Freecode camp

<https://react-projects.netlify.app/>

https://github.com/john-smilga/react-projects

top of name

"homepage": "https://Ganesh-Paulraja.github.io/react-nav-one",

npm install --save gh-pages

npm install gh-pages --save-dev

yarn add gh-pages –-save

yarn add gh-pages --save-dev

top of scripts

"predeploy": "npm run build",

"deploy": "gh-pages -d build",

npm run deploy

yarn run deploy

<https://react-projects.netlify.app/>

https://github.com/john-smilga/react-projects

React is a open source javascript library for building user interfaces

* Popular UI library
* Great community support
* Component based architecture
* Easy to create complex ui

**Install**

node js

Browser → chrome

Vs code

* es7 react extension
* emmet jsx → user settings → extension → emmet → include language →

javascript : javascriptreact

* user settings → tab size → 2 (optional)

Comment:

npx create-react-app [folder name]

npx create-react-app . → for current folder

(or)

yarn create react-app my-app

yarn start

yarn build

yarn eject

yarn start

yarn install // or simply, yarn

yarn add react react-dom

—------------------------------------------------------------

tailwind

yarn add -D tailwindcss postcss autoprefixer

yarn tailwindcss init -p

tailwind.config.js

module.exports = {

content: ["./src/\*\*/\*.{js,jsx,ts,tsx}"],

theme: {

extend: {},

},

plugins: [],

};

index .css

@tailwind base;

@tailwind components;

@tailwind utilities;

—------------------------------------------------------

for starting → public inside index only need

npm i sass – save

yarn add sass --save

all names capitalise

outer error

import export;

props

* props get passed to the component
* Function parameters
* props are immutable

State

* State is managed within the component
* Variables declared in the function body
* state can be changed
* useState Hook

Short circuit operator

React 18 – march 2022

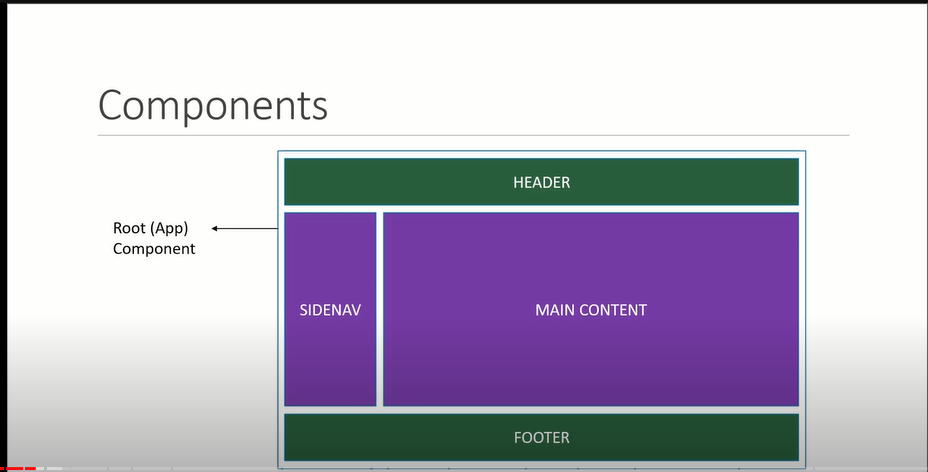
React is a open source javascript library for building user interfaces

\*only library

\* Building user interfaces

* React has a rich eco system
* great support form user and developers
* component based architecture

npx create-react-app name



render

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<React.StrictMode>

<App name = "What" work= "part" />

</React.StrictMode>

);

—-----------------------------------------------------

Functional Component

const App = () => {

return (

<div>

<h1>Hello New Function</h1>

</div>

)

}

we can use any name for importing this component

use capitalise

export default App;

import AppTwo from './App';

Named Exports

export const App = () => {

import {App} from './App';

—-----------------------------------

Jsx – Javascript extended

html + js

class replaced by classname

for -> htmlFor

camelCase property naming convention for html attribute

onclick -> onClick

easy arithmatic

return (

<div>

<h1>Hello New Function {5+6}</h1>

</div>

)

props

value cannot be change (immutable)

export const App = (props) => {

return (

<div>

<h1>Hello New Function {props.name} {props.work}</h1>

</div>

)

