**BỘ CÂU HỎI PHỎNG VẤN MODULE 1**

**BOOTCAMP PREPARATION 2.0**

|  |  |
| --- | --- |
| **STT** | **Câu hỏi** |
| 1 | Hệ thống các kiến thức đã học ở Module1 |
| 2 | Giải thuật/Thuật toán là gì? : giải thuật là các bước thứ tự để giải quyết một bài toán. |
| 3 | So sánh việc trình bày giải thuật bằng pseudo-code và flowchart?  + 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, dễ triển khai với những bài tooán đơn giản, dễ hiểu. Khó khăn khi dùng để trình bày những bài toán có độ khó lớn, dài. Do viết bằng ngôn ngữ tự nhiên nên không mang tính đại chúng.  +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, có tính đa dạng, sinh động, thể hiện ràng luồng thực hiện của chương trình. |
| 4 | Tại sao lại cần trình bày giải thuật trước khi bắt tay vào code? : để biết nên làm những gì, code chính xác, bớt tốn thời gian |
| 5 | Tại sao lại nên sử dụng flowchart hơn pseudo-code khi thể hiện giải thuật của bài toán phức tạp? trực quan, dễ hiểu, có quy tắc rõ ràng, chỉ rõ hướng đi của chương trình |
| 6 | Tại sao giải thuật lại cần hữu hạn các bước thực thi? để tránh các bài toán vô hạn |
| 7 | Trình bày ý nghĩa các ký hiệu sử dụng khi vẽ flowchart?  oval: bắt đầu, kết thúc  chữ nhật: mô tả các bước tính toán  elip: các bước nhập xuất dữ liệu  thoi: điều kiện rẽ nhánh  mũi tên: hướng đi, luồng thực hiện của chương trình |
| 8 | Vẽ giải thuật giải phương trình bậc nhất?  BEGIN  INPUT a,b  IF a = 0 THEN  IF b = 0 THEN  PRINT “PT VÔ SỐ NGHIỆM”  ELSE PRINT “PT VÔ NGHIỆM”  END IF  ELSE PRINT “PT CÓ NGHIỆM x = -b/a  END IF  END |
| 9 | Bạn biết gì về VCS – Version Control System (hệ thống quản lý phiên bản)? Là một phần mềm hỗ trợ: khôi phục phiên bản cũ của các file, khôi phục lại phiên bản cũ của toàn bộ dự án, xem các thay đổi đã thực hiện theo thời gian, xem ai là người thay đổi gây nên sự cố, khôi phục lại các file đã bị xóa. |
| 10 | Phân biệt hệ thống quản lý mã nguồn tập trung và hệ thống quản lý mã nguồn phân tán? Hệ thống quản lý mã nguồn tập trung là toàn bộ mã nguồn sẽ được lữu trữ tập trung trên một nơi lưu trữ nhất định. Hệ thống quản lý mã nguồn phân tán thì mã nguồn sẽ được lưu trữ ở nhiều nơi (Remote Repo và Local Repo: có thể có nhiều Local Repo) |
| 11 | Git là gì? Tại sao cần sử dụng git? Git là hệ thống quản lý phiên bản mã nguồn theo hình thức phân tán. Để quản lý mã nguồn và ghi nhận các thay đổi, khôi phục |
| 12 | Git thuộc hệ thống quản lý mã nguồn tập trung hay phân tán? Phân tán |
| 13 | Repository là gì? Phân biệt LocalRepository và RemoteRepository? Là nơi lữu trữ mã nguồn. LocalRepo: là các máy con, có thể có nhiều LocalRepo. RemtoteRepo: là nơi lưu trữ mã nguồn tập trung từ các local và update phiên bản mã nguồn. |
| 14 | Đồng bộ giữa các repository (local và remote) để làm gì? Để lưu trữ, update mã nguồn và kiểm soát sự thay đổi. |
| 15 | Thao tác push của git để làm gì? Cú pháp câu lệnh? Đẩy mã nguồn từ local lên remote. git push |
| 16 | Thao tác pull của git để làm gì? Lấy các thay đổi từ remote về local |
| 17 | Thao tác clone của git để làm gì? Cú pháp câu lệnh? Sao chép Remote về máy để tạo Local |
