channels for react :

* Nerd's lesson
* programming with mosh(old video)
* edureka
* simplily learn
* pedro tech (spl for projects)
* clever programmer
* scrimba – free code camp

create react app in VS studio terminal:

1. You can now create a new React application by typing:

npx create-react-app my-app

2. Let's quickly run our React application by navigating to the new folder and typing npm start to start the web server and open the application in a browser:

cd my-app

npm start

You should see the React logo and a link to "Learn React" on http://localhost:3000 in your browser. We'll leave the web server running while we look at the application with VS Code.

\*React is an open source js library for creating user interface.

\*component based architecture - combination of small component

-react js and react native.

--->web app- react js .

--->mobile app- react native

React two components: funtional and class

1. fun:

function welcome(){

return <h1>Heloo</h1>

};

2. Class:

class welcome extends React.Component

{

render(){

return <h1>Heloo</h1>

};

};

* **COMPONENT name must start with uppercase.**

(Because react will treat smaller case regular html tag. so have to use uppercase for react component)

React Structure:

Public folder:

--> index .html - is one of the main which is having html code (root id <div> is here )

-->logo files are here.

-->manifest file- this file for concern for web app.

Src folder:

-->index.js - this file is starting point for react js. Here we are getting access to the root file. (reactDOM is here). All the import files are here.

App.js folder:

--> This is used in index .js file.

-->index.html-- contains root element (<div class="root"> )

-->logo.svg is logo file.

--> Report file is here . We won’t use it .

--> Setup file is also here. Which will use in testing.

Hooks

components ----> it is contain every single components in your entire application.

\_tests\_

assets -----> it contains all CSS and images, font files.

React component- core building block of react.

-react components

class component --> it is having render() method which will return something

syntax:

import {component} form react;

class message extends component{

render(){

return <h1> this is class component </h1>

}

}

export default message;

->> state

->> render

-react components always start with caps(uppercase)

-event handlings

-Jsx = html+css+js code

class component-- render(), inside render you always have return

HOW IT’S work??

-Node js- to run the react codes. Just like an application

-to start a project

1. Install node js

2. Create create app

npm install -g create-react-app

3. Start a project

create -react-app app-name

* ctrl+c for stop react -to terminate

- In editor

Index.js file - main file for display content in browser.

1. Import library

2. Create component it can be either class or function

Example fun syntax:

const FunName =() =>{

var message="welcome "

return ( <div> {message} </div>)

} OR

const FunName =() =>{

return ( <div> <h1>"welcome"</h1> </div>)

}

3. return should be in the class or fun

4. data display--> Reactdom. render( )

for project purpose react splitting:

step 1. index.js file ---> import library and Reactdom .render

step 2. App.js ---> for writing functions and classes, and then import this file into the index.js file. import App from "./app";

step 3. App.css file for creating styles for app.js

steps to remember:

--> 1. import

---> 2. data App- functions and classes

---> 3. display - render

for reuse code:

step 1: create separate js file

step2: inside function

--> copy the code which you want to reuse

step3: export the function (export default funname)

step4: inside js file ,inside return you can write function name how many times you want. this will look like a nesting concept .

# using props you can change name or anything in react js

what is props??

it is useful for passing properties and attributes.

composition--> Component composition is the name for passing components as props to other components, thus creating new components with other components

JSON File in react js:

* In React, a JSON file can be used to store and manage data in a structured way. JSON, which stands for JavaScript Object Notation, is a lightweight data format that is easy to read and write. JSON files can be used to store data such as configuration settings, user preferences, or content for a web application.
* In a React application, a JSON file can be imported into a component and then used to render content on the page.
* When working with a JSON file in React, you can use built-in JavaScript functions such as fetch() to load the data from the file and then process it as needed in your application. You can also use third-party libraries such as axios or jQuery to simplify the process of making requests and handling responses from the server.
* Json data types- strings, numbers, Boolean, null, arrays- [1,2,3], objects-{“key” : “value”}.
* Object syntax - {

“name” : “venba”,

“age” : 26,

“interest” :”cooking”

}

1. Create json file inside the src folder with the following extension of .json (for example records.json)
2. Store the data in json file .

Inside the square brackets we need to crate objects in json. [ ]

1. Each object can have properties and values. Json object can contains strings, numbers, arrays or anything.
2. Json object stucture :

[

{

“id” : 1,

“name” : “venba”,

“icon” : “link or path “,

“color” : “green”,

“details” :[

{

“width” : “23px”,

“padding”: “10px”

}

]

},

{

-------

-------

}

]

1. Once created json file we need to import that json file in app.js or react app.

Import Records from “./records.json”;

Links:

https://egghead.io/courses/the-beginner-s-guide-to-react

to create progress bar:

https://www.npmjs.com/package/react-circular-progressbar

my first react responsive page:

<https://codesandbox.io/s/stoic-scott-nyo22l?file=/src/cards.js>

<https://www.interviewbit.com/blog/react-projects/> -- for projects excerise.