Bộ câu hỏi phỏng vấn MODULE 1

Câu 1: Giải thuật/thuật toán là gì ?

* Là một chuỗi các hành động được thực hiện để giải quyết một vấn đề cụ thể hoặc thực hiện một nhiệm vụ cụ thể
* Các cách khác nhau để mô tả thuật toán là
  + Mã giả (seudo-code)
  + Lưu đồ (flowchart)

Câu 2: So sánh việc trình bày giải thuật bằng pseudo-code và flowchart

|  |  |
| --- | --- |
| Pseudo-code | Flowchart |
| * Ưu điểm : cho phép mô tả thuật toán bằng ngôn ngữ tự nhiên gần gũi, giúp người đọc dễ hiểu, không bị ràng buộc bởi cú pháp hoặc ngôn ngữ lập trình cụ thể, giúp trong việc trình bày ý tưởng của thuật toán. * Nhược điểm : Không có ràng buộc nên dẫn đến các bên liên quan không hiểu được thuật toán của nhau dẫn đến khó hiểu. | * Ưu điểm : Có các quy định cụ thể về các khối trong flowchart nên khi biểu diễn các bên liên quan sẽ hiểu được luồng thực thi và ý nghĩa. * Nhược điểm : vì có các quy định cụ thể nên nhiều lúc gây khó khăn cho 1 số bên. |

Câu 3: Tại sao cần trình bày giải thuật trước khi bắt tay vào code?

* Hiểu rõ vấn đề : hiểu rõ hơn về vấn đề cần giải quyết và cách thức để giải quyết nó
* Phân tích và thiết kế : xác định được cấu trúc và các thành phần cần thiết trong việc triển khai thuật toán
* Giảm thiểu sai sót : phát hiện và sửa các lỗi hoặc sơ xuất một cách dễ dàng, giảm thiểu việc phải sửa lỗi sau khi viết mã
* Dễ dàng truyền đạt được ý tưởng : giúp truyền đạt ý tưởng một cách dễ dàng cho người khác
* Tối ưu hóa và tăng hiệu suất : tìm ra các cách tiếp cận tối ưu hóa và cải thiện hiệu suất trước khi viết mã

Câu 4: Tại sao lại nên sử dụng flowchart hơn pseudo-code khi thể hiện giải thuật của bài toán phức tạp ?

* Minh họa rõ ràng và trực quan : thể hiện rõ các luồng thực thi của bài toán giúp người khác để hình dung hơn
* Giảm sự phúc hợp : có thể chia nhỏ thuật toán thành các phần nhỏ và thể hiện rõ luồng thực thi và chức năng của mỗi phần
* Dễ dàng tìm kiếm và sữa đổi :
* Hỗ trợ trong việc học và giản dạy : giúp dễ hiểu và hiểu được cách thức hoạt động của thuật toán
* Dễ dàng chia sẻ và truyền đạt ý tưởng

Câu 5 : Tại sao giải thuật lại cần hữu hạn các bước thực thi ?

* Thực thế của máy tính : là một thiết bị cần phải thực hiện từng bước một, nên chỉ rõ các bước để máy tính thực hiện theo
* Hiệu suất và tốc độ : thuật toán với số lượng thực hiện lớn có thể tốn kém về thời gian và tài nguyên
* Độ chính xác và dễ kiểm tra :
* Dễ dàng giải quyết vấn đề :
* Trành vòng lặo vô hạn :

Câu 6 : Trình bày ý nghĩa các ký hiệu sử dụng khi vẽ flowchart ?

* Hình elip : đánh dấu điểm bắt đầu và kết thúc của thuật toán
* Hình chữ nhật : đánh dấu các thao tác tính toán
* Hình bình hành : đánh dấu các lệnh nhập xuất dữ liệu
* Hình thoi : đánh dấu các bước cần rẽ nhánh
* Hình tròn : nối các phần trong lưu đồ lại với nhau
* Mũi tên : đánh dấu luồng thực thi của thuật toán

Câu 7: vẽ giải thuật vã phương trình bậc nhất ?

A diagram of a flowchart

Description automatically generated

Câu 8 : Bạn biết gì về VCS – Version Control System (hệ thống quản lý phiên bản)?

* Là một phần mềm hỗ trợ :
  + Khôi phục lại phiên bản cũ của các file
  + Khôi phục lại phiên bản cũ của toàn bộ dự án
  + Xem lại các thay đổi đã được thực hiện theo thời gian
  + Xem ai là người thực hiện thay đổi cuối cùng có thể gây ra sự cố
  + Khôi phục lại các file vô tình xóa mất

Câu 9: Phân biệt hệ thông quản lý mã nguồn tập trung và hệ thông quản lý mã nguồn phân tán :

|  |  |
| --- | --- |
| Quản lý mã nguồn tập trung | Quản lý mã nguồn phân tán |
| * Toàn bộ mã nguồn và lịch sử phiên bản của dự án được lưu trữ tại một máy chủ tập trung * Người dùng phải kết nối trực tiếp đến máy chủ tập trung để thực hiện các thao tác. * Ưu điểm : Dễ dàng quản lý và kiểm soát phiên bản, truy cập đồng nhất vào mã nguồn tùy nơi * Nhược điểm : yêu cầu kết nối internet để thực hiện các thao tác quản lý mã nguồn | * Mỗi máy tính trong hệ thống lưu trữ một bản sao hoàn chỉnh của toàn bộ lịch sử phiên bản và mã nguồn (mỗi máy khách là một hệ thống riêng biệt) * Ưu điểm : linh hoạt và độc lập, cho phép làm việc ngoại tuyến, bảo mật tốt hơn vì mỗi bản sao là độc lập * Nhược điểm : có thể gây ra không nhất quán giữa các bản sao nếu không quản lý cẩn thận, yêu cầu kĩ năng hơn để quản lý đồng thời nhiều phiên bản |

Câu 10: GIT là gì ? Tại sao cần sử dụng Git ?

