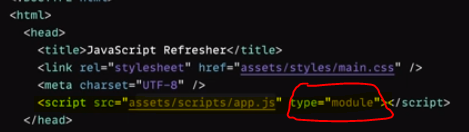
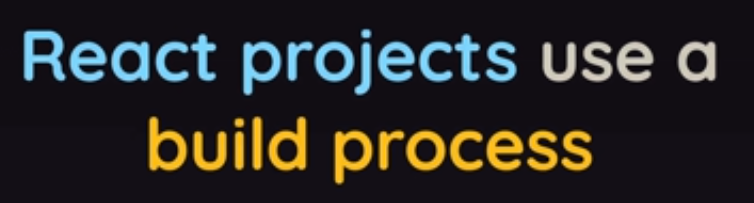
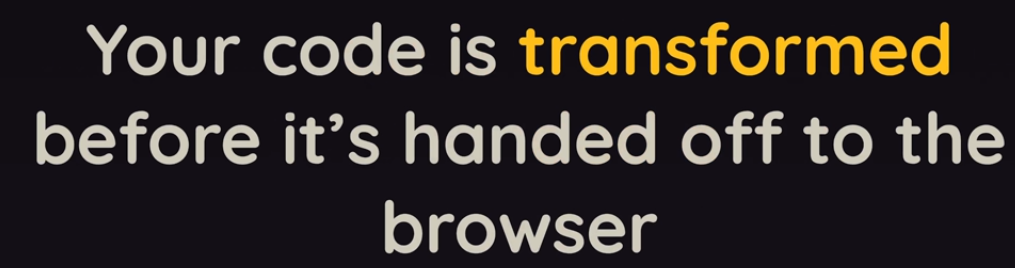


In header in <script></script>we need to use src.



So that tha the javascript file will be treated as module.

So that we can import from js file another js file easily  




Let scope

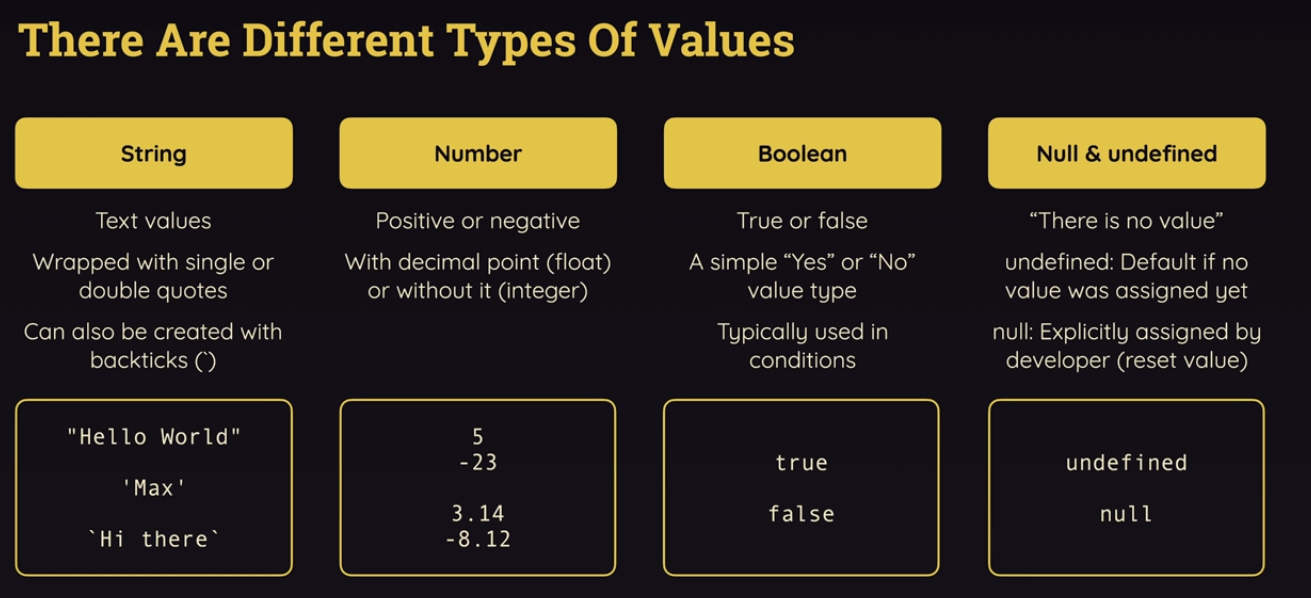
Function hello(){

Let say=”hello”;

