Learning React - Tutorial 01(Getting Started)

Hello World with Reactjs.

Useful link for deeper learning https://reactjs.org/tutorial/tutorial.html

Description:

This first tutorial is meant to familiarize you with Reactjs which is a Javascript library that helps you make powerful responsive web apps. This shall guide you through installation and creating your first simple web app.

Build Workflow: (I recommend this workflow or stack). To optimize code, be able to use next gen Javascript features and in general make things easier as a beginner)

- Use dependency management tool (npm or yarn)
- Use a bundler. (WebPack)
- Use next gen Javascript Compiler (Babel along side presets)
- Use a Development server

To do all of this we are going to make use of create-react-app as highlighted here.
https://github.com/facebook/create-react-app

Installation Instructions:

Download latest version nodejs compatible with your machine from nodejs.org and install in order to use node package manager.

To confirm the install open a command terminal and execute
>node -v

You should get the version number as a response To confirm npm (node package manager) is installed execute
>npm -v

Once you do this go into your systems terminal or command prompt and run this command.

> npm install create-react-app -g

Run with sudo in front of it if your requires permissions.

This installs it globally so you can run from anywhere.

Instructions:

Now that you have create react app installed. Create a new app and name it tutorial1 with the following command. This may take a while to setup.

>create-react-app tutorial1

Move into new directory

>cd tutorial1

Run the following command to start your server.

>npm start

Your new web app should come up on localhost:3000. This also comes with hot reload so you do not need to manually reload webpage to view changes.

I also recommend an ide as there are a lot of folders and files and it becomes hard to manage it all. Download visual studio code https://code.visualstudio.com/download.

Overview:

Components are the foundation of react apps. In fact a react app can generally be represented as a component tree, which has a parent component or container usually named “App.js” and then a number of nested children components. A component is required to return JSX. This tells react which html to render however JSX is not html. It is just a way of using javascript to write html in an easier way instead of separating stuff.

A typical React app therefore could be depicted as a **component tree** - having one root component ("App") and then an potentially infinite amount of nested child components.

Each component needs to return/ render some **JSX** code - it defines which HTML code React should render to the real DOM in the end.

There are two types of components:

Functional Components: These are just usually for representing real world objects the do not have state(will be explored in later tutorials)

Class-based Components: These are often used as containers can manipulate state

Problem 1)

Create a new functional component called Student and return a single paragraph inside it.

Problem 2)

Import it into your app class and have it display this student.