Week1

-JSX is an XML/HTML-like syntax used by React that extends ECMAScript so that XML/HTML-like text can co-exist with JavaScript/React code.

-React is all about components.

-The createElement function is the main function in the React top-level API.

-You can use JavaScript expressions anywhere in JSX.

-JSX stands for JavaScript XML.

-You can write React components with JavaScript classes.

-All React elements attributes (events included) are named using camelCase.

-An actual JavaScript function reference is passed as the event handler, rather than a string.

-Every React component has a story.

-React components can have a/an private state.

-React get its name from the fact that it reacts to state changes.

-We can think of React as the agent we hired to communicate with the browser.

-A component might need to re-render whet its stat get updated or when its parent decides to change the props that it passed to the component.

-Lifecycle methods are actually escape hatches.

-If you’re not doing anything special, you can create full applications without lifecycle methods.

-React lets you compose complex User Interfaces (UI) from small and isolated pieces of code called components.

-When our data changes, React will efficiently update and re-render our components.

-The render method returns a description of what you want to see on the screen.

-JSX comes with the full power of JavaScript.

-Each React component is encapsulated and can operate independently.

Week2

* Props are read-only.
* To add just a few style properties, inline styling is the best option.
* In React, you usually render lists inside a component.
* A component with a/an render prop takes a function that returns a React element and calls it instead of implementing its own render logic.
* To reuse style properties in the same file, then style-components are optimal.
* A key is a special string attribute you need to include when creating lists of elements.
* All React components must act like pure functions with respect to their props.
* In React, inline styles are specified with an object whose key is the camelCased version of the style name, and whose value is the style’s value, usually a string
* The best way to pick a key is to use a string that uniquely identifies a list item among its siblings.
* A render prop is a function prop that a component uses to know what to render.
* Styled-components is a library for React and React Native that allows you to use component-level styles in your application that are written with a mixture of JavaScript and CSS.
* All you need to render a list in React is Array.map.
* You can implement most higher-order components (HOC) using a regular component with a render prop.
* CSS classes are generally better for performance than inline styles.
* It’s been said that most web apps are just lists of things.
* The term render prop refers to a technique for sharing code between React components using a prop whose value is a function.
* Inline styles are used when a single HTML element needs unique styles.
* In React, when you use lists, each list item needs a unique key.
* Any prop that is a function that a component uses to know what to render is technically a “render prop”.
* To style and element with an inline style attribute, the value must be a JavaScript object.

Week3