### Data is the new oil🔥

1. **Two Layer in React JS**

* UI Layer — JSX, Tailwind CSS
* Data Layer — State and Props (pass data from one component to another)
* Parcel bundles both layers

**2. Props Drilling**

Prop drilling is **a situation where data is passed from one component through multiple interdependent components until you get to the component where the data is needed**.

Parent -> children

**3. React dev tools extension**

[**React Developer Tools**  
Adds React debugging tools to the Chrome Developer Tools. Created from revision 47f63dc54 on 12/6/2022.chrome.google.com](https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi?hl=en)

**4. State lifting?**

If two child has different state, we cannot modify a child state of another child state

we can maintain the child state in its parent

Take the control of child and give it to the parent (state lifting)

**5. Profiler**

Profiler helps us in performance testing

Available in React Developer Tools

**6. UseContext Hook**

Any comp can use data (sharing data)

Can i have multiple UserContext? Yes

Utils --> UserContext.js  
  
import {createContext} from "react"  
  
const UserContext = createContext({  
name: "Jasmine",  
email: "jasmine@gmail.com"  
})  
  
export default UserContext

import UserContext from "../utils/UserContext"  
import {useContext} from "react"  
  
consr {user} = useContext(UserContext)  
  
  
{user.name}

7.Class Comp

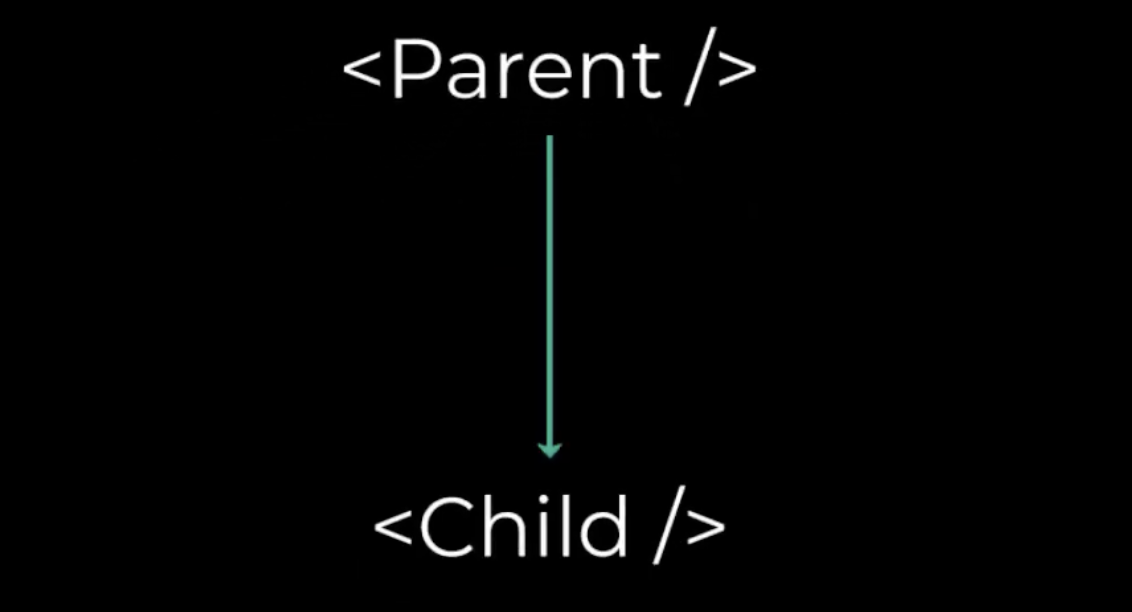
import UserContext from "../utils/UserContext"  
  
  
<UserContext.Consumer>  
  
{(value) => console.log(value);  
  
