

HTML5 CSS3

STT	Câu hỏi	Câu trả lời
1	DOM trong HTML là gì? Code HTML chúng ta viết có phải là đang viết DOM không?	<ul style="list-style-type: none"> - Document Object Model (DOM) là 1 interface mà tại đó nó coi HTML/XML là một cấu trúc cây. Mỗi node trong cấu trúc này tượng trưng cho một phần của interface. - Khi viết HTML, thì không hẳn là chúng ta đang viết DOM. Khi mở Inspect lên, tab Elements sẽ hiện DOM dưới dạng visualize. Và nó khác HTML ở chỗ nếu ta có viết sai/thiếu HTML thì DOM có thể sẽ fix lại. Ví dụ như <code><p></p></code> trong HTML sẽ thành <code><p></p></code> ở layout cuối cùng, hay như thiếu <code><html></code>, <code><head></code> thì browser sẽ thêm vào giúp mình trước khi gán lại thành 1 DOM hoàn chỉnh
2	Các thẻ Meta là gì? Kể tên một số thẻ quan trọng mà bạn biết?	<ul style="list-style-type: none"> - Các thẻ meta trong web mô tả metadata của trang đó (metadata hiểu đơn giản là data về data), và thường được dùng để qui định các keywords, author, title, v.v của trang web - Một số thẻ meta quan trọng: meta description, title tag, canonical tag, robots meta tag, responsive design meta tag, v.v
3	Trong CSS Pseudo-class là gì?	<ul style="list-style-type: none"> - Pseudo-class hiểu đơn giản là một "selector" mô tả một trạng thái đặc biệt nào đó của element hiện tại - Một số loại pseudo-class hiện có: <code>:hover</code>, <code>:active</code>, <code>:has</code>, <code>:not</code>, ...
4	Điểm khác biệt giữa relative, fixed, sticky, absolute và statically positioned element là gì?	<ul style="list-style-type: none"> - relative: không làm thay đổi cách hiển thị ban đầu, và để xác định vị trí tương đối cho thành phần so với vị trí hiện tại của nó - fixed: xác định vị trí tuyệt đối theo vùng hiển thị của trình duyệt - sticky: xác định vị trí theo vị trí cuộn của người dùng (scroll) - absolute: thay đổi cách hiển thị ban đầu, thành phần sẽ đi ra khỏi không gian ban đầu của nó (thành không gian trống). Vị trí tuyệt đối sẽ được so với thẻ cha chứa position: relative, không thì sẽ đi theo <code><body></code> - static: giá trị mặc định của các element, không bị ảnh hưởng bởi top/right/bottom/left
5	Phân biệt các giá trị inline, block, inline-block của thuộc tính display	<ul style="list-style-type: none"> - inline: không xuống dòng, không xét được height/width, chỉ xét được margin và padding <i>left-right</i> - inline-block: không xuống dòng, xét được height/width, padding/margin <i>4 hướng</i> - block: xuống dòng, xét được height/width, padding/margin <i>4 hướng</i>
6	Thuộc tính data trong HTML dùng để làm gì?	Dùng để lưu trữ dữ liệu trực tiếp trên trang HTML. data- phải gồm ít nhất một ký tự trở lên và không dùng chữ in hoa. Dữ liệu này có thể lấy thông qua JavaScript.
