

HTML INTERVIEW Q&A

Q-1: What are the differences between html4 and html5?

A-1: The main difference between HTML 4 and HTML 5 is that HTML 4 is the 4th and older version of HTML. HTML4 Lengthy Doctype declaration, On the Other hand, HTML5 has a Simple Doctype declaration. HTML 4 does not support multimedia while HTML 5 supports multimedia.

Q-2: What are semantic tags in html? Give me some examples.

A-2: A semantic element clearly describes their meaning in a human- and machine-readable way

the developer.

Example:

- <article>
- <aside>
- <details>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>

Q-3: What is the purpose of Article, div, section, nav, aside?

A-3:

purpose of <article>: The <article> tag is one of the new semantic elements introduced with HTML5.

According to the HTML5 specification : The article element represents a section of content that forms an independent part of a document or site; for example, a magazine or newspaper article, or a blog entry.

purpose of <aside>: The <aside> HTML element represents a portion of a document whose content is only indirectly related to the document's main content. Asides are frequently presented as sidebars or call-out boxes.

purpose of <div>: The <div> tag defines a division or a section in an HTML document. The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

purpose of <section>: The HTML <section> tag is used to define sections in a document. When you put your content on a web page, it may contains many chapters, headers, footers, or other sections on a web page that is why HTML <section> tag is used. HTML <section> is a new tag introduced in HTML5.

purpose of <nav>: The <nav> HTML element represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes

Q-4: Why will you use Meta tag?

A-4: The <meta> tag defines metadata about an HTML document. Metadata is data about data. <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

Q-5: Inline block vs inline-block?

A-5:

inline: The element doesn't start on a new line, and only occupy just the width it requires. You can't set the width or height.

inline-block: It's formatted just like the inline element, it doesn't start on a new line, but you can set width and height values.

block: The element will start on a new line and occupy the full width available. And you can set width and height values.

Q-6: Difference between strong, b, em, i?

A-6: b or i means you want the text to be rendered as bold or italics. strong or em means you want the text to be rendered in a way that the user understands as "important". The default is to render strong as bold and em as italics, but some other cultures might use a different mapping.

Q-7: What are properties and attributes in HTML?

A-7: Attributes are additional information which we can put in the HTML to initialize certain DOM properties. Properties are formed when the browser parses the HTML and generates the DOM.

Q-8: What is a Viewport?

A-8: The browser's viewport is the area of the window in which web content can be seen. This is often not the same size as the rendered page, in which case the browser provides scrollbars for the user to scroll around and access all the content.

Q-9: Have you used Audio and Video tags? How does they work?

A-9: Yes I did.

<audio> tag working principle: The <audio> HTML element is used to embed sound content in documents. It may contain one or more audio sources, represented using the src attribute or the <source> element: the browser will choose the most suitable one. It can also be the destination for streamed media, using a MediaStream

<video> tag working principle: The HTML5 video element tells the browser to load a video file from another source by specifying the video file's location, similar to the way a browser loads an image file (the image itself is not stored in the HTML file — the browser pulls it from somewhere else).

Q-10: What is hyperlink in html? what tag and attribute will you use for hyperlink?

A-10: A hyperlink, also called a link or web link, contains an address for a destination and acts as a reference to data. A user can easily follow, jump to, and be directed to the destination by either clicking, tapping on, or hovering over the link. I used `<a>` tag for hyperlink. `<a>` has some attributes also. Usually I used href and target attribute.

Q-11: What is the difference between HTML elements and tags?

A-11: HTML tag is just opening or closing entity. For example:

`<p>` and `</p>` are called HTML tags HTML element encompasses opening tag, closing tag, content (optional for content-less tags) Ex: `<p>This is the content</p>` : This complete thing is called a HTML element

Q-12: What is charset in html? why will you use it?

A-12: The charset attribute specifies the character encoding for the HTML document. The HTML5 specification encourages web developers to use the UTF-8 character set, which covers almost all of the characters and symbols in the world!

CSS INTERVIEW Q&A

Question-13: What Flex layout? Difference Between Flex and grid layout?

