the shortcut

React Tutorial
Lesson 1

Today’s lesson will cover

- Intro to React
- Setting up development environment and project
- Intro to React component
- JSX
- Lists in React
- Components basics
- React DOM

the shortcut

Intro to React

What is React in a nutshell

- ReactJS is an open-source, component based front end library responsible only for the **view** layer of the application
- ReactJS uses virtual DOM based mechanism to fill in data (views) in HTML DOM.
- A React application is made up of multiple components
- Components can be nested within other components to allow complex applications to be built out of simple building blocks.
- A component may also maintain internal state
- React allows us to write components using a domain-specific language called JSX
- React "reacts" to state changes in your components quickly and automatically to rerender the components in the HTML DOM by utilizing the virtual DOM

Development environment requirements

- IDE or text editor of your choice
- Node and NPM installed

Setting up a React Project

- We will use create-react-app to bootstrap our projects
- Everything you need is located in the src/ folder
- The main focus lies on the src/App.js file. It will be used to implement application, but later you might want to split up your components into multiple files
- You will also find a src/App.test.js file for your tests, and a src/index.js as an entry point to the React world. There is also a src/index.css and a src/App.css file to style your general application and components, which comes with the default style when you open them. You will modify them later as well.

React Component

- First React component called App component is just a JavaScript function. It's commonly called **function component**, because there are other variations of React components (we will talk about **component types** later)
- Second, the App component doesn't receive any parameters in its function signature yet (we will talk about **props** later)
- Third, the App component returns code that resembles HTML which is called JSX
- The function component possess implementation details like any other JavaScript function. You will see this in practice in action throughout your React journey
- Variables defined in the function's body will be re-defined each time this function runs, like all JavaScript functions

JSX

- Output of the App component that resembles HTML is called JSX (JavaScript XML), which mixes HTML and JavaScript
- JSX replaces a handful of internal HTML attributes, but you can find all the supported HTML attributes in React's documentation, which follow the camel case naming convention. Expect to come across more JSX-specific attributes like `className` and `onClick` instead of `class` and `onclick`, as you learn more about React
- JSX was initially invented for React, but it became useful for other modern libraries and frameworks after it gained popularity. It is one of the nice things about React that without any extra templating syntax (except for the curly braces), we are now able to use JavaScript in HTML.

Lists in React

- Since we use JSX in React, you can use the built-in JavaScript map method for arrays to iterate over each item of the list and return a new version of each
- Avoid using the index of the item in the array to make sure the key attribute is a stable identifier. If the list changes its order, for example, React will not be able to identify the items properly
- Not only can JavaScript in JSX be used to display items, but also to assign HTML attributes dynamically
- Read more about why React's key attribute is needed, Don't worry if you don't understand the implementation yet, just focus on what problem it causes for dynamic lists.:
    a. https://dev.to/jtonzing/the-significance-of-react-keys---a-visual-explanation--56l7
    b. https://reactjs.org/docs/lists-and-keys.html
- Recap the standard built-in array methods -- especially *map*, *filter*, and *reduce* -- which are available in native JavaScript.

Standalone components

- Larger React applications have **component hierarchies** (also called **component trees**)
- There is usually one uppermost **entry point component** (e.g. App) that spans a tree of components below it
- The App is the **parent component** and can have **child component** of the App
- In a component tree, the App is the **root component**, and the components that don't render any other components are called **leaf components** (e.g. Item)
- If the App has another child component, the additional child component on same heirarchy is called a **sibling component**

Component instantiation

- If a JavaScript class definition exists, one can create multiple instances of it. It is similar to a React component, which has only one component definition, but can have multiple component instances
- Once we've defined a **component**, we can use it like an HTML **element** anywhere in our JSX. The element produces an **component instance** of your component, or in other words, the component gets instantiated.
- You can create as many component instances as you want. It's not much different from a JavaScript class definition and usage.

React DOM

- Next to React, there is another imported library called `react-dom`, in which a `ReactDOM.render()` function uses an HTML node to replace it with JSX. The process integrates React into HTML. `ReactDOM.render()` expects two arguments; the first is to render the JSX. It creates an instance of your App component, though it can also pass simple JSX without any component instantiation.

- The second argument specifies where the React application enters your HTML. It expects an element with an `id='root'`, found in the *public/index.html* file. This is a basic HTML file.

React Component Definition

- All components in the src/App.js file are function component. JavaScript has multiple ways to declare functions. So far, we have used the function statement, though arrow functions can be used more concisely
- If an arrow function doesn't do *anything* in between, but only returns *something*, -- in other words, if an arrow function doesn't perform any task, but only returns information --, you can remove the **block body** (curly braces) of the function. In a **concise body**, an **implicit return statement** is attached, so you can remove the return statement

Handler function in JSX

- For the change event of the input field in React, there are functions called **(event) handler** which can be passed to the `onChange` attribute (JSX named attribute) of the input field.
- HTML and JavaScript work well together in JSX. JavaScript in HTML can display objects, can pass JavaScript primitives to HTML attributes (e.g. `href` to `<a>`), and can pass functions to an element's attributes for handling events.

Self Study Assignments

To Dos

- Start working on React or Fullstack JavaScript project, OR complete FCC Frontend Library Certification : Bootstrap, React, Redux, React & Redux, projects; APIs and Microservices Certification : https://www.freecodecamp.org/learn