web development interview questions and answers:

1. HTML Questions

1. **What is HTML?**
   - HTML (HyperText Markup Language) is the standard language used to create web pages and applications.
2. **What are HTML5 new features?**
   - New elements (`<article>`, `<section>`, `<nav>`, `<video>`, `<audio>`)
   - Local storage
   - Canvas and SVG support
   - Geolocation API
3. **What is the difference between `<div>` and `<span>`?**
   - `<div>` is a block-level element, while `<span>` is an inline element.
4. **What is the purpose of the `<meta>` tag?**
   - Provides metadata about the web page (e.g., character set, viewport settings).
5. **What is semantic HTML?**
   - Using meaningful elements like `<header>`, `<footer>`, `<article>`, etc.

---

2. CSS Questions

6. **What are the different types of CSS?**
   - Inline CSS, Internal CSS, External CSS
7. **What is the difference between `relative`, `absolute`, and `fixed` positioning?**
   - `relative`: Relative to its normal position.
   - `absolute`: Positioned relative to its nearest positioned ancestor.
   - `fixed`: Stays fixed relative to the viewport.
8. **What is Flexbox in CSS?**
   - A layout model for distributing space in a container with `display: flex`.
9. **What is the difference between `em`, `rem`, `%`, `vw`, and `vh`?**
   - `em`: Relative to parent font size.

   - `rem`: Relative to the root element’s font size.
   - `%`: Relative to the parent element.
   - `vw`: Percentage of the viewport width.
   - `vh`: Percentage of the viewport height.

10. **What is Grid Layout in CSS?**

- A two-dimensional layout system using `display: grid`.Here are **20 coding questions** for web development interviews, covering **HTML, CSS, JavaScript, React, and Node.js**.

---

1. HTML & CSS Coding Questions

11: Create a simple responsive navigation bar using HTML & CSS.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Responsive Navbar</title>
    <style>
        * { margin: 0; padding: 0; box-sizing: border-box; }
        body { font-family: Arial, sans-serif; }
        .navbar { background: #333; padding: 10px; display: flex; justify-content: space-between; align-items: center; }
        .navbar a { color: white; text-decoration: none; padding: 10px; }
        .menu { display: flex; }
        .menu a:hover { background: #555; }
        @media (max-width: 600px) {
            .menu { flex-direction: column; display: none; }
            .menu.show { display: flex; }
        }
    </style>
</head>
<body>
    <nav class="navbar">
        <a href="#">Logo</a>
        <div class="menu">
            <a href="#">Home</a>
            <a href="#">About</a>
            <a href="#">Contact</a>
        </div>
    </nav>
</body>
</html>
```

12: Center a `div` both vertically and horizontally.

```
.center-div {
   display: flex;
   justify-content: center;
   align-items: center;
   height: 100vh;
}
```

13: Create a button with a hover effect using CSS.

```
.button {
   background-color: blue;
   color: white;
   padding: 10px 20px;
   border: none;
   transition: 0.3s;
}
.button:hover {
   background-color: darkblue;
}
```

---

2. JavaScript Coding Questions

14: Reverse a string in JavaScript.

```
function reverseString(str) {
   return str.split('').reverse().join('');
}
console.log(reverseString("hello")); // Output: "olleh"
```

15: Find the largest number in an array.

```
function findLargest(arr) {
   return Math.max(...arr);
}
console.log(findLargest([2, 8, 1, 9, 3])); // Output: 9
```

16: Check if a number is prime.

```
function isPrime(num) {
   if (num < 2) return false;
   for (let i = 2; i < num; i++) {
      if (num % i === 0) return false;
   }
   return true;
}
console.log(isPrime(7)); // Output: true
```

17: Write a function to remove duplicates from an array.

```
function removeDuplicates(arr) {
    return [...new Set(arr)];
}
console.log(removeDuplicates([1, 2, 2, 3, 4, 4])); // Output: [1, 2, 3, 4]
```

18: Find the factorial of a number using recursion.

```
function factorial(n) {
    return n === 0 ? 1 : n * factorial(n - 1);
}
console.log(factorial(5)); // Output: 120
```

19: FizzBuzz Problem

```
for (let i = 1; i <= 20; i++) {
    console.log(i % 3 === 0 && i % 5 === 0 ? "FizzBuzz" : i % 3 === 0 ? "Fizz" : i % 5
=== 0 ? "Buzz" : i);
}
```

20: Find missing number in an array.

```
function findMissingNumber(arr) {
    let n = arr.length + 1;
    let sum = (n * (n + 1)) / 2;
    let actualSum = arr.reduce((acc, num) => acc + num, 0);
    return sum - actualSum;
}
console.log(findMissingNumber([1, 2, 4, 5, 6])); // Output: 3
```

---

3. React.js Coding Questions

21: Create a simple React component that displays a button and a counter.

