**CÂU HỎI ÔN TẬP FONT-END CƠ BẢN**

1. **PHẦN CÂU HỎI LÝ THUYẾT**
2. ***Có bao nhiêu cách nhúng CSS ?***

👉 Có 3 cách: Internal, External, Inline

1. ***Độ ưu tiên trong css?***

👉 Inline ➔ Internal ➔ External

1. ***Độ ưu tiên của selector trong css?***

👉Important! ➔ id ➔ class ➔ tag name ➔ specific (độ cụ thể, ví dụ div#hello.hello sẽ cụ thể hơn).

👉 Trong trường hợp hai seclector hoàn toàn giống nhau thì cái nào viết sau sẽ được ưu tiên.

1. ***Thẻ body có phải là một HTML Semantic hay không?***

👉 Không.

1. ***HTML là gì?***

👉 Ngôn ngữ đánh dấu siêu văn bản

1. ***Thẻ nào thường dùng làm tiêu đề cho bài viết?***

👉 Heading.

1. ***Cách viết transion nào sau đây là sai?***
2. Transition: width 2s
3. Transiton: width 2s, height 0.3s
4. **Transiton: width 2s/height 0.3s**
5. Transiton: 2s
6. ***Kết quả của `1 && null || false`là gì?***

👉 False.

1. ***Kết quả của `1==”1”` là gì?***

👉 True. Do “==” không xét kiểu dữ liệu.

1. ***Khi nào sử dụng được z-index***
2. Chỉ khi thẻ cha có thuộc tính position:absolute
3. Thẻ cha có position hoặc display: flex
4. Khi thẻ cha có display:flex
5. **Khi thẻ có position hoặc thẻ cha có display:flex (thẻ cha có display flex mới cập nhật).**
6. ***Thuộc tính nào không có trong transform?***
7. **Zoom**
8. Rotate
9. translateZ
10. rotateX
11. ***Mục đích của function?***

👉Tổ chức code và tái sử dụng.

1. ***Kể tên một số buil-in function (những function có sẵn không cần viết lại)?***

👉 alert, prompt, Math.random,…

1. ***Có bao nhiêu cách định nghĩa function?***

👉 Có 3 cách: function declaration, funcion expression, function arrow.

1. ***Phân biệt function và method?***

👉method là một function nhưng ở bên trong một object và được gán cho property name.

👉 function được gọi bằng tên của function đó, method được gọi bằng dấu chấm kèm theo thên của property.

1. ***Kết quả của 1 + ‘0’ - 1 là?***

👉Số 9 . Trong một phép cộng nếu có số hạng ở dạng string thì sẽ chuyển thành phép cộng chuỗi, nếu gặp phép trừ thì đổi thành phép cộng số.

1. ***Cách sử dụng for nào không hợp lệ?***
2. for (;;)
3. for (let i =0;i<10;i=i++2)
4. **for (const name of user) – user là object**
5. for (let i of user) – user là array
6. ***Kết quả của 1/0***

👉 Infinity

1. ***Css after, before sử dụng được cho?***

👉 Chỉ những thẻ có thẻ đóng.

1. ***Thuộc tính css nào không tác dụng lên thẻ inline?***

👉 Width,height. Padding vẫn được nhưng không đẩy.

1. ***Kết quả của đoạn code dưới đây:* Let a = 10; a = ’10’; console.log(typeOf a);**

***là gì.?***

👉 String

1. ***Thuộc tính nào liên quan đến display:fex ?*** 👉 **z-index**, top, margin, height
2. ***Kết quả của 1 – ‘11’ + 2\*’-2’?***

👉 Số -14

1. ***Kết quả của đoạn code sau:*** **let i = 10; for (;i<12;i) {console.log(++1)}**

👉 11,12

1. ***Cách nào*** **không *dùng để tạo 1 Object trong Js?***

A. new Object(),

B. Object.create(),

C. **factory partten (một kiểu design)** ,

D. litteral notation (dấu ngoặc nhọn {})

1. ***Kết quả của đoạn code sau:***

**var scope = ‘global’;**

**getScope = () => {**

**Console.log(scope);**

**var scope = 10;}**

**getScope();**

👉 Undefined. Do block scope và hoisting, JS tự động đưa **var scope;** lên đầu block scope và gán cho nó giá trị undefined.

1. **.text()** ***và*** .**html()** ***trong jquery có khác nhau không?***

👉 Có. **.text()** chỉ cho text, còn .**html()** vừa cho text vừa cho thẻ html

1. ***fadeIn tronng jquery là sự kết hợp của?***

👉 Display:block + opacity:1

👉 Tương tự fadeOut là sự kết hợp của display:none và opacity:0;

1. ***Khai báo biến với var nhưng không gán giá trị thì giá trị của biến đó sẽ là gì ?***

👉Undefined.

1. ***Khai báo biến với let nhưng không gán giá trị thì giá trị của biến đó sẽ là gì ?***

👉Undefined

1. ***Khai báo biến với const nhưng không gán giá trị thì giá trị của biến đó sẽ là gì ?***

👉Lỗi vì hằng số phải có giá trị.

1. ***Thẻ nào sau đây là thẻ inline?*** **👉** **Small**, ol, footer, header
2. ***Ý nghĩa của đoạn code bên dưới?***

**Math.foor(Math.random() \* 10) +1**

👉 Random 1 số từ 1 đến 10

1. ***Một function có thể return một function không?***

👉 Có

1. ***Có thể đặt biến trong css không?***

👉 Có. Ví dụ:

**:root {color:red}**

**H1 {color: var(--color);}**

1. ***Kết quả của đoạn code sau đây***: **[1,0,2].filter (val=>val != true)** ?

👉 [2,0]

1. ***Thẻ nào sau đây là thẻ block?*** 👉 **ul**, br,input, textarea.
2. ***Font-size mặc định của h2 là gì?***

👉 24px

1. ***Kết quả của đoạn code sau:*** **let x = 5; let x = 10; console.log(x); *là gì***

👉 Throw and error

1. ***Kết quả của đoạn code sau:***  **console.log(true===’true’); *là gì?***

👉False. Vì === là so sánh có xét kiểu dữ liệu nên so sánh **boolean** với **string** sẽ ra false.

