1. **PropTypes**: By using props we can pass different types of information like integers, strings, arrays, etc. as props to the components. We can either create **default props** or pass props directly to the component. Passing props from outside of a component and receiving them inside another component. But we never check the data types of the props of the components. It’s totally upon us whether we validate props in the components or not.

But It’s a good practice for the large application to validate the props type of the components.

This will help to debug easily and avoid bugs in the future. We can use the prototype for validating any data we are receiving from props. But before using it we must add the dependency in our project and import it. Just like default props, **propTypes are also objects** whereas keys are the prop names and values are their types.

1. **State-Props:** Props are used for passing the data from one component to another. The state is something that passed the data within the component only. **Props are immutable and cannot be modified**, where **the state is mutable and can be updated or modified**. Props can be used within the state and functional components. Both of them hold the information.
2. **JSX**: JSX stands for **JavaScript XML**. Writing HTML code with JavaScript is known as JSX. JSX provides the syntactic sugar for the react.createElement (). JSX doesn’t return multiple parent elements. After compilation, JSX expressions become regular JavaScript function calls and evaluate JavaScript objects. In JSX JavaScript code can be written with curly braces. JSX always allows a single-line Code for JavaScript. JSX allows declaring variables with Html.
3. **Component Lifecycle:** Three phases introduce the react component life cycle. The phases are **Mounting**, **Updating**, and **Unmounting**.

**Mounting** means adding elements to the DOM. Four built-in methods of react have been called for Mounting.

These are constructor (), getDerivedStateFromProps (), render (), and componentDidMount ().

The next phase in the lifecycle is **updating**. A react component is updated when its state or props are being updated.

The last phase of react life cycle is **Unmounting**. Unmounting happens when a component is removed from the virtual DOM.

1. **Hooks**: Hooks specified the functional component to access the react components state and to allow the other features. Hooks are one kind of function that lets us hook into react state and lifecycle features. Because of hooks, there is no need to use the class components in react. Hooks generally replace the class components. Hooks are mainly used to “hook” for react features such as state and components lifecycle methods. There are a few kinds of hooks that are being introduced to the react these are useState, useEffect, useLocation, use history, useParam, etc.
2. **Custom Hooks**: The custom hook is created by the user. The custom hook is a JavaScript function that is created by the programmer or developer. To share logic among some components and other JavaScript functions custom hook is very useful. Custom hook allows sharing a piece of code to share different levels of several parts of the app
3. **Context-API**: In React sometimes the props are being sent to the 3rd,4th, and nth, components. But the intermediate component has no connection with the props. It’s called prop drilling. To solve this problem context API is introduced. Context API is a way to produce global variables effectively. This is the process of alternating prop drilling. These are also mentioned as props transferred from grandparents to children to parents and so on.
4. **Virtual-DOM and diff Algorithm**: DOM stands for Document Object Model. Dom is very much expensive. When to change one element into the DOM. It changes the whole structure of the model. So, DOM always supervises the changes. Virtual Dom makes a copy of real DOM and compares it. The Virtual Dom compares with the real Dom and identifies only the possible changes.
5. **Prop Drilling**: In React sometimes the props are being sent to the 3rd,4th, and nth, components. But the intermediate component has no connection with the props. It’s called prop drilling. To solve this problem context API is introduced. Context API is a way to produce global variables effectively. This is the process of alternating prop drilling. These are also mentioned as props transferred from grandparents to children to parents and so on.
6. **Optimize React-JS Application:** There have a few techniques to optimize the React Js Application.

* Using an immutable Data Structure revolves around a strict unidirectional data flow. Immutable data objects are simpler to create, test and use.
* Use React. Fragment to avoid additional Element wrapper: react. fragment allows a group of children without adding an extra node.