## REACT Session Day-1

Design Patterns:

MVC

Factory

Observer

Dependency Injection

Development Environment – MERN – Mongo, Express, React & Node MEAN – A is Angular

Node.js

Visual Studio code.

JavaScript Framework

Ajax based framework

Single page Application

Component based UI Application

Ajax – DOM Manipulation, Part of page is rendered, HTML page is loaded only once.

No more multiple pages development – Move in to Component based application

NPM [node package Manager ]manages the dependencies for JavaScript’s. – Update & uninstall the dependencies.

NPM is a CLI based tool.

Check NPM version

npm –v

Dependencies installed in Global Mode or Local Mode

npm –install uninstall or update

Options –S or –D

list , init, start command

For any Project there needs to be the configuration file to take care of Project related [Meta data for Projects]

Package.json starts with { and ends with } that will be used by npm.

Can be created by developer or created by npm using npm init.

Npm –init –y (Override y the creation of config file )

D:\ReactTraining\Day1Project>npm init -y

Wrote to D:\ReactTraining\Day1Project\package.json:

{

"name": "Day1Project",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

"license": "ISC"

}

CSS – Add the Head section

In the style

Link to external file

Selectors

Bootstrap - npm install bootstrap –S

[Saves to the configuration file in the package.json

D:\ReactTraining\Day1Project>npm install bootstrap -S

npm notice created a lockfile as package-lock.json. You should commit this file.

npm WARN bootstrap@4.1.1 requires a peer of jquery@1.9.1 - 3 but none is install

ed. You must install peer dependencies yourself.

npm WARN bootstrap@4.1.1 requires a peer of popper.js@^1.14.3 but none is instal

led. You must install peer dependencies yourself.

npm WARN Day1Project@1.0.0 No description

npm WARN Day1Project@1.0.0 No repository field.

+ bootstrap@4.1.1

added 1 package from 2 contributors and audited 1 package in 22.92s

found 0 vulnerabilities]

Now ackage.json is updated with Bootstrap

{

"name": "Day1Project",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

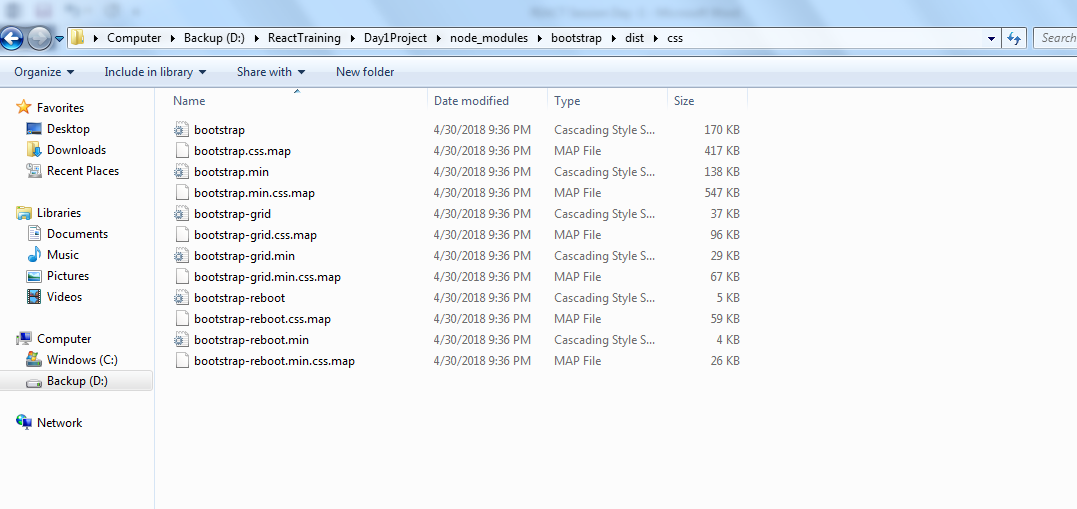
"license": "ISC",

"dependencies": {

"bootstrap": "^4.1.1"

}

}



D:\ReactTraining\Day1Project>npm uninstall bootstrap -S

npm WARN Day1Project@1.0.0 No description

npm WARN Day1Project@1.0.0 No repository field.

removed 1 package in 1.652s

found 0 vulnerabilities

D:\ReactTraining\Day1Project>npm install bootstrap@3.3.7 -S

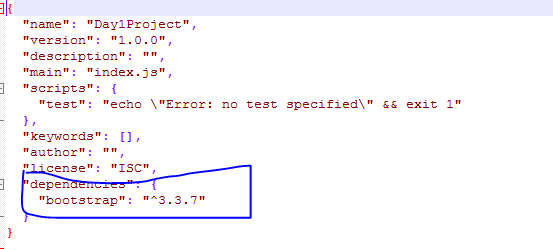
npm WARN Day1Project@1.0.0 No description

npm WARN Day1Project@1.0.0 No repository field.

