhiên, giá trị của biến sẽ là undefined cho đến khi nó được gán giá trị.  Khả năng tái khai báo: Biến có thể được tái khai báo bằng var trong cùng phạm vi.  Khai báo biến bằng let:  Phạm vi: Biến được khai báo bằng let có phạm vi khối (block scope), nghĩa là nó chỉ có thể truy cập trong cùng một khối mã (ví dụ: một khối if, for, hoặc một hàm).  Hoisting: Biến được khai báo bằng let cũng có hiện tượng hoisting, nhưng chúng sẽ không được khởi tạo với giá trị undefined. Thay vào đó, chúng sẽ ở trạng thái "temporal dead zone" (vùng chết thời gian) cho đến khi chúng được khởi tạo.  Khả năng tái khai báo: Biến không thể tái khai báo bằng let trong cùng một phạm vi. | |
| 14 | Trong JavaScript có bao nhiêu loại kiểu dữ liệu ?. Làm thế nào để xác định được biến có kiểu dữ liệu gì?  Number: Kiểu số, bao gồm cả số nguyên và số thực.  String: Kiểu chuỗi, dùng để lưu trữ văn bản.  Boolean: Kiểu boolean, chỉ có hai giá trị là true hoặc false.  Undefined: Biến chưa được khởi tạo hoặc không có giá trị. Giá trị mặc định của biến chưa được khai báo là undefined.  Null: Biến có giá trị null, thường được sử dụng để chỉ định rằng biến không có giá trị hoặc không tồn tại.  Object: Kiểu dữ liệu đối tượng, được sử dụng để đại diện cho đối tượng hoặc cấu trúc dữ liệu phức tạp.  Sử dụng Typeof | |
| 15 | Các cách tạo chuỗi chứa dấu nháy ? | |
| 16 | NaN là gì? NaN === NaN có đúng không? Vì sao?  NaN: not a number | |
| 17 | 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?  ++ trước: đặt trước biến, giá trị của biến sẽ tăng lên 1 đơn vị trước khi biểu thức khác được thực hiện, kết quả của biểu thức là giá trị của biến sau khi đã tăng lên  ++ sau: đặt sau biến, giá trị của biến sẽ tăng lên 1 đơn vị sau khi biểu thức khác được thực hiện, kết quả của biểu thức là giá trị ban đầu của biến trước khi tăng lên | |
| 18 | Các loại toán tử trong js? Cho biết độ ưu tiên của các toán tử trong một biểu thức?  Toán tử số học: +; -; \*; /; %; \*\*  Toán tử gán: gán(=); cộng và gán (+=); trừ và gán (-=); \*=; /=; %=  Toán tử so sánh: so sánh giá trị (==), so sánh giá trị và kiểm tra kiểu dữ liệu (===); không bằng (!=;!==); >;<;)=;<=  Toán tử logic: &&; ||; ! (phủ định)  Toán tử chuỗi: + (nối chuỗi)  Toán tử ba ngôi: condition? exp1: exp2  Toán tử type of: trả về kiểu dữ liệu  Độ ưu tiên của toán tử:  () (Ngoặc đơn)  \*\* (Lũy thừa)  \*, /, % (Nhân, chia, lấy phần dư)  +, - (Cộng, trừ)  + (Nối chuỗi)  <, >, <=, >= (So sánh)  ==, !=, ===, !== (So sánh giá trị)  && (AND logic)  || (OR logic)  ? : (Ternary Operator)  =, +=, -= và các toán tử gán khác | |
| 19 | Toán tử ba ngôi là gì? Cú pháp?  Là toán tử điều kiện có chức năng tạo một biểu thức điều kiện ngắn gọn dựa trên 1 điều kiện cho trước  Cú pháp: condition ? expression1 : expression2  Khi biểu thức điều kiện đúng sẽ trả về expression1, sai trả về expression2 | |
| 20 | So sánh null và rỗng | |
| 21 | Toán tử == và === khác và giống nhau như thế nào  ==: so sánh giá trị  ===: so sánh giá trị và kiểu dữ liệu | |
| 22 | Phân biệt giữa 2 toán tử && và || trong JavaScript? &&: đáp ứng đủ tất cả điều kiện đưa ra ||: đáp ứng đủ một trong các điều kiện đưa ra | |
| 23 | Các hàm dùng để tạo thông báo trong JavaScript ?  alert(); confirm(); prompt(); console.log(); | |
| 24 | Có bao nhiêu Statement control(câu lệnh điều khiển) trong Java Script ? | |
| 25 | So sánh sự khác nhau giữa if và switch case?. | |
| 26 | 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 | |
| 27 | 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? | |
| 28 | Đặt ra 1 bài toán. Xác định bài toán cần sử dụng vòng lặp nào? | |
| 29 | So sánh giống và khác nhau giữa for, while và do…while | |
| 30 | So sánh break và continue | |
| 31 | Đặc điểm mảng một chiều trong JavaScript | |
| 32 | Các cách khởi tạo một mảng kiểu String trong JavaScript | |
| 33 | - 1 số hàm thao tác với mảng? - Phân biệt push() và unshift()? - Phân biệt push() và pop()? - Phân biệt shift() và unshift()? | |
| 34 | Phân biệt tham trị và tham chiếu trong JavaScript | |
| 35 | Phân biệt giữa Hàm có return và hàm không có return | |
| 36 | Lập trình hướng đối tượng là gì | |
| 37 | Phân biệt class và object | |
| 38 | Các đặc điểm trong Lập trình hướng đối tượng. Đưa ra 1 ví dụ 1 trong 4 tính chất. (lưu ý cần phải trình bày được 4 tính chất bằng tiếng anh) | |
| 39 | Constructor là gì? Trong 1 class có nhiều hơn 1 constructor được hay không? | |
| 40 | Những phương thức nào cho phép tương tác với chuỗi | |
| 41 | Các câu lệnh để vẽ 1 hình tròn. Giải thích từng câu lệnh? | |