</UserContext.Consumer>

8. vscode extensions

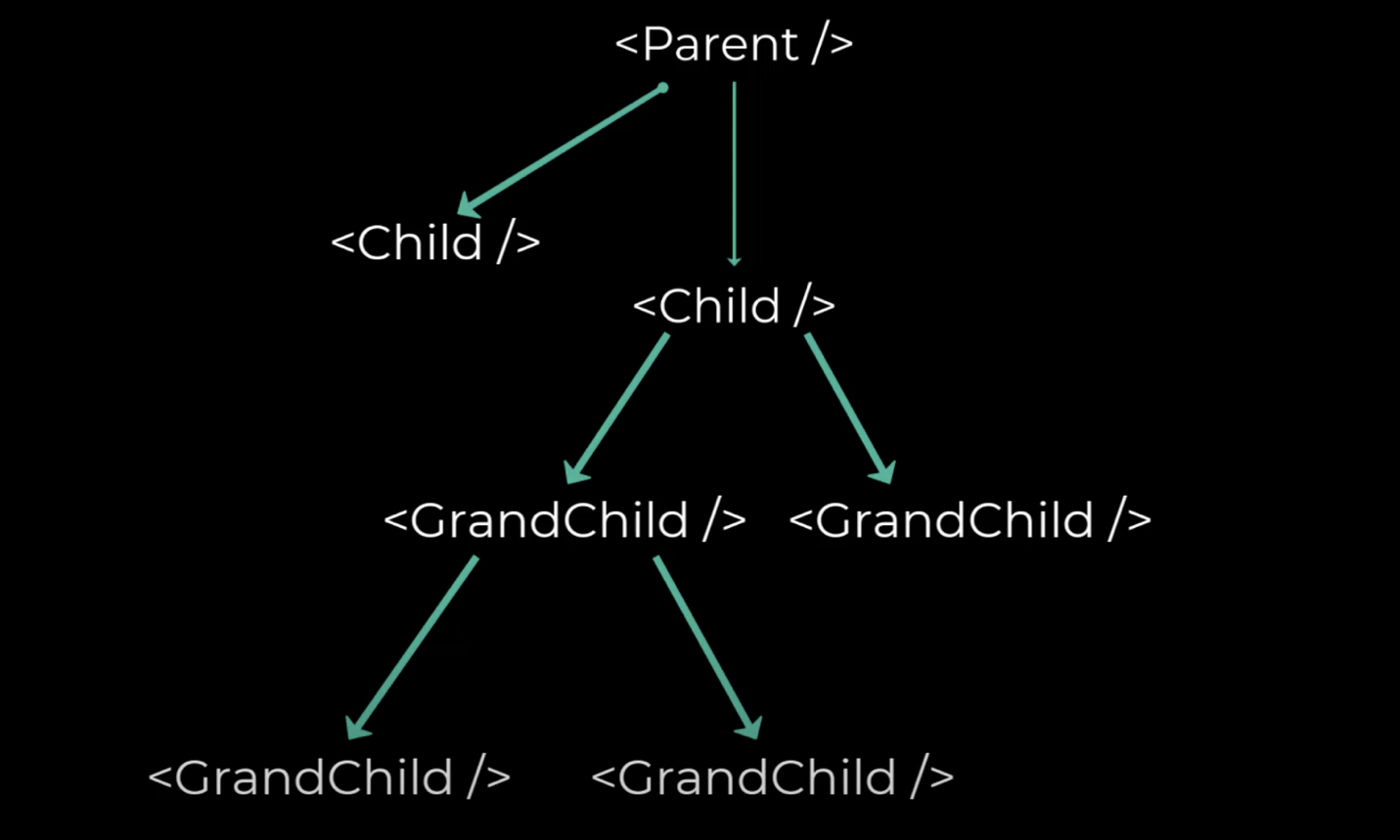
<https://resource.jsmasterypro.com/vsc-extensions>

**React Hooks — useContext Hook 🔥**

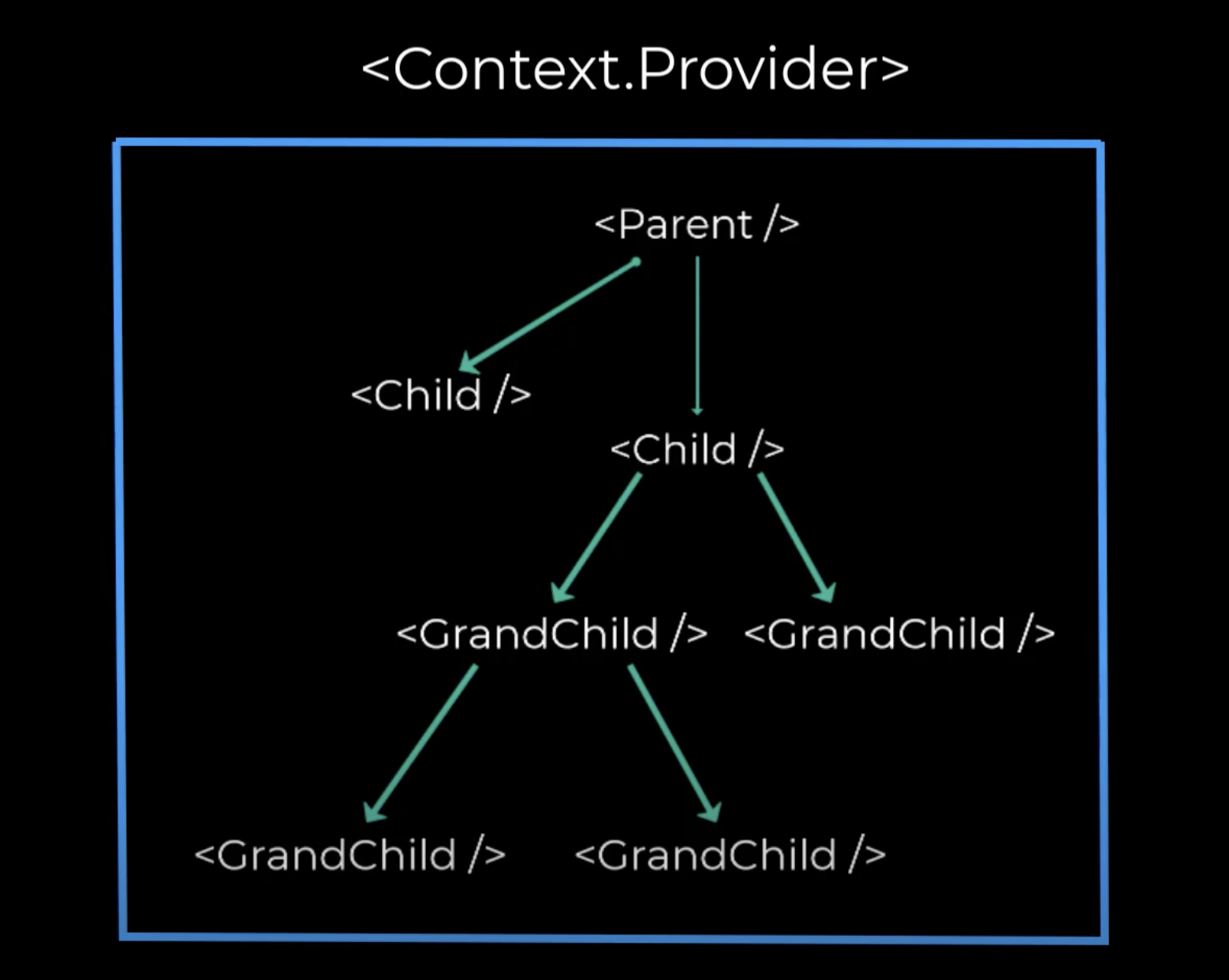
React uses useState hook to manage the state of a single component. React uses props to pass data from parent to child component.



