**Code With Harry Tutorials**

1. npx create-react-app app-name
2. Play with Files in of src and public Folder, to get handy.
3. In src folder index.js is entry point and you can note <App /> there. In the beginning of the code, you can see import statement also that imports app component.

* Also note that in App.js at the end it’s exporting that component.

1. Tutorial 3 : JS Refresher.

* Var, let, const
* Variable objects
* Object functions
* Event-listeners, alerts, confirm
* String methods in JavaScript and substring
* Arrays
* Dates, for and while loops, for each loop
* Arrow function
* New: when we write “use strict” in JavaScript that means now JavaScript will be execute in strict mod. Just for good code. Always on this thing.
* This keyword.
* Callback function like set timeout. Dur to asynchronized behavior
* Promises New :::: Doing something and as soon as I will complete this I will do this. If resolved or rejected.

1. Tutorial 4: JSX (JavaScript xml)

https://reactjs.org/docs/introducing-jsx.html

* In a function when you are returning it, it must be in a single tag
* Invalid :
  + <h1>Hello</h1>
  + <div>aaa</div>
* Valid :
  + <>
  + <h1>Hello</h1>
  + <div>aaa</div>

</>

Here if we really need other tags then we can wrap them in empty tags like above.

* className instead of class
* htmlFor instead of for
* tabIndex

note that : if you want to open this in your phone, you can do this just give this Ip address in phone.

Embedding Expressions in JSX

const name = 'Josh Perez';

const element = <h1>Hello, {name}</h1>;

Note that always close the tag at the end. It’s compulsory in JSX.

Tutorial 5: // Adding Bootstrap and Textutils project

Note: there is node modules folder that contains all the packages and libraries we need. We can delete it because we already have that all details in package.json … we can install all the dependency using following command

* npm install / npm i

Adding Bootstrap:

We can add CDN of bootstrap in public/index.html.

Creating Textutils project.

Tutorial 7: Understanding Props and PropTypes

JavaScript default import and export default names.

Suppose we create module1.js, then to run it we can write:

* node .\module1.js

Keep in mind that it says any error then ,we can add ‘m’ before js extension like following:

(module1.mjs)

We can create component in another folder inside src folder and use them in our Apps.js

We can pass props in function-based component like parameter in function. Now we can pass the arguments from the Apps.js by adding them like following:

<Navbar title="TextUtils" />

Here title is argument pass in props. We can use if by {props.title}

We can make proptype fix and also make it require by importing and using props-type.

import PropTypes from 'prop-types';

Navbar.prototype = {

    title: PropTypes.string.isRequired

}

I have created module1.mjs and module2.mjs to understand the concept of named and default import and export of names.

Here note that if we use .js extension then we will get the following error

PS C:\Users\Dell\OneDrive\Desktop\React-with-Git> node .\react-learning\src\components\module1.js

(node:10376) Warning: To load an ES module, set "type": "module" in the package.json or use the .mjs extension.

(Use `node --trace-warnings ...` to show where the warning was created)

C:\Users\Dell\OneDrive\Desktop\React-with-Git\react-learning\src\components\module1.js:1

import aa from './module2.js';

^^^^^^

SyntaxError: Cannot use import statement outside a module

at Object.compileFunction (node:vm:352:18)

at wrapSafe (node:internal/modules/cjs/loader:1027:15)

at Module.\_compile (node:internal/modules/cjs/loader:1063:27)

at Object.Module.\_extensions..js (node:internal/modules/cjs/loader:1153:10)

at Module.load (node:internal/modules/cjs/loader:975:32)

at Function.Module.\_load (node:internal/modules/cjs/loader:822:12)

at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run\_main:77:12)

at node:internal/main/run\_main\_module:17:47

To solve this, we use .mjs extension instead of .js

Note that we can not have multiple defaults for export statements. **One more thing to be noted is props are Read only, we cannot change them.**

Default Props:

Navbar.defaultProps = {

    title: "I am Good Title"

}

**React Hooks: React Hooks are that something that let us use the react class component-based features in function-based components.**

**State:** this is class-based feature, to use this in function-based component we need hook which is useState.

Normally, variables “disappear” when the function exits but state variables are preserved by React.

