React JS

What is react? A javascript library for building user interfaces.
Library= Collection of codes
User Interface= Front End

Why react?
React ashse 2013 te. Er age Vanilla JS & jQuery use kore user interface handle kora hoto. jQuery diya e to hossilo tahole React kano?

Index.html: (Normal Code)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>React Tutorial</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div class="container">
      <div>
        <h1 id="display">0</h1>
        <div>
          <button id="button">Increment +</button>
        </div>
      </div>
    </div>
    <script src="./script.js"></script>
  </body>
</html>
```

Script.js:

```
let number = 0;

const display = document.querySelector("#display");
const button = document.querySelector("#button");

button.addEventListener("click", () => {
  number++;
  display.textContent = number;
});
```

Aija kaj javascript diya kora holo problem kothi?

Problem holo eta larger application er khatre scaleable na. Ai khane ekta matro counter ase tai kono problem hoche na. Kintu onek gula counter hole problem ase.

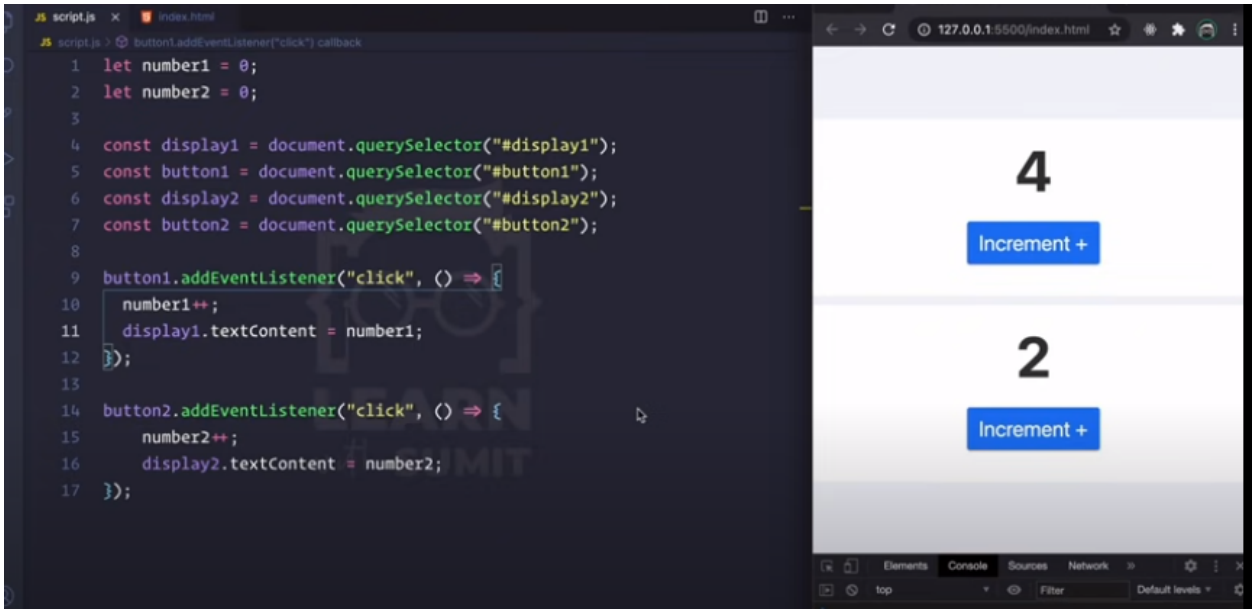


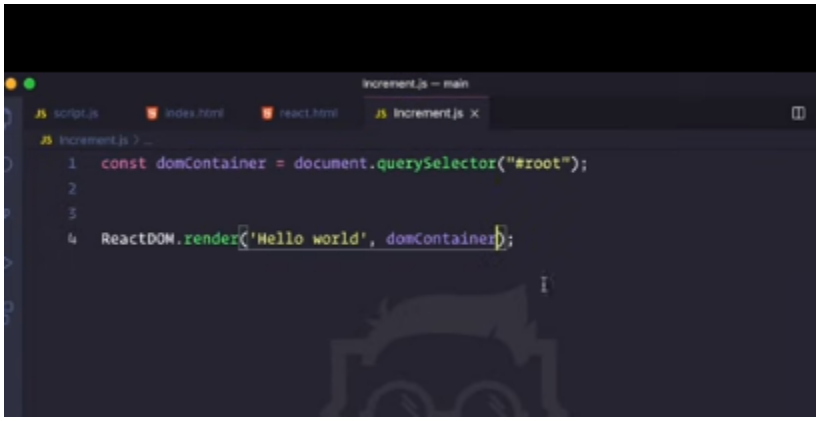
Fig: 2 ta counter howate code onek bare gelo.

React er prototype prothom toiri kore fb er ekjon Engineer Jordan Walke and name den fax js. Normally use react:

React.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>React Tutorial</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div id="root"></div>

    <!-- Load React -->
    <script
      src="https://unpkg.com/react@17/umd/react.development.js"
      crossorigin
    ></script>

    <script
      src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"
      crossorigin
    ></script>
```

```
    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
    <script type="text/babel" src="./Increment.js"></script>
  </body>
</html>
```



Output: Hello world

React library amder k html generate kore dibe. React dom library ta amder dom e render kore dibe. Root er modda generate & render korbe.

React element create:

```
const domContainer = document.querySelector('#root');

const myElement = React.createElement('div', null, [
    React.createElement("p", null, "Hello world"),
])
ReactDOM.render(myElement, domContainer);
```

JSX = javascript XML

Markup syntax use kore element create:

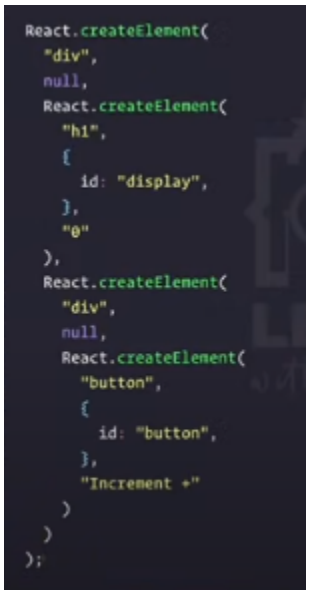
```
let p = document.createElement("p");
p.innerHTML = "Hello world";
domContainer.appendChild(p);
```

JSX diya element create:

```
const domContainer = document.querySelector('#root');

const myElement = (
    <div>
        <h1 id='display'>0</h1>
        <div>
            <button id="button">Increment+</button>
        </div>
```

```
    </div>
);
ReactDOM.render(myElement, domContainer);
```

Uporer code tukor uqivelent:



Ai vabe amder likte hose na aita e valo kotha. Amra likbo JSX. Akhon JS to r JSX buje na. So ai jonno amder use korte hosse babel. Babel JSX k vanilla js convert kore.

```
const domContainer = document.querySelector('#root');
const Increment = () => {
    return (
        <div>
            <h1 id='display'>0</h1>
        <div>
            <button id="button">Increment</button>
          </div>
        </div>
    );
};
ReactDOM.render(<Increment/>, domContainer)
```

/OR

```
ReactDOM.render(Increment(), domContainer)
```

JSX a / (slash) dita e call kora jai.
React er jQuery use kora ekta chorom vule.
State use kore code korbo easy way te:

```
const domContainer = document.querySelector("#root");
```

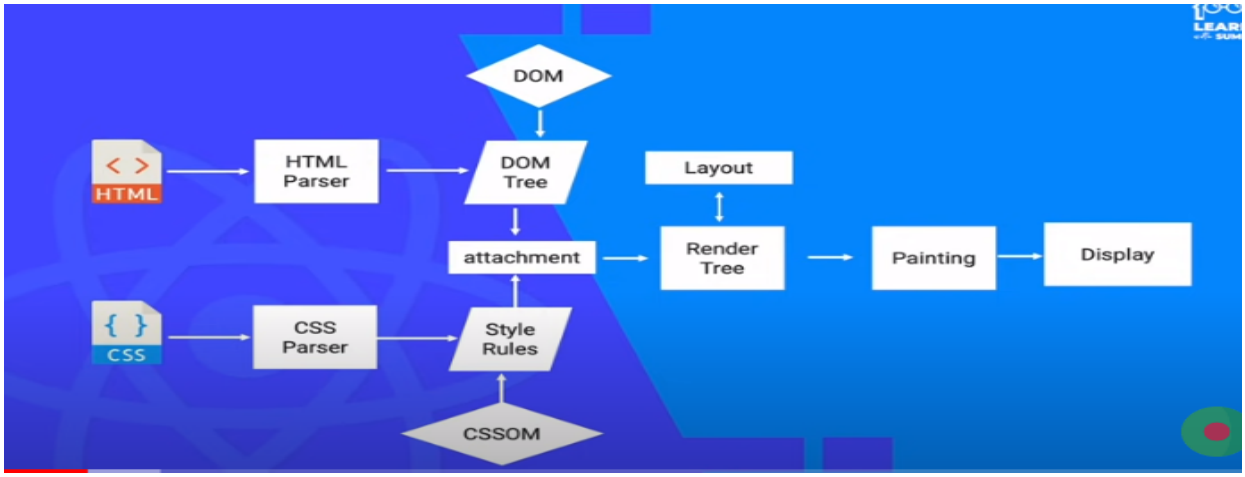
```
const Increment = () => {
  const [counter, setCounter] = React.useState(0);
  return (
    <div>
      <h1 id="display">{ counter }</h1>
      <div>
        <button id="button" onClick={ () => setCounter(counter + 1) }>Increment
+</button>
      </div>
    </div>
  );
};

ReactDOM.render(
  <div className="container">
    <Increment />
  </div>,
  domContainer
);
```

React Virtual DOM

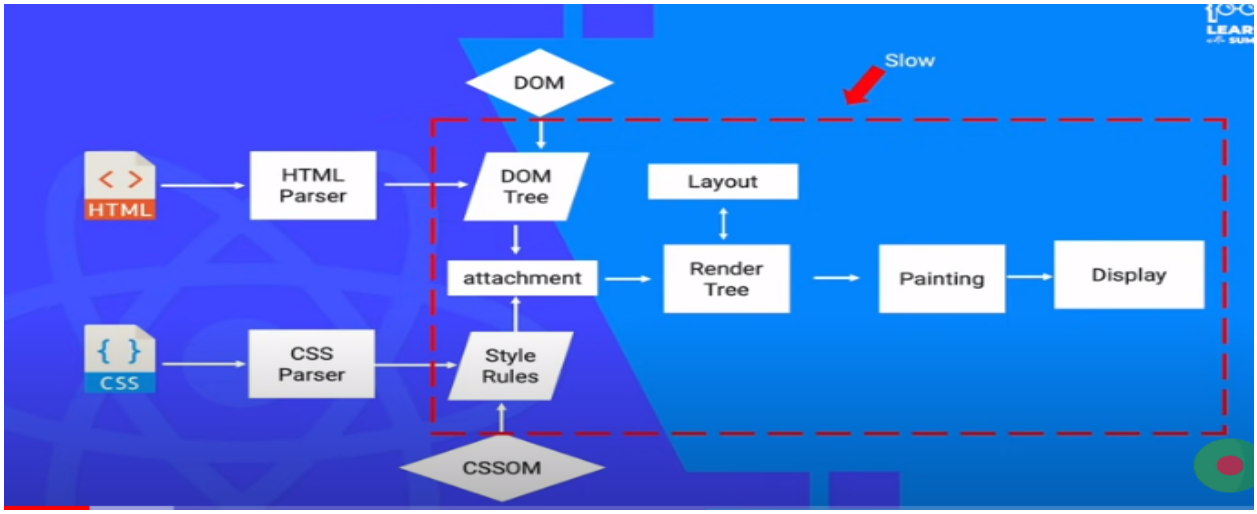
Ekta kotha procholito ase Browser dom(Document object model) slow ai kothar kono jukti nai. React virtual dom shompoke bola hoi aita DOM er cheya efficient way te web application er view render korte pare.

DOM কি আসলেই slow? Dom যথেষ্ট fast. মূলত Dom change হওয়ার পর আমাদের চখের সামনে দেখানর জন্য Browser কে যে কাজটা করতে হয় সেটা Slow.

Browser Work Flow:



Slow Part:



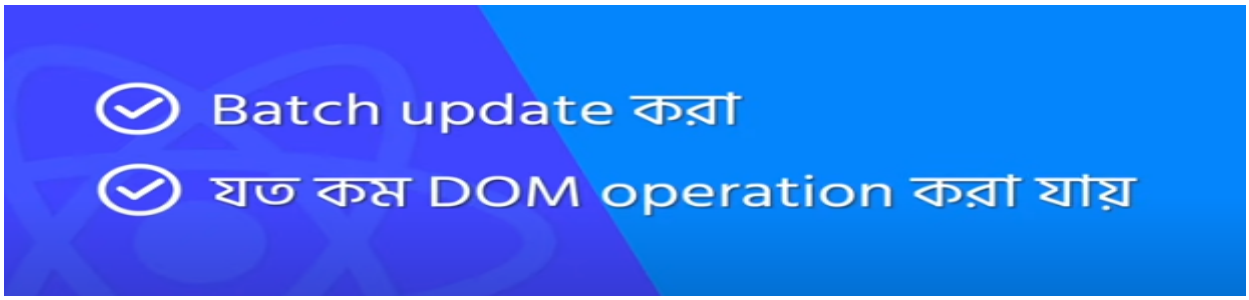
যখনই আমরা DOM এ কোন change or update করি Browser কে এই Rander Tree আবার Update করতে হয়। এবং paint করতে হয়। এই কাজটাই Slow.

Browser maker ra continuously Brower gulo ke update kore efficient banate chashta kore jasse jeno ai re painting process ta faster hoi. Abong tara korcen o aita. Modern brower a amon shoto shoto repainting baper e na. Single page application বেশি জনপ্রিয়. User experience er jonno page reload kora ta r kew e like kore na. Tai 1 ta page amader onek onek dom operation handle korte hoi.

Facebook এর কথাই যদি চিন্তা করি এক জায়গায় click করলে কত জায়গায় change হচ্ছে। মানে কত বেশি DOM operation সামাল দিতে হচ্ছে Browser কে।

So এটা performance কে এক সময় effect করবে। তাহলে DOM ই যদি change করি maximum কি করার আছে আমাদের?

So আমরা ২ টা কাজ করে maximum optimized করতে পারি।

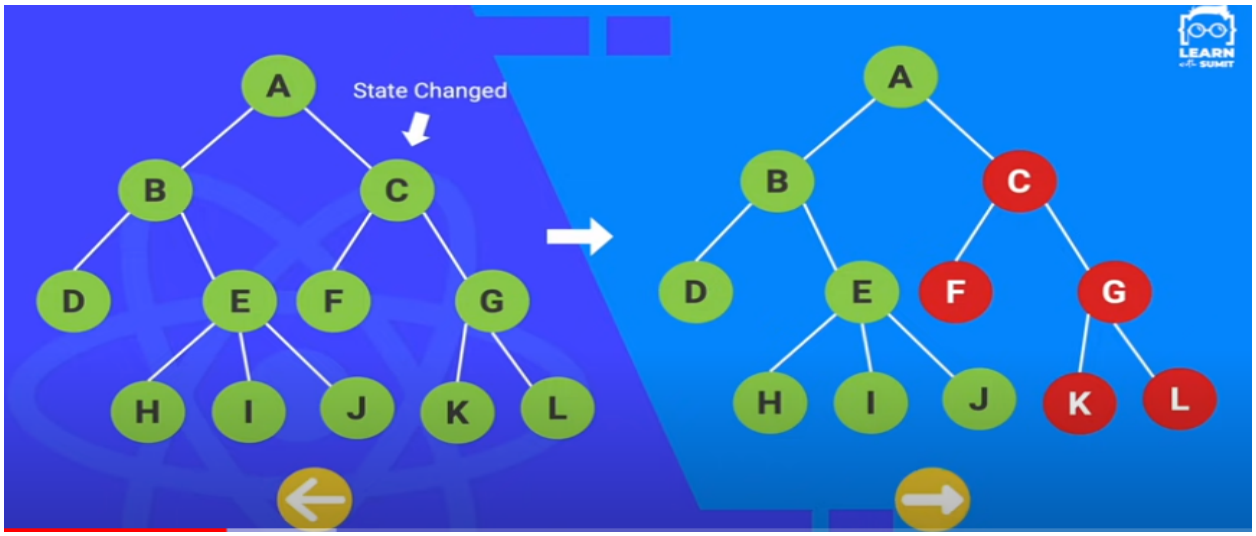


Batch update:
Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="container"></div>
  <script src="dom.js"></script>
</body>
</html>
```

Batch update holo shob kisu batch kore ekta dom operation kore kaj ta sesh kore fellam. So repaint process ta ekbar e holo ta hole.
Virtual DOM ছাড়াও batch update করা যায়। DOM direct manipulate করে ও করতে পারি। কিন্তু exectly যে element ta change হয়েছে শুধু সেটাকে update করার এই beauty টাকে achieve করার জন্য React কে update করার আগের অবস্থা এবং পরের অবস্থা ২ টার separate snapshort রাখতে হবে। যেন সে ২ টার মধ্যে compare করতে পারে। React সেই কাজ টাই করে। কিন্তু DOM এর মধ্যে থেকে সেটা করা problematic. এ জন্য React আলাদা একটা DOM বানিয়ে নিয়েছে। যেখানে repaint er জামেলা নাই। তাকে browser এর নিয়ম মানতে হবে না। যেখানে সে Javascript object নিয়ে কাজ করে।

Virtual DOM কি ভাবে কাজ করে?
Virtual DOM কে আমরা একটা simple tree হিসেবে কল্পনা করতে পারি। যখনই আমরা কোন Data Change করি fast এ নতুন একটা Tree create হয়। যেখানে তার change component & child component আবার নতুন করে create হয়।



React simply 2 ta oboshta ke compare kore. একটা Efficient Algorithm এর মাধ্যমে decide করে কোন কোন জিনিস change হয়েছে। এই Algorithm কে Reconciliation or Diffing Algorithm বলা হয়।
Virtual DOM slow o na fast o na she fast enough. Shudu performance er jonno amra React use kori na. React is not only popular for it's Performance only. React is famous for amazing Developer Experience.
Performance depend kore bibinno factor er upor. Use case er upor & Application er upor. আমি শুধু state change করে দিব বাকি কাজ React ই করবে।

Ekta component change hole tar under a joto child ase shob e abr rerender hoi. So Ami chile shob rerender na o korte pari UseMemo(), shouldComponentUpdate() use kore.

React Environment Setup

Node, npm install আছে কিনা দেখব। Verson check: node –v, npm –v.

Update version na thakle abr download kore install korle e hobe.

Globally yean install korbo: npm i –g yarn

Create-react-app install kora ase kina dekbo Create-react-app type kore. Jodi thake tahole remove kore dibo: *yarn global remove create-react-*app

React app create: npx create-react-app myreact

Editor Setup

You can use any editor but as I personally prefer VS Code. I will give some instructions about how I prefer VS code to be setup for React applications.

Plugins

You need to install the below plugins:

- ESLint by Dirk Baeumer
- Prettier - Code formatter by Prettier
- Dracula Official Theme (optional)

Settings

Follow the below settings for VS Code -

1. Create a new folder called ".vscode" inside the project root folder
2. Create a new file called "settings.json" inside that folder.
3. Paste the below json in the newly created settings.json file and save the file.

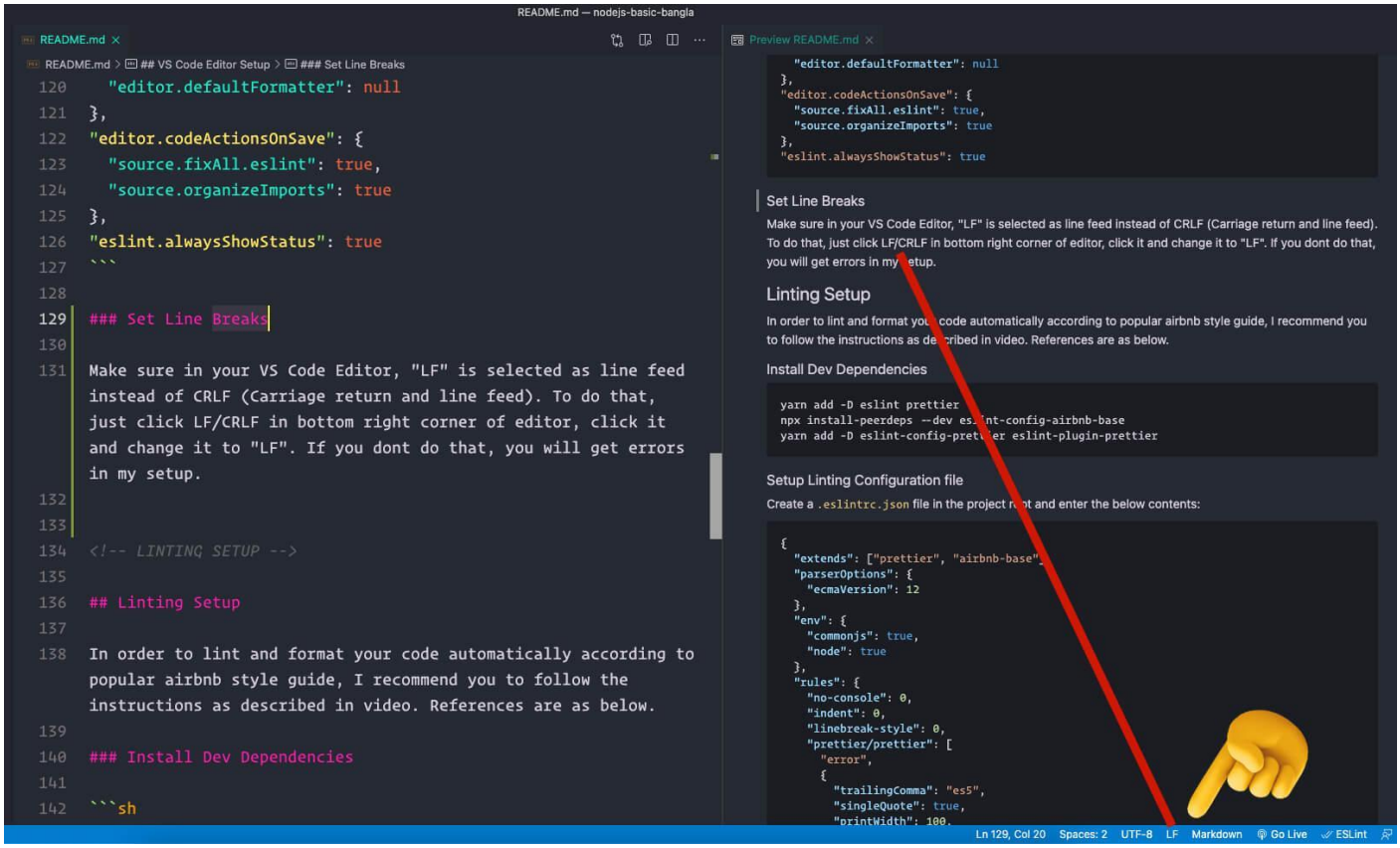
```
{
  // Theme
  "workbench.colorTheme": "Dracula",


