BỘ CÂU HỎI PHỎNG VẤN MODULE 1

BOOTCAMP PREPARATION 2.0

**1. Hệ thống các kiến thức đã học ở Module1**

- Học cách học

- Nhập môn tư duy lập trình

- Mô tả thuật toán bằng pseudo code và flowchart

- Git

- HTML Form Table

- Tổng quan JavaScript

- Biến kiểu dữ liệu và toán tử

- Cấu trúc điều kiện

- Vòng lặp

- Mảng

- Hàm

- Lập trình hướng đối tượng

**2. Giải thuật/Thuật toán là gì**

Thuật toán/giải thuật bao gồm các chỉ thị để giải quyết một vấn đề

Có các cách giải thuật thông dụng: Pseudo code, Flowchart, ngôn ngữ lập trình

**3. So sánh việc trình bày giải thuật bằng pseudo-code và flowchart?**

-1-Pseudo code (mã giả) Liệt kê tuần tự các bước bằng ngôn ngữ tự nhiên để biểu diễn

thuật toán

• Ưu điểm

• Đơn giản, không cần kiến thức về cách biểu diễn (lưu đồ, ngôn ngữ lập trình)

• Nhược điểm

• Dài dòng, không cấu trúc

• Đôi lúc khó hiểu, không diễn đạt được thuật toán

-2- flowchart lưu đồ

Lưu đồ mô tả giải thuật bằng các sơ đồ hình khối, mỗi khối quy định một hành động.

**4. Tại sao lại cần trình bày giải thuật trước khi bắt tay vào code?**

- Để khái quát những việc cần làm, vừa tiết kiệm thời gian, vừa biết điểm bắt đầu và kết thúc khi code.

**5. Tại sao lại nên sử dụng flowchart hơn pseudo-code khi thể hiện giải thuật của bài toán phức tạp?**

Bởi vì các thông tin được trình bày dưới dạng flowchart vô cùng ngắn gọn, hiệu quả. Điều này giúp ích rất lớn trong việc truyền đạt thông tin

**6. Tại sao giải thuật lại cần hữu hạn các bước thực thi?**

- Trên thực tế, bất cứ sự vật sự việc gì cũng có sự hữu hạn. Trong giải thuật, việc giới hạn hay có điểm kết thúc các bước thực thi là cần thiết, vì sẽ có 1 lúc cũng phải kết thúc

**7. Trình bày ý nghĩa các ký hiệu sử dụng khi vẽ flowchart?**

O-van: điểm bắt đầu – kết thúc

HBH: nhập xuất dữ liệu

HCN: xử lí, các bước tính toán

Hình thoi: các quyết định và rẽ nhánh

0-> bộ nối hai phần trong chương trình

Mũi tên: dòng chảy

**8. Vẽ giải thuật giải phương trình bậc nhất?**

**9. Bạn biết gì về VCS – Version Control System (hệ thống quản lý phiên bản)?**

 Version Control System (VCS) là một hệ thống được sử dụng trong việc phát triển phần mềm, giúp lưu trữ và ghi nhận sự thay đổi của các tập tin, mã nguồn theo thời gian, nhờ đó hệ thống có thể theo dõi việc cập nhật sửa đổi mã nguồn của các thành viên trong suốt quá trình làm dự án.

Version Control System (VCS) – Hệ thống quản lý phiên bản mã nguồn

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 xoá mất

**10. Phân biệt hệ thống quản lý mã nguồn tập trung và hệ thống quản lý mã nguồn phân tán?**

Mô hình tập trung (Centralized version control - CVC)

Cách giải thích 1: Mô hình sử dụng kiến trúc Client-Server. Server đóng vai trò là một kho dữ liệu (Repository) duy nhất lưu trữ tất cả các tập tin, mã nguồn, thay đổi… Các client sẽ kết nối đến server để lấy phiên bản mới nhất của mã nguồn về máy (check out) và cập nhật tất cả các thay đổi lên server (commit).

Cách giải thích 2 :các VCS dạng tập trung thì source code của dự án sẽ được lưu trữ trên một kho tập trụng (hay kho trung tâm) trên một máy chủ. Mỗi lập trình viên muốn tạo ra sự thay đổi cho source code lưu trữ trong kho trung tâm trung (centralized repository) thì họ cần phải thực hiện một công việc trước đó là update source code trên máy tính của họ với kho tập trung trước sau đó mới được thay đổi.

Mô hình phân tán (Distributed version control - DVC)

VCS dạng phân tán thì mỗi lập trình viên sẽ có riêng một kho (repository) của mình và có thể tự do phát triển khác feature và tạo version mới (hay đúng hơn là một commit) mà không cần phải đồng bộ với kho trên server. Điều này sẽ giúp lập trình viên tập trung vào việc hoàn thành feature(tính năng) hay fix bug mà không cần lo về việc phải liên tục update code trên máy tính của họ với code trên server để giảm thiểu xung đột. Việc xử lý xung đột (nếu có) sẽ được thực hiện một lần duy nhất.

**11 Git là gì? Tại sao cần sử dụng git?**

giúp bạn lưu lại các phiên bản của những lần thay đổi vào mã nguồn và có thể dễ dàng khôi phục lại dễ dàng mà không cần copy lại mã nguồn rồi cất vào đâu đó… Và một người khác có thể xem các thay đổi của bạn ở từng phiên bản,

Là nơi lưu trữ mã nguồn, theo từng phiên bản cho phép nhiều người có thể ttruy cập phiên bản đó để làm việc chung.