+ bootstrap@3.3.7

added 1 package from 1 contributor and audited 1 package in 1.767s

found 0 vulnerabilities



{

"name": "Day1Project",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

"license": "ISC",

"dependencies": {

"bootstrap": "^3.3.7"

}

}

# LITE SERVER

Install the lite- server for Development purposes.

npm i lite-server –D

D:\ReactTraining\Day1Project>

D:\ReactTraining\Day1Project>npm i lite-server -D

D:\ReactTraining\Day1Project>npm i lite-server -D

npm WARN deprecated uws@9.14.0: stop using this version

> uws@9.14.0 install D:\ReactTraining\Day1Project\node\_modules\uws

> node-gyp rebuild > build\_log.txt 2>&1 || exit 0

npm WARN Day1Project@1.0.0 No description

npm WARN Day1Project@1.0.0 No repository field.

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node\_modules\fse

vents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@

1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}

)

+ lite-server@2.3.0

added 232 packages from 198 contributors and audited 644 packages in 403.534s

found 5 low severity vulnerabilities

run `npm audit fix` to fix them, or `npm audit` for details

D:\ReactTraining\Day1Project>

# After Installing start the webserver from .bin and execute the server

D:\ReactTraining\Day1Project>node\_modules\.bin\lite-server

Did not detect a `bs-config.json` or `bs-config.js` override file. Using lite-se

rver defaults...

\*\* browser-sync config \*\*

{ injectChanges: false,

files: [ './\*\*/\*.{html,htm,css,js}' ],

watchOptions: { ignored: 'node\_modules' },

server: { baseDir: './', middleware: [ [Function], [Function] ] } }

[Browsersync] Access URLs:

--------------------------------------

Local: http://localhost:3000

External: http://192.168.2.205:3000

--------------------------------------

UI: http://localhost:3001

UI External: http://192.168.2.205:3001

--------------------------------------

[Browsersync] Serving files from: ./

[Browsersync] Watching files...

18.06.04 11:50:41 200 GET /index.html

18.06.04 11:50:41 200 GET /node\_modules/bootstrap/dist/css/bootstrap.css

18.06.04 11:50:42 404 GET /favicon.ico

Package.json looks like after installing the lite-server entry

{

"name": "Day1Project",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

"license": "ISC",

"dependencies": {

"bootstrap": "^3.3.7"

},

"devDependencies": {

"lite-server": "^2.3.0"

}

}

# NPM as a Task Runner

Common web development tasks

1. Built in task [start, test] 🡪 Scripts
2. Custom task

Mentioned as scripts in package .json

"scripts": {

"test1": "echo No Test Specified",

"lite": "lite-server"

},

For Loop: **var** is always a global variable

for (var i=0;i<=5;i++){

console.log(i)

}

console.log(i)

ECMA SCRIPT – responsible for standards in Javascript earlier for loop – ES5

ES6 – Blocked code

Const keyword and let keyword added

Arrow functions

for (var i=0;i<=5;i++){

console.log(i) }

console.log(i)

for (let j=0;j<=5;j++){

console.log(j)

}

console.log(j)

Assignment to COntsnat variable Throws error

for (var i=0;i<=5;i++){

console.log(i)

}

console.log(i)

for (let j=0;j<=5;j++){

console.log(j)

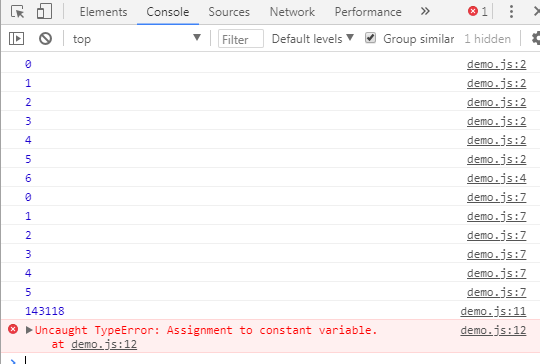
}

//console.log(j)

const g =143118;

console.log(g)

console.log(g++)



for (var i=0;i<=5;i++){

console.log(i)