1. ***Kết quả của console.log(new String(‘yes’) === new String(‘yes’)) là gì?***

👉 False. Vì khi khai báo với new thì JS sẽ tạo ra một object rỗng,mà so sánh hai object sẽ ra false.

1. ***Từ khoá nào không có trong scss?*** 👉 Mixin, function, content, **constructor**.
2. ***Giá trị nào sau đây thuộc reference type?*** 👉 **[]**, true, ‘[]’, 1
3. ***Kết quả của:*** **console.log(11&&3)**? 👉 3
4. ***Thứ tự thực thi code trong một file HTML?***

👉Code chạy từ trên xuống dưới, cái nào nhúng trước thì thực thi trước.

1. ***Nếu một function không có return thì giá trị trả về của nó là gì?*** 👉 Undefined
2. ***Có thể nhúng file css dưới thẻ body không?*** 👉 Không
3. ***Có thể nhúng file js ở trên thẻ head được không?*** 👉 Có
4. ***. Ý nghĩa của đoạn code sau:*** **Math.ceil(Math.random()\*10)+1 ?**

👉 Random một số từ 1 đến 11.

1. ***Kết quả của đoạn code:*** **const number= 10; number = 20; console.log(number); *là gì?***

👉 Throw an error. Vì hằng số không thể gán lại.

1. ***Dấu $ trong jquery là gì?*** 👉 Là một function.
2. ***Kết quả của đoạn code:* [1,2,0].map(val=>val==false) *là gì?***

👉 [false,false,true]. Vì == so sánh không xét kiểu dữ liệu nên 0 == flase => true.

1. ***Khác nhau giữa ++i và i++?***

👉 ++i : tăng trước rồi mới thực hiện câu lệnh.

👉 i++: thực hiện câu lệnh rồi mới tăng.

👉 ví dụ:

Let a = 5;

Console.log(a++); // kết quả là 5 và giá trị của a lúc này là 6

Console.log(++a); // kết quả là 7

1. ***Khác nhau giữa position: relative và position:absolute?***

👉 position:relative tách element sang một layer mới nhưng vẫn giữ nguyên khoảng trống ở vị trí cũ nếu ta di chuyển element đó đi.

👉 positon:absolute tách element sang một layer mới nhưng không để lại khoảng trống khi di chuyển element đó đi, làm ảnh hưởng đến bố cục xung quanh nó. Thuộc tính này sẽ tự động tìm đến element cha nào có thuộc tính postion, vị trí sẽ phụ thuộc vào element cha đó, nếu không tìm thấy sẽ nhảy lên đầu trang.

1. ***Nguyên lý của && và || trong JS?***

👉&&: Kiểm tra vế trái có false hay không, nếu vế trái false thì trả về vế trái, ngược lại trả về vế phải

👉|| : kiểm tra vế trái có true hay không, nếu vế trái true thì trả về vế trái, ngược lại trả về vế phải.

* Với false: “”, null, undefined, 0, false
* True: còn lại. Lưu ý, [] và {} là true vì không nằm ở trường hợp phía trên.

1. ***Javascript chạy được trên môi trường nào? Font-end hay back-end?***

👉 Javascript chạy được ở cả font-end (các trình duyệt) và back-end (như NodeJs).

1. ***Trong javascript có những kiểu dữ liệu nào?***

👉 Javascript không cần khai báo kiểu dữ liệu rõ ràng. Các loại dữ liệu như: number, string, boolean, array, object.

1. ***Kiểu dữ liệu trong javascript được chia thành những loại nào?***

👉 Primitive type: number, string, boolean, undefined, null. Giá trị của các biến khác nhau không liên quan đến nhau do được lưu trên stack. Ví dụ, khi gán a=10 và gán b =a, sau đó thay đổi biến a thành 100 thì giá trị của b vẫn là 10.

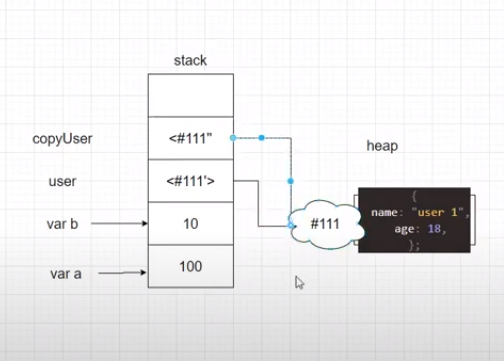
👉 Reference type: array, object, function. Được lưu trong heap một cách ngẫu nhiên, trên stack sẽ lưu con trỏ trỏ đến địa chỉ của biến trong heap, do đó khi thay đổi giá trị của biến này sẽ thay đổi luôn giá trị của biến kia. Ví dụ:

Var user = {name:’vinh’, age:23};

Var userCopy = user;

User.age=25;

Console.log(usercopy.age) // 25

******

1. ***Hoisting là gì? Tác hại của hoisting? Cách khắc phục?***

👉 Là một cơ chế mặc định của JS, khi khai báo biến bằng từ khoá var thì JS tự động đưa phần khai báo biến hoặc hàm lên đầu scope của nó (chỉ lấy phần khai báo, không động tới các phần còn lại), giá trị biến là undefined vì mặc định giá trị của biến không được gán giá trị sẽ là undefined.

👉 Tác hại: biến chưa khai báo khi mình gọi lại trả về undefined.

👉 Cách khắc phục: sử dụng let, const. Sẽ có báo lỗi khi biến chưa được khai báo.

1. ***Sự khác biệt giữa var, let, const?***

👉 let, const có block scope, không xảy ra hoisting.

👉 var có function scope và xảy ra hiện tượng hoisting.

👉 var, let nếu không gán giá trị thì biến sẽ có giá trị undefined, còn const nếu không được gán giá trị sẽ báo lỗi, vì hằng số bắt buộc phải có giá trị.

👉 Biến được khai báo bằng const thì không được gán lại giá trị.

👉 var cho phép khai báo lại biến, còn let và const khi khai báo lại biến sẽ báo lỗi.

1. ***Phép so sánh == và === trong Javascript khác nhau điểm nào?***

👉 “==” là phép so sánh không xét đến kiểu dữ liệu, JS tự convert kiểu dữ liệu phù hợp để so sánh, ví dụ 1==”1” sẽ trả về true.