**12** ***Git thuộc hệ thống quản  lý mã nguồn tập trung hay phân tán?***

Git là một hệ thống quản lí phiên bản phân tán

Phân tán: có 1 máy có 1 local riêng thì có 1 mã nguồn riêng khác nhau

**13 Repository là gì? Phân biệt LocalRepository và RemoteRepository?**

Thường được gọi ngắn gọn là repo

• Repository là nơi chứa toàn bộ mã nguồn

• Repository bao gồm toàn bộ các file và lịch sử của các file đó

• Repository chứa tất cả các commit

Local Repository : là ở trên máy của lập trình viên

Remote Repository: là máy chủ trên m chia sẽ (github)

**14 Đồng bộ giữa các repository (local và remote) để làm gì?**

 repository phân thành 2 loại là local và remote nên với những công việc bình thường thì có thể sử dụng local repository và thực hiện trên toàn bộ máy sẵn có. Khi muốn công khai nội dung công việc mà bản thân đã làm trên local repository, thì sẽ upload lên remote repository rồi công khai. Thêm nữa, thông qua remote repository cũng có thể lấy về nội dung công việc của người khác.

**15 Thao tác push của git để làm gì? Cú pháp câu lệnh?**

Đồng bộ mã nguồn của local repository lên remote repository

Có 2 cách push: Git push

Git push origin (nhánh)

**16 Thao tác pull của git để làm gì?**

Đồng bộ mã nguồn remote repository về local repository

**17 Thao tác clone của git để làm gì? Cú pháp câu lệnh?**

sao chép một repository về máy của lập trình viên clinet

git clone <web>

**18 Nêu 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 trong vùng stages

git commit ghi nhận các thay đổi

git push : đưa các thay đổi từ local repository lên remote repository

Git fetch: đồng bộ dữ liệu giữa remote repo và local repo

Git pull: kéo phiên bản, thay đổi mới nhất về local repo

**19 www là gì?**

-WWW là viết tắt của cụm từ World Wide Web (*dịch nhanh là* *Mạng lưới toàn cầu*), là một không gian thông tin nơi chứa các tài liệu và nguồn tài nguyên khác của website

Nơi lưu trữ những dữ liệu mà các máy tính đẩy lên, muốn truy cập vào dữ liệu nào đó thì chỉ cần có địa chỉ của www đó (1 trong rất nhiều thư viện của internet )

Internet là sự kết nối giữa nhiều máy tính lại với nhau có mạng lan

Kết nối giữa các máy tính của các máy tính

**20 HTML là gì?**

Là ngôn ngữ sử dụng để tạo ra các website

HyperText makeup language (ngôn ngữ đánh dấu siêu văn bản)

**21 Nêu tên các thẻ làm việc với văn bản trong tài liệu html?**

**<p>**  hiển thị đoạn văn bản

**<h1> >h6**  hiển thị tiêu đề

**<b>** in đậm văn bản

**<pre>** ghi văn bản thành nhiều dòng

**<i>** In nghiêng chữ viết

**<a>** gán các liên kết cho văn bản

**22 Thẻ div dùng để làm gì? <div> khác gì <span>**

div – viết tắt của division – là thẻ được dùng để nhóm nhiều phần tử HTML lại với nhau.. Thẻ <div> là thẻ đánh dấu một khối (block) gồm nhiều thẻ khác trong khối. Như tên gọi, thẻ <div> giúp các tài liệu HTML được chia thành các phần (khối) riêng biệt.  Thẻ div được dùng để nhóm các phần tử lại với nhau trong các khu vực của website.

 thẻ <span> được thêm vào đoạn văn nhưng không làm thay đổi thuộc tính cũng như hiển thị của đoạn văn đó. Vì vậy chúng được gọi là thẻ trung tính trong HTML. Sự thay đổi của chúng chỉ được thay đổi khi chúng ta tác động chúng bằng CSS. Một điều quan trọng nữa là thẻ <span> là một thẻ inline. (khi khai báo vào nội dung thì nó sẽ vẫn nằm chung một dòng với các thẻ khác trong văn bản)

**23 URL là gì? Phân biệt url tuyệt đối và url tương đối?**

Uniform Resource Locator (viết tắt URL, nghĩa tiếng Việt: định vị tài nguyên thống nhất, được gọi một cách thông thường là một địa chỉ web

Sự khác biệt chính giữa URL tuyệt đối và URL tương đối là, URL tuyệt đối là địa chỉ hoàn chỉnh trỏ đến tệp hoặc tài nguyên, trong khi URL tương đối trỏ đến tệp liên quan đến thư mục hoặc tệp hiện tại. URL tuyệt đối chứa nhiều thông tin hơn URL tương đối, nhưng việc sử dụng URL tương đối dễ dàng hơn nhiều vì chúng ngắn hơn và dễ di chuyển hơn.

**24 Siêu liên kết là gì?**

Hyperlink hay còn gọi là liên kết(link)  Một siêu liên kết trỏ đến toàn bộ tài liệu hoặc một phần tử cụ thể bên trong một tài liệu.

**25 Muốn liên kết đến một phần trên cùng trang thì làm thế nào?**

Đầu tiên ta tạo các book mark (1 điểm đánh dấu vị trí trên trang web, bằng cách thêm thuộc tính id=”tên bookmark”

Tạo liên kết trong sau khi tạo 1 liên kết sau đó ta thêm tên book mark chỉ định đến 1 vị trí trang web

**26 Muốn liên kết đến một phần trên trang khác thì làm thế nào?**

<a href="URL">nội dung đại diện cho cái liên kết</a>

**27 Phân biệt internal link và external link?**

External link  Liên kết ngoài là các siêu liên kết trỏ đến (target) bất kỳ miền nào khác với tên miền liên kết tồn tại trên (nguồn)

Internal links (liên kết nội bộ) là các liên kết đi từ 1 trang trên 1 tên miền đến 1 trang khác cùng 1 tên miền

External Links = liên kết trỏ đến tên miền riêng

Internal Links = các liên kết trỏ đến nội dung trong cùng một tên miền

Internal: chỉ những liên kết từ trang này sang trang khác trong cùng 1 miền

External: chỉ những liên kết ngoài trang web, từ trang web bên ngoài dẫn đến trang web của bạn

**28 Có mấy loại danh sách? (ul, ol, li)**

-Có 3 kiểu danh sách (list type)

  +Kiểu sắp xếp (ordered list)

  +Kiểu kh sắp xếp ( unordered list)

  +Kiểu danh sách mô tả (description list)

-Kiểu sắp xếp (Ordered List): Là kiểu hiển thị một danh sách mà các mục con của nó được sắp xếp theo thứ tự bằng số hoặc chữ cái.