* Git là một hệ thống điều khiển phiên bản (version control system) theo hình thức phân tán
* Nên sử dụng Git :
  + Quản lý mã nguồn hiệu quả : theo dõi và quản lý lịch sử của mã nguồn một cách hiệu quả, cho phép bạn dễ dàng quay lại phiên bản trước đó
  + Linh hoạt trong phát triển : có thể phát triển các tinh năng mới mà không ảnh hưởng đến mã nguồn chính sau đó hợp nhất chúng một cách dễ dàng bằng 2 phương pháp branching và merging.
  + Làm việc song song : nhiều người có thể làm việc trên cùng một dự án mà không gặp xung đột
  + Cộng đồng lớn : dễ dàng tìm kiếm tài nguyên và trao đổi hỗ trợ
  + Phổ biến và tiêu chuẩn :
  + Miễn phí và mã nguồn mỡ :

Câu 11 : Git thuộc quản lý mã nguồn tập trung hay phân tán

* Git thuộc quản lý mã nguồn phân tán.

Câu 12: Repository là gì ? phân biệt Remote Repository và Local Repository ?

* Repository là một kho chứa dữ liệu, thường được sử dụng để lưu trữ và quản lý mã nguồn của một dự án phần mềm.

|  |  |
| --- | --- |
| Local Repository | Remote Repository |
| * Là phiên bản của Repository mà bạn đang làm việc trên máy tính cá nhân. | * Là phiên bản của Repository được lưu trữ trên một máy chủ từ xa, thường là trên internet hoặc trong một hệ thống mạng nội bộ. |

Câu 13 : Đồng bộ giữa các Repository (local và remote) để làm gì ?

* Chia sẻ và cộng tác : có thể chia sẻ và cộng tác trên một mã nguồn. Cho phép người khác làm việc trên các tính năng mới , sửa lỗi hoặc thực hiện các thay đổi mà không cần phải đợi người khác hoàn thành công việc của họ.
* Sao lưu và bảo vệ dữ liệu : Remote Repository là nơi sử dụng để sao lưu và bảo vệ mã nguồn.
* Quản lý phiên bản : việc đồng bộ Local lên Remote giúp theo dõi lịch sử phiên bản của dự án một cách chính xác.
* Kiểm tra và phê duyệt : kiểm tra trước khi chia sẻ với mọi người

Câu 14 : Thao tác push của Git để làm gì ? cú pháp câu lệnh

* Chia sẻ các thay đổi : chia sẻ mã nguồn của bạn lên Remote Repository để nhóm có thể tiếp tục làm việc trên cùng một phiên bản dự án
* Sao lưu dữ liệu : dữ liệu lưu trữ không chỉ trên máy tính của bạn mà còn trên máy chủ từ xa
* Bảo vệ dữ liệu : tránh mất mát dữ liệu khi máy tính bị mất hoặc gặp sự cố
* Đồng bộ hóa lịch sử phiên bản : giúp các thành viên trong nhóm xem được các thay đổi mới nhất và lịch sử của dự án
* Cú pháp :
  + Git push origin main

Câu 15 : Thao tác pull của git để làm gì ?

* kéo các thay đổi từ Remote Repository về Local của bạn
  + Cập nhật mã nguồn :
  + Đồng bộ hóa lịch sử phiên bản
  + Kiểm tra và đối chiếu thay đổi
  + Hợp nhất (merge) các thay đổi

Câu 16 : Thao tác clone của git để làm gì ? cú pháp câu lệnh

* Sao chép dự án từ Remote Repository
* Chia sẻ và cộng tác
* Đơn giả hóa quá trình bắt đầu
* Đảm bảo tính đồng bộ
* Cú pháp :
  + Git clone <url>

Câu 17 : Nếu tên và ý nghĩa các câu lệnh cơ bản đã được học để làm việc với git

* Git clone : sao chép một Repository
* Git init : khởi tạo một Repository
* Git add : đưa các file vào vùng staged
* Git commit : ghi nhận các thay đổi hoặc gì một message
* Git push : đưa các thay đổi từ local lên remote
* Git pull : kéo các thay đổi từ remote về local

Câu 18 : www là gì ?

* Thường là viết tắt cho “World Wide Web” là hệ thống thông tin toàn cầu dựa trên mậng internet. Nó bao gồm hàng tỷ trang web chưa thông tin da dạng như văn bản, hình ảnh, video, âm thanh và nhiều loại dữ liệu khác.

Câu 19 : HTML là gì ?

* Là viết tắt của từ Hypertext Markup Language ngôn ngữ đánh dấu siêu văn bản để tạo ra các trang web, dùng để định nghĩa cấu trúc và nội dung của một trang web bằng cách sử dụng các thẻ và các phần tử.

Câu 20 : Nêu các thẻ làm việc với văn bản trong HTML ?

* <p></p> : hiển thị dưới dạng một đoạn văn bản độc lập
* <h1> đến <h6> : là các thẻ tiêu đề khac nhau <h1> là lớn nhất và <h6> là bé nhất
* <strong> và <b> : làm cho văn bản in đậm
* <em> và <i> : làm cho văn bản in nghiêng
* <u> : làm cho văn bản được gạch chân

Câu 21: Thẻ div dùng để làm gì ? <div> khác gì với <span> ?

* Thẻ div dùng để tạo ra một khối trên trang web. Thường được sử dụng để nhóm các tổ chức HTML khac lại với nhau, để dễ dàng quản lý và định dạng chúng bằng CSS

|  |  |
| --- | --- |
| DIV | SPAN |
| * Được sử dụng để nhóm một khối lớn trên trang web * Nhóm các phần tử HTML với nhau để dễ dàng định dạng bằng css hoặc thao tác với javaScript * Sẽ xuống dòng mới | * Được sử dụng để định dạng và làm việc với văn bản hoặc một phần nhỏ của nội dung * Dùng để định dạng văn bản hoặc phần nhỏ của văn bản, hoặc để áp dụng kiểu dáng cụ thể của một phần đoạn văn * Không tao ra dòng mới |

Câu 22 : URL là gì ? Phân biệt url tuyệt đối và tương đối ?

