Tutorial 1


Is React JS Library or Framework?
React is not a framework. React is a JavaScript library for building user interfaces.
It is also known as ReactJS and React.js, so don't get confused if you read different notation in different places.
React knows only one thing that is to create an awesome UI.


React History
React was first designed by Jorden Walke, a software engineer at Facebook
It was first deployed for Facebook News feed around 2011.
In 2013, React was open sourced at JS conference.

Open Source

An open source project is where the code to a certain project is completely open source.

That means that anybody can readily see the code that went into a project.

Open source projects also are usually community-based, and accept help from other programmers.

About React

Component based approach

A Component is one of the core building blocks of React.
In other words, we can say that every application you will develop in React will be made up of pieces called components.
Components make the task of building UIs much easier.

About React

Component based approach

uses a Declarative Approach

Declarative programming is a programming paradigm ... that expresses the logic of a computation without describing its control flow.

About React

Component based approach

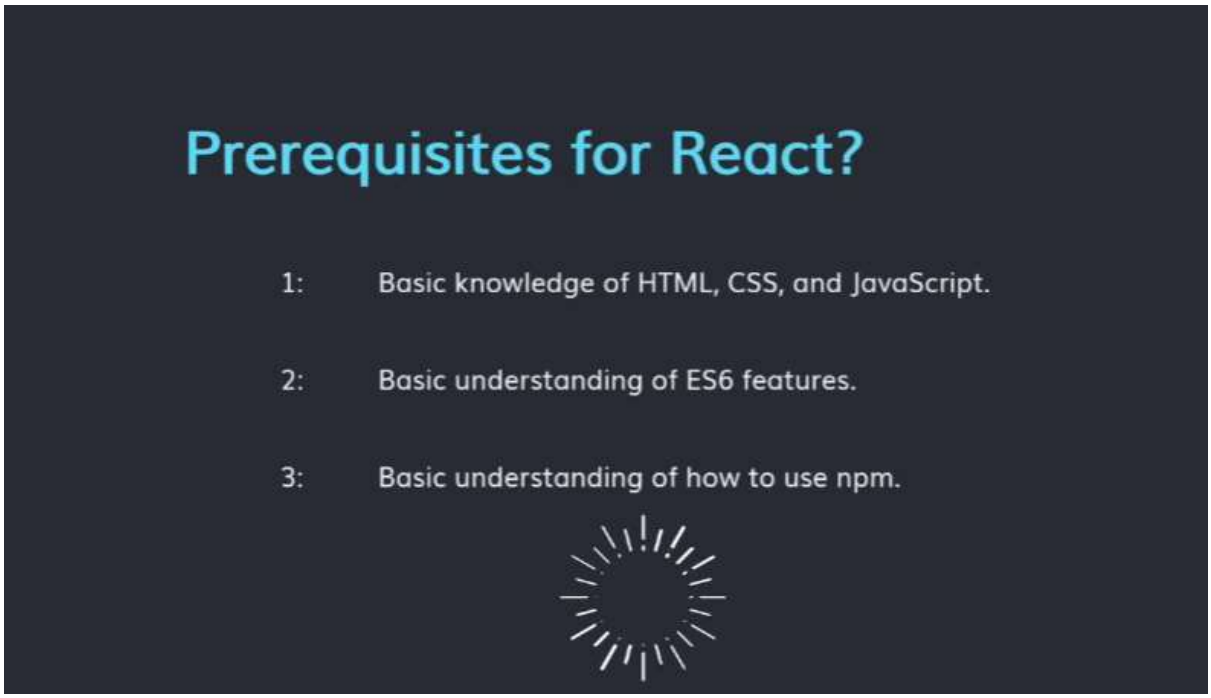
uses a Declarative Approach

DOM updates are handled gracefully

Reusable Code

React is designed for speed, speed of implementing the application simplicity and scalab