-Kiểu không sắp xếp (Unordered List): Là kiểu hiển thị danh sách mà các mục con của nó sẽ không được sắp xếp theo thứ tự mà chỉ được đánh dấu bằng một ký tự đặc trưng.

-Kiểu mô tả (Description List): Là kiểu hiển thị danh sách mà các mục con của nó sẽ không được đánh dấu thứ tự, nhưng sẽ có kèm theo một đoạn miêu tả.

**29 Muốn hiển thị danh sách có thứ tự bắt đầu C. D. E… thì làm thế nào?**

Dùng kiểu sắp xếp (ordered list)

<ol type=”A”> <li> </li> </ol>

 dùng thuộc tính start của thẻ ol,xác định danh mục bắt đầu đc đánh số hay thứ tự bao nhiêu,giá trị start là một

chữ số. cú pháp <ol type="A" start="3"> : giá trị bắt đầu là "C".

**30 Form hay biểu mẫu dùng để làm gì?**

Biểu mẫu (form) được yêu cầu để thu thập các loại dữ liệu đầu vào khác nhau của người dùng như : tên,tuổi… và gửi về server

Trong thẻ <form> sẽ có những hành động chỉ định tạo các sự kiện

**31 Thuộc tính action của form dùng để làm gì?**

Thuộc tính action có giá trị là 1 URL(siêu liên kết) quy định nơi gửi dữ liệu đến

**32 Phân biệt 2 method là get và post?**

Thuộc tính method được dùng để quy định cách thức dữ liệu gửi lên sever

GET

Dữ liệu được gửi sẽ hiển thị lên thanh địa chỉ

Không nên sử dụng những dữ liệu nhạy cảm

Thường đc sử dụng vs dữ liệu nhỏ

POST

Dữ liệu gửi đi KHÔNG hiển thị lên thanh địa chỉ mà gửi lên SERVER

Sử dụng để gửi dữ liệu nhạy cảm

Khônng hạn chế dung lượng dữ liệu

**33 Phân biệt các nút bấm có type là: button, submit, reset?**

Button tạo ra 1 nút xử lí các thao tác người dung

Submit tạo ra 1 nút nhấn, gửi các dữ liệu lên sever

Reset để xóa dữ liệu ng dùng vừa nhập vào form

**34 Muốn submit form đến trang CodeGym.vn thì làm thế nào?**

<form action=”codegym.vn”>

<button type=’submit’ > </button>

**35 Nếu muốn nối 3 ô trên cùng 1 hàng thì làm thế nào?**

Colspan=”3”

Table tr td col=”3”

**36 Nếu muốn nối 2 ô trên cùng 1 cột thì làm thế nào?** Rowspan =”2”

**37 Phân biệt thuộc tính cellpadding và thuộc tính cellspacing?**

**Cellspacing**: Tạo khoảng cách giữa hai ô hay còn gọi là các viền

**Cellpadding**: tạo giữa nội dung với khoảng cách trong ô so với đường viền

**38 Trình bày các cách nhúng javascript vào 1 tài liệu html?**

Viết mã Javascript bên trong thẻ <script></script>

• Sử dụng file .js và nhúng vào trang web

• Sử dụng file .js từ bên ngoài

• Viết trực tiếp mã JavaScript trong thẻ html

**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?**

Đúng. Vì nó là cầu nối, để các client tập hợp ra các hướng dẫn thực thi đưa lên sever

**40 Phát biểu “JavaScript phân biệt chữ hoa chữ thường ” đúng hay sai?** (đúng)

**41 Trình bày về quy tắc  camelCase và PascalCase.**

Camel : Từ đầu viết thường từ sau viết  Hoa vd: nhatTruong doHungHuy

PascalCase: viết hoa các chữ cái đầu vd: NhatTruong PhanVanThien DoHungHuy

Underscore: sử dụng các gạch chân giữa các từ vd: nhat\_truong chữ viết snake do\_hung\_huy

**42 Biến là gì? Phân biệt biến cục bộ (local) và biến toàn cục (global)?**

Biến là một giá trị dữ liệu có thể được thay đổi trong chương trình. Giá trị dữ liệu có thể là loại chuỗi (string), số tự nhiên (interger), số trôi nổi (float), logic (boolean), mảng (array), đối tượng (object) hoặc dữ liệu rỗng (NULL).

**biến toàn cục** là biến được khai báo bên ngoài hàm và được sử dụng ở tất cả các vị trí,

**biến** **cục bộ** là biến khai báo ở trong hàm và chỉ sử dụng được trong nội bộ của hàm đó thôi, sau khi hàm thực thi xong thì ngay lập tức nó sẽ bị xóa khỏi bộ nhớ

**43 Phân biệt: var, let, const?**

Var ở phạm vi toàn cục có thể gán lại và khai báo lại

Let phạm vi khối có thể gán nhưng không thể khai báo lại

Const phạm vi khối không thể gán và cũng kh thể khai báo lại

**44 Phát biểu “JavaScript là ngôn ngữ định kiểu yếu hay động” nghĩa là gì? Trình bày?**

Là kiểu yếu, vì phụ thuộc vào không cần phải khai báo kiểu của các biến trước khi dùng. Kiểu dữ liệu của biến phụ thuộc vào giá trị chúng ta gán cho biến.