| 18 | Nêu tên và ý nghĩa các câu lệnh cơ bản đã được học để làm việc với git?  git add . :đánh dấu để chuẩn bị các nội dung cho commit  git commit: tạo snapshot các thay đổi trong thư mục  git push: cập nhật các thay đổi từ LocalRepo lên RemoteRepo  git pull: cập nhật các thay đổi từ RemoteRepo xuống LocalRepo  git clone: sao chép một Repo từ một URL về máy cá nhân  git init: tạo một Repo mới |
| 19 | www là gì? World Wide Web: mạng toàn cầu, là 1 phần của internet, là hệ thống thông tin toàn cầu liên kết các tài liệu và nguồn thông tin trên internet bằng cách sử dụng các siêu liên kết văn bản |
| 20 | HTML là gì? Hyper Text Markup Language: ngôn ngữ đánh dấu siêu văn bản. Thể hiện ở các thẻ đóng/mở. |
| 21 | Nêu tên các thẻ làm việc với văn bản trong tài liệu html?  <p> : định dạng đoạn văn bản  <h1> <h2>...<h6>: tạo tiêu đề với độ ưu tiên khác nhau. h1 cao nhất  <b> : in đậm  <i> : in nghiêng  <u>: gạch chân  <br>: tạo dòng mới  <ul><ol>: tạo danh sách  <li>: tạo các mục trong danh sách  <table><tr><td><th>: tạo bảng |
| 22 | Thẻ div dùng để làm gì? <div> khác gì <span> Dùng để chia các khối. Div và span chứa các câu lệnh bên trong tuy nhiên div chứa khối và độ rộng của khối tùy thuộc vào các câu lệnh ở bên trong, span cũng chứa các câu lệnh tuy nhiên chỉ nằm trong một hàng (chủ yếu thao tác với text (văn bản). |
| 23 | URL là gì? Phân biệt url tuyệt đối và url tương đối?  Là một định dạng chuỗi để xác định một tài nguyên trên internet.  URL tuyệt đối: là một URL hoàn chỉnh và chính xác, bao gồm tất cả các thành phần  URL tương đối: là một URL không bao gồm giao thức và tên miền, thường dùng để liên kết đến các tài nguyên trong cùng 1 trang web |
| 24 | Siêu liên kết là gì?  là một kỹ thuật trong các tài liệu đa phương tiện và trình duyệt web cho phép nhấp vào văn bản, hình ảnh hoặc biểu tượng để chuyển đến 1 trang web, tài liệu hoặc tài nguyên khác có liên quan. thường được tạo bằng cách sử dụng thẻ <a> |
| 25 | Muốn liên kết đến một phần trên cùng trang thì làm thế nào?  sử dụng <a href = “ ”></a> |
| 26 | Muốn liên kết đến một phần trên trang khác thì làm thế nào? |
| 27 | Phân biệt internal link và external link? Internal: link thuộc trang web. External: link ngoài trang web |
| 28 | Có mấy loại danh sách? Có 2 loại: danh sách có đánh thứ tự (<ol>) và không có thứ tự (<ul>) |
| 29 | Muốn hiển thị danh sách có thứ tự bắt đầu C. D. E… thì làm thế nào? Sử dụng phương thức type và start = 3 |
| 30 | Form hay biểu mẫu dùng để làm gì? Dùng để thu thập thông tin và gửi về máy chủ |
| 31 | Thuộc tính action của form dùng để làm gì? Gán địa chỉ nhận thông tin biểu mẫu |
|  |  |
| 32 | Phân biệt 2 method là get và post?  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 |
| 33 | Phân biệt các nút bấm có type là: button, submit, reset? Buttuon: tạo một nút bấm thông thường hoạt động dựa trên các sự kiện. Submit: nút bấm gửi thông tin thu thập từ form. Reset: trả về giá trị trước khi load. |
| 34 | Muốn submit form đến trang CodeGym.vn thì làm thế nào?  <form action = “Codegym.vn”> |