Answer-13: Flex Layout: Flex Layout is a flexible box layout , that provides a more efficient way to layout, align, and distribute space among items in the container. Difference between flex and grid: The main difference between flex and grid layout is- flex is one dimensional. In a flex layout, items can be arranged either in rows or columns. On the other hand, grid layout can arrange items both in rows and columns at a time.

Question-14: Explain CSS position property? What are some differences between absolute position and relative position?

Answer-14: The CSS position property defines the position of an element in a document. CSS position property has five values. They are - static, sticky, relative, absolute, fixed. This property works with left, right, top, bottom to specify final position of an element.

Difference between absolute and relative position: absolute position means the element is positioned absolutely to its first positioned parent. On the other hand, relative position means the element is positioned related to the browser window.

Question-15: What is a box model? And what are the different elements of a box model?

Answer-15: box model: box model usually referred to CSS box model is a box containing all the elements like content, margin, padding, and border.

Elements of box model:

Content: the main area where text and images can take place

Padding: space inside content

Margin: space outside the main content

Border: a border around the padding and main content.

Question-16: What is a Hover effect? What is the purpose of the active class?

Answer-16: hover effect allows us to give some styles to an element only while hovering the mouse over it. Purpose of the active class: active is a CSS pseudo-class. This class selects an element based on its state. And when the state matches, you can style that element as your desired style.

Question-17: What are the different types of Selectors in CSS

Answer-17: id selector, class selector, element selector, universal selector, group selector

Question-18: What is CSS Specificity?

Answer-18: Specificity is the algorithm used by browsers to determine the CSS declaration that is the most relevant to an element, which in turn, determines the property value to apply to the element. Specificity priority: inline style > id selector > (class selector, pseudo-class selector, attribute selectors) > element selector

Question-19: What is a CSS Preprocessor? What are some benefits of Sass?

CSS preprocessor: By using CSS preprocessor we can generate unique styles that don't exist in pure CSS such as mixin, nesting selector, and inheritance selector.

Benefits of SASS(Syntactically Awesome Style Sheet):

- a. Using SASS we can generate our unique styles
- b. SASS allows us to mixin
- c. SASS uses nested syntax
- d. We can use variable
- e. It is pure CSS friendly with the regard of syntax

Question-20: What is a Pseudo element? Give an example of pseudo-element

Answer-20: CSS pseudo-element is used to give style to a specific part of an element.

Example: ::after, ::before

Question-21: How will you use media queries to make a website responsive?

Answer-21: At first I have to ensure the breakpoint of small, medium and large device in regards of width. Usually I use breakpoint as follows:

Small: $\leq 576\text{px}$

Medium: > 576 and $\leq 872\text{px}$

Large: $> 872\text{px}$

Then i use for small device:

```
@media only screen and (max-width:576px){
```

```
}
```

For medium device :

```
@media only screen and (min-width:576px and max-width:872px){
```

```
}
```

The default will have the property of large device

Question-22: How Will You Make Font-Size Responsive?

Answer-22: I use relative units to make font size responsive. I use %, rem, vw, vh insted of using px.

Question-23: Difference between transition and transform

Answer-23: The transform property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skew, etc. On the other hand CSS transitions allows you to change property values smoothly, over a given duration

Question-24: How will you horizontally and vertically center a div inside a div

Answer-24: I will use following style to the parent element

Display: flex,

justify-content: center,

align-items: center

JAVASCRIPT INTERVIEW Q&A

Q-25: How does javascript work?

A-25: JavaScript is a synchronous, blocking, single-threaded language. That just means that only one operation can be in progress at a time.

Q-26: How does javascript code is executed in the browser?

A-26: In Chrome browser, Javascript code is executed “v8 engine. “v8 engine” is open-source, high-performance engine which is written in C++. Javascript is an interpreted language. v8 engine interprets javascript code line by line and converts it to optimised machine code. Just In Time (JIT) compiler makes this process quicker and more efficient. Then machine code is converted into byte code which is runnable by the browser .Different browsers have their own javascript engines to run javascript in the browser. Firefox uses SpiderMonkey.

Q-27: What is the difference between “==” and “===” in js?