Mạnh: số gán là số. Định hình ra kiểu dữ liệu

Yếu : lúc khai báo có thể kiểu dử liệu khác

**45 Nêu tên các kiểu dữ liệu nguyên thủy trong JavaScript?**

Number: Các số nguyên hoặc số thực. Ví dụ: 5 hoặc 5.05

String: là các text như “Các kiểu dữ liệu trong JavaScript”, text có thể có một hoặc nhiều ký tự.

Boolean: chỉ có 2 giá trị là true hoặc false.

Undefine:  là các giá trị không xác định.

Null: đơn giản là không có giá trị nào cả.

Empty : có vị trí có thứ tự, nhưng kh có giá trị

**46 Trình bày về toán tử typeof? Khi khai báo “let x=10;” thì x có kiểu là gì?**

Từ khóa typeof được dùng để kiểm tra kiểu dữ liệu của một biến nào đó (hoặc một giá trị nào đó).

Khi khai báo “let x=10;” thì x có kiểu number

**47 Trình bày các cách đưa thông báo trong JavaScript?**

 alert, confirm, prompt.

Thông báo hiển thị cảnh báo (alert) cho người dùng

Thông báo yêu cầu người dùng xác nhận (confirm) một cái gì đó

Và thông báo yêu cầu người dùng nhập thông tin (prompt).

**48 Trình bày về các loại toán tử trong JavaScript?**

. Toán tử toán học

• Toán tử gán

• Toán tử cộng chuỗi

• Toán tử so sánh

• Toán tử logic

• Toán tử typeof

**49 Phân biệt giữa 2 toán tử == và ===?**

== so sánh value

=== so sánh value và type date

Chỉ so sánh value

=== so sánh cả value và cả kiểu dử liệu của nó luôn

**50 Phân biệt giữa 2 toán tử && và || trong JavaScript?**

&& và

|| hoặc

Bổ sung: ! phủ định vd: kp a+b -> !(a+b)

**51 Kết quả biểu thức +0===-0 là true hay false?**

true . Số 0 không có âm hay dương;

**52 Trình bày về toán tử ++,--  trong JavaScript?**

++ tăng 1 giá trị -- giảm đi 1 giá trị

**53 Phân biệt ++a và a++?**

++a tính luôn trên giá trị ban đầu.

a++ tạo ra bản copy của giá trị ban đầu.

**54 Câu lệnh điều kiện/rẽ nhánh là gì?**

Câu lệnh điều kiện còn được gọi là câu lệnh ra quyết định (decisionmaking)

• Cho phép thay đổi luồng thực thi của chương trình

• Lựa chọn thực thi một khối lệnh dựa trên việc đánh giá một điều kiện cho trước

• JavaScript hỗ trợ các câu lệnh điều kiện:

**55 Có những loại câu lệnh điều kiện nào?** If else, if lồng nhau , if bậc thang, switch

**56 Trình bày về câu lệnh if?**

If (true) {

 Hàm xử lí

}

**57 Trình bày về câu lệnh switch?**

switch-case là một cấu trúc điều kiện cho phép lựa chọn thực thi các khối lệnh khác nhau dựa trên kết quả của việc so sánh. Switch dùng để xét  những trường hợp từ trường hợp 1 . Truyền vô  cái điều kiện, r nó sẽ xét tất cả các điều kiện nếu điều kiện nào đúng với điều kiện đó thì nó sẽ brek, còn k có case nào thỏa mãn thì defaut

**58 Tại sao lại nói else là một tùy chọn (optional) của câu lệnh if?**

Vì trường hợp điều kiện ở if tức là (true) khi đúng thì nó sẽ xử lí biểu thức còn không thì sẽ là else tức là option khác. Để việc tuần hoàn của giải thuật

**59 Khi nào nên sử dụng câu lệnh switch?**

Đưa ra nhiều trường hợp, tình huống switch-case so sánh giá trị của một biến với lần lượt từng giá trị một, nếu có giá trị phù hợp với biến thì khối lệnh tương ứng sẽ được thực thi

**60 Câu lệnh break dùng để làm gì? Nếu một case mà không có break thì chương trình chạy thế nào?**

Break dùng để phân tách các trường hợp. khi gặp break thi sẽ kết thúc thực thi khối switch-case nếu không có break thì các trường hợp sẽ bị nối nhau

**61 Mệnh đề default có bắt buộc trong câu lệnh switch?**

Không bắt buộc, đây là khối lệnh mặc định nếu không chọn được trường hợp nào

**62 Trình bày về toán tử 3 ngôi?**

Condition (biểu thức đánh giá xem có đúng hay không) ?  “biểu thức đúng” : “biểu thức sai”

**63 Trình bày về sự kiện và trình quản lý sự kiện trong javascript?**

Sự kiện là một hành động tác động lên các đối tượng HTML, qua đó ta có thể bắt được sự kiện và yêu cầu javascript thực thi một chương trình nào đó

1 onclick Xảy ra khi click vào thẻ HTML

2 ondbclick Xảy ra khi double click vào thẻ HTML

3 onchange Xảy ra khi giá trị (value) của thẻ HTML đổi. Thường dùng trong các đối thẻ form input

4 onmouseover Xảy ra khi con trỏ chuột bắt đầu đi vào thẻ HTML

5 onmouseout Xảy ra khi con trỏ chuột bắt đầu rời khỏi thẻ HTML

6 onmouseenter Tương tự như onmouseover

**64 Sự kiện onchange phát sinh khi nào?**

Xảy ra khi giá trị (value) của thẻ HTML đổi. Thường dùng trong các đối thẻ form input