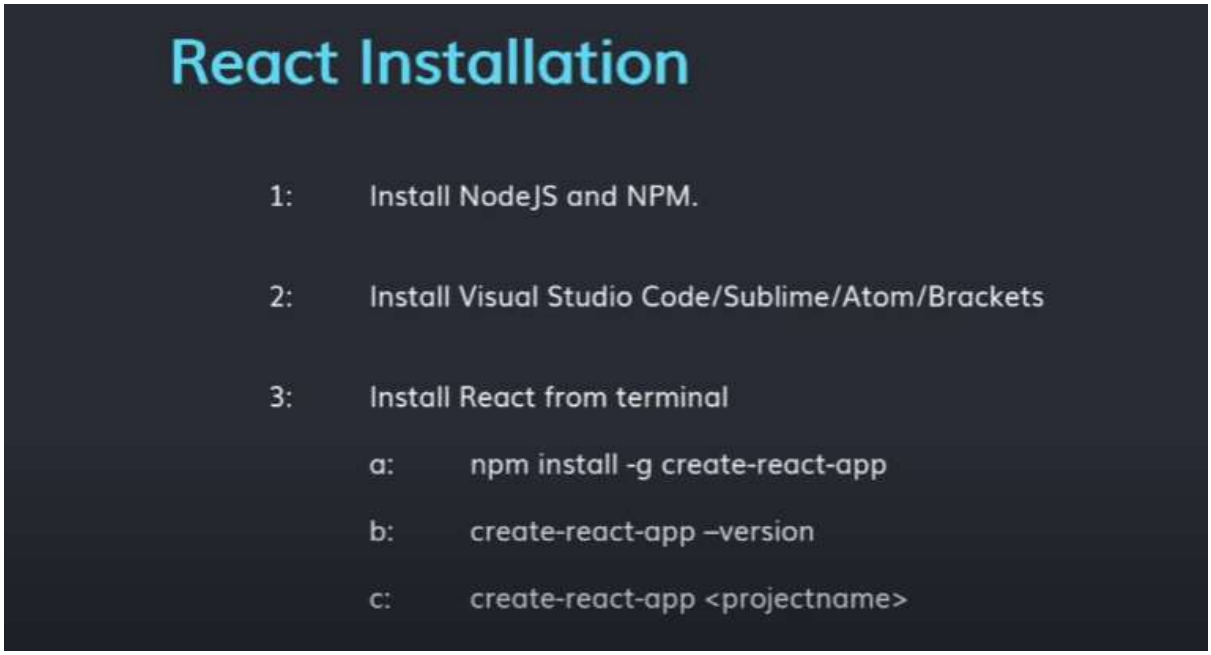
TUTORIAL 2


Prerequisites for React?
1: Basic knowledge of HTML, CSS, and JavaScript.
2: Basic understanding of ES6 features.
3: Basic understanding of how to use npm.

TUTORIAL 3

Babel code is just highlight es6 code so we know that the code will be es6 code.

TUTORIAL 4

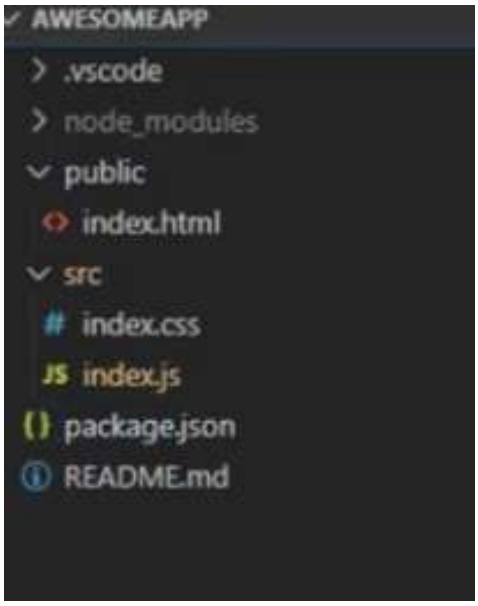


TUTORIAL 5

React Folder Structure| js babel and weback

1. Node_module folder is the repository of modules/library which you are using inside your project. What ever you are importing in your project that module or library should present inside the node_module folder. When you do npm install that time that module or the library install inside the node_module folder and one entry added in package.json file.
2. Package.json:- react app meta data like version,scripts,etc.
3. Index.js is most important in which we can create our full code.

4. Index.css:- it is used for styling component to look better ui.
5. App.js:- it is basically a component we can learn this component later.
6. Roboto.txt:- it is basically used for website crawling.
7. Manifest.json:- it provide the information about over app.
8. Index.html:- this file is basically used for import cdn and extra js file which is usefull when we are creating website template.



This is the structure that is important most to work as a beginner.