}

console.log(i)

for (let j=0;j<=5;j++){

console.log(j)

}

//console.log(j)

const g =143118;

console.log(g)

//console.log(g++)

//ANONYMOUS FUNCTION

let M= function(){

return "WellsFargo"

}

console.log(M)

console.log(M())

//ARROW FUNCTION

N=() => "HYDERABAD"

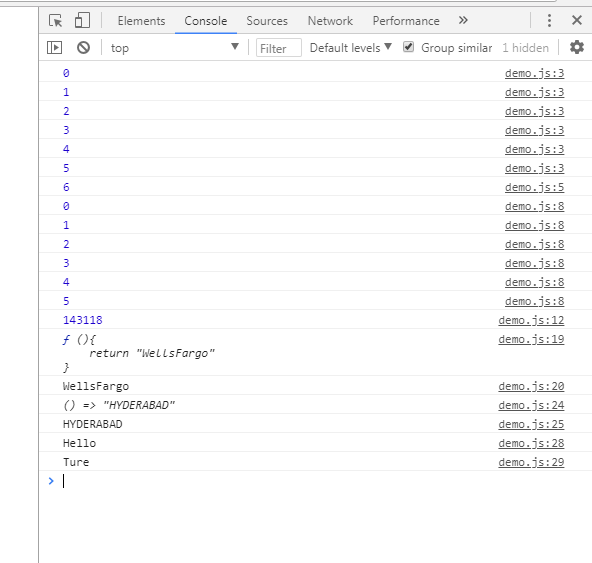
console.log(N)

console.log(N())

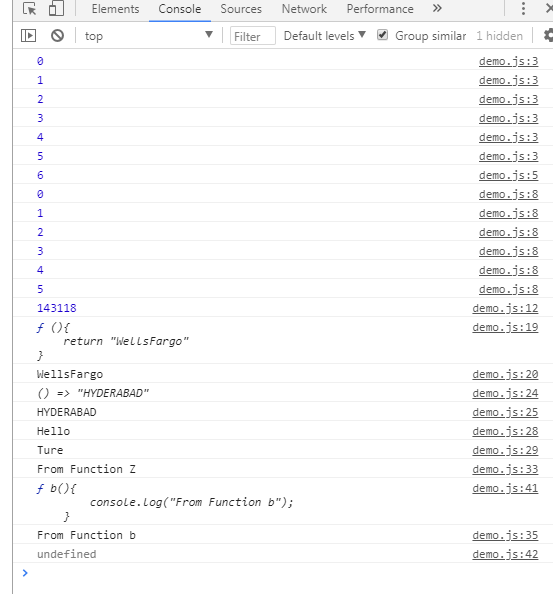
x=(msg) => msg // argument to Function and returning from Function

console.log(x("Hello"));

console.log(x("Ture"));



**ANONYMOUS AND ARROW FUNCTIONS**



//ANONYMOUS FUNCTION

let M= function(){

return "WellsFargo"

}

console.log(M)

console.log(M())

//ARROW FUNCTION

N=() => "HYDERABAD"

console.log(N)

console.log(N())

x=(msg) => msg // argument to Function and returning from Function

console.log(x("Hello"));

console.log(x("Ture"));

//Function returning a function

function Z() {

console.log("From Function Z");

function b(){

console.log("From Function b");

}

return b;

}

let s=Z();

console.log(s);

console.log(s());

INSTALL REACT

D:\ReactTraining\Day1Project>npm install create-react-app -g

C:\Users\admin\AppData\Roaming\npm\create-react-app -> C:\Users\admin\AppData\Ro

aming\npm\node\_modules\create-react-app\index.js

+ create-react-app@1.5.2

updated 1 package in 26.229s

D:\ReactTraining>create-react-app reactproject1

Creating a new React app in D:\ReactTraining\reactproject1.

Installing packages. This might take a couple of minutes.

Installing react, react-dom, and react-scripts...

> uglifyjs-webpack-plugin@0.4.6 postinstall D:\ReactTraining\reactproject1\node\_

modules\uglifyjs-webpack-plugin

> node lib/post\_install.js

+ react-dom@16.4.0

+ react-scripts@1.1.4

+ react@16.4.0

added 1314 packages from 810 contributors and audited 14349 packages in 284.826s

found 0 vulnerabilities

Success! Created reactproject1 at D:\ReactTraining\reactproject1

Inside that directory, you can run several commands:

npm start

Starts the development server.

npm run build

Bundles the app into static files for production.

npm test

Starts the test runner.

npm run eject

Removes this tool and copies build dependencies, configuration files

and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

cd reactproject1

npm start

Happy hacking!