  // config related to code formatting
  "editor.defaultFormatter": "esbenp.prettier-vscode",
  "editor.formatOnSave": true,
  "[javascript]": {
    "editor.formatOnSave": false,
    "editor.defaultFormatter": null
  },
  "[javascriptreact]": {
```

```
    "editor.formatOnSave": false,
    "editor.defaultFormatter": null
  },
  "javascript.validate.enable": false, //disable all built-in syntax checking
  "editor.codeActionsOnSave": {
    "source.fixAll.eslint": true,
    "source.fixAll.tslint": true,
    "source.organizeImports": true
  },
  "eslint.alwaysShowStatus": true,
  // emmet
  "emmet.triggerExpansionOnTab": true,
  "emmet.includeLanguages": {
    "javascript": "javascriptreact"
  }
```

} If you followed all previous steps, the theme should change and your editor should be ready.

Set Line Breaks

Make sure in your VS Code Editor, "LF" is selected as line feed instead of CRLF (Carriage return and line feed). To do that, just click LF/CRLF in bottom right corner of editor, click it and change it to "LF". If you dont do that, you will get errors in my setup.



Linting Setup

In order to lint and format your React project automatically according to popular airbnb style guide, I recommend you to follow the instructions below.

Install Dev Dependencies

```
yarn add -D prettier
yarn add -D babel-eslint
npx install-peerdeps --dev eslint-config-airbnb
yarn add -D eslint-config-prettier eslint-plugin-prettier
```

or You can also add a new script in the scripts section like below to install everything with a single command:

```
scripts: {
    "lint": "yarn add -D prettier && yarn add -D babel-eslint && npx install-peerdeps
--dev eslint-config-airbnb && yarn add -D eslint-config-prettier
eslint-plugin-prettier"
}
```

and then simply run the below command in the terminal –

```
yarn lint #or 'npm run lint'
```

Create Linting Configuration file manually

Create a `.eslintrc` file in the project root and enter the below contents:

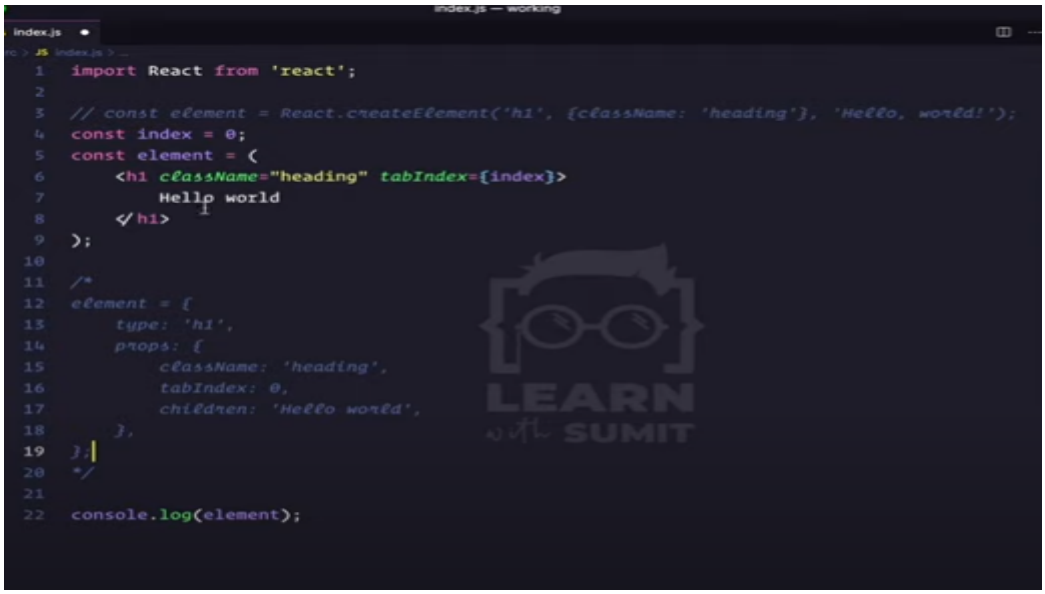
```
{
  "extends": [
    "airbnb",
    "airbnb/hooks",
    "eslint:recommended",
    "prettier",
    "plugin:jsx-a11y/recommended"
  ],
  "parser": "babel-eslint",
  "parserOptions": {
    "ecmaVersion": 8
  },
  "env": {
    "browser": true,
    "node": true,
    "es6": true,
    "jest": true
  },
  "rules": {
    "react/react-in-jsx-scope": 0,
    "react-hooks/rules-of-hooks": "error",
    "no-console": 0,
    "react/state-in-constructor": 0,
    "indent": 0,
    "linebreak-style": 0,
    "react/prop-types": 0,
    "jsx-a11y/click-events-have-key-events": 0,
    "react/jsx-filename-extension": [
      1,
      {
        "extensions": [".js", ".jsx"]
      }
    ],
    "prettier/prettier": [
      "error",
      {
        "trailingComma": "es5",
        "singleQuote": true,
        "printWidth": 100,
        "tabWidth": 4,
        "semi": true,
        "endOfLine": "auto"
      }
    ]
  },
  "plugins": ["prettier", "react", "react-hooks"]
}
```

Akhon yarn start korle error dekhabe karon react er o eslint package ase & ami o eslint package use korsi & 2 tar version binno. (Ai problem ta hoi nai akhon amr)

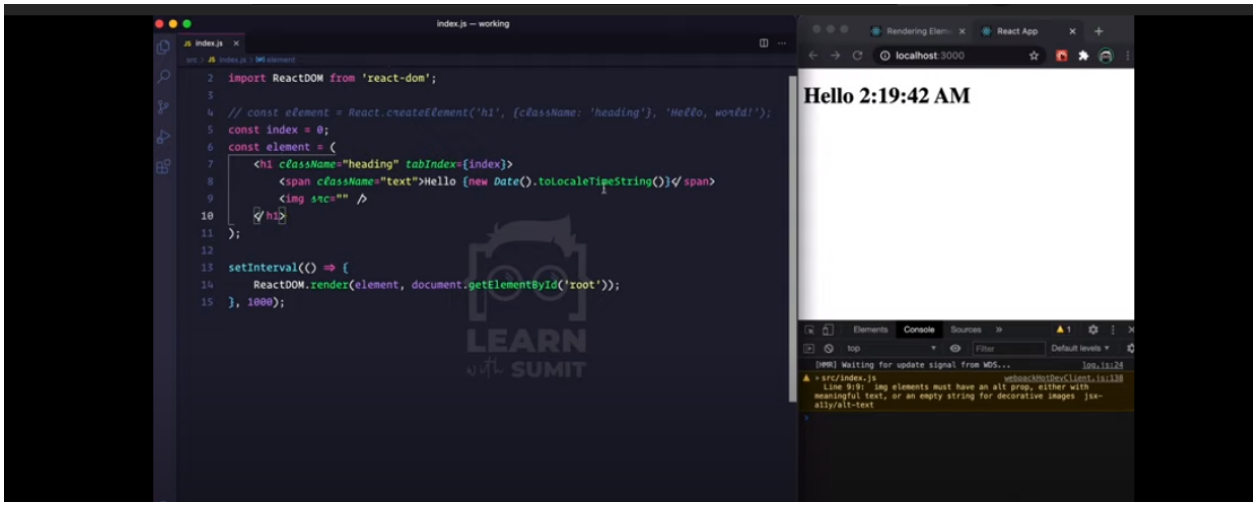
React element hose ekta valid js object.

Index.js

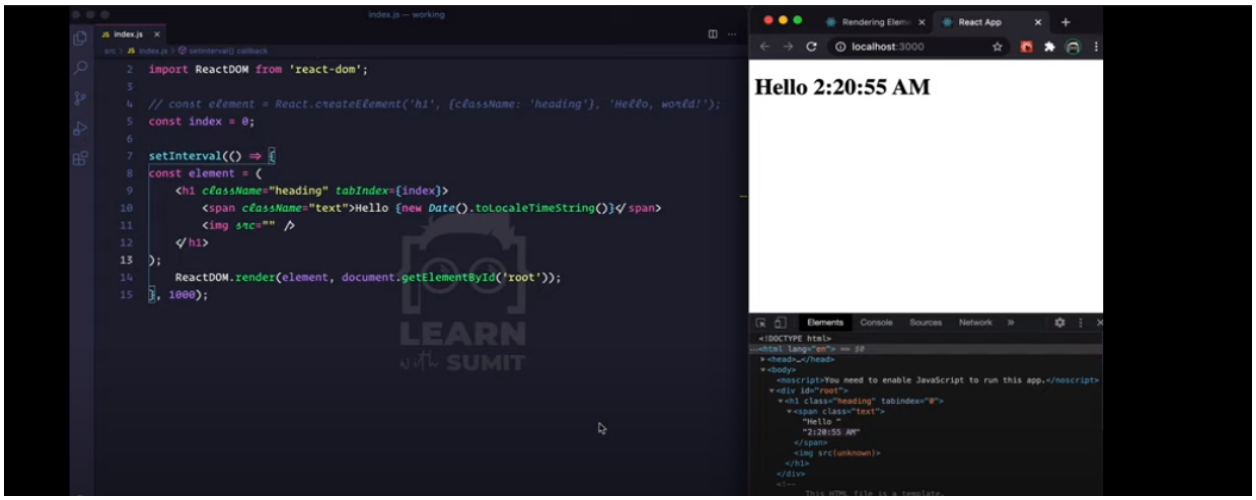
```
import React from 'react';

const element = React.createElement('h1', null, 'Hello world');
console.log(element);

/* babel convert kore banase:
{
    type:'h1',
    props: {
        children: 'Hello world'
    }
}
*/
```



Index.js



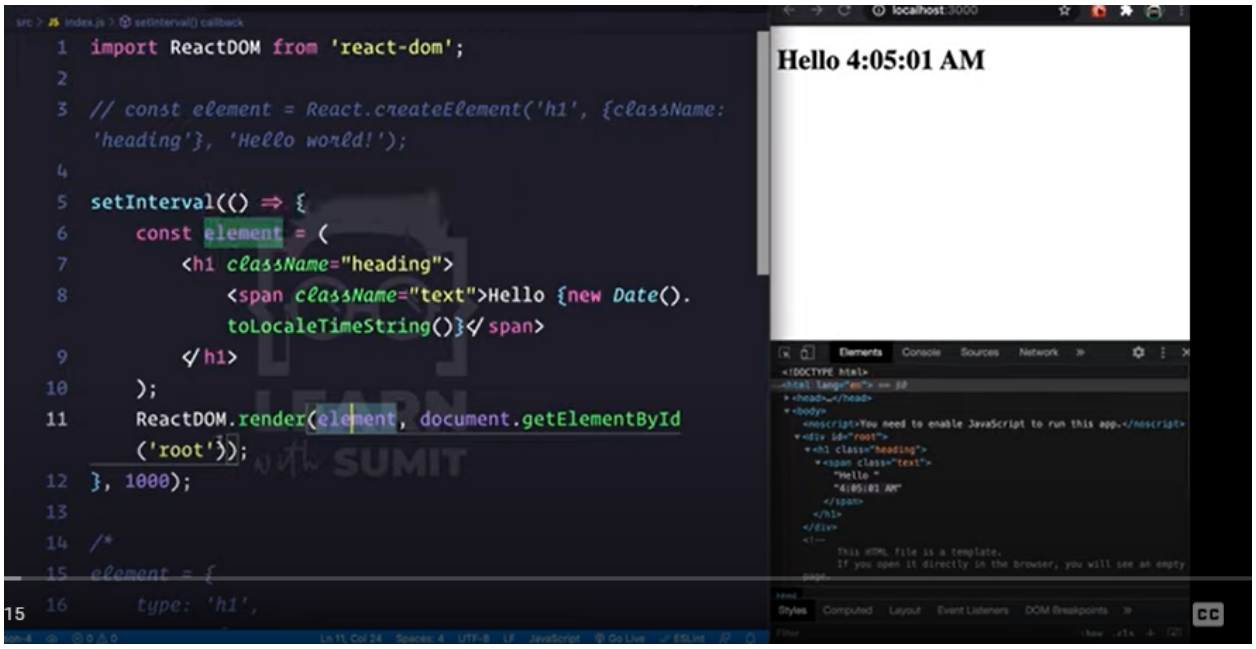
Ai gori ta update hose na karon element ta global variable.



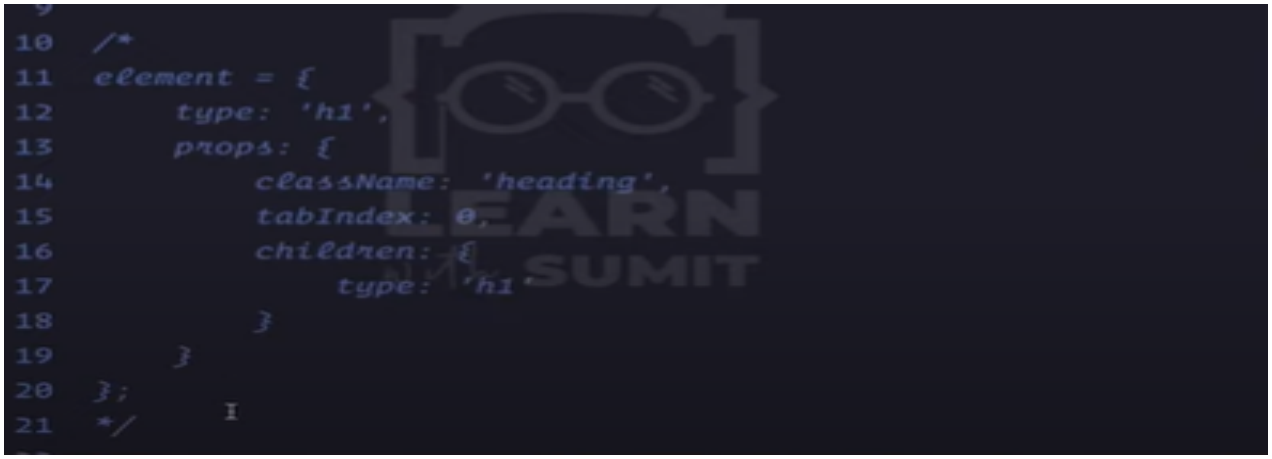
Akhon gori ta kaj korse.

React component & props

React er element hosse react er core building block. React shudu matro tar element kei chinte pare. Normally Browser a amra jokhon kono html elements likhi like div tokhon amder browser she take parse kore tar modda je div & onnano element gula ase shob gulo k niya ekta tree er moto structure toiri kore jetake DOM bola hoi. DOM kono valid js object na. Sheta node type er ekta jinish. Sheta k niya amra js er maddome manipulate korte pari & bibino functional kaj korte pari. React er element gulo html element er motoi but kono vabei html element na. React er element hosse valid js object. React hosse ekta vanilla js library. React er element k amra change korte pari na.



Object representation:



Ekhane element take je change korte hobe amra bole dissi manually setInterval diya. Akhane 1 sec por por element call hosse. Bar bar call hoi react dom er kase notun element jasse then ager element & porer element compare korse, compare kore je jinish ta change hosse sheta shudu

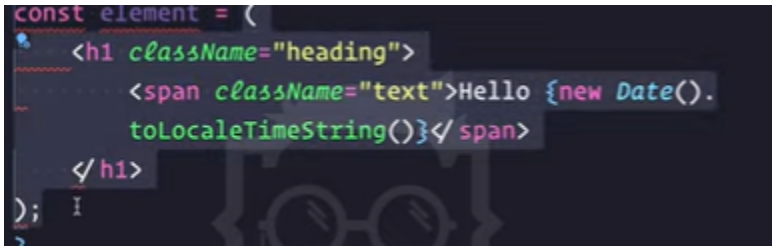
dome change kore disse. Ata hosse React er element er core concept. Element k amra change korte pari na. Change korer jonno React amder kono kisu dai ni. Aita hosse fundamental concept React er.

Ajk amra janbo component. **Component kano ashlo?** Karon Uporer je element ta sheta kono functional element na. Ai element ta just dump ekta jinish. Jeta change kora jasse na. Ekta string er moto. Ai element ta just ami dom er kase pathiya dissi. Ai rokom dump element diya to amder kono kaj nei. Amader functional boishishto wala kisu dorkar. So shei functional boishishto wala jinish tai hosse component.

```
import ReactDOM from 'react-dom';

const element = (
  <h1 className='heading'>
    <span className='text'>Hello {new
Date().toLocaleTimeString()}</span>
  </h1>
);
ReactDOM.render(element,
document.getElementById('root'));
```

Akhon clock er change hobe na. Karon ekta e render kora hoiyase. Er change kora hosse na.

Akhon ei element take functional banate amder ki kora uchit? Ekta simple vanilla js function likte pari.



Ai ja first bracket er bitor ja ase she gulo ek ekta balid js syntax. React element are valid javascript Expression. Tar mane element take kono function theke shetake return o kora jabe abr kono function sheta e parameter hishabe pate o pare.