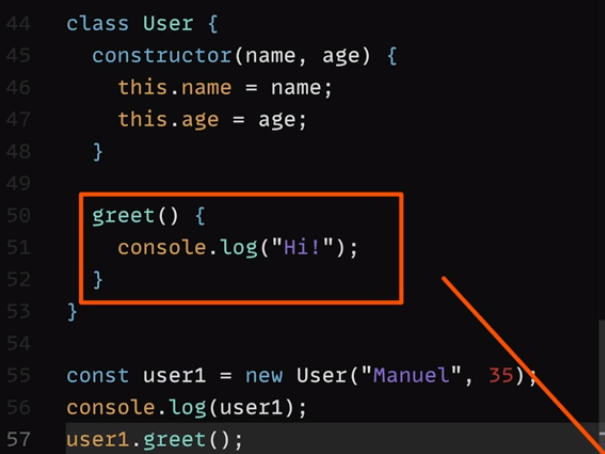
}

Console.log({say});---we get the error here.scope is here upto function only.

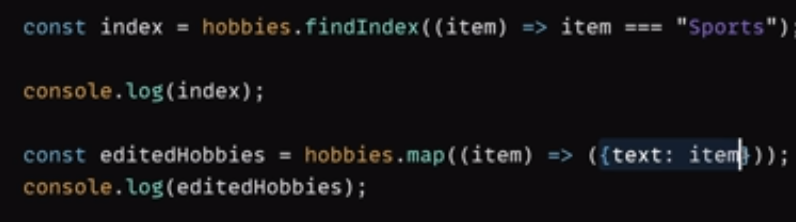




Javascript class



Returning js object by using map



**Section 3: React Essentials**

npm install

npm run dev

**PS D:\UI-code\Udemy-maxschwarzmueller\react-complete-guide-course-resources\code\03 React Essentials> npm install**

**npm ERR! code ENOENT**

**npm ERR! syscall open**

**npm ERR! path D:\UI-code\Udemy-maxschwarzmueller\react-complete-guide-course-resources\code\03 React Essentials/package.json**

**npm ERR! errno -4058**

**npm ERR! enoent ENOENT: no such file or directory, open 'D:\UI-code\Udemy-maxschwarzmueller\react-complete-guide-course-resources\code\03 React Essentials\package.json'**

**npm ERR! enoent This is related to npm not being able to find a file.**

**npm ERR! Enoent**

**Solution:**

<https://stackoverflow.com/questions/45531633/npm-warn-enoent-enoent-no-such-file-or-directory-open-c-users-nuwanst-packag>

Have you created a **package.json** file? Maybe run this command first again.

C:\Users\Nuwanst\Documents\NodeJS\3.chat>npm init

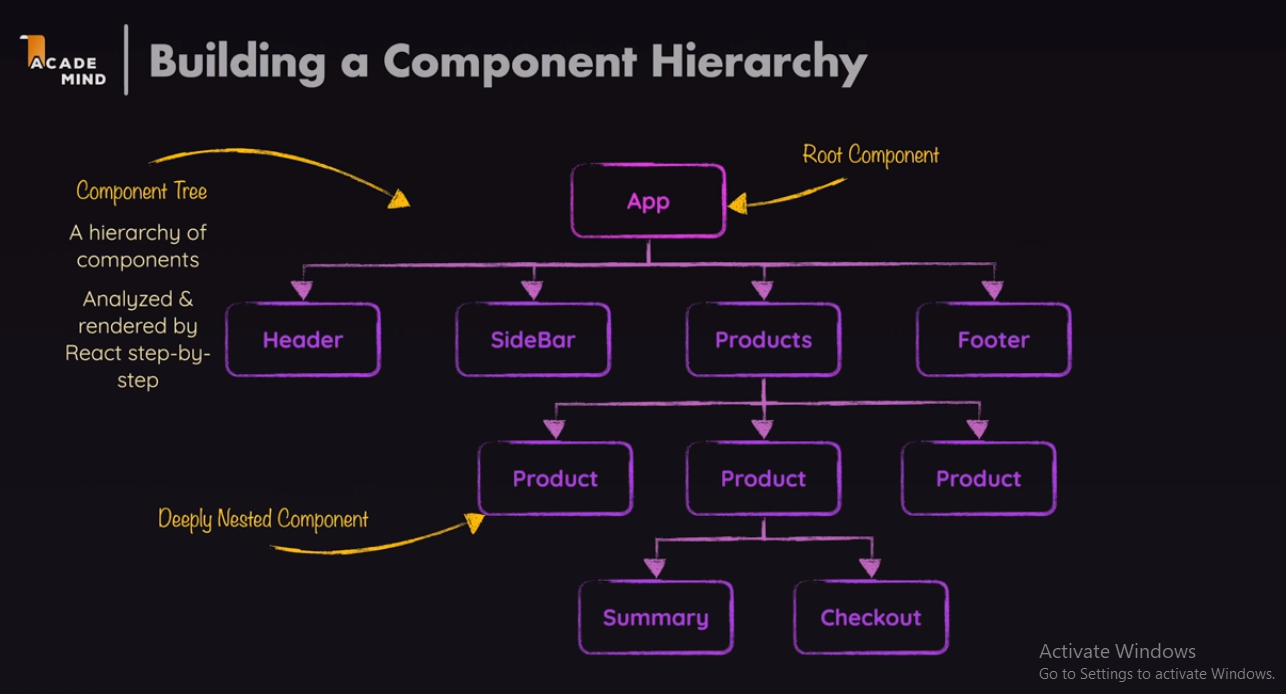
It creates a **package.json** file in your folder.

