**Bộ 60 hỏi phỏng vấn**

|  |  |  |
| --- | --- | --- |
| stt | **Câu hỏi** | **Câu trả lời** |
|  | Hệ thống các kiến thức đã học ở Module1: | -[Học cách học & Quản lí công việc bằng Kanban](https://james.codegym.vn/course/view.php?id=62#section-1)  -[Nhập môn tư duy lập trình](https://james.codegym.vn/course/view.php?id=62" \l "section-2)  -[Mô tả thuật toán bằng Pseudo code & Flowchart](https://james.codegym.vn/course/view.php?id=62" \l "section-3)  -[Git & HTML](https://james.codegym.vn/course/view.php?id=62#section-4)  -[HTML Form & Table](https://james.codegym.vn/course/view.php?id=62#section-5)  -[Tổng quan JavaScript](https://james.codegym.vn/course/view.php?id=62" \l "section-6)  -[Biến, kiểu dữ liệu và toán tử](https://james.codegym.vn/course/view.php?id=62" \l "section-7)   -Cấu trúc điều kiện  -Cấu trúc lặp  -Mảng  -Hàm  -Lập trình hướng đối tượng |
|  | Giải thuật/Thuật toán là gì? | Giải thuật là một tập hợp hữu hạn các chỉ thị để được thực thi theo một thứ tự nào đó để thu được kết quả mong muốn. |
|  | So sánh việc trình bày giải thuật bằng pseudo-code và flowchart? | - Pseudo-code là ta sẽ vay mượn các cú pháp của một ngôn ngữ lập trình nào đó và một phần ngôn ngữ tự nhiên để thể hiện thuật toán  - Flowchart Lưu đồ hay sơ đồ khối là một công cụ trực quan để diễn đạt các thuật toán. Biểu diễn thuật toán bằng lưu đồ sẽ giúp người đọc theo dõi được sự phân cấp các trường hợp và quá trình xử lý của thuật toán |
|  | Tại sao lại cần trình bày giải thuật trước khi bắt tay vào code? | Trước hết cần ý tưởng, giải pháp giải quyết bài toán rồi sau đó mới thực hiện các bước sau |
|  | Tại sao lại nên sử dụng flowchart hơn pseudo-code khi thể hiện giải thuật của bài toán phức tạp? | Tại vì khi một bài toán phức tạp số lượng các bước thực hiện lớn thì cần có một quy tắc và thể hiện được luồng thực thi của chương trình để mọi người đọc đều có thể hiểu được mà pseudo-code ko đáp ứng được |
|  | Tại sao giải thuật lại cần hữu hạn các bước thực thi? | Để tránh việc không kết thúc được công việc |
|  | Trình bày ý nghĩa các ký hiệu sử dụng khi vẽ flowchart? | - Hình oliu dùng cho: bắt đầu và kết thúc chương trình  - Hình thoi: điều kiện rẽ nhánh (lựa chọn)  - Mũi tên luồng xử lý  - Hình bình hành nhập xuất  - Hình chữ nhật xử lý tính toán  - Hình tròn đầu nối hai phần chương trình |
|  | Vẽ giải thuật giải phươngtrình bậc nhất? |  |
|  | Bạn biết gì về VCS – Version Control System (hệ thống quản lý phiên bản)? | - Là một hệ thống lưu giữ các phiên bản của [mã nguồn](https://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n" \o "Mã nguồn) của sản phẩm phần mềm, giúp các [lập trình viên](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_vi%C3%AAn" \o "Lập trình viên) có thể dễ dàng lấy lại phiên bản mong muốn.  - Hệ thống này hỗ trợ khôi phục lại phiên bản cũ, xem lại các thay đổi, xem được ai là người cuối cùng thực hiện thay đổi có thể gây ra lỗi và khôi phục lại các file vô tình xóa mất |
