What Is React?

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”.

In React, you write code that looks like HTML right in your JavaScript code.

Built by Facebook and huge companies such as Uber, Twitter, Airbnb etc tried React and realized that it could help teams build better products faster.

React was release first in 2013.

Before digging into React, it’s important to know JavaScript. Not all of JavaScript, not every pattern, but having a comfort with arrays, objects, and functions before jumping into this book will be useful.

Obstacles and Roadblocks:

-**React is a library not a framework** (Does not have all the tools that is expected in a framework)

-**New ECMAScript Syntax**: As of 2015, new language features and syntax additions will be released every years. This often means that documentation assumes knowledge of the latest ECMAScript syntax. If you are not familiar with the latest spec, looking at React code can be daunting.

-**Popularity of Functional JavaScript**: React emphasizes functional programming over object-oriented programming.

-**Java Script Tooling Fatigue**: Now you need to be a webpack expert, handling code splitting, compression, testing and on and on.

-**React Doesn’t have to be Hard to learn**.

React’s Past and Future:

React was first created by Jordan Walke, a software engineer at Facebook.

It was incorporated into Facebook’s newsfeed in 2011 and later on Instagram when it was acquired by Facebook in 2012.

At JSConf 2013, React was made open source, and it joined the crowded category of UI libraries like jQuery, Angular, Dojo, Meteor, and others. At that time, React was described as “the V in MVC.” In other words, React components acted as the view layer or the user interface for your JavaScript applications.

From there, community adoption started to spread. In January 2015, Netflix announced that they were using React to power their UI development.

Later that month, React Native, a library for building mobile applications using React, was released. Facebook also released ReactVR, another tool that brought React to a broader range of rendering targets.

In 2015 and 2016, a huge number of popular tools like React Router, Redux, and Mobx came on the scene to handle tasks like routing and state management. After all, React was billed as a library: concerned with implementing a specific set of features, not providing a tool for every use case.

Another huge event on the timeline was the release of React Fiber in 2017. Fiber was a rewrite of React's rendering algorithm that was sort of magical in its execution. It was a full rewrite of React's internals that changed barely anything about the public API. It was a way of making React more modern and performant without affecting its users.

More recently in 2019, we saw the release of Hooks, a new way of adding and sharing stateful logic across components. We also saw the release of Suspense, a way to optimize

asynchronous rendering with React.

In the future, we'll inevitably see more change, but one of the reasons for React's success is the strong team that has worked on the project over the years. The team is ambitious yet cautious, pushing forward-thinking optimizations while constantly considering the impact any changes to the library will send cascading through the community.

Keeping Up with the changes:

As changes are made to React and related tools, sometimes there are breaking changes. There's a lot of React code that

was written years ago using old styles that still works well and must be maintained.

Working with the Files:

React Developer Tools: We'd highly recommend installing React Developer Tools to support your work on React projects. These tools are available as a browser extension for Chrome and Firefox and as a standalone app for use with Safari, IE, and React Native.

Once you install the dev tools, you'll be able to inspect the React component tree, view props and state details, and even view which sites are currently using React in production. These are really useful when debugging and when learning about how React is used in other projects.

To install, head over to the GitHub repository. There, you'll find links to the Chrome and Firefox extensions.

Once installed, you'll be able to see which sites are using React.

https://github.com/MoonHighway/learning-react/tree/second-edition/chapter-01

https://github.com/MoonHighway/learning-react/tree/second-edition/chapter-01

Installing Node.js	node -v
Install NPM	npm -v

React in Existing Application

- Create a new folder in your project
  “ReactSPA”
- Install following libraries
  - **React Core**
    > npm install react -save
  - **React DOM library**
    > npm install react-dom -save
  - **Babel Stand alone**
    > npm install @babel/standalone -save

Check React version from Command Prompt: npm view react version

Note:

- React core library provides properties and methods for react environment.
- React-DOM library is used to manipulate Virtual DOM.
- Babel is a compiler for React

Add a new Page into “ReactSPA” folder

Index.html

```
<!DOCTYPE html>

<html>

  <head>

    <title>React Demo</title>

    <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">

    <script src="../node_modules/react/umd/react.development.js"></script>

    <script src="../node_modules/react-dom/umd/react-dom.development.js"></script>

    <script src="../node_modules/@babel/standalone/babel.js"></script>

    <script type="text/babel">

      ReactDOM.render(
            <h2>Welcome to React.js</h2>,
            document.getElementById("container")
          )
        </script>
      </head>
      <body class="container-fluid">
        <h1 class="bg-primary text-white text-center p-2">React Application</h1>
        <div id="container">


        </div>
      </body>
</html>
```