```
import React, { useState } from 'react';

function Counter() {
    const [count, setCount] = useState(0);
    return (
        <div>
            <h2>Counter: {count}</h2>
            <button onClick={() => setCount(count + 1)}>Increase</button>
        </div>
    );
}
```

```

export default Counter;


```

22: Fetch data from an API and display it in React.

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

function Users() {
    const [users, setUsers] = useState([]);

    useEffect(() => {
        fetch("https://jsonplaceholder.typicode.com/users")
            .then(response => response.json())
            .then(data => setUsers(data));
    }, []);

    return (
        <div>
            <h2>User List</h2>
            <ul>
                {users.map(user => <li key={user.id}>{user.name}</li>)}
            </ul>
        </div>
    );
}

export default Users;


```

23: Create a to-do list in React.

```
import React, { useState } from 'react';

function TodoList() {
    const [tasks, setTasks] = useState([]);
    const [task, setTask] = useState('');

    const addTask = () => {
        setTasks([...tasks, task]);
        setTask('');
    };

    return (
        <div>
            <input type="text" value={task} onChange={(e) => setTask(e.target.value)} />
            <button onClick={addTask}>Add Task</button>
            <ul>
                {tasks.map((t, index) => <li key={index}>{t}</li>)}
            </ul>
        </div>
    );
}

export default TodoList;
```

4. Node.js & Backend Coding Questions

24: Create a simple Express.js server.

```
const express = require('express');
const app = express();
const PORT = 3000;

app.get('/', (req, res) => {
    res.send('Hello World!');
});

app.listen(PORT, () => {
    console.log(`Server is running on port ${PORT}`);
});
```

25: Create a simple API endpoint using Express.js.

```
app.get('/api/users', (req, res) => {
    res.json([{ id: 1, name: 'John' }, { id: 2, name: 'Jane' }]);
});
```

26: Connect to MongoDB using Mongoose.

```
const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost:27017/mydatabase', { useNewUrlParser: true, useUnifiedTopology: true })
    .then(() => console.log('Connected to MongoDB'))
    .catch(err => console.error(err));
```

27: Hash a password using bcrypt.

```
const bcrypt = require('bcrypt');
const password = 'mypassword';

bcrypt.hash(password, 10, (err, hash) => {
    console.log(hash);
});
```

- **28: Implement JWT authentication.**
- const jwt = require('jsonwebtoken');
- const token = jwt.sign({ id: 1 }, 'secretkey', { expiresIn: '1h' });
- console.log(token);

3. JavaScript Questions

29.What is the difference between `==` and `===` in JavaScript?

- `==` checks for value equality (allows type coercion).
- `===` checks for both value and type equality.

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

- `map()`: Returns a new array.
- `forEach()`: Executes a function for each element, does not return a new array.
- `filter()`: Returns a new array with elements that satisfy a condition.

31. **What is SQL vs NoSQL?**
- SQL: Structured, relational databases.
- NoSQL: Flexible, non-relational databases.

32. **What is normalization in databases?**
- Organizing data to minimize redundancy.

33. **What are the different types of joins in SQL?**
- INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN.

34. **What is indexing in a database?**
- A technique to speed up data retrieval.

35. **What is ACID in databases?**
- Atomicity, Consistency, Isolation, Durability.

---

8. General Web Development Questions

36. **What is the difference between HTTP and HTTPS?**
- HTTPS is secure and uses SSL/TLS encryption.

37. **What are cookies, localStorage, and sessionStorage?**
- Cookies: Small data stored by websites.
- localStorage: Stores data with no expiration.
- sessionStorage: Stores data per session.

38. **What is a CDN?**
- A Content Delivery Network used for faster content delivery.

39. **What is CORS?**
- Cross-Origin Resource Sharing, allows web pages to request resources from different origins.

40. **What is the difference between client-side and server-side rendering?**

- Client-side: Renders pages on the browser.
- Server-side: Renders pages on the server before sending them.

Here are **questions 41-100** for web development interviews:

---

9. Web Performance & Optimization Questions

41. **What is lazy loading?**
- Lazy loading delays the loading of non-essential resources until they are needed.

42. **What are WebP images?**
- WebP is a modern image format that provides better compression than JPEG or PNG.

43. **What is the difference between throttling and debouncing?**
- Throttling: Executes a function at most once in a given period.
- Debouncing: Delays execution until a pause occurs.

44. **What is a Service Worker?**
- A script that runs in the background for caching and push notifications.

45. **What are critical rendering paths?**
- The steps a browser takes to convert HTML, CSS, and JavaScript into a rendered page.

46. **How does DNS lookup work?**
- It translates domain names into IP addresses.

47. **What are Content Security Policies (CSP)?**
- A security feature that prevents cross-site scripting (XSS) attacks.

48. **What is TTFB (Time to First Byte)?**
- The time it takes for the first byte of a page to be received from the server.

49. **What is the difference between minification and compression?**
- Minification removes unnecessary characters, while compression reduces file size.

50. **What is a PWA (Progressive Web App)?**
- A web app that behaves like a native app, using Service Workers and Web Manifests.

---

10. JavaScript Advanced Concepts

51. **What is memoization?**
- A technique to store computed results and reuse them for performance optimization.

52. **What is the event loop in JavaScript?**
- A mechanism that handles asynchronous operations in JavaScript.

53. **What is the difference between `apply()`, `call()`, and `bind()`?**
- `call()`: Calls a function with arguments individually.
- `apply()`: Calls a function with arguments as an array.
- `bind()`: Returns a new function with the specified `this`.

