REACT.JS QUESTIONS

1.Why React was introduced?

Answer: React was introduced to make building **dynamic and interactive user interfaces (UI)** easier and faster.
Before React, developers had to manually update the webpage every time data changed — which was slow and complex.
React solves this by using a **Virtual DOM** that updates only the changed parts of a webpage efficiently.

Example:
When you “like” a post on Facebook, only that like button updates — the entire page doesn’t reload.
This is possible because of React’s Virtual DOM.

2. What is React?

Answer:
React is a **JavaScript library** used for building **user interfaces (UI)**, especially **single-page applications (SPA)**.

It helps developers create reusable UI components like buttons, forms, cards, etc.

Example:

```jsx
function Welcome() {
return

# Hello, Naveen!

;
}
```

This small code creates a **component** that displays “Hello, Naveen!”

3️⃣ Who developed React?

Answer:
React was developed by **Jordan Walke**, a software engineer at **Facebook (now Meta)**, and it was released in **2013**.

4️⃣ Why do we use React?

Answer:
We use React because:

* ✅ It’s **fast** (Virtual DOM updates only necessary parts)
* ✅ It’s **reusable** (components can be reused)
* ✅ It’s **maintainable** (code is modular)
* ✅ It allows **real-time UI updates** without refreshing the page

Example:
In a shopping website, when you add an item to your cart, React updates the cart count instantly.

5️⃣ Difference between HTML webpage and React webpage

Feature	HTML Webpage	React Webpage
Structure	Built using HTML, CSS, JS separately	Built using components (JSX)
Reload	Page reloads for every change	No reload (Single Page App)
Data Update	Manual DOM manipulation	Virtual DOM updates automatically
Speed	Slower for dynamic content	Faster and efficient
Reusability	Hard to reuse code	Components are reusable

6️⃣ Difference between React, Angular, and Vue

Feature	React	Angular	Vue
Type	Library	Framework	Framework
Language	JavaScript / JSX	TypeScript	JavaScript
Developer	Facebook	Google	Evan You
Learning Curve	Easy to learn	Difficult	Easy
DOM	Virtual DOM	Real DOM (with optimization)	Virtual DOM
Use Case	Dynamic, fast apps (Facebook, Instagram)	Large enterprise apps	Lightweight apps

---

7️⃣ What is TypeScript?

Answer:
TypeScript is a **superset of JavaScript** that adds **type safety**.
It helps catch errors during development (before running the code).

Example:

```typescript
let age: number = 25; // ✅ valid
age = "twenty"; // ❌ Error: Type 'string' is not assignable to type 'number'
```

TypeScript makes code more reliable and easier to debug — it’s commonly used with **React and Angular**.

🔹 **JAVASCRIPT QUESTIONS**

8️⃣ What is a Variable?

Answer:
A **variable** is used to store data in memory.
Example:

```javascript
let name = "Naveen";
let age = 21;
```

Here, `name` and `age` are variables.

9️⃣ What is Declaration?

Answer:
Declaration means creating a variable (without giving value).
Example:

```javascript
let a; // declaration
```

🔟 What is Initialization?

Answer:
Initialization means assigning a value to a declared variable.
Example:

```javascript
a = 10; // initialization
```

Or together:

```javascript
let a = 10; // declaration + initialization
```

11️⃣ What is Scope?

Answer:
Scope means the area where a variable can be accessed.

There are mainly:

* Global scope
* Function scope
* Block scope

12️ What is Lexical Scope?

Answer:
Lexical scope means **a function can access variables from its parent (outer) function**.

Example:

```javascript
function outer() {
let name = "Naveen";
function inner() {
console.log(name); // inner can access outer variable
}
inner();
}
outer(); // Output: Naveen
```

13️ What is Block Scope?

Answer:
Variables declared inside `{}` using `let` or `const` are **block scoped** (cannot be accessed outside).

Example:

```javascript
{
let x = 10;
}
console.log(x); // ️ Error: x is not defined
```

14️ What is Functional Scope?

Answer:
Variables declared inside a function using `var` are **function scoped** (accessible only inside that function).

Example:

```javascript
```

```
function demo() {
var x = 20;
}
console.log(x); // ❌ Error: x is not defined
```

15️⃣ What is Hoisting?

Answer:
Hoisting means JavaScript moves variable and function declarations to the top of their scope before execution.

Example:

```javascript
console.log(a); // undefined
var a = 10;
```

Here, JS internally does this:

```javascript
var a;
console.log(a); // undefined
a = 10;
```

16️⃣ What are Closures?

Answer:
A **closure** is a function that remembers the variables from its outer function even after the outer function has finished executing.

Example:

```javascript
function outer() {
let count = 0;
return function inner() {
count++;
console.log(count);
}
}

const counter = outer();
counter(); // 1
counter(); // 2
```

The `inner` function remembers `count` even after `outer()` is done — that’s a closure.

---

17 In web development process — why Node is used? Which came first: Node, React, or Angular?

Answer:

* **Node.js** is a **runtime environment** that lets JavaScript run **outside the browser** (like on a server).
* It allows developers to build **backend (server-side)** applications using JavaScript.

Example:
Before Node.js, JavaScript could only run inside browsers.
With Node.js, we can create servers, connect databases, and build APIs.

 Order of release:

1. **Node.js** – 2009
2. **AngularJS** – 2010
3. **React.js** – 2013

 So, **Node.js came first**, then Angular, then React.

Would you like me to prepare these same answers in **short interview format (question + 2-line answer + example)** for easy last-minute revision?