|  | Phân biệt hệ thống quản lý mã nguồn tập trung và hệ thống quản lý mã nguồn phân tán? | - Tập trung:bao gồn một máy chủ lưu tất cả tập tin được phiên bản hóa và các máy khách kết nối với server để lấy một phiên bản copy đầy đủ dự án hoặc cập nhật thay đổi mới nhất từ dự án làm việc trên phiên bản đó và sau đó lưu lại thay đổi của họ  - Phân tán: mỗi thành viên sẽ có một repo ở local và tất cả các thao tác commit hay add.. đều làm việc trực tiếp với nó giống như vừa là client vừa là một server. |
|  | Git là gì? Tại sao cần sử dụng git? | - Git là cách thức ( mô hình, giao thức, công cụ, quy tắc) để lưu trữ mã nguồn và hỗ trợ cộng tác trong quá trình phát triển dự án.  - Sử dụng git để quản lý code và lịch sử thay đổi cũng như giúp công việc nhóm thực hiện dễ dàng. |
|  | Git thuộc hệ thống quản lý mã nguồn tập trung hay phân tán? | Git là vcs phiên bản phân tán |
|  | Repository là gì? Phân biệt LocalRepository và RemoteRepository? | - Repository là nơi lưu trữ quản lý tất cả những thông tin cần thiết cũng như các thay đổi và lịch sử của toàn bộ dự án.  - localRepository là ở máy client  - remoteRepository là ở trên máy chủ |
|  | Đồng bộ giữa các repository (local và remote) để làm gì? | Để biết được thay đổi mới nhất của local repository và remote repository |
|  | Thao tác push của git để làm gì? Cú pháp câu lệnh? | Push để đẩy nội dung trong mấy local lên remote  // cú pháp : git push origin master |
|  | Thao tác pull của git để làm gì? | Lấy mã nguồn từ trên remotrepository về local |
|  | Thao tác clone của git để làm gì? Cú pháp câu lệnh? | - Copy một bản sao của kho chứa git có sẵn. thao tác chỉ làm 1 lần khi bắt đầu dự án.  - Git clone <https://link> đường dẫn của repository |
|  | Nêu tên và ý nghĩa các câu lệnh cơ bản đã được học để làm việc với git? | - Git init là khởi tạo một repo  - Git clone là copy một bản repo trên nguồn có sẵn về máy cá nhân  - Git add là đưa thêm file vào vùng staged(Đây là vùng chứa thông tin về những gì sẽ được commit trong lần commit sắp tới)  - Git commit ghi nhận các thay đổi  - Git push đẩy thay đổi từ local lên remote  - Git pull lấy cập nhật mã nguồn từ remote về local |
|  | www là gì? | Viết tắt của World Wide Wed : một không gian thông tin toàn cầu mà mọi người có thể truy cập (đọc và viết) qua các thiết bị kết nối với mạng Internet |
|  | HTML là gì? | Hypertext markup language dùng để cấu trúc nên trang wed bằng bộ các thẻ (<tag>) |
|  | Nêu tên các thẻ làm việc với văn bản trong tài liệu html? | <h1>-…<h6> thẻ tiêu đề  <p> đoạn văn bản  <pre> đoạn văn bản  <span> chuỗi ký tự |
|  | Thẻ div dùng để làm gì? <div> khác gì <span> | - Thẻ div là tạo một khu vực hay là khối (block) có chiều rộng chiều cao có thể chứa các thẻ khác bên trong dùng để tùy chỉnh layout cho trang wed. khi khai báo các thẻ div khác nhau sẽ ở mỗi dòng riêng biệt  - Thẻ <span> thẻ inline khi khai báo thì nội dung nó sẽ nằm trên một hàng |
|  | URL là gì? Phân biệt url tuyệt đối và url tương đối? | - url là đường đẫn đến địa chỉ của trang web mà bạn muốn chuyển đến khi người dùng bấm vào liên kết  - url tuyệt đối là một chuỗi đầy đủ bao gổm http://, tên miền của trang web, đường dẫn đến tập tin  - url tương là một phần nhỏ của đường dẫn tuyệt đối, thông thường đường dẫn tương đối là phần đường dẫn đến tập tin) |