**65 Sự kiện onclick phát sinh khi nào?**

Xảy ra khi click vào thẻ button hoặc bất kì chổ nào đã setting sẵn

**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?**

Confirm

**67 Phân biệt onmousevover và onmouseout, onmousemove?**

4 onmouseover Xảy ra khi con trỏ chuột bắt đầu đi vào thẻ HTML

5 onmouseout Xảy ra khi con trỏ chuột bắt đầu rời khỏi thẻ HTML

8 onmousemove Xảy ra khi con chuột di chuyển bên trong thẻ HTML

**68 Phân biệt onkeyup, onkeydown và onkeypress?**

Onkeyup: Xảy ra khi bạn gõ phím nhưng lúc bạn nhã phím ra sẽ được kích hoạt

Onkeydown Xảy ra khi gõ một phím bất kì vào ô input

Onkeypress Xảy ra khi bạn nhấn môt phím vào ô input

Press đè vào nó sẽ chạy

Up đè vào thả ra

Down nhấn vào cái là chạy luôn

**69 Phân biệt onfocus và onblur?**

Onblur Xảy ra khi con trỏ chuột rời khỏi ô input

Onfocus xảy ra khi trỏ chuột vào ô input

Onfocus: đưa vào ô input

Onblur:  rút ra ô input

**70 Sự kiện onload phát sinh khi nào?**

Xảy ra khi đối tượng đã được tải thường nằm trong thẻ <body>

Xảy ra khi trang html nớ được tải lên trang web, hắn sẽ chyaj kèm theo 1 cái chức năng chi đó bị load chậm  ảnh huởng tới lỗi

**71 Phân biệt innerHTML và innerText?**

innerText: đặt hoặc trả về nội dung văn bản của nút được chỉ định và tất cả các nút con của nó

Thuộc tính innerHTML: đặt hoặc trả về nội dung HTML (HTML bên trong) của một phần tử.

**72 Nếu muốn vô hiệu hóa (disabled) một button khi click vào 1 checkbox thì làm thế nào?**

Thuộc tính disabled là 1 thuộc tính Boolean. Khi xuất hiện thì nó chỉ định nút button đã được tắt <button type=”checkbox” disabled> </button>

**73 Phân biệt thuộc tính readonly và disabled của html?**

-Thuộc tính **readonly**

Thuộc tính readonly xác định rằng trường dữ liệu đầu vào chỉ có thể đọc (không thể bị thay đổi). Khôgn thể copy

-Thuộc tính **disabled**

Thuộc tính disabled xác định rằng trường dữ liệu đầu vào bị vô hiệu hóa, tức là không thể sử dụng, không thể click vào và giá trị cũng không được gửi khi biểu mẫu gửi đi. {không thể copy luôn}

**74 Vòng lặp là gì?**

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

**75 Nêu các loại vòng lặp cơ bản trong JavaScript?** Vòng lặp for, while, do- while

**76 Phân biệt vòng lặp for, while, do…while?**

For thực hiện số lần lặp mà đã cho trước

While thực hiện số lần lặp mà ta kh hề biết là bao nhiêu

Do while sẽ thực hiện khối lệnh ở “do” trước, sau đó mới xét điều kiện “while”, nếu thỏa mãn, thì nó sẽ lặp lại

**77 Khi nào nên sử dụng vòng lặp do…while?**

Vòng lặp do while ứng dụng trong những trường hợp mà ta xác định được điều kiện để lặp, cho dù thỏa mãn điều kiện hay không, thì bản thân nó vẫn thực hiện khối lệnh tối thiểu 1 lần {Sử dụng do while khi cần làm công việc rồi mới kiểm tra điều kiện, ví dụ như kiểm tra nhập giá trị hợp lệ

**78 Khi nào nên sử dụng vòng lặp for?**

Khi xác định được số vòng lặp, biết điểm bắt đầu và điểm dừng

**79 Khi nào nên sử dụng vòng lặp while?**

Vòng lặp while luôn xét điều kiện cho trước sau đó mới thực hiện khối lệnh

**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?**

do while vì người nhập sẽ được nhập vào trước dữ liệu, sau đó ta sẽ xét điều kiện nếu đó là số chẵn thì break, còn không thì continue;

**81 Yêu cầu người dùng tính tổng các số chẵn từ 10 đến 100 thì nên dùng vòng lặp loại nào?** Vòng lặp for

**82 Trình bày về vòng for thiếu? For (;;)**

Các biểu giá trị ở trong có thể đưa vào bên trong for hoặc bên ngoài

**83 Nếu muốn hiển thị các cặp số nguyên dương có tổng là 10 thì viết thế nào?**

Dùng vòng lặp for chạy từ 0 -> 10, cộng các cặp phần tử nguyên dương trong đó, nếu = 10 thì trả kết quả về, kết quả sẽ được lưu vào biến total

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

Mình sử dụng 2 vòng lặp lòng nhau => dùng mảng trong mảng hiển thị hình sao

Let i = 0 i < chiều dài mảng i ++

Trong sẽ là j, nếu muốn vẽ tam giác vuông thì j=i; j < i; j++

Document.write \*

**85 Trình bày về cách chạy của vòng for lồng nhau?**

Đầu tiên vòng for1 sẽ chạy lần 1, for 2 sẽ chạy hết số vòng lặp của nó, rồi ra lại vòng for1 chạy lần 2, cứ tiếp diễn như vậy cho đến khi for1 kh còn thỏa mãn điều kiện

Câu lệnh khởi tạo, điều kiện , biểu thức thực thi, biểu thức để sai

Vòng ngoài chạy đầu tiên, xong sẽ lặp vòng trong

R lại ra vòng ngoài

Như ví dụ đếm số người trong quân ngũ

Hàng đầu tiên là vòng ngoài từng người trong đó là vòng 2 rồi tiếp tục hàng thứ 2

**86 Phân biệt giữa câu lệnh break và continue?**

Break; khi gặp điều kiện sẽ trả về kết quả điều kiện và dừng hẵn tại đó

Còn continue thì bỏ qua điều kiện và chạy tiếp

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

for (let x= 0; x< 100; x++) {

     if (x %5 == 0){

        continue;

     }

console.log(x)

}

**88 Mảng là gì?**

Mảng là 1 biến (đặc biệt) có thể lưu trữ nhiều giá trị thay vì chỉ 1 giá trị thông thường.