Then run,

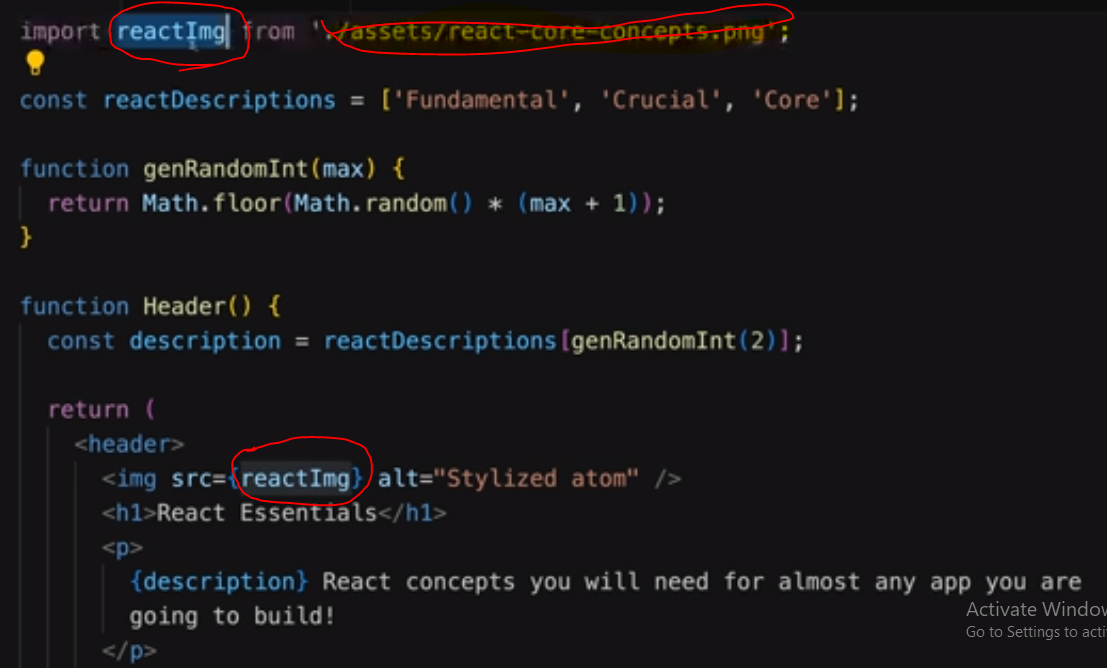
C:\Users\Nuwanst\Documents\NodeJS\3.chat>npm install socket.io --save

The --save ensures your module is saved as a dependency in your **package.json** file.