* Là viết tắt của “Uniform Resource Locator” (định vị tài nguyên đồng nhất) và là một chuỗi ký tự được sử dụng để chỉ định vị trí của một tài nguyên trên internet. URL thường được sử dụng để truy cập cac trang web, tệp tin, ảnh, video và các tài nguyên khác trên mạng.

|  |  |
| --- | --- |
| URL tuyệt đối | URL tương đối |
| * Bao gồm toàn bộ thông tin cần thiết để xác định vị trí chính xác của một tài nguyên trên internet * Bắt đầu bằng giao thức http:// hoặc https://, theo sau là tên miền hoặc địa chỉ IP của máy chủ, và cuối cùng là đường dẫn đến tài nguyên * Ví dụ : https://www.example.com/images/logo.png. | * Chỉ định vị trí của một tài nguyên dựa trên vị trí hiện tại của trang web tài nguyên * Nó không bao gồm giao thức hay tên miền, nhưng chỉ cung cấp đường dẫn đến tài nguyên trong cùng một trang web hoặc thư mục * Ví dụ :   images/logo.png hoặc ../images/logo.png. |

Câu 23 : Siêu lên kết là gì ?

* Được sử dụng để tạo ra các liên kết giữa các trang web và các tài nguyên trên internet. Khi một người dùng nhân vào một siêu liên kết, trình duyệt sẽ chuyển đến địa chỉ được chỉ định trong siêu liên kết
* Được tạo bằng thẻ <a> href=”link liên kết” taget=”phương thức mở”

Câu 24 : Muốn liên kết đến một phần trên cùng trang thì làm thế nào?

* Tạo thẻ HTML với thuộc tính “id”
  + <h1 id = “name” >Tiêu đề</h1>
  + <a href = “#name” >Về đầu trang</a>

Câu 25 : Muốn liên kết đến một phần trên trang khác thì làm thế nào ?

* <a href="https://example.com/page#top">Liên kết đến phần trên cùng của trang khác</a>
* Đưa trình duyệt đến phần có ID là “top”

Câu 26 : Phân biệt internal link và external link ?

* Internal link : liên kết từ trang chủ của một trang web đến trang sản phẩm cụ thể cùng trang web đó
* External link : liên kết từ trang web của một công ty đến trang web bên ngoài

Câu 27 : Có mấy loại danh sách ?

* Danh sách không có thứ tự:
  + Tạo bằng thẻ <ul>
  + Mỗi mục được đại diện bằng thẻ <li>
* Danh sách có thứ tự :
  + Tạo bằng thẻ <ol>
  + Mỗi mục được đại diện bằng thẻ <li>
* Danh sách mô tả :
  + Tạo bằng thẻ <dl>
  + Mỗi mục trong danh sách bao gồm thẻ <dt> để định nghĩa và thẻ <dd> để mô tả

Câu 28 : Hiển thi trang danh sách có thứ tự bắt đầu C.D.E.... thì làm thế nào ?

* Sử dụng phương thức sort();
* Kiểm tra từ bắt đầu phải là C không
* So sánh

Câu 29 : Form hay biểu mẫu dùng để làm gì ?

* Thu thập thông tin : chủ yếu để thu thập thông tin từ người dùng như tên, mail, địa chỉ, số điện thoại
* Gửi dữ liệu : sau khi nhập dữ liệu vào thì dữ liệu sẽ gửi đếm máy chủ để xử lý
* Xác thực dữ liệu :
* Tương tác với người dùng :
  + Chọn các tùy chọn
  + Thực hiện hành động
  + Gửi thông tin

Câu 30 : Thuộc tính action của form dùng để làm gì ?

* Thuộc tính “action” của một form trong HTML được sử dụng để chỉ định URL của trang hoặc tập lệnh xử lý dữ liệu khi form được gửi đi. Khi người dùng nhấn submit trong form, dữ liệu sẽ được gửi đến địa chỉ được chỉ định trong thuộc tính action

Câu 31 : Phân biệt 2 method ‘GET’ và ‘POST’

|  |  |
| --- | --- |
| GET | POST |
| * Dữ liệu được gửi đi sẽ hiển thị trên thanh địa chỉ của trình duyệt * Không nên sử dụng để gửi các dữ liệu nhạy cảm (chẳn hạn như mật khẩu) * Thường được sử dụng để gửi các dữ liệu nhỏ | * Dữ liệu được gửi đi KHÔNG hiển thị trên thanh địa chỉ của trình duyệt * Được sử dụng để gửi những dữ liệu nhạy cảm * Không hạn chế dung lượng gửi đi, thường được dùng để gửi dữ liệu lớn lên server |

Câu 32: Phân biệt các nút bấm có type là : button, submit, reset ?

* Nút button (type=’button’) : nút này không gửi dữ liệu form khi được nhấn. Sử dụng để thực hiện các hành động javaScript cụ thể đã được xác định. Hoặc xử lý sự kiện
* Nút gửi (type=’submit’) : gửi dữ liệu trong form đến một URL được chỉ định trong thuộc tính ‘action’ của form
* Nút thiết lập lại (type=’reset’) : nút này được sử dụng để đặt lại tất cả các trường nhập dữ liệu trong form về giá trị ban đầu của chúng.

Câu 33: Muốn submit form đến CodeGym.vn thì là thế nào ?

<form action="http://demo.CodeGym.vn/6/registration\_form/register.php">

<input type="text" name="q"/>

<input type="submit" value="form"/>

</form>

Câu 34 : Nếu muốn nối 3 ô trên cùng 1 hàng thì làm thế nào ?

* Colspan=’3’

Câu 35 : Nếu muốn nối 2 ô trên cùng 1 cột thì làm thế nào ?

* rowspan=’2’

câu 36 : Phân biệt thuộc tính cellpadding và cellspacing ?

* cellpadding xác định khoảng cách giữa nội dung của mỗi ô và ranh giới của ô đó
* cellspacing xác định khoảng cách giữa các ô

câu 37 : Trình bày cách nhúng javaScript vào 1 tài liệu HTML ?

* gôm có 3 cách để nhúng 1 file javaScript vào HTML:
  + Viết mã javaScript bên trong thẻ <script>
  + Sử dụng file.js rồi nhúng vào trang web
  + Viết trực tiếp mã javaScript trong thẻ HTML

CÂU 38: 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?

