React

JavaScript library for user interface designing

Declarative

Component based-class based and Functional component

Consist of JSX

Pre-requisite

Intall node js

To install react

npx create-react-app my-app

To run application

npm start

simple code:

import ‘./App.css’;

const App=()=> {

return(

<div className=”App”>

<h1>Hello {name}!</h1>

</div>

);

}

}

export default App;

React fragment used to render multiple/different element of html

{

name?(

<> 🡨 react fragment

test

</>

): (

<>

<h1>test</h1>

<h2>test2</h2>

</>

)

}

Creating a functional component

Const Person = () => { //component name to be started with capital

Return (

<>

<h1>test</h1>

<h2>test2</h2>

<h1>test3</h1>

<h2>test4</h2>

</>

)

}

const App=()=> {

return(

<div className=”App”>

<Person />

<Person />

<Person />

</div>

);

}

}

Props

Allow u to pass navig data through react components. Passed via attributes.

Const Person = (props)=> {

Return (

<>

<h1>Name: {props.name}</h1>

<h2>Last Name: {props.lastName}</</h2>

<h2>Age: {props.age}</</h2>

</>

)

}

const App=()=> {

return(

<div className=”App”>

<Person name={‘John’} lastName={‘Doe’} age={25}/>

<Person />

<Person />

</div>

);

}

}

State

Plain JavaScript object represent a piece of information about the components current situation

Pre-requiste

import {useState} from ‘react’;

Hook

Whenever we start a function and we start with use then it is called hook

import {useState} from ‘react’;

import ‘./App.css’;

const App=()=> {

const [counter, setCounter] = useState(0);

useEffect(()=>{

setCounter(100);

},[]);

return(

<div className=”App”>

<button onclick={()=> setCounter((prevCount)=> prevCount -1)}>-</button>

<h1>{counter}</h1>

<button onclick={()=> setCounter((prevCount)=> prevCount +1)}>+</button>

</div>

);

}

}

export default App;

Building application

App.css-from git

search.svg-form git

Index.js

import React from ‘react’;

import ReactDOM form ‘react-dom’;

import App from ‘./App’;

ReactDOM.render(<App/>, document.getElemenyById(‘root’));

MovieCard.jsx-react component

Import React form ‘react’;

Const MovieCard=({movie})=>{//used movie1 instead of props, if props used should be mentionsed everywhere

return(

<div className=”movie”>

<div>

<p>{movie.Year}</p>

</div>

<div>

<img src={movie.Poster}!==’N/A’? movie.Poster: ‘https://via.placeholder.com/400’ alt={movie.Title}/>

</div>

<div>

<span>{movie.Type}</span>

<h3>{movie.Title}</h3>

</div>

</div>

)

}

Export deault MovieCard;

App.js

Import React from ‘react’;

Import ‘./App.css’;

Import SearchIcon from ‘./search.svg’

Import MovieCard from ‘./MovieCard’;

const API\_URL=”https:www.omdbi.com?apikey=num.”;//num to be put when created an account

const movie1={

json from git

}

const App=()=> {

const [movies, setMovies] = useState([]);

const [searchTerm, setSearchTerm] = useSate([]);

const searchMovies = async(title)=> {

const response = await fetch(‘$(API\_URL)’&s=$(title)’);

const data = await response.json();

console.log(data.Search);

}

useEffect(()=> {

searchMovies(‘Spiderman’);

},[]);

return (

<div className=”app”>

<h1>MovieLand</h1>

<div className=”search”>

<input

Placeholder=”Search for movies”

Value={searchTerm}

Onchange={(e)=>setSearchTerm(e.target.value)}

/>

<img

Src={SearchIcon}

Alt=”search”

Onclick={()=>searchMovies(searchTerm)}/>

</div>

{

movies?.length > 0

?(

<div className=”container”>

{movies.map((movie)=>(

<MovieCard movie1={movie}/>

))}

</div>

): (

<div className=”empty”>

<h2>No movies found</h2>

</div>

)

}

/\*

<div className=”container”>

/\*

<div className=”movie”>

<div>

<p>{movie1.Year}</p>

</div>

<div>

<img src={movie1.Poster}!==’N/A’? movie1.Poster: ‘https://via.placeholder.com/400’ alt={movie1.Title}/>

</div>

<div>

<span>{movie1.Type}</span>

<h3>{movie1.Title}</h3>

</div>

</div>

\*/

<MovieCard movie1={movie1}/>

</div>

\*/

</div>

)

}

**Web dev simplified**

Props is passed into a component and props handled outside

State is inside a component and handled inside

State is re-rendered when it changes

Hook-only for function component

Cannot be put inside the conditional statement

useState

it consists of two parameter inside a array as it returns array of two parameter

const [count, setCount]= useState(4)

count is the current state and setCount is the updated state(function) to be set.

Fucntion decre(){

setCount(prevCount=>prevCount-1)

}

onclick={decre} inside a button

instead of 4 we can give an function to only run once, otherwise everytime u render

it renders the value which is not needed

const [count, setCount]= useState(()=> {

console.log(“run function”);

return 4;

})

For merging property when a single property is updated …prevState is to be used

Or can use multiple state

Const [state, setState]= useState({count: 4, theme:’blue’})

Const count = state.count

Const theme = state.theme

Fucntion decrementCount(){

setState(orevState => {  
return {…prevState, count: prevState.count-1}

})

}

useEffect

Freecodecamp

Why react

Composable- component based

Declarative- like variables so no need to repeatedly declare

Hireable skill

Actively maintained by skilled people

Props used as function for creating dynamic components. Like youtube home page tiles.

It cannot be just imported as it becomes static without change in data.