React Components

- React is component based.
- Components are building blocks for react application.
- Component is combination of
  - Logic
  - Presentation and
  - Styles
- **Logic** is defined using JavaScript
- **Presentation** is defined by using HTML
- **Styles** are defined by using CSS
- In react components can be designed by using
  - JavaScript functions
  - JavaScript classes
- React components are classified into 2 types
  - Function Components [JavaScript Functions]
  - Class Components [JavaScript Classes]

Ex: Function Component

```
<!DOCTYPE html>
<html>
   <head>
      <title>React Demo</title>
      <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
      <script src="../node_modules/react/umd/react.development.js"></script>
      <script src="../node_modules/react-dom/umd/react-dom.development.js"></script>
      <script src="../node_modules/@babel/standalone/babel.js"></script>
      <script type="text/babel">
         function HeaderComponent(){
            const title = "Amazon Shopping";
            return(
               <div className="text-center bg-danger text-white p-1 mt-2">
                  <h2>{title}</h2>
                  <p>Online Shopping - 60% OFF on Electronics</p>
                </div>
            )
         }
         ReactDOM.render(
            <HeaderComponent />,
            document.getElementById("container")
         )
      </script>
   </head>
   <body class="container-fluid">
```

```
        <div id="container">


        </div>
    </body>
</html>
```

Ex: Function Component as Arrow function

```
<!DOCTYPE html>
<html>
    <head>
        <title>React Demo</title>
        <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
        <script src="../node_modules/react/umd/react.development.js"></script>
        <script src="../node_modules/react-dom/umd/react-dom.development.js"></script>
        <script src="../node_modules/@babel/standalone/babel.js"></script>
        <script type="text/babel">
            const HeaderComponent = () => {
                const title = "Amazon Shopping";
                const offer = "Online Shoppine - 70% OFF on Footwear";
                return(
                    <div className="text-center text-white bg-danger mt-2 p-2">
                    <h2>{title}</h2>
                    <p>{offer}</p>
                    </div>
                )
            }
            ReactDOM.render(
                <HeaderComponent />,
                document.getElementById("container")
            )
        </script>
    </head>
    <body class="container-fluid">
        <div id="container">

        </div>
    </body>
</html>
```

Ex: **Multiple Components**

```
<!DOCTYPE html>
<html>
    <head>
        <title>React Demo</title>
        <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
        <script src="../node_modules/react/umd/react.development.js"></script>
        <script src="../node_modules/react-dom/umd/react-dom.development.js"></script>
        <script src="../node_modules/@babel/standalone/babel.js"></script>
        <script type="text/babel">
            const HeaderComponent = () => {
                const title = "Amazon Shopping";
```

```
   const offer = "Online Shoppine - 70% OFF on Footwear";
   return(
      <div className="text-center text-white bg-danger mt-2 p-2">
      <h2>{title}</h2>
      <p>{offer}</p>
      </div>
   )
}
const FooterComponent = () => (
   <div className="text-center bg-danger mt-2 text-white">
      &copy; copyright 2021 - Amazon India
   </div>
)
const LoginComponent = () => (
   <div>
    <dl>
       <dt>User Name</dt>
       <dd><input type="text" /></dd>
       <dt>Password </dt>
       <dd><input type="password" /></dd>
    </dl>
    <button>Login</button>
   </div>
)
const MainComponent = () => (
   <div>
```

```
      <h1>User Login</h1>
      <LoginComponent />
   </div>
)
const SectionComponent = () => (
   <div style={{height:'400px'}}>
     <div className="row">
       <div className="col-2">
         <ul>
            <li>Home</li>
            <li>Electronics</li>
            <li>Footwear</li>
         </ul>
       </div>
       <div className="col-10">
          <MainComponent />
       </div>
     </div>
   </div>
)
ReactDOM.render(
    <div>
     <HeaderComponent />
     <SectionComponent />
     <FooterComponent />
    </div>,
            document.getElementById("container")
         )
      </script>
   </head>
   <body class="container-fluid">
      <div id="container">

      </div>
   </body>
</html>
```