You will note that if you add onclick event on button and changing the text value that is contained inside textarea that will theow an error in console that you should add onchange event in textarea.

State is One type of variable that can be changed using setvar like of.

Handling Event. We can watch the commit online.

Noted Point: I have noted that in react also if button type is submit then the page will surly realoded. If you want that your page should not reload then remove type=”submit” form the button tag.

I have learned to make the text area writable. If we are using the value form another variable.

**Tut 8 : Add your own functionality (I am adding copy button)**

I have desided to to add jquery in my react app to add copy functionality.

To add the Jquery, Following Steps only

* npm install jquery –save
* 2. Now open project in your favorite IDE and import jquery to the file where you want to manipulate DOM.
* import $ from 'jquery'

Note that we can not change the props so we use state instead. The other thing is when we are using object of object then we should use the {{ }} “double curly braces”.

**Tutorial 12:**

* we can use “ ` ”, this is backtics under Esc, for templating. We can not use template strings inside string. To use templating, we have to use backtics.
* We can set style by using {{ }}, That means object of object. Ex:
  + Style = {{backgroundColor: props.mode === ‘dark’ ? ‘grey’:’white’, color: props.mode === ‘dark’ ? ‘white’:’black’}}

**===============================================**

**Updating React component:**

* **npm install react@18 react-dom@18**

**Note that in react 18 we can not use ReactDom.render. (in index.html)**

**We can use as following:**

**const root = ReactDom.createRoot(document.getelementId(‘root));**

**root.render(**

**<App />; //component.**

**)**

**React Router:**

React router used to change the particular component without reloading the page. To install the router:

* npm install react-router-dom

In react Router we have to import it.

import {

  BrowserRouter as Router,

  Routes,

  Route,

  Link

} from "react-router-dom";

<Router>

      <Navbar title="TextUtils" mode = {mode} toogleMode={tooglemode} switchMode={switchMode}/>

      <Alert alert={alert}/>

      {/\* I am trying to use Router For better user experience.  \*/}

      <Routes>

            {/\* <Navbar title="TextUtils" mode = {mode} toogleMode={tooglemode} switchMode={switchMode}/> \*/}

            <Route exact path='/' element={<Form heading="Enter the Text to Analyze" showAlert={showAlert}/>} />

            <Route exact path='/about' element={<About />} />

      </Routes>

    </Router>

Note in above code that Everything under Router, and we can use Routes tag to set paths n all like above

* Note “exact” keyword also here. This is because react do partial checking for url if we do not use exact keyword.
* For example there is : /users, and /users/home this two urls. And now what react will do is if exact is not defined then for users/home it will go for /users inseted, by doing partial checking. So it’s important to use exact keyword here.
* Now we have to use Link and to instead of a and href.
* For suppose in Navbar there is about to set it,
* <Link className="nav-link" to="/about">About</Link>

Like this.

We should know that if we are using link and to then the page will not reloding. We can use a, and href also but the page will reload dur to this, if want reloding then we can use a and href.

**Tutorial: 20**

**Note that OnClick Function take the arrow function not function calling.**

If we Give function calling then the onClick will be automatically called infinite loop.

  {/\* This will not work, it will go in infinit loop of a functon, Because if you want to pass the parameter to any function on click then you need to make it arrow fuction First.  \*/}

                {/\* OnClick attribute takes funciton as a parameter only. \*/}

                {/\* <div type="button" className="bg-primary border rounded mx-2" onClick={props.toogleMod} style={{height: '30px', width:'30px'}}></div>  Here on click is Funciton Call \*/}

                {/\* Below we will make the funciton call as a function for onclick  \*/}

                <div type="button" className="bg-primary border rounded mx-2" onClick={() => customTheme('primary')} style={{height: '30px', width:'30px'}}></div>  {/\*Here on click is Funciton Call\*/}

The above is code example.

**Project2**