Monday, May 15, 2023 11:45 AM

ECMA Script 2015 also called as ES6 = Specification based on which javascript syntax is created

Spread operator = used for copy of arrays and objects
= we can use it to achieve DEEP copy also

DEEP copy Vs Shallow copy

Destructuring Syntax = extracting properties of objects or arrays in local variables

a. Object destructuring (LHS curly brackets)
let { a} = obj //for property "a" is extracted in local variable "a"
let {a:x} = obj // for property "a" is extracted in local variable "x"

b. Array destructuring (LHS square brackets)
let [p] =arr //extract 0th element of array in local variable p

---

Callback ---- The functions are added to callback queue.
This queue is executed after the main stack is empty

Promises ---- the promise executor is added to one more data structure called as promise queue
It is different from callback queue
It is having higher precedence than the callback queue
It is also executed after the main stack is empty

Framework Vs Library

Framework = the architecture / design of application is already created, We can customize some customizable parts

Library = some readymade functionality or code is available , we can use it and design our own architechture

ANGULAR = Framerwork
REACT UI Library !!
Component based Library
UI can be divided into components like the header component, footer component, signup component , displaycomponent , etc
Final application = Integrating all the compoments

We can REUSE some components in other applications !!

React Library is BASED on Javascript !!!

To start working in react ----

GET a readymade project template	npx = node pacakge executor
npx create-react-app APPName	

In the lab = create a folder ReactApps
Cd to this folder
Type the command   npx create-react-app IetMay23

Once the REACT app is created
Observe the contents

1	node_modules folder	JS -Libraries are present here
2	public	Has some images , manifest.json , index.html
		Only one html is created in the entire application - index.html }} this goes to the browser where is it rendered
3	src	folder where all our JS files are kept . All the components are placed here
		index.js = this is the **starting point** of a react app we are adding a root TAG , it is a custom tag . we add it to the <div id="root" > </div>
4	package.json	It contains the depencies and scripts used in the application

React Pages are rendered from a web server = React provides a DEVELOPMENT web server . We can use this server as long as we are developing our application.

For DEPLOYMENT phase = we have to build a react "build" and add it to the Webserver of your choice !!!

To start React development web server -- use the command   npm start
Always start the web server from the project folder

Access the web server from the browser using the URL as follows ---
  http://localhost:4000

____

Modify the App.js , save it and Refresh the browser .
Whateve changes are made to the js files are auto-deployed (HOT -DEPLOYMENT) = server need not be restarted after every change .

Make changes in App.js , see that the server auto deploys . Do some mistake .
Observe the error on the server window

Go to the browser - inspect window see the Elements -- observe the app tag is added to the div root
____

React components are of two types
1. Functional components
2. Class components

Create a class Component Welcome !!
1. Open a file name it as Welcome.js in src folder (please make first letter

capital)

Tags with all small letters are considered as HTML tags

Tags with first letter capital and other small is considered as custom tag

2. Write a class Welcome , extend it from React.Component
3. Export the class
4. Render Html in the class
5. Connect the component with APP component
6. Save everything and observe the browser

____

Create a Functional Component Greeting

1. Create a file Greeting.js
2. Write a function Greeting
3. Export default
4. Return the html
5. Connect the component with App Component
6. Save and observe the browser

____

(NOTE) the HTML returned by the component MUST be a single tag !!!

Write a class component Login

Return the div that contains 2textfields and button

Write a function component CurrencyConvertor

Return the <div> containing currency convertor elements

Add both of them to the App.js

____