* đúng, vì nó sử dụng các đối tượng để tổ chức mã và dữ liệu, javaScript không chỉ dùng để phát triển ứng dụng internet dựa trên client mà còn có thể sử dụng trên phía server thông qua Node.js

câu 39 : Phát biểu “javaScript phân biệt chữ hoa hay chữ thường là đúng hay sai”

* Đúng javaScript phân biệt chữ hoa và chữ thường

Câu 40 : Trình bài quy tắc CamelCase và camelCase ?

* CamelCase :
  + Là quy tắc đặt tên mà mỗi từ trong tên bắt đầu bằng một chữ cái viết hoa (trừ từ đầu tiên)
  + Được ghi liền nhau mà viết hoa chữ cái đầu tiên của mỗi từ
* camelCase :
  + cũng giống như CamelCase nhưng khác là chữ cái đầu tiên là viết thường

câu 41: Biến là gì ? Phân biệt biến cục bộ (local) và biến toàn cục (glocal)?

* Biến là :
  + Tên gọi được gán cho một vùng nhớ dữ liệu
  + Dữ liệu được lưu trong biến được gọi là giá trị (value)
  + Có thể truy nhập, gán hay thay đổi giá trị của biển
  + Khi gán 1 giá trị mới thì giá trị cũ sẽ bị ghi đè lên
  + Cần khai báo biến trước khi sử dụng
* Phân biệt local và glocal

|  |  |
| --- | --- |
| Biến cụ bộ (local) | Biến toàn cục(glocal) |
| * Sử dụng khi bắt đầu được khai báo cho đến điểm kết thúc của khối lệnh đó | * Sử dụng bất cứ ở đâu trong chương trình |

Câu 42: Phân biệt var, let, const ?

|  |  |  |
| --- | --- | --- |
| Var | Let | Const |
| * Có Hoisting * Có thể khai báo lại * Có phạm vi là function scope | * Không có Hoisting * Không thể khai báo lại * Có phạm vi là block scope | * Không thể thay đổi giá trị của hằng số |

Câu 43: Nêu tên các biểu thức dữ liệu nguyên thủy trong javaScript ?

* Number
* String
* Boolean
* Null
* Undefined
* Symbol

Câu 44: Trình bày về toán tử Typeof ? khai báo let x = 10; thì x có kiểu dữ liệu là gì ?

* Typeof được dùng để trả về kiểu dữ liệu của một biến hoặc một giá trị
* X sẽ trả về kiểu dữ liệu là Number

Câu 45: Trình bày cách đưa thông báo trong JavaScript ?

* Dùng alert() : hàm này hiển thị một hộp thoại thông báo với một tin nhắn và một nút OK
* Dùng confirm() : hàm này hiển thị thông báo với một tin nhắn, có một nút OK và một nút cancel , được sử dụng để xác nhận hay hủy bỏ một hành động
* Dùng prompt() : hiển thị một hộp thoại thông báo với một tin nhắn , một trường nhập dữ liệu và 2 nút OK với Cancel .sử dụng để nhận thông tin từ người dùng.

Câu 46: Trình bày về các loại toán tử trong javaScript ?

* Toán tử số học ( +, -, \*, /, %, \*\*)
* Toán tử gán ( =, +=, -=, \*=, /=)
* Toán tử so sánh (==, ===, !=, !==, <, >, <=, >=)
* Toán tử logic ( &&, ||, !)
* Toán tử ba ngôi ( biểu thức điều kiện ? trả về gí trị true : trả về giá trị false ;
* Toán tử typeof

Câu 47 : Kết quả của biểu thức +0 === -0 là true hay false ?

* Trả về giá trị là true vì +0 và -0 là cùng 1 giá trị

Câu 48: trình bày về toán tử ++, --, và – trong javaScript ?

* Toán tử ++ :
  + Nếu toán tử ++ được đặt trước biến ( ++a ) thì giá trị của biến sẽ được tăng lên trước khi biểu thức được đánh giá
  + Nếu toán tử ++ được đặt sau biến ( a++ ) thì giá trị của biến sẽ tăng lên sau khi biểu thức được đánh giá
* Toán tử -- :
  + Tương tự như ++
* Toán tử - :
  + Chuyển đổi số dương thành số âm và ngược lại

câu 49 : Câu lệnh điều kiện / rẻ nhánh là gì ?

* Thực hiện các câu lệnh từ trên xuống dưới và có thể thay đổi luồng thực thi thông qua các điều kiện.
* Được dùng để thực hiện các hành động khác nhau dựa trên điều kiện của chương trình, câu lệnh này cho phép chương trình lựa chọn giữa các khối mã được thực thi dựa trên một điều kiện cụ thể.

Câu 50 : Có những loại câu lệnh điều kiện nào ?

* Câu lệnh điều kiện if
* Câu lệnh điều kiện if – else
* Câu lệnh điều kiện switch -case
* Toán tử 3 ngôi
* Câu lệnh điều kiện logic (&& , ||)

Câu 51 : Trình bày về câu lệnh if ?

* Dùng để kiểm tra một điều kiện và thực thi nếu điều kiện đó đúng

Câu 52: Trình bày về câu lệnh switch – case ?

* Kiểm tra một biểu thức cho nhiều giá trị khác nhau và thực thi các hành động tương ứng với từng giá trị.

Câu 53 : Tại sao nói else là một tùy chọn ( optional) của câu lệnh if ?

* Vì nó không bắt buộc đi kèm với câu lệnh “ if “

Câu 54 : Khi nào nên sử dụng câu lệnh switch ?

* Khi so sánh bằng (===)
* Vượt quá 3 biểu thức điều kiện
* Khi muốn kiểm tra một biến có nhiều giá trị khác nhau
* Khi cần phải xử lý nhiều trường hợp cùng một lúc

Câu 55 : Câu lệnh break dùng để làm gì ? nếu một case mà không có break thì chương trình sẽ chạy thế nào ?