Mỗi giá trị trong mảng là 1 phần tử. Mảng gồm: phần tử, chỉ số, độ dài

**[element, index,length]**

**89 Trình bày cú pháp khai báo mảng trong JavaScript?**

Var array = [] của javaScript

Var array = new Array()  của EMAs script

**90 Chỉ số của mảng là gì? Chỉ số của mảng xuất phát từ 0 hay 1?**

Chỉ số hay đc gọi là index, là thứ tự  trong mảng, bắt đầu bằng số 0

**91. Kích thước của mảng là gì?**Là độ dài của mảng, từ điểm xuất phát đến mảng -1

**92 Phân biệt push() và unshift()?**

Push() cho phép ta thêm một phần tử vào phía sau mảng

Còn unshift thì thêm 1 phần tử vào đầu mảng

**93 Phân biệt push() và pop()?**

Push() thêm 1 phần tử vào cuối mảng

Pop() xóa 1 phần tử ở cuối mảng

**94 Phân biệt shift() và unshift()?**

Shifft() xóa 1 phần tử đầu mảng

Unshift theme 1 phần từ đầu mảng

**95 Nếu muốn xóa phần tử cuối cùng của mảng dùng hàm nào?**Hàm pop()

**96 Nếu xóa phần tử đầu tiên của mảng dùng hàm nào?** Shifft()

**97 Muốn sắp xếp mảng thì dùng hàm nào?**

Sort() reverse()

Sort() sắp xếp theo thứ tự a-z và theo  chiều tăng dần

Reverse() đảo ngược thứ tự trong mảng

**98 Có một mảng gồm tên của các bạn học viên. Nếu muốn sắp xếp tên này theo thứ tự alphabet làm thế nào?** Dùng phương thức sort()

**99 *Thao tác duyệt mảng là gì? Triển khai thao tác bằng code?***

là thao tác duyệt qua tất cả các phần tử có trong mảng theo yêu cầu của bản thân

Duyệt mảng không điều kiện. in ra các giá trị trong mảng

For(;;) {console.log(a[i])}

duyệt mảng có điều kiện. in ra các giá trị thỏa mãn điều kiện

for() {if{

}}

**100 Thao tác duyệt mảng 2 chiều?**

Là việc ta dùng 2 vòng lặp for lồng nhau để duyệt qua mảng 2 chiều

**101 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 1 phần tử gồm

A[i][j]

Trong đó i thể hiện duyệt qua hàng

j thể hiện duyệt qua cột

**102 *Viết code tính tổng các phần tử của ma trận số (ý tưởng)?***

Duyệt hai lần for để duyệt qua các hàng và cột, sau đó tạo 1 biến total rồi cộng vào đó

**103 *Viết code tính tổng các ô trên cùng hàng của ma trận số (ý tưởng)?***

Duyệt hai lần for, ở vòng for ngoài (vd: duyệt hàng số 3) let i= 0; if ( i = 2) thì duyệt vòng for ở trong rồi cộng vào total

**104 *Viết code tính tổng các ô trên cùng cột của ma trận số (ý tưởng)?***

- Duyệt vòng for ngoài (vd : tính tổng các cột 2) cho biến i vòng ngoài chạy tới arr.length - 1, và giữ nguyên số cột [2].

**105. *Hàm (function) là gì?***là một nhóm các câu lệnh thực hiện một nhiệm vụ nhất định

**106 *Phân biệt giữa tham số hình thức (formal arguments) và tham số thực (actual arguments)?***

* Các tham số (đối số) được truyền trong một lệnh gọi hàm được gọi là các đối số thực tế.

  Các đối số này được định nghĩa trong hàm gọi. Vd: sum (a,b) -> sum(2,3) ở đây 2,3 là các tham số thực.

* Các tham số hình thức là các tham số được định nghĩa trong hàm và chỉ dùng cục bộ trong hàm đó và chỉ thuộc hàm đó.  Vd: sum(a,b) { let a = 2; b = 3; c = a+b  return c}

**107 Gọi hàm (invoke) là gì?**

• Gọi (call hoặc invoke) phương thức là cách để thực thi một hàm đã được định nghĩa trước đó

• Khi gọi hàm thì cần truyền đối số vào

Hàm có thể được gọi bởi: Các sự kiện (như click khi nhấn một button). Một lệnh hay khối lệnh JavaScript. Tự động

**108 *Phân biệt hàm gọi (calling function) và hàm được goi (called function)?***

Hàm gọi (calling function) là việc ta gọi 1 hàm khác nữa trong hàm

Vd: function 1() {

Ab = function a()}

Function a(){}

**109 *Phân biệt truyền tham trị và truyền tham chiếu?***

Tham trị là việc ta truyền giá trị vào hàm số (giá trị bên ngoài k thay đổi)

Tham chiếu là truyền biến vào. (giá trị hàm thay đổi thì biên bên ngoài theo)