7	Phân biệt các giá trị của box-sizing	<ul style="list-style-type: none"> - content-box (default): width và height chỉ áp dụng cho content, không bao gồm padding, border và margin - border-box: width và height bao gồm content, padding, border, và không bao gồm margin - padding-box: width và height bao gồm content, padding, và không bao gồm border, margin
8	Sự khác nhau giữa Responsive Design (RD) và Adaptive Design (AD)	<ul style="list-style-type: none"> - RD: Thay đổi cách hiển thị trên các thiết bị khác nhau - AD: Thay đổi thiết kế cho các thiết bị và màn hình khác nhau - AD có thể xây dựng trên một trang đã có từ trước, RD có thể phải suy nghĩ lại từ đầu - AD có thể dùng một url hoàn toàn khác (m.fb), RD dựa vào HTML/CSS/JS
9	Tại sao HTML lại sử dụng các tag khác nhau thay vì chỉ sử dụng <div>?	<ul style="list-style-type: none"> - Do <div> không có "semantic meaning" (không có nghĩa về mặt ngữ nghĩa), nên mình dùng các thẻ khác nhau để người viết dễ maintain, người đọc dễ access (maintainability, accessibility, SEO, ...) - Vd xoay quanh semantic: sẽ có nghĩa về mặt semantic hơn . Do định nghĩa là chữ này sẽ mạnh/quan trọng hơn mấy chữ khác (dành cho người khiếm thị ...), còn chỉ là một style (bold text) và không có định nghĩa tình quan trọng của text đó
10	Nêu vai trò của HTML/CSS/JS trong một trang web một cách ngắn gọn?	<ul style="list-style-type: none"> - HTML: Nền móng, là cái khung của nhà - CSS: Bề ngoài của căn nhà - JS: Các chuyển động trong nhà (như bật tắt đèn, bật tivi, ...) - (BE: đường ống, đường dây điện)
11	Mức độ ưu tiên của CSS Style?	<ul style="list-style-type: none"> - Inline styling (ghi chung thẻ tag) > Internal styling (ghi trong <style> trên <head>) > External styling (ghi ở file css bên ngoài) - Mức độ ưu tiên còn tùy thuộc vào vị trí đặt External và Internal styling
12	Bạn có biết gì về JSSS (Javascript styled sheet)?	<ul style="list-style-type: none"> - JSSS là một cách gán style cho HTML, với JavaScript, thay vì dùng CSS - VD: <pre><style type="text/javascript"> tags.h1.color = "blue" </style></pre>
13	Nếu muốn select một phần tử cha từ một phần tử con thì viết scss như thế nào?	<pre>.right-to-left { transform: translateX(100%); @at-root .active { transform: translateX(0) } }</pre>

STT	Câu hỏi	Câu trả lời
1	Sự khác biệt giữa cookies, session storage, and local storage?	<ul style="list-style-type: none"> - Quyền truy cập: Local storage và cookies ở trên các windows; session storage chỉ tồn tại ở tab hiện tại - Hạn kết thúc (bị huỷ/xoà): + Cookie tự set (manual) + Local không bao giờ bị huỷ (trừ khi mình dùng JS hoặc xoà tay) + Session bị huỷ khi close tab - Lưu trữ: Local và Session storage lưu trên Browser (client-side); Cookies lưu trên Browser và Server - Dựa vào cookies, trang web hiện tại có thể track người dùng và thêm những features user-specific ví dụ như recommendation
2	CORS là gì?	Cross-origin resource sharing (CORS) là một cơ chế cho phép các nguồn tài nguyên trong một trang web được request bởi một domain khác (ví dụ như trang lotteria có thể gọi sang api của trang kfc nếu không bị block CORS)
3	Promise là gì? Các state khác nhau trong Promise?	<ul style="list-style-type: none"> - Promise là một đối tượng thể hiện cho sự hoàn thành hoặc thất bại của một tiến trình bất đồng bộ, giúp thực thi các tác vụ bất đồng bộ mà không rơi vào callback hell - State: + pending + fulfilled + rejected
4	Hoisting trong JavaScript là gì?	Cơ chế hoisting là cơ chế đưa các phần khai báo biến và hàm lên các dòng đầu tiên của Scope chứa nó. Tuy nhiên, chỉ di chuyển các phần khai báo, phần gán giá trị vẫn giữ nguyên vị trí cũ
5	Các giá trị falsy trong JavaScript là gì? Có cần lưu ý gì khi làm việc với giá trị falsy không?	<ul style="list-style-type: none"> - false, 0, -0, "", null, undefined, NaN - Array là falsy ([] == false). Nên nếu cần thì phải check độ dài của array để biết rỗng hay không. - Object rỗng là truthy. Nên check object thì cần tự check rỗng
6	Kể tên các loại data type trong JavaScript	- String - Number - BigInt - Boolean - undefined - null - Symbol - Object https://www.programiz.com/javascript/data-types
7	Phân biệt call, apply, và bind (xác định con trỏ this)	<ul style="list-style-type: none"> - call: Gọi hàm và cho phép bạn truyền vào một object và các arguments phân cách nhau bởi dấu phẩy function.call(thisArg, arg1, arg2, ...) - apply: Gọi hàm và cho phép bạn truyền vào một object và các arguments thông qua mảng