* Lệnh break :
  + Dùng để thoát khỏi vòng lặp mà không đi đến điểm cuối
  + Khi gặp câu lệnh này trong vòng lặp, chương trình sẽ thoát ra khỏi vòng lặp và chỉ đến câu lệnh liền sau đó
  + Nếu có nhiều vòng lặp thì sẽ thoát ra vòng lặp gần nhất
  + Ngoài ra break còn dùng trong switch – case
* Khi case mà không có break thì chương trình sẽ tiếp tục thực thi từ trường hợp đúng cho đến khi gặp câu lệnh break hoặc kết thúc switch dẫn đến kêt quả không mong muốn

Câu 56: Mệnh đề default có bắt buộc trong câu lệnh switch ?

* Default không bắt buộc trong câu lệnh switch, mệnh đề default để xác định hành động mặc định mà khi không có trường hợp nào đúng với giá trị của biểu thức trong switch

Câu 57: Trình bày toán tử 3 ngôi ?

* Gồm một biểu thức điều kiện ? biểu thức true : biểu thức false ;

Câu 58: Trình bày về sự kiện và trình quản lý sự kiện trong javaScript ?

* Sự kiện là các hành động hoặc trạng thái có thể xảy ra trong trình duyệt, các sự kiện có thể là do người dùng thao tác như click chuột, gõ phím, di chuyển chuột, ...
* Trình quản lý sử kiện cho phép đính kèm các hàm vào các sự kiện cụ thể, khi sự kiện xảy ra thì hàm đó sẽ được gọi để xử lý sự kiện đó.

Câu 59 : Sự kiện onchange phát sinh khi nào ?

* Khi một giá trị của một phần tử HTML có thay đổi
* Ví dụ như khi người dùng chọn hoa quả từ danh sach lựa chọn, sự kiện “onchange” sẽ được kích hoạt , và hàm ứng với tên của sự kiện mình đã đặt sẽ được gọi, nó sẽ thực hiện hành động mà ta đã cài đặt từ trước như in ra lựa chọn của bạn

Câu 60 : sự kiện onclick phát sinh khi nào ?

* Phát sinh khi người dùng nhấp chuột vào một phần tử HTML có thể là nhấn nút, một liên kết, một hình ảnh, hoặc bất kì phần tử nào nhận sự kiện nhấp chuột.

Câu 61: Muốn hiển thị 1 hộp thoại xác nhận “Bạn chắc chắn xóa không ?” với người dùng khi nhấn vào nút delete thì làm thế nào ?

* Đầu tiên tạo 1 hàm có tên là delete()
* Sao đó tạo 1 thông báo confirm và gán cho biến result
* Xét điều kiện nếu result trả về là true khi người dùng nhấn OK thì hành động xóa được thực hiện
* Còn nếu người dùng nhấn vào cancel thì result trả về giá trị false và không có hành động xóa nào xảy ra
* Tạo 1 nút button ứng với sự kiện onclick cùng với tên hàm đã tạo trên

Câu 62: Phân biệt onmousevover, onmouseout, onmousemove ?

|  |  |  |
| --- | --- | --- |
| Onmouseover | Onmouseout | Onmousemove |
| * Sự kiện xảy ra khi con trỏ chuột dy chuyển vào một phần tử HTML | * Sự kiện xảy ra khi con trỏ chuột rồi khỏi phần tử HTML mà nó đã di chuyển vào trước đó bằng sự kiện onmouseover | * Sự kiện này xảy ra khi con trỏ chuột di chuyển bên trong một phần tử HTML |

CÂU 63: Phân biệt onkeyup và onkeydown, onkeypress ?

|  |  |  |
| --- | --- | --- |
| Onkeyup | Onkeydown | Onkeypress |
| * Sự kiện được xảy ra sau khi phím được nhả ra sau khi nhấn | * Sự kiện được xảy ra khi phím được nhấn xuống | * Sự kiện xảy ra khi phím nhấn giữ |

câu 64: Sự kiện onload phát sinh khi nào ?

* Xảy ra khi một phần tử HTML hoặc toàn bộ trang HTML đã được tải hoàn toàn trong trình duyệt web của người dùng.
* Thường được sử dụng để thực hiện các hành động hoặc chức năng mà cần phải chờ đợi đến khi trang hoàn toàn được tải, chẳng hạn như hiển thị nội dung, khởi động các chức năng của javaScript

Câu 65: Phân biệt innerHTML và innerText ?

|  |  |
| --- | --- |
| InnerHTML | innerText |
| * Thay đổi nội dung của HTML cũ bằng một chuỗi HTML mới cho nó * Trình duyệt sẽ phân tích chuỗi HTML và tạo ra các phần tử tương ứng | * Thay thế thẻ HTML và in ra văn bản |

Câu 66 : Nếu muốn vô hiệu hóa (disabled) một button khi click vào 1 checkbox thì là thế nào ?

* Tạo ra input có type là checkbox có id là mycheckbox
* Tạo ra 1 nút có id là mybutton
* Gán 2 biến cho 2 nút
* Gắn sự kiện click cho checkbox bằng hàm addEventListener
* Sau đó kiểm tra trạng thái của checkbox nếu true thì vô hiệu hóa còn false thì không

Câu 67 : Phân biệt thuộc tính readonly và disabled của HTML ?

|  |  |
| --- | --- |
| Readonly | Disabled |
| * Chỉ định rằng một phần tử không thể thay đổi, nhưng vẫn nhận được sự tương tác từ người dùng * Khi bạn đặt thuộc tính này ở một ô input hay thẻ textarea thì người dùng vẫn có thể xem nội dung nhưng k thể thay đổi nó. | * Để vô hiệu hóa một phần tử HTML, làm cho nó không thể nhận được sự tương tác từ người dùng * Khi được vô hiệu hóa nó thường có giao diện là mờ hoặc không thể nhấp chuột cũng như không thể tương tác từ bàn phím |

Câu 68 : Vòng lặp là gì ?

* Vòng lặp cho phép tự động thực hiện một khối lệnh lặp đi lặp lại nhiều lần dựa trên một điều kiện cho trước

Câu 69 : Nếu các loại vòng lặo cơ bản trong javaScript ?

