1. Define JSX

JSX (JavaScript XML) is a syntax extension for JavaScript used in React to describe what the UI should look like.
It looks similar to HTML, but it's used **within JavaScript** code and **compiled to React.createElement()** calls behind the scenes.

Example:

jsx

const element = <h1>Hello, world!</h1>;

This is syntactic sugar for:

javascript

CopyEdit

const element = React.createElement('h1', null, 'Hello, world!');

2. Explain ECMAScript (ES)

ECMAScript is the standard upon which JavaScript is based.
It defines how the language should work.

- **ES6 (ECMAScript 2015)** introduced major features like:
    - let, const
    - Arrow functions () =>
    - Classes
    - Template literals
    - Destructuring
    - Modules (import, export)
    - Promises
    - Map, Set, etc.

React heavily uses ES6+ features.

3. Explain React.createElement()

React.createElement() is a **core React API** used to create virtual DOM elements.

Syntax:

Javascript

React.createElement(type, props, ...children);

Example:

javascript

```
React.createElement('h1', { className: 'title' }, 'Welcome!');
```

This returns a React element:

javascript

```
{
  type: 'h1',
  props: { className: 'title', children: 'Welcome!' }
}
```

JSX is just a **shorthand** for this function.

4. How to Create React Nodes with JSX

You can create UI elements (called "React nodes") using JSX:

jsx

```
const element = <h2>Hello React</h2>;
```

You can also nest them:

jsx

```
const layout = (
  <div>
    <h1>Title</h1>
    <p>Description here</p>
  </div>
);
```

JSX must return **one root element**.

5. How to Render JSX to the DOM

React uses the **ReactDOM.render()** method to render JSX into the actual browser DOM.

Example:

jsx

```
import React from 'react';
import ReactDOM from 'react-dom/client';
```

```
const element = <h1>Hello React</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(element);
```

This renders the JSX inside the element with id root in index.html.

6. How to Use JavaScript Expressions in JSX

You can **embed JavaScript expressions** inside JSX using curly braces {}.

Examples:

jsx

CopyEdit

```
const name = "Vaishnavi";
const element = <h2>Hello, {name}</h2>;

const age = 20;
const message = <p>{age >= 18 ? "Adult" : "Minor"}</p>;
```

Note: Only **expressions** are allowed, not full statements like if, for, etc.

7. How to Use Inline CSS in JSX

React allows you to apply **inline styles** using the style attribute and an **object**.

Example:

jsx

```
const titleStyle = {
  color: "blue",
  fontSize: "24px",
  marginTop: "10px"
};

const element = <h1 style={titleStyle}>Styled Heading</h1>;
```

You can also use it directly:

jsx

CopyEdit

```
<p style={{ color: "green", fontWeight: "bold" }}>Welcome!</p>
```

CSS property names use **camelCase** (e.g., backgroundColor, fontSize).