**CÂU HỎI AUDIT MODULE 1**

**BOOTCAMP PREPARATION 2.0**

|  |  |  |
| --- | --- | --- |
| **STT** | **Câu hỏi** | |
| 1 | | Repository là gì? có bao nhiêu loại repository? | |
|  | | Repository là nơi chứa toàn bộ mã nguồn. Repository 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 LTV  Remote repository: ở trên máy chủ chia sẻ (chẳng hạn như GitHub) | |
| 2 | | Các cấu lệnh căn bản để làm việc với git? | |
|  | | Git clone: sao chép repository từ remote về local  Git init: Khởi tạo một repository  Git add: Đưa các file vào vùng staged  Git commit: Ghi nhận các thay đổi  Git push: Đưa các thay đổi từ local repository lên remote repository  Git pull: Cập nhật thay đổi từ remote -> local  Git status: Kiểm tra trạng thái thay đổi | |
| 3 | | Giải thuật là gì? các cách để biểu diễn thuật toán? | |
|  | | Thuật toán/giải thuật bao gồm các chỉ thị (bước) để giải quyết một vấn đề  Có thể sử dụng các cách khác nhau để biểu diễn thuật toán:  Mã giả (Pseudo-code)  Lưu đồ (Flowchart)  Ngôn ngữ lập trình | |
| 4 | | Mô tả thuật toán tìm kiếm 1 phần tử trong mảng bằng mã giả? | |
|  | | Begin  Input a  for (let i=0; i<arr.length<i++)  if (a==arr[i])  Display “Phần tử a nằm trong mảng”  Else  Display “Phần tử a không nằm trong mảng”  End. | |
| 5 | | Mô tả thuật toán sắp xếp mảng số nguyên bằng mã giả? | |
|  | |  | |
| 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> ? | |
|  | | Các thẻ tạo danh sách: <dl>,<dt>,<dd>,<ul>,<li>,<ol>  Sự khác nhau:  <ul> : tạo danh sách không có thứ tự  <ol> : tạo danh sách theo thứ tự | |
| 8 | | Các thẻ tạo nên 1 table? Phân biệt th và td? | |
|  | | Các thẻ tạo table: <table>,<tr>,<th>,<td>  Phân biệt <th>,<td>:  <th>: dùng để tạo tiêu đề (thuộc tính: căn giữa và in đậm)  <td>: dùng để nhập giá trị | |
| 9 | | Phân biệt innerHTML và innerText? | |
|  | | innerHTML: hiển thị nội dung và định dạng thẻ HTML  innerText: chỉ hiển thị nội dung text của thẻ | |
| 10 | | Nêu một số thẻ HTML cơ bản mà bạn biết | |
|  | | <h1>,<p>,<table>,<ul>,<li>,<ol>,… | |
| 11 | | Các thuộc tính cơ bản của thẻ form | |
|  | | Bao gồm 2 thuộc tính cơ bản: action và method  Action: quy định nơi mà dữ liệu gửi đến  Method: cách thức mà dữ liệu được gửi đi | |
| 12 | | Phân biệt phương thức get và post trong thẻ form. | |
|  | | GET: không bảo mật bằng POST  GET: gửi dữ liệu nhanh hơn POST  GET chỉ có thể gửi dữ liệu kiểu text, còn POST có thể gửi đi theo nhiều kiểu file như mp3, video, hình ảnh,…  GET được lưu trữ trong history của trình duyệt và bookmark, POST thì không | |
| 13 | | Cách khai báo biến bằng từ khóa let và var khác nhau như thế nào ?. Phạm vi của biến? | |
|  | | Var được khai báo lại tên biến còn let thì không. Khi sử dụng let thì chỉ được gán lại giá trị mới cho biến chứ không được khai báo lại biến đã tồn tại  Var có cơ chế hoisting, cho phép dòng khai báo được đưa lên đầu tài liệu nên có thể xem như sử dụng trước khi khai báo  Phạm vi của biến:  Var có scope là function còn let là block scope  Const là hàm số đặc biệt dung để khai báo hằng số. Khi khai báo hàm const phải có giá trị cụ thể. Chỉ có hiệu lực trong blockScope | |
| 14 | | Trong Javascript có bao nhiêu loại kiểu dữ liệu? Làm thế nào để xác định được biến có kiểu dữ liệu gì? | |