D:\ReactTraining>

CLASSES

M2.js – Export class EMP🡪 available for any other js file

Use IMPORT where u need the class to be Instantiated

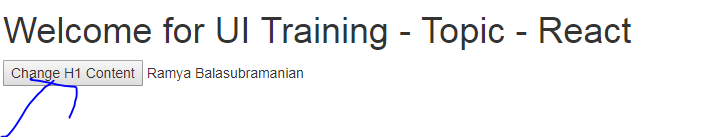
MODULE – Concept works with Export & Import

# HANDS ON DOM MANIPULATION

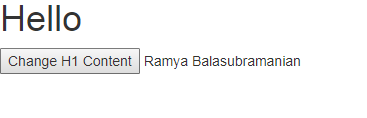
Use Myname Tag

* Built in tags & Attributes
* Custom Tags & Custom Attributes <myname> Tage
* function change(){
* let value=document.getElementById("h");
* let replyvalue=value.getAttribute("reply");
* value.innerHTML=replyvalue;
* }

DOM Manipulated



Changed with Change function



<https://reactjs.org/> - Overcome DOM Manipulation

React has 2 Libraries - React & React DOM

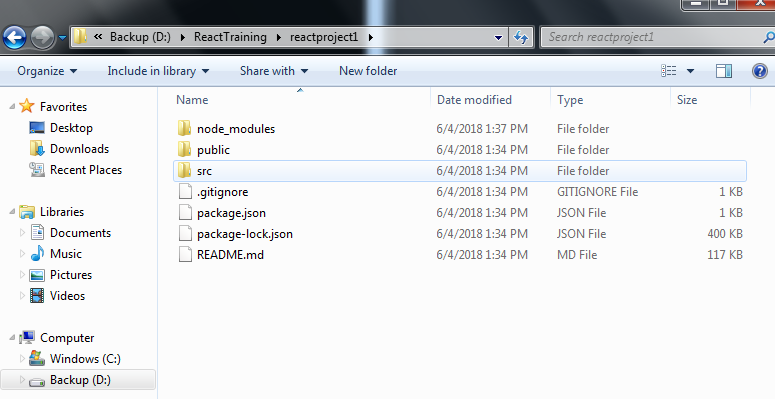
Works on the principal of VIRTAL DOM- Check with IN Memory Object – To Update the DOM

REACT JS PROJECT SET UP

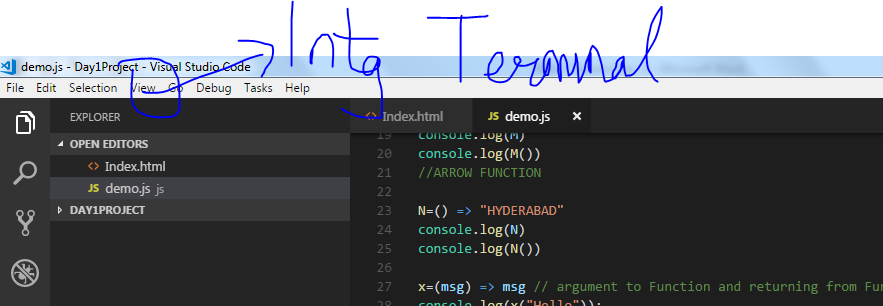
1. Manual Set up
2. Create-React-app [CLI] – sets up with zero configuration