8	Browser xử lý JS như nào ?	Thông qua JS engine, JS engine nổi tiếng nhất là V8 của chrome
9	Khi nào thì dùng IIFE?	
10	Closure trong JavaScript là gì? Nếu ví dụ.	<p>Closure là một hàm được viết lồng vào bên trong một hàm khác (hàm cha) mà nó có thể sử dụng và thay đổi các tham số của hàm cha lần của chính nó.</p> <pre> const increase = () => { let x = 0; const increaseInner = () => ++x; return increaseInner; }; const useClosure = increase(); console.log(increase()); //output: 1 console.log(useClosure()); //output: 1 console.log(useClosure()); //output: 2 </pre>
11	undefined khác gì null?	<ul style="list-style-type: none"> - undefined là 1 type - null là 1 giá trị - Khi một biến được khai báo, và không được gán giá trị thì biến đó có type là undefined - Còn null là một giá trị có thể gán vào cho một biến nào đó đã được khởi tạo
12	this là gì?	Khi gọi this với như là phương thức của một object, thì thông thường this lúc này sẽ trỏ tới "OBJECT ĐỨNG TRƯỚC DẤU "
13	Arrow function khác gì so với function thường?	<ul style="list-style-type: none"> - Arrow function không bind this, không làm thay đổi ngữ cảnh của con trỏ this (this ứng với ngữ cảnh gần nhất) - Arrow function có thể bỏ qua từ khoá 'return' - Arrow function không bind arguments - Arrow function không gặp vấn đề về hoisting
14	So sánh var, let và const?	<ul style="list-style-type: none"> - var bị ảnh hưởng bởi cơ chế hoisting và hoạt động ở mọi nơi trong global scope - let không bị ảnh hưởng bởi hoisting, có thể tái gán giá trị và hoạt động trong block scope - const không bị ảnh hưởng bởi hoisting, không thể tái gán giá trị và hoạt động trong block scope
15	Khi nào thì cần "use strict"?	
16	Viết duplicate([1,2,3,4,5]) => [1,2,3,4,5,1,2,3,4,5]	- arr.concat(arr)

STT	Câu hỏi	Câu trả lời
17	Kể tên các tính chất của đối tượng (OOP) trong JavaScript?	- VI: Tính đóng gói, Tính đa hình, Tính thừa kế, Tính trừu tượng - EN: Encapsulation, Polymorphism, Inheritance, Abstraction
18	Higher order function là gì? Kể tên các higher order function trong ES6?	- Higher order function là một function thực hiện ít nhất 2 việc dưới: - Nhận function(s) làm tham số - Trả về 1 function (closure) - Các higher order function thường thấy trong ES6: reduce, map, filter, forEach
19	Spread operator dùng khi nào ?	Dùng khi muốn copy Array, hoặc object. Dùng khi muốn thêm 1 phần tử mới vào mảng, hay object (merge array/object)
20	Nếu có 3 promise đang chạy, nếu muốn chờ xong cả 3 mới làm thì dùng cách nào ?	Dùng promise.all
22	Giải thích shallow copy và deep copy, nêu ví dụ?	- Trường hợp deep copy: dùng spread operator, cắt đứt mối liên hệ với giá trị cũ - Trường hợp shallow copy: copy giá trị, nhưng c
23	Phân biệt toán tử "==" và "===" trong Javascript?	- "==": so sánh giá trị - "===": so sánh giá trị và kiểu dữ liệu - Đối với array, object: so sánh địa chỉ vùng nhớ (reference)
24	Callback function là gì?	Callback function là một function A được truyền vào một function B dưới dạng một tham số. Tại một thời điểm nào đó tùy thuộc vào cách function B được xây dựng mà function A sẽ được function B gọi để thực thi.
25	Callback hell là gì? Và cách tránh callback hell?	- Callback hell là cách code không tối ưu, khi các hành động asynchrone được thực thi 1 cách liên tiếp, dẫn tới việc các callbacks bị lồng ghép vào lẫn nhau, khó theo dõi. - Một trong các cách tránh: Dùng async await promise
26	Có mấy cách khai báo function trong js. Khác nhau là gì?	- Declaration function (DF): function logSomething() {} - Expression function (EF): const logSomething = function () {} - DF bị ảnh hưởng bởi cơ chế hoisting, còn EF thì không. Vì vậy, nếu gọi hàm viết dưới dạng EF chạy trước khi khai báo sẽ bị lỗi.