👉 “===” là phép so sánh có xét kiểu dữ liệu, ví dụ 1===”1” sẽ trả về false vì so sánh một **number** và một **string**.

1. ***Callback là gì?Tại sao phải dùng callback?***

👉 Callback là một function được gọi bởi một function khác như là một đối số.

👉 Dùng để thực hiện một chức năng sau khi một chức năng khác đã hoàn tất.

👉 Cần dùng callback vì trong JS thay vì chờ phản hồi trước khi tiếp tục, JavaScript sẽ tiếp tục thực thi trong khi lắng nghe các sự kiện khác.

1. ***Callbackhell là gì?***

 👉 Callback hell trong javascript chỉ là bạn thực hiện quá nhiều callback lồng nhau làm cho code khó đọc và khó bảo trì. Trong dây truyền callback, nếu có một callback xảy ra lỗi thì các callback sau không thực hiện được.

1. ***Promise là gì?***

👉 Promise là một cơ chế trong JavaScript giúp bạn thực thi các tác vụ bất đồng bộ mà không rơi vào callback hell.

1. ***Ansync / await là gì?***

👉Cú pháp để sử dụng promise nhanh gọn hơn.

1. ***JS dùng để làm gì?***

👉 javascript được sử dụng để tính toán và thêm hành vi vào các trang web.

1. ***So sánh .filter(), .map(), for()?***

👉 .map() và .filter() chạy luôn tuồng, for() thì ngắt được.

1. ***Khác biệt giữa arrow function và function bình thường?***

👉 Cú pháp: Arrow function sử dụng “=>” thay cho từ khoá “function”.

👉 Arrow function thường ngắn gọn hơn do với những function chỉ có 1 tham số ta có thể bỏ qua cặp dấu {} và từ khoá “return”.

👉 Arrow function không được hoisted: nghĩa là phải định nghĩa arrow function trước khi sử dụng nó.

👉 Arrow function không bind this

👉 Arrow function không bind arguments

👉 Arrow function không phù hợp làm method cho object

👉 Arrow function không thể sử dụng làm hàm constructor

👉 Arrow function không có thuộc tính prototype

1. ***Chức năng chính của console.log() là gì?***

👉 Dùng để troubleshoot (gỡ lỗi).

1. ***Chức năng của prompt() là gi?***

👉 Hiện hộp thoại kèm lời nhắn và khoảng trống để người dùng nhập vào một giá trị nào đó.

1. ***Khái niệm closure? Tại sao phải dùng closure?***

👉 Closure là một hàm được tạo ra từ bên trong một hàm khác (hàm cha), nó có thể sử dụng các biến toàn cục, biến cục bộ của hàm cha và biến cục bộ của chính nó.

👉 Việc viết hàm theo kiểu closure trong một số trường hợp sẽ giúp code nhìn sáng và dễ quản lý hơn, linh hoạt hơn trong việc xử lý các chức năng.

1. ***Nút scroll to top làm như thế nào?***

👉Đặt id cho một thẻ nằm trên top của trang web, sau đó cho id đó là href của thẻ a nằm trong nút scroll to top.

👉Dùng window.scrollTop () trong js.

1. ***ES6 có gì hay hơn ES5?***

👉Let, const (chống hoisting)

👉 Có thể đặt các giá trị mặc định khi khai báo tham số trong function.

👉 Template Literals (biến trong chuỗi).

👉 Dấu backticks (``) để sử dụng chuỗi có nhiều dòng.

👉 Class : thay cho prototype khi lập trình hướng đối tượng.

👉Cho phép export và import module để sử dụng giữa các file JS khác nhau.

1. ***Khi chưa có class người ta lập trình hướng đối tượng bằng gì?***

👉Dùng function. Nếu muốn kế thừa… thì dùng prototype.

1. ***Khác nhau giữa setInterval và setTimeOut?***

👉 setTimeout(): thực thi đoạn code sau một khoảng thời gian ấn định.

👉 setInterval(): thực hiện lặp lại việc thực thi đoạn code sau một khoảng thời gian ấn định.

1. ***Khác biệt giữa alert() và confirm()?***

👉 cả hai đều hiện message bằng hộp thoại trên trình duyệt, tuy nhiên alert() chỉ có nút [ok] còn confirm() có xác nhận với hai button [ok] và [cancel].

1. ***Phân biệt margin và padding?***

👉 Margin: tạo không gian xung quanh element.

👉 Padding: tạo không gian xung quanh nội dung của element.

1. ***Công thức tính tổng chiều dài của một element?***

👉 total element width = width + left padding + right padding + left border + right border + left margin + right margin.

1. ***Công thức tính tổng chiều cao của một element?***

👉 total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin.

1. ***Ul và ol khác nhau điểm nào?***

👉 ul là unorder list: liệt kê với ký hiệu.

👉 ol là order list: liệt kê với số thứ tự.

1. ***Thế nào là thẻ block? Kể tên một số thẻ block?***

👉 Thẻ block là những thẻ mặc định bắt đầu bằng new-line (xuống dòng). Thẻ block chứa được các thẻ block và thẻ inline.

👉 Một số thẻ block: p, h1-h6, div, ul, li, form,…

1. ***Thế nào là thẻ inline? Kể tên một số thẻ inline?***

👉Thẻ inline là những thẻ mặc định **không** bắt đầu bằng new-line. Thẻ inline chứa được các thẻ inline khác.

👉Một số thẻ inline: a, b, i, input, button, select, textarea, …

1. ***HTML semantic là gì?***

👉 Là những thẻ được định nghĩa để giúp browser hay developer phân biệt được nội dung chính trong thẻ đó thuộc phần nào của web. Ví dụ: <header>, <section>,…

1. ***API là gì?***

👉 API được viết tắt bởi cụm từ tiếng Anh Application Programming Interface, tức giao diện lập trình ứng dụng. Đây là phương tiện cho hai hoặc nhiều ứng dụng trao đổi, tương tác với nhau, tạo ra tương tác giữa người dùng với ứng dụng hiệu quả và tiện lợi hơn. Với API, các lập trình viên có thể tiếp cận, truy xuất dữ liệu từ máy chủ thể hiện chúng trên ứng dụng phần mềm hoặc website của mình một cách dễ dàng hơn.