* Vòng lặp for
* Vòng lặo while
* Vòng lặp do – while

Câu 70 : phân biệt vòng lặp for, while, do – while ?

|  |  |  |
| --- | --- | --- |
| For | While | Do – while |
| * Lặp với số lần biết trước | * Lặp với số lần không biết trước | * Tương tự như while * Sử dụng trong trường hợp muốn lặp ít nhất 1 lần |

Câu 71: khi nào nên sử dụng vòng lặp do – while ?

* khi muốn lặp ít nhất một lần

câu 72 : khi nào nên sử dụng vòng lặp for ?

* khi biết trước số lần lặp

câu 73 : khi nào nên sử dụng vòng lặp while ?

* khi k biết trước số lần lặp

câu 74 : 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?

* Có và nên sử dụng vòng lặp do – while

Câu 75: Yêu cầu người dùng tính tổng các số chẵn từ 10 đến 100 thì nên dùng vòng lặp loại nào?

* Vòng lặp for vì biết trước số lần lặp

Câu 76: Nếu muốn hiển thị các cặp số nguyên dương có tổng là 10 thì viết thế nào?

function hienThiCapSoTong10() {

for (let i = 1; i <= 9; i++) {

let j = 10 - i;

console.log(`${i} + ${j} = 10`);

}

}

hienThiCapSoTong10();

Câu 77 : Nếu muốn hiển thị hình tam giác vuông \* thì nên viết thế nào?

* for (*i* = 1; *i* <= 5; *i*++) {  
   let cham = '\* ';  
   for ( *j* = 2; *j* <= *i*; *j*++) {  
   cham += '\* ';  
   *document*.write('<p></p>');  
   }  
   *document*.write(cham);  
   }

câu 78 : Trình bày cách chạy vòng lặp for lồng nhau ?

câu 79 : Phân biệt câu lệnh break và continue ?

|  |  |
| --- | --- |
| Break | Continue |
| * Dùng để thoát khỏi vòng lặp mà không đi đến điểm cuối * Khi gặp câu lệnh này trong vòng lặp, chương trình sẽ thoát ra khỏi vòng lặp và chỉ đến câu lệnh liền sau đó * Nếu nhiều vòng lặp thì câu lệnh break sẽ thoát ra khỏi vòng lặo gần nhất | * Để nhảy qua một phần tử của khối lệnh và bắt đầu dòng tiếp theo |

Câu 80 : Hiển thị các số không chia hết cho 5 từ 1-100 có sử dụng continue thì làm thế nào?

* For (let i = 1; i < 100 ; i++) {

If ( i % 5 === 0 ) {

Continue;

}

Console.log (i);

}

Câu 81 : Mảng là gì ?

* Là một loại biến đặc biệt, có thể lưu được nhiều giá trị thay vì một giá trị như các biến thông thường
* Mỗi giá trị trong mảng được gọi là một phần tử
* Các phần tử được lưu ở vị trí kế tiếp nhau trong bộ nhớ

Câu 82 : Trình bày cú pháp khai báo mảng trong javsScript ?

* Có 2 cách để khai báo mảng:
  + Để các phần tử trong dấu ngoặc vuông.
  + Dùng từ khóa new Array và các phần tử được nằm trong dấu ngoặc tròn
  + Ngoài ra ta có thể để mỗi Array cũng đc

Câu 83 : Chỉ số của mảng là gì ? chỉ số của mảng xuất phát từ 0 hay 1 ?

* Chỉ số của mảng là một số nguyên dương được sử dụng để định vị các phần tử trong mảng và cho biết vị trí của các phần tử trong mảng.
* Chỉ số của mảng bắt đầu bằng 0 kết thúc bằng độ dài mảng – 1

Câu 84 : Kích thước của mảng là gì?

* Là số lượng phần tử có trong mảng đó

Câu 85 : Thao tác duyệt mảng là gì? Triển khai thao tác bằng code?

* Sử dụng vòng lặp foreach :
  + let arr = [1, 2, 3, 4, 5];

arr.forEach(function(item) {

console.log(item);

});

* sử dụng vòng lặp for :
  + let arr = [1, 2, 3, 4, 5];

for (let i = 0; i < arr.length; i++) {

console.log(arr[i]);

}

* sử dụng vòng lặp for – of :
  + let arr = [1, 2, 3, 4, 5];

for (let item of arr) {

console.log(item);

}

Câu 86 : Thao tác duyệt mảng 2 chiều ?

* có thể sử dụng vòng lặp for lồng nhau để duyệt mảng 2 chiều hoặc foreach

câu 87 : Mảng 2 chiều cần mấy chỉ số để xác định duy nhất 1 phần tử?

* mảng 2 chiều cần 2 chỉ số để xác định duy nhất 1 phần tử

câu 88 : Viết code tính tổng của phần tử của ma trận số (ú tưởng) ?

* tạo 1 biến count và gán giá trị là 0
* dùng 2 vòng lặp lồng nhau duyệt qua các phần tử của mảng rồi cộng chúng lại với nhau

câu 89 : Viết code tính tổng các ô trên cùng hàng của ma trận số (ý tưởng)?

function tinhTongHang(matrix, hang) {

let tong = 0;

for (let j = 0; j < matrix[hang].length; j++) {

tong += matrix[hang][j];

}

return tong;

}

// Một ma trận ví dụ

let maTran = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

];

let hangCanTinh = 1; // Hàng cần tính tổng (tính từ 0)

// Gọi hàm tính tổng và in kết quả

let tongHang = tinhTongHang(maTran, hangCanTinh);

console.log("Tổng các ô trên hàng " + hangCanTinh + " là:", tongHang);

Câu 90 : Viết code tính tổng các ô trên cùng cột của ma trận số (ý tưởng)?

function tinhTongCot(matrix, cot) {

let tong = 0;

for (let i = 0; i < matrix.length; i++) {

tong += matrix[i][cot];

}

return tong;

}

// Một ma trận ví dụ

let maTran = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

];

let cotCanTinh = 1; // Cột cần tính tổng (tính từ 0)

// Gọi hàm tính tổng và in kết quả

let tongCot = tinhTongCot(maTran, cotCanTinh);

console.log("Tổng các ô trên cột " + cotCanTinh + " là:", tongCot);

câu 91 : Hàm (function) là gì ?