A-27: The “==” and “===” both are comparison operators. The double equal compare only values of operands. It ignores the data type of operands. Actually “==” does the type conversion before comparison. On the other hand, the “===” operator compares both the values and data types of operands.

Q-28: What is a callback function in js?

A-28: We can use a function inside another function. A callback function is a function that is passed as an argument to another function so that it can be executed in that function. The callback function is executed after another function(parent function) has finished execution.

Q-29: When will you return something from a function?

A-29: I will return a value from a function when I want to return that value to the caller. Also I use return when I need more than one value from the function. I also use return to terminate the execution of a function. In this case I wrap the desired values into an Array and return the Array.

Q-30: How many arguments can call(), apply() and bind() methods take?

A-30:

- a. The call() method takes arguments separately.
- b. The apply() method takes arguments as an array.

Q-31: What is a Closure in JavaScript? How does it work?

A-31: A closure is the combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment). In other words, a closure gives you access to an outer function's scope from an inner function. In JavaScript, closures are created every time a function is created, at function creation time.

Q-32: What does “this” keyword indicate in js?

A-32:

- a. In a method: “this” refers to an object.
- b. Alone: When “this” is used alone, it refers to a global variable.
- c. In strict mode: “this” refers to a global object.
- d. In event handlers: “this” refers to an HTML element.

Q-33: How does event delegation work in javascript?

A-33: Event Delegation is basically a pattern to handle events efficiently. Instead of adding an event listener to each and every similar element, we can add an event listener to a parent element and call an event on a particular target using the .target property of the event object.

Q-34: Explain hoisting in javascript.

A-34: In js, hoisting is a behavior in which a function or a variable can be used before declaration. In terms of variable hoisting, a variable with let or const can not be hoisted. A variable with var can be hoisted.

Q-35: What is a recursive function in javascript?

A-35: A function that calls itself is called a recursive function. A recursive function must have a condition to stop calling itself. Otherwise it will call itself infinite times and it will never stop. When the given condition is met, the function stops calling itself.

Q-36: What is the difference between “undefined” and “null” in js?

A-36: In javascript, undefined is a data type whereas null is an object. “Undefined” means a variable that has been declared but it has not assigned a value yet. On the other hand “null” is an assignment value. It can be assigned to a variable as a representation of no value.

Q-37: What are the different data types in js?

1.Primitive data types: In javascript, a primitive (predefined) data is not an object and it has no methods or properties. All primitives are immutable. A primitive can be replaced but it can't be directly altered. There are 7 primitive data types.

- a. String
- b. Number
- c. BigInt
- d. Boolean
- e. Undefined
- f. Symbol
- g. null

2. Non-primitive data types: Non-primitive data types are called as reference types because they refer to an object. Non-primitives are mutable and can be altered.

There are 3 non-primitive data types:

- a. Object
- b. Array
- c. regex

Q-38: What is DOM?

A-38: The Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document. With DOM javascript can access and change all elements of an HTML document. Javascript can also create dynamic HTML elements with DOM.

Q-39: Is javascript a static type or a dynamic type?

A-39: Javascript is a dynamically typed (loosely-typed) language. You can use variables, function, object or anything without declaring the type. The type checking is done during the runtime rather than compile time. Type is determined by the value of the variable in javascript.

ES6 INTERVIEW Q&A

Q-40: What is ES6?

A-40: ES6 stands for ECMAScript 6 or ECMAScript 2015 is the newer version of Javascript that was introduced in 2015. ECMAScript was created to standardise JavaScript. ECMAScript provides the specification on how Javascript programming language should work.

ES6 features:

- a. let and const keyword
- b. Template string
- c. default parameter
- d. arrow function
- e. spread operator
- f. destructuring
- g. multi-line string
- h. promise
- i. classes
- j. modules

Q-41: What are the differences between var, let, and const?

A-41: The scope of a var variable is functional scope. The scope of the let variable is block scope. The scope of the const variable is block scope. var variable can be updated or re-declared in this scope. let variable can be updated but can't be re declared. const variables can't be updated and re-declared into the scope. Var and let variables can be declared without initialization but const variables can't be declared without initialization.

Q-42: Why will you use default parameters?

