**CÂU HỎI AUDIT MODULE 1**

**BOOTCAMP PREPARATION 2.0**

|  |  |  |
| --- | --- | --- |
| **STT** | **Câu hỏi** | |
| 1 | | Repository là gì? có bao nhiêu loại repository?   * Repository là một nơi chứa các mã nguồn. * Có 2 loại repository là local repository và remote repository * Local repository để chỉ mã nguồn lưu trữ tại máy tính của lập trình viên * Remote repository để chỉ mã nguồn lưu trữ tập trung ở máy chủ | |
| 2 | | Các cấu lệnh căn bản để làm việc với git?   * Các câu lệnh căn bản: * Git init: khởi tạo 1 local repository * Git clone: sao chép 1 repository về máy * Git add . : theo dấu các sự thay đổi trong folder * Git commit: tạo 1 snapshot và ghi nhận các thay đổi * Git push: đưa các thay đổi từ local repository lên remote repository * Git pull: cập nhật mã nguồn từ remote repository -> local repository | |
| 3 | | Giải thuật là gì? các cách để biểu diễn thuật toán?   * Các câu lệnh căn bản: Là một tập hợp hữu hạn các chỉ thị được thực thi theo một thứ tự nhất định để đạt được kết quả mong muốn * Có 2 cách biểu diễn thuật toán: mã giả và lưu đồ * Mã giả là sử dụng các ngôn ngữ tự nhiên để diễn tả các bước thuật toán * Lưu đồ thì ngược lại với mã giả, mô tả thuật toán bằng các sơ đồ với các hình dạng được quy định cụ thể: hình elip là bắt đầu hoặc kết thúc; hình bình hành là input hoặc output dữ liệu. Hình chữ nhật là nhập các phép tính toán. Hình thoi là rẽ nhánh điều kiện | |
| 4 | | Mô tả thuật toán tìm kiếm 1 phần tử trong mảng bằng mã giả?  Begin;  While (i<array.length) do  If array [i] = ‘Minh’;  Count++;  End do;  Display ‘count’  End; | |
| 5 | | Mô tả thuật toán sắp xếp mảng số nguyên bằng mã giả?  Begin;  Let temp;  While (i<array.length) do  If a[i] <a[i+1];  Temp=a[i];  A[i]=a[i+1];  A[i+1]=temp;  End do;  End; | |
| 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: * Ol: ordered list: tạo danh sách có thứ tự (chứa các cặp thẻ li) * UL: unordered list: tạo danh sách không thứ tự (chứa các cặp thẻ li) * Dl: Definition List được dùng để xác định danh sách (có đề mục và phần mô tả đề mục) | |
| 8 | | Các thẻ tạo nên 1 table?   * Các thẻ tạo nên table: table, th, td, tr   Phân biệt th và td?   * TH: là các ô tiêu đề * TD: là các ô bình thường | |
| 9 | | Phân biệt innerHTML và innerText?   * Inner text: là sẽ truy xuất nhưng chỉ in nội dung text cho thẻ đó * Inner HTML: là sẽ truy xuất và in nội dung HTML cho thẻ đó | |
| 10 | | Nêu một số thẻ HTML cơ bản mà bạn biết   * Thẻ h1-h6 là các ô tiêu đề * Thẻ p: thẻ văn bản * Thẻ a: liên kết * Thẻ img: hình ảnh * Thẻ form: cho phép người dung nhập dữ liệu để gởi về server | |
| 11 | | Các thuộc tính cơ bản của thẻ form   * Thuộc tính action: có giá trị là URL (đường dẫn) nơi dữ liệu được gởi đến * Thuộc tính method: xác định phương thức gởi dữ liệu, có 2 phương thức là get và post * Thuộc tính name: quy định tên form | |
| 12 | | Phân biệt phương thức get và post trong thẻ form.   * Get: dung để gởi dữ liệu thong thường (không chưa thong tin nhạy cảm), dữ liệu sẽ hiển thị trên thanh URL * Post: dung để gởi dữ liệu nhạy cảm, dữ liệu không hiển thị trên thanh URL, gởi chậm hơn GET | |
| 13 | | Cách khai báo biến bằng từ khóa let và var khác nhau như thế nào ?. Phạm vi của biến?   * VAR: Có thể sử dụng biến rồi mới khai báo sau (cơ chế hoisting). Var có thể khai báo lại, lần khai báo sau sẽ đè lên lần trước. Function scope: phạm vi sử dụng trong 1 function * Let: Phải khai báo trước khi sử dụng. Không thể khai báo lại. Block scope: Phạm vi sử dụng trong 1 dấu ngoặc nhọn | |
| 14 | | Trong Javascript có bao nhiêu loại kiểu dữ liệu ?. Làm thế nào để xác định được biến có kiểu dữ liệu gì?   * Các kiểu dữ liệu: String, number, Boolean, object, null, undefined * Dùng typeof để xác định biến có chứa kiểu dữ liệu gì | |
| 15 | | Các cách tạo chuỗi chứa dấu nháy ? | |