1. ***Ouline trong css là gì?***

👉 Là đường viền nằm ngoài border của element làm cho element nổi bậc hơn, thường các thẻ input, select, textarea, button,… sẽ tự động có outline.

1. ***Số lượng phần tử tối đa có thể cho vào một mảng?***

👉 Không có giới hạn, nhưng phụ thuộc vào bộ nhớ của máy tính.

1. ***Kết quả khi truy cập mảng bằng index nhỏ hơn 0 hoặc lớn hơn chiều dài mảng -1?***

👉undefined.

1. ***Nếu một mảng có 10 phần tử mà ta thêm giá trị cho phần tử có index 20, giá trị của các phần tử có index = 10 đến index = 19 là?***

👉undefined.

1. ***Điều gì xảy ra khi thêm vào object một key trùng với key đã tồn tại?***

👉 thay đổi value của key đã tồn tại bằng value của key mới.

1. ***Khi khai báo một mảng với const thì có thêm được phần tử mới không?***

👉 Được. Thêm khác với gán.

1. ***Scope là gì?***

👉 Là phạm vi sử dụng của một biến hoặc hàm.

1. ***Block scope?***

👉 Những biến được khai báo trong if, while, swicth, for bằng từ khoá **let** hoặc **const** thì có block scope. Khi sử dụng biến ngoài phạm vi này sẽ báo lỗi “is not defined”.

👉 Biến được khai báo với **var** không thể có **Block Scope**. Vì thế biến được khai báo trên trong block **{ }** có thể được truy cập từ bên ngoài block.

1. ***Function scope?***

👉 Những biến được khai báo trong function bằng từ khoá **var** sẽ có phạm vi cả function (kể cả block cha hay block con). Khi qua function khác thì biến sẽ “is not defined”.

1. ***Global scope?***

👉 Những biến được khai báo ngoài **function scope** sẽ được sử dụng ở mọi nơi. Hạn chế sử dụng cách này để tránh những bất cẩn không mong muốn. (Khi khai báo biến bằng từ khoá **var** trong if(), while(),for(), switch() thì biến đó là **global scope**).

1. ***Kết quả của:* console.log(typeof(undefined)); *là gì?*** 👉 undefined.
2. ***Kết quả của:* console.log(typeof(NaN)); *là gì?*** 👉number.
3. ***Kết quả của:* console.log(typeof(null)); *là gì?*** 👉object.
4. ***Kết quả của:* console.log(typeof([1,2,3])); *là gì?*** 👉object.
5. ***Kết quả của:* console.log(Number(false)); *là gì?*** 👉0
6. ***Kết quả của:* console.log(Number(null)); *là gì?*** 👉0
7. ***Kết quả của:* console.log(Number([])); *là gì?*** 👉0
8. ***Kết quả của:* console.log(Number(“”)); *là gì?*** 👉0
9. ***Kết quả của:* console.log(Number({})); *là gì?*** 👉NaN
10. ***Kết quả của:* console.log(Number(undefined)); *là gì?*** 👉NaN
11. ***Kết quả của:* console.log(NaN==NaN); *là gì?*** 👉 false
12. ***Kết quả của:*console.log(null==0); *là gì?*** 👉false
13. ***Kết quả của:* console.log(0/0); *là gì?*** 👉NaN
14. ***Kết quả của:* console.log(typeof Infinity); *là gì?*** 👉number
15. ***Kết quả của:* console.log(undefined == null); *là gì?*** 👉true
16. ***Kết quả của:* console.log(10>==9); *là gì?*** 👉 lỗi. Không có “>==” hay “<==”. Chỉ có “===” và “!==”.
17. ***Kết quả của:* console.log(Infinity – 1); *là gì?*** 👉Infinity.
18. ***Kết quả của:* console.log([1,2,23] == [1,2,23]); *là gì?*** 👉false. Hai object chỉ bằng nhau khi biến chứa điểm tham chiếu đến cùng một object.
19. ***Kết quả của đoạn code bên dưới:***

**Let a = {};**

**Console.log(a.length);**

👉 undefined.

1. ***Cách xác định độ dài (số lượng key) của một object?***

👉 console.log(Object.keys(object).length);

1. ***Kết quả của đoạn code sau:* console.log(‘bcd’>’dbc’); *là gì?***

👉 false. Vì trong bảng mã ASCII ‘b’ có mã nhỏ hơn ‘d’.

1. ***Kết quả của đoạn code sau:* console.log(‘Abc’<’abc’); *là gì?***

👉 true. Vì trong bảng mã ASCII ‘A’ có mã nhỏ hơn ‘a’.

1. ***Kết quả của đoạn code sau:* console.log(‘abc’ >= 1); *là gì?***

👉 false. Vì JS sẽ convert ‘abc’ thành number: ‘abc’=>NaN. Mà NaN so sánh với bất cứ giá trị nào cũng trả về false.

1. ***Kết quả của đoạn code sau là gì:* console.log(undefined >=150); ?**

👉false. Vì **undefined** được JS convert sang kiểu **number** sẽ là NaN do đó so sánh với mọi giá trị với mọi phép so sánh đều trả về false.

1. ***Kết quả của đoạn code sau là gì:* console.log(null >= 0); ?**

👉true. Vì **null** được JS convert sang kiểu **number** sẽ là 0 nên phép so sánh trên trả về true.

1. ***Kết quả của đoạn code sau là gì:* console.log(true >= 2)?**

👉false. Vì **true** được JS convert sang kiểu **number** sẽ là 1 do đó phép so sánh trên trả về false.

1. ***Ý nghĩa của* div#hello > p {} *là gì?***

👉 Áp dụng các thuộc tính css cho tất cả thẻ p là con của thẻ div có id = “hello”.

1. ***Ý nghĩa của* div + p {} *là gì?***

👉 Áp dụng các thuộc tính css cho thẻ p cùng cấp đầu tiên bên dưới thẻ div.

1. ***Ý nghĩa của* div ~ p {} *là gì?***

👉 Áp dụng các thuộc tính css cho tất cả những thẻ p cùng cấp bên dưới thẻ div.

