

HTML Interview Topics and Questions

What is HTML?

All about meta tags in HTML .

What are the different types of HTML elements?

Explain the difference between HTML and XHTML.

What is the difference between `<div>` and ``?

What are semantic HTML elements?

Explain the difference between `<section>`, `<article>`, and `<aside>`.

What is the purpose of `<header>`, `<footer>`, and `<nav>`?

What is the difference between `` and ``?

What are block-level and inline elements? Give examples.

What is the difference between `` and `<i>`?

Forms & Input Elements

What are the different types of input elements in HTML5?

Explain the use of `<fieldset>` and `<legend>`.

What is the difference between `GET` and `POST` methods?

How can you validate a form in HTML?

. HTML5 Features

What new elements were introduced in HTML5?

What is the `<canvas>` element used for?

What is the `<video>` and `<audio>` tag?

What is the purpose of the `data-*` attribute?

HTML Accessibility

What are ARIA attributes?

How do you make a website more accessible using HTML?

What is the `alt` attribute used for?

Does a hyperlink only apply to text?

What is a style sheet?

Explain the layout of HTML?

How many tags can be used to separate a section of texts?

What are empty elements?

What is the use of a span tag? Give one example.

What is SVG?

Which type of video formats are supported by HTML5?

Is there any need to change the web browsers to support HTML5?

CSS Interview Topics and Questions

What is CSS and why is it used?

Explain the different ways to apply CSS to a webpage.

What is the difference between `relative`, `absolute`, `fixed`, and `sticky` positioning?

What are the different types of CSS selectors?

What are the components of the CSS Box Model?

How does `border-box` differ from `content-box`?

How do `margin` and `padding` differ?

CSS Flexbox & Grid

What is Flexbox and how does it work?

Explain the difference between `justify-content` and `align-items`.

What are the different Flexbox properties?

How does CSS Grid differ from Flexbox?

What are `grid-template-rows` and `grid-template-columns`?

CSS Transitions & Animations

What is the difference between a CSS transition and an animation?

How do you create a smooth transition effect?

What are keyframes in CSS animations?

Responsive Design & Media Queries

What are media queries in CSS?

How can you make a website responsive?

What is the difference between `em`, `rem`, `vh`, and `vw` units?

What are the different variations of CSS?

What are the advantages of CSS?

How can you integrate CSS on a web page?

What are the CSS frameworks?

Explain universal selector.

Which type of css holds the highest priority?

JavaScript

JavaScript Basics

Data Types (Primitive & Reference)

Variables (`var`, `let`, `const`)

Hoisting

Operators (Arithmetic, Logical, Comparison)

Type Coercion & Type Conversion

Conditional Statements (`if`, `switch`)

Loops (`for`, `while`, `do-while`)

Functions (Regular, Arrow, Anonymous, IIFE)

Scope (Global, Function, Block)

Closures

Temporal Dead Zone(TDZ)

Interview Questions

What are the different data types in JavaScript?

What is hoisting in JavaScript?

What is the difference between `let`, `const`, and `var`?

Explain the difference between `==` and `===` in JavaScript.

What is the difference between `null` and `undefined`?

How does the `typeof` operator work?

What is Closures and how to work?

Objects and Arrays

Object Literals & Constructors

Object Methods

Prototypes & Inheritance

__proto__

prototype property, constructor

Polyfills

ES6 Classes

Array Methods

(`map()`, `filter()`, `reduce()`, etc.)

Destructuring

Interview Questions

How do you create an object in JavaScript?

What is the difference between deep and shallow copies in JavaScript?

How does prototypal inheritance work in JavaScript?

What are the different ways to clone an object in JavaScript?

What is the difference between `map()`, `forEach()`, and `filter()`?

Functions & Execution Context

Function Expressions vs. Declarations

First-Class Functions & Higher-Order Functions

Callbacks & Promises

Event Loop & Concurrency Model

`this` Keyword

`bind()`, `call()`, `apply()`

IIFE - Immediately Invoked

Function Expression

Interview Questions

What is the difference between function declaration and function expression?

What is a higher-order function in JavaScript?

Explain how `this` works in JavaScript.

What is the difference between `bind()`, `call()`, and `apply()`?

Explain how the JavaScript event loop works.

Asynchronous JavaScript

Callbacks

Promises & Chaining

`async/await`

Event Loop, Microtasks, and Macrotasks

Fetch API vs. Axios

Interview Questions

What is the difference between synchronous and asynchronous JavaScript?

Explain how promises work in JavaScript.

What are `async/await`, and how do they improve readability?

How does the JavaScript event loop handle asynchronous code?

What is the difference between microtasks and macrotasks?

DOM Manipulation & Events

DOM Traversal & Manipulation

Event Bubbling & Capturing

`addEventListener()`

Preventing Default Behavior

Debouncing & Throttling

Interview Questions

What is the DOM, and how does JavaScript interact with it?

What is event delegation in JavaScript?

What is the difference between event bubbling and event capturing?

How do you debounce and throttle a function in JavaScript?

What is the difference between `innerHTML` and `textContent`?

ES6+ Features

Arrow Functions

Template Literals

Spread & Rest Operators

Destructuring

Modules (`import` / `export`)

Default Parameters

Optional Chaining (`?.`)

Interview Questions

What is the difference between function expressions and arrow functions?