| 35 | Nếu muốn nối 3 ô trên cùng 1 hàng thì làm thế nào? Dùng rowspan |
| 36 | Nếu muốn nối 2 ô trên cùng 1 cột thì làm thế nào? Dùng colspan |
| 37 | Phân biệt thuộc tính cellpadding và thuộc tính cellspacing?  cellpadding: xác định khoảng cách giữa nội dung của một ô và viền của ô đó  cellspacing: xác định khoảng cách giữa các ô trong bảng |
| 38 | Trình bày các cách nhúng javascript vào 1 tài liệu html?  inline: code trực tiếp trong thẻ HTML  internal: code trong cặp thẻ script. thường được đặt ở cuối sourcecode, trước thẻ đóng body  external: nhúng file js bằng thẻ link |
| 39 | Phát biểu: “JavaScript là một ngôn ngữ script dựa trên đối tượng nhằm phát triển các ứng dụng Internet dựa trên client và server” là đúng hay sai? Giải thích? Sai. dựa trên client |
| 40 | Phát biểu “JavaScript phân biệt chữ hoa chữ thường ” đúng hay sai? Đúng |
| 41 | Trình bày về quy tắc PascalCase và camelCase?  PascalCase: tên biến hoặc hàm được bắt đầu bằng một chữ viết hoa và mỗi từ tiếp theo trong tên cũng bắt đầu bằng một chữ viết hoa, các từ được viết liên tiếp với nhau không có nối tiếp bằng dấu gạch  camelCase: tên biến hoặc hàm được bắt đầu bằng một chữ viết thường và mỗi từ tiếp theo trong trong tên được viết hoa, các từ được viết liên tiếp nối nhau không có dấu gạch |
| 42 | Biến là gì? Phân biệt biến cục bộ (local) và biến toàn cục (global)?  Là một tên gọi được gắn cho một vùng nhớ chứa dữ liệu  Biến cục bộ: được khai báo trong phạm vi một hàm hoặc một khối mã cụ thể, chỉ có thể được khi báo và sử dụng trong phạm vi được khai báo và không thể truy cập từ bên ngoài.  Biến toàn cục: có thể truy cập từ bất kỳ đâu trong chương trình, có thể truy cập từ bất kỳ hàm nào. Sử dụng nhiều biến toàn cục làm phức tạp và khó bảo trì mã nguồn |
| 43 | Phân biệt: var, let, const?  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 |
| 44 | Phát biểu “JavaScript là ngôn ngữ định kiểu yếu hay động” nghĩa là gì? Trình bày? |
| 45 | Nêu tên các kiểu dữ liệu nguyên thủy trong JavaScript?  Number, string, boolean, undefined, object, null |
| 46 | Trình bày về toán tử typeof? Khi khai báo “let x=10;” thì x có kiểu là gì?  Toán tử typeof được dùng để lấy về kiểu dữ liệu của 1 biến hoặc giá trị.  number |
| 47 | Trình bày các cách đưa thông báo trong JavaScript?  alert(); prompt(); confirm(); |
| 48 | Trình bày về các loại toán tử trong JavaScript?  Toán tử số học:  Toán tử gán:  Toán tử cộng chuỗi: Toán tử so sánh:  Toán tử logic: |
| 49 | Phân biệt giữa 2 toán tử == và ===?  == So sanh về giá trị  === So sánh về giá trị và kiểu dữ liệu |
| 50 | Phân biệt giữa 2 toán tử && và || trong JavaScript?  && đảm bảo đáp ứng tất cả điều kiện đưa ra là đúng sẽ trả về kết quả đúng  || đảm bảo một trong các điều kiện đưa ra là đúng sẽ trả về kết quả đúng |
| 51 | Kết quả biểu thức +0===-0 là true hay false? true |
| 52 | Trình bày về toán tử ++,-- và – trong JavaScript? |
| 53 | Phân biệt ++a và a++?  ++a: 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  a++: 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 |