```
import ReactDOM from 'react-dom';

function Clock() {
  return (
    <h1 className="heading">
      <span className="text">Hello {new
Date().toLocaleTimeString()}</span>
    </h1>
  );
}

ReactDOM.render(Clock(),
document.getElementById('root'));
```

So Ai Clock function theke element take return kore dilam & render a call kore dilam. So amra ki korlam? Component toiri kore fellam. Karon ai function take e bola hoi react er component. Component and element ek na kintu. Ai khane ai pura function ta component & jeta return korse sheta element.

React component return React Element. One React Component return one single React Element.

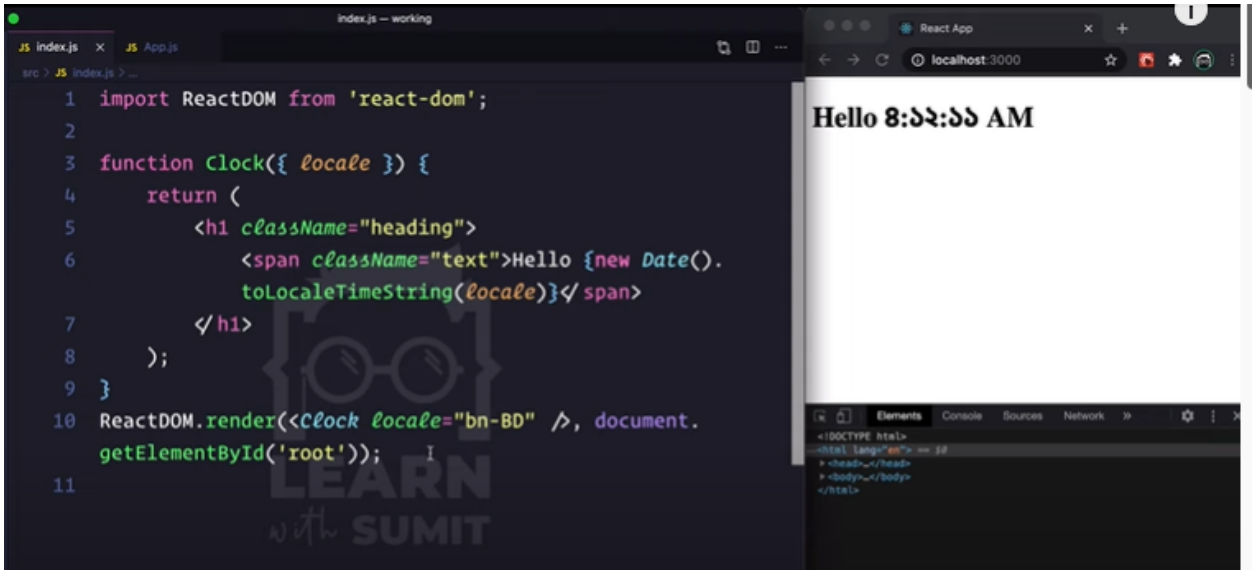
```
ReactDOM.render(<Clock/>,
document.getElementById('root'));
```

Amra jahutu react use korsi So React akhon dekbe ai Clock name upore kisu ase ki na. Jokhon pabe shetake call kore dibe. And shate ultimately ekta React element return korse.

, <img/> html er ai element gula ka amra bar bar re-use korte pari. So akhon je hutu amra ekta functional component banai fellam & (<Clock/>)

Ai vabe liklam so amra akhon aita k bar bar reuse korte pari chaile e.

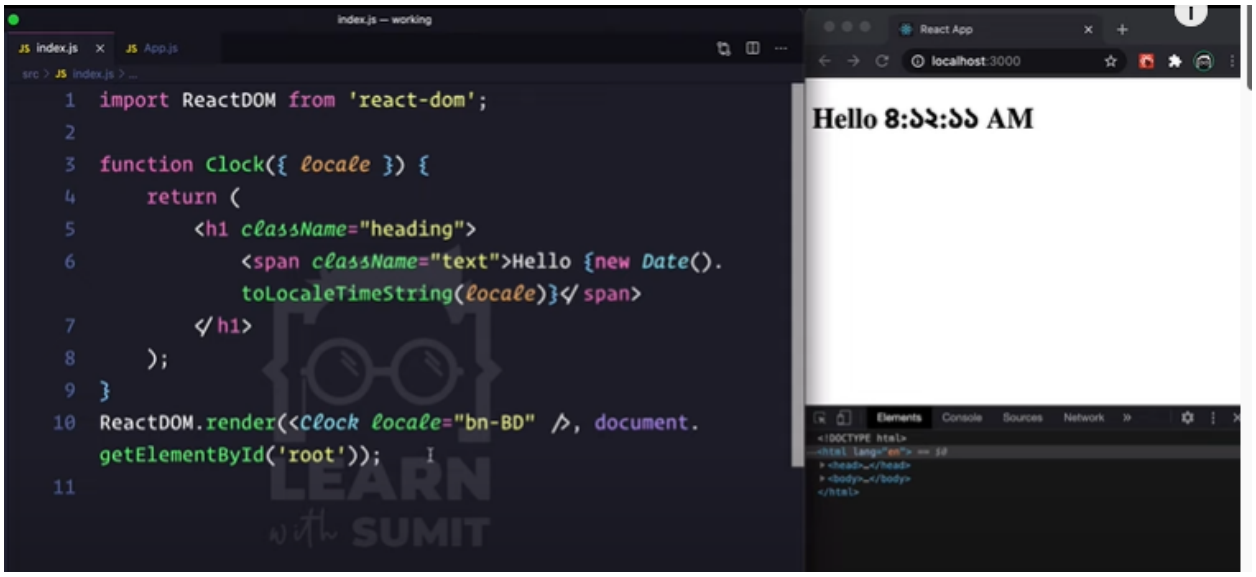
We can re-use React Component. Jokhon element chilo tokhon take baire theke kono element dite partam na. So jahatu eta akhon ekta valid js function amra parameter dite parbo.



Akhon clock function call korsi with parameter & ai parameter take function ta object akare pabe & function ta object akare distructure kore nisse. Akhon gori ta ami banglai pabo.

Component receive properties from outside. Babel will transpile this to valid Javascript.

Functional Component chara o React a Component bananor jonno er ekta syntax ase sheta holo class syntex. Class hose ekta statefull component.



Ai khane ai gori ta kintu change hosse na. Gori ta je 1 sec por por change hobe change take listen korer moto kew nai. Tar mane amk setInterval kore bole dite hobe kokhon she change hobe. But react shai purpose toiri hoi nai.

Akhon ai change take listen korte class use korte pari.

Amra kintu jani class k call kora jai na.

Valid js Way:

```
import ReactDOM from 'react-dom';

class Clock {
    print() {
        return (
            <h1 className='heading'>
                <span className='text'>Hello {new
Date().toLocaleTimeString()}</span>
            </h1>
        )
    }
}
const ClockComponent = new Clock();//class k call kora jai na class ekta blue
print tai ai object ta baniya nilam
ReatDOM.render(ClockComponent.print(), document.getElementById('root'));
```

```
ReatDOM.render(<ClockComponent/>, document.getElementById('root'));
```

Ai vabe aita kaj korbe na. Karon ai khane shudu she function call aita bujte pare. Class k buje na ai vabe. Class likle react amder ekta statefull বৈশিষ্ট দিবে। And state toiri korte and change korte parbe.

```
import React from 'react';
import ReactDOM from 'react-dom'
class Clock extends React.Component {
    render() {
        return (
            <h1 className='heading'>
                <span className='text'>Hello {new
Date().toLocaleTimeString(this.props.locale)}</span>
            </h1>
        )
    }
}

ReactDOM.render(<Clock locale='bn-BD'/>, document.getElementById('root'));
```

Function k react tar component akare chinte pare Tamon Class ke o chinate react.component extends kore nibo. React.component er maddome ja ja ase shob inheret kore niya ashlam.

Class er render name ekta method thake. React automatically calls this render method. Class er method & property thake.

Class component a amra jokhon kono parameter pass kori tokhon React shetake react.component er modda je props name property ase tar modda dukiya dai. So kono class er property jokhon accept korte chai tahole this diya call korte hoi. this.props ekta object ai khane. Jamon function a parameter jeto object hishabe. Component ektar modda er ekta thakte pare. Because Component eventually return an Element. Amar koto gula component lagbe ami age theke thik kore nibo. Shudu shudu ojotha component create korbo na. Component ekta element na component ek ek ta application er moto chinta korbo. Borto mane functional component er modda o state control kora jai hook use kore. Hook… Hook kore kore bibbino jaigai duke jai. Class component onek shomoi ektu boro hoiya jai tai shobi functional component e prafer kore. Ami jodi shob class component use kori kono problem nai. But functional component dakte shubida.

Presentational component: je component ta dump mane je component shudu baire theke props tops pathabo tar nijer kono state nai or tamon kono functionality nai jeta shomoi er shate shate change hobe or click korle user er intraction er jonno shate change hobe amon complex jinish Jodi na hoi tar kaj hoi screen a hello print kora tahole shetake bola hoi presentational component.

(Jai object ta react toiri kore disse sheta hosse this.)

Nested Component:

```
import React from 'react';
import ReactDOM from 'react-dom'
class Clock extends React.Component {
    render() {
this.props.locale= ‘hello’ Amon kisu korbo na.
        return (
            <h1 className='heading'>
                <span className='text'>Hello - {this.props.children} {new
Date().toLocaleTimeString(this.props.locale)}</span>
            </h1>
        )
    }
}

ReactDOM.render(<Clock locale='bn-BD'>test</Clock>,
document.getElementById('root'));
```

`locale='bn-BD'`

Never change props inside Component. Ai ja props pathalam aita bitore change korbo na kokhono.

Jokhon e props change hoi react rerender kore. React component will re-render whenever props change. Props shob shomoi baire theke change hobe bitore na. State change hoi bitore. JS ekta functional programming language. React a amra dorkari jinish pass korsi baire theke then component sundor moto encapsulate way te kaj korse . Encapsulated means all functionalities are self-contained.

SO amra amder application ke bibbino component a bag kore felbo & amra component k reuse korte parbo & component er modda er ekta component nested korte parbo, Akta component k bangay tar theke er ekta component nite parbo. Je jinish ta bar bar use hobe she jinish take bangbo.

State and LifeCycle

State is data that change. (State = oboshta)

Props change hobe baire theke. State hosse component er nijasho ekta jinish. Like jijer ekta database. Component it self ekta app. State is a javacript Object.

Class er state k initial ekta value dite hobe database toiri korte chile. Kono class a initialize korte hole constructor function nite hoi. Super just calls the base class constructor. Jokhon e state k ekta particular way te poribotton korbo tokhon e react.. react korbe. And state er value change hole e she render korbe & notun updated state pabe.

componentDidMount() runs after component has been renderd to the DOM.

Aita jahutu single page application page reload hobe na so amra jokhon onno page a jabo ai timer ta kintu cholte e thakbe (setInterval). Reload dile hoito timer ta chole jeto. 1 page e jehutu shob hosse dom theke chole jasse but timer on theke jabe. So amder timer take off kore dewa uchit. So amder jante hobe kokohn aita dom theke chole gese or unmounts hoiyase. Component mount hower pore componentDidMount call hoi. Component unmount hower thik age componentWillUnmount call hoi. Chole jawer age call kore diya chole jai she.

Never ever change props inside component.

```
class Clock extends React.Component {
    constructor(props){
        super(props);
        this.state = {date: new Date()};
    }
```

Ai khane props ta lage nai kaje.

this.state er modda jokhon props ta dorkar or kaje lagate hoi tokhon e shudu constructor props ai vabe anbo.

Clock.js

```
import React from 'react';

class Clock extends React.Component {
    constructor(props){
        super(props);
        this.state = {date: new Date()};
    }

    ComponentDidMount() {
        this.clockTimer = setInterval(() => this.tick(), 1000)
    }
    componentWillUnmount(){
        clearInterval(this.clockTimer)
    }
    tick(){
        this.setState({
            date: new Date(),
        })
    }

    render() {
        return(
            <h1 className='heading'>
                <span
className='text'>{this.state.date.toLocaleTimeString(this.props.locale)}</span>
            </h1>
        )
    }
}
export default Clock;
```

Short Cart way the korle:

Clock.js

```
import React from 'react';

class Clock extends React.Component {
  state = { date: new Date()};

    componentDidMount() {
        this.clockTimer = setInterval(() => this.tick(), 1000)
    }
    componentWillUnmount(){
        clearInterval(this.clockTimer)
    }
```

```
    tick(){
        this.setState({
            date: new Date(),
        })
    }

    render() {
        return(
            <h1 className='heading'>
                <span
className='text'>{this.state.date.toLocaleTimeString(this.props.locale)}</span>
            </h1>
        )
    }
}
export default Clock;
```

Always make a copy of Reference values. React has One way data flow. Also called top down or uni-directional flow.

Handling Events & Control Re-render

React events are named using camelCase, rather than lowercase.

```
    handleClick(e) {
        e.preventDefault();
        console.log('The button was clicked')
    }
```

Page reload na hower code.

This:

```
<script>
    const javascript = {
        name:"Javascript",
        printLibraries: ['react', 'vue'],
        printLibraries: function(){
            console.log(this)
        this.libraries.forEach(function(library){
            console.log(this)
            console.log((`${this.name} loves ${library}`));// Ai khane this.name
ta passe na karon normal function tai ai this k chinte parse na. Bairer this & ai
this same na.
        })

        }
    }
</script>
```

Arraw function this er confusion clear kore dai.

```
<Button change = {this.handleClick.bind(this, 'en-US')}></Button>
```

this k bind kore dilam & parameter o pathalam. Arrow function na use korle aita kora jete pare. Aita 1 ta way. Normally function er modda ai vabe amra parameter diya pathabo na for best practice.

Controller re-render: 2, 1 component Jodi bar bar re-render hoi voi er kisu nai. Kono component Re-render hobe ki hobe na aita amra shouldComponentUpdate use kore bole dibo.

```
shouldComponentUpdate()
```

shouldComponentUpdate accept kore nextProps & nextSate.

Jekhane shekhane shouldComponentUpdate use korbo na. Component Jodi medium size er o hoi shouldComponent use na kora e valo.

Button.js

```
import React from 'react';
class Button extends React.Component {
   shouldComponentUpdate(nextProps){
       const {change: currentChange} = this.props;
       const {change: nextChange} = nextProps;
       if(currentChange === nextChange){
           return false;
       }
       return true;
   }
    render() {
        const {change} = this.props;
        return (
            <button type='button' onClick={change}>Click Here</button>
        )
    }
}
export default Button;
```

React Conditional Rendering, List and Keys

```
render() {

    }
```

Render k amder call korte hoi React e render k call kore.

App.js

```
import ClockList from './component/ClockList'
```

```
function App() {
    const quantities = [1,2,3]
    return (
        <div>
            <ClockList quentity = {quantities}/>
        </div>
    );
}

export default App;
```

ClockList.js

```
import Clock from "./Clock";
function ClockList ({quantity = []}){
    return (

    )
}
```

byDefault value blank array diya dilam user Jodi value na dai error jeno na ashe.

```
import Clock from "./Clock";
function ClockList ({quantity = []}){
    return (
       <div>

            {quantities.map(() => (
                <Clock key={}/>
            ))}
        </div>

    )
}
```

Amon List populate korbo unique key dite hobe.

React Form Handling –Controlled VS Uncontrolled

Form.js

```
import React from 'react';
export default class Form extends React.Component {
    render(){
        return(
            <div>
                <form>
```

```
                    <input type='text' placeholder='Enter title'
value='Javascript'/>
                </form>
            </div>
        )
    }
}
```

Ai vabe from a value diya dele bujbe take control dissi. Value na dile react bujbe amr programmer amk dite chasse na mane dom diya control korte chasse.

Controll way tai best.

Uncontrol unit mane DOM operation. Amader normal je html form shetai uncontrol component react er khatre. Jar uppor react er kono control nai. Aita amr daietto dom je je vabe control kore she vabe e control korbo. Control bolte bujjase ja component ami banaci shai component er state er uppor…control component/control input

React Conditional Rendering, List and Keys

Click.js

```
render() {
        const { date, locale } = this.state;
        let button;
        if(locale === 'bn-BD'){
        button = (
            <Button change = {this.handleClick.bind(this, 'en-US')}>Click
Here</Button>
        )

        }else{
            button = (
                <Button change = {this.handleClick} locale="bn-BD">Click
Here</Button>
            )
        }
        return (
            <div>
            <h1 className="heading">
                <span className="text">{date.toLocaleTimeString(locale)}</span>
            </h1>
            <button onClick={this.handleClick}>Click here</button>
            </div>
        );
    }
}
```

Conditional rendering: ekta component render hobe ki hobe na sheta jodi shai component er biton theke e control korte chai tar koita upai er modda ekta holo vanilla js er concept diya.

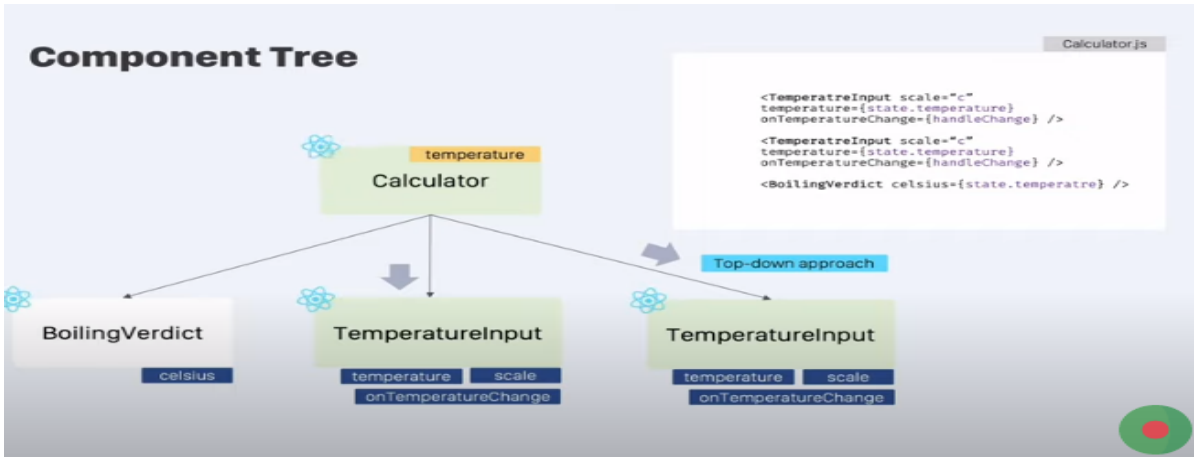
For List & Key:

```
<div>
    {quantities.map((key) => (
        <Clock key={key}/>
//or
        <Clock key={Math.random()}/>
    ))}
</div>
```

Key na thakle map er index use korte pari but index use kora uchit na tai amra Math.random use korbo. Ekta particular array er modda unique hole e hobe globaly unique hote na.

Lifting State UP

Ekta input area diya amra 2 ta input nibo tai baire ekta object baniya nibo.



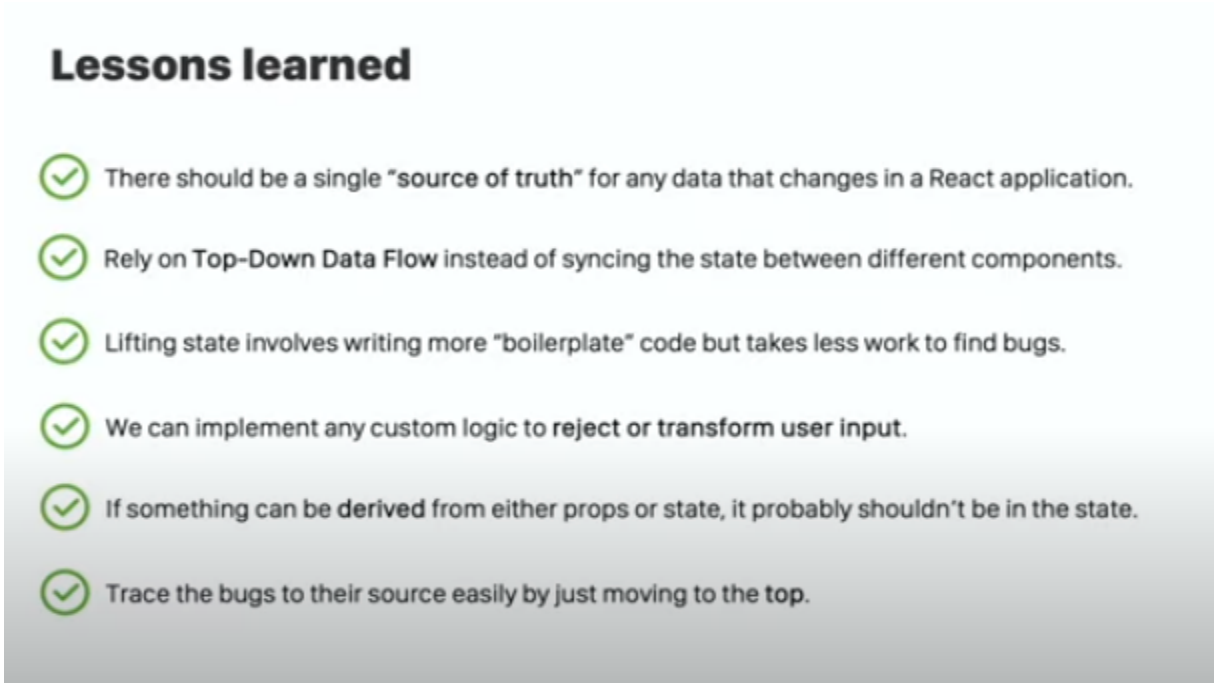
TemperatueInput.js

```
import React from 'react';
const scaleNames = {
   c: 'Celsius',
   f: 'Fahrenheit'
}
export default function temperatureInput({temperature, scale, onTemperature}){
    <fieldset>
    <legend>Enter temperature in {scaleNames[scale]}:</legend>
    <input type='text' value={temperature} onChange={onTempareture}></input>
    </fieldset>
}
```

Calculator.js

```
import React from 'react';
```

```
import temperatureInput from './Calculator';

export default class Calculator extends React.Component {
    state = {temperature: '', scale: 'c'};

    handleChange = (e, scale) => {
        this.setState({
            temperature: e.target.value,
            scale,
        })
    }
    render() {
        return(
            <div>
                <temperatureInput scale='c'/>
                <temperatureInput scale='f'/>
            </div>
        )
    }
}
```



React Composition Vs Inheritance

Suppose Jokhon amra class use kori Inheritance holo ekta class jokhon tar parent class theke kono boishito inharit kore niya ashe. Inharit holo niya asha.

