Lecture No 10

```
npm start -> react-scripts start
                                        -> index.js
                                                       -> index.html
                                                                      (Render)
```

npm start ---> WebPack(Bundler) ---> Babel(JS Compiler) ----> Website launch

Newer JS Code ---> Babel(JS Compiler) ----> Older JS Code

Expression Evalulator

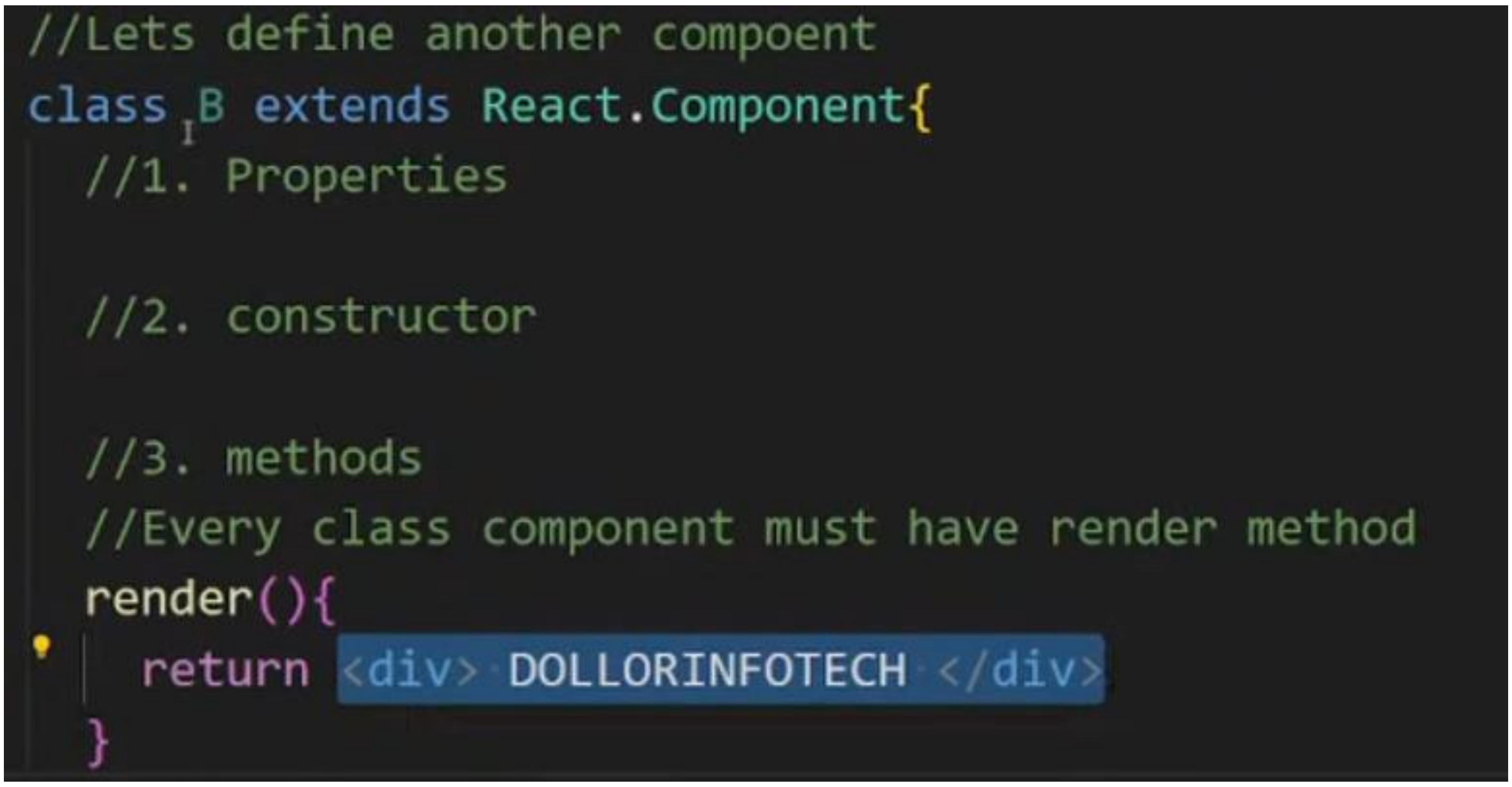
`` `${}` `` Pure JS

{} ReactJs

Import/Export

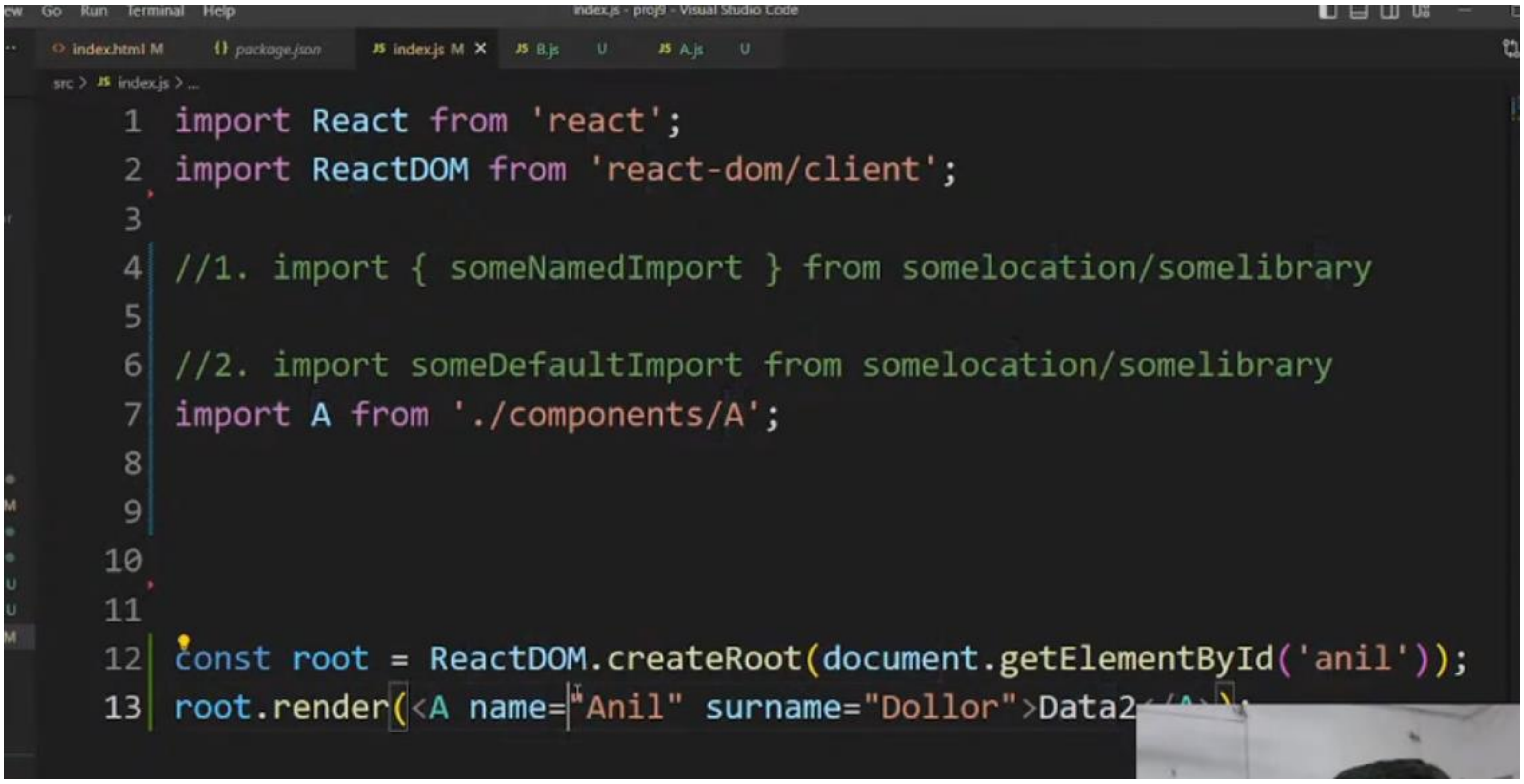
Every Component can have its own data

1. Inside
2. Outside <------- Some APIs