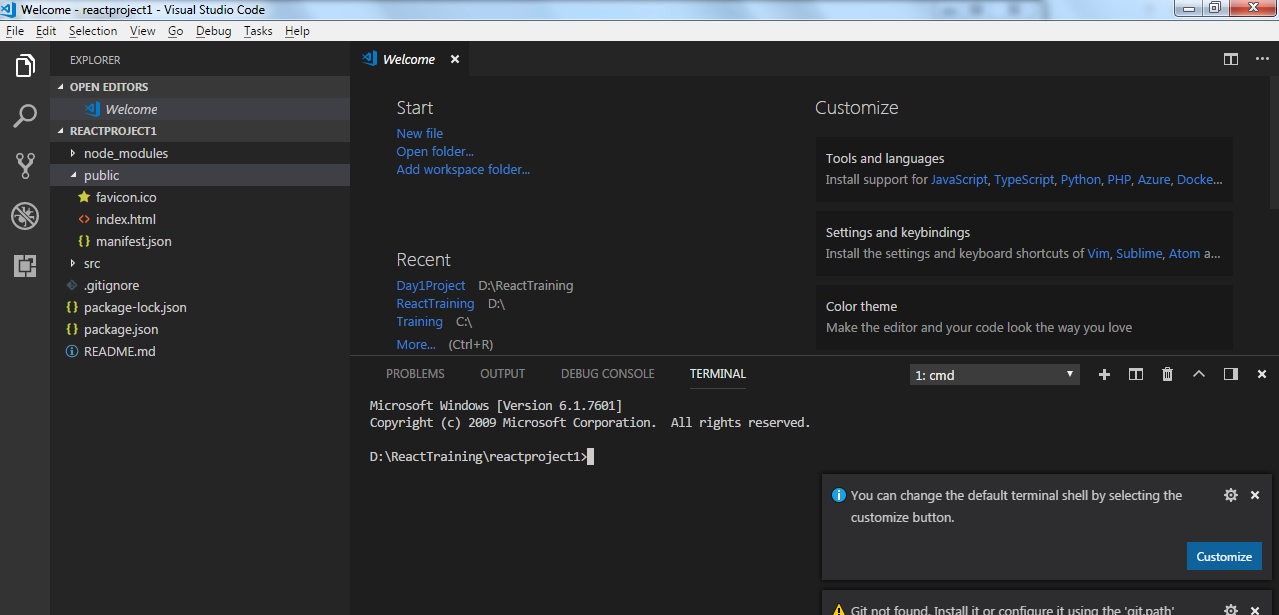
nm I - g create-react-app

pm test 0 Runs the testwatcher in the interactive mode & builds for Prod

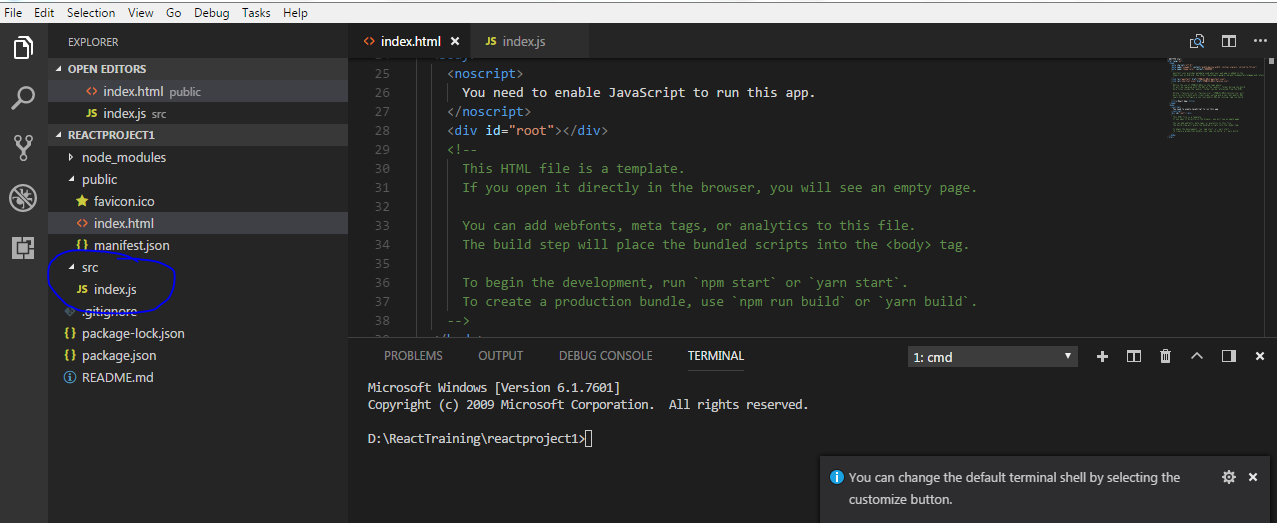


Delete the file in SRC folder





Whe react starts looks for Index.js 🡪 created.