```
xport default class Calculator extends React.Component {
}
```

React er ja builtin componrnt ase take extend kore nai. Ai ja React.Component korlam aita e inheritance. Inheretance amra js a korte pari through class. Class er maddome inheritance achieve korte pari.

React bolse tmi class & function component a inheritance use korba na Composition use koro shob shomoi.

Inheretance:

```
import Emoji from "./imoji";

export default class Text extends Emoji {
    constructor(props){
        super(props);
    }
}
```

Amra jani Extends korbo inherentace korte gele amk ekta constructor function nite hobe & super call kore dite hoi.

Inheretance Example:

Imoji.js

```
import React from 'react';

export default class Emoji extends React.Component {
    addEmoji = (text, emoji) => `${emoji} ${text} ${emoji}`;
    render(){
        const text = 'I am the Emoji Component';
        return <div>{text}</div>
    }
}
```

Text.js

```
import Emoji from "./imoji";

export default class Text extends Emoji {
    constructor(props){
        super(props);
    }
}
```

Akhon Suppose…. Ai text.js child component theke imoji.js parent component er addEmoji function k call korbo So er jonno amk super.addEmoji call korte hobe.

Final Code Inheretance:

Imoji.js

```
import React from 'react';

export default class Emoji extends React.Component {
    addEmoji = (text, emoji) => `${emoji} ${text} ${emoji}`;
    render(override){
        let text = 'I am the Emoji Component';
        if(override) text = override;
        return <div>{text}</div>
    }
}
```

Text.js

```
import Emoji from "./imoji";

export default class Text extends Emoji {
    constructor(props){
        super(props);
    }
    render(){
        const decoratedTexr = this.addEmoji('I am JS Language', '😋');
        return super.render(decoratedTexr);
    }
}
```

App.js

```
import Text from "./component/text";
function App() {
    return <Text/>
}
export default App;
```

Aita just bujar jonno Example.

Higher Order Component(HOC)

App.js

```
import ClickCounter from "./HOC/ClickCounter";
import HoverCounter from "./HOC/HoverCounter";
function App() {
    return <div className="app">
        <ClickCounter/>
        <HoverCounter/>
```

```
        </div>
}
export default App;
```

ClickCounter.js

```
import React from 'react';

class ClickCounter extends React.Component {
    state = {
        count: 0,
    }
    incrementCount = () => {
        this.setState((preState) => ({ count: preState.count + 1}))
    }
    render() {
        return (
            <div>
                <button type='button' onClick={incrementCount}>Clicked
{count}times</button>
            </div>
        )
    }
}
export default ClickCounter;
```

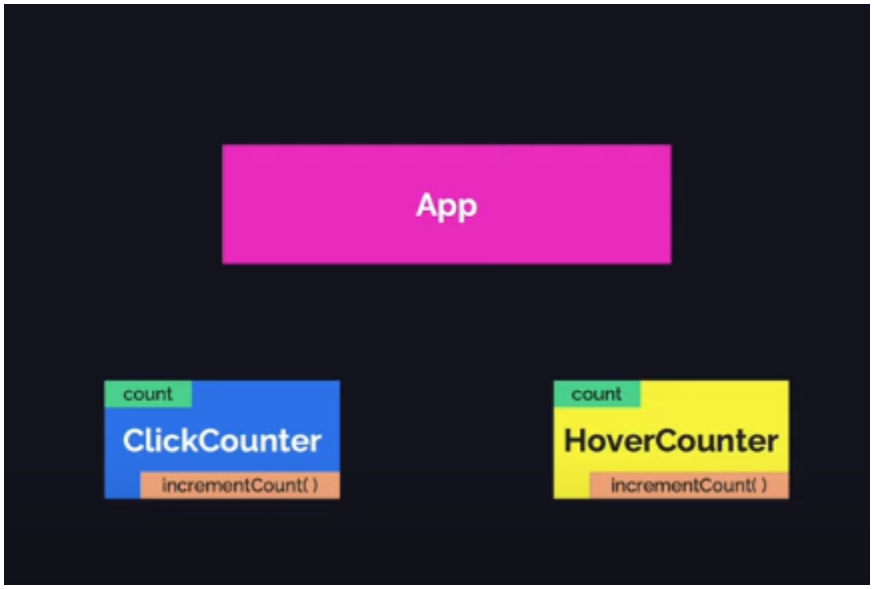
HoverCounter.js

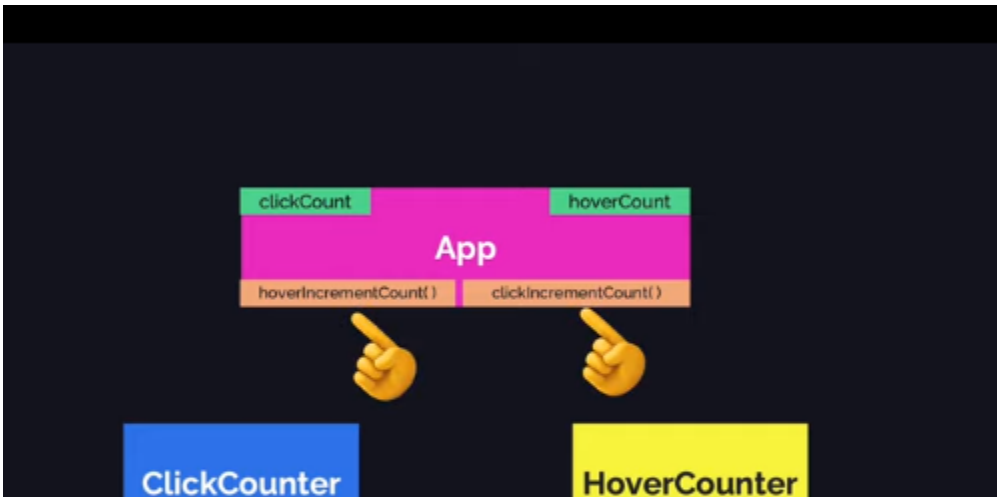
```
import React from 'react';

class HoverCounter extends React.Component {
    state = {
        count: 0,
    }
    incrementCount = () => {
        this.setState((preState) => ({ count: preState.count + 1}))
    }
    render() {
        return (
            <div>
                <h1 type='button' onMouseOver={this.incrementCount}>Clicked
{count}times</h1>
            </div>
        )
    }
}
export default HoverCounter;
```

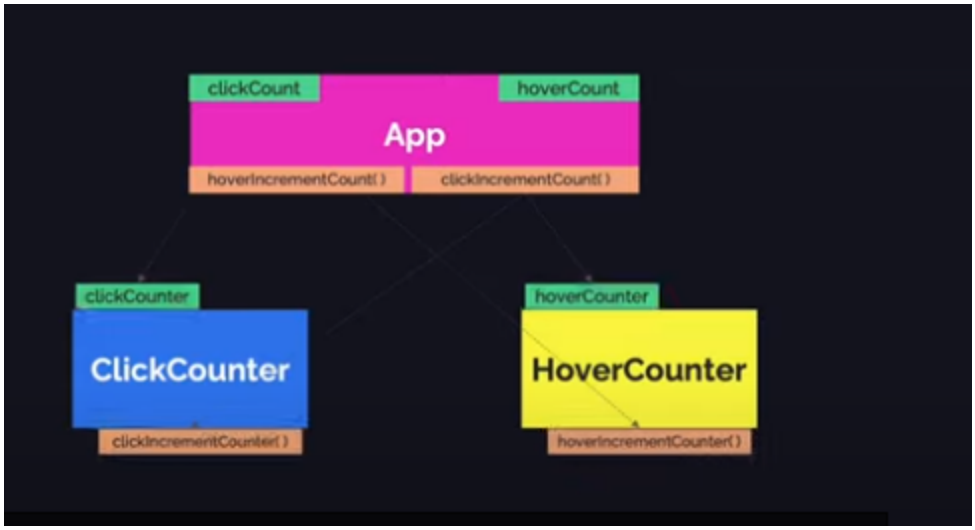
Output:

So amr code kaj korse All Ok. Akhon kisu din por amr client Jodi er ekta requirement niya ashe ja ekta input element er modda user koibar key press korlo sheta track korte hobe. Tahole ami ai component k copy kore abr banate pari. Ate ki hosse ek e kaj amr bar bar kora hosse. Code repeat hosse. So amk Jodi amon similar functionality er function banate hoi tahole kisu din pore aita amr kase e manageable thakbe na. Kisu change korte hole shob gulo component dhore dhore change korte hobe. Ja scalable na.

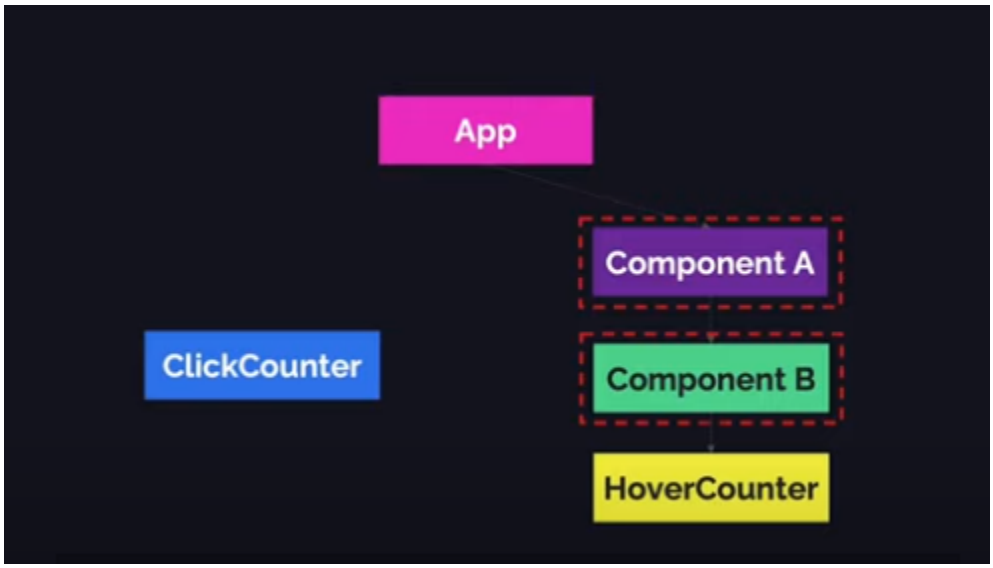
So Amder amon kono upai bar korte hobe jeno similar code gulo reuse korte pari. So ki korte pari… Er ekta solution hote pare state take ek dhap upore tule dai using listingStateUp Patarn. Ai khatre amder App.js er state er modda 2 ta value rakte hobe clickCounter & Hover counter And handler o 2 ta banate hobe and pore props er maddome pathiya dite hobe.







Akhon props akare pass korle amon kisu component er modda diya jete hoto jader kase oi props gulo o proio jonio.

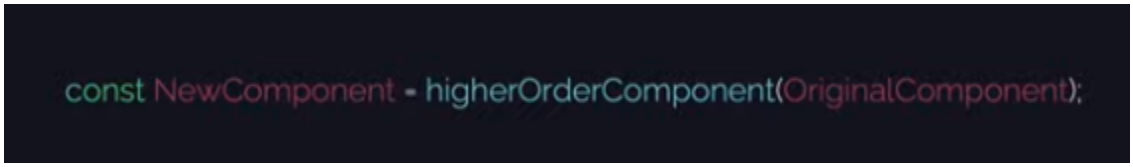


So LiftingStateUp ai khatre correct solution na. So amra bujte e parsi code reapt na kore Component er modda common funcationality share korer 1 ta need ase. Ai khane e Higherorder component er Concept ta chole ashe.

Now What is Higher order component?

A higher order component is a function that takes a component as parameter and return a new component.

Component Example:



Now Create HOC:

withCounter.js

```
import React from "react";

//withCounter ekta function nicci
const withCounter = (OriginalComponent) => {
    class NewComponent extends React.Component {
        //Ager code theke common jinish tuko niya ashi
        state = {
            count: 0,
        }
        incrementCount = () => {
            this.setState((preState) => ({ count: preState.count + 1}))
        }
        render() {
            const {count} = this.state;
            return <OriginalComponent/>
        }
    }
    return NewComponent;
}
export default withCounter;
```

clickCounter.js er state ta bad diya dibo. And class Component take function component baniya felbo.

Final Code:

withCounter.js

```
import React from "react";

//withCounter ekta function nicci
const withCounter = (OriginalComponent) => {
    class NewComponent extends React.Component {
        //Ager code theke common jinish tuko niya ashi
        state = {
            count: 0,
        }
        incrementCount = () => {
            this.setState((preState) => ({ count: preState.count + 1}))
        }
        render() {
            const {count} = this.state;
            return <OriginalComponent count={count}
incrementCount={this.incrementCount}/>
        }
    }
    return NewComponent;
}
export default withCounter;
```

ClickCounter.js

```
import withCounter from "./withCounter";

const ClickCounter = () => {
    const {count, incrementCount} = props;
    return (
        <div>
            <button type="button" onClick={incrementCount}>
                Clicked {count} times
            </button>
        </div>
    )
}
export default withCounter(ClickCounter);
```

\HoverCounter.js

```
import React from 'react';
import withCounter from './withCounter';

const HoverCounter = (props) => {
    const { count, incrementCount} = props;
    return (
        <div>
            <h1 onMouseOver={incrementCount}>Hovered {count} times</h1>
        </div>
    )
}
export default withCounter(HoverCounter);
```

Rendering Props Pattern

User.js

```
export default function User({ name }) {
    return name;
}
```

App.js

```
return (
    <div>
        <User name='sumit'/>
    </div>
)
```

Amra EKta function kew o props hishabe pathate pari. Example:

```
export default function User({ name }) {
    return name();
}
```

```
return (
    <div>
        <User name={() => 'Sumit'}/>
    </div>
)
```

Function er modda parameter pathate chile:

```
export default function User({ name }) {
    return name(True); //output: sumit
    return name(false); //output: Guest
}


return (
    <div>
        <User name={(isLoggedIn) => (isLoggedIn? 'sumit'
: 'Guest')}/>
    </div>
)
```

Ai khane name Props take kore dai render tahole e render props:

```
export default function User({ render }) {
    return render(True); //output: sumit
}


return (
    <div>
        <User render={(isLoggedIn) => (isLoggedIn?
'sumit' : 'Guest')}/>
    </div>
)
```

Render Prop: Prop that defines render Logic. Render prop simply ekta function je function ta ultimately she component er render logic ta baire theke niya nijer modda render kore.

Counter.js

```
import React from 'react';

class Counter extends React.Component {
    state = {
```

```
        count: 0,
    }
    incrementCount = () => {
        this.setState((preState) => ({ count: preState.count + 1}))
    }
    render() {
        const {children} = this.props;
        // const {render} = this.props;
        const {count} = this.state;
        return render(count, this.incrementCount);
    }
}
export default Counter;
```

WithCounter.js

```
import React from 'react';

export default function ClickCounter({count, incrementCount}){
    return (
        <div>
            <button type='button' onClick={incrementCount}>Clicked {count}
times</button>
        </div>
    )
}
```

App.js

```
import ClickCounter from "../src/HOC/ClickCounter";
import Counter from "./Counter";
import HoverCounter from "../src/HOC/HoverCounter";