27	Clone 1 object thì dùng những cách gì	Spread operator, hoặc stringify()
28	"typeof Array" trả về kết quả là gì? Làm sao biết array là một Array?	- typeof(array) -> object - Dùng Array.isArray(array) để kiểm tra
29	Bạn hiểu gì về event loop	https://topdev.vn/blog/hiieu-ve-javascript-bat-dong-bo-vong-lap-su-kien/
30	Sử dụng gì để call api, ngoài axios?	Fetch và AJAX
31	Giải thích kiểu tham trị và kiểu tham chiếu trong JavaScript. Nêu ví dụ cho passByValue và passByReference trong JS nếu có	- Kiểu tham trị: Dành cho các biến, tham số có các kiểu dữ liệu cơ bản. Ví dụ như String, Number - Kiểu tham chiếu: Chứa địa chỉ vùng nhớ trỏ tới đối tượng, và địa chỉ đó dùng để truy cập vào vùng nhớ mỗi khi cần lưu hay lấy dữ liệu của biến đó. Ví dụ như Object - VD truyền kiểu passByValue và byReference: https://viblo.asia/p/javascript-truyen-tham-tri-va-truyen-tham-chieu-gAm5yoAqZdb
32	Khác biệt giữa Asynchronous và Non-blocking	
33	Để lấy 1 thẻ bằng ID có mấy cách, nếu có thì trả về gì, không tìm ra thì trả về gì?	- document.getElementById("myId") ; document.querySelector("#myId") - Nếu có thì trả về thẻ đó, nếu không thì trả về null
34	ES6 offer các tính năng mới gì cho mình sử dụng, kể tên?	- Các tính năng mới: Arrow function, classes, promise, let & const, template string, spread operator, destructuring, ...
35	Làm sao để thực thi 1 cái hàm nhưng không cần gọi hàm?	Dùng Immediate-Invoked Function Expression (IIFE)
36	Kể tên các sự kiện tương ứng với 'onmousedown', 'onmouseup', 'onmousemove' trên touch screen?	touchstart, 'touchend', 'touchmove'
37	GET và POST đều có thể dùng để gửi dữ liệu về cho server xử lý, hãy so sánh điểm khác biệt?	GET: - Tốc độ nhanh - Không bảo mật do data hiện lên thanh url. - Giới hạn kí tự, giới hạn độ dài trên thanh url. POST: - Tốc độ không nhanh bằng GET - Bảo mật hơn GET - Không bị giới hạn độ dài

ReactJS

STT	Câu hỏi	Câu trả lời
1	Bạn hiểu gì về state, props. Giống và khác	<ul style="list-style-type: none"> - Props được truyền vào Component cha và có thể truy cập được ở các Component con - State được dùng để biểu diễn trạng thái của Component. State là một thành phần của Component và chỉ có thể bị thay đổi bởi chính Component chứa nó. - Props và State đều được sử dụng để lưu trữ dữ liệu trong Component - Khác nhau: + Công dụng + Phạm vi sử dụng + Trong component, props cố định ở các Component con, state có thể bị thay đổi ở các Component con
2	React là oneway binding hay two way binding?	<p>Oneway vì khi đổi model thì view không đổi (html không đổi), phải thay đổi state</p> <p>Trong React, data được thiết kế là chỉ đi một chiều từ cha xuống con (trong Angular đi 2 chiều được)</p>
3	Phân biệt functional (stateless) component và class (stateful) component?	<ul style="list-style-type: none"> - Như cái tên "stateless component", functional component không thể quản lý React state (data) và life-cycle. Nhưng sau này đã có React Hooks nên functional component không hẳn là stateless nữa (nó đã có state, đã có life-cycle). - Stateless: + là một function + trả về 1 React element + nhận props làm tham số (nếu cần) - Stateful: + là một class + nhận props trong hàm khởi tạo (nếu cần) + phải có 1 phương thức render() trả về JSX/null