1. ***Khác nhau giữa .find() và .findIndex()?***

👉 .findIndex(): nếu không tìm thấy sẽ trả về -1. Nếu tìm thấy sẽ trả về index đầu tiên.

👉 .find(): nếu tìm thấy sẽ trả về giá trị trong điều kiện, còn không tìm thấy sẽ trả về undefined.

1. ***Destructuring là gì?***

👉 Destructuring Assignment là một cú pháp cho phép tách dữ liệu được lưu trữ bên trong (nested) Objects hoặc Arrays (tổng quát hơn là các iterable values) và gán chúng cho các biến riêng biệt.

1. ***Spread operator là gì?***

👉Là một cú pháp cho phép biểu diễn một số lượng vô hạn đối số dưới dạng một mảng, từ đó một hàm có thể được gọi với bất kỳ số lượng đối số nào.

1. ***Rest operator là gì?***

👉 Spread Operator cho phép chuyển đổi một chuỗi thành nhiều argument (trong trường hợp gọi với hàm) hoặc thành nhiều phần tử (cho array).

1. ***DOM là gì?***

👉DOM( document object modal) là một chuẩn được định nghĩa bởi W3C dùng để cho phép các chương trình và tập lệnh truy cập động và cập nhật nội dung, cấu trúc và kiểu của tài liệu.

👉 DOM (viết tắt của Document Object Model) là một giao diện lập trình ứng dụng, cho phép Javascript hoặc các một loại script khác đọc và thao tác nội dung của document  
(trong trường hợp này là HTML).

1. ***Điều gì xảy ra khi sử dụng document.getElementbyId() và cho vào một id không tồn tại?***

👉trả về null.

1. ***Cách chuyển trang bằng JS ?***

👉 window.location.href = “<url>”.

1. ***Cách trang mới bằng Js?***

👉 window.open(url,<kiểu>).

1. ***Cách load lại trang web bằng JS?***

👉location.reload().

1. ***Regular Expression (regex) dùng để làm gì?***

👉 dùng để làm việc với string, kiểm tra chuỗi có khớp với mẫu hay không.

1. ***Tác dụng của các thẻ heading là gì?***

👉Tối ưu SEO (Seach Engine Optimiztion).

1. ***Thuộc tính alt trong thẻ img dùng để làm gì?***

👉Hiển thị thông tin khi tài nguyên không tìm thấy (hoặc chưa load được).

1. ***AJAX là gì?***

👉 Asynchornous Javascript and XML là một kỹ thuật trao đổi dữ liệu với máy chủ và cập nhật các phần của trang web mà không cần tải lại toàn bộ trang.

1. ***Hướng đối tượng là gì?***

👉 Lập trình hướng đối tượng (Object Oriented Programming – OOP) là một kỹ thuật lập trình cho phép [lập trình viên](https://topdev.vn/blog/lap-trinh-vien/) tạo ra các đối tượng trong code trừu tượng hóa các đối tượng.

👉 Mỗi đối tượng bao gồm hai thông tin: thuộc tính và phương thức

👉 Class là một khuôn mẫu, đối tượng là một thực thể được thể hiện dựa trên khuôn mẫu đó.

👉 Các tính chất của OPP: kế thừa, đa hình, trừu tượng, đóng gói.

1. ***Phân biệt parametter và argument?***

👉 Parameter là tên biến được khai báo trong phần định nghĩa hàm.

👉 Argument là giá trị biến thực được truyền vào khi gọi hàm.

1. ***BEM là gì?Tại sao lại sử dụng BEM?***

👉 BEM hay còn được biết đến là Block, Element, Modifier – một phương thức để viết HTML theo hướng module, viết CSS có tổ chức và đạt hiệu quả trong quá trình làm việc.

👉 Việc sử dụng BEM sẽ giúp cho các nhà lập trình có thể biết được các item cụ thể thuộc block nào và có class ra sao.

1. ***Phân biệt UI và UX?***

👉 UI (User Interface) – Giao diện người dùng: Đây là những gì người dùng nhìn thấy và giúp họ tương tác với hệ thống (giao diện Web, button, …).

👉 UX (User Experience) – Trải nghiệm người dùng: Đây là những gì người dùng trải nghiệm khi sử dụng sản phẩm và dịch vụ (Bao gồm cảm xúc, suy nghĩ trong quá trình sử dụng). UI chỉ là một phần của UX.

1. ***HTML5 có gì mới?***

👉 các thẻ mới như: header, footer, nav, main, section,…

👉 Bổ sung các type mới cho thẻ input như: tel, number, email, url, search,..

👉 Bổ sung một số thuộc tính mới cho thẻ input như: placeholder, step, required,…

👉 Đồ họa trên web: <canvas>, <svg> (đồ hoạ dạng vector)

👉 Lưu trữ dữ liệu thay thế cho cookie: localStorage (lưu trữ dữ liệu không có ngày hết hạng), secssionStorage (lưu trữ dữ liệu cho một phiên – dữ liệu sẽ mất khi đóng tab)

👉 Bổ sung các thẻ media cho phép nhúng audio hoặc video.

1. ***CSS3 có gì mới?***

👉 Opacity, border-radius, drop-shadown, linear-gradients, radial-gradients, multipe background images.

1. ***Khác nhau giữa …rest và arguments?***

👉 arguments lấy tất cả các tham số truyền vào dưới dạng một array-like object

👉 …rest lấy một số hoặc tất cả tham số truyền vào dưới dạng một array

1. ***Boostrap là gì?***

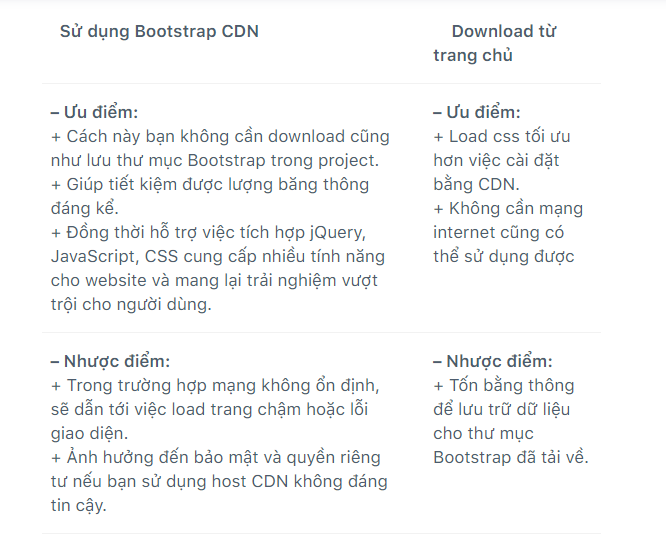
👉 Bootstrap là một trong những front-end framework phổ biến nhất trên thế giới hiện nay, được tạo ra để phát triển các trang web, có thể hoạt động tối ưu trên mọi kích thước màn hình, trên điện thoại màn hình nhỏ hoặc máy tính để bàn màn hình lớn.