function App() {
    return(
        <div className="app">
            <Counter>
                {(Counter, incrementCount) => (
                   <ClickCounter Counter={Counter}
incrementCount={incrementCount}/>
                )}
            </Counter>
            <Counter>
                {(Counter, incrementCount) => (
                   <HoverCounter Counter={Counter}
incrementCount={incrementCount}/>
                )}// Atai prop.Children
            </Counter>
            {/* <User render={(isLoggedIn) => (isLoggedIn? 'sumit' : 'Guest')}/>
*/}
            <Counter render={(Counter, incrementCount) => (
```

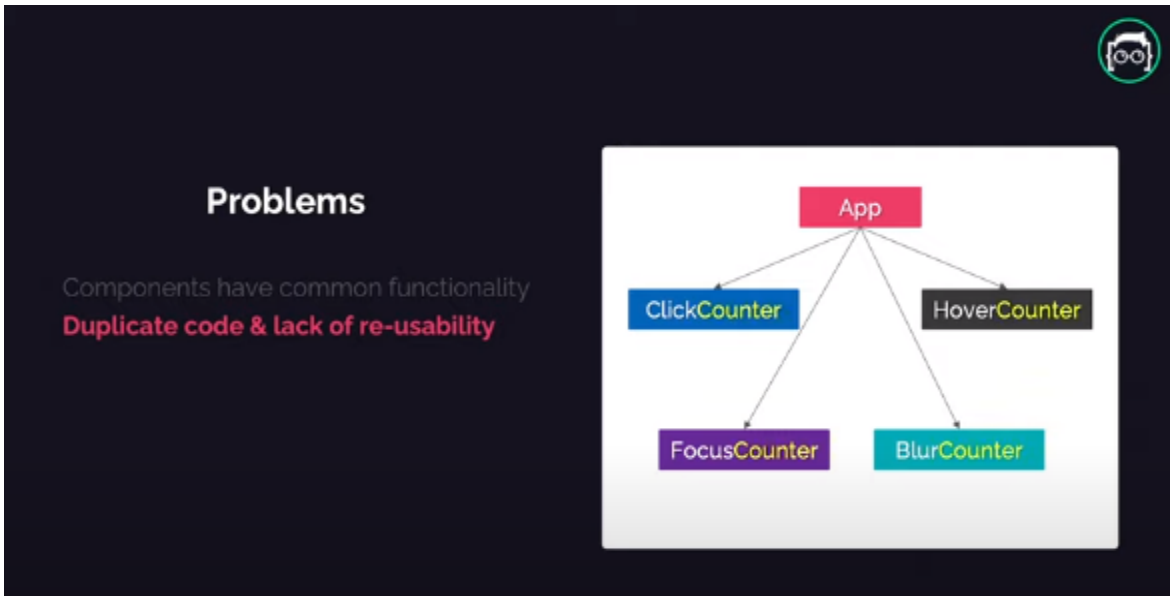
```
                <ClickCounter Counter={Counter} incrementCount={incrementCount}/>
            )}/>
        </div>
    )
}
```

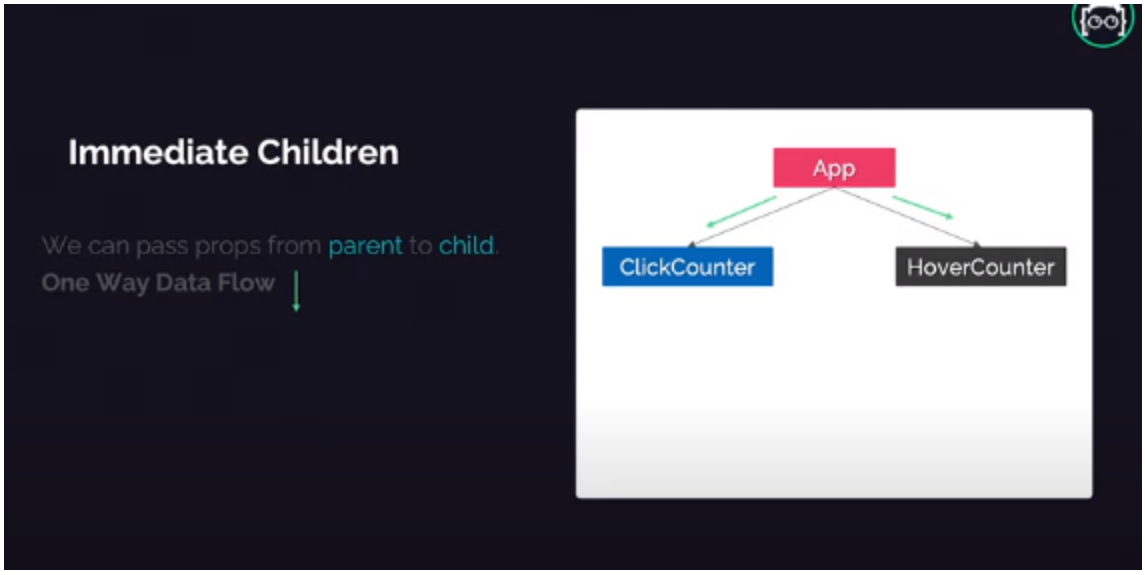
Render props diya ja slove korlam ai gula e context API diya o solve korte parbo. Next Topic context API.

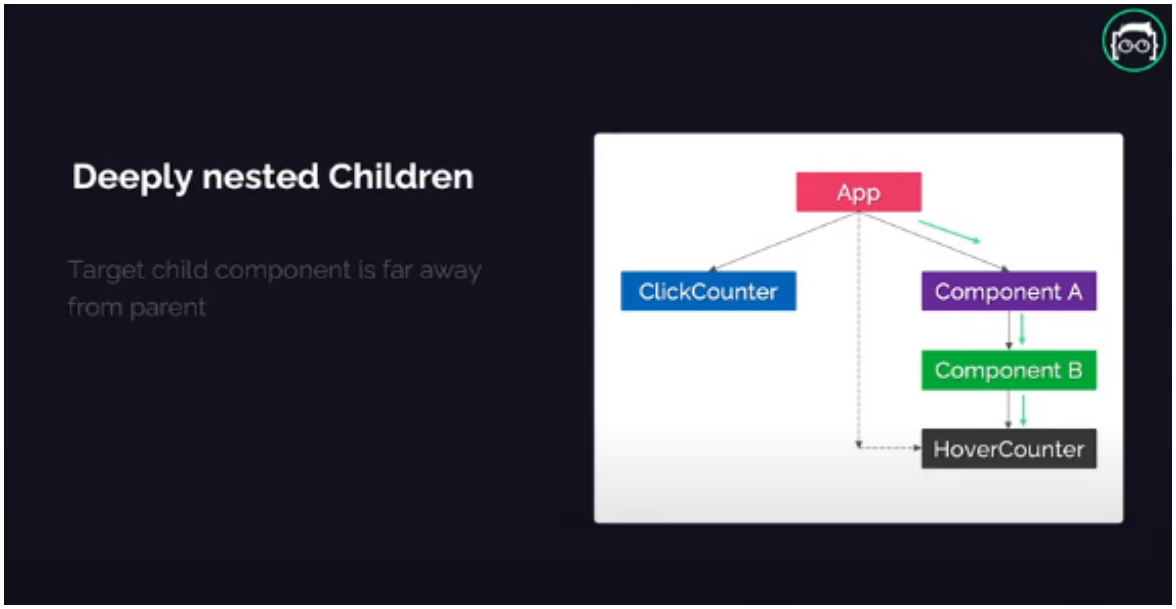
React Context API

Ager tutorial gular summary:

Onek gula component ek e dhoroner kaj kore. User interaction track kore and ekta function er modda update kore. But different different component er jonno ek e code bar bar likte hossilo. Component er reuse ability kore jassilo amder ek e code bar bar likte hossilo.







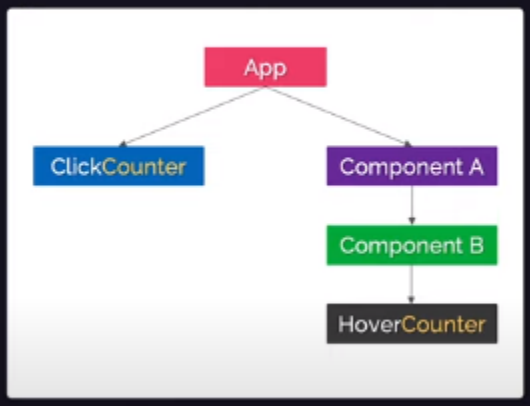
Ai probem solve korte amra er ager tutorial a HOC & rendering props dakhisi.

Props Drilling

- Higher Order Component (HOC)
- Render Props pattern

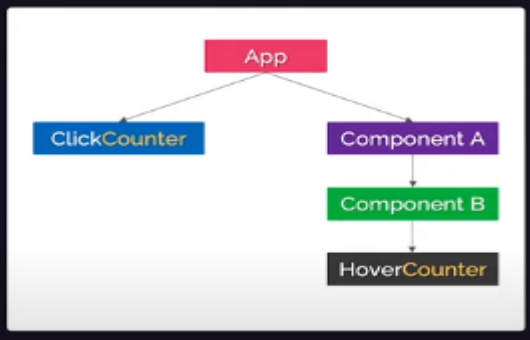
Higher Order Component

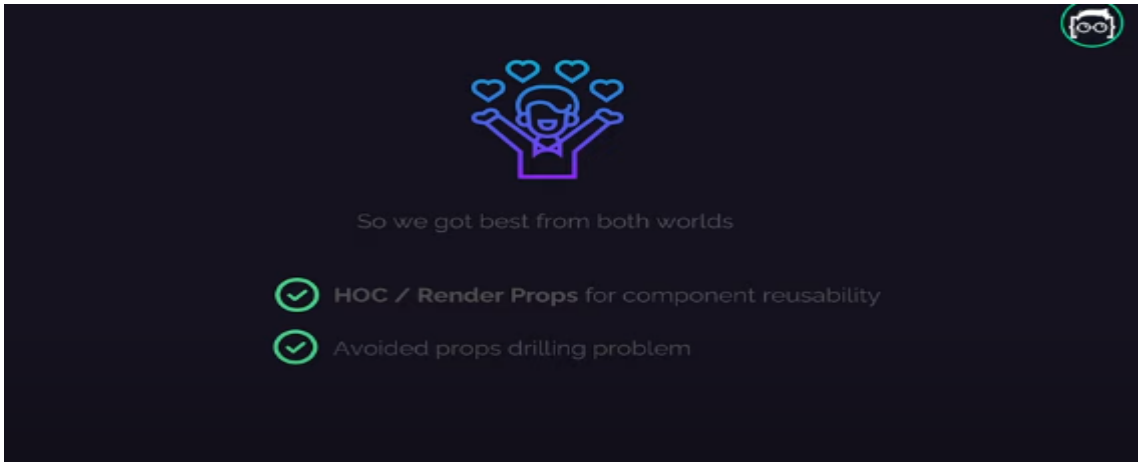
```
withCounter(Component){
  state or data
  .....
  return <Component {...props} />
}

withCounter(HoverCounter)
```

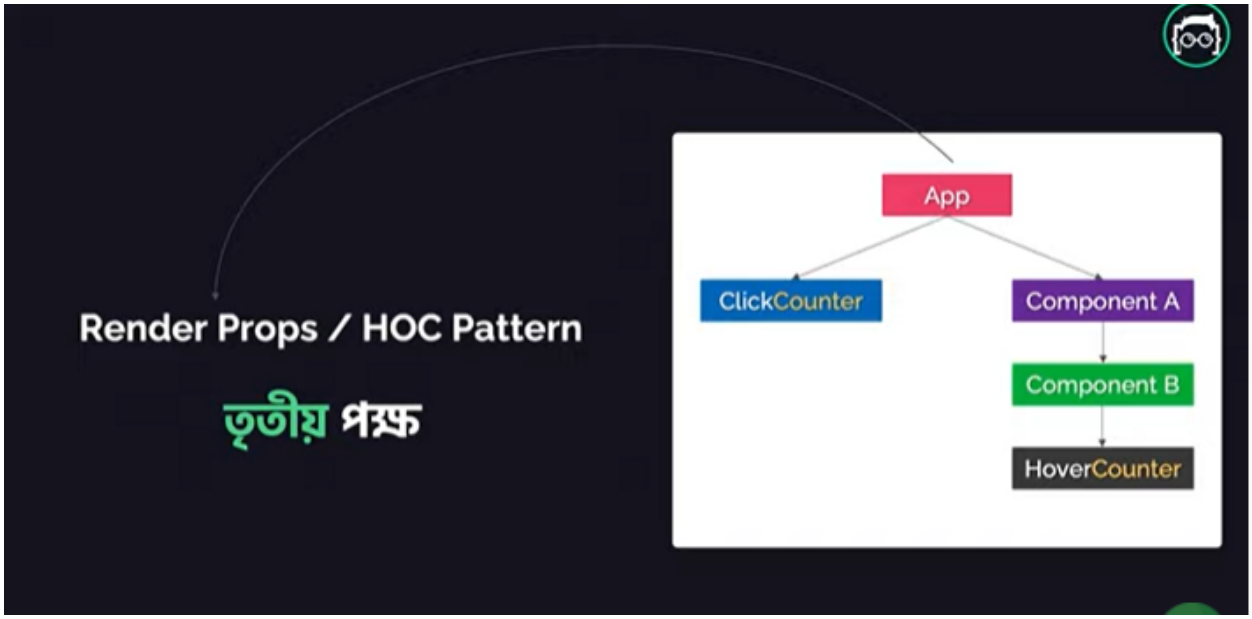


Render Props Pattern

```
class Counter {
  state or data
  render() {
    return this.props.children( anything );
  }
}

<Counter>
  ( anything ) => < HoverCounter anything />
</Counter>
```





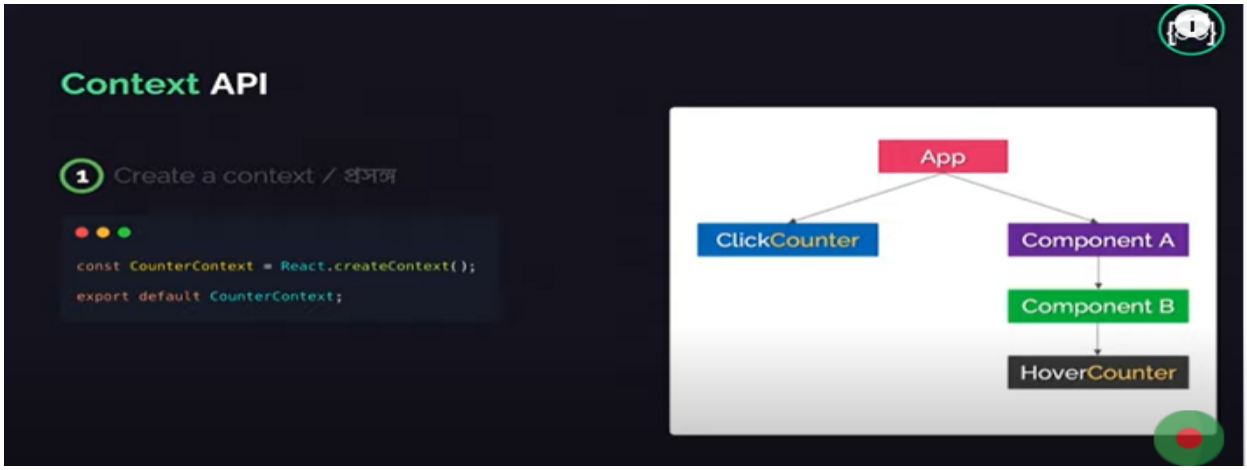


App theke dril na kore Direct HoverCounter a jete parsi na Shehutu titio ekjon poker shahajjo nite hosse sheta render props(er khatre Component), HOC(er khatre function). So titilo ekta poker modda amk funcationality rakhe shekhan theke props bore diya amk HoverCounter er kase patate hoiyase.

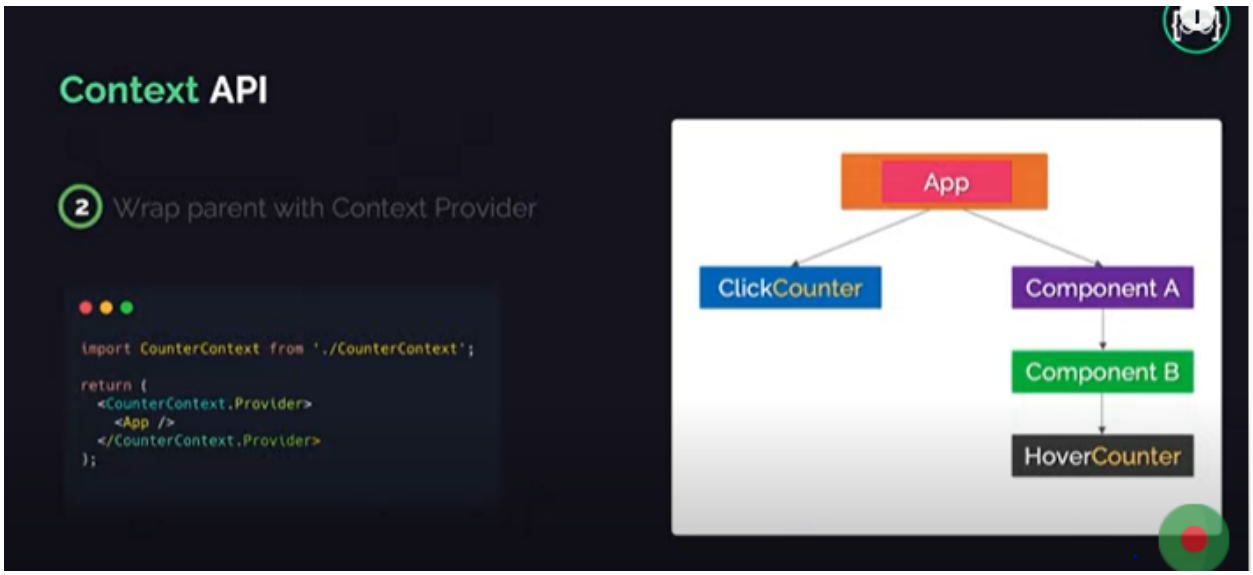
Akhon amon er ekta titio pokko holo Context ApI Amra tar shomporke e akhon janbo.

Context = proshongo

App theke dibo HoverCounter a nibo maj khaner component diya she jabe na. HoverCounter er Jodi App er kono proshongo k tene ante hoi tahole she context API or proshongo API use kore she shai proshongo take tene ante parbe.



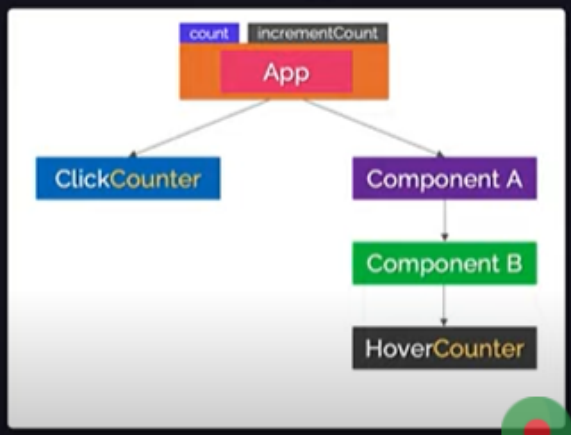
Create context ApI & aita amder 2 ta proshongo diya dai provider & consumer



Context API

3 Provide a value of the context as prop

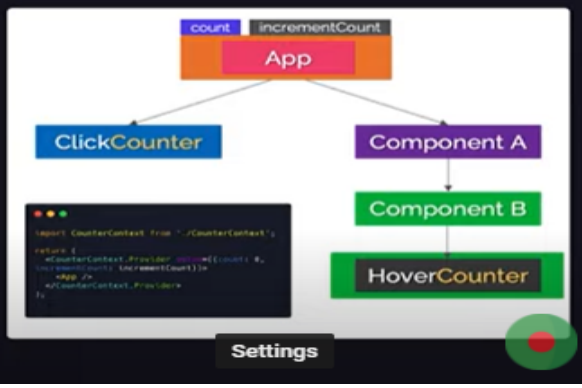
```
import CounterContext from './CounterContext';

return (
  <CounterContext.Provider value={{count: 0,
incrementCount: incrementCount}}>
    <App />
  </CounterContext.Provider>
);
```



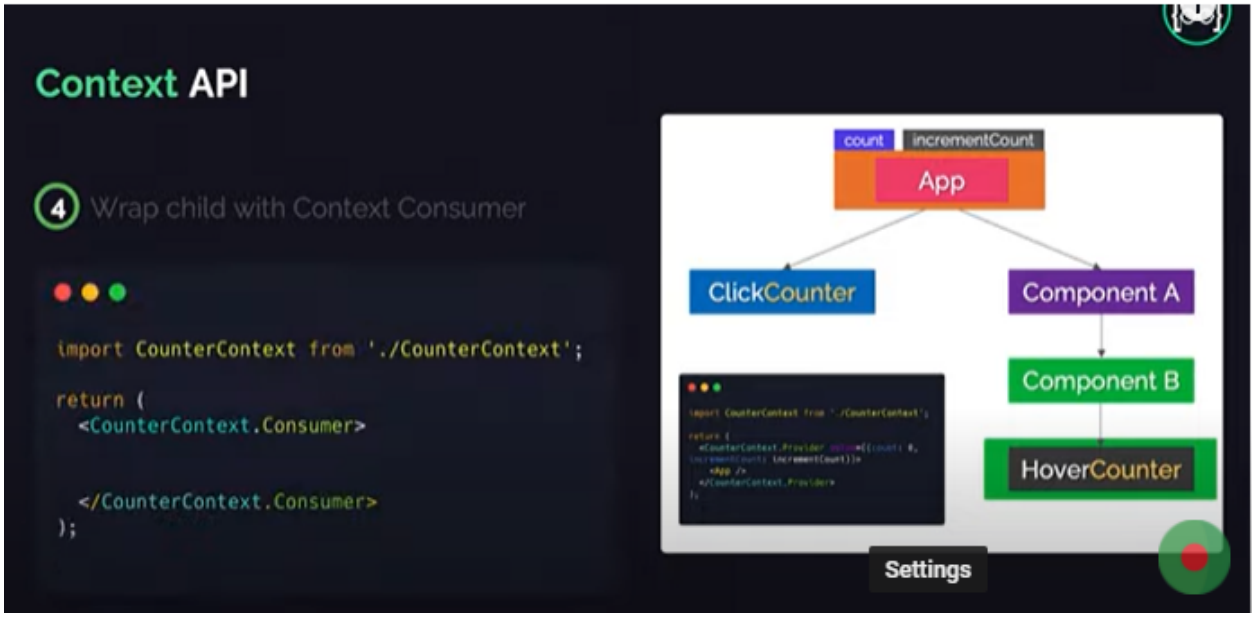
Context API

4 Wrap child with Context Consumer

```
import CounterContext from './CounterContext';

return (
  <CounterContext.Consumer>


  </CounterContext.Consumer>
);
```



Settings



Context API k ekta stores er moto chinta korte pari.

Use of Context:

Context is primarily used when some data needs to be accessible by *many* components at different nesting levels. Apply it sparingly because it makes component reuse more difficult.

If you only want to avoid passing some props through many levels, component composition is often a simpler solution than context.

React Context API_ contextType & useContext Hook

Ai tutorial a Context API er boishisto dekbo.

App.js