When we are working in react we need two module compulsory which we need requirement and that is (remove all from index.js).

```
Var React = require('react');
Var ReactDOM = require('react-dom');
{if you use ES5 with npm, you can write var
ReactDOM = require('react-dom').}
```

ReactDOM.render():- it means jst display and it takes 3 parameter
ReactDOM.render('kya dikhana hai','kaha dikhana hai','callback function');

ReactDOM.render(<h1>Hello World</h1>,document.getelementbyID("root"));
This is not html this is jsx and we can learn jsx deeply after some time.

How bable compile the code to understand by browser and this will automatically install

To write this,
Var React = require('react');
Var ReactDOM = require('react-dom');

We can use import thing so its easy to write code

Import React from 'react';
Import ReactDOM from 'react-dom';

(this will convert using babel so its better to understand)

TUTORIAL 6

React JSX

Jsx stands for javascript extension/javascript XML.

When we want to use html attribute in react then we must use jsx and for using jsx we need

Import React from 'react';

Normal	Using jsx
ReactDOM.render(/*#__PURE__*/ React.createElement("h1", null, "Shiva Mahadev"), document.getElementById("root"));	ReactDOM.render(Shiva Mahadev ,document.getElementById("root"));
ReactDOM.render(/*#__PURE__*/ React.createElement("h1", { className: "std" }, "Shiva Mahadev"), document.getElementById("root"));	ReactDOM.render(<h1 className="std">Shiva Mahadev</h1>,document.getElementById("root"));

Or we can write using javascript :-

Var h1=document.createElement("h1");
H1.innerHTML="hello shiva";

```
Document.getElementById(‘root’).appendChild(h
1);
```

TUTORIAL 7

How to Render Multiple Elements inside ReactDOM.render()

Render take only takes single element or node but if we used multiple element then we can do use div tag then they cant show parse error.

```
ReactDOM.render(
<div>
<h1>Hello</h1>
<p>Shiva</p>
</div>,
Document.getElementById(“root”));
```

But in react v16 its possible for render() to return an array of element

```
ReactDOM.render(
[
<h1>Hello</h1>,
<p>Shiva</p>,
<h3>Shiva</h3>
],
Document.getElementById(“root”));
```

TUTORIAL 8

React Fragment

In inspect screen we can see that from using
ReactDOM.render(

<h1>Hello</h1>
Shiva
,
Document.getElementById("root"));

They can show

<h1>Hello</h1>
Shiva

One more div they can show but if we are using React.fragment then this will not show another div so its benifical and don't take to much space{a special fragment syntax. React 16.2 also introduce a syntactical sugar fragments <> </>}.

<React.Fragment>
<h1>Hello</h1>
Shiva
</React.Fragment>,
Document.getElementById("root"));

Or(this will be possible using babel js)

<>
<h1>Hello</h1>
Shiva
</>,

Document.getElementById(“root”));

TUTORIAL 9

React Challenge 1

1. create react app from scratch
2. create one h1 element in it.
3. create one p element in it.
4. create list of 5 name of shiva.

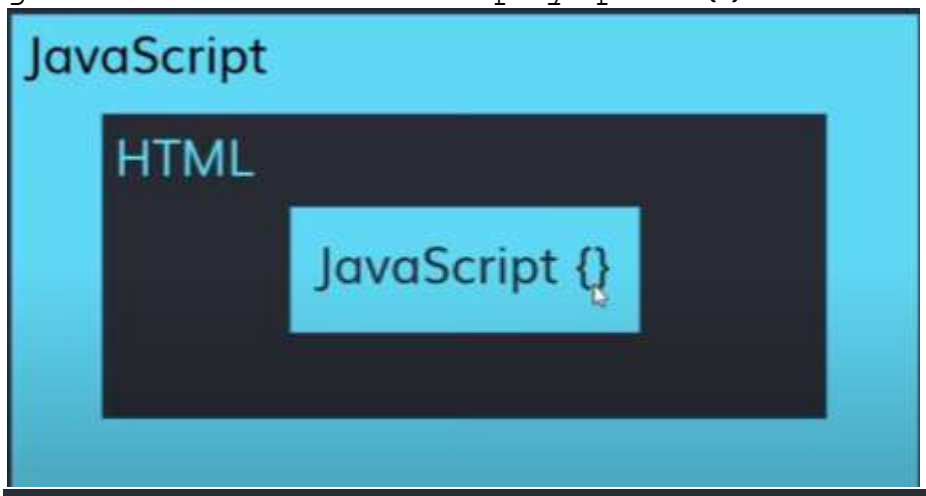
Completed done perfectly

TUTORIAL 10

Expression In JSX

```
const flname= "shiva sankar"
ReactDOM.render(
<h1>i love flname</h1>
    ,document.getElementById('root')
);
```