A-42: If the desired number of parameters is not passed to a function, the default parameter allows named parameters to be initialised with default values. If the function is passed the correct number of parameters then default values will not work. Default values will only work if the function is not passed the correct number of parameters.

Q-43: How does the spread operator work?

A-43: spread operator expands the elements of an array into a new array. The JavaScript spread operator (...) allows us to quickly copy all or part of an existing array or object into another array or object.

Q-44: What is the difference between class and object?

A-44: A class defines object properties including a valid range of values, and a default value. A class also describes object behaviour. An object is a member or an "instance" of a class. An object has a state in which all of its properties have values that you either explicitly define or that are defined by default settings.

Q-45: What is a prototype chaining in js?

A-45: Prototype chaining means all the objects in JavaScript inherit the properties and methods from Object. prototype. This is called Prototype chaining. This is a very powerful and potentially dangerous mechanism to override or extend object behaviour. Objects created using new keywords inherit from a prototype called Object.

Q-46: Explain Call by value vs call by reference

A-46: If two primitive(number/string/boolean) variables become the same by copying the value but in 2 separate spots in the memory, then this approach is called call by value. If two not-primitive(array/object) variables become the same by copying the value and situated in the same spot in the memory, then this approach is called call by reference.

Q-47: What is the scope in javascript?

A-47: In javascript, scope determines the accessibility (visibility) of a variable. After ES6 was introduced in 2015 javascript has 3 scopes.

- a. **global scope:** Variables declared Globally (outside any function) have Global Scope. Global variables can be accessed from anywhere in a JavaScript program. They all have global scope.
- b. **local scope:** Variables declared within a JavaScript function, become LOCAL to the function. Local variables have Function Scope. They can only be accessed from within the function. Since local variables are only recognized inside their functions, variables with the same name can be used in different functions. Local variables are created when a function starts, and deleted when the function is completed.
- c. **Block scope:** After “let” and “const” were introduced in 2015, block scope arrived. Variables declared inside a { } block cannot be accessed from outside the block. A var variable can not have block scope.

Q-48: What is a Higher-order Function in js?

A-48: Higher order function is directly related to callback function. We can use a function inside another function. A callback function is a function that is passed as an argument to another function so that it can be executed in that function. In which function, another function is passed as an argument is called higher order function.

Q-49: What is API? Difference between Get vs post?

A-49: API: API stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses.

Difference between GET and POST: Both GET and POST method is used to transfer data from client to server in HTTP protocol but Main difference between POST and GET method is that GET carries request parameter appended in URL string while POST carries request parameter in message body which makes it more secure way of transferring data from client to server.

Q-50: Difference between local storage and Session storage?

A-50: The main difference between LocalStorage and SessionStorage is LocalStorage data never expires but the SessionStorage data cleared when the page session ends. The data storage capacity of LocalStorage is almost double of the capacity of SessionStorage.

Q-51: What are cookies? And why will you use it?

A-51: Cookies: Cookies (also referred to as HTTP cookies or browser cookies) are small text files stored in a web user's browser directory or data folder. We will use cookies to record the information of the user.

Q-52: What is Object Oriented Programming?

A-52: Object-oriented programming (OOP) is a style of programming characterised by the identification of classes of objects closely linked with the methods (functions) with which they are associated. It also includes ideas of inheritance of attributes and methods.

Q-53: Difference between Array and Linkedlist?

A-53:

Array: Array is a collection of elements of similar data types.

LinkedList: A linked list is a collection of objects known as a node where node consists of two parts - data and address.

Q-54: How will you debug javascript applications?

A-54: Usually I use `console.log()` frequently to display where the error occurred. I use breakpoints. I also use the debugger keyword to debug my js application. I have a clear concept of chrome devtools.

REACT INTERVIEW Q&A

Q-55: What is react?

A-55: React is a flexible front-end Javascript library for building user interfaces. React allows us to compose complex UIs to small isolated pieces of code called "components".

Advantages of react:

- a. It increases the performance of applications.
- b. Because of JSX, code readability increases.
- c. react is easy to integrate with other frameworks like Meteor, Angular, etc
- d. It is easy to learn and use.