```
import React from 'react';
import ClickCounter from './components/ClickCounter';
import Counter from './components/Counter';
import Section from './components/Section';
import ThemeContext from './contexts/themeContext';
export default class App extends React.Component {
  state = {
    theme: 'dark',
  }
  switchTheme = () => {
    this.setState(({ theme }) => {
      if (theme === 'dark') {
        return {
          theme: 'light',
        }
      }
      return {
        theme: 'dark',
      }
    })
  }
  render() {
    const { theme } = this.state;
    return (
      <div className="app">
        <Counter>
          {(counter, incrementCount) => (
            <ClickCounter count={counter}
incrementCount={incrementCount}/>

          )}
```

```
        </Counter>
        <ThemeContext.Provider value={{
theme,switchTheme: this.switchTheme }}>// switchTheme
function take o value hishabe pathi dilam
          <Section/>
        </ThemeContext.Provider>
      </div>
    )
  }
}
```

`Theme : theme` but same name howate ekbar e likha hoiyase `switchTheme: this.switchTheme`

Property: value

switchTheme function ta uporer state er theme take update korbe. Light hole dark & dark hole light korbe. Ai switch theme k call korbe Hover.

Content.js

```
import ThemeContext from "../contexts/themeContext";
import Counter from "./Counter";
import HoverCounter from './HoverCounter'
export default function Content() {
  return (
    <div>
      <h1>This is a content</h1>
      <Counter>
        {(counter, incrementCount) => (
          <ThemeContext.Consumer>
            {({ theme, switchTheme }) => (
              <HoverCounter
                count={counter}
                incrementCount={incrementCount}
                theme={theme}
```

```
                switchTeme={switchTheme} //props akare
pathiya dissi.
                />
            )}
          </ThemeContext.Consumer>
        )}
      </Counter>
    </div>
  );
}
```

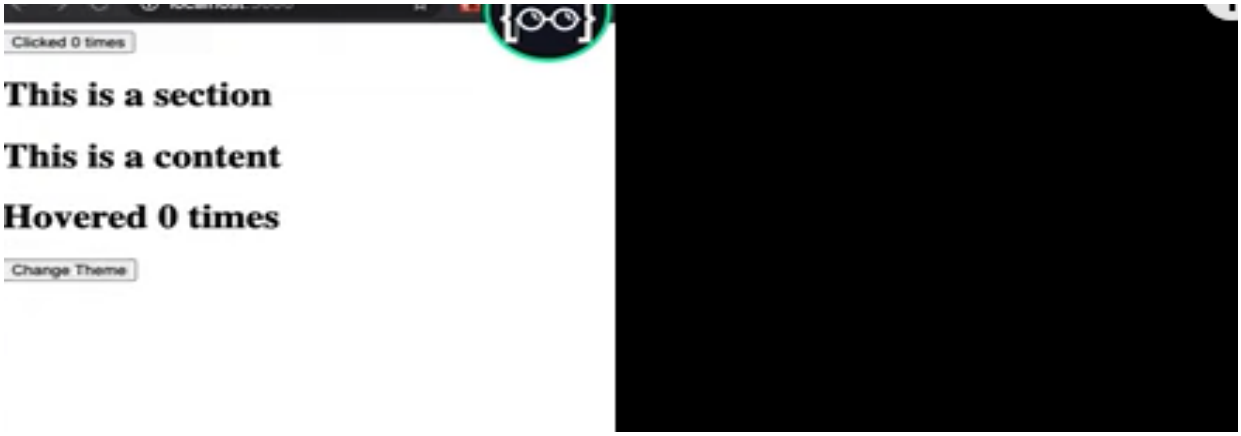
HoverCounter.js

```
import React from 'react';

export default function HoverCounter({ count,
incrementCount, theme, switchTheme }) {
    const style = theme === 'dark' ? { backgroundColor:
'#000000', color: '#ffffff' } : null;

    console.log('HoverCounter rendered');
    return (
        <div>
            <h1 onMouseOver={incrementCount}
style={style}>
                Hovered {count} times
            </h1>
            <button type="button" onClick={switchTheme}>
                Change Theme
            </button>
        </div>
    );
}
```

Counter.js

```
import React from 'react';

class Counter extends React.Component {
    state = {
        count: 0,
    };

    incrementCount = () => {
        this.setState((prevState) => ({ count:
prevState.count + 1 }));
    };

    render() {
        const { children } = this.props;
        const { count } = this.state;
        return children(count, this.incrementCount);
    }
}

export default Counter;
```

Output:



Akhon content.js Jodi class component hoto? Akhon tai class component a er ekta patern dekhbo..

Content.js

```
import ThemeContext from "../contexts/themeContext";
import Counter from "./Counter";
import HoverCounter from './HoverCounter'
import React from 'react';
export default class Content extends React.Component {
  componentDidMount() {
    console.log(this.context);
  }
  render() {
    const { theme, switchTeme } = this.context;
    return (
      <div>
        <h1>This is a content</h1>
        <Counter>
          {(counter, incrementCount) => (
            <HoverCounter
              count={counter}
              incrementCount={incrementCount}
              theme={theme}
```

```
              switchTeme={switchTeme}
            />
          )}
        </Counter>
      </div>
    );
  }
}
Content.contextType = ThemeContext;
```

Akhon Abr dekbo ai patern function er jonno.

```
 <ThemeContext.Consumer>
          </ThemeContext.Consumer>
```

So Akhon o ai consumer er dorkar nai. UseContext Use korbo.

Content.js

```
import ThemeContext from "../contexts/themeContext";
import Counter from "./Counter";
import HoverCounter from './HoverCounter'
import React from 'react';
import { useContext } from "react";
export default function Content() {
  const context = useContext(ThemeContext);
  const { theme, switchTeme } = context;
  return (
    <div>
      <h1>This is a content</h1>
      <Counter>
        {(counter, incrementCount) => (
          <HoverCounter
            count={counter}
            incrementCount={incrementCount}
            theme={theme}
```

```
            switchTeme={switchTeme}
          />
        )}
      </Counter>
    </div>
  );
}
```

Provider change hole consumer rerender hoi. Context a provider er shate consumer er relation.

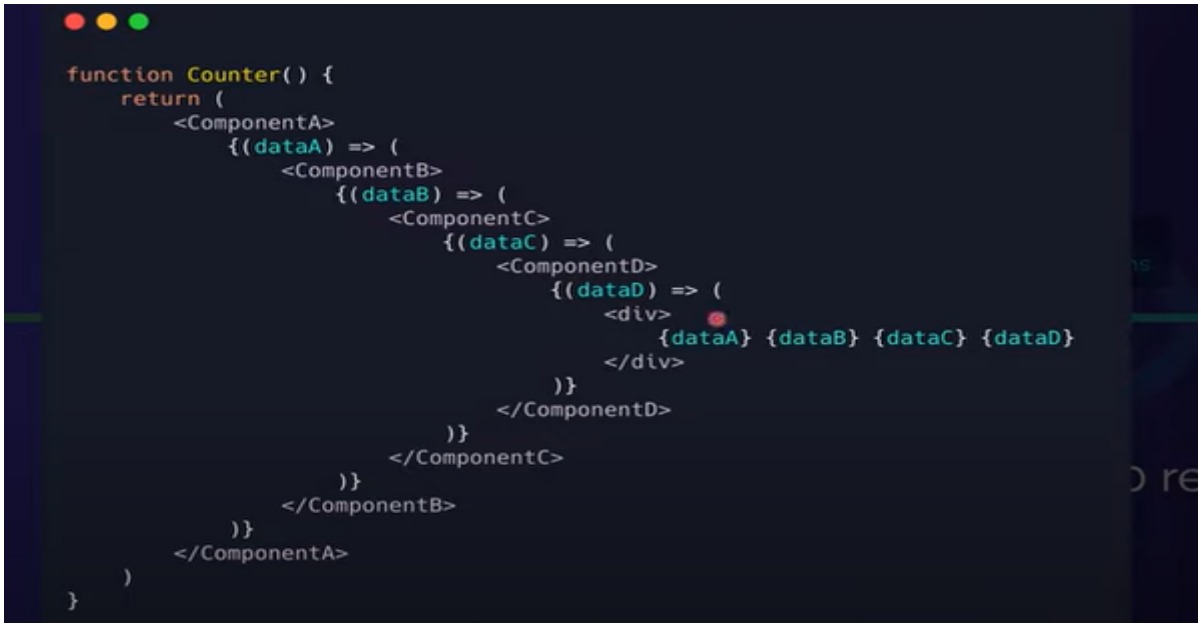
Hooks

Hook Introdruction:
Why React Hooks?

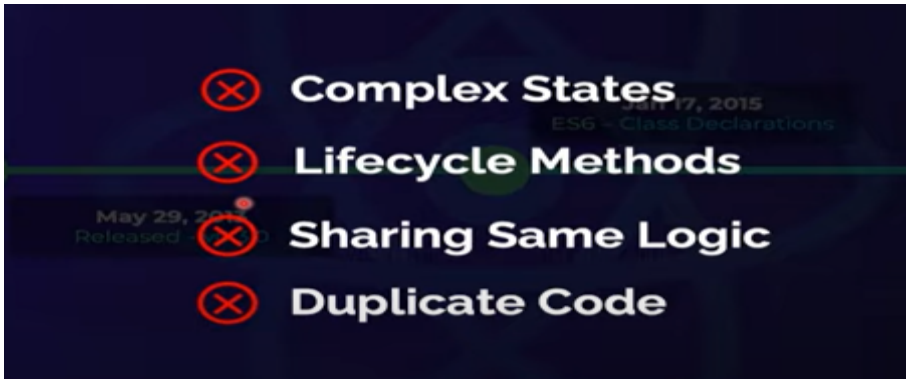
React 2013 a fast release hoi and tokhon react dekte amon chilo. Aita pura ta ekta function tar modda object.

Higher order ekta pattern aita react er kisu na. ER ekta patern amra dekhisi sheta holo Render Props Pattern.

But render props kore kore jete gele ai rokom ekta kun akriti hote pare.



React er problem:

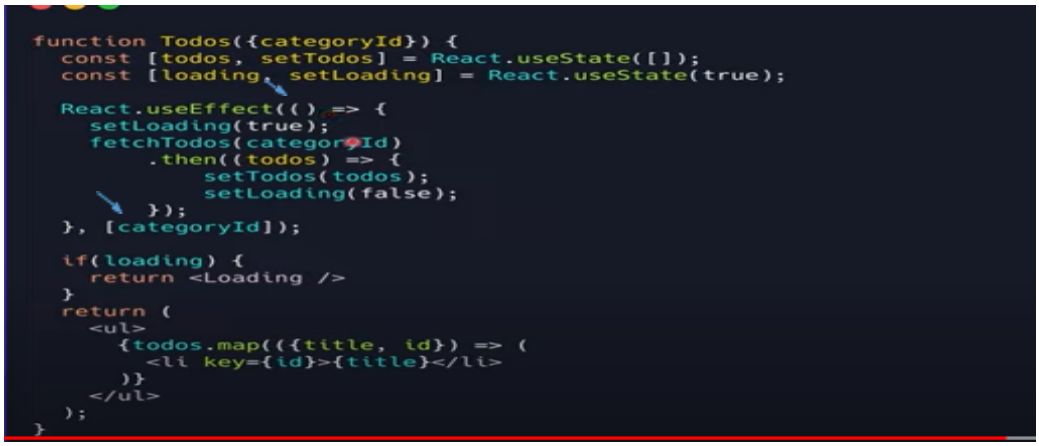


React Hooks come Feb 16, 2019.

useState & useEffect Hook.

useEffect:

User interface banate giya je data dorkar hosse she data gulo synchronise korte amder kisu side effect proijon hoi jamon server theke data ante hoi sita holo side effect. So ai kaj gulo amra rect er life cycle use kore kortam. Jamon componentDidMount use kortam. Jeta amder manually kore dite hoto. But hooks er maddome amra sheta indirectly korte pari React life cycle method use na kore.



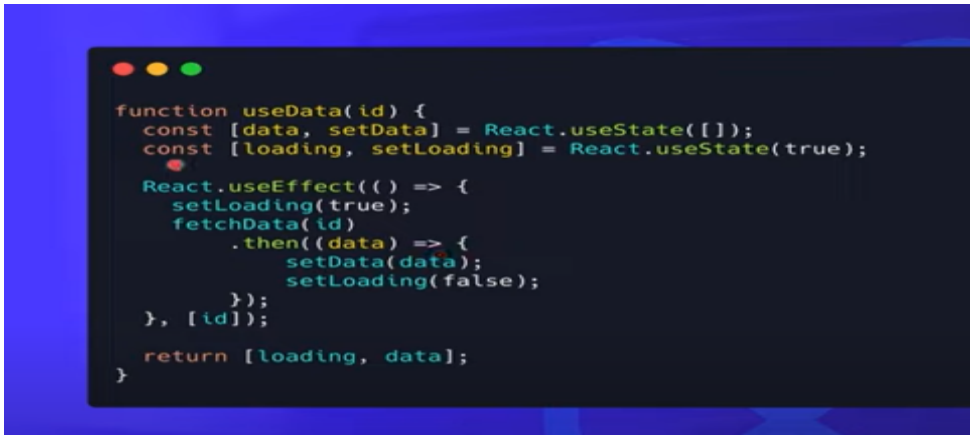
```
function Todos({categoryId}) {
  const [todos, setTodos] = React.useState([]);
  const [loading, setLoading] = React.useState(true);

  React.useEffect(() => {
    setLoading(true);
    fetchTodos(categoryId)
        .then((todos) => {
            setTodos(todos);
            setLoading(false);
        });
  }, [categoryId]);

  if(loading) {
    return <Loading />
  }
  return (
    <ul>
      {todos.map(({title, id}) => (
        <li key={id}>{title}</li>
      ))}
    </ul>
  );
}
```

.useEffect ai khane ekta call back function parameter hishabe niyase. And 2nd parameter a ekta array niyase.

So akhon amra er maddome ki ki avoid korte parlam?

1. State
2. Lifecycle method
3. Duplicate Code
4. Sharing Same Logic

Custom Hook:



```
function useData(id) {
  const [data, setData] = React.useState([]);
  const [loading, setLoading] = React.useState(true);

  React.useEffect(() => {
    setLoading(true);
    fetchData(id)
        .then((data) => {
            setData(data);
            setLoading(false);
        });
  }, [id]);