EXPORT & IMPORT

Module.js

export class Point{

constructor(x,y){

this.x=x; //this.x = Instance Variable & x= local variable

this.y=y;

}

display(){

console.log(this.x + ":" + this.y);

}

}

export class Point3D extends Point{

constructor(x,y,z){

super(x,y);

this.z=z; //this.x = Instance Variable & x= local variable

}

display(){

super.display(); // displays only x & Y values

console.log(this.x + ":" + this.y + ":" + this.z);

}

}

Index.js

import { Point,Point3D} from "./module1"

console.log( "After Import From Index.js react");

//Define the class Point wiyh data

let P= new Point(10,15);

P.display();

//Inheritance

let P3= new Point3D(20,30,40);

P3.display();

DEFUALT EXPORT – Shd not have the curly Braces

NAMED EXPORT 🡪 Import within curly braces {} after the class ends we can mention as export default Point

There can be only one default in a JavaScript Class.

BUILT IN TAG 🡪 shd be in Lower case, CUSTOM TAG 🡪 CamelCase

**REACT IMPLEMENTATION with JSX**

import React from "react";

import ReactDOM from "react-dom"

import Point, {Point3D} from "./module1"

//Create ab Element

//let element = React.createElement("u",null,"My first React Example");

let root= document.getElementById("root");

//Functioanlity to Implement - for display we use the REACTDOM

//ReactDOM.render(element,root);

ReactDOM.render(<h3>JSX Example</h3>,root);

console.log( "After Import From Index.js react");

//Define the class Point wiyh data

let P= new Point(10,15);

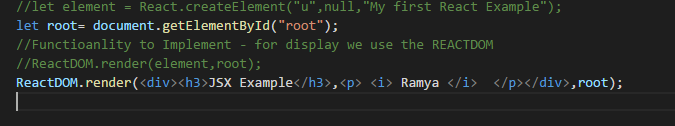
P.display();

//Inheritance

let P3= new Point3D(20,30,40);

P3.display();

For Multiple elements in JSX we need to use DIV or SPAN tag -Image below

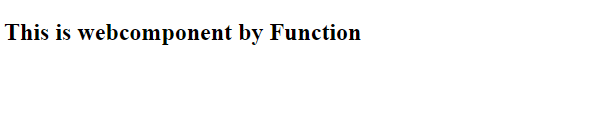


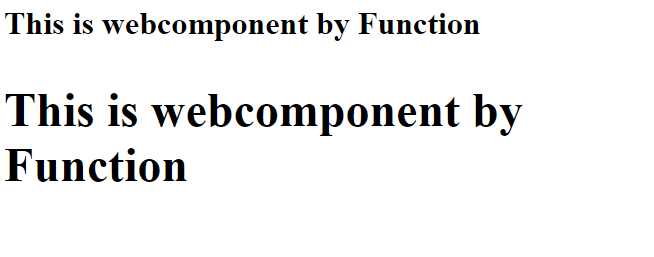
RENDERING NON\_ELEMENT CONTENT

SYNTAX – will be in CURLY Braces {}

**Create a Web Component**

Two types of creation of Creating Web Component

1. Function based creation (Arrow based creation)
2. 
3. //Create a Web Component
4. let Mytag =() => <h1>This is webcomponent by Function</h1>
5. ReactDOM.render(Mytag(),root);
6. //Create a Web Component
7. // Modified to include Arrow function learnt earlier
8. let Mytag1 =() => <h2>This is webcomponent by Function</h2>
9. let Mytag =() => <h1>This is webcomponent by Function<Mytag1/></h1>
10. ReactDOM.render(Mytag(),root);



1. Class based Web component

STEP -1 – Create<<App>> as a component <<APP.js>>

import React,{Component} from "react";

App.js

//Component Mmber is a named Export

import React,{Component} from "react";

class App extends Component{

//Shd override the Render Method

render(){

return <div> <h1> Welcome to React from Class Based Component </h1></div>

}

}

// We need to Use it in index.js --> export defalut

export default App;

index.js

//Steps for Component Based on Class

import App from './components/App';

import ReactDOM from "react-dom"

import React from "react"

ReactDOM.render(<App/>,document.getElementById("root"));

Component based – Class Based is best with Data , Function based is Dumb

# STATE IN REACT COMPONENT

State is stored as Object specified in Curly braces,[Json Format] Stored as JAvascript Variable

Mutable React Data.

State – {key : value} [Shd be a Json Object]

class App extends Component{

//Shd override the Render Method

//Definingthe state component

state= {Compheading : "Values"}

render(){

return <div> <h1> Welcome to React from Class Based Component {this.state.Compheading}</h1></div>

}

}

CHILD COMPONENTS – NESTED COMPONENTS