|  | | Trong JS, có 2 kiểu dữ liệu: Kiểu nguyên thủy và kiểu object  Sử dụng hàm typeof để xác định biến có kiểu dữ liệu gì. | |
| 15 | | Các cách tạo chuỗi chứa dấu nháy? | |
|  | | Sử dụng một dấu nháy (‘) khi muốn hiển thị hai dấu nháy (“) và ngược lại | |
| 16 | | NaN là gì? NaN === NaN có đúng không?Vì sao? | |
|  | | NaN: Not a Number  NaN == NaN sẽ trả về False. Xảy ra khi không thể chuyển dữ liệu về dạng số  Vd: parseInt(a) = NaN;  Vì a không phải là số nên không thể chuyển về dạng số => xuất hiện NaN | |
| 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? | |
| 18 | | Các loại toán tử trong js? Cho biết độ ưu tiên của các toán tử trong một biểu thức? | |
|  | | Các loại toán tử:  Toán tử Toán học  Toán tử So sánh  Toán tử Logic  Toán tử Tăng giảm  Toán tử Typeof  Toán tử 3 ngôi  Độ ưu tiên:  Toán tử Tăng giảm -> Toán tử Nhân chia phần trăm -> Toán tử Cộng trừ -> Toán tử So sánh -> Toán tử Logic -> Toán tử 3 ngôi -> Toán tử gán | |
| 19 | | Toán tử ba ngôi là gì? Cú pháp? | |
|  | | Toán tử ba ngôi là biểu thức điều kiện đánh giá một biểu thức dựa vào một điều kiện cho trước  Biểu thức điều kiện là một toán tử 3 ngôi  Cú pháp:  Điều kiện ? Lệnh thực thi khi đk đúng : Lệnh thực thi khi đk sai | |
| 20 | | So sánh null và rỗng | |
|  | | Null là object đại diện cho vùng nhớ chưa được cấp phát  Còn rỗng là vùng nhớ được cấp phát rồi nhưng chưa có giá trị | |
| 21 | | Toán tử == và === khác và giống nhau như thế nào | |
|  | | Toán tử ==: chỉ so sánh giá trị  Toán tử ===: so sánh giá trị và kiểu dữ liệu | |
| 22 | | Phân biệt giữa 2 toán tử && và || trong JavaScript? | |
|  | | &&: Thỏa mãn cả 2 biểu thức cùng kiểu dữ liệu  ||: Ngừng đánh giá khi biểu thức đầu tiên đủ điều kiện (cùng kiểu dữ liệu) | |
| 23 | | Các hàm dùng để tạo thông báo trong Javascript? | |
|  | | Các hàm tạo thông báo: alert, confirm | |
| 24 | | Có bao nhiêu Statement control (câu lệnh điều khiển) trong Java Script? | |
|  | | Có 2 loại câu điều kiện:  If khuyết và if đầy đủ | |
| 25 | | So sánh sự khác nhau giữa if và switch case? | |
|  | | |  |  | | --- | --- | | **If** | **Switch-case** | | * Có thể sử dụng để so sánh lớn hơn, nhỏ hơn,.. * Mỗi câu lệnh if có một biểu thức điều kiện với giá trị trả về là TRUE hoặc FALSE * Biểu thức điều kiện cần trả về giá trị kiểu Boolean * Chỉ có một khối lệnh được thực thi nếu điều kiện đúng | * Chỉ có thể sử dụng để so sánh bằng hoặc khác * Tất cả các trường hợp (case) đều so sánh với giá trị của biểu thức điều kiện duy nhất * Biểu thức điều kiện trả về giá trị là kiểu byte, short, chart, int hoặc string * Nếu điều kiện đúng mà không có câu lệnh break thì tất cả các khối lệnh ở phía sau cũng được thực thi | | |
| 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 | |
|  | | Switch case là so sánh === | |
| 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? | |
|  | | Các biểu thức: biểu thức khởi tạo, biểu thức điều kiện, biểu thức sau mỗi lần lặp  Luồng thực thi:  Biểu thức khởi tạo -> biểu thức điều kiện -> lệnh thực thi -> biểu thức sau mỗi lần lặp -> lặp lại  Nếu thiếu 1 hoặc tất cả thì vòng for vẫn thực hiện bình thường và kéo dài vô tận | |