| 54 | Câu lệnh điều kiện/rẽ nhánh là gì?  Câu lệnh điều kiện/rẽ nhánh trong lập trình là một cấu trúc kiểm tra một điều kiện nào đó và thực hiện các hành động khác nhau dựa trên kết quả của điều kiện đó. Câu lệnh điều kiện cho phép chương trình thực hiện lựa chọn giữa các khối mã khác nhau dựa trên giá trị của biểu thức điều kiện |
| 55 | Có những loại câu lệnh điều kiện nào?  If-else; switch-case; |
| 56 | Trình bày về câu lệnh if?  Đánh giá 1 điều kiện và đưa ra lựa chọn thực thi 1 khối lệnh dựa trên điều kiện đó. Nếu điều kiện có giá trị đúng thì khối lệnh bên trong if được thực thi, nếu điều kiện sai thì khối lệnh bên trong if được bỏ qua, luồng thực thi sẽ chuyển xuống khối lệnh sau if |
| 57 | Trình bày về câu lệnh switch?  Là 1 cấu trúc điều kiện cho phép lựa chọn thực thi các khối lệnh khác nhau dựa trên kết quả của việc so sánh. Switch-case so sánh giá trị của biến với lần lượt từng giá trị một |
| 58 | Tại sao lại nói else là một tùy chọn (optional) của câu lệnh if?  Có thể có else hoặc không có else cũng không ảnh hưởng đến câu lệnh |
| 59 | Khi nào nên sử dụng câu lệnh switch?  Khi kiểm tra nhiều trường hợp cho cùng một biểu thức. |
| 60 | Câu lệnh break dùng để làm gì? Nếu một case mà không có break thì chương trình chạy thế nào?  Dùng để thoát khỏi vòng lặp hoặc cấu trúc điều kiện 🡺 Dừng thực thi khối mã hiện tại và tiếp tục thực thi ở ngoài khối mã  Sẽ tiếp tục thực hiện các câu lệnh trong các case liên tiếp phía sau |
| 61 | Mệnh đề default có bắt buộc trong câu lệnh switch?  Không bắt buộc |
| 62 | Trình bày về toán tử 3 ngôi?  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 |
| 63 | Trình bày về sự kiện và trình quản lý sự kiện trong javascript? |
| 64 | Sự kiện onchange phát sinh khi nào? |
| 65 | Sự kiện onclick phát sinh khi nào? Khi ấn chuột vào |
| 66 | Muốn hiển thị 1 hộp thoại xác nhận “Bạn chắc chắn xóa không?” với người dùng khi bấm vào nút delete thì làm thế nào? Sử dụng confirm() |
| 67 | Phân biệt onmousevover và onmouseout, onmousemove? |
| 68 | Phân biệt onkeyup, onkeydown và onkeypress? |
| 69 | Phân biệt onfocus và onblur? |
| 70 | Sự kiện onload phát sinh khi nào? |
| 71 | 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 |
| 72 | Nếu muốn vô hiệu hóa (disabled) một button khi click vào 1 checkbox thì làm thế nào? |
| 73 | Phân biệt thuộc tính readonly và disabled của html? |
| 74 | Vòng lặp là gì?  là một cấu trúc điều khiển được sử dụng để lặp lại một khối mã hoặc 1 tập hợp các lệnh nhiều lần cho đến khi 1 điều kiện cụ thể được đáp ứng |
| 75 | Nêu các loại vòng lặp cơ bản trong JavaScript? |
| 76 | Phân biệt vòng lặp for,while,do…while? |
| 77 | Khi nào nên sử dụng vòng lặp do…while? |
| 78 | Khi nào nên sử dụng vòng lặp for? |
| 79 | Khi nào nên sử dụng vòng lặp while? |
| 80 | Yêu cầu người dùng nhập đi nhập lại giá chị của một số cho đến khi số đó là số chẵn thì có nên sử dụng vòng lặp không? Nếu có, nên sử dụng loại vòng lặp nào? |
| 81 | Yêu cầu người dùng tính tổng các số chẵn từ 10 đến 100 thì nên dùng vòng lặp loại nào? |