- e. It comes with a good supply of documentation, tutorials, and training resources. Any developer who comes from a JavaScript background can easily understand and start creating web apps using React in a few days.
- f. It uses virtual dom instead of real dom. Virtual dom is faster than real dom.
- g. Creating dynamic web applications with React is easier.
- h. You can reuse react components wherever you need them.
- i. Traditional javascript frameworks have an issue in dealing with SEO. React overcomes this problem because React.js applications can run on the server, and the virtual DOM will be rendering and returning to the browser as a regular web page.
- j. ReactJS applications are extremely easy to test. It offers a scope where the developer can test and debug their codes with the help of native tools.

Disadvantage of React:

- a. Since the environment continually changes so fast, some of the developers are not feeling comfortable relearning the new ways of doing things regularly.
- b. React technologies update and accelerate so fast that there is no time to make proper documentation.
- c. React uses JSX. Some developers who are in their learning curve are not comfortable with JSX. Because JSX allows HTML with javascript mixed together. It increases complexity sometimes.
- d. React.js is just a library. It is not a full framework.
- e. It can be little difficult for the novice programmers to understand

Q-56: Why will you use React.js?

A-2: React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”. React uses virtual dom which is faster than the real dom. React allows developers to create large scalable web applications that can change data, without reloading the page. React allows us to create a single page application (SPA). I can reuse react components wherever I need. React is SEO friendly. React applications are super easy to test and debug.

Q-3 : What is Virtual dom? What are the differences between virtual and real dom?

A-3: virtual DOM: In React, for every DOM object, there is a corresponding “virtual DOM object.” A virtual DOM object is a representation of a DOM object, like a lightweight copy. Virtual DOM can update faster than real DOM. Virtual DOM manipulation is easier than Real DOM. Real DOM creates a new DOM if element updates. Virtual DOM updates the JSX id element updates.

Q-4 : What is the difference between props and state?

A-4: Props can be passed from one component to another. State can be passed within the components only. Props can't be modified. But the state can be modified. Prop is read only. State is both read and write.

Q-5 : What is the purpose of useState? When and why will you use it?

A-5: The useState hook is a special function that takes the initial state as an argument and returns an array of two entries. The first parameter is the initial state and the second one is a function that is used for updating the state.

When the value of a variable can be changed by the user or by page rerendering, then we need to keep that variable under an useState() hook. Because the useState() hook allows us to track state in a functional component.

Q-6: What is a context API? How does it work?

Context API: The context API is a feature of React that allows us to share data across the entire app. It is a way for a React app to effectively produce global variables that can be passed around. Context API is the alternative to props-drilling. Context API allows a react application to pass data directly to the child component without sharing data to the every level of components from the grand parent component.

React.createContext() returns a consumer and a provider. Provider is a component that as its name suggests provides the state to its children. It will hold the "store" and be the parent of all the components that might need that store. Consumer as it so happens is a component that consumes and uses the state.

Q-7 : Difference between useEffect and useState?

A-7:

useEffect () hook: The useEffect Hook allows us to perform side effects in our components. Some examples of side effects are: fetching data, directly updating the DOM, and timers. The useEffect hook takes two arguments. The first argument is a function and the second one is dependency array. The second one is optional.

useState () hook: The useState hook is a special function that takes the initial state as an argument and returns an array of two entries. The first parameter is the initial state and the second one is a function that is used for updating the state.

Q-8 : What is JSX? How does it work?

A-8: JSX: It stands for Javascript XML. JSX is a feature in react that allows us to mix HTML with javascript in React application. It is just an XML-like extension that allows us to write JavaScript that looks like HTML and have it returned from a component.

JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement() and/or appendChild() methods. JSX converts HTML tags into react elements. You are not required to use JSX, but JSX makes it easier to write React applications.

Q-9 : Tell us about React Component lifecycle

A-9: Every react component has a lifecycle with 3 different phases:

- a. mounting: Mounting means putting elements into the DOM
- b. updating: A component is updated whenever there is a change in the component's state or props.
- c. unmounting: The next phase in the life cycle is when a component is removed from the DOM

Q-10 : What is the purpose of a custom hook? How will you create a custom hook? Give us an example.

