**Question Interview**

1. **CSS**
2. ***Phân biệt border-box và content-box***

* 2 cái đều là thuộc tính của box-sizing, mặc định của box-sizing là content-box làm thay đổi kích thước của content, không bao gồm padding, border và margin.
* Còn border-box làm thay đổi kích thước content, padding và border nhưng không bao gồm margin.

***Trả lời đơn giản:*** content-box là default của box-sizing gồm các border padding + bên ngoài còn border-box là ngược lại nó sẽ cộng bên trong không ảnh hưởng đến bên ngoài, giải thích ngắn gọn dễ hiểu sẽ đc điểm đánh giá cao.

1. ***Fontawesome được làm từ svg vẽ từ các điểm vector***

* Fontawesome ngoài dùng svg thì còn cả png hay jpeg nhưng phạm vi CSS không đề cập tới nó. CSS nó sử dụng psudo element là before after để code.

***3. Bootstrap***

- Bootstrap là một Framework HTML, CSS, và JavaScript cho phép thiết kế phát triển các website hỗ trợ responsive.

- Vì boostrap là một thư viện Opensource nên sử dụng và config khá dễ, vd muốn config lại 1 button abc, ta chỉ cần thêm 1 class vào button abc, rồi CSS tùy biến.

**4. *Asynchronous là hàm bất đồng bộ***

- Asynchronous là bất đồng bộ vì ta sẽ không xác định đc thời điểm sẽ nhận về result data.

***5. Function scope***  
- Khi khởi tạo 1 function lúc đó đã khởi tạo 1 scope. Scope nằm trong phạm vi của function và các function nhỏ hơn.

***(tìm hiểu về let, var, const ví dụ minh hoạt cụ thể là loop for và while. nên cho keyword thì sẽ đc đánh giá cao)***

*Let: tạo ra một biến chỉ có thể truy cập được trong block bao quanh nó, khác với var - tạo ra một biến có phạm vi truy cập xuyên suốt function chứa nó.*

*Const: dùng để khai báo một hằng số - là một giá trị không thay đổi được trong suốt quá trình chạy.*

- Khi một function nằm trong 1 function khác, mà function đó có thể truy cập các biến của scope của function cha thì gọi đó là closure.

***6. Phân biệt padding với margin.***

- Hiểu 1 cách đơn giản thì thuộc tính padding là khoảng trống nằm giữa nội dung và viền, padding bao quanh 4 mặt của nội dung, còn margin là khoảng trống nằm giữa viền và phần tử tiếp theo.

***Đơn giản:***

Margin là khoảng cách so với tag bên cạnh

Padding là khoảng cách so với nội dung bên trong

***7. Grid vs Flex***

- Flex : dàn layout theo 1 chiều.

- Grid: dàn layout theo 2 chiều || grid-template-columns || grid-template-rows

***8. Khi code CSS gặp issue the cách giải quyết ntn?***

- F12 debug xem elements lỗi chỗ nào rồi fix.

***9. Cách thức hoạt động của website ngày nay khác gì ngày xưa? (Hỏi FE)***

- Hồi xưa fe xử dụng HTML, CSS, JS, jQuery và thêm kĩ thuật ajax để không bị load lại trang

- Nay sự dụng các fw Angular… sử dụng single application page. Nhiều tính năng trên 1 page

(thêm ý là server nay chỉ cần xây dựng sẵn các hàm API trả về dữ liệu và FE sẽ phụ trách công việc get dữ liệu đó. giảm chịu tải cho server làm quá nhiều việc gây load chậm,... )

***10. Khi gặp 1 request select 1 triệu record the sẽ làm cách nào ? (Hỏi leader)***

- Chia partition, cache, loadlazy.

***11. Các engine sql mà bạn biết và từng làm? (Câu này bí)***

- Engine của mysql vd như: myisam, mariadb, innodb...

***12. Các web server mà bạn biết và đã từng làm?***

- NodeJS, Apache, nginx, tomcat. (thằng nhanh nhất hiện tại là speedlight nữa và khá hot.)