| 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 | |
|  | | For: sử dụng cho cấu trúc lặp xác định  While: sử dụng cho cấu trúc lặp không xác định  Do while: thường sử dung cho các cấu trúc lặp yêu cầu một hành động bắt thuộc thực hiện trước khi kiểm tra | |
| 30 | | So sánh break và continue | |
|  | | Break: dùng để kết thúc vòng lặp  Continue: dùng để bỏ qua lần lặp hiện tại và bắt đầu vòng lặp tiếp theo | |
| 31 | | Đặc điểm mảng một chiều trong Javascript | |
|  | | Các phần tử trong mảng được lưu liên tiếp nhau trong ô nhớ, mỗi phần tử có index, kích thước mảng là không giới hạn | |
| 32 | | Các cách khởi tạo một mảng kiểu String trong JavaScript | |
|  | | Let ten\_mang = “”;  Let ten\_mang = new Array(‘’);  Let ten\_mang = Array(‘’); | |
| 33 | | - 1 số hàm thao tác với mảng? - Phân biệt push() và unshift()? - Phân biệt push() và pop()? - Phân biệt shift() và unshift()? | |
|  | | Một số hàm: push(),pop(),shift(),unshift(),sort(),reverse(),concat(),join(),…  Phân biệt push() và unshift(): push là thêm phần tử vào cuối mảng còn unshift là thêm phần tử vào đầu mảng  Phân biệt push() và pop(): push là thêm phần tử còn pop là xóa phần tử ở cuối mảng  Phân biệt shift() và unshift(): shift là xóa phần tử còn unshift là thêm phần tử ở đầu mảng | |
| 34 | | Phân biệt tham trị và tham chiếu trong Javascript | |
|  | | Tham trị: trong lúc gọi hàm truyền đối số vào thì đối số là giá trị copy. Thông thường là nguyên thuỷe  Tham chiếu: giá trị mà đối số nhận được sẽ được lưu trữ ở bộ nhớ if nên tham chiếu sẽ bị thay đổi khi đối số thay đổi. | |
| 35 | | Phân biệt giữa Hàm có return và hàm không có return | |
|  | | Hàm return sẽ có giá trị trả về và các lệnh sau return sẽ không được thực thi | |
| 36 | | Lập trình hướng đối tượng là gì | |
|  | | Lập trình hướng đối tượng (OOP: Object Oriental Programming) là phương pháp lập trình mô tả các đối tượng trong cuộc sống vào ngôn ngữ lập trình  Tất cả mọi thứ OOP đều quy về đối tượng. Một chương trình phần mềm được xem như là một thế giới bao gồm các đối tượng tương tác với nhau | |
| 37 | | Phân biệt class và object | |
|  | | Class: là một khuôn mẫu mô tả các hành vi, thuộc tính chung của một nhóm các đối tượng  Object: là thể hiện cụ thể của class, có các giá trị cụ thể cho thuộc tính và hành vi mà class quy định | |
| 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 contrustor được hay không? | |
|  | | Constructor là hàm khởi tạo để tạo ra các đối tượng khi người sử dụng từ khóa new ten\_class()  Từ khóa “this” đại diện cho đối lượng đang được tưởng tác. Ngoài ra, còn giúp  Trong 1 class chỉ có thể chứa 1 constructor | |
| 40 | | Những phương thức nào cho phép tương tác với chuỗi | |
|  | | Các phương thức: toLowerCase(), toUpperCase(), substring(), indexOf(), charAt(), replace(), … | |
| 41 | | Các câu lệnh để vẽ 1 hình tròn. Giải thích từng câu lệnh? | |
|  | | <canvas>  Tạo class bao gồm thuộc tính x,y,radius  Tạo function để vẽ hình tròn  Let ctx = document.getElementById(“…”).getContext(“2d”);  Let circle = New Circle (x,y,radius);  ctx.beginPatch();  ctx.stroke(5 tham số);  ctx.arc(circle.x, circle.y, circle.radius, 0, 2 \* Math.PI);  ctx.fill();  Gọi hàm createCircle(); | |