Purpose of custom hook: By using custom hook, we can reduce code repeat. The main reason to write a custom hook is for code reusability. For example, instead of writing the same code across multiple components that use the same common stateful logic (say a “setState” or localStorage logic), you can put that code inside a custom hook and reuse it.

The way I create a custom hook: (Example)

```
import { useState, useEffect } from "react";
import React from "react";

const UseServices = () => {
  const [services, setServices] = useState([]);

  useEffect(() => {
    fetch("services.json")
      .then(res => res.json())
      .then(data => setServices(data));
  });
}
```

```
    }, []);  
  
    return [services, setServices];  
  
}  
  
Export default UseServices;
```

Q-11 : How would you optimise a react js application?

A-11: Some ways to optimise react application as follows:

- a. using fragments (<> </>) instead of using unnecessary HTML tags.
- b. Avoid using index as key or map.
- c. Avoid Inline Function Definition in the Render Function
- d. Using a CDN
- e. CSS Animations Instead of JS Animations
- f. Avoid spreading props in a DOM element

CONCEPTUAL REACT INTERVIEW

Q&A

Q-1 : How will you send data from a Child Component to the parent component?

A-1: I can pass data from a child component to a parent component by following this steps:

- a. Pass a function as a prop to the child component
- b. Call the function from the child component and pass data as arguments
- c. Access the data in the function in the parent.

Q-2 : What is the best way to send 4 or more props to a child component?

A-2: Using context api is the best way to send 4 or more props to a child component.

Q-3 : What is Redux and and what is the purpose of Redux?

A-4: Redux is an open-source JavaScript library used to manage application state. React uses Redux for building the user interface.

React Redux is the official React binding for Redux. It allows React components to read data from a Redux Store, and dispatch Actions to the Store to update data. Redux helps apps to scale by providing a sensible way to manage state through a unidirectional data flow model.

Q-4 : What is React native? What do you know about React Native?

A-4: React Native (also known as RN) is a popular JavaScript-based mobile app framework that allows you to build natively-rendered mobile apps for ios and android. The framework lets you create an application for various platforms by using the same codebase.

Q-5 : What are Higher order components? Give us an example.

A-5: Higher-Order Component in React is a function that takes a component as an argument and returns a new component. It is the state-of-the-art technique in React for reusing component logic. The HOC adds additional data or functionality to an original component.

Q-6 : Is there any reason to return something from a useEffect hook?

A-6: Yes, there are some reasons. The useEffect Hook is built in a way that we can return a function inside it and this return function is where the cleanup happens. The cleanup function prevents memory leaks and removes some unnecessary and unwanted behaviours.

Q-7 : How will you optimise a react application?

A-7: Some ways to optimise react application as follows:

- a. using fragments (<> </>) instead of using unnecessary HTML tags.
- b. Avoid using index as key or map.
- c. Avoid Inline Function Definition in the Render Function
- d. Using a CDN
- e. CSS Animations Instead of JS Animations
- f. Avoid spreading props in a DOM element

Q-8 : What are the different ways to manage state in a React Application

A-8: There are four main types of state you need to properly manage in your React apps:

- a. Local state
- b. Global state
- c. Server state
- d. URL state

Q-9 : Why do we inject dependency inside a useEffect hook?

A-9: The dependency array basically tells the hook to "only trigger when the dependency array changes". If you have multiple elements in a dependency array, the hook will trigger if any element of the dependency array changes:

Q-10 : How will you prevent re-render in react applications?

A-10: Some ways to prevent re-render

- a. Using useMemo() and useCallback() hook
- b. Api call optimisation with react query
- c. using useRef() instead of useState()
- d. using react fragments

Q-11 : Some disadvantages of react?

A-11: Disadvantage of React:

- a. Since the environment continually changes so fast, some of the developers are not feeling comfortable relearning the new ways of doing things regularly.
- b. React technologies update and accelerate so fast that there is no time to make proper documentation.
- c. React uses JSX. Some developers who are in their learning curve are not comfortable with JSX. Because JSX allows HTML with javascript mixed together. It increases complexity sometimes.
- d. React.js is just a library. It is not a full framework.
- e. It can be little difficult for the novice programmers to understand

Q-12: Does React perform one-way data binding or two-way data binding?