1. ***Ưu / Nhược điểm của Boostrap***

👉 Ưu điểm: dễ sử dụng, responsive nhanh, tương thích với trình duyệt.

👉 Nhược điểm: Bootstrap sở hữu rất nhiều thư viện có sẵn nên dẫn tới việc website của bạn sẽ phải tải thêm rất nhiều dòng code không cần thiết khi mà bạn chỉ cần chưa đến 10% những gì Bootstrap cung cấp và sẽ khiến tốc độ load trang không được tối ưu.

1. ***Có mấy cách cài đặt Boostrap? Ưu nhược điểm của các cách?***

******

1. ***So sánh Java và Javascript?***

👉 Java là ngôn ngữ lập trình tĩnh, hướng đối tượng, hoạt động trên nhiều nền tảng. [JavaScript là ngôn ngữ lập trình động](https://quantrimang.com/javascript-la-gi-155978) (hay ngôn ngữ kịch bản - scripted language) được sử dụng để làm cho các trang web và ứng dụng trở nên sinh động.

👉 Java là một ngôn ngữ độc lập. JavaScript phụ thuộc nhiều hơn, nghĩa là nó hoạt động với [HTML](https://quantrimang.com/html) và [CSS](https://quantrimang.com/hoc-css) trên các trang web để tạo nội dung động.

👉 Cả Java và JavaScript thường được sử dụng nhiều nhất trong các ứng dụng phía máy khách.

1. ***$(document).ready() là gì?***

👉 document.ready là bộ quản lý sự kiện cơ bản của Jquery. Hiểu nôm na thì nó cũng là 1 sự kiện trong jquery, $(document).ready() được kích hoạt khi tài liệu html của trang web được load xong và cây DOM được tạo thành.

👉 HTML render từ trên xuống dưới theo 1 thread (single thread), do đó cái chưa được render ra sẽ bị lỗi khi gọi nó.

1. ***Hàm on() trong Jquery dùng để làm gì?***

👉 Hàm on có thể dùng để gắn sự kiện cho tag hiện hành hoặc những tag sẽ tạo trong sau này, tức hiện tại chưa có (các tag con sẽ được tạo sau này).

1. ***Contrustor là gì?***

 👉 Constructor là hàm được dùng để định nghĩa các thuộc tính và phương thức ban đầu cho đối tượng được tạo ra sử dụng hàm này nhờ sử dụng từ khoá new.

👉 Lưu ý: có thể thêm mới hoặc sửa chữa thuộc tính hoặc phương thức cho đối tượng được tạo ra từ constructor thông qua từ khoá new, chứ không thêm mới cho constructor đó được, nếu muốn thêm / thay đổi phải dùng .prototype.

1. ***Khác biệt giữa constructor và biến kiểu object?***

👉 Biến kiểu object chỉ tạo ra được một đối tượng, còn contrustor (hàm khởi tạo) tạo được n đối tượng.

1. ***Prototype là gì?***

👉 Prototype là cơ chế mà các object trong javascript kế thừa các tính năng từ một object khác. Tất cả các object trong javascript đều có một prototype, và các object này kế thừa các thuộc tính (properties) cũng như phương thức (methods) từ prototype của mình.

1. ***Phân biệt thư viện và framework?***

👉 Thư viện là một tập hợp các chức năng (functions), các lớp (class) được viết sẳn để có thể tái sử dụng. Mỗi function hoặc class phục vụ cho một công việc cụ thể nào đó.

👉 Framework là một tập hợp các Library đã được đóng gói để hỗ trợ phát triển ứng dụng dựa trên framework đó. Đồng thời, Framework cung cấp các nguyên tắc, cấu trúc của ứng dụng mà chúng ta phải tuân thủ theo nó.

* Framework và Library đều cung cấp các tính năng (functions) được viết sẵn để chúng ta có thể tái sử dụng.
* Framework lớn hơn và phức tạp hơn Library.
* Sử dụng Framework bạn phải thay đổi cấu trúc code của dự án (project’s structure) theo các quy tắc của framework đó để có thể sử dụng được các functions mà framework đó cung cấp.
* Chúng ta có thể sử dụng các functions của Library một cách trực tiếp mà không cần thay đổi cấu trúc code của dự án.
* Framework có thể hiểu là một khung chương trình, người dùng bổ sung code và tuân theo quy tắc để tạo ra ứng dụng. Còn Library chỉ cung cấp các chức năng tiện ích hay các class để sử dụng trong quá trình xây dựng ứng dụng.
* Framework hoạt động chủ động. Nghĩa là nó có thể đưa ra các quyết định gọi hoặc bị gọi bởi các Library hay ứng dụng nào đó.
* Library hoạt động bị động. Nghĩa là nó chỉ được gọi khi nào chúng ta cần dùng nó.

1. ***ReactJS là gì?***

👉 Là một thư viện Javascript cho việc xây dựng giao diện người dùng do Facebook thiết kế.

👉 Được sử dụng cho các SPA (single page aplication).

👉 Cho phép tạo ra các component có thể tái sử dụng.

👉 React tạo ra một DOM VIRTUAL trong bộ nhớ. Thay vì thao tác trực tiếp với DOM của trình duyệt, React tạo một DOM ảo trong bộ nhớ, nơi nó thực hiện tất cả các thao tác cần thiết, trước khi thực hiện các thay đổi trong DOM của trình duyệt. React chỉ thay đổi những gì cần thay đổi! React tìm ra những thay đổi đã được thực hiện và chỉ thay đổi những gì cần thay đổi.