|  | Siêu liên kết là gì? | - liên kết trong mã HTML còn được gọi là siêu liên kết ( hyperlink)  - ta có thể nhấp chuột vào một liên kết để đi đến trang wed khác |
|  | Muốn liên kết đến một phần trên cùng trang thì làm thế nào? | - Đầu tiên tạo bookmark điểm đánh dấu vị trí trang wed với cú pháp id=”tên bookmark”  - Tạo liên kết có #tên bookmark . |
|  | Muốn liên kết đến một phần trên trang khác thì làm thế nào? | - Đầu tiên tạo bookmark điểm đánh dấu vị trí trang wed với cú pháp id=”tên bookmark”  - Tạo liên kết có #tên bookmark phía sau đường dẫn đến trang wed đó. |
|  | Phân biệt internal link và external link? | - [**External Links**](https://lamvt.vn/backlink-la-gi-nhung-loi-ich-ma-backlink-mang-lai/) chỉ từ một tên miền đến miền hoàn toàn riêng biệt  - [**Internal Link**](https://lamvt.vn/internal-link-ky-thuat-seo-khong-bo-qua-giup-website-len-top-google/) chỉ trỏ trong [trang web](https://lamvt.vn/36-cach-toi-uu-toc-load-trang-web-theo-loi-khuyen-cua-google/) hoặc miền cụ thể của riêng bạn  Hay có thể hiểu:  - [External Links](https://lamvt.vn/backlink-la-gi-nhung-loi-ich-ma-backlink-mang-lai/) = liên kết trỏ đến tên miền riêng  - [Internal Links](https://lamvt.vn/internal-link-ky-thuat-seo-khong-bo-qua-giup-website-len-top-google/) = các liên kết trỏ đến [nội dung](https://lamvt.vn/quy-trinh-viet-bai-chuan-seo-toan-tap/) trong cùng một tên miền |
|  | Có mấy loại danh sách? | 3 loại danh sách:  -Danh sách không xếp thứ tự  <ul>  <li></li>  <li></li>  </ul>  -Danh sách xếp thứ tự  <ol>  <li></li>  <li></li>  </ol>  -Danh sách miêu tả  <dl>  <dt></dt>  <dd></dd>  <dt></dt>  <dd></dd>  </dl> |
|  | Muốn hiển thị danh sách có thứ tự bắt đầu C. D. E… thì làm thế nào? | <ol type=”A” start=”3”> |
|  | Form hay biểu mẫu dùng để làm gì? | Cho phép người dùng nhập dữ liệu và gửi về server |
|  | Thuộc tính action của form dùng để làm gì? | Thuộc tính **action** quy định nơi dữ liệu được gửi đến. Địa chỉ này có thể là trên cùng server với trang web hiện tại hoặc trên bất kỳ server nào khác. |
|  | Phân biệt 2 method là get và post? | - GET:  + Dữ liệu của phương thức này gửi đi thì hiện trên thanh địa chỉ (URL) của trình duyệt.  + HTTP GET có thể được cache bởi trình duyệt  + HTTP GET có thể duy trì bởi lịch sử đó cũng là lý do mà người dùng có thê bookmark được.  + HTTP GET không được sử dụng nếu trong form có các dữ liệu nhạy cảm như là password, tài khoản ngân hàng  + HTTP GET bị giới hạn số trường độ dài data gửi đi  - POST:  + Dữ liệu được gửi đi với METHOD POST thì không hiển thị trên thanh URL.  + HTTP POST không cache bởi trình duyệt  + HTTP POST không thể duy trì bởi lịch sử đó cũng là lý do mà người dùng không thê bookmark HTTP POST được.  + HTTP POST không giới hạn dữ liệu gửi đi |
|  | Phân biệt các nút bấm có type là: button, submit, reset? | Chỉ định kiểu của button - button: nút nhấn  - reset: nút reset form đưa về giá trị mặc định - submit: nút submit form. Đẩy dữ liệu từ form lên server |
|  | Muốn submit form đến trang CodeGym.vn thì làm thế nào? | <form action =” CodeGym.vn”>  Content  <input type =”submit” >  </form> |