  return [loading, data];
}
```

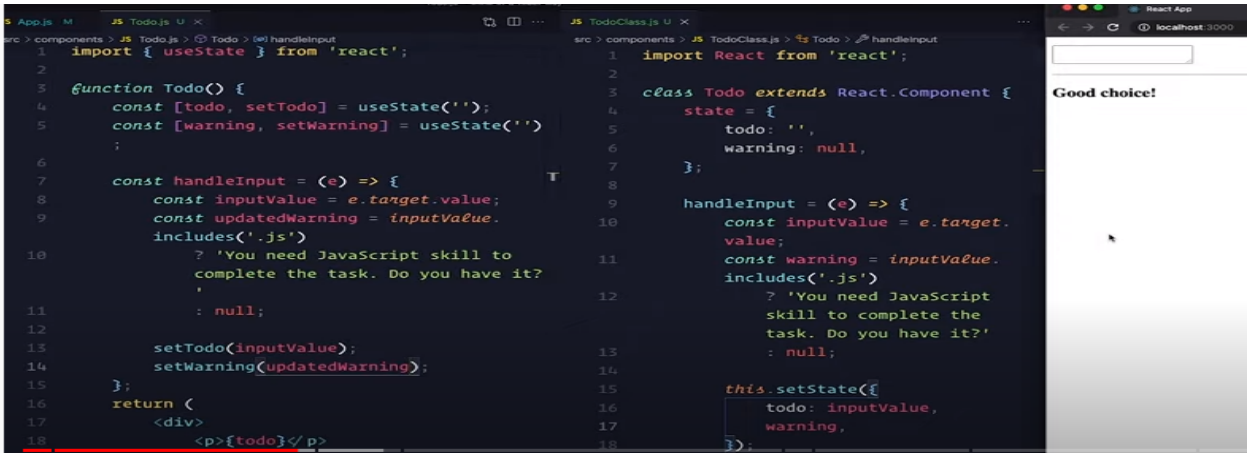
Some points to note

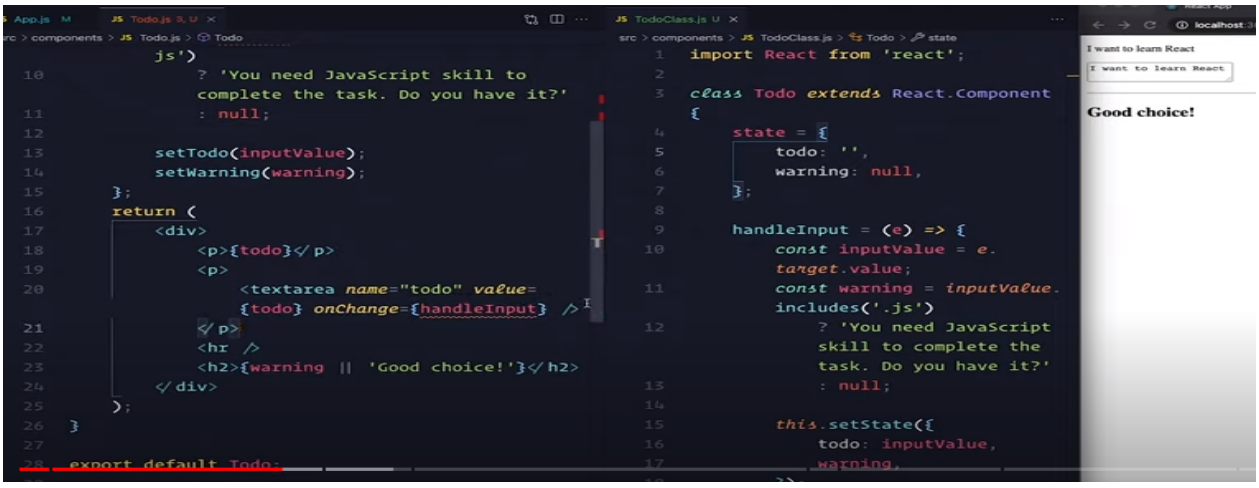
- Hooks are available from React version 16.8 +
- Hooks don't contain any breaking changes & it's 100% backward compatible.
- Hooks are optional & classes won't be removed from React.
- Hooks can't be used inside class component.
- Hooks don't replace your existing React knowledge. It just provides a more direct API to the React concepts you already know.

UseState Hook

Hook vs State:

Hook state





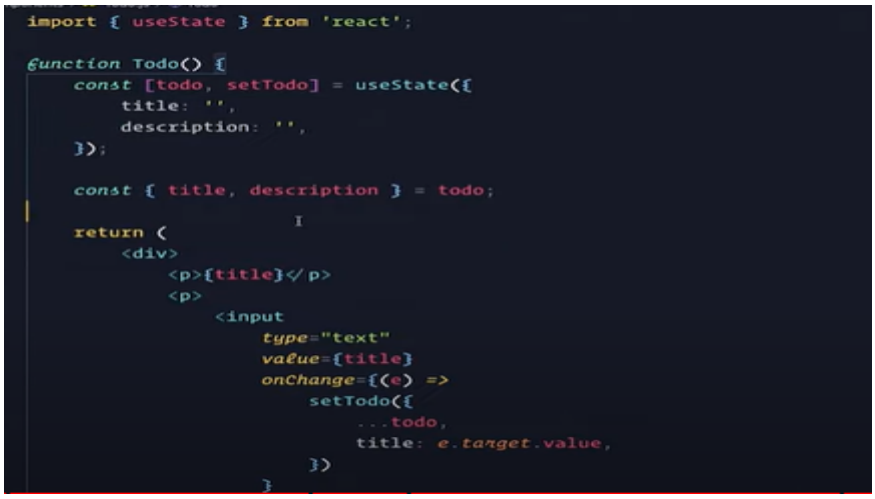
How useState work?

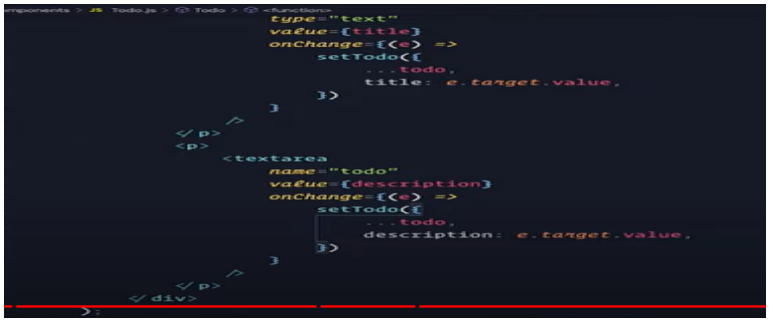
Ai khane fast a state initialize korlo. Er por return korbe JSX. So ai khane fast a todo er value blank.

My own useState hook implementation:

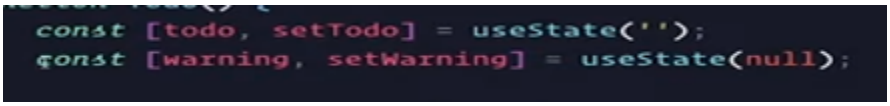
Hook shob shomi top level Mne upore e use hoi.

UseState with object/Array:





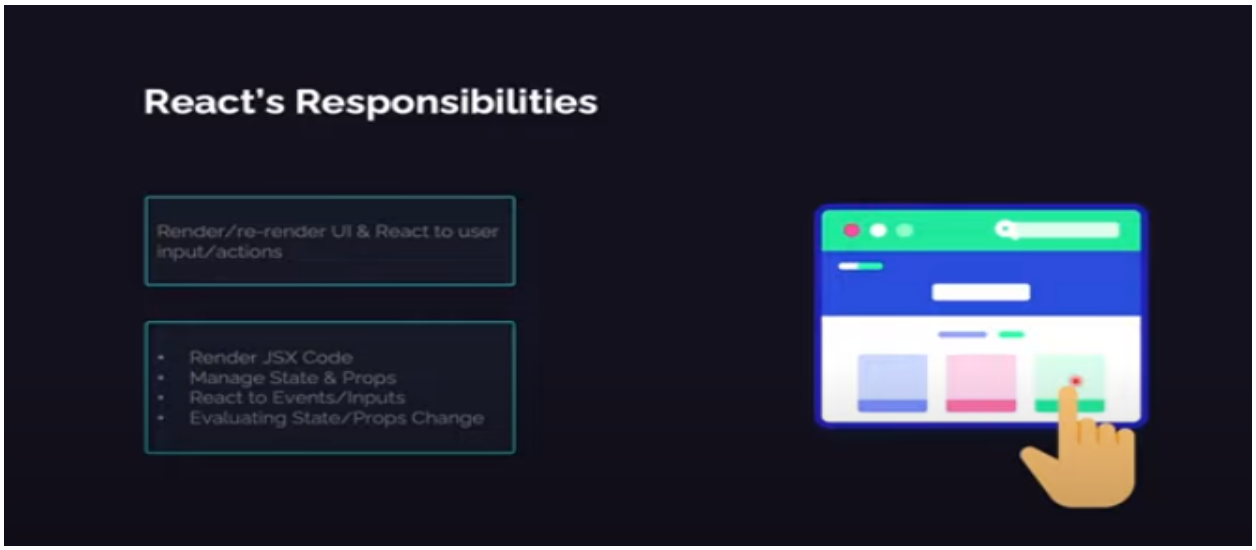
Hook & state use er modda ekta difference ase..



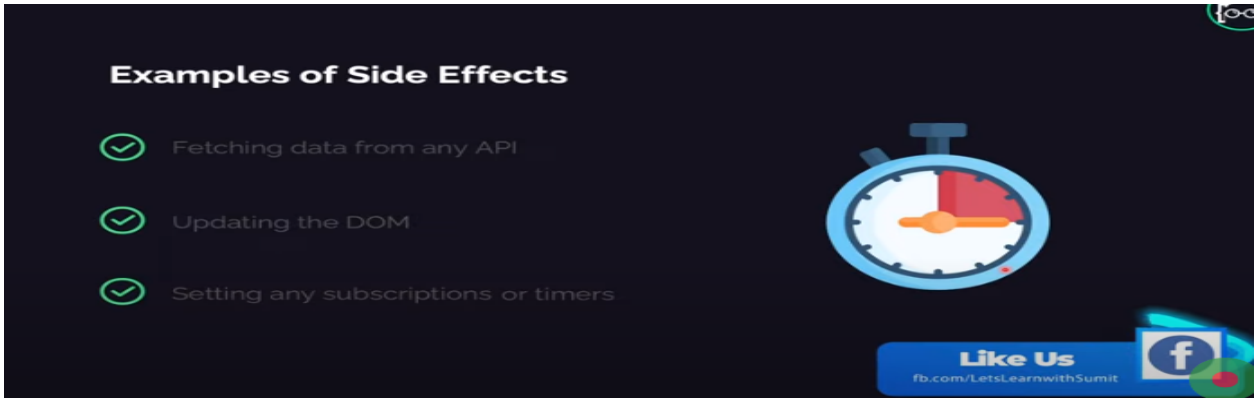
(‘ ’) = string ase akhon
Hook a proti ta state alada alada hoi. Pic a deya ase. But ai ta Jodi single value na hoiya object hoi? Tahole problem karon marge hobe na.

Akhon ekta object hoiya gelo..

What is react Responsibilities?



React er responsibilities UI render kora.
Responsibilities er baire ja ja ase ta shob e side effect.
What are side Effects?
Mane React er bairer kaj.



Side effect amra class component er modda ki vabe handle korbo?

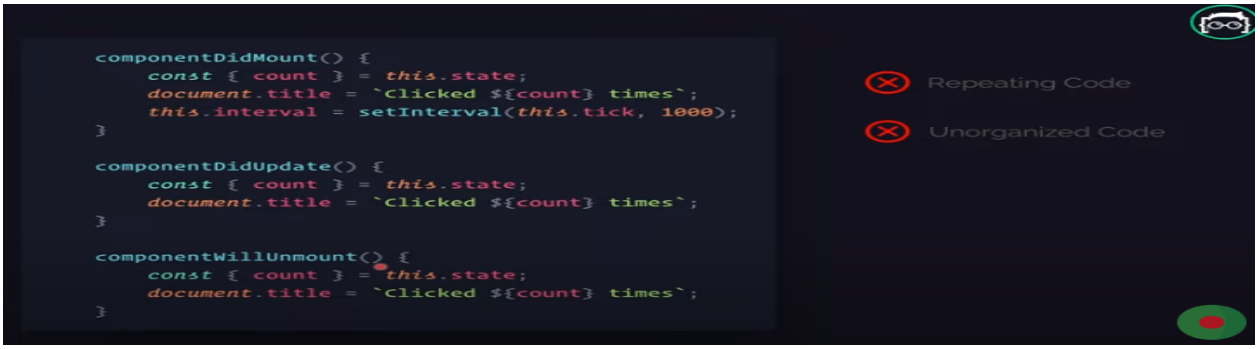
Life Cycle method diya handle korbo.

- ComponentDidMount()
- ComponentDidUpdate()
- ComponentDidUnmount()

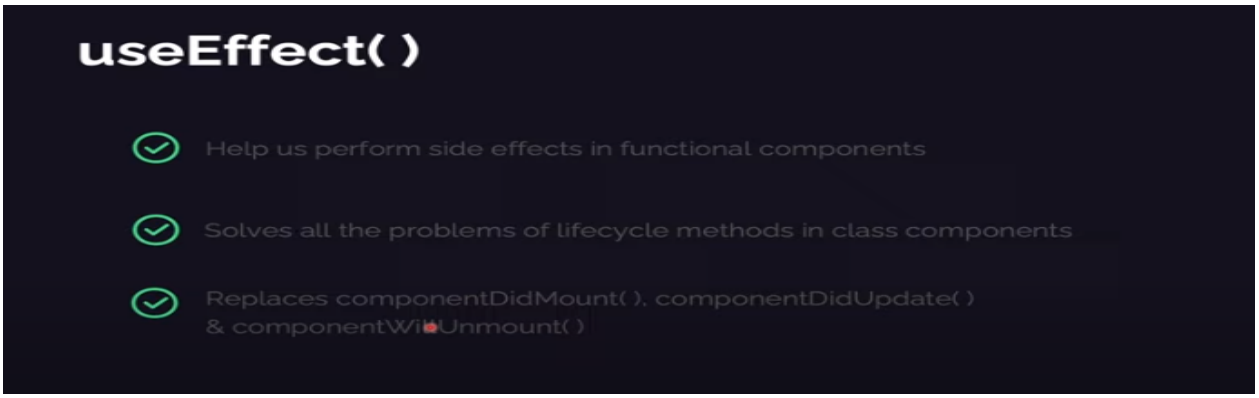
Example:

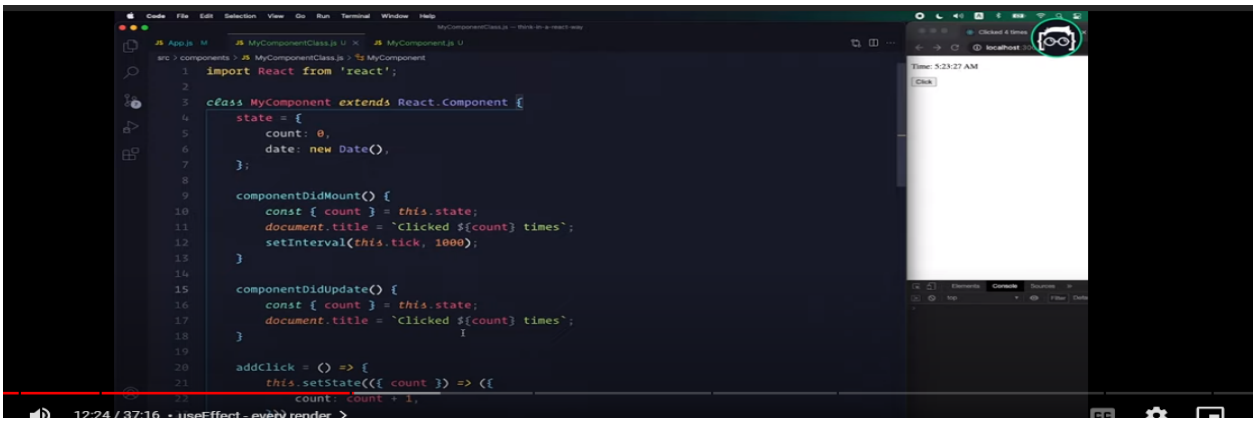
```
componentDidMount() {
    const { count } = this.state;
    document.title = `Clicked ${count} times`;
    this.interval = setInterval(this.tick, 1000);
}

componentDidUpdate() {
    const { count } = this.state;
    document.title = `Clicked ${count} times`;
}

componentWillUnmount() {
    const { count } = this.state;
    document.title = `Clicked ${count} times`;
}
```

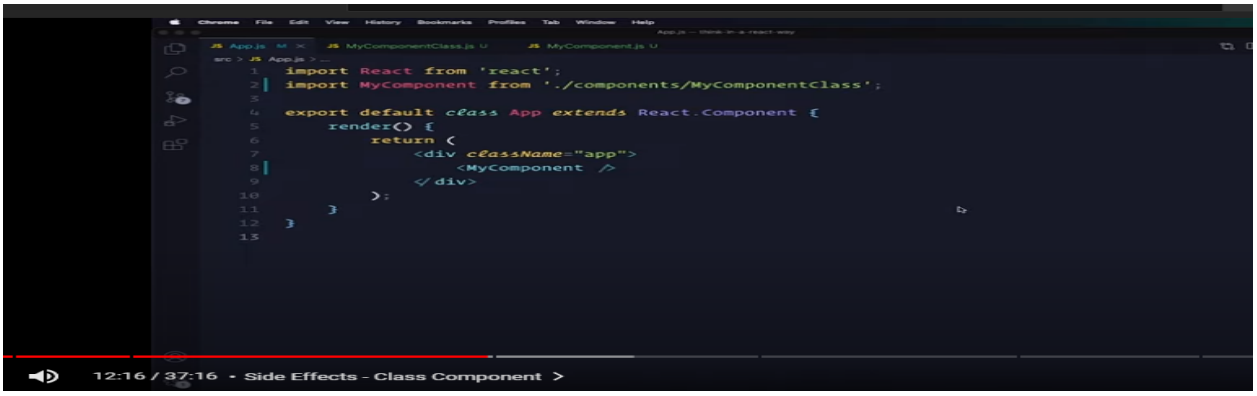
Why UseEffect?



Aita holo class component er khatre amra lifecycle method diya handle korasi. But problem holo Repeating Code & UnOrganized Code.

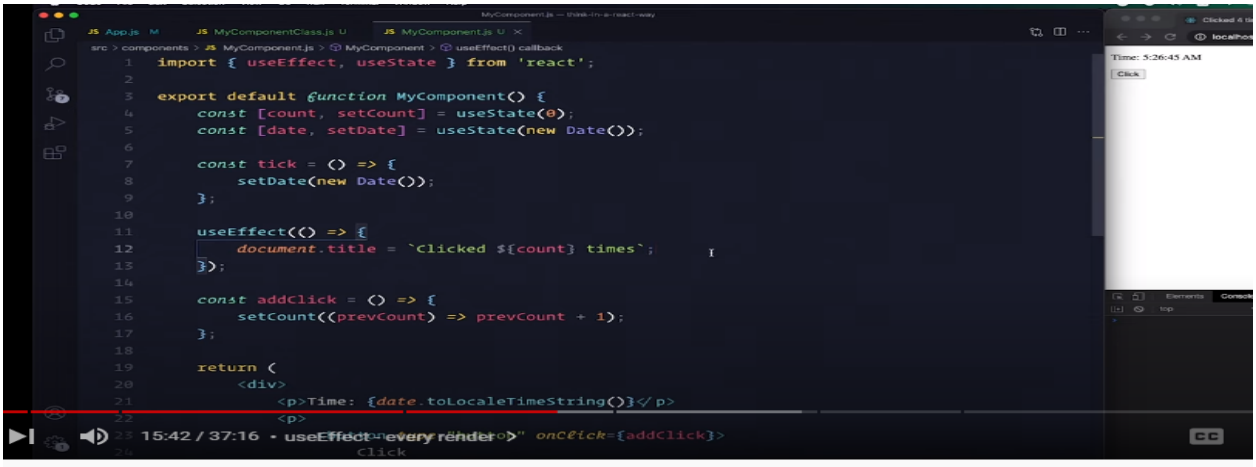






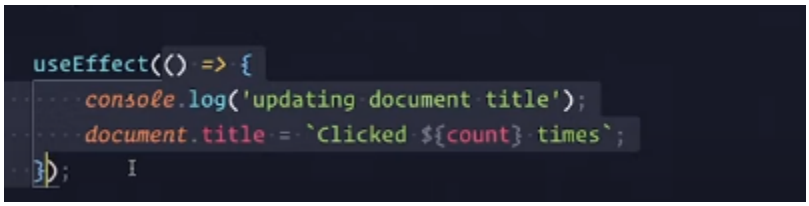
Uporer kaj ta Jodi amra function component a korte chai:

Functional component a lifecycle method nai.

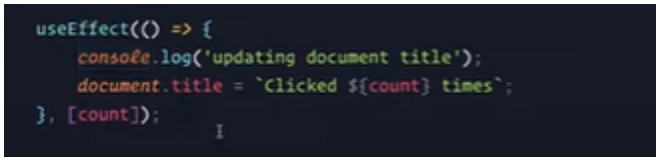


UseEffect er bitore je function ta proti ta render a run hoi. But componentDidMount Shudu first time jokhon component mount hoto tokhon run hoto.

<u>UseEffect render Conditionally:</u>



UseEffect er second er ekta parameter ase. Se khane amra bole dite pari shudu matro kon jinish ta change hole she render korbe. Rerender korer shomoi bitor er function take call kore dai She. Akhon Adeo call korbe kin a she take control korer ekta way ase. 2nd parameter k amra ekta array dite pari. And ai array er modda amra bolte pari kon jinish ta change hole amder ai useEffect er jinish ta change hote pare.



Life Cycle method ekta use kora geleo useEffect je hutu ekta simple function tai bar bar call kora or use kora jai.

```
useEffect(() => {
    console.log('updating document title');
    document.title = `Clicked ${count} times`;
}, [count]);

useEffect(() => {
    console.log('starting timer');
    const interval = setInterval(tick, 1000);