1. ***Ưu điểm ReactJs?***

👉 Component độc lập có thể tái sử dụng.

👉 React làm cho front-end javascript dễ sử dụng hơn, nhanh hơn bằng cách sử dụng Virtual DOM.

👉 Dễ dàng làm việc nhóm.

👉 Được đảm bảo bởi Facbook.

1. ***JSX là gì?***

👉 JavaScript XML Là một chuẩn của Javascript XML, cho phép viết HTML trong React, tạo sự thuận tiện cho việc viết và thêm HTML trong React.

1. ***Cơ chế hoạt động của Virtual DOM?***

👉 Bất cứ khi nào có yếu tố mới được thêm vào UI, một Virtual DOM sẽ được tạo.  
 👉 Khi state của bất cứ yếu tố nào thay đổi, React sẽ cập nhật Virtual DOM trong khi vẫn giữ phiên bản Virtual DOM trước để so sánh và tìm ra đối tượng Virtual DOM thay đổi. Khi tìm được sự thay đổi, React chỉ cập nhật đối tượng đó trên DOM  
thật.

1. ***Components trong React là gì?***

👉 Component mã code độc lập và có thể tái sử dụng. Chúng phục vụ mục đích như hàm trong Javascript, nhưng hoạt động riêng lẻ và trả về HTML thông qua hàm render().

👉 Component có hai loại là Class Components và Function Components.

1. ***So sánh function component và class component?***

|  |  |
| --- | --- |
| Function | Class |
| phải trả về 1 React element | phải có 1 method render() trả về 1 React element (JSX), or null |
|  |  |
| một function Javascript / ES6 function | là một class ES6, nó sẽ là một component khi nó "kế thừa" React component. |
| nhận props làm tham số nếu cần | có thể nhận props (trong hàm khởi tạo) nếu cần. |
|  | Có phương thức khởi tạo, life-cycle, hàm render() và quản lý state (data). |

1. ***Props là gì?***

👉 Props là các đối số được truyền vào React Components

👉 Props được truyền đến components thông qua các thuộc tính HTML.

1. ***State là gì?***

👉 State là một object được tích hợp sẵn trong React

👉 Là nơi lưu trữ các giá trị thuộc tính của components

👉 Khi state thay đổi thì componetnt sẽ được re-render.

👉 Để thay đổi giá trị trong state, sử dụng *this.setState()*

1. ***Vòng đời của React component gồm mấy giai đoạn?***

👉 Gồm 3 giai đoạn: Mounting (đặt elements vào DOM), Updating(components được cập nhật lại bất cứ khi nào state hoặc props có sự thay đổi) và Unmounting (component bị xoá khỏi DOM).

1. ***Cookie là gì?***

👉 Cookies là một file tạm được tự động tạo ra trong máy tính, mỗi khi người dùng truy cập một trang web nào đó, nó sẽ lưu những thông tin liên quan đến cá nhân như tài khoản đăng nhập để sử dụng cho lần sau.

1. ***Khi nào dùng function components và khi nào dùng class components***
2. ***Tên của component được đặt thế nào?***

👉PascalCase( Viết hoa tất cả các chữ cái đầu).

# **PHẦN CÂU HỎI TƯ DUY**

## **CẤP ĐỘ DỄ**

1. ***Cách tìm số lớn thứ n trong mảng?***

👉 Sắp xếp mảng theo thứ tự giảm dần, lấy ra phần tử có index = n-1.

1. ***Trong một chuỗi, làm sao để đổi ký tự hoa thành thường và ngược lại?***

👉 Tạo một chuỗi mới bằng lowerCase() của input. So sánh từng ký tự của hai chuỗi, Nếu kí tự thứ i của input bằng kí tự thứ i của chuỗi mới thì Uppercase() kí tự thứ i của input còn khác thì lowerCase().

1. ***Cách tính tổng các chữ số của một số (kiểu number) bất kì ?***

👉 Biến input thành chuỗi bằng cách +”” và cắt thành mảng bằng .split() sau đó dùng hàm reduce((acc,val)) để tính tổng. Lưu ý biến val phải được ép kiểu sang number bằng Number().

1. ***Cho một mảng các object, mỗi object có thể có hoặc không có key tên là “budget”, tính tổng budget của mảng.***

👉 Dùng hàm reduce(). Nếu (budget in val) trả về true thì acc + val, không thì trả về acc.

1. ***Cho một chuỗi bất kì gồm nhiều từ, làm sao để với mỗi từ: ký tự đầu của từ viết hoa, các ký tự còn lại viết thường?***

👉 Dùng split(“ “) để tách chuỗi ra thành mảng các từ.

👉 Dùng map() với:

arr[index]= val.splice(0,1).toUpperCase()+val.splice(1).toLowerCase().

1. ***Cho một mảng, làm sao để lấy ra các phần tử có chứa số?***

👉 Dùng filter(), với điều kiện val.split(“”).some((a)=>typeof a == number) == true.

1. ***Cách kiểm tra mật khẩu mạnh hay yếu với điều kiện: ít nhất 6 ký tự và phải có chữ hoa, chữ thường, số*** ➔***mạnh.***

👉Nếu length của input lớn hơn hoặc bằng 6 thì xét input có khớp với regex = /[a-z][A-Z][0-9]/ hay không.

1. ***Tính tổng các số chia hết cho 2 trong mảng?***

👉Dùng hàm reduce(). Nếu val%2==0 thì trả về acc+val, ngược lại thì trả về acc.

1. ***Kiểm tra các ký tự trong một chuỗi có trùng nhau hay không (có phân biệt hoa thường)?***

👉 Dùng hàm every(). Nếu val.UpperCase() == input[0].UpperCase() thì trả về true.

1. ***Cho một mảng với các phần tử là các mảng lồng nhau, làm sao để tính tổng các phần tử trong mảng đó?***

👉 Sử dụng .join(“”).split(“”) gộp lại thành một chuỗi rồi tách ra thành mảng các ký tự.

👉Dùng reduce() để tính tổng, kiểm tra xem phần tử val có phải là số hay không bằng !isNaN(), nếu là số thì acc +val, không thì trả về acc.