}

export const App = ({name,work}) => {

return (

<div>

<h1>Hello New Function {name} {work}</h1>

</div>

)

}

<React.StrictMode>

<App name = "What" work= "part" />

</React.StrictMode>

STATE

export const App = () => {

const[message, setMessage] = useState('Welcome');

const btnCss = {

background: 'green',

cursor: 'pointer',

}

return (

<div>

<h1>{message}</h1>

<button onClick={() => setMessage('Changed')} style = {btnCss}>Change</button>

</div>

)

}

PARENT TO CHILD

export const ChildComponent = (props) => {

return (

<div>

<button onClick= {props.clickFun} >I AM CHILD</button>

</div>

)

}

import { ChildComponent } from "./Child"

export const ParentComponent = () => {

const clickFun = () => {

alert('work');

}

return (

<div>

<ChildComponent clickFun = {clickFun} />

</div>

)

}

CONDITION RENDERING

ternary operator js

{message == 'Welcome'? 'Change': 'Completed'}

{message == 'Welcome' && 'Change'}

LIST RENDERING

export const App = () => {

const frootList = ['Apple', 'Orange', 'Graphs'];

return (

<div>

{

frootList.map((val, i) => {

console.log(val, i)

let c = i+1;

return (

<div>{c}. {val} </div>

)

})

}

</div>

)

}

EFFECT HOOK (LIFE CYCLE)

// componentDidMount, componentDidUpdate == after rendering the component == effect hook == for side effects

\*It’s looping when updating useState

useEffect(() => {

console.log('work');

})

\*It’s not looping when updating useState

useEffect(() => {

console.log('work');

},[])

\*It’s looping when updating exact useState

useEffect(() => {

console.log('work');

},[msg])

Two way binding

When writing something inside input also state updating

When updating state also frontend will updating

—------------------------------------------------------------------

**React Nav**

**npm i react-router-dom**

import { BrowserRouter, Routes, Route} from 'react-router-dom';

<div className="App">

<div className="full-wrap" style={{textAlign: 'center', color : 'red'}}>

<BrowserRouter>

<Header/>

<Routes>

<Route exact path="/" element={<Home/>} />

<Route path="/LogIn" element={<LogIn />} />

</Routes>

</BrowserRouter>

</div>

</div>

import { Link, NavLink } from "react-router-dom";

<Link to="/LogIn">Login</Link>

<NavLink to="/about" activeclassname="active">About</NavLink>

—-----------------------------------------------------------

Formik

https://www.youtube.com/watch?v=vJtyp1YmOpc

Hello world

REDUX

npm i redux react-redux @reduxjs/toolkit

Create store in index.js

import {configureStore} from ‘@reduxjs/toolkit’

Import {Provider} from ‘react-redux’

-----s2

import userReducer from ‘./features/user’

Cost store = configureStore({

reducer: {

-----s2 → after that we can access user in any component

user: userReducer,

}

})

Wrap app inside provider

<Provider store={store}>

<App/>

</Provider>

—-------------------s2

features – > user.js

Import {createSlice} from ‘@reduxjs/toolkit’

Const userSlice = createSlice({

name: ‘user’,

initialState: {

Value: {

name: ‘’,

age: 0,

email: ‘’

},

reducers: {

login: (state, action) => { → action name

state.value = action.payload

}

}

}

});

export default userSlice.reducer;

—------------------------------------------S3

access state

import {useSelector} from ‘react-redux’’

const user = useSelector(state => state.user.value)

<>

<div>user.name</div>

<div>user.age</div>

<>

—------------------------------------------S4

update State

import {useDispatch} from ‘react-redux’

import {login} from ‘./features/user’

<button onClick = {() => dispatch(

{

name: ‘’,

age: 0,

email: ‘’,

}

)}>Btn</button>

swiper – react

npm i swiper

—----------------------------------------

import {Swiper, SwiperSlider) from ‘swiper/react’’;

import ‘swiper/swiper.scss’

MERN ecommerce

REACT QUESTIONS

1. what is react?

React is a popular JavaScript library for building user interfaces (UIs) \

React has rich ecosystem.