***13. htaccess là gì và sử dụng ntn, ví dụ cơ bản?***

- Htaccess là tập tin để cấu hình máy chủ. Dùng để điều hướng, cấp quyền thư mục.

- Nó được viết theo quy tắc regular expersssion

RewriteEngine On

RewireBase /…

***14. Trình bày mô hình MVC một hệ thống?***

- Client -> middlware -> route->controller->model ->return view

***15. Trình bày các giải pháp tăng tốc độ load website?***

- Cache bằng htaccess: img, css,js, nén gzip,…

- Cache file….

- Cache nginx

- Load lazy

- DB: index, partition

(thêm: optimize image, bỏ thêm tiền mua vps xịn hơn).

***16. Lazyload là gì? Optimize image như thế nào?***

- Load lazy là load từ từ, có 1 sự kiện lắng nghe khi có sự kiện kích hoạt sẽ load tiếp

 Optimize image Giảm độ bit của ảnh xuống.

17. Từng dùng API nào, ví dụ ?

- Axios hoặc Fetch.

-

----------------------------------------------------------------------------------------------------------------

***II. Công ty***

1. ***Bạn đã biết gì về công ty chúng tôi chưa ?***

***2. Tại sao bạn lại nghỉ việc công ty cũ (nếu đã làm việc trước đó)?***

***3. Lý do tại sao bạn chọn công ty chúng tôi ?***

***4. Nếu được làm việc ban hy vọng môi trường làm việc sẽ như thế nào?***

***5. Bạn hy vọng điều gì để có thể làm việc lâu dài cùng công ty?***

***6. Mức lương bạn nghĩ sẽ phù hợp với bạn?***

***7. Bạn có câu hỏi gì cho công ty không?***

----------------------------------------------------------------------------------------------------------------

1. ***JAVASCRIPT***
2. ***Có bao nhiêu scope trong JS ?***

- 2 scope là: block scope, function scope.

1. ***Hoisting là gì ? Có bao nhiêu trường hợp gây ra hoisting trong JS***

* Hoisting là khi khai báo 1 biến nó sẽ chuyển phần khai báo của biến và hàm lên trên đầu function scope, chỉ đưa phần khai báo biến, không đưa phần khai báo giá trị, phần khai báo hàm lên trước phần khai báo biến
* Có 2 kiểu tạo ra hoisting là defined function và var.

***3. Prototype là gì ?***

- Là nó là một đối tượng của object, tất cả các object trong Js đều có một prototype, và các object này kế thừa các thuộc tính (properties) cũng như phương thức (methods) từ prototype của mình.

***4.Arrow function là gì ?***

- Là 1 tính năng mới trên ES6, nó This out ra ngoài 1 cấp không có this và không có argument.

***5. Class***

- Trong ES6 mới có class và ES6 trở xuống không có class, trong ES5 để định nghĩa class là function và object. Và hàm constructor trong class tương đương với các hàm define trong function trong ES5.

***6. Closure***

- Là 1 function a được define trong function, array, objects. và hàm function a đó gọi các biến this, var, let, const… và sử dụng từ function cha.

***7. var let const là gì ? Điểm giống và khác nhau giữa 3 thằng***

- var let const đều define 1 giá trị or 1 địa chỉ. khác nhau là var nó chịu ảnh hưởng fnc scope, không chịu ảnh hưởng block scope, còn let const đều chịu ảnh hưởng.

***8. Inheritance trong JS là gì? Cho ví dụ***

- new object or new function

***9.* Synchronous - *Asynchronous là gì? Có bao nhiêu methods gọi API XMLHttpRequest***

- **Synchronous có nghĩa là xử lý đồng bộ**, chương trình sẽ chạy theo từng bước và chỉ khi nào bước 1 thực hiện xong thì mới nhảy sang bước 2, khi nào chương trình này chạy xong mới nhảy qua chương trình khác.

**- Asynchronous** được ví như một chương trình hoạt động không chặt chẽ và không có quy trình nên việc quản lý rất khó khăn. Nếu một hàm A phải bắt buộc chạy trước hàm B thì với Asynchronous sẽ không thể đảm bảo nguyên tắc này luôn đúng.