Use expression this:- we can use javascript under jsx then we can simply put {} braces.




```
const flname= "shiva sankar"
ReactDOM.render(
<h1>i love {flname}</h1>
    ,document.getElementById('root')
);
```

```
const flname= "shiva sankar"
ReactDOM.render(
    <>
<h1>i love {flname}</h1>
<p>Total No of Two No is { 2+3 }</p>
    </>
    ,document.getElementById('root')
);
```

In curly braces we only use expression and we cant use statement like if else condition and much more

TUTORIAL 11

Template Literals In JSX

Use template literal which we use in javascript

` statement which you want to write or display ${variable name} ${another variable name}.`

TUTORIAL 12

Display Current Date And Time In JSX

Challenge #2

1. Create react app from scratch
2. Add one h1 element with your name
3. Add one p element in it with current date
4. Add one p element in it with current time.

```
5.    const curdate = new Date().toLocaleDateSt
  ring();
6.    const curtime =  new Date().toLocaleTimeS
  tring();
7.    ReactDOM.render(<>
8.    <h1>Shiva</h1>
9.    <p>Today's date is {curdate}</p>
10.   <p>Time is {curtime}</p>
11.   </>,document.getElementById("root"));
```

Completely done

TUTORIAL 13

JSX Attribute

Rules of JSX

1. Use camel case eg. contentEditable
2. All tag will be closed in it or element.eg. <img />
3. It gives warning in img when we are not using alt tag in react so it tells us that you should use alt

In html we are using contenteditable

TUTORIAL 14

Styling Attribute

We can use className instead of class.
Because react have its class component so two same name element and component will not allowed in react so we can give className instead of class.

TUTORIAL 15

Adding Google Fonts in React

Copy cdn of your favourite font and paste it into public/index.html
And put style into your css class and see that result

TUTORIAL 16

Inline Styling in jsx

we can use camel case in jsx and {{}} for this see the tut-6.js
consider css as a object and use pascalCase in all attributes now from that process we are using inline css.

```
const heading= {
    color: '#b20029',
    textAlign: 'center',
    textTransform: 'capitalize',
    fontWeight: 'bold' ,
    textShadow: '0px 2px 4px #ffe9c5',
    margin: '70px 0',
    fontFamily:'"Poppins",sans-serif'

};
<h1 style={heading}>My name is {name}</h1>
```

TUTORIAL 17

React Challenge #3

1. Create react app from scratch
2. Show a heading that says hello sir, good morning if time is between 1am to 11am.
3. Good afternoon if 12pm to 7pm.
4. Good night if 7pm to 12am
5. Apply css in it, then dynamically change the color of greeting parts only using inline css style. Ex. Green, orange,black,etc.

TUTORIAL 18

React JS Components

In react every thing is component
For example we are showing puzzles pieces and the pieces are the component In react and the whole puzzle after creation is like react.

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML via a render() function.

Components come in two types, Class components and Function components, in this tutorial we will concentrate on Class components.

Create a Class component called Car

```
class Car extends React.Component {
  render() {
    return <h2>Hi, I am a Car!</h2>;
  }
}
ReactDOM.render(<Car />,
document.getElementById('root'));
```

TUTORIAL 19

React JS Components Challenge #4

1. Do same as do in challenge 3 but this time you have to use component

TUTORIAL 20

React JS Import Export

in import export all the element take from one file and transfer to another file using import export keyword. But we can import in two ways

1. All member

   If we are calling all member then all the element are treated as a object so we can use Import * as xyz from ‘./app’;

   Xyz.element
   Xyz.function();

   Eg.

```
import * as AllExport from "./ExportAllElement";
<ol>
        <li>{AllExport.default}</li>
        <li>{AllExport.nms}</li>
        <li>{AllExport.Aabc()}</li>
        <li>{AllExport.ASub()}</li>
</ol>
```

2. Selected member

   We can call all member of the component by the help of import export but if we want some element to call then this we can use

```
import NormalExport, { names, Sub, Abc } from "./
ImportExportInReact";
<ol>
        <li>{NormalExport}</li>
```

```
        <li>{names}</li>
        <li>{Sub()}</li>
        <li>{Abc()}</li>
      </ol>
```

And it treated as a component element not object.