|  | Nếu muốn nối 3 ô trên cùng 1 hàng thì làm thế nào? | Thêm thuộc tính colspan =”3” vào ô cần gộp( thẻ th hay td) |
|  | Nếu muốn nối 2 ô trên cùng 1 cột thì làm thế nào? | Thêm thuộc tính rowspan =”2” vào ô cần gộp( thẻ th hay td) |
|  | Phân biệt thuộc tính cellpadding và thuộc tính cellspacing? | - Cellspacing là tạo khoảng cách giữa các đường viền của các ô  - Cellpadding là tạo khoảng cách giữa nội dung trong ô với các đường viền |
|  | Trình bày các cách nhúng javascript vào 1 tài liệu html? | - Sử dụng thẻ <script> bên trong mã HTML  Vd: <script>      document.getElementById("demo").innerHTML = "My First JavaScript"; </script>  - Sử dụng file javascript riêng biệt  Vd: nhúng file myScript.js vào HTML  <html>  <body>  <script src="myScript.js"></script>  </body>  </html>  - Viết mã javascript ngay trong các thẻ htm  Vd: <button onclick="alert('Xin chào bạn!');">Click me</button> |
|  | 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.  -JavaScript là một ngôn ngữ kịch bản (scripting language) được dùng để tạo các script ở máy client (client-side script) và máy server (server-side script). Các script ở máy client được thực thi tại trình duyệt, các script ở máy server được thực hiện trên server. |
|  | Phát biểu “JavaScript phân biệt chữ hoa chữ thường ” đúng hay sai? | đúng |
|  | Trình bày về quy tắc CamelCase và camelCase? | -Với CamelCase là các chữ cái đầu mỗi từ viết hoa  -Còn camelCase thì từ chữ đầu tiên của từ thứ 2 mới viết hoa |
|  | Biến là gì? Phân biệt biến cục bộ (local) và biến toàn cục (global)? | - Biến là một tên gọi gắn cho một vùng nhớ chứa dữ liệu  - Biến cục bộ là biến chỉ được khai báo và sử dụng trong một khối code và nó tồn tại khi khối code nó được thực thi  - Biến toàn cục là biến được khai báo bên ngoài tất cả các hàm. Nó được sử dụng mọi hàm trong chương trình và nó tồn tại cho tới hết chương trình |
|  | Phân biệt: var, let, const? | - Var biến toàn cục  - Còn let khai báo biến cục bộ  - Const là khai báo biến là hằng số không thể thay đổi trị trong suốt chương trình |
|  | Phát biểu “JavaScript là ngôn ngữ định kiểu yếu hay động” nghĩa là gì? Trình bày? | Điều đó nghĩa là không cần phải khai báo kiểu của các biến trước khi dùng. Kiểu sẽ được xác định tự động trong khi chương trình được thực thi. Điều đó cũng có nghĩa là một biến có thể chứa giá trị của các kiểu dữ liệu khác nhau:  Let x = 40 ; //x là một số  x= ‘javascript’; // x giờ là một chuỗi  x= true; // giờ x là một kiểu boolean |
|  | Nêu tên các kiểu dữ liệu nguyên thủy trong JavaScript? | - Kiểu số number  - Kiểu chuỗi string  - Kiểu logic boolean  - Undefine giá trị không xác định  - Null không có giá trị nào |
|  | Trình bày về toán tử typeof? Khi khai báo “let x=10;” thì x có kiểu là gì? | -Toán tử typeof là dùng để lấy về kiểu dữ liệu của một biến hoặc một giá trị  -X là number |
