CODEGYM
Raising the bar

CÂU HỎI AUDIT MODULE 1
BOOTCAMP PREPARATION 2.0

STT	Câu hỏi
1	Repository là gì? có bao nhiêu loại repository? - Là nơi lưu trữ mã nguồn, lưu trữ các file, các đoạn mã được viết ra trong suốt quá trình phát triển dự án. - Local Repository: Là nơi lưu trữ mã nguồn trên máy LTV. - Remote Repository: Là nơi lưu trữ mã nguồn ở trên máy chủ chia sẻ.
2	Các cấu lệnh căn bản để làm việc với git? - Git init : khởi tạo 1 Local Repo - Git add : đưa các thay đổi vào vùng theo dõi stage - Git commit : đưa sự thay đổi từ lên Local Repo - Git push : đưa sự thay đổi từ Local lên Remote Repo - Git remote add + URL : kết nối local và remote - Git clone : sao chép mã nguồn từ remote về local - Git satus: - Git pull: lấy file từ Remote về Local
3	Giải thuật là gì? các cách để biểu diễn thuật toán? - Giải thuật(thuật toán): là bao gồm các bước để giải quyết 1 vấn đề. - Các cách biểu diễn thuật toán: • Mã giả (Pseudo code) • Lưu đồ (Flow char) • Ngôn ngữ lập trình.
4	Mô tả thuật toán tìm kiếm 1 phần tử trong mảng bằng mã giả?
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 và ? - Các thẻ tạo danh sách là
8	- Các thẻ tạo nên 1 table? o <table> dùng để tạo 1 bảng o <tr> dùng để tạo 1 hàng o <td> o <th> - Phân biệt th và td? o Thẻ <th> dùng để tạo 1 ô **tiêu đề** bên trong hàng o Thẻ <td> sẽ tạo 1 ô **bình thường** bên trong hàng
9	Phân biệt innerHTML và innerText? - InnerText truy xuất và đặt nội dung của thẻ dưới dạng văn bản thuần túy.

	- innerHTML thì có thể truy xuất và đặt nội dung thẻ ở định dạng HTML.
10	Nêu một số thẻ HTML cơ bản mà bạn biết
11	Các thuộc tính cơ bản của thẻ form - action : đường dẫn của sever để gửi dữ liệu - method : phương thức gửi dữ liệu đi (get và post)
12	Phân biệt phương thức get và post trong thẻ form. - Get : o Dữ liệu gửi lên sever hiện trên URL o Kém bảo mật o Kiểu dữ liệu gửi dưới dạng String và giới hạn dữ liệu gửi đi o Gửi nhanh - Post: o Dữ liệu được gửi ngầm o Bảo mật o Gửi được nhiều dữ liệu (string,hình ảnh, file, …) không giới hạn dữ liệu gửi đi o Gửi chậm so với Get.
13	Cách khai báo biến bằng từ khóa let và var khác nhau như thế nào ?. Phạm vi của biến? - Let: o Không thể khai báo lại cùng tên o Không thể sử dụng được trước khi khai báo (không có cơ chế hosting). o Phạm vi truy cập của biến được khai báo let nằm trong phạm vi block scope{ }. - Var: o Có thể khai báo lại cùng tên. o Có thể sử dụng trước khi khai báo (có cơ chế hosting). o Phạm vi truy cập nằm trong function scpoe (phạm vi toàn cục). ➢ Nên sử dụng từ kháo “let” để khai báo biến (vì “let” có tính chặt chẽ hơn “var”
14	Trong Javascript có bao nhiêu loại kiểu dữ liệu ?. Làm thế nào để xác định được biến có kiểu dữ liệu gì? Có 2 loại kiểu dữ liệu: - Kiểu nguyên thủy: o String o Number o Boolean o Undefined o Null o Symbol - Kiểu đối tượng: o Array o Date