4	React là thư viện hay framework? Thư viện và framework khác nhau ở điểm nào?	<ul style="list-style-type: none"> - React là thư viện (Angular là framework) - https://viblo.asia/p/su-khac-nhau-giua-framework-va-library-maGK7D6DZjz
5	Kể tên các giai đoạn khác nhau trong vòng đời của 1 component	<ul style="list-style-type: none"> - Vòng đời Creation: constructor --> getDerivedStateFromProps --> render --> componentDidMount --> componentWillUnmount - Vòng đời Update: getDerivedStateFromProps --> shouldUpdate --> render --> componentDidUpdate
6	Khi thay đổi state, component render lại, có cách nào để khi thay đổi state mà component ko cần render lại ko?	Dùng shouldComponentUpdate => return false
7	Sử dụng react hook thì làm sao để quản lý render?	<ul style="list-style-type: none"> - Rerender: dùng useMemo (class dùng PureComponent), useCallback - Lifecycle: dùng useEffect (bao gồm componentDidMount, componentDidUpdate, componentWillUnmount)
8	Sử dụng lifecycle trong React Hooks như thế nào?	<ul style="list-style-type: none"> - componentDidMount: useEffect(() => { //do something }, []) - componentDidUpdate: useEffect(() => { //do something if flag changed }, [flag]) - componentWillUnmount: useEffect(() => { return () => { //do something before component is removed from } }, []);
9	Em đã từng sử dụng các thư viện quản lý state nào?	Redux, mobx
10	Cơ chế so sánh của redux?	Shallow equality checking (hay so sánh tham chiếu) là phép so sánh chỉ cần kiểm tra xem 2 biến được so sánh có cùng tham chiếu đến 1 object hay không, ngược lại với nó là thuật ngữ deep equality checking (hay so sánh giá trị) phải kiểm tra mỗi giá trị của từng thuộc tính ở cả 2 objects
11	Công dụng của middleware là gì?	Là cầu nối giữa DB và App, xử lý các request/response từ actions và handle nếu có lỗi xảy ra
12	componentWillUnmount chạy trước hay sau khi component bị hủy?	Chạy trước khi Component bị hủy
13	PureComponent và memo dùng để làm gì?	Để ngăn việc re-render không cần thiết
14	JSX là gì?	JSX là 1 cú pháp mở rộng cho JavaScript, là kết hợp của JavaScript và XML. Nó chuyển đổi cú pháp dạng gần như XML về thành JavaScript, giúp chúng ta có thể code ReactJS bằng cú pháp của XML thay vì phải dùng JavaScript
15	Higher Order Component là gì? Thường trong dự án sử dụng khi nào?	<ul style="list-style-type: none"> - HOC là 1 function nhận vào 1 component, và trả về 1 component - Sử dụng khi cần wrap các Screen trong một Layout chung như có chung Header, Foot, Side drawer
16	Một số cách để tối ưu ứng dụng react	<ul style="list-style-type: none"> - Sử dụng useCallback, memo, useMemo để ngăn việc re-render không đáng có - Tối ưu dung lượng ảnh, hạn chế sử dụng svg - Sử dụng lazyload
17	Bạn đã sử dụng context API bao giờ chưa? context API truyền dữ liệu như nào? có thể truyền giữa các component anh em được không?	
18	Khi sử dụng các hook của react như useCallback, useEffect, ta hay truyền 2 tham số, callback function và 1 mảng dependencies, sự khác biệt của truyền mảng rỗng và không truyền là gì?	Truyền mảng rỗng tương ứng với việc hook chỉ chạy một lần và không bao giờ chạy lại trong suốt vòng đời của Component.