Component based architecture and virtual dom

by building reusable components we can reduce the development time

virtual dom

The virtual DOM is a lightweight representation of the real DOM

can manipulate much faster than the real DOM.

reacts components and virtual dome make a website fast and efficient

2.) lifecycle methods vs hooks

lifecycle methods with react class components

React Hooks are functions that let you "hook into" React state and lifecycle features from functional components

useState, useEffect,

3.)uni directional data flow?

https://www.youtube.com/watch?v=W7U-4GfcYvc

two way binding?

it’s used in form input

4.) what is react memo?

replacing the shouldcomponentupdate in react hook

https://www.youtube.com/watch?v=HUSXvYbN3gA&t=102s

redux is efficient in transferring data child to paent

may be we try contact api if the application is too small

5.) what is hoc any use case?

**lazy loading, atuhundication**

https://www.youtube.com/watch?v=2zqWG2--0pc

6.)what is mounting

component did mount and component unmount

7.)how to optimize react application give me any example

https://chatgpt.com/c/9b177ba8-64fa-4f40-889e-248866cd4194

### **1. Memoization**

Memoization helps to avoid unnecessary re-renders by caching the results of expensive calculations.

### **2. Code Splitting**

Code splitting helps to load only the necessary parts of the application, reducing the initial load time.

### **3. Avoiding Anonymous Functions**

Passing anonymous functions directly in props or attributes can cause unnecessary re-renders. Instead, use useCallback to memoize these functions.

### **4. Virtualize Long Lists**

Rendering long lists can be expensive. Using libraries like react-window or react-virtualized can help by only rendering the visible items.

### **6. Optimizing Images**

Ensure images are optimized for the web. This can include:

### **7. Use a Performance Profiler**

React DevTools includes a Profiler that helps you identify performance bottlenecks in your application. By analyzing the components that take the most time to render, you can target specific optimizations.

8.) how to pass data child to parent

1. Call back method

2.contxt api

3.stat management libraries (Redux)

9.) ways to call api

### **1.Using Fetch API**

### **2. Using Axios**

### **3. Using Async/Await with Fetch**

Using async and await with Fetch can make the code cleaner and more readable.

10.) React contact api?

11.) new futures of react latest release

12.) what is redux? how it works?

13.)what are middlewares what happen if they are not there?

14.) redux thunk vs saga?

15.) promis in js? async and await?

16.) what is call back hell

17.)Explain some feature of ES6

18.) find min value in array?

19.) useEffect, useState

20.) features in react

21.) jsx, component

22.) diff between virtual dom and real dom

23.) reat native?

24.) redux flow explain?

25.) what is js?

26.) map, filter, find, forEach

27.) map vs forEach → map creating new array

28.) props → parent to child

29.) sate vs props

30.) react router

31.) html , html layout

32.) What are tags

33.) strong vs bold in HTML

34.) Inline vs external

35.) type of css

36.) tell me the out put

“B” - “A”

“2” - 2

“2” + 2

37.) event loop, eventque , callback que

38.) in callback promis, set time whic one act first

39.) prototype, constructor function

40.) this keyword in js

41.) closure, var , const , let

42.) advantage and disadvantage of react js

43.) what makes react faster?

44.) can we use jsx directly

45.) can we rerender without updating state

Website

Browser --> Client --> Server --> DataBase

What is React? What is the Role of React in software development? V.Im

\* React is an open source Javascript Library for building User Interfaces

\* It has rich ecosystem, Greate community support, easy to use with other libraries and moduels

\* React simplifies the creation of complex UI and SPA by using reasable components

Key futures

\* Virutal Dom

\* Component Based Architecture

\* Reusability

\* JSX

\* Community & Ecosystem

\* React Hooks

What is Dome? What is the difference between HTML and DOM

Dom stands for document object model

Dom is a top part of dom tree representation

we can manuplate dome to make changes in html

Dom vs virtual dom

The virtual DOM is a lightweight representation of the real DOM

can manipulate much faster than the real DOM.

when we are making changes in real Dom for every single changes It will take more time

Bue when we are making changes in virtual DOM it collects all changes and update real Dom

Components? What are the main elements of it?