| 16 | | NaN là gì? NaN === NaN có đúng không?Vì sao?   * NAN: là not a number. Nếu không phải là số thì nó sẽ trả về True, nếu là số thì nó sẽ trả về false * NAN ===NAN là sai, vì có thể 2 cái NAN đó là khác nhau. Ví dụ a= NAN, 3:0=NAN, nhưng a khác 3:0 | |
| 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?  X=3  Y=X++ -4 => Y=-1 và X=4  Y=++X -4=> Y=0 và X=4  ++ sau thì việc tang lên đó sẽ diễn ra sau biểu thức tính toán  ++ trước thì việc tang len đó sẽ diễn ra trước khi tính toán biểu thức | |
| 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 cộng chuỗi * Toán tử so sánh * Toán tử logic * Toán tử gán * Toán tư typeof * Độ ưu tiên như trên | |
| 19 | | Toán tử ba ngôi là gì? Cú pháp?   * Toán tử 3 ngôi là 1 cú pháp điều kiện cho trả về value 1 nếu điều kiện đúng, nếu điều kiện sai trả về value 2 * Cú pháp: (condition) ? value1 : value2 * Ví dụ: Điểm thi >5 ? ‘Qua môn’ : ‘Rớt môn’ | |
| 20 | | So sánh null và rỗng   * Null: không tồn tại * Rỗng là có tồn tại kiểu dữ liệu nhưng không có giá trị (giá trị đang trống) | |
| 21 | | Toán tử == và === khác và giống nhau như thế nào   * == chỉ so sánh về giá trị * === so sánh giá trị và kiểu dữ liệu | |
| 22 | | Phân biệt giữa 2 toán tử && và || trong JavaScript?   * && là và * || là hoặc | |
| 23 | | Các hàm dùng để tạo thông báo trong Javascript ?   * Alert: tạo 1 thông báo * Confirm: tạo 1 thông báo xác nhận: ok (hàm trả về true) cancel (hàm trả về false) * Promt: yêu cầu nhập vào dữ liệu | |
| 24 | | Có bao nhiêu Statement control(câu lệnh điều khiển) trong Java Script ?   * IF, IF else, IF bậc thang, IF ELSE lồng nhau * Switch case | |
| 25 | | So sánh sự khác nhau giữa if và switch case?.   * IF: có thể so sánh >,<,==. Mỗi câu lệnh sẽ có 1 điều kiện. Giá trị trả về là kiểu Boolean. Chi có 1 khối lệnh được thực thi nếu điều kiện đúng * Switch case: chỉ so sánh === hoặc !==. Tất cả trường hợp đều so sánh với 1 điều kiện duy nhất. Giá trị trả về là kiểu Boolean,number hoặc string. Nếu điều kiện đúng mà sau khối lệnh thực thi không có break, thì sẽ thực thi hết toàn bộ khối lệnh phía sau | |
| 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   * Điều kiện switch case so sánh === * Không nên dung Switch case trong trường hợp có so sánh > ,< . Switch case chỉ nên sử dụng trong trường hợp phải so sánh nhiều lần =. Nếu bài toán ít điều kiện nên sử dụng if else | |
| 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?   * Biểu thức for (khởi tạo, điều kiện, tang/giảm) {khối lệnh thực hiện} * Đầu tiên Khởi tao -> so sánh điều kiện -> thực hiện khối lệnh -> tang giảm * Khởi tạo và tang giảm có thể không có tùy trường hợp * Nếu biến đã được khởi tạo trước thì không cần khởi tạo lại * Nếu trong khối lệnh thực hiện có tăng/giảm, thì có thể bỏ tang giảm trong dấu ngoặc tròn của biểu thức | |
| 28 | | Đặt ra 1 bài toán. Xác định bài toán cần sử dụng vòng lặp nào?   * For: biết trước số lần lặp: ví dụ tìm bạn tên Minh trong lớp học codegym, ta đã biết sỉ số lớp nên biết được số lần lặp, nên dung for * While: không biết trước số lần lặp. Ví dụ cho 1 dãy số bất kì, đếm số khác 0. Do không biết được số lần lặp nên dùng while * Do while: Với do-while cho phép chạy biểu thức 1 lần roi mới so sánh điều kiện đúng hay sai, phù hợp với các bài toán cần phải nhập trước 1 giá trị đầu vào. Ví dụ tìm 1 bạn đầu tiên đạt 10 điểm thi. Muốn vậy thì phải nhập điểm thi của 1 bạn đã rồi mới so sánh điều kiện | |
| 29 | | So sánh giống và khác nhau giữa for, while và do..while   * For: biết trước số lần lặp: * While: không biết trước số lần lặp. * Do while: Với do-while cho phép chạy biểu thức 1 lần roi mới so sánh điều kiện vòng lặp * While và do while cần phải khai báo biến trước khi chạy. | |
| 30 | | So sánh break và continue   * Continue: sẽ bỏ qua các câu lệnh bên dưới để đến vòng lặp tiếp theo * Break: được dùng kết thúc vòng lặp hoặc cấu trúc Switch | |