Tham chiếu: tức là sẽ copy địa chỉ nhớ ra thành 1 địa chỉ mới nhưng cả 2 địa chỉ cùng trỏ vô 1 vùng nhớ duy nhất

Nếu thay đổi dử liệu 1 cái thì cái còn lại cũng sẽ bị thay đổi

Tham trị:  tạo ra 1 vùng nhớ mới mượn giá trị

Khi thay đổi địa chỉ này thì địa chỉ kia k bị thay đổi

**110. *Câu lệnh return dùng để làm gì?*** Return là việc trả về kết quả của hàm đã định nghĩa

**111 Có được phép viết “return;”?** Vẫn được, nhưng nó sẽ trả về kết quả rỗng

**112 Phân biệt return và break?**

- **Return** và **break** đều là câu lệnh có thể kết thúc 1 hàm hoặc 1 vòng lặp.   
Nhưng ở return có thể trả về kết quả còn break thì không.

Return hoạt động ở hàm

**Break** hoạt động ở vòng lặp for và switch

Break thực hiện tới 1 đk nào đó đúng thì bỏ qua câu lệnh đó, mà thực hiện else luôn

Còn return sẽ thực hiện ngược lại.

**113 *Hàm thực hiện việc kiểm tra nên trả về giá trị không? Nếu có, nên trả về gì?***

*-* có, nên trả về kết quả của việc kiểm tra.

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

Việc hoán vị chỉ diễn ra trong hàm, và chỉ hoán vị giá trị, còn địa chỉ lưu trữ vẫn được giữ nguyên, khi kết thúc và ra khỏi hàm thì khi gọi các địa chỉ lưu trữ, thì vẫn sẽ lấy giá trị  tham số gốc

**115 Trình bày về OOP?**

OOP hay còn gọi là lập trình hướng đối tượng Object Oriented Programing là một phương thức lập trình dựa trên lớp và đối tượng.OOP thường tập trung vào các đối tượng thao tác hơn là logic để có thể thao tác chúng.

**116. Phân biệt giữa lập trình hướng đối tượng và lập trình hướng thủ tục/chức năng?**

Lập trình hướng thủ tục (POP) và Lập trình hướng đối tượng (OOP),sử dụng ngôn ngữ cấp cao.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **OOP**   |  |  | | --- | --- | | **CƠ SỞ ĐỂ SO SÁNH** | **POP** | |
| **Căn bản** | **Thủ tục / Cấu trúc định hướng.** | **Hướng đối tượng.** |
| **Hướng tiếp cận** | **Từ trên xuống** | **Từ dưới lên** |
| **Nền tảng** | **Trọng tâm chính là "làm thế nào để hoàn thành nhiệm vụ" tức là về thủ tục hoặc cấu trúc của một chương trình.** | **Trọng tâm chính là "bảo mật dữ liệu". Do đó, chỉ các đối tượng được phép truy cập các thực thể của một lớp.** |
| Division | Chương trình lớn được chia thành các đơn vị gọi là chức năng. | Toàn bộ chương trình được chia thành các đối tượng. |
| Chế độ truy cập thực thể | Không có specifier truy cập. | Có tình xác định truy cập là "public", "private", "protected". |
| Overloading or Polymorphism | Neither it overload functions nor operators. | It overloads functions, constructors, and operators. |
| **Thừa kế** | **Không hỗ trợ thừa kế** | **Thừa kế được hỗ trợ ở ba trạng thái "public", "private", "protected"** |
| **Bảo mật** | **Không có cách ẩn dữ liệu thích hợp, vì vậy dữ liệu không an toàn** | **Dữ liệu được ẩn trong ba chế độ"public", "private", "protected" do đó bảo mật dữ liệu tăng lên.** |
| Chia sẻ dữ liệu | Dữ liệu toàn hệ thống được chia sẻ giữa các chức năng trong chương trình. | Dữ liệu được chia sẻ giữa các đối tượng thông qua các chức năng. |
| **Function và Class** | **Không có khái niệm về Function và Class** | **Các lớp hoặc hàm có thể trở giao tiếp với nhau với từ khóa. (tùy các ngôn ngữ khác nhau thì từ khóa sẽ khác nhau)** |
| Virtual classes hoặc virtual function | Không có khái niệm về lớp ảo | Khái niệm về chức năng ảo xuất hiện trong quá trình kế thừa. |
| Các ngôn ngữ thường sử dụng | TC, VB, GIỚI THIỆU, Pascal | C ++, JAVA, [VB.NET](http://vb.net/), C # .NET, Ruby,... |

**117 Phân biệt giữa object và class?**

Object là đối tượng

Class lớp

Tập hợp của nhiều object lại thành class , vậy object là con của class.

**Class** là một thực thể xác định đối tượng sẽ có những hành vi nào và những đối tượng sẽ có.

Một **Object** (đối tượng) nó chứa trong đó bao gồm các **method** (phương thức) và **properties** (thuộc tính) để tạo ra một kiểu dữ liệu hữu ích.

Một **Class** là một **Blueprint** (kế hoạch) hay **Prototype** (nguyên mẫu) xác định biến và các phương thức (hay function) chung với tất cả các đối tượng cùng loại.

Một **Object** (đối tượng) là một cụ thể, thể hiện của một **Class**.

**118 Hai khái niệm “Môn học” và “Môn học JavaScript” thì đâu là object?**

Môn học javascript là một object

**119 Mô tả về sơ đồ lớp?**

Class = {

contrucster (value,...)