These are Javascript Functions or Classes returning some JSX value

JSX is looks like html but we can perform some js operation with it

SAP ?

without reloading the page update the data of web page and switching one page to another page that is called SAP

React advantage and disadvantage

adv:

simple to build single page application

react is cross plantform and open source

Light weight and very Fast because of virtual dom

Large community and ecosystem

easy testing

disadvantage:

we need to depend on other library still we are installing what we want only so it’s not a big problem

not very good choise for static very small websites (webswites not going to update component reuse component)

Jsx?

JavaScript XML

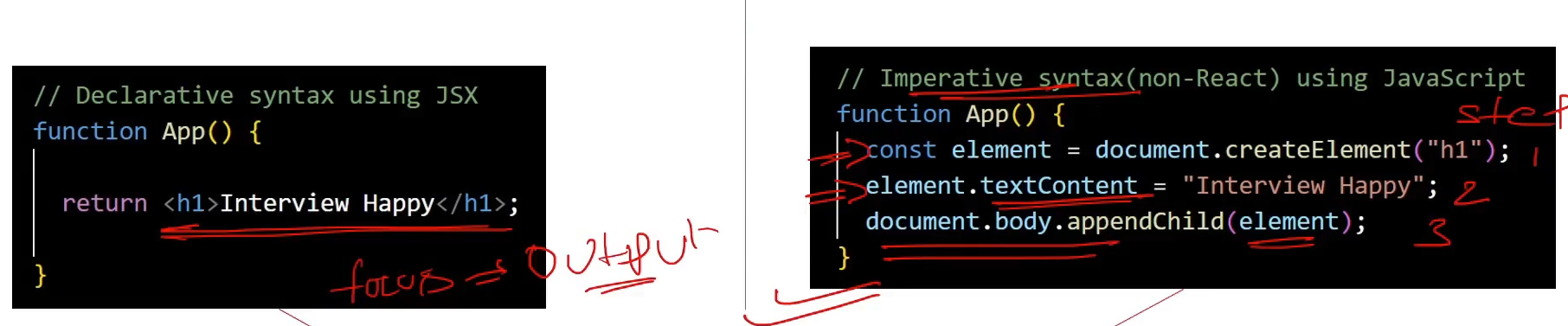
JSX is looks like html but we can perform some js operation with it

className instead of class and htmlFor instead of for

with the help of bable library we are converting this to js

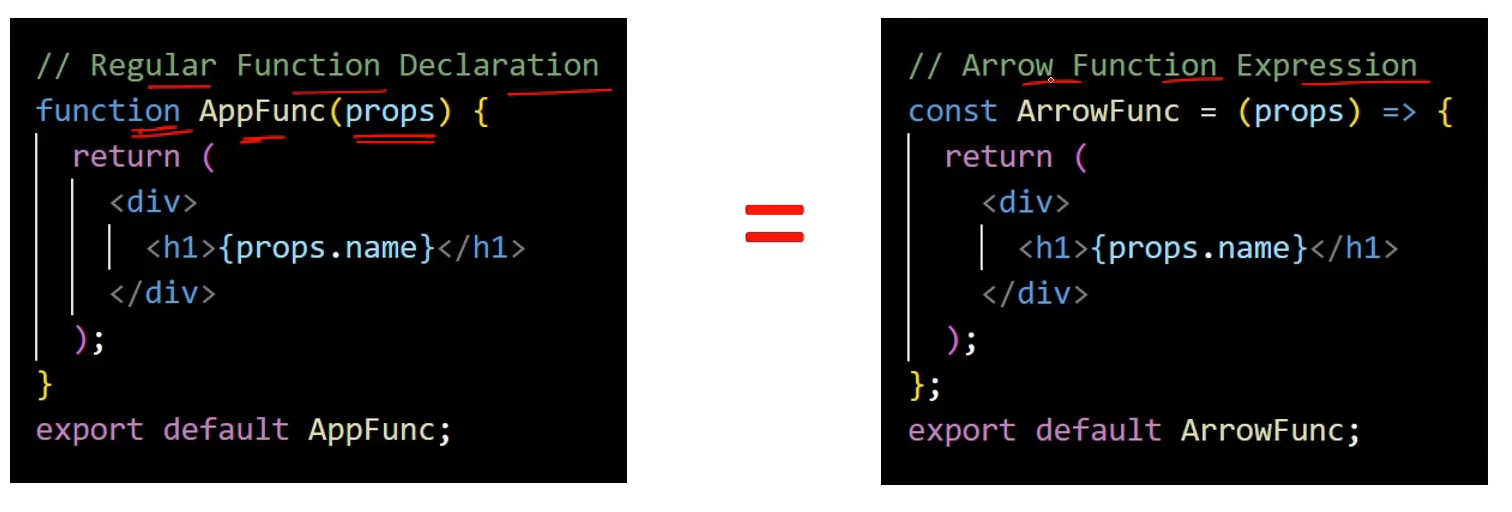
It’s maket he process simpler for developers