* Là một nhóm các câu lệnh thực thi một nhiệm vụ nhất định
* Mỗi hàm chỉ nên có 1 chức năng duy nhất

Câu 92 : Phân biệt giữa tham số hình thức (formal arguments) và tham số thực (actual arguments)?

* Tham số hình thức (formal arguments) :
  + Đây là các tham số được khai báo trong định nghĩa của hàm
* Tham số thực (actual arguments)
  + Đây là các giá trị được truyền vào khi gọi hàm
  + Là những là trị cụ thể được sử dụng bởi hàm trong thời điểm thực thi
  + Tham số thực cần tương thích với tham số hình thức theo số lượng và kiểu dữ liệu

Câu 93 : Gọi hàm (invoke) là gì ?

* Thực hiện một hàm trong chương trình bằng cách sử dụng tên của hàm với các tham số thực được truyền vào nếu có.

Câu 94 : Phân biệt hàm gọi (calling function) và hàm được goi (called function)?

* Hàm gọi (calling function):
  + Hàm được sử dụng để kích hoạt (invoke) của hàm khác
  + Trong quá trình gọi hàm thì có thể truyền các tham số thực vào nếu có
* Hàm được gọi (called function)
  + Là hàm được gọi khi hàm gọi được kích hoạt

Câu 95 : Phân biệt truyền tham trị và truyền tham chiếu ?

* Truyền tham trị (pass but value) :
  + Giá trị được sao chép và truyền vào hàm
  + Hàm chỉ làm việc với bản sao mà không ảnh hướng đến bản gốc
  + Áp dụng với number, string, boolean
* Truyền tham chiếu (pass by reference):
  + Tham chiếu đến đối tượng hoặc mảng được truyền vào hàm
  + Hàm thao tác trên đối tượng hoặc mảng được truyền, ảnh hưởng đến giá trị gốc nằm ngoài hàm
  + Áp dụng cho array, object

Câu 96 : Câu lệnh return dùng để làm gì ?

* Đóng vai trò quan trọng khi chúng ta muốn kết thúc việc thực thi của một hàm và trả về một giá trị cụ thể
* Return được gọi trong một hàm thì quá trình thực thi của hàm đó sẽ dừng lại
* Nếu được chỉ định, giá trị cung cấp sẽ được trả về trong cuộc gọi hàm
* Trong trường hợp không có biểu thức nào được cung cấp thì giá trị trả về là undefined

Câu 96 : Có được phép viết “return;” ?

* Có thể sự dụng “return;” khi muốn kết thúc hàm mà không cần trả về bất cứ giá trị nào

Câu 97 : Phân biệt break và return ?

|  |  |
| --- | --- |
| Break | Return |
| * Là một câu lệnh được sử dụng để thoát khỏi vòng lặp hoặc câu lệnh switch * Khi được gọi trong vòng lặp, nó sẽ kết thúc vòng lặp ngay lập tức và chương trình sẽ tiếp tục thực thi từ lệnh sau vòng lặp đó * Trong switch thoát ra 1 switch và thực hiện từ lệnh sau switch | * Là một câu lệnh được sử dụng để trả về một giá trị và kết thúc hàm * Khi return được gọi hàm sẽ kết thúc ngay lập tức và trả về giá trị chỉ định ở return, và được trả về nơi gọi hàm * Có thể sử dụng ở bất kỳ đâu trong hàm và có thể có hoặc không giá trị trả về |

Câu 98 : Hàm thực hiện việc kiểm tra nên trả về giá trị không? Nếu có, nên trả về gì?

* Có , nên trả về kiểu boolean

Câu 99 : Tại sao hàm hoán vị swap(a,b) lại không thực hiện được việc hoán vị của các tham số gốc?

* Vì khi đó hàm đang làm việc với bản sao của a, b nên các giá trị gốc ngoài hàm sẽ không ảnh hưởng

Câu 100 : Trình bày về OOP ?

* Là phương pháp lập trình dựa trên các khái niệm về đối tượng và quan hệ giữa chúng, có thể tái sử dụng.

Câu 101 : Phân biệt giữa lập trình hướng đối tượng và lập trình hướng thủ tục/chức năng?

* Lập trình hướng đối tượng (OOP) :
  + Tập trung vào các đối tượng : các đối tượng là các thực thể có thuốc tính và hành vi riêng.
  + Kế thừa (inheritence) :tái sử dụng mã nguồn từ việc kế thừa các thuộc tính và phương thức của lớp cha
  + Đóng gói (encapsulation ) : ẩn thông tin đảm bảo tính bảo mật và chỉ thể hiện thông qua các phương thức công khai
* Lập trình hứng thủ tục/ chức năng :
  + Tập trung vào các hàm và thủ tục : mã nguồn được tổ chức thành các hàm và thủ tục, mỗi hàm và thủ tục thực hiện một nhiệm vụ cụ thể
  + Không có khái niệm về đối tượng :
  + Thực hiện theo dòng lệnh từ trên xuống dưới

Câu 102 : Phân biệt Object và class

|  |  |
| --- | --- |
| Object | Class |
| * Là một thể hiện của lớp * Một đối tượng được khai báo nhiều lần theo yêu cầu * Đối tượng được khai báo bằng từ khóa “ new” hoặc tên lớp * Ví dụ là : lớp car ta có xe BMW, Toyota,.. | * Là bản thiết kế hoặc khuông mẫu mà từ đó các đối tượng được tạo ra * Là tập hợp các đối tượng có thuộc tính giống nhau * Một lớp chỉ được khai báo 1 lần * Được khai báo bằng từ khóa “class” * Ví dụ : lớp car,lớp human |

Câu 103: Hai khái niệm “môn học” và “môn học javaScript” thì đâu là object ?

* Môn học javaScript là Object

Câu 104 : Mô tả về sơ đồ lớp ?

* Lớp ( class) gồm các thuộc tính và phương thức (method) của lớp đó có thể có
* Thuộc tính (property)
* Phương thức (method/0
* Các mối quan hệ như kế thừa

Câu 105 : thuộc tính là gì ? nên dùng từ loại nào để biểu diễn thuộc tính ?