| 31 | | Đặc điểm mảng một chiều trong Javascript   * Mảng là 1 biến đặc biệt có thể lưu trữ được nhiều giá trị * Kiểu dữ liệu của mảng là Array (1 kiểu dữ liệu thuộc object) * Mỗi giá trị của mảng được gọi là một phần tử * Các phẩn tử được lưu trữ ở các vị trí liền kề nhau trong bộ nhớ * Các phần tử được phân biệt bởi index của chúng. Index bắt đầu từ 0 cho đến length-1 | |
| 32 | | Các cách khởi tạo một mảng kiểu String trong JavaScript   * Let array = [] * Let array = newArray () | |
| 33 | | - 1 số hàm thao tác với mảng?   * Push, pop, splice, concat, join - Phân biệt push() và unshift()? * Push: them 1 phần tử vào cuối mảng * Unshift: them 1 phần tử vào đầu mảng - Phân biệt push() và pop()? * Push: them 1 phần tử vào cuối mảng * Pop: xóa 1 phần tử ở cuối mảng - Phân biệt shift() và unshift()? * Shift: xóa 1 phần tử ở đầu mảng * Unshift: them 1 phần tử vào đầu mảng | |
| 34 | | Phân biệt tham trị và tham chiếu trong Javascript   * Tham số là các biến được khai báo ở header của function * Đối số là các giá trị truyền vào các tham số này khi gọi hàm | |
| 35 | | Phân biệt giữa Hàm có return và hàm không có return   * Hàm có return là hàm có dùng lệnh return trong hàm và kết thúc hàm sẽ trả về 1 giá trị * Hàm không có return là hàm không có dùng lệnh return trong hàm | |
| 36 | | Lập trình hướng đối tượng là gì   * Lập trình hướng đối tượng là kĩ thuật lập trình hướng các đối tượng trong thực tế vào lập trình | |
| 37 | | Phân biệt class và object   * Class là tập hợp các đối tượng có các thuộc tính và hành vi tương tự nhau * Object là sự thể hiện cụ thể của class có những thuộc tính và hành vi nhằm phân biệt nó với các đối tượng khác | |
| 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)   * Tính bao đóng (encapsulation) đảm bảo truy xuất vào các thuộc tính, hành vi nhưng vẫn đảm bảo tính che dấu các đặc tính riêng bên trong của đối tượng. Ví dụ: nhân viên phòng sẽ có các thuộc tính tên, tuổi, địa chỉ mail, sdt, trình độ. Đảm bảo có thể truy xuất được các thong tin này, nhưng bình thường vẫn đảm bảo tính bảo mật * Tính kế thừa (inheritance) tính kế thừa thể hiện ở việc class này có thể kế thừa thuộc tính và hành vi của class khác mà không cần phải định nghĩa lại. Ví dụ: Tạo danh sách NCC hóa chất. Tạo danh sách NCC đồ điện, kế thừa các thuộc tính và hành vi của NCC hóa chất * Tính trừa tượng (abstraction):Loại bỏ những thuộc tính và hành vi không cần thiết trong ứng dụng, chỉ giữ lại thuộc tính và hành vi liên quan đến vấn đề giải quyết. Ví dụ: Với 1 NCC thì có rất nhiều thuộc tính như tên cty, địa chỉ, mã số thuế, số nhân viên, tài khoản ngân hàng, doanh thu,…Tuy nhiên với danh sách NCC thì mình chỉ lấy các thong tin liên quan lúc mua hàng là tên cty, địa chỉ, mã số thuế, stk ngân hàng * Tính đa hình (polymorphism) Tính đa hình là cùng 1 phương thức nhưng có thể xử lý theo nhiều cách khác nhau: Ví dụ đi lấy hàng, với NCC ở Đà Nẵng thì đi lấy hàng là lấy xe xuống địa chỉ NCC lấy hàng. Với NCC ở ha nội gởi vào bến xe khách, thì lấy hàng là ra bến xe lấy hàng | |
| 39 | | Constructor là gì? Trong 1 class có nhiều hơn 1 contrustor được hay không?   * Constructor là cấu trúc khởi tạo các thuộc tính của class * Mỗi class chỉ có 1 constructor | |
| 40 | | Những phương thức nào cho phép tương tác với chuỗi   * Split: chia chuỗi thành các phần tử của mảng * Length: đếm độ dài chuỗi * Slice : cắt chuỗi (start, end) * Search * toUpperCase * toLowerCase | |
| 41 | | Các câu lệnh để vẽ 1 hình tròn. Giải thích từng câu lệnh?   * Lấy thẻ xuống: Let ctx =document.getElementById().getContext(‘2d’) * Khai báo đối tượng hình tròn với 3 thông số tọa độ X, Y, bán kính: let circle = New circle (x,y,radius) * Ctx.arc (circle.x,circle.y,circle.radius,0,2\*Math.Pi): vẽ cung tròn * Ctx.fill: để vẽ đường tròn (có thể dùng ctx.stroke) | |