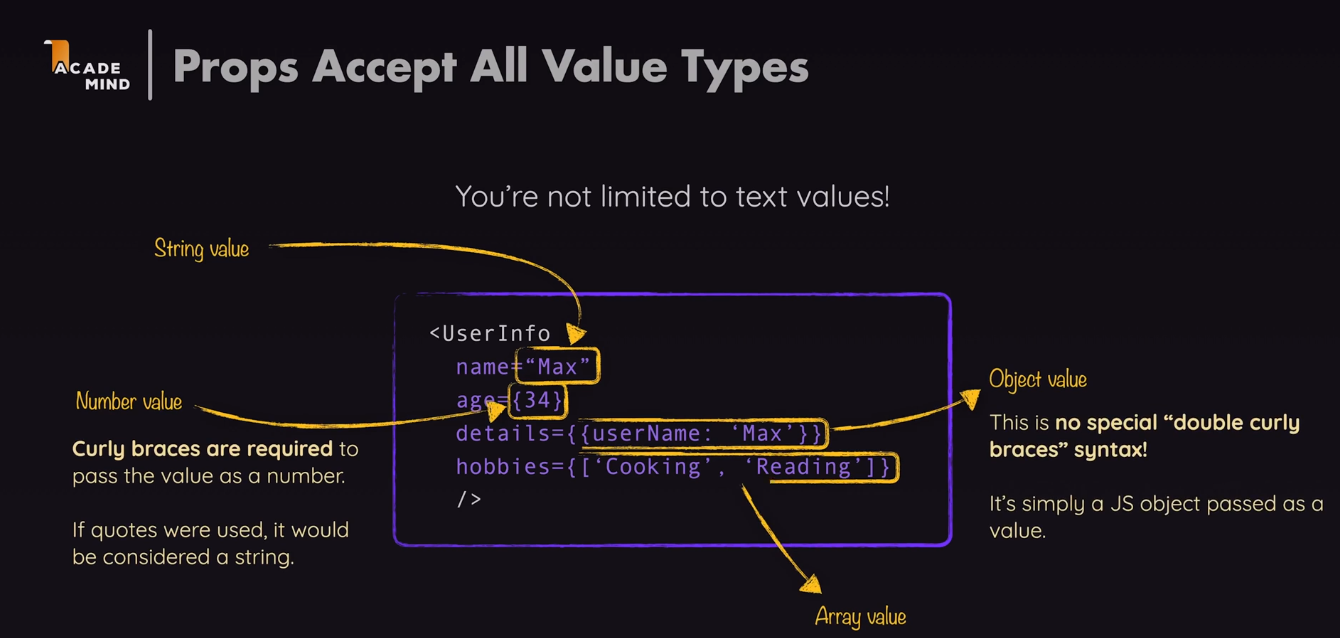
Let me know if this works.



How to import a image in jsx fle dynamically?



43. making components reusable with **props**



function App() {

  return (

    <div>

      <Header />

      <main>

        <section id="core-concepts">

          <h2>Core Concepts</h2>

          <ul>

            <CoreConcept

              title="Components"

              description="The core UI building block."

              image={componentsImg}

            />

            <CoreConcept title="Props" />

            <CoreConcept />

            <CoreConcept />

          </ul>

        </section>

      </main>

    </div>

  );

}

import reactImg from './assets/react-core-concepts.png';

import { CORE\_CONCEPTS } from './data.js';

const reactDescriptions = ['Fundamental', 'Crucial', 'Core'];

function genRandomInt(max) {

  return Math.floor(Math.random() \* (max + 1));

}

function Header() {

  const description = reactDescriptions[genRandomInt(2)];

  return (

    <header>

      <img src={reactImg} alt="Stylized atom" />

      <h1>React Essentials</h1>

      <p>

        {description} React concepts you will need for almost any app you are

        going to build!

      </p>

    </header>

  );

}

function CoreConcept({ image, title, description }) {

  return (

    <li>

      <img src={image} alt={title} />

      <h3>{title}</h3>

      <p>{description}</p>

    </li>

  );

}

function App() {

  return (

    <div>

      <Header />

      <main>

        <section id="core-concepts">

          <h2>Core Concepts</h2>

          <ul>

            <CoreConcept

              title={CORE\_CONCEPTS[0].title}

              description={CORE\_CONCEPTS[0].description}

              image={CORE\_CONCEPTS[0].image}

            />

            <CoreConcept {...CORE\_CONCEPTS[1]} />

            <CoreConcept {...CORE\_CONCEPTS[2]} />

            <CoreConcept {...CORE\_CONCEPTS[3]} />

          </ul>

        </section>

      </main>

    </div>

  );

}

export default App;

