**React Interview Question:**

1. What is JSX ?

Ans: JSX stands for JavaScript XML. It is simply a syntax extension of JavaScript. It allows us to directly write HTML in React (within JavaScript code). It is easy to create a template using JSX in React, but it is not a simple template language instead it comes with the full power of JavaScript.

1. What is StateLess Component?

Ans: Stateless components are those components which don't have any state at all, which means you can't use this. setState inside these components.

1. What is Statefull Component?

Ans: In React, a stateful component is a component that holds some state.

1. ReactJS Reconciliation

Ans: The reconciliation process makes React work faster. Reconciliation is the process through which React updates the Browser DOM.

Important concepts behind the working of the Reconciliation process are:

1. Virtual DOM
2. Diffing Algorithm

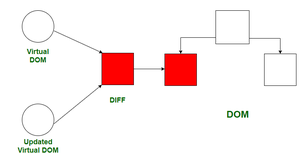
The term rendering in React can closely be identified as *making* or *becoming*. In traditional rendering, Browser does the following tasks:

* Creates a DOM (Document Object Model) represented by a tree structure.
* Renders any new data to the DOM even if data is similar to previous ones.

This rendering by Browser has a sequence of steps and is rather costly in nature. The concept of Virtual DOM used by React makes rendering much faster.

**Virtual DOM:** React renders JSX components to the Browser DOM, but keeps a copy of the actual DOM to itself. This copy is the Virtual DOM. We can think of it as the twin brother of the real or Browser DOM. The following actions take place in React:

* React stores a copy of Browser DOM which is called Virtual DOM.
* When we make changes or add data, React creates a new Virtual DOM and compares it with the previous one.
* Comparison is done by Diffing Algorithm. The cool fact is all these comparisons take place in the memory and nothing is yet changed in the Browser.
* After comparing, React goes ahead and creates a new Virtual DOM having the changes. It is to note that as many as 200,000 virtual DOM nodes can be produced in a second.
* Then it updates the Browser DOM with the least number of changes possible without rendering the entire DOM again. *(Ref: Fig.1)* This changes the efficiency of an application tremendously



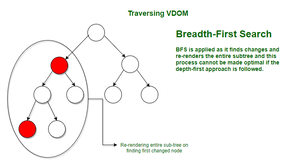
*Fig.1: Calculating Diff and updating Browser DOM*

Suppose we have new data similar to the previous one, Virtual DOM compares previous and new structures and sees that it has no change, so nothing gets rendered to the Browser. This is how the Virtual DOM is helping to enhance the Browser performance.

1. How does this Virtual DOM compare itself to its previous version?

Ans: This is where the Diffing Algorithm comes into play. Some concepts used by this Algorithm are:

1. Two elements of different types will produce different trees.
2. Breadth-First Search (BFS) is applied because if a node is found as changed, it will re-render the entire subtree hence Depth First Approach is not exactly optimal. *(Ref: Fig.2)*
3. When comparing two elements of the same type, keep the underlying node as same and only update changes in attributes or styles.
4. React uses optimizations so that a minimal difference can be calculated in O(N) efficiently using this Algorithm.



*Fig.2: BFS for traversing Virtual DOM*