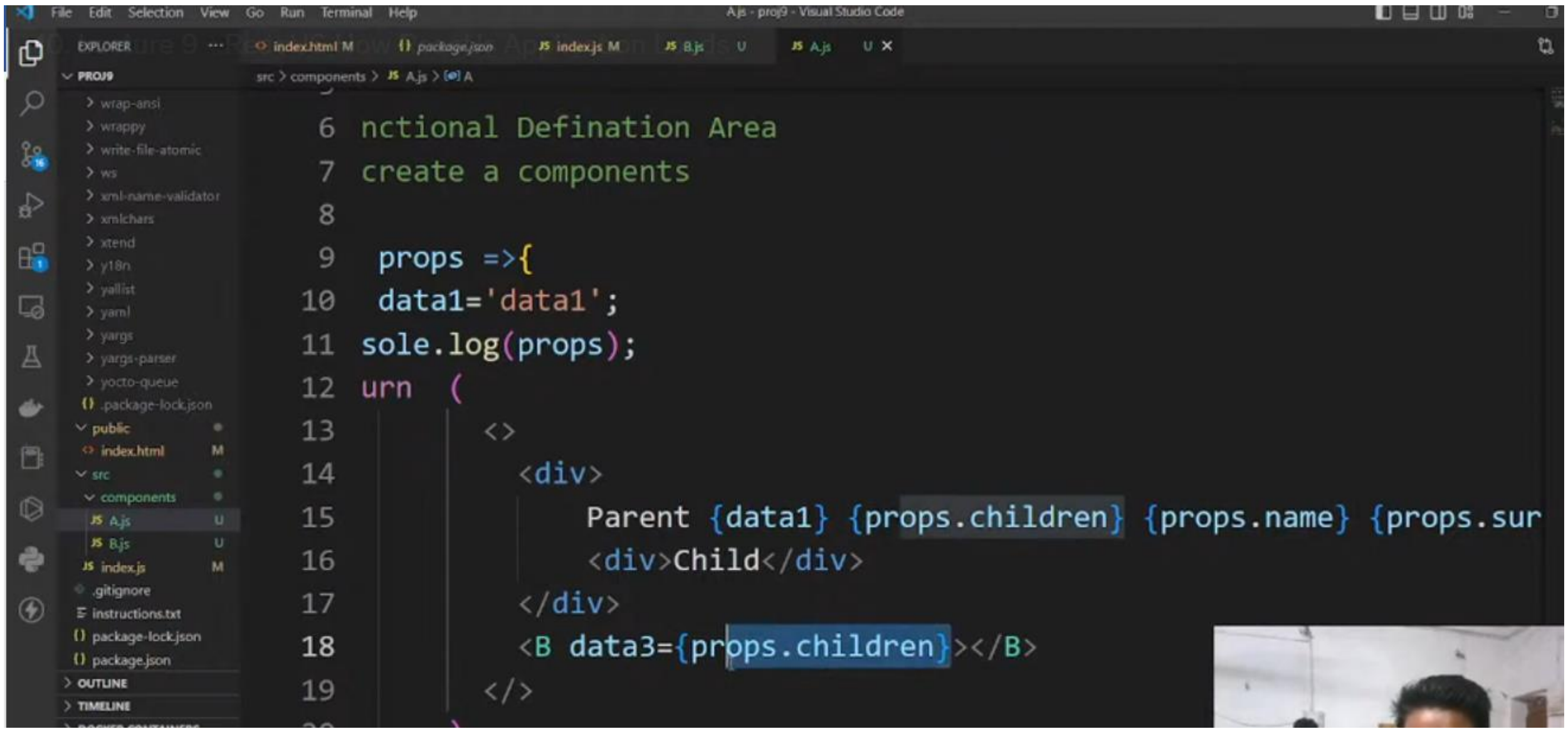
```
//Lets define another compoent
class B extends React.Component{
  //1. Properties

  //2. constructor

  //3. methods
  //Every class component must have render method
  render(){
    return <div> DOLLORINFOTECH </div>
  }
```

Data flow from one class compt to other :