## **CẤP ĐỘ TRUNG BÌNH**

1. ***Cho một chuỗi là password của một user, làm sao để bảo vệ password bằng cách thay các ký tự bằng ký tự “\*” (trừ ký tự đầu và cuối đối với những password có số ký tự lớn hơn 2).***

👉 Dùng hàm repeat(), nếu độ dài password nhỏ hơn hoặc bằng 2 thì lặp lại “\*” input.length lần, ngược lại thì input[0] + lặp lại “\*” input.lenght-2 + input[input.length-1].

1. ***Làm sao để tính số lần xuất hiện của phần tử trong mảng và lưu dữ liệu dưới dạng object?***

👉 Dùng reduce() với giá trị khởi tạo là một object rỗng, xét nếu val chưa có trong object thì acc[val] =1, còn đã có thì acc[val]++.

1. ***Cho một mảng, xuất ra một mảng gồm các phần tử sắp xếp theo thứ tự tăng dầng, không tính các phần tử kiểu string.***

👉 Dùng hàm filter() để lấy ra các phần tử là số với đk: typeof val == “number”.

👉 Dùng hàm sort() để sắp xếp.

1. ***Cho một mảng gồm các string, làm sao để biết các string trong mảng có giống nhau không? (không phân biệt hoa thường).***

👉Dùng hàm every(). Nếu val.toLowerCase() == input[0].toLowerCase() thì true.

1. ***Cho một mảng gồm các number, làm sao để tạo được một mảng gồm các mảng con là các cặp số (số đầu và số cuối)?***

👉 Cho vòng for() chạy đến input.length/2.

👉 Mỗi lần lặp tạo mảng rỗng, sau đó push(input[i]) và push(input[input.length-1-i]). Push mảng rỗng đó vào mảng result.

1. ***In ra mảng các ký tự trùng liên tiếp trong một chuỗi.***

👉 Tạo một mảng rỗng.

👉 Chạy for(), dùng includes để xét xem phần tử đó có trong mảng chưa. Nếu chưa thì xét index của nó có bằng firstIndex của nó không. Nếu không thì push vào mảng.

1. ***Làm sao để cắt một chuỗi thành các chuỗi con dựa trên số lượng ký tự cần có ở mỗi chuỗi con?***

👉 Dùng vòng lặp for() và hàm splice()

👉 Gọi k là số lượng ký tự cần cắt, bắt đầu bằng 0.

👉 push splice(i,k) vào mảng rỗng rồi sau đó i+=k;

1. ***Làm sao để loại bỏ các phần tử trùng nhau trong một mảng?***

👉 Dùng hàm filter() để lấy những val mà có index = firstIndex của nó.

1. ***Làm sao để kiểm tra một chuỗi có phải là sự lặp lại của các chuỗi con hay không?***

👉 Liệt kê ra tất cả các chuỗi con có thể có.

👉 Nếu có bất kì chuỗi con nào lặp lại n lần thì bằng chuỗi input thì trả về true, với n = input.length/subString.length.

1. ***Thay thế các ký tự đặc biệt (không thuộc a-z A-z 0-9) bằng kí tự “-“, nếu có nhiều ký tự đặc biệt kề nhau thì chỉ thay bằng một ký tự “-“?***

👉 Dùng regex = /[^a-zA-z0-9]+/g và hàm replace() để thay thế những vị trí có ký tự đặc biệt bằng “-“.

1. ***Tính tổng các number có trong một mảng gồm các string?***

👉 Dùng join(“”) để gộp tất cả các chuỗi lại thành 1 chuỗi.

👉 Tách ra thành mảng các ký tự bằng split(“”);

👉 Dùng reduce() để tính tổng, với đk: nếu typeof(val) == “number” thì trả về acc+val, ngược lại thì trả về acc.

## **CẤP ĐỘ KHÓ**

1. ***Sắp xếp mảng thứ tự giảm dần kết hợp với tần số xuất hiện. Ưu tiên tần số xuất hiện trước.Ví dụ:***

input: [3,6,3,7,4,5,3,4]

output: [3,3,3,4,4,7,6,5]

1. ***Cho mảng được sắp xếp lộn xộn gồm number và string.Sắp xếp lại mảng đó theo quy tắc: number được sắp xếp giảm dần, xen kẽ string được sắp xếp độ dài tăng dần.***
2. ***Xác định \*số lượng\* subarray, có tích các phần tử nhỏ hơn(<) 100 với subarray: chứa các phần tử kề nhau. Ví dụ:***

input: [10,5,2,6]

subarray: [10],[5],[2],[6],[10,5],[5,2],[2,6],[5,2,6]

output: 8

1. ***Tính khoảng cách gần nhất giữa các kí tự trong string so với một nguyên âm.***

**\* Nguyên âm : a, e, i, o, u**

**\* Nếu kí tự là nguyên âm thì return 0**

1. ***Cách tìm số lớn nhất có thể, từ các phần tử trong mảng?***

👉 Sắp xếp mảng theo thứ tự giảm dần, sau đó join() lại thành một chuỗi và chuyển sang kiểu number bằng Number().

1. ***Tính tổng các số(number) có trong mảng. Lưu ý: 2 hoặc nhiều nhiều số đứng cạnh nhau được đc tính là 1 số.Ví dụ: "1ab23" --> 24 (tổng của 1 và 23).***

👉Ghép mảng lại thành chuỗi, tách ra thành từng ký tự

👉 Dùng regex = /a-zA-z/g và replace() để thay thế những vị trí string bằng “-“.

👉 Có thể dùng map(), vị trí nào không phải là số thì trả về “-“.

👉 Ghép mảng lại rồi split(“-“).

👉 Dùng reduce(), nếu val là number thì acc+val không thì trả về acc thôi.

1. ***Cách tìm số nguyên dương nhỏ nhất không có trong mảng?***

👉 Sắp xếp lại mảng theo thứ tự tăng dần.

👉 Cho vòng for chạy từ 0 đến số lớn nhất trong mảng

👉 Dùng includes(), nếu i không nằm trong mảng thì trả về i và thoát vòng lặp.

1. ***Làm sao để tính tổng lớn nhất có thể có được từ các mảng con của một mảng?***

👉 Liệt kê ra tất cả các mảng con có thể có

👉 Tính tổng từng mảng con, rồi dùng Math.max().