How does destructuring work in JavaScript?

What is the purpose of the spread and rest operators?

Explain how JavaScript modules work.

What is optional chaining, and how does it help in JavaScript?

Error Handling & Debugging

`try...catch`

Throwing Custom Errors

Debugging Techniques (`console.log`, breakpoints)

Interview Questions

How does error handling work in JavaScript?

What is the difference between `throw` and `return`?

How do you debug JavaScript code efficiently?

What is the `finally` block used for?

How can you catch errors in an `async/await` function

Memory Management & Performance

Garbage Collection

Memory Leaks

Performance Optimization

JavaScript Execution Optimization

Interview Questions

How does JavaScript handle memory management?

What causes memory leaks in JavaScript?

How can you optimize JavaScript performance?

What are some ways to improve load times in JavaScript?

How does lazy loading work in JavaScript?

Advanced JavaScript Concepts

Currying

Memoization

Functional Programming

Metaprogramming ([Proxy](#), [Reflect](#))

Interview Questions

What is currying, and why is it useful?

Explain memoization in JavaScript.

How do Proxies work in JavaScript?

What is functional programming, and how does JavaScript support it?

What is the difference between imperative and declarative programming?

React.js

Basics

What is React?

Key features of React

Virtual DOM vs Real DOM

React Components

(Functional vs Class Components)

JSX and Babel

Props and State

Handling Events in React

Questions

What is React, and why is it used?

Explain the difference between functional and class components.

What is JSX, and how does it work?

How does the Virtual DOM improve performance?

What are props in React? Can props be modified?

What is state in React? How is it different from props?

How do you handle events in React?

React Component Lifecycle

Lifecycle methods in Class Components

Hooks and Functional Component Lifecycle

useEffect and Cleanup Functions

Questions

Explain the lifecycle methods of a React class component.

What is the significance of `componentDidMount()`?

How does `useEffect` replace lifecycle methods in functional components?

What is the difference between `useEffect` and `componentDidUpdate`?

How do you prevent unnecessary re-renders in React?

State Management

useState Hook

Lifting State Up

Context API

Redux (Basics & Advanced)

Questions

What is the difference between `useState` and `useReducer`?

Why and when should you use Context API?

How does Redux work, and what are its core principles?

What are actions, reducers, and stores in Redux?

How does the `useContext` hook work?

React Hooks

What are hooks?

useState, useEffect, useRef, useMemo, useCallback

Custom Hooks

Questions

What are React Hooks? Why were they introduced?

Explain the difference between `useEffect` and `useLayoutEffect`.

What is the purpose of `useRef`?

How does `useMemo` help in optimization?

How do you create a custom hook in React?

React Routing

React Router v6

Route Parameters & Nested Routes

Protected Routes

Redirects and Navigation

Questions

What is React Router, and why is it used?

How do you implement dynamic routing in React?

What is the difference between `useNavigate` and `useHistory`?

How do you create a private (protected) route in React?

What are nested routes in React Router?

Forms and Controlled Components

Controlled vs Uncontrolled Components

useRef in Forms

Form Validation (Formik, Yup)

Questions

What is the difference between controlled and uncontrolled components?

How do you handle form submission in React?

What are the benefits of using Formik over traditional form handling?

How can you validate forms in React?

When should you use `useRef` instead of state for form inputs?

Server Communication (API Handling)

Fetch API vs Axios

Handling API Requests with `useEffect`

Error Handling and Loading States

React Query

Questions

How do you fetch data from an API in React?

What is the difference between Fetch API and Axios?

How do you handle API errors in React?

What are the benefits of using React Query?

How do you implement caching in React Query?

Advanced React Concepts

Higher-Order Components (HOC)

Portals

Error Boundaries

Suspense and Concurrent Mode

Questions

What are Higher-Order Components, and how do they work?

When should you use React Portals?

What are Error Boundaries? How do they help in debugging?

What is Suspense, and how does it improve React performance?

How does Concurrent Mode enhance the user experience?

Testing in React

Unit Testing with Jest

React Testing Library

Mocking API Requests

Cypress for End-to-End Testing

Questions

1. What is the purpose of testing in React applications?
2. How do you write a unit test using Jest?
3. How does React Testing Library differ from Enzyme?
4. How do you mock an API call in Jest?
5. What are the advantages of Cypress for UI testing?

Redux Interview Questions

What is Redux, and how does it work?

What are the core principles of Redux?

What are actions in Redux? Provide an example.

Explain the role of reducers in Redux.

What is the Redux store, and how does it work?

How do you dispatch an action in Redux?

What is the purpose of the `combineReducers` function?

Explain the role of middleware in Redux.

What are the advantages and disadvantages of using Redux?

What is Redux Thunk, and how does it work?

What is the difference between Redux Thunk and Redux Saga?

How do you handle side effects in Redux?

How can you improve Redux performance?

What is Redux Toolkit, and how does it simplify Redux development?

What is `redux-select`, and why is it useful in Redux?

How would you handle large amounts of data in a Redux store?

What would you do if a Redux application becomes slow?

How would you structure a Redux store for a complex application?

How do you persist the Redux state across page reloads?

When would you choose Redux over the Context API?