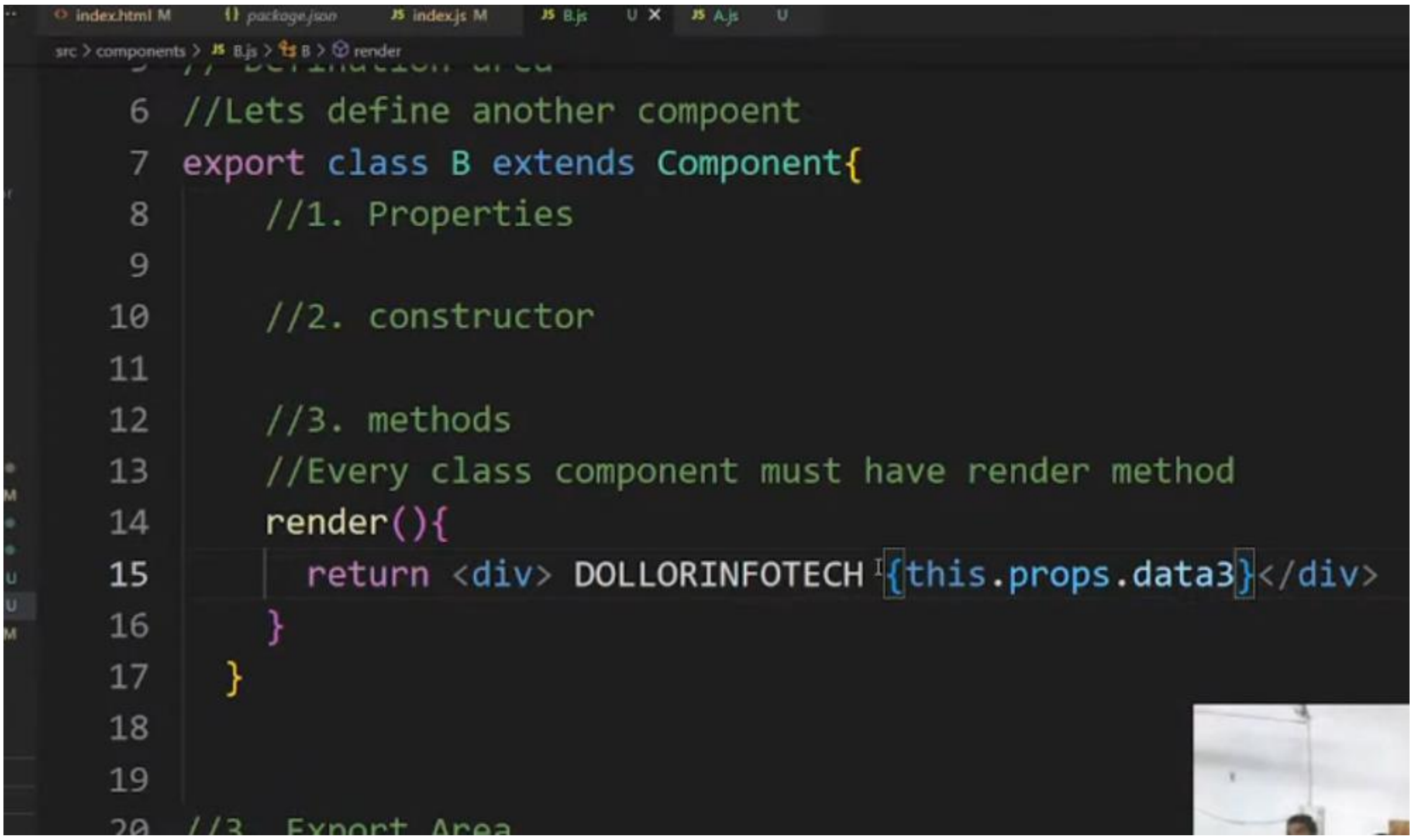
```
import React from 'react';
import ReactDOM from 'react-dom/client';

//1. import { someNamedImport } from somelocation/somelibrary

//2. import someDefaultImport from somelocation/somelibrary
import A from './components/A';




const root = ReactDOM.createRoot(document.getElementById('anil'));
root.render(<A name="Anil" surname="Dollor">Data2
```

"A" compt se me data behj raha hu name="Anil" jab A ko invoke kr rah hu index.js me (kahen bhi kr skty hen hen invoke is compt ko), ab isko me "A" k ander receive kro ga as a prop.



```
nctional Defination Area
create a components

 props =>{
 data1='data1';
sole.log(props);
urn  (
       <>
         <div>
             Parent {data1} {props.children} {props.name} {props.sur
             <div>Child</div>
         </div>
         <B data3={props.children}></B>
       </>
```

Ab "A" me ham 'B' ko invoke kr rahy hen, or us "B" me ham us name=anil ko pass kr dety hen or B ki definition me receive krke use kr lety hen .



```
//Lets define another compoent
export class B extends Component{
    //1. Properties

    //2. constructor

    //3. methods
    //Every class component must have render method
    render(){
      return <div> DOLLORINFOTECH {this.props.data3}</div>
    }
  }

//3. Export Area
```

In React class components, this.props ka use karna zaroori hai kyunki props class component ke context me this ke andar hoti hain.

Explanation:

1. **Class Components**:
   - Jab aap ek class component banate hain, toh aapko this keyword ka istemal karna padta hai.
   - this ek reference hota hai current instance ka class ka.
2. **Props**:
   - props wo data hota hai jo parent component se child component ko pass kiya jata hai.
   - Class component me props ko access karne ke liye, aapko this ka istemal karna padta hai.
3. **Usage**:
   - Jab aap this.props.x likhte hain, toh iska matlab hai ki aap props object me se x property ko access kar rahe hain, jo ki current class instance ka part hai.

Agar aap functional components ka use karte hain, toh aap props ko directly function ke parameters ke roop me access kar sakte hain, bina this ke.