React Concept

1->Don’t touch the DOM, (declarative)->one big object that act as component, you change it and I will change the dom. React-> this is the state make the change

2-> small component=> plain java script function()=> except prop and has html like structure and then render. It will have state

Component can be function or class component

3 -> one way data flow (state+ jsx)=> give component render in virtual dom

React developer tools

Node -v

npm -v

yarn -v

Install dependencies from package.json: npm install == yarn

Install a package and add to package.json: npm install package --save == yarn add package

Install a devDependency to package.json: npm install package --save-dev == yarn add package --dev

Remove a dependency from package.json: npm uninstall package --save == yarn remove package

Upgrade a package to its latest version: npm update --save == yarn upgrade

Install a package globally: npm install package -g == yarn global add package

npx create-react-app

cd

code .

npm start

json file=>

* react
* react dom
* script
  + start
  + build-> production build (build folder) => javascript
  + test
  + eject-> manual and granular control

babel-> code to basic javascript

web pack=> chunk (break it into chunck)

public folder ->index.html(root)->src

public folder has other static stuff

src folder->index.js->(react;react-dom)->App-> component

index.js=> reactdom.render()

classes are java script

hooks are react specific . hooks are use with function

Set state=> we are doing object. Assign() so it is create new object in memory

Set state do=> Shallow merge , will I will update the state value of the key,

In case of shallow merge it only merge the key that is asked for even though it is a complex.

State

Set State

Map

Map with id : Map optimised by using id