| 82 | Trình bày về vòng for thiếu? |
| 83 | Nếu muốn hiển thị các cặp số nguyên dương có tổng là 10 thì viết thế nào? |
| 84 | Nếu muốn hiển thị hình tam giác vuông \* thì nên viết thế nào? |
| 85 | Trình bày về cách chạy của vòng for lồng nhau? |
| 86 | Phân biệt giữa câu lệnh break và continue? |
| 87 | Hiển thị các số không chia hết cho 5 từ 1-100 có sử dụng continue thì làm thế nào? |
| 88 | Mảng là gì? |
| 89 | Trình bày cú pháp khai báo mảng trong JavaScript? |
| 90 | Chỉ số của mảng là gì? Chỉ số của mảng xuất phát từ 0 hay 1? |
| 91 | Kích thước của mảng là gì? |
| 92 | Phân biệt push() và unshift()? |
| 93 | Phân biệt push() và pop()? |
| 94 | Phân biệt shift() và unshift()? |
| 95 | Nếu muốn xóa phần tử cuối cùng của mảng dùng hàm nào? |
| 96 | Nếu xóa phần tử đầu tiên của mảng dùng hàm nào? |
| 97 | Muốn sắp xếp mảng thì dùng hàm nào?  sort() |
| 98 | Có một mảng gồm tên của các bạn học viên. Nếu muốn sắp xếp tên này theo thứ tự alphabet làm thế nào? |
| 99 | Thao tác duyệt mảng là gì? Triển khai thao tác bằng code? |
| 100 | Thao tác duyệt mảng 2 chiều? |
| 101 | Mảng 2 chiều cần mấy chỉ số để xác định duy nhất 1 phần tử? |
| 102 | Viết code tính tổng các phần tử của ma trận số (ý tưởng)? |
| 103 | Viết code tính tổng các ô trên cùng hàng của ma trận số (ý tưởng)? |
| 104 | Viết code tính tổng các ô trên cùng cột của ma trận số (ý tưởng)? |
| 105 | Hàm (function) là gì? |
| 106 | Phân biệt giữa tham số hình thức (formal arguments) và tham số thực (actual arguments)? |
| 107 | Gọi hàm (invoke) là gì? |
| 108 | Phân biệt hàm gọi (calling function) và hàm được goi (called function)? |
| 109 | Phân biệt truyền tham trị và truyền tham chiếu? |
| 110 | Câu lệnh return dùng để làm gì? |
| 111 | Có được phép viết “return;”? |
| 112 | Phân biệt return và break? |
| 113 | Hàm thực hiện việc kiểm tra nên trả về giá trị không? Nếu có, nên trả về gì? |
| 114 | Tại sao hàm hoán vị swap(a,b) lại không thực hiện được việc hoán vị của các tham số gốc? |
| 115 | Trình bày về OOP? |
| 116 | Phân biệt giữa lập trình hướng đối tượng và lập trình hướng thủ tục/chức năng? |
| 117 | Phân biệt giữa object và class? |
| 118 | Hai khái niệm “Môn học” và “Môn học JavaScript” thì đâu là object? |
| 119 | Mô tả về sơ đồ lớp? |
| 120 | Vẽ sơ đồ lớp của lớp Học viên? |
| 121 | Thuộc tính là gì? Nên dùng từ loại nào để biểu diễn thuộc tính? |
| 122 | Phương thức là gì? Nên dùng từ loại nào để biểu diễn phương thức? |
| 123 | Cú pháp định nghĩa 1 class trong JavaScript? |
| 124 | Cú pháp tạo 1 đối tượng từ class? |
| 125 | Trình bày một số class có sẵn trong JavaScript mà em biết? |
| 126 | Cú pháp truy cập đến 1 thuộc tính của đối tượng? |
| 127 | Cú pháp truy cập đến 1 phương thức của đối tượng? |
| 128 | Từ khóa this dùng để làm gì? |
| 129 | Trình bày về DOM (Document Object Model – Mô hình đối tượng tài liệu)? |
| 130 | Thẻ <canvas> dùng để làm gì? |
| 131 | Vẽ hình chữ nhật trên canvas? |
| 132 | Vẽ hình tròn trên canvas? |
| 133 | Vẽ text trên canvas? |
| 134 | Nếu muốn di chuyển một hình tròn trên canvas cần làm gì? |