**48. component composition : The special children prop**

**Children data is , which available between opening and closing braces.**

**Componenets, jsx, propsand state from below**

    <section id="examples">

          <h2>Examples</h2>

          <menu>

            <TabButton>Components</TabButton>

            <TabButton>JSX</TabButton>

            <TabButton>Props</TabButton>

            <TabButton>State</TabButton>

          </menu>

        </section>

export default function TabButton({ children }) {

  return (

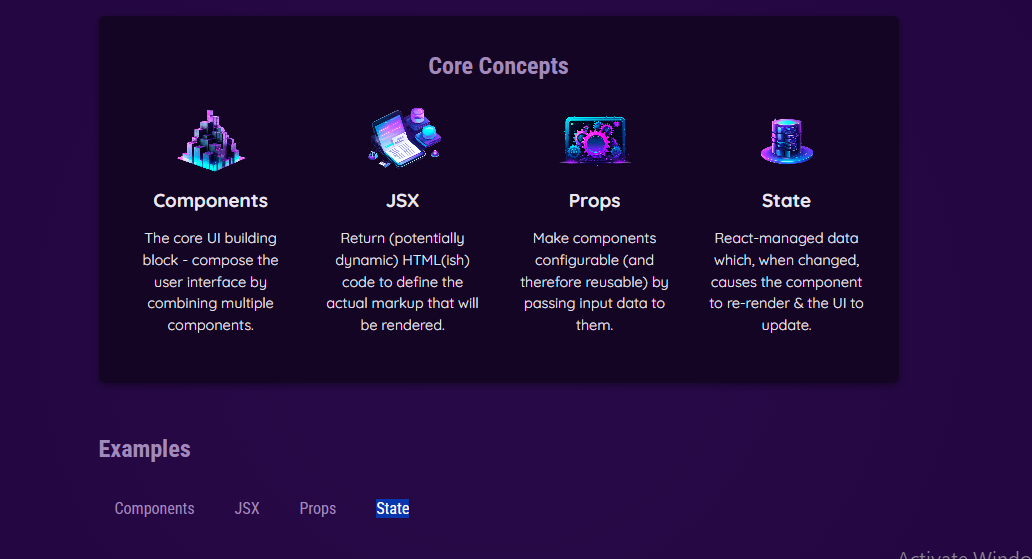
    <li>

      <button>{children}</button>

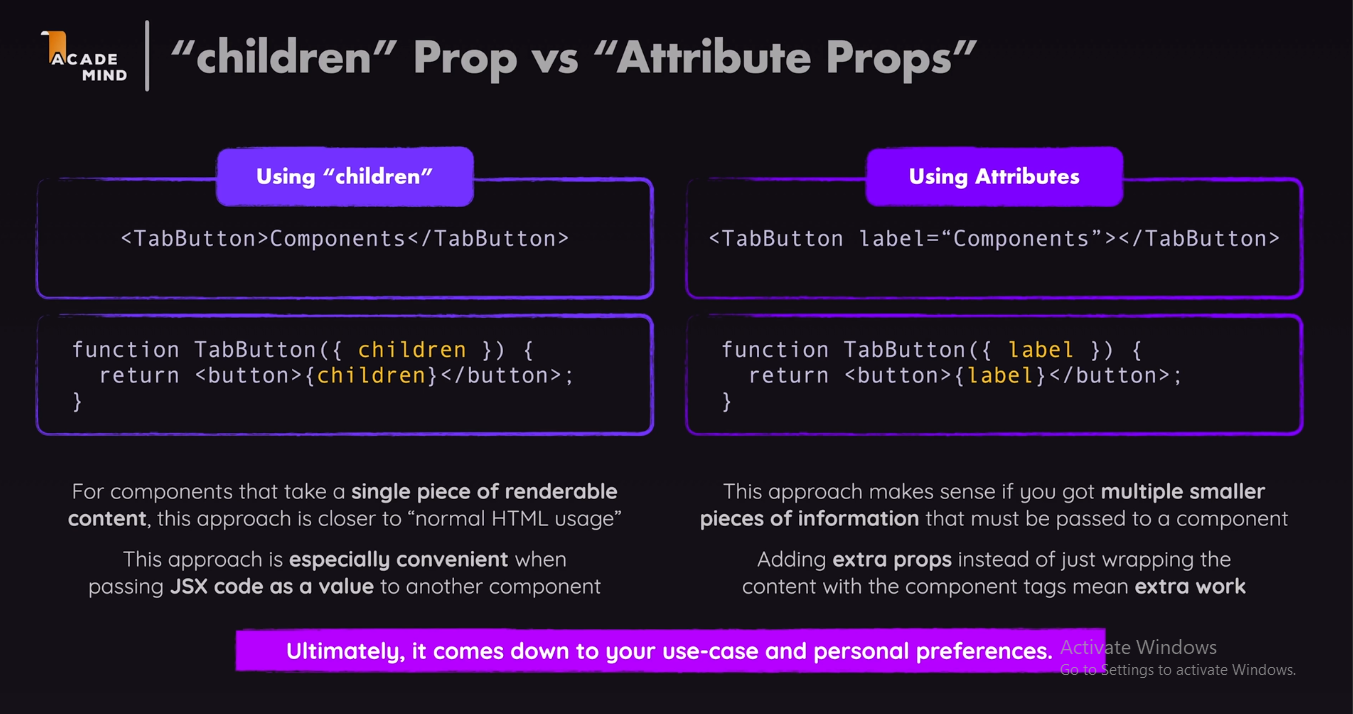
    </li>

  );

}



**Same can be done using lables**



**49. Reacting to events**

**How we react to events in venila java scripts.**

**Document.queryselector(‘button’).addEventListener(‘click’,()=>{})**

export default function TabButton({ children }) {

  function handleClick() {

    console.log('Hello World!');

  }

  return (

    <li>

      <button onClick={handleClick}>{children}</button>

    </li>

  );

}

**50. Passing functions as values to props.**

<section id="examples">

          <h2>Examples</h2>

          <menu>

            <TabButton onSelect={handleSelect}>Components</TabButton>

            <TabButton onSelect={handleSelect}>JSX</TabButton>

            <TabButton onSelect={handleSelect}>Props</TabButton>

            <TabButton onSelect={handleSelect}>State</TabButton>

          </menu>

          Dynamic Content

        </section>

function App() {

  function handleSelect() {

    console.log('Hello World - selected!');

  }

export default function TabButton({ children, onSelect }) {

  return (

    <li>

      <button onClick={onSelect}>{children}</button>

    </li>

  );

}

**51. Passing custom arguments to Event Functions.**

 <section id="examples">

          <h2>Examples</h2>

          <menu>

            <TabButton onSelect={() => handleSelect('components')}>

              Components

            </TabButton>