|  | Trình bày các cách đưa thông báo trong JavaScript? | - Phương thức alert(nội dung thông báo); hiện thị một thông báo cho người dùng. Có thể ấn nút ok để đóng nó  - Phương thức confirm(nội dung); hiển thị thông báo cho người dùng lựa chọn. nếu ok trả về giá trị true còn cancel trả về false  - Phương thúc prompt(content); hiển thị cho người dùng nhập vào một thông tin. Nếu ok thì trả về nội dung được nhập và cancel trả về null. |
|  | Trình bày về các loại toán tử trong JavaScript? | Toán tử số học: + thực hiện phép cộng  - phép trừ  \* phép nhân  / phép chia  % chia lấy phần dư  ++ tăng một giá trị  -- giảm một giá trị  =Toán tử gán  += cộng xong rồi gán  -= trừ xong rồi gán  \*= nhân xong rồi gán  /= chia xong rồi gán  %= chia lấy dư xong rồi gán  \*\*= lũy thừa rồi gán  Toán tử cộng chuỗi  Toán tử so sánh (==, ===, !=, !==, <,>, <=, >=)  Toán tử logic( &&, ||, !)  Toán tử typeof |
|  | Phân biệt giữa 2 toán tử == và ===? | == so sánh giá trị: bằng nhau thì trả về true, khác nhau thì trả về false  === so sánh giá trị và so sánh kiểu dữ liệu. nếu giá trị bằng nhau và cùng kiểu dữ liệu trả về true |
|  | Phân biệt giữa 2 toán tử && và || trong JavaScript? | - && (AND) là toán tử là true khi cả 2 biểu thức logic là true  - || (or) chỉ trả về giá trị false khi cả hai biểu thức logic là false |
|  | Kết quả biểu thức +0===-0 là true hay false? | -True  -Số nguyên duy nhất có hai đại diện: 0 được đại diện bởi -0 và +0 |
|  | Trình bày về toán tử ++,-- và – trong JavaScript? | ++ là tăng lên một đơn vị (x++ tương đương với x=x+1)  -- là giảm đi một đơn vị (x-- tương đương với x=x-1)  - phép trừ giữa 2 toán hạng |
|  | Phân biệt ++a và a++? | ++a và a++ đều tăng lên 1 chỉ khác nhau khi  Vd: a=2  c=++a thì c=3 và a=3  b=a++ thì b=2 và a=3 |
|  | Câu lệnh điều kiện/rẽ nhánh là gì? | Là câu lệnh làm thay đổi luồng thực thi của một chương trình |
|  | Có những loại câu lệnh điều kiện nào? | If và switch-case |
|  | Trình bày về câu lệnh if? | - if(biểu thức điều kiện){  Nếu biểu thức điều kiện trả về true thì thực hiện chương trình trong này còn false thì bỏ qua if chuyển xuống câu lệnh ngay sau if  }  - If(biểu thức điều kiện){  Chương trình thực thi khi biểu thức dk là true  }  else{  chương trình thực thi khi biểu thức điều kiện là false  } |
|  | Trình bày về câu lệnh switch? | Là một cấu trúc điều kiện cho phép lựa chọn thực thi các khối lệnh khác nhau dựa trên kết quả việc so sánh.  Switch(biểu thức){  Case a:  Khối lệnh thực thi  Break;  Case a:  Khối lệnh thực thi  Break;  Default:  Khối lệnh  }  Nếu biểu thức so sánh với từng trường hợp case nếu có giá trị true thì khối lệnh sau nó sẽ được thực hiện và xong sẽ break thoát khỏi khối switch -case. Nếu sai nó sẽ so sánh với case tiếp theo.  Nếu không có case nào phù hợp thì nó sẽ thực hiện lệnh trong default |
|  | Tại sao lại nói else là một tùy chọn (optional) của câu lệnh if? | Tại vì else là thực thi chương trình khi điều kiện sai của biểu thức nên nó là một lựa chọn khác của if |
|  | Khi nào nên sử dụng câu lệnh switch? | Khi có số trường hợp cần xử lý lớn hơn 3 thì nên sử dụng switch – case. |
|  | Câu lệnh break dùng để làm gì? Nếu một case mà không có break thì chương trình chạy thế nào? | - Break để thoát khỏi khối lệnh  - Khi một case không có break thì nó thực hiện case tiếp theo ngay sau nó |