* Thuộc tính là các dữ liệu mà một đối tượng có thể chưa, thường mô tả các đặc điểm trạng thái của đối tượng đó.
* Nên dùng từ : this.tên thuộc tính
  + Tên (Name) : sử dụng chuỗi (string)
  + Tuổi (Age) : sử dụng số nguyên (int)
  + Chiều cao (Height) : sử dụng số thực (float)
  + Địa chỉ (Address) : sử dụng chuỗi (string)
  + Điểm số (score) : sử dụng số thực (float)
  + Tình trạng (Status) : sử dụng kiểu (boolean)

Câu 106 : Phương thức là gì ? nên dùng từ loại nào để biểu diễn phương thức?

* Phương thức là một khối mã được sắp xếp để thực hiện một nhiệm vụ cụ thể .
* Các từ loại để biểu diễn phương thức :
  + Được biểu diễn bằng các hàm (function) và gán cho các thuộc tính

Câu 107 : cú pháp định nghĩa 1 class trong js ?

* Có thẻ định nghĩa 1 lớp (class) bằng từ khóa “class”
  + className : tên của lớp
  + constructor : là phương thức để khởi tạo các thuộc tính của đối tượng
  + method : các phương thức khai báo
  + this : là một tham chiếu đến đối tượng hiện tại mà phương đang hoạt động

câu 108 : Cú pháp tạo 1 đối tượng từ class?

* sử dụng đối tượng mới bằng cách sử dụng từ khóa “new”
* và truyền các đối số vào constructor của lớp đó nếu có

câu 109 : Trình bày một số class có sẵn trong javaScript mà em biết ?

* Object
* Array
* Number
* Boolean
* String
* Date
* Function

Câu 110 : Cú pháp truy cập đến một thuộc tính của đối tượng ?

* Sử dụng toán tử chấm “.” Sau tên của đối tượng

Câu 111: Cú pháp truy cập đến một phương thức của đối tượng ?

* Sử dụng toán tử chấm “.” Sau tên của đối tượng, theo sau là tên của phương thức sau đó là dấu ngoặc dơn () .

Câu 112 : Từ khóa this dùng để làm gì ?

* Dùng để tham chiếu đến đối tượng hiện tại mà phương thức đó đang hoạt động bên trong đó.
* Thường được sử dụng trong phương thức của một đối tượng để tham chiếu đến các thuộc tính hoặc phương thức của đối tượng đó

Câu 113 : Trình bày về DOM (Document Object Model – Mô hình đối tượng tài liệu)?

* Là mô hình lập trình được sử dụng trong các ngôn ngữ như HTML để biểu diễn và tương tác với cấu trúc tài liệu web hoặc tài liệu dữ liệu.
* Được tổ chức theo cấp bậc của các phần tử trong tài liệu
* Cho phép javaScript tương tác và thay đổi cấu trúc và nội dung của tài liệu web

Câu 114 : Thẻ <canvas> dùng để làm gì ?

* Được sử đụng để vẽ đồ họa và đồ thị động trên trình duyệt web bằng cách sử dụng javaScript
  + Vẽ hình ảnh và đồ họa động
  + Xử lý hình ảnh
  + Vẽ và ghi chú
  + Tạo trò chơi
  + Tạo hiệu ứng động

Câu 115 : Vẽ hình chữ nhật trên canvas?

<canvas id="myCanvas" width="200" height="200" style="border:1px solid black;"></canvas>  
<script >  
 // Lấy ra đối tượng canvas từ DOM  
 var *canvas* = *document*.getElementById('myCanvas');  
 // Lấy ra đối tượng context 2D để vẽ lên canvas  
 var *ctx* = *canvas*.getContext('2d');  
  
 // Đặt thuộc tính fillStyle để xác định màu sắc cho hình chữ nhật  
 *ctx*.fillStyle = 'blue';  
 // Vẽ hình chữ nhật với toạ độ (x, y) là (50, 50), chiều rộng là 100, chiều cao là 50  
 *ctx*.fillRect(50, 50, 100, 50);  
  
</script>

Câu 116 : Vẽ hình tròn trên canvas ?

// Lấy ra đối tượng canvas từ DOM  
var *canvas* = *document*.getElementById('myCanvas');  
// Lấy ra đối tượng context 2D để vẽ lên canvas  
var *ctx* = *canvas*.getContext('2d');  
  
// Đặt thuộc tính fillStyle để xác định màu sắc cho hình tròn  
*ctx*.fillStyle = 'red';  
// Bắt đầu một đường vẽ mới (path)  
*ctx*.beginPath();  
// Vẽ một hình tròn tại toạ độ (x, y) là (100, 100) và bán kính là 50  
*ctx*.arc(100, 100, 50, 0, 2 \* *Math*.PI);  
// Đóng path  
*ctx*.closePath();  
// Lấp đầy hình tròn với màu sắc đã được đặt trước đó  
*ctx*.fill();

Câu 117 : vẽ text trên canvas ?

* Sử dụng phương thức fillText() để ghi văn bản
* Ví dụ:
  + Ctx.fillText(‘hello, world’, 100, 100)
  + Trong đó :
    - 100 đầu là tọa độ x bắt đầu vẽ văn bản
    - 100 sau là tọa độ y bằng đầu vẽ văn bản

Sử dụng ctx.font = ‘20px Arial’ để định nghĩa font chữ và kiểu chữa

Câu 118 : Nếu muốn di chuyển một hình tròn trên canvas cần làm gì?

let canvas = document.getElementById('myCanvas');

let ctx = canvas.getContext('2d');

// Khai báo tọa độ x và y của hình tròn

let x = 100;

let y = 100;

let radius = 50;

function moveCircle() {

// Xóa hình tròn cũ

ctx.clearRect(0, 0, canvas.width, canvas.height);

// Cập nhật vị trí mới của hình tròn

x += 1; // Di chuyển sang phải mỗi lần gọi hàm

// Vẽ hình tròn mới ở vị trí mới

ctx.beginPath();

ctx.arc(x, y, radius, 0, 2 \* Math.PI);

ctx.fillStyle = 'red';

ctx.fill();

}

setInterval(moveCircle, 16);