What if it is multiple levels deep (Component Tree) ?



React Context is a way to manage state globally. useContext Hook **provides a way to pass data or state through the component tree without having to pass props down manually through each nested component**.



React uses context api to provide state from parent component to children no mater how nested the child component is.

useContext can be used together with the useState Hook to share state between deeply nested components more easily than with useState alone.

**Example**

App component

import React from "react"  
import Nav from './Nav'  
  
function App() {  
return (  
<>  
<Nav/>  
<h1> Signed In </h1>   
</>  
 )  
}

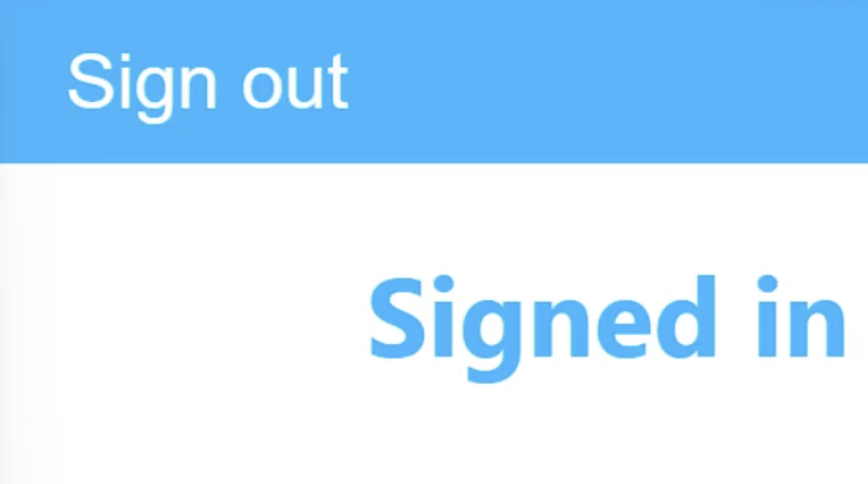
Nav component

import React from "react"  
import Btn from './Btn'  
  
export default function Nav() {  
return (  
<div className="nav">  
<Btn/>  
</div>  
 )  
}

Btn component

import React, {useState} from "react"  
import Btn from './Btn'  
  
export default function Btn() {  
  
const [signedIn, setSignedIn]  
return (  
<button onClick= {() => setSignedIn(!setSignedIn) }>  
{signedIn ? 'Sign out' : 'Sign in'}  
</button>  
 )  
}

Only Child changes. Not the Parent.



App component

import React from "react"  
import Nav from './Nav'  
  
function App() {  
return (  
<>  
<Nav/>  
<h1> Signed In </h1>   
</>  
 )  
}

Nav component

import React from "react"  
import Btn from './Btn'  
  
export default function Nav() {  
return (  
<div className="nav">  
<Btn/>  
</div>  
 )  
}

Btn component

import React, {useState} from "react"  
import Btn from './Btn'  
  
export default function Btn() {  
const [signedIn, setSignedIn]  
return (  
<button onClick= {() => setSignedIn(!setSignedIn) }>  
{signedIn ? 'Sign out' : 'Sign in'}  
</button>  
 )  
}