This.value;

}

class <class\_name> {

    field;

    method;

}

* class: Là từ khóa để tạo class
* <class\_name>: Là tên class bạn định tạo nó sẽ được viết theo quy tắc PascalCase (Viết liên và viết HOA chữ cái đầu tiên của từng từ)
* field: Các biến đối tượng
* Method Class Student {
* Contrucster (name,age,adress,corse) {
* This.name = name;
* This.age = age;
* This.adress = adress;
* this.corse = corse;
* }
* }
* Var student1 = new Student(truong,23,84KTD,C1121I1);
* console.log(student1): Các phương thức

**120 Vẽ sơ đồ lớp của lớp Học viên?**

**121 Thuộc tính là gì? Nên dùng từ loại nào để biểu diễn thuộc tính?**

**Thuộc tính (properties)**

Thuộc tính trong lớp là lớp mô tả đặc điểm của đối tượng.

Những từ ngắn những vẫn mỏ tả đúng chính xác đặc điểm của đối tượng.

**122 Phương thức là gì? Nên dùng từ loại nào để biểu diễn phương thức?**

Phương thức trong lớp là những hành vi của đối tượng

**123. Cú pháp định nghĩa 1 class trong JavaScript?** Class (ClassName: đặc theo PascalCase)

**124 Cú pháp tạo 1 đối tượng từ class?**

class <class\_name> {

    field;

    method;

}

* class: Là từ khóa để tạo class
* <class\_name>: Là tên class bạn định tạo nó sẽ được viết theo quy tắc PascalCase (Viết liên và viết HOA chữ cái đầu tiên của từng từ)
* field: Các biến đối tượng
* method: Các phương thức

**125. Trình bày một số class có sẵn trong JavaScript mà em biết?** Math, String, Date:

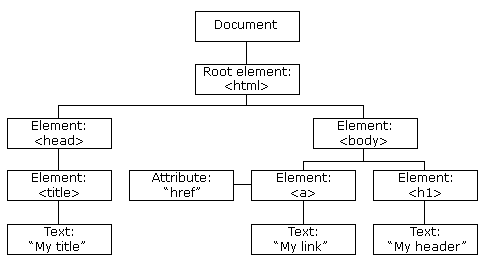
**126 Cú pháp truy cập đến 1 thuộc tính của đối tượng?** this.name;

**127 Cú pháp truy cập đến 1 phương thức của đối tượng?** this.info();

**128 Từ khóa this dùng để làm gì?** This. ý ám chí đối tượng

**129 Trình bày về DOM (Document Object Model – Mô hình đối tượng tài liệu)?**

DOM là 1 thư viện khởi tạo nên HTML. được khởi tạo bởi W3C.DOM được dùng để truy xuất và thao tác trên các tài liệu có cấu trúc dạng HTML hay XML bằng các ngôn ngữ lập trình thông dụng như Javascript, PHP…



Document object model

Dùng để truy cập cá phần tử có trong trang web HTml hoặc các thuộc tính của HTml có trong trang web

Dùng có thể thêm bớt, thay đổi 1 thẻ html

Thêm bớt, thay đổi 1 thuộc tính của thẻ html

Truy cập truy suất các thẻ html để lấy ra cái giá trị có trong html hoặc các thuộc tính của html

**130 Thẻ <canvas> dùng để làm gì?**

Phần tử HTML <canvas> được sử dụng để vẽ đồ họa một cách nhanh chóng, thông qua JavaScript. Phần tử <canvas> chỉ là vùng chứa cho đồ họa. Bạn phải sử dụng JavaScript để thực sự vẽ đồ họa. Canvas có một số phương pháp để vẽ đường dẫn, hộp, hình tròn, văn bản và thêm hình ảnh.

**131 Vẽ hình chữ nhật trên canvas?**

<canvas id="myCanvas" width="200" height="100" style="display :inline-block; border: 2px solid #d3d3d3">

|  |  |
| --- | --- |
| let Rectangle = function(height, width) { |  |
|  | this.height = height; |
|  | this.width = width; |

|  |  |
| --- | --- |
| function runRectangle() { |  |
|  | let height = Number(document.getElementById('lengthRec').value); |
|  | let width = Number(document.getElementById('widthRec').value); |
|  | if (height && width) { |
|  | let rectangle = new Rectangle(height, width); |
|  | rectangle.drawRec(); |
|  | are = rectangle.getArea(); |
|  | perimeter = rectangle.getPerimeter(); |
|  |  |
|  | document.getElementById('showArea').innerHTML = `rectangular are ${are}`; |
|  | document.getElementById('showPerimeterOfARectangle').innerHTML = `Perimeter of a rectangle  ${perimeter}`; |
|  |  |
|  | } |

**132 Vẽ hình tròn trên canvas?**

arc(x,y, bán kính, điểm bắt đầu, điểm kết thúc)

**133 Vẽ text trên canvas?**

**134 Nếu muốn di chuyển một hình tròn trên canvas cần làm gì?**

Cần thay đổi vị trí trục X và trục y;

BOM là chữ viết tắt của (**B**rowser **O**bject **M**odel), hay còn gọi là các đối tượng liên quan đến trình duyệt browser. Mỗi browser sẽ có những đối tượng khác nhau nên nó không có một chuẩn chung nào cả. Tuy nhiên, để có tính thống nhất giữa các trình duyệt thì người ta quy ước ra các loại BOM sau:

* [window](https://freetuts.net/bom-window-trong-javascript-385.html)
* screen
* [location](https://freetuts.net/location-trong-javascript-386.html)
* [history](https://freetuts.net/bom-history-trong-javascript-387.html)
* navigator
* popup
* timing
* cookies

Đối tượng có những phương thức do mình tự tạo ra

Chủ động lập trình có những thuộc tính và kiểu dữ liệu mà mình mong muốn để đáp ứng những nhu cầu lập trình của mình

Function tên class ( các thuộc tính mà mình đặt ra)

Overloading:

Cùng dữ liệu khác kiểu tham số