	○ Function ○ Object Sử dụng toán tử "type of" để kiểm tra dữ liệu của 1 biến
15	Các cách tạo chuỗi chứa dấu nháy ? Thêm \+" hoặc " ' "
16	NaN là gì? NaN === NaN có đúng không?Vì sao? - NaN là viết tắt của "Not a Number". Khi một function hoặc operation trong JavaScript không thể trả về một số cụ thể, nó sẽ trả về giá trị NaN thay thế. NaN sẽ được coi như một kiểu Number. - Không thể so sánh NaN với chính nó giống như việc kiểm tra như các giá trị khác trong JavaScript. Cả hai phương pháp operator so sánh bằng, đó là == và === đều trả ra false khi chúng ta kiểm tra giá trị NaN là 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? • Toán tử số học: +, -, / , *, ++, -- • Toán tử so sánh: <,>, ==, ===, >=, <=, !=, !== • Toán tử logic: &&, ||, ! • Toán tử gán: =, +=, -+, /=, %= • Toán tử cộng chuỗi • Toán tử "type of": kiểm tra kiểu dữ liệu của 1 biến • Độ ưu tiên của toán tử: số học -> so sánh -> logic-> gán • Có thể sử dụng dấu "()" để thay đổi độ ưu tiên.
19	Toán tử ba ngôi là gì? Cú pháp? - Là toán tử sử dụng đê chay cho if-else (if đủ) để rút gọn code - Cú pháp: ○ (bt_điều_kiện) ? Khối lênh 1 : Khối lệnh 2 ○ Khối lệnh 1 đc thực thi nếu biểu thức điều kiện đúng ○ Khối lệnh 2 đc thực thi nếu biểu thức điều kiện sai.
20	So sánh null và rỗng False
21	Toán tử == và === khác và giống nhau như thế nào? - Toán tử "==" chỉ so sánh giá trị - Toán tử '===' so sánh cả giá trị và kiểu dữ liệu
22	Phân biệt giữa 2 toán tử && và || trong JavaScript? - Toán tử "&&" là toán tử AND trong logic và có giá trị là True khi cả 2 điều kiện đều True. - Toán tử " || " là toán tử OR trong logic và nó sẽ có giá trị True khi 1 trong điều kiện là True.
23	Các hàm dùng để tạo thông báo trong Javascript ? - Alert ("Hello World")

- Console.log("Hello World")
- Prompt ("Hello World")
- Confirm("1 câu hỏi bất kì")
- Document.write
- Document.getElementById("Id").value
- Document.getElementById("Id").innerText
- Document.getElementById("Id").innerHTML

24

Có bao nhiêu Statement control(câu lệnh điều khiển) trong Java Script ?

Có 3 nhóm câu lệnh điều khiện:

- Cấu trúc điều kiện (if, switch – case)
- Cấu trúc lặp (for, while, do-while)
- Cấu trúc nhảy (break, continue)

25

So sánh sự khác nhau giữa if và switch case?.

If	Switch - case
Sử dụng để so sánh: <, >, <=, >=, ==, ===, !=, !==	Chỉ sử dụng để so sánh: ===
Biểu thức điều kiện trả về giá trị kiểu Boolean (True | False)	Biểu thức điều kiện có thể trả về giá trị (number, string, boolean)
Mỗi khối if chỉ có 1 biểu thức điều kiện trả về giá trị True / False	1 biểu thức điều kiện sẽ được so sánh với tất các case
Mỗi biểu thức điều kiện đúng thì chỉ thực thi 1 câu lệnh	Sẽ thực thi tất cả các case ở phía sau nếu **case đúng** ko có "break"

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 so sánh ===
- Đối với bài toán có 4 case trở lên thì dùng 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?