Declerative vs Imperative syntax



—-------------------

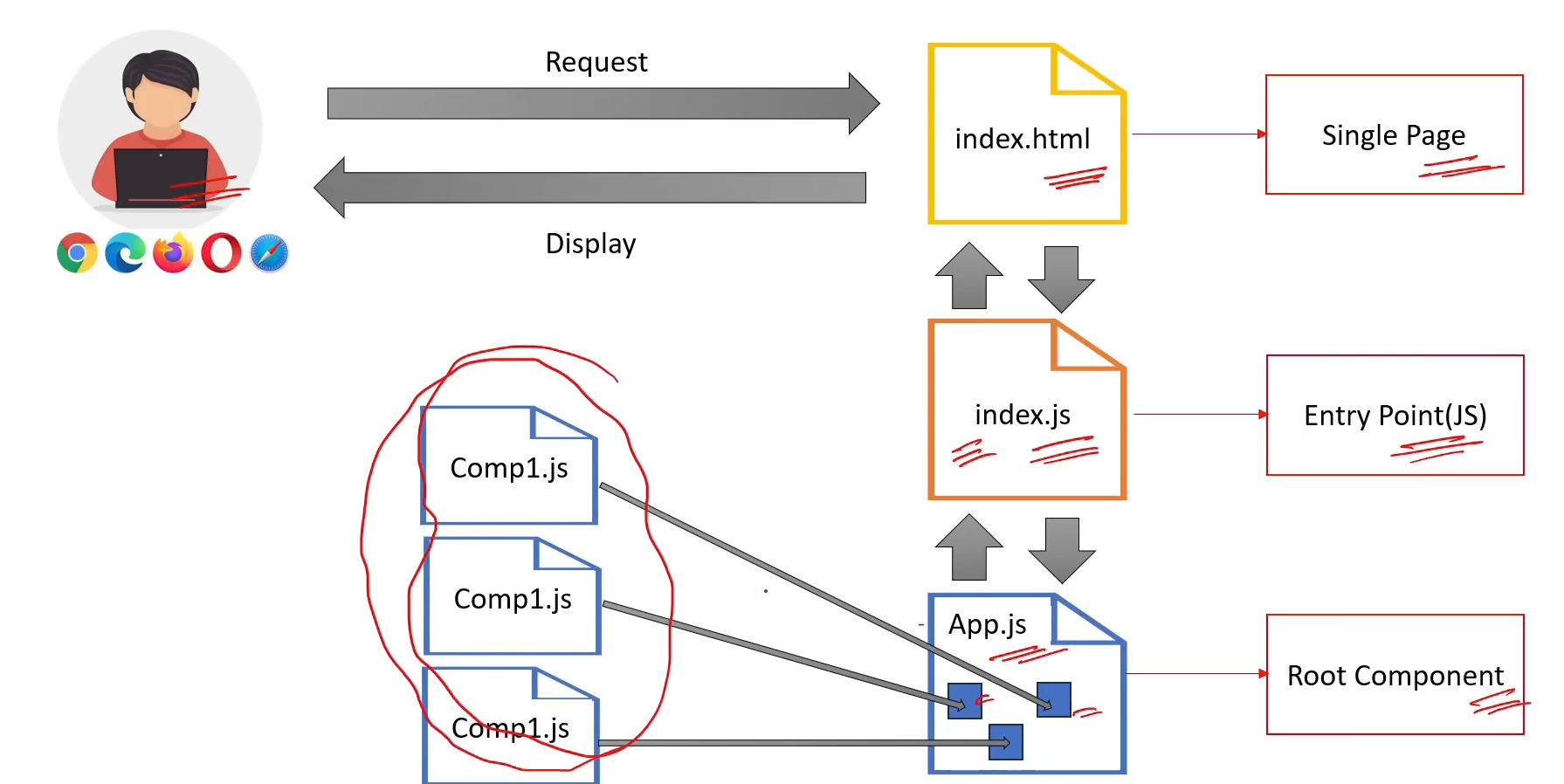
What is the Arrow Function Expression

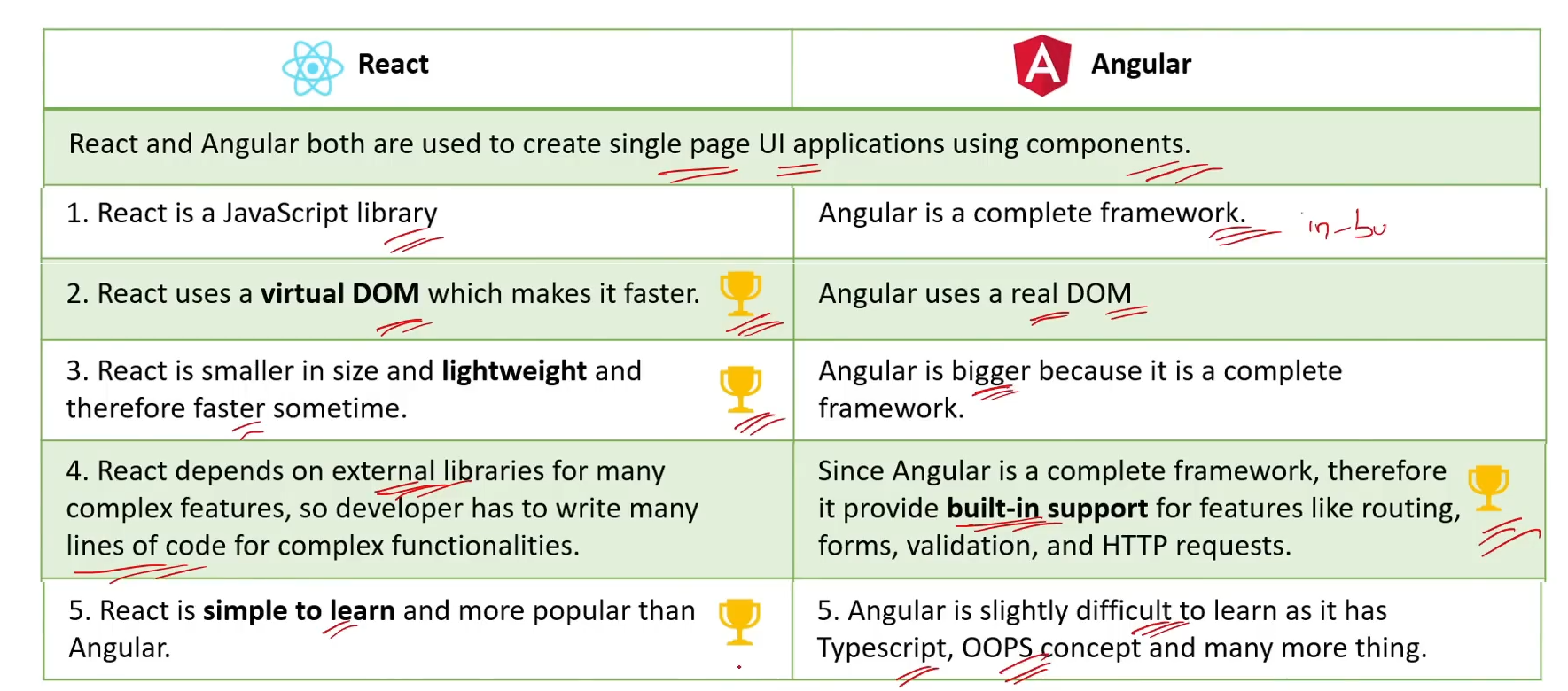
  
The arrow function expression syntax is a concise way of defining functions

Function Extraction

Assigning function value to a variable eg: arrow function, named function

How React App Load and display the component in browser?





React is a Framework or a Library? What is the difference?



**How React Reusability and Composition?**