54. **What is the difference between deep copy and shallow copy?**
- Shallow copy copies object references, while deep copy clones nested objects.

55. **What is hoisting in JavaScript?**
- Moving variable and function declarations to the top of their scope before execution.

56. **What are higher-order functions?**
- Functions that take other functions as arguments or return them.

57. **What are JavaScript generators?**
- Special functions that allow pausing and resuming execution using `yield`.

58. **What is `this` in JavaScript?**
- `this` refers to the execution context (object that calls the function).

59. **What is currying in JavaScript?**
- Transforming a function with multiple arguments into a sequence of functions.

60. **What is a polyfill?**
- A piece of code that adds modern functionality to older browsers.

---

11. React.js Advanced Questions

61. **What are React hooks?**
- Functions like `useState`, `useEffect`, and `useContext` that let you use state and lifecycle features in functional components.

62. **What is the Virtual DOM?**
- A lightweight JavaScript representation of the actual DOM that improves performance.

63. **What is React Fiber?**
- A new reconciliation algorithm in React for rendering.

64. **What is the difference between controlled and uncontrolled components?**
- Controlled: Managed by React state.
- Uncontrolled: Uses the DOM for state management.

65. **What is `useEffect` used for?**
- Side effects like API calls, subscriptions, and DOM manipulations.

66. **What is context API in React?**
- A way to share state between components without prop drilling.

67. **What are React Portals?**
- A way to render components outside the parent hierarchy.

68. **What is the difference between `React.Fragment` and a `<div>`?**
- `React.Fragment` groups elements without adding an extra DOM node.

69. **What is the difference between SSR (Server-Side Rendering) and CSR (Client-Side Rendering)?**
- SSR: Renders on the server, faster initial load.
- CSR: Renders in the browser, more dynamic.

70. **What are React Suspense and React Lazy?**
- Used to load components asynchronously for better performance.

---

12. Node.js & Backend Questions

71. **What is event-driven programming in Node.js?**
- A programming paradigm where actions (events) trigger code execution.

72. **What is the difference between process and thread?**
- Process: An instance of a program.
- Thread: A smaller unit of execution within a process.

73. **What is middleware in Express.js?**
- Functions that process requests before they reach the final handler.

74. **What is a stream in Node.js?**
- A way to handle I/O efficiently using chunks.

75. **What is the difference between blocking and non-blocking I/O?**
- Blocking I/O waits for an operation to complete.
- Non-blocking I/O allows other operations to continue.

76. **What is clustering in Node.js?**
- Running multiple instances of Node.js to use multi-core CPUs.

77. **What is the difference between synchronous and asynchronous execution in Node.js?**
- Synchronous: Executes in sequence, blocking further execution.
- Asynchronous: Executes tasks independently without blocking.

78. **What is the difference between `fs.readFileSync()` and `fs.readFile()`?**
- `readFileSync()` is synchronous, `readFile()` is asynchronous.

79. **What is REST vs GraphQL?**
- REST: Uses fixed endpoints for structured data retrieval.
- GraphQL: Allows flexible queries on a single endpoint.

80. **What is WebSocket?**
- A full-duplex communication protocol for real-time apps.

13. Security Questions

81. **What is XSS (Cross-Site Scripting)?**
- Injecting malicious scripts into web applications.

82. **What is CSRF (Cross-Site Request Forgery)?**
- Attacker tricks a user into making an unwanted request.

83. **What is SQL Injection?**
- Injecting malicious SQL queries into databases.

84. **What is HTTPS vs HTTP?**
- HTTPS encrypts data, HTTP does not.

85. **How does OAuth work?**
- A secure way to authenticate users via third-party providers.

86. **What is a Same-Origin Policy?**
- A security feature that blocks scripts from different origins.

87. **What is Two-Factor Authentication (2FA)?**
- A security method requiring two types of verification.

88. **What are security headers in HTTP?**
- Headers like `Content-Security-Policy` and `X-Frame-Options`.

89. **What is rate limiting?**

- Restricting the number of requests to prevent abuse.

90. **What is an SSL/TLS certificate?**
- Encrypts communication between a client and a server.

---

14. DevOps & Deployment

91. **What is Docker?**
- A tool for creating and managing lightweight, portable containers.

92. **What is CI/CD?**
- Continuous Integration and Continuous Deployment for automated testing and releases.

93. **What is the difference between a monolithic and microservices architecture?**
- Monolithic: All features in one codebase.
- Microservices: Separate services for different features.

94. **What is a load balancer?**
- Distributes traffic across multiple servers.

95. **What is Kubernetes?**
- A system for automating containerized applications.

96. **What is the difference between Git and GitHub?**
- Git: A version control system.
- GitHub: A platform for hosting Git repositories.

97. **What is Jenkins?**
- A CI/CD automation server.

98. **What is WebAssembly?**
- A binary format for running code in the browser.

99. **What is GraphQL?**
- A query language for APIs.

100. **What is the difference between SOAP and REST APIs?**
- SOAP: XML-based, strict protocol.
- REST: Flexible, JSON-based.