TOP 50 INTERVIEW QUESTIONS FOR WEB DEVELOPMENT

1. What is the difference between id and class in HTML?
Answer:

- id is unique and used once per page.
- class can be used multiple times for styling or scripting.

2. How do HTML5 semantic elements improve SEO and accessibility?
Answer:
Semantic elements like <article>, <section>, <nav>, and <header> describe content structure, helping search engines index better and screen readers understand content context.

3. What is the box model in CSS?
Answer:
The box model includes: content → padding → border → margin. It defines how elements are sized and spaced.

4. How do you center a div using Flexbox?
Answer:

display: flex;

justify-content: center;

align-items: center;

5. What's the difference between relative, absolute, and fixed positioning?
Answer:

- relative: moves relative to its normal position.
- absolute: positions relative to the nearest positioned ancestor.
- fixed: stays fixed to the viewport.

◈ CSS Concepts

6. What is specificity in CSS?
Answer:
It defines which rule takes priority. Calculated using: Inline styles > IDs > Classes > Elements.

7. What are media queries?
Answer:
They make websites responsive.

@media (max-width: 768px) { ... }

8. What is the difference between em, rem, px, and % units?
Answer:

- px: absolute.

- em: relative to parent.
- rem: relative to root.
- %: relative to container.

9. How does z-index work?
Answer:
It controls stacking order of overlapping elements. Higher z-index = on top.

10. What are pseudo-elements and pseudo-classes?
Answer:

- Pseudo-class: :hover, :first-child (user interaction).
- Pseudo-element: ::before, ::after (inject content).

◈ JavaScript Fundamentals

11. What is hoisting?
Answer:
Variable and function declarations are moved to the top of their scope at compile time.

12. Difference between == and ===?
Answer:

- == checks value only.
- === checks value **and** type.

13. What is a closure?
Answer:
A function that remembers its lexical scope even after the outer function has closed.

```
function outer() {
  let x = 10;
  return function inner() {
    console.log(x);
  };
}
```

14. Difference between var, let, and const?
Answer:

- var: function-scoped, hoisted.
- let: block-scoped.
- const: block-scoped, cannot be reassigned.

15. What is the event loop?
Answer:
It allows non-blocking async operations by handling the call stack and the task queue.

🔹 Advanced JavaScript

16. How does this behave in different contexts?
Answer:

- In global scope: window.
- In object methods: the object.
- In arrow functions: lexically scoped (this is inherited).

17. Synchronous vs. Asynchronous code?
Answer:

- Synchronous: blocks further execution.
- Asynchronous: doesn’t block, uses callbacks/promises.

18. What are promises and async/await?
Answer:
Promises handle async operations; async/await provides cleaner syntax.

19. Debouncing vs Throttling?
Answer:

- Debounce: delays function until pause.
- Throttle: limits function to once per interval.

20. Difference between null and undefined?
Answer:

- undefined: a variable declared but not assigned.
- null: intentional empty value.

🔹 React

21. What are React hooks?
Answer:
Functions like useState, useEffect, useContext to manage state and side-effects.

22. Controlled vs Uncontrolled components?
Answer:

- Controlled: form data managed via React state.
- Uncontrolled: DOM handles input state.

23. What is the virtual DOM?
Answer:
A lightweight copy of real DOM. React updates it efficiently, then syncs with real DOM.

24. What are props and state?
Answer:

- props: read-only, passed by parent.
- state: local, managed within component.

25. How does lifting state up work?
Answer:
Moving state to the closest common ancestor to share data between components.

26. What are keys in React?
Answer:
Unique identifiers for list items. Helps React optimize rendering.

27. Difference between useEffect and useLayoutEffect?
Answer:

- useEffect: after render.
- useLayoutEffect: before paint (used for layout read/write).

28. How does React handle reconciliation?
Answer:
By comparing virtual DOM trees and updating only changed parts.

29. What is Context API?
Answer:
Provides global state without prop drilling.

30. What are Higher Order Components (HOCs)?
Answer:
Functions that take a component and return a new component with added behavior.

◈ Node.js & Express

31. What is Node.js?
Answer:
A runtime for executing JavaScript on the server using V8 engine.

32. What are middleware functions in Express?
Answer:
Functions that run before request handlers to process requests, like logging or authentication.

33. How does the event-driven model work?
Answer:
Node uses non-blocking I/O and event loops to handle concurrency with callbacks.

34. Benefits of using Express.js?
Answer:
Minimal, fast, flexible with robust routing and middleware support.

35. How do you handle errors in Node.js?
Answer:
Using try...catch, error-handling middleware, and next(err).

36. What is package.json?
Answer:
A file that manages project metadata and dependencies.

🔷 APIs, Databases & Security

37. HTTP Methods and use cases?
Answer:

- GET: retrieve
- POST: create
- PUT: replace
- PATCH: update
- DELETE: remove

38. What are streams in Node.js?
Answer:
Efficient data handling via readable/writable pipelines, especially for large files.

39. Difference between require and import?
Answer:

- require: CommonJS (Node).
- import: ES Modules (modern JavaScript).

40. Managing env variables in Node.js?
Answer:
Use .env with dotenv package:

```
require('dotenv').config();
```

41. SQL vs NoSQL?
Answer:

- SQL: relational, structured.
- NoSQL: non-relational, flexible schema.

42. What are CRUD operations?
Answer:
Create, Read, Update, Delete — basic DB operations.

43. What is REST?
Answer:
Representational State Transfer — API design principle using stateless communication.

44. PUT vs PATCH?
Answer:

- PUT: full update.
- PATCH: partial update.

45. HTTP Status Codes:

- 200: OK
- 404: Not Found
- 500: Server Error

46. How to secure a REST API?
Answer:

- Use HTTPS
- Validate input
- Use JWT/auth
- Rate limit
- Sanitize data

47. What is CORS?
Answer:
Cross-Origin Resource Sharing. It controls access from other origins. Handled via headers.

48. What is indexing in databases?
Answer:
A data structure to improve search/query speed.

49. What is JWT?
Answer:
JSON Web Token — used for stateless authentication. Contains user data and is signed.

50. How do you structure a RESTful API?
Answer:
Organize by resource:

```
GET    /users

POST   /users

PUT    /users/:id

DELETE /users/:id
```

Would you like this as a **PDF**, a **quiz**, or **interactive flashcards** next?