We have learned so far that React Apps are basically a collection of interactive Components, and from the article, on ReactJS Components we have known how to create components but even with that knowledge, it will not be sufficient to create a full-fledged React App as in order to create a collection of Components we first need to know the way to use and re-use components that may have been defined elsewhere. To do so we need to know two operations broadly known as Importing and Exporting.
We may not have told earlier, but we have been using the import operation in every one of our previous articles when we were importing react and react-dom itself. Similarly, we can also import user-defined classes, components or even a part of the same. Let us shift our discussion over Importing.

- **Importing default export:** Every module is said to have at most one default export. In order to import the default export from a file, we can use only the address and use the keyword import before it, or we can give a name to the import making the syntax as the following.
  `import GIVEN_NAME from ADDRESS`
- **Importing named values:** Every module can have several named parameters and in order to import one we should use the syntax as follows.

```
import { PARA_NAME } from ADDRESS
```

And similarly for multiple such imports we can use a comma to separate two parameter name within the curly braces.

- **Importing a combination of Default Exports and Named Values:** The title makes it clear what we need to see is that the syntax of the same. In order to import a combination, we should use the following syntax.

```
import GIVEN_NAME, { PARA_NAME, ... } from ADDRESS
```

Exporting

Now, importing is an operation that requires the permission of the module. Importing is possible only if the module or named property to be imported has been exported in its declaration. In React we use the keyword **export** to export a particular module or a named parameter or a combination. Let us now see the different ways we can use the import operation in React

- **Exporting default export:** We have already learned that every module is said to have at most one default export. In order to export the default export from a file, we need to follow the syntax described below.

```
export default GIVEN_NAME
```

- **Exporting named values:** Every module can have several named parameters and in order to export one we should use the syntax as follows.

```
export { PARA_NAME }
```

TUTORIAL 21

React JS Import Export Challenge

Challenge done easily create calculator

TUTORIAL 22

React JS Props

Props are arguments passed into React components.

Props are passed to components via HTML attributes.

React Props

React Props are like function arguments in JavaScript *and* attributes in HTML.

To send props into a component, use the same syntax as HTML attributes:

Giving attributes and call it into more than one element for ex. Create structure of cards and then all cards have different data then call card element as call app.js in index.js and in called element give diff value and it will show in cards more than 1 time..

```
import React from "react";
import Card from "./top-movie-list/Card";
function App() {
  return (
    <>
      <Card
```

```
        links="https://www.google.com"
        title="X-man Hollywood"
        imgsrc="https://upload.wikimedia.org/wikipedia/en/8
/8a/The_Avengers_%282012_film%29_poster.jpg"
        sname="X-man"
      />

      <Card
        links="https://www.google.com"
        title="Avengers Hollywood"
        imgsrc="https://sm.ign.com/t/ign_ap/gallery/e/every
-movi/every-movie-x-man_r3bu.1080.jpg"
        sname="Avengers"
      />

      <Card
        links="https://www.google.com"
        title="Radhe Bollywood"
        imgsrc="https://upload.wikimedia.org/wikipedia/en/5
/5b/Radhe-_Your_Most_Wanted_Bhai_First_Look.jpg"
        sname="Radhe"
      />
    </>
  );
}
export default App;
```

```
import React from "react";

function Card(props) {
  return (
    <>
      <div className="cards">
```

```
        <div className="card">
          <img src={props.imgsrc} alt="myPic" className="ca
rd__img" />
          <div className="card__info">
            <span className="card__category">{props.title}<
/span>
            <h3 className="card__title">{props.sname}</h3>
            <a href={props.links} target="_blank">
              <button>Watch Now</button>
            </a>
          </div>
        </div>
      </div>
    </>
  );
}

export default Card;
```