- Biểu thức khởi tạo
- Biểu thức thức điều kiện
- Biểu thức tăng giảm
- Nếu thiếu 1 biểu thức vòng for sẽ chạy vô hạn

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	While	Do-while
Thường dùng khi biết số lần lặp	Không biết trước số lần lặp	Không biết số lần lặp
Kiểm tra điều khiện trước khi lặp	Kiểm tra điều kiện trước khi lặp	Kiểm tra điều kiện sau khi lặp. Nó sẽ thực hiện lặp ít nhất 1 lần cho dù điều kiện sai

<table>
<tr><td></td><td></td></tr>
<tr><td>30</td><td>So sánh break và continue

<table><tr><th>Continue</th><th>Break</th></tr><tr><td>Khi gặp từ khóa “continue” thì chương trình sẽ bỏ qua tất cả các câu lệnh các câu lệnh dưới trong cùng vòng lặp và nhảy tới vòng lặp tiếp theo</td><td>Khi gặp “break” thì chương trình sẽ thoát khỏi vòng lặp gần nhất</td></tr><tr><td>Không sử dụng trong Switch-case</td><td>Có thể sử dụng trong Switch-case</td></tr></table></td></tr>
<tr><td>31</td><td>Đặc điểm mảng một chiều trong Javascript
- Là 1 biến đặc biệt có thể lưu trữ được nhiều giá trị. Mỗi giá trị trong mảng được gọi là phần tử. Độ dài của mảng là tổng số phần tử trong mảng (size). Các phần tử trong mảng được sắp xếp liền nhau từ index = 0 đến index = length - 1.</td></tr>
<tr><td>32</td><td>Các cách khởi tạo một mảng kiểu String trong JavaScript
- Cách 1: let arr = [value1,value2,…]
- Cách 2: let arr = new Array (value1, value2, …)
- Cach 3: let arr = Array (value1, value2, …)</td></tr>
<tr><td>33</td><td>- 1 số hàm thao tác với mảng?
- Phân biệt push() và unshift()?
- Phân biệt push() và pop()?
- Phân biệt shift() và unshift()?</td></tr>
<tr><td>34</td><td>Phân biệt tham trị và tham chiếu trong Javascript?

<table><tr><th>Tham trị</th><th>Tham chiếu</th></tr><tr><td>Truyền đối số: kiểu nguyên thủy</td><td>Truyền đối số: kiểu đối tượng</td></tr><tr><td>Copy giá trị của biến truyền vào hàm</td><td>Copy địa chỉ của biến truyền vào hàm</td></tr><tr><td>Khi thay đổi trong hàm, ngoài hàm vẫn không thay đổi.</td><td>Khi thay dổi trong hàm, ngoài hàm bị thay đổi.</td></tr></table></td></tr>
<tr><td>35</td><td>Phân biệt giữa Hàm có return và hàm không có return
- Hàm có return thì khi thực hiện xong sẽ có giá trị trả về
- Hàm không có return thì khi thực hiện xong sẽ ko có giá trị trả về (undefined)</td></tr>
<tr><td>36</td><td>Lập trình hướng đối tượng là gì
- OOP (Object Oriented Programming) – lập trình hướng đối tượng là kỹ thuật lập trình ánh xạ các đối tượng ngoài thực tế vào ngôn ngữ lập trình.</td></tr>
</table>

	- Tất cả mọi thứ trong OOP đều là các "đối tượng". Một chương trình phần mềm được xem như là 1 thế giới bao gồm các đối tượng tương tác với nhau.
	- Đối tượng bao gồm:
	  o Thuộc tính: đặc điểm, tính chất của các đối tượng.
	  o Phương thức: các khả năng, hành động mà đối tượng có thể thực hiện.

37	Phân biệt class và object - Class: là 1 khuôn mẫu mô trả đặc điểm và hành vi chung cho 1 nhóm các đối tượng tương đồng nhau. - Object: là 1 thể hiện cụ thể của 1 class, có đặc điểm và hành vi cụ thể.
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 một hàm khởi tạo các thuộc tính bên trong class. - Bên trong 1 các chỉ có 1 contrustor.
40	Những phương thức nào cho phép tương tác với chuỗi - Trim() - Replace() - indexOf() - charAt() - Split() - toUpperCase() - toLowerCase()
41	Các câu lệnh để vẽ 1 hình tròn. Giải thích từng câu lệnh?