**React Docs**

**DOM vs Virtual DOM:**

DOM or Document object model is a tree like representation of the html document that browser keeps the DOM represents the document as nodes and objects, that way programming languages can interact with the page.

V-DOM or Virtual DOM is a light weight representation of the original DOM which react keeps in memory and in sync with the DOM.

Now making changes directly to the DOM is very expensive and slower than making changes to the virtual DOM as virtual DOM doesn’t has the power to directly change what’s on the screen hence nothing gets drawn on screen.

When anything new is added to the application, a virtual DOM is created and it is represented as a tree. Each element in the application is a node in this tree. So, whenever there is a change in the state of any element, a new Virtual DOM tree is created. This new Virtual DOM tree is then compared with the previous Virtual DOM tree and make a note of the changes. After this, it finds the best possible ways to make these changes to the real DOM. Now only the updated elements will get rendered on the page again.

**React Fiber**

Fiber is the new reconciliation engine in React16. Its main goal is to enable incremental rendering of the virtual DOM.

The entire process of transforming changes to the real DOM is called Reconciliation.

**React Architecture:**

**Events in React:**

React supported events are:

* Clipboard events (onCopy, onCut, onPaste)
* Composition events
* Keyboard events (onKeyDown onKeyPress, onKeyUp)
* Focus events( onFocus, onBlur)
* Form events (onChange, onInput, onInvalid, onReset, onSubmit)
* Generic events (onError, onLoad)
* Mouse events ( onClick, onDoubleClick, onDrag, onMouseEnter, …)
* Pointer events (onPointerDown, onPointerMove, ….)
* Selection events (onSelect)
* Touch events (onTouchCancel, onTouchEnd, onTouchMove, onTouchStart)
* UI events (onScroll)
* Wheel events (onWheel)
* Media events (onLoad, onError)
* Animation events (onAnimationStart, onAnimationEnd, onAnimationIteration)
* Other events (onToggle)

**Stateful components:**

In React, a stateful component is a component that holds some state. Stateless component, by contrast, have no state. Note that both types of components can use props.

**Binding in react event handlers:**

**prop Types:**

**Working of react/ imp features:**

**JSX:**

JSX, is a syntax extension/ syntactic sugar to JS. We recommend using it with React to describe what the UI should look like. JSX produces React “elements”.

React doesn’t require JSX, but most people find it helpful as a visual aid when working with UI inside the JavaScript code.

**Can browsers read JSX:**

No, Browsers can’t read JSX because there is not inherent implementation for the browser engines to read and understand them. JSX is not intended to be implemented by the engines or browsers.

**Features, advantages and Limitations of React:**

**How rendering works in React:**

**States vs Props:**

The Sate represents parts of an application that can change. Each component can have its state. The state is mutable and it is local to the component only

Props are know as properties it can be used to pass data from one component to another. Props cannot be modified, read-only, and immutable.

**Use of an arrow function:**

**Higher order components(HOC’s) need/use:**

**3 phases of a components life cycle:**

**How are events created:**

**How is routing in react diff from conventional**

**routing:**

**Diff b/w flux and redux:**

**Synthetic events:**

**refs in React:**

**Purpose of render():**

**LifeCycle methods in REACT:**

**Controlled v/s Uncontrolled components:**

**Pure Components:**

**Keys?**

**Redux? Principles? Components of Redux?**

**Advantages?**

**Use of Switch Keyword?**

**React-Router?**

**Hooks?**

**Axios?**

**Instance Property vs State property:**

**Why is Router required in React**

**How can you tell react to build in prod mode:**

**Diff b/w clone elemnet & create Eelement:**

**Strict mode component:**

**Hooks**