TUTORIAL 23

Array In React JS

```
const Sdata = [
  {
    links: "https://www.google.com",
    title: "X-man Hollywood",
    imgsrc:
      "https://upload.wikimedia.org/wikipedia/en/8/8a/The_Avengers_%282012_film%29
_poster.jpg",
    sname: "X-man",
  },
  {
    links: "https://www.google.com",
    title: "Avengers Hollywood",
    imgsrc:
      "https://sm.ign.com/t/ign_ap/gallery/e/every-movi/every-movie-x-
man_r3bu.1080.jpg",
    sname: "Avengers",
  },
  {
    links: "https://www.google.com",
    title: "Radhe Bollywood",
    imgsrc:
      "https://upload.wikimedia.org/wikipedia/en/5/5b/Radhe-
_Your_Most_Wanted_Bhai_First_Look.jpg",
    sname: "Radhe",
  },
  {
    links: "https://www.google.com",
    title: "42 Hollywood",
    imgsrc:
      "https://i.pinimg.com/originals/6c/88/64/6c8864a83846fc440ed4998ae165699e.jp
g",
    sname: "42",
  },
  {
    links: "https://www.google.com",
    title: "Hollywood",
    imgsrc:
      "https://image.tmdb.org/t/p/original/au1ktwgL9QtW5jwAqsd4Z2ykUfX.jpg",
    sname: "Bullet Of The Head",
  },
];
export default Sdata;
```

```
 <Card
```

```
        links={Sdata[0].links}
        title={Sdata[0].title}
        imgsrc={Sdata[0].imgsrc}
        sname={Sdata[0].sname}
      />

      <Card
        links={Sdata[1].links}
        title={Sdata[1].title}
        imgsrc={Sdata[1].imgsrc}
        sname={Sdata[1].sname}
      />
```

TUTORIAL 23

Map Method in Array

The Map() method create a new array with the result calling a function for every array element.
The map() method calls the provided function once for each element in an array, in order.
Syntax:-
Array.map(function(currentValue, index , arr), thisValue);
Argument description
currentValue requires the value of current variable
index Optional the array of the current element
arr Optional the array object the current element belongs to.

```
const newcall = ["shiva", "sankar", "bholenath"];
const newarr = newcall.map(function (cvalue) {
  return cvalue;
});
console.log(newarr);
     // const newcall = ["shiva", "sankar", "bholenath"];


// const newarr = newcall.map(function (cvalue) {
//   return cvalue;
// });
// console.log(newarr);

// const Sdatamap = Sdata.map(function (cvalue) {
//   return cvalue;
// });

// console.log(Sdatamap);
```

TUTORIAL 24

Map Method in Array and fetal function

```
function ncard(val) {
  return (
    <Card
```

```
      imgsrc={val.imgsrc}
      title={val.title}
      sname={val.sname}
      links={val.links}
    />
  );
}
function App() {
  return (
    <>
      <h1 className="heading_style">List Of Top Netflix Series
Of 2020</h1>
      {Sdata.map(ncard)}
    </>
  );
}
```

You just need to use map function in which all array will display at once no need to give index no just one time use card element and then it display from array.

On app component we need to set only filename.map and then function name then after we can create that function which we will introduce before and in that function we only need index and put card component there
And we can also put in another way

```
{Sdata.map(function ncard(val) {
        return (
          <Card
            imgsrc={val.imgsrc}
            title={val.title}
            sname={val.sname}
            links={val.links}
          />
        );
```

=>(this symbole is define as fat arrow function)

Normal function:-

```
Function fun_name(param){
    Statement with return type
}
```

Fat arrow function:-

```
Const fun_name = (param)=>{
    Statement with return type
}
```

In normal function if we are forgot to put bracket or return type then that function throw error and in fat arrow function if we know that we only want 1 statement return then we don't need to put return type we can understand this better using example

```
Function add(a,b){
    Return (a + b);
}

Const Add = (a,b) => a+b;
```

Live example: arrow function

```
{Sdata.map((val) => {
      return (
        <Card
          imgsrc={val.imgsrc}
          title={val.title}
          sname={val.sname}
          links={val.links}
        />
      );
```

Normal function

```
{Sdata.map(function ncard(val) {
      return (
```

```
          <Card
            imgsrc={val.imgsrc}
            title={val.title}
            sname={val.sname}
            links={val.links}
          />
        );
```

TUTORIAL 25

React Developers Tools Debugging and Error Handling

Download react developer tools from google chrome extension and put key in component otherwise It show error