    // do the cleanup - stop the timer
    return () => {
        console.log('component unmounted');
        clearInterval(interval);
    };
}, []);
```

2nd useEffect ta prothom parameter a ekta function niyase ai function ta ekta kisu return o korte pare So akhon jodi amra return kori sheta componentWillUnMount er kaj kore.

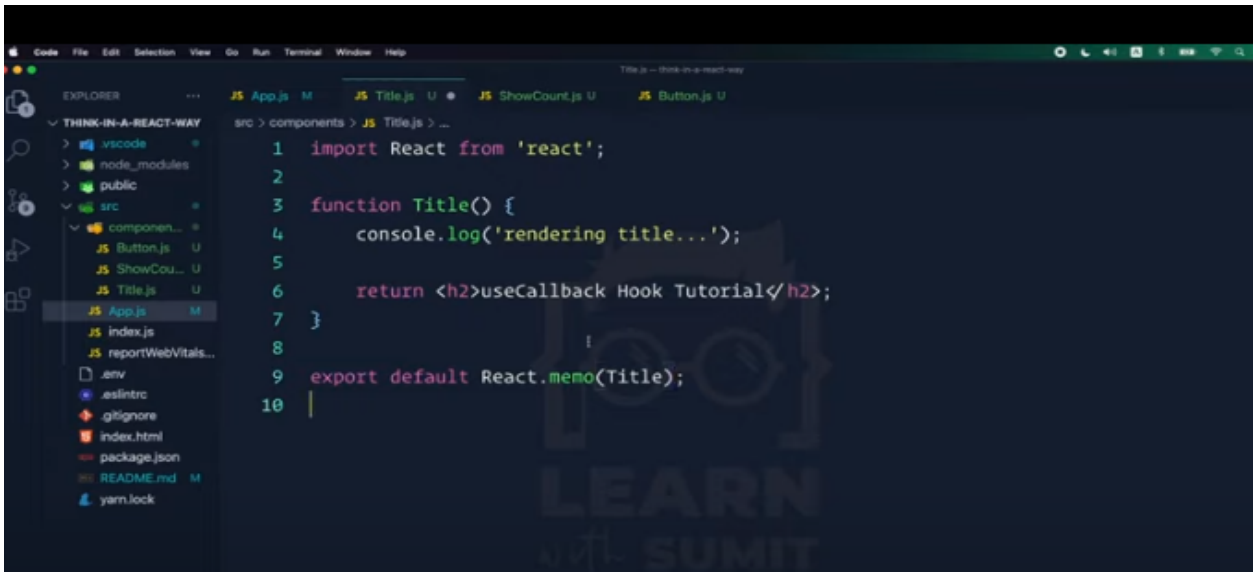
ai khane setInterval ta cholte e thakbe So akhon return korate er cholte thakbe na.

Array er bitore diya dite hobe jodi local variable na hai she jamon count k diya desi.

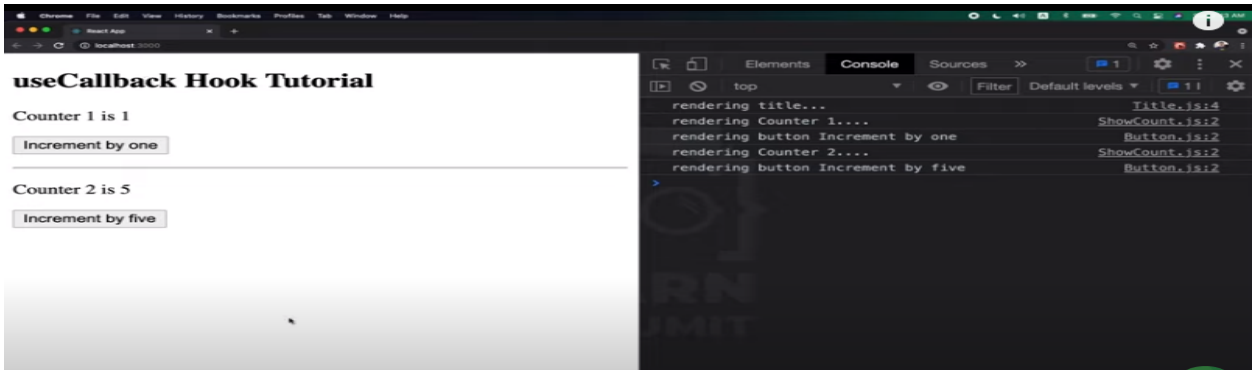
UseCallback & useMemo Hooks

Why react.memo?

Memo performance optimization a help kore.



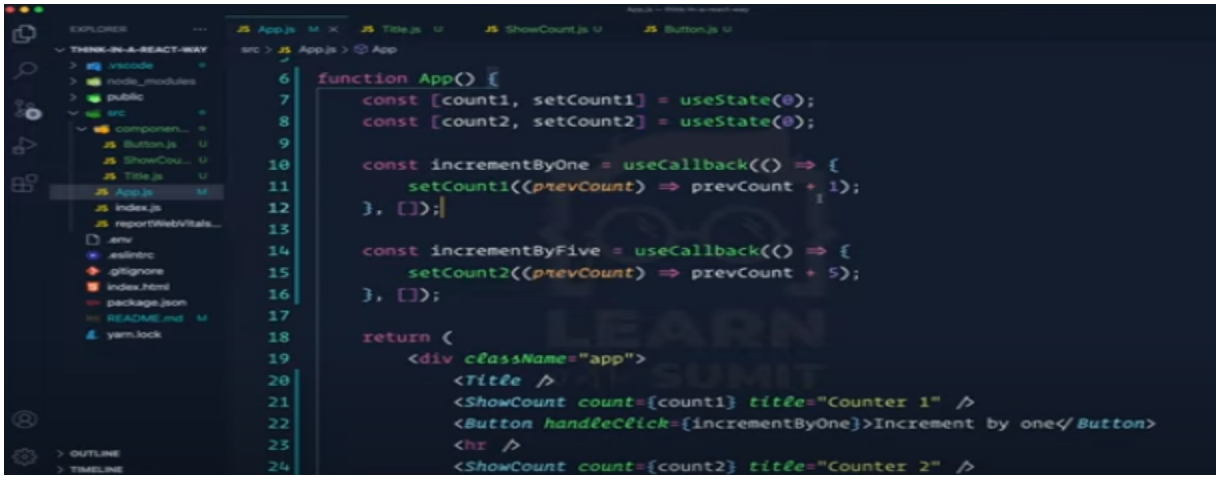
Ai khane title to bar bar rerender hower dorkar nai. React.memo er modda title take rape kore dilam. Aita kore dile e react amder component take cash kore felbe mane memorize kore felbe.



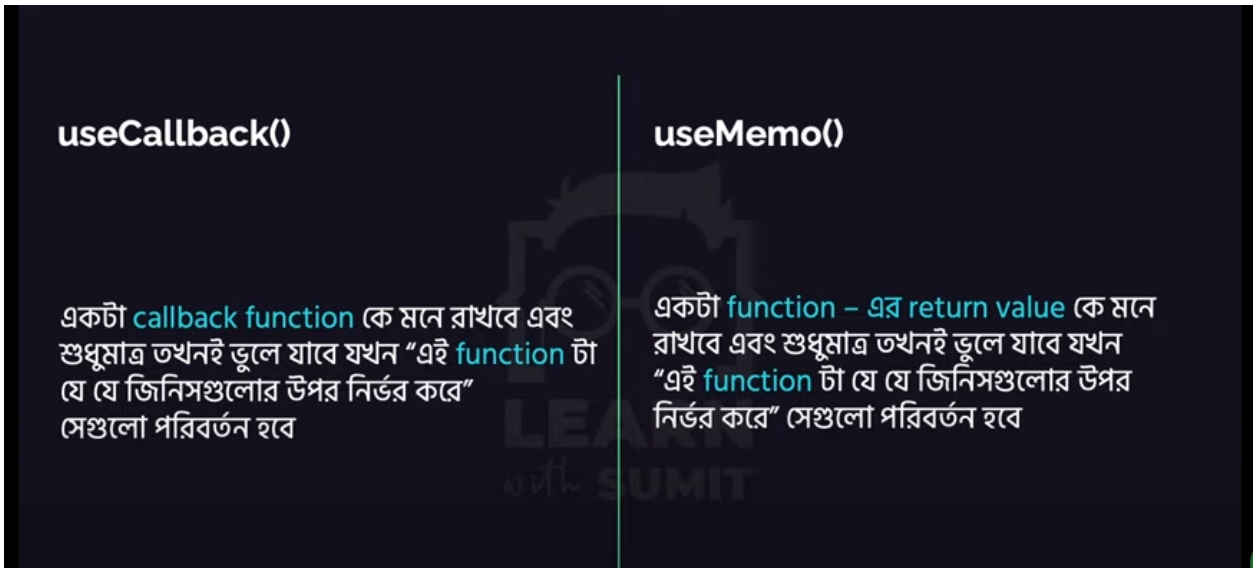
Ami jokhon button a click korsi ai button gula jodi onek expensive kaj kore thake and ai component gula jodi click er shate shate bar bar rerender hoi sheta ashole oproijonio.

Mane Button 1 change hosse er shate jodi button 2 o ojotha change hoiya jai shudu shudu aita ashole bukami. React.memo kono hook na.

String primitive value, object reference value,function o js a ekta object jeta reference type. React.memo diya optimized ekta component er rerender jodi amra thakate chai & ai khane function reference er problem jodi shorate chai she jonno use korte hobe useCallback hook. Callback function mone rakhe. Shob jaigai usecallback or usememo use kora jabo na. Chutto ekta compnent bar bar rerender hole kono problem nai. React.memo ai function tar o to nijjasho cost ase. tai React.memo function er chaya jodi component notun kore rerender er cost bashi hoi tokhon e amra React.memo use korbo.



Je function ta memorize korte hobe useCallback() er bitor diya dibo 1st parameter hishabe and 2nd parameter a dependence gula bolte hobe kisu na thakle faka array diya rakbo. Akhon react ai jinish take mone rakbe shob shomoi karon blank array kono dependency nai.




```
function App() {

    const [count1, setCount1] = useState(0);

    const [count2, setCount2] = useState(0);


    const incrementByOne = useCallback(() => {

        setCount1((preCount) => pevCount + 1)

    }, []);


    const incrementByFive = useCallback(() => {

        setCount2((preCount) => preCount + 5 )

    }, []);

}
```

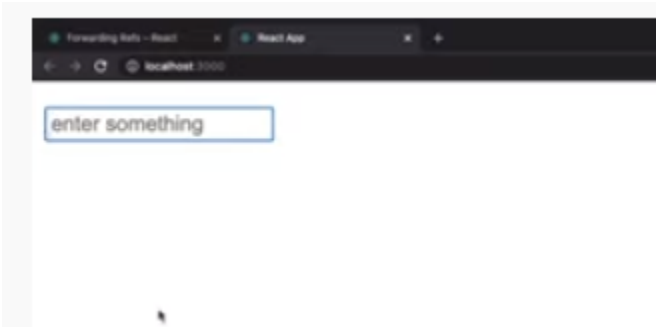
2nd parameter mane dependency change hole she vule jabe mone rakbe na.

React Hook useRef, forwardRef

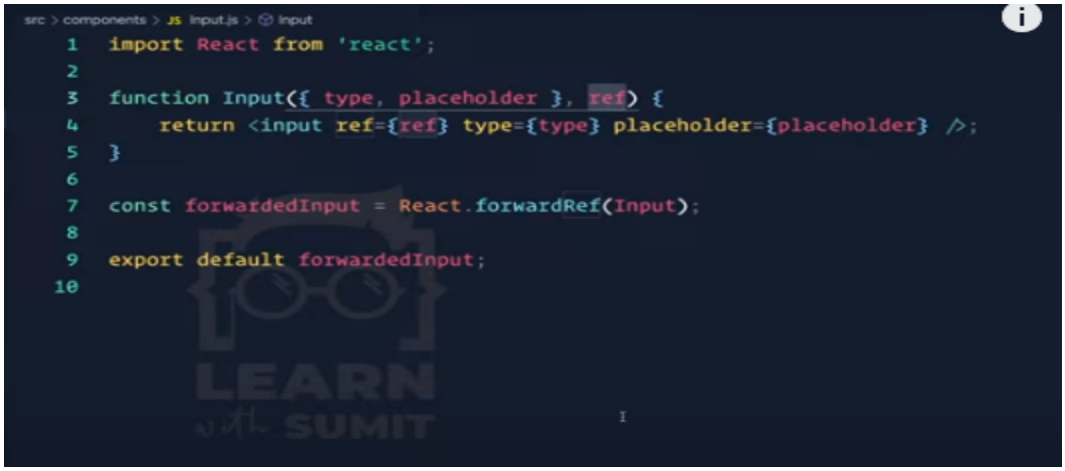
Dom Manipulation React a jodi nije kora lage eta thik na. Jamon document.getElementById, document.querySelector egulu joto bashi para jai avoid kora e better. React Dom er upor control ta nije e maintain kore.

Onek shomoi kisu kisu situation chole ashe Dom a haat dite e hoi. But sheta o react er maddome na. Amra eta shate or props change kore kori. Er er onujai e react nije e Dom er kaj kore fele.Kintu amon situtation o asbe jekhane Dom er upor amder kisu kaj kora lagbe. Jar shate state/props change er shomporko na o thakte pare. She khatre amra useRef use korbo.

Output:



ForwardRef:



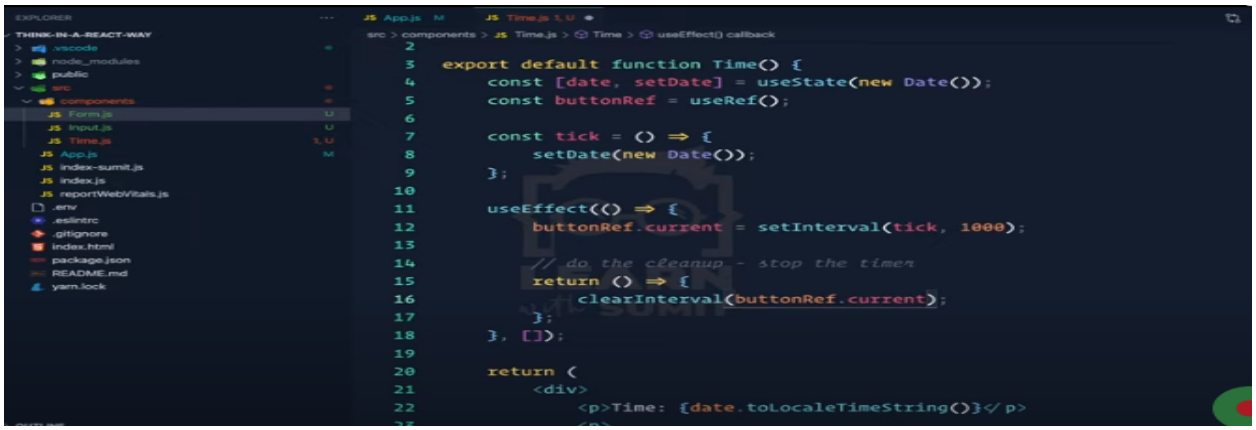
UseRef k er ek vabe o use kora jai:

Suppose cleanUp ta button a click korle hobe.

React a jamon state change hole rerender hoi tamon useRef a o hoi.

useRef muloto jahutu Dom elememt a excess dite ashse. useRef jetake return kore sheta amder je object ta dai shetar current er modda react element take dukiya diyase. And ai inputRef.current er modda chile amra o amder value dukate pari.

Button er modda useRef use: useRef k ekta storage hishabe use korbo.



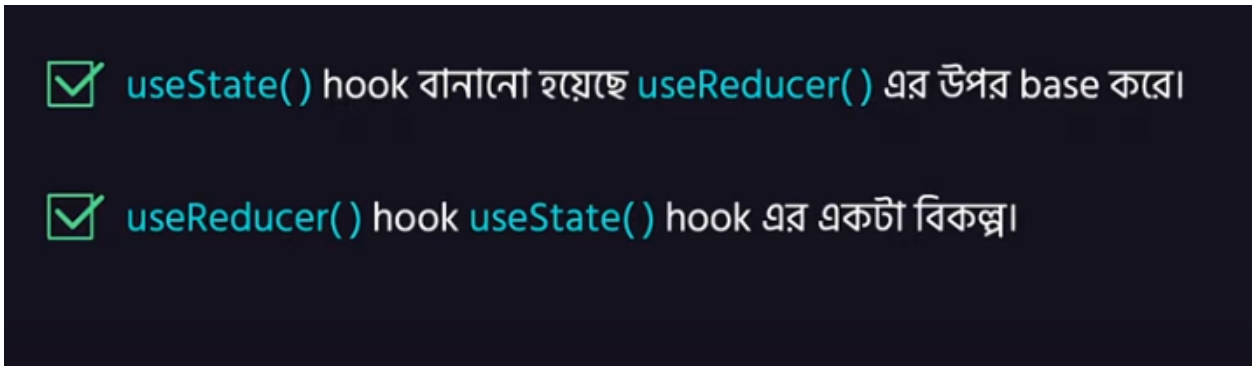
output:

useRef call korer por amra je jinish ta pai shei jinish ta ekta special type of jinish mane buttonRef jeta change hole o ai component ta rerender hobe na & jodi ai component ta onno kono karone rerender hoi state change or props change er karone tahole ager je reference ta sheta roiya jabe tar mane amr je interval ta jeta perfect candidate karon amr interval ta store kore rakha dorkar jeno ami porobotite sheta cleanup korte pari shetai kintu hosse ai khane.So amra useRef k storage hishabe use korlam.

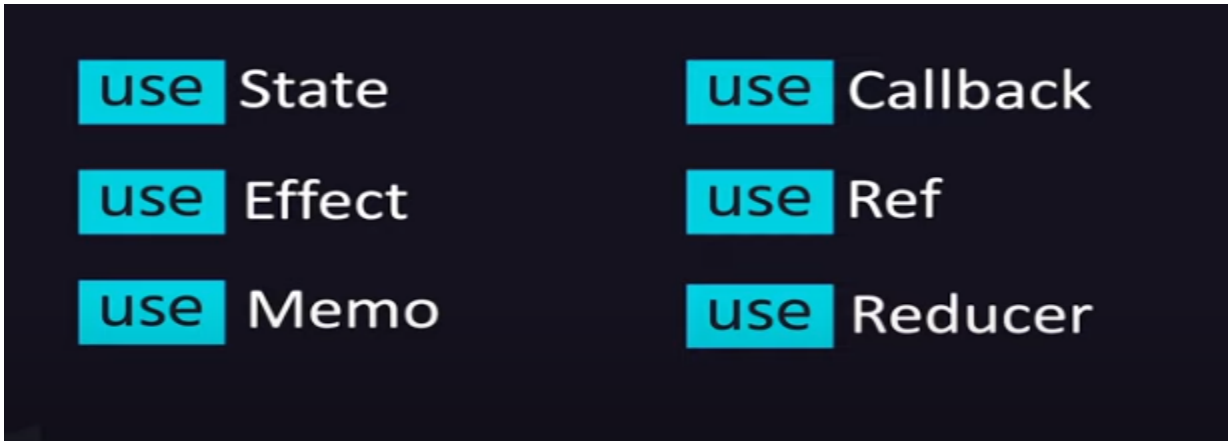
React useReducer() Hook

useReducer ekta hook jeta state management korte bebohar hoi. Kintu amra already jani

useState hook state management korte babohar hoi.



Akhon proshno holo 2 ta hook er modda pathokko ki & kokhon konta babohar korbo?



Ai khane pottek ta word diya e buja jasse konta kon purpose a use hoi.

useState: useState use hoi state management er jonno.

useEffect: side effect niya kaj korer jonno.

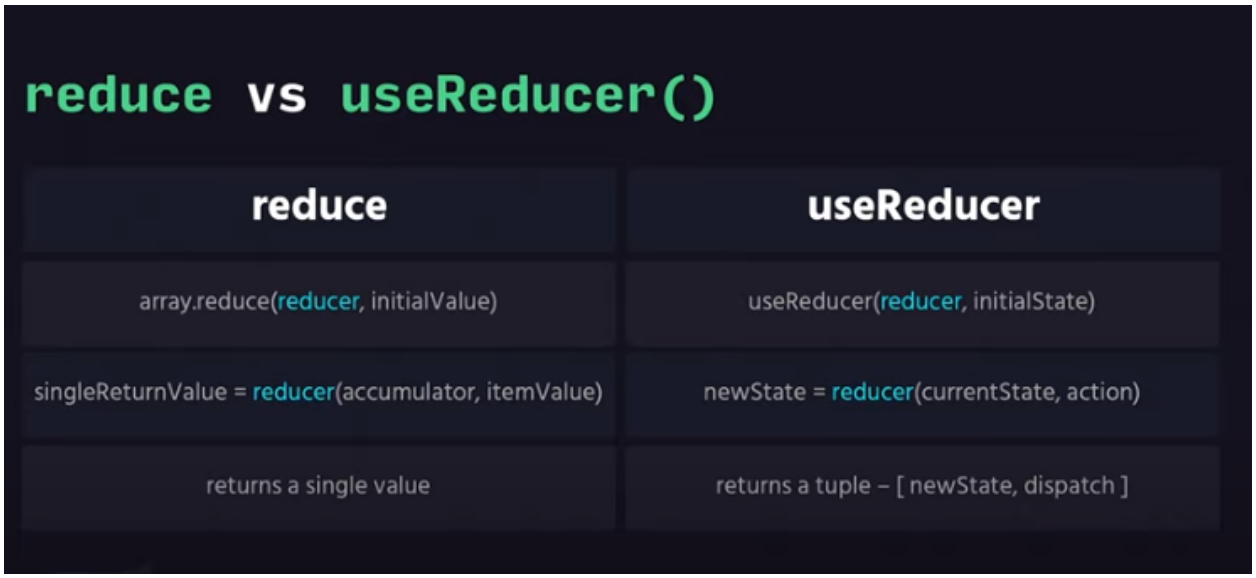
useMemo: Memorization niya kaj korer jonno.

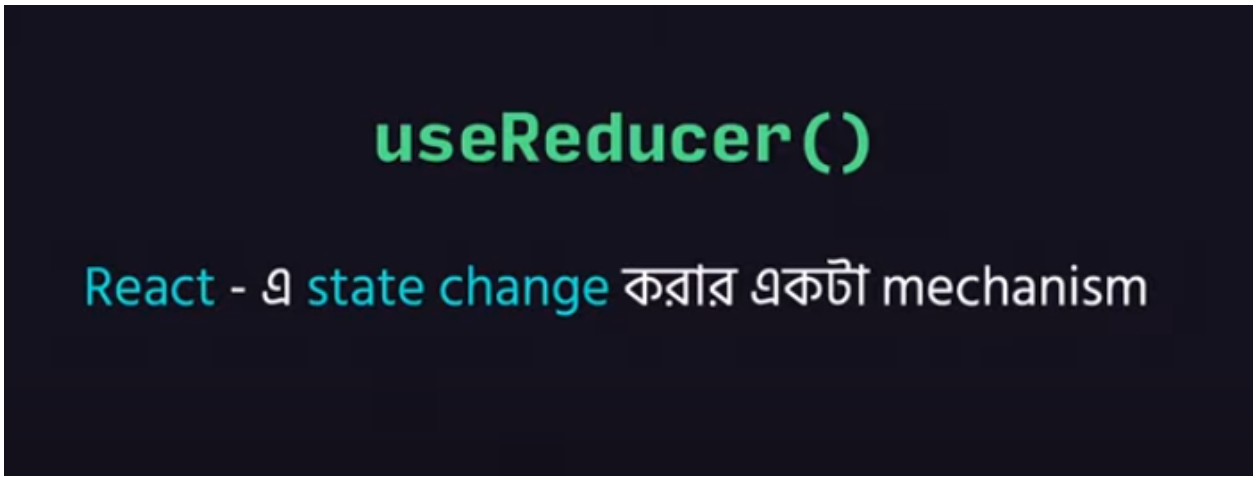
use Callback: Callback hosse callback function niya kaj korer jonno.

use Ref: reference niya kaj korer jonno.

use Reducer: Reducer niya kaj korer jonno.

Reducer react er kisu na aita JS er ekta jinish. So Akhon bujbo js a reducer ta ki?





useReducer Example1:

In VS code Install ESLint & prettier

- Go to setting search default formatter select prettier
- Workplace setting.json:

```
{
"editor.formatOnSave": true,
"editor.codeActionsOnSave": {
    "source.fixAll.eslint": true,
    "source.fixAll.tslint": true,
    "source.organizeImports": true
},
"javascript.validate.enable": false,
"typescript.validate.enable": false
}
```