***10. So sánh giống và khác nhau giữa promise và async/await***

- Promise nâng cao lên xử lý nhiều vấn đề. muốn lấy giá trị phải dùng callback, trong then catch sử dụng callback rất nhiều.

- Async/await biến hàm bất động bộ thành hàm động bộ. sử dụng try catch, catch của nó sẽ bắt tất cả lỗi của nhiều API.

***11. Cơ chế hoạt động của browser khi thực thi một trang web từ bắt đầu đến kết thúc .***

- Queue stack, call stack, realtime( cơ chế watch của browser), eventloop(“kích liên tục sự kiện”)

- Promise ra trước timeout, setinterval,...

***12. This trong Javascript? Cho ví dụ minh họa***

- 1 cái tham chiếu 1 từ khóa đại diện cho ngữ cảnh gần nhất, context. khi dùng bind sẽ làm mất ngữ cảnh đó

***13. Có bao nhiêu kiểu khai báo function? Điểm khác nhau giữa function viết thường và arrow function***

- 4 kiểu:

khai báo thẳng

khai gọi ngay (life function)(function execute)

khai báo gắn vào 1 biến

khai báo arrow function

***14. Các câu hỏi bonus xử lí tình huống:***

*\* viết thuật toán xử lý  memorized function để cache kết quả củ*

*\* làm sao để tối ưu tốc độ website*

*\* quy trình một trang web hoạt động từ lúc nhập vào một địa chỉ đến lúc render ra kết quả màn hình trà về cho user (fe -> be)*

*\* một số câu đố về iq.*

----- UPDATE 2 ------

**Object trong JavaScript là gì?**

Về mặt định nghĩa, một đối tượng (một object) là một **danh sách các item**, mỗi item là một cặp name-value, trong đó value có thể là: các kiểu dữ liệu cơ bản, function, hay cũng có thể là một object khác (kiểu dữ liệu phức hợp).

Ta gọi mỗi item là một property(thuộc tính) của object nếu value của item đó có kiểu dữ liệu là kiểu phức hợp hoặc các kiểu dữ liệu cơ bản, ngược lại nếu value của item nó là một hàm (một function) thì ta gọi nó là methodcủa object (phương thức của object).

có 2 cách tạo:

Object literals: tạo bình thường

object constructor: tạo bằng new object()

----------------------------------------------------------------------------------------------------------

Phân biệt asys vs promise, cái nào dùng tối ưu hơn, tại sao ?

Local storage, session storage, cookie là gì, khác nhau chỗ nào ?

Bootstrap có các component nào mà em biết ?

Em dùng component nào trong project ? cứ nói chủ yếu dùng grid system để chia cột

Các type của button bs ?

Tạo 1 bảng có 3 cột như thế nào ?

Em có biết scss không ? nói vài cái như nested, mixin… là gì thôi

Biết dãy fibonacy không ? ý ông đấy hỏi thuật toán? Học cách đệ quy đi

Tính múi giờ ở Mỹ với múi giờ ở vn ? lệch nhau 11 giờ, vd bên mình 12h trưa thì nó 1h sáng

Học hết các thuộc tính của flex

Cái mà để responsive trong css là gì ? media query

Em có biết về api không,từng dùng api nào, tao nói axios vs fetch nhưng chủ yếu dùng axios, có push để tạo bài viết thì phải, get để tải về, del là xoá và cái gì nữa đó, rồi hỏi thử lấy cái gì đó.

Nhảy từ ô số 1 sang số 3 bằng tab ? tab index

Project của em cái này làm như thế nào, cứ nói gì gì đó như truyền props rồi state gì đó cho ông đấy tin thôi chứ ít khi hỏi

Design partern hỏi gì méo biết

Class vs id khác nhau gì, cái nào ưu tiên hơn

Cách mà bs style như thế nào ?

Asyschyous vs prochyous em hiểu ntn ? cái ở trên giải thích rồi đó

Phân biệt var, let, const

Hoisting là gì