

# Universal playbook

## Placement Preparation Resources:

[https://docs.google.com/document/d/1iVOvpMbk2wLlb30Lmo4\\_OHRucYQFsTe2McADRbejQyc/edit](https://docs.google.com/document/d/1iVOvpMbk2wLlb30Lmo4_OHRucYQFsTe2McADRbejQyc/edit)

## Commonly asked questions in previous interviews.

### HTML

- What is html how it works ?
- Is some compiler required to run html?
- What new update in html?
- What is the meaning of<DOCTYPE html>?
- Difference between HTML and XML.
- What do you mean by a markup language?
- Can you share examples of other markup languages and how they differ from HTML?
- What version of HTML do you use in your projects? How is HTML 5 different from HTML 4?
- What are attributes in HTML?
- What are data- attributes good for?
- Describe the difference between <script>, <script async> and <script defer>.
- Why is it generally a good idea to position CSS <link>s between <head></head> and JS <script>s just before </body>? Do you know any exceptions?

### CSS

- What is Position related
- What is Pseudo css?
- What is the Box model in **CSS**?
- What are the advantages of using **CSS**?
- What are the limitations of **CSS**?
- What is CSS and how does it work with HTML?
- What are selectors and what are their different types?
- What is CSS selector specificity and how does it work?
- Describe z-index and how stacking context is formed.
- Describe BFC (Block Formatting Context) and how it works.
- Have you ever used a grid system, and if so, what do you prefer?
- Have you used or implemented media queries or mobile specific layouts/CSS?
- Explain how a browser determines what elements match a CSS selector.
- Describe pseudo-elements and discuss what they are used for.
- Explain your understanding of the box model.
- What does \* { box-sizing: border-box; } do? What are its advantages?
- What is the CSS display property and can you give a few examples of its use?

- What's the difference between inline and inline-block?
- What's the difference between the "nth-of-type()" and "nth-child()" selectors?
- What's the difference between a relative, fixed, absolute and statically positioned element?
- What existing CSS frameworks have you used locally, or in production? How would you change/improve them?
- Have you used CSS Grid?

## JavaScript

- js java object-oriented
- how we will apply inheritance to js
- let vs const
- undefine & not define
- and some more question i forget
- What are the primitive data types in JS?
- What's the difference between a variable that is: null, undefined or undeclared?
- What is the difference between while and do-while loops in JavaScript?
- What language constructions do you use for iterating over object properties and array items?
- What are the promises and how do they work?
- What are IIFEs and explain with an example where they can be used?
- Explain event delegation.
- Explain how *this* works in JavaScript.
- a. Can you give an example of one of the ways that working with *this* has changed in ES6? Explain how prototypal inheritance works.
- What is a closure, and how/why would you use one?
- Can you describe the main difference between the Array.forEach() loop and Array.map() methods and why you would pick one versus the other?
- What's a typical use case for anonymous functions?
- What's the difference between host objects and native objects?
- Explain the difference between: function Person() {}, var person = Person(), and var person = new Person()?
- Explain the differences on the usage of foo between function foo() {} and var foo = function() {}
- Can you explain what Function.call and Function.apply do? What's the notable difference between the two?
- Explain Function.prototype.bind.
- What's the difference between feature detection, feature inference, and using the UA string?
- Explain "hoisting".
- Describe event bubbling.
- Describe event capturing.
- What's the difference between an "attribute" and a "property"?
- What are the pros and cons of extending built-in JavaScript objects?
- What is the difference between == and ===?
- Explain the same-origin policy with regards to JavaScript.
- Why is it called a Ternary operator, what does the word "Ternary" indicate?

- What is strict mode? What are some of the advantages/disadvantages of using it?
- What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?
- What tools and techniques do you use debugging JavaScript code?
- Explain the difference between mutable and immutable objects.
- a. What is an example of an immutable object in JavaScript?
- b. What are the pros and cons of immutability?
- c. How can you achieve immutability in your own code?
- Explain the difference between synchronous and asynchronous functions.
- What is an event loop, What is the difference between call stack and task queue?
- What are the differences between variables created using let, var or const?
- What are the differences between ES6 class and ES5 function constructors?
- Can you offer a use case for the new arrow => function syntax? How does this new syntax differ from other functions?
- What advantage is there for using the arrow syntax for a method in a constructor?
- What is the definition of a higher-order function?
- Can you give an example for destructuring an object or an array?
- Can you give an example of generating a string with ES6 Template Literals?
- Can you give an example of a curry function and why this syntax offers an advantage?
- What are the benefits of using spread syntax and how is it different from rest syntax?
- How can you share code between files?
- Why might you want to create static class members?

## React

- What is react?
- Differentiate between functional and class components. What do you prefer to use and in which situation?
- What is State and Props in react?
- What is the difference between State and Props? How do you decide what is controlled by state and what is controlled by props?
- What is Virtual DOM and how it works?
- What is the importance of Key in the map?
- What are the life-cycle methods in react?
- Which life-cycle method will you use when calling an API?
- How do we access the parent component in a child and vice versa. (in react)
- Have you used Context?
- How to connect react with backend?
- Why use Redux in react?
- Explain how to work with redux?
- What is the container pattern?
- What is middleware used for?

## REDUX

- What is redux?
- Why we need it?
- We can store our data in local storage rather than why we are using Redux.

- What is an action?
- What is typescript?

### **Backend Implementation**

- How can you implement filtering in APIs?
- How can you implement pagination in APIs?
- How does authentication work?
- What is indexing?

### **Conceptual-Node**

- What is Node.js?
- How does Node.js work?
- Differentiate between JavaScript and Node.js
- List down the major benefits of using Node.js?
- Where Node.js can be used?
- How many types of API functions are there in Node.js? Why is Node.js single-threaded?
- What happens when you open a new tab in the browser? Does it create a new thread or a new process? What are the benefits of using threads?
- What is an Event loop in Node.js and how does it work?
- Explain REPL in the context of Node.js?
- What are callbacks and what is callback hell in NodeJS?

### **DSA CONCEPTS:**

<https://www.w3schools.in/data-structures-tutorial/intro/>