ReactJS

STT	Câu hỏi	Câu trả lời
19	Ưu và nhược điểm của single page application so với multiple page application?	SPA không cần phải load lại trang nhiều lần, tăng trải nghiệm người dùng. Nhược điểm: Giảm SEO
20	Luân chuyển dữ liệu trong redux?	dispatch (action) --> reducer --> new state (store) --> re-render (view)
21	Hook là gì, và có các loại hook nào?	Hook cho phép sử dụng state và các tính năng khác của React (ví dụ như lifecycle) mà không phải dùng đến Class. Các hook cơ bản: useState, useEffect, useReducer, useSelector, useDispatch, useCallback, useMemo, useContext, useRef, v.v
22	Khi 1 component setState mà nó đã bị unmount trước thì sẽ như nào, cách khắc phục?	- Do setState bất đồng bộ, nên lúc component unmount thì có thể setState chưa chạy xong, dẫn tới memory leak (nếu có nhiều setState chưa chạy xong, có thể làm chậm web, 1 cái thì có thể không ảnh hưởng lớn tới performance). Trường hợp tệ nhất là miss lúc bỏ các event listener (ví dụ như .resize sẽ lắng nghe hoài), hoặc miss lúc bỏ các interval (setInterval) - Dùng componentWillUnmount và một biến theo dõi component đã được mounted hay chưa, để đề phòng setState trên 1 unmounted component didMount: this.mounted = true willUnmount: this.mounted = false
23	Redux-thunk là gì? Thunk hoạt động như thế nào? Thunk nhận vào cái gì và trả về cái gì?	
24	Khác nhau giữa redux-thunk và redux-saga?	
25	React.memo() hay useMemo() trong Hooks dùng để làm gì?	- Tránh việc re-render nhiều lần ảnh hưởng đến performance, tương tự như PureComponent hay shouldComponentUpdate
26	Nếu muốn thay đổi element khi kích cỡ màn hình browser thay đổi thì làm như thế nào trong React?	- Tạo 1 custom hook useWindowResize theo dõi width, height thông qua sự kiện 'resize'. Dựa vào đó để nhận thấy sự thay đổi
27	Tại sao không nên thay đổi state trực tiếp (VD: this.state.a = b)? Thay đổi trực tiếp như vậy dẫn đến điều gì?	
28	Các cách thay đổi state và tại sao lại có nhiều cách?	- setState(currState: 'newerState') - setState(prevState => ((currState: prevState+1))) //prevState luôn là state mới nhất để mình thực hiện các thao tác trên nó. Với cách dùng 1 thì có thể currState hiện tại chưa kịp update lên cái mới nhất
29	Sự khác biệt giữa Element và Component	
30	Lợi ích khi dùng refs là gì, và cách tạo refs?	
31	Sự khác biệt giữa Shadow DOM và Virtual DOM	- React dùng Virtual DOM - Shadow DOM là một công nghệ của browser, được dùng chủ yếu để scope các biến, và CSS của các thành phần trên web. Stylesheet hay JS bên ngoài đều không tác dụng vào Shadow DOM được - Virtual DOM là một Object chứa các thông tin cần thiết để tạo ra một DOM. Nó có khả năng tính toán, cập nhật các node mà không cần sử dụng DOM APIs. Sau khi cập nhật trên DOM ảo, các thay đổi sẽ được thực hiện với Original DOM => không cần render lại toàn bộ DOM tree
32	Lifting State Up trong React nghĩa là gì?	Trong React, để chia sẻ State, chúng ta sẽ di dời nó lên "tổ tiên" chung gần nhất của các Components cần state đó (bỏ local state). Việc này gọi là Lifting State up. Như tên gọi thì, lifting state up đơn giản là đem state của các Component con lên Component cha, và các Component con có thể dùng chung state đó
33	Tại sao React dùng className thay vì class như HTML thông thường?	- Vì 'class' là một keyword được dùng bên JavaScript, mà JSX là một phiên bản mở rộng của JS => phải đổi để không nhầm lẫn và dùng sai trong 1 số trường hợp
34	Fragments là gì?	Là 1 pattern chung trong React, dùng cho Component để trả về nhiều elements mà không cần tạo thêm các nodes vào DOM
35	Chuyện gì sẽ xảy ra khi dùng setState trong constructor	- Khi dùng setState() thì thường nó sẽ re-render nếu có sự thay đổi, nhưng tại constructor(), component chưa được render nên việc mình setState không có ý nghĩa => đang gọi setState trên một unmounted component - Lúc này mình sẽ gặp lỗi "Cannot read property 'firstState' of null" => gán thẳng this.state = {} trong constructor
36	Cách tạo đường dẫn tới Notfound page	- Cách 1: Để Route của NotfoundPage dưới cùng <Route component={NotFoundPage} /> - Cách 2: Điều hướng tới trang khác (cùng để ở dưới cùng) <Route path="/404" component={NotFoundPage} /> <Redirect to="/404" />