A-12: React.js use one-way data binding

Node MongoDB Interview Q&A

Q-1: What is node.js?

A-1: Node.js is an open-source and cross-platform event-driven JavaScript runtime environment. Node.js is designed to build scalable network applications.

Q-2 : What is the difference between javascript and node?

A-2: javascript is a high-level programming language on the other hand, node is a javascript runtime environment. Javascript can only be run in the browsers. But with the help of node.js we can run javascript outside the browsers. Javascript is used in front-end development, node.js is used in backend development.

Q-3 : Nodejs single threaded or multi threaded?

A-3: Nodejs is single threaded.

Q-4: Nodejs blocking or non-blocking?

A-4: Nodejs based on a non-blocking I/O model

Q-5 : What is npm?

A-5: npm stands for node package manager. npm is the world's largest Software Registry. The registry contains over 800,000 code packages. Open-source developers use npm to share software. npm is a package manager for the JavaScript programming language.

Q-6 : What is the purpose of a database?

A-6: A database is an organised collection of structured information, or data, typically stored electronically in a computer system. A good database is crucial to any company or organisation. This is because the database stores all the pertinent details about the company such as employee records, transactional records, salary details etc.

Q-7 : What is the difference between a sql and nosql database?

A-7: SQL database stores data or information in an organised way usually in a tabular form. On the other hand, the NOSQL database stores information in a less organised way. At present nosql database like mongoDB is been used more than a sql database.

Q-8 : Why do we use mongoDB with nodejs in a react application?

A-8: Even though Node.js works well with a MySQL database, the perfect combination is a NoSQL like MongoDB wherein the stored data/information need not be well-structured. MongoDB represents the data as a collection of documents rather than in a tabular form. This makes it possible for the varied types of data dealt over the internet to be stored decently and accessed in the web applications using Node.js. The main advantage of using mongoDB with nodejs is that mongoDB stores data in BSON form which is easily accessible by nodejs. Because nodejs is an event-driven runtime environment of javascript.

Q-9 : What do you mean by database design, database schema?

A-10 : Database design: Database design is the organisation of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database management system manages the data accordingly.

Database schema: Database schema design organises the data into separate entities, determines how to create relationships between organised entities, and how to apply the constraints on the data. Designers create database schemas to give other database users, such as programmers and analysts, a logical understanding of the data.

Q-10 : What do you do when your server side crashes?

A-10: I usually do as follows-

- a. Try to identify the root causes
- b. Go through the error messages carefully
- c. Try to find out the status
- d. Try to fix issues

Q-11 : How does an API work?

A-11: The application (such as a website or a mobile app) will make an API call for a set of data to display for the end user to consume. The request is made via the API that accesses the web server to retrieve the requested data, which is populated in the user interface.

Q-12 : What is CRUD

A-12: CRUD is an acronym that comes from the world of computer programming and refers to the four functions that are considered necessary to implement a persistent storage application: create, read, update and delete.

Q-13 : What is the difference between get and post

A-13: GET is used to request data from a specified resource. On the other hand, POST is used to send data to a server to create/update a resource.

Get requests can be cached but post requests can never be cached. Get request remains in the browser's history. Post request does not remain in browser history.

Q-14 : What is the difference between put and patch?

A-14: PUT is used to send data to a server to create/update a resource. The PATCH method is used to apply partial modifications to a resource. The put method can duplicate information if that information already exists. But the patch method does not duplicate a particular piece of information. If the information does not exist then the patch method updates. But if the information already exists then the patch method does not update.

Q-15 : How will you secure an api?

A-15: Usually I use Json Web Token (JWT) to secure api

Q-16 : What is mongoose, how does it work?

A-16: Image result for what is mongoose mongodb

Mongoose is an Object Data Modelling (ODM) library for MongoDB and Node. js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

Q-17 :What is webpack?

A-17: Webpack is a popular module bundling system built on top of Node. js. It can handle not only combination and minification of JavaScript and CSS files, but also other assets such as image files (spriting) through the use of plugins.

Weaknesses:

Ans: 1.I like to play cricket very much, this cricket game wastes a lot of my time.
2.I can not lie.