TUTORIAL 26

If else Statement in React js

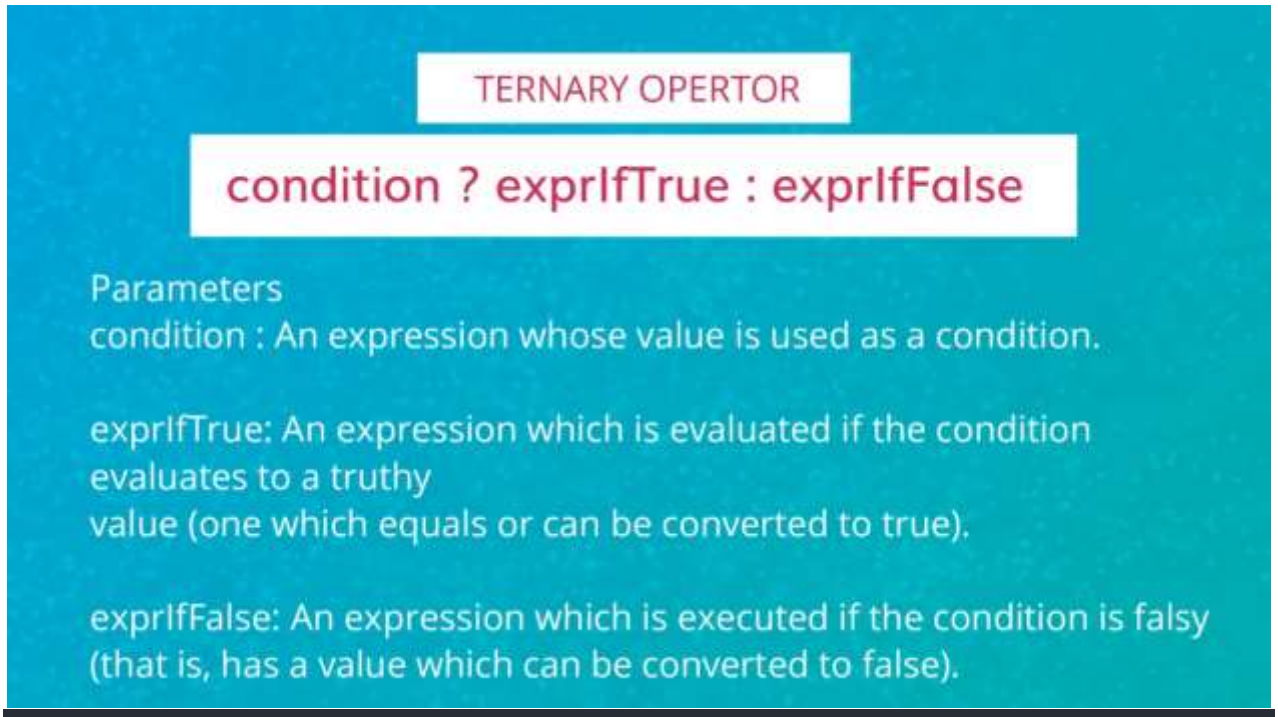
Use if else in showing component on filter like we see in amazon or any ecommerce website so its better to code.

```
const FavSeries = "netflix";

// const Favs = () => {
//   if (FavSeries === "netflix") {
//     return <Netflix />;
//   } else {
//     return <Amazon />;
//   }
// };
import React from "react";
import Sdata from "./top-movie-list/Sdata";
import Card from "./top-movie-list/Card";

const Amazon = () => {
  return (
    <Card
      key={Sdata[2].id}
      imgsrc={Sdata[2].imgsrc}
      title={Sdata[2].title}
      sname={Sdata[2].sname}
      links={Sdata[2].links}
    />
  );
};
```

```
export default Amazon;

import React from "react";
import Sdata from "./top-movie-list/Sdata";
import Card from "./top-movie-list/Card";

const Netflix = () => {
  return (
    <Card
      key={Sdata[1].id}
      imgsrc={Sdata[1].imgsrc}
      title={Sdata[1].title}
      sname={Sdata[1].sname}
      links={Sdata[1].links}
    />
  );
};

export default Netflix;
```

TUTORIAL 27

Ternary Operator in React js




```
{FavSeries === "netflix" ? <Netflix /> : <Amazon />}
```