            <TabButton onSelect={() => handleSelect('jsx')}>JSX</TabButton>

            <TabButton onSelect={() => handleSelect('props')}>Props</TabButton>

            <TabButton onSelect={() => handleSelect('state')}>State</TabButton>

          </menu>

          Dynamic Content

        </section>

function App() {

  function handleSelect(selectedButton) {

    // selectedButton => 'components', 'jsx', 'props', 'state'

    console.log(selectedButton);

  }

**52. How not to update the UI?**

import { CORE\_CONCEPTS } from './data.js';

import Header from './components/Header/Header.jsx';

import CoreConcept from './components/CoreConcept.jsx';

import TabButton from './components/TabButton.jsx';

function App() {

  let tabContent = 'Please click a button';

  function handleSelect(selectedButton) {

    // selectedButton => 'components', 'jsx', 'props', 'state'

    tabContent = selectedButton;

    console.log(tabContent);

  }

  console.log('APP COMPONENT EXECUTING');

  return (

    <div>

      <Header />

      <main>

        <section id="core-concepts">

          <h2>Core Concepts</h2>

          <ul>

            <CoreConcept

              title={CORE\_CONCEPTS[0].title}

              description={CORE\_CONCEPTS[0].description}

              image={CORE\_CONCEPTS[0].image}

            />

            <CoreConcept {...CORE\_CONCEPTS[1]} />

            <CoreConcept {...CORE\_CONCEPTS[2]} />

            <CoreConcept {...CORE\_CONCEPTS[3]} />

          </ul>

        </section>

        <section id="examples">

          <h2>Examples</h2>

          <menu>

            <TabButton onSelect={() => handleSelect('components')}>

              Components

            </TabButton>

            <TabButton onSelect={() => handleSelect('jsx')}>JSX</TabButton>

            <TabButton onSelect={() => handleSelect('props')}>Props</TabButton>

            <TabButton onSelect={() => handleSelect('state')}>State</TabButton>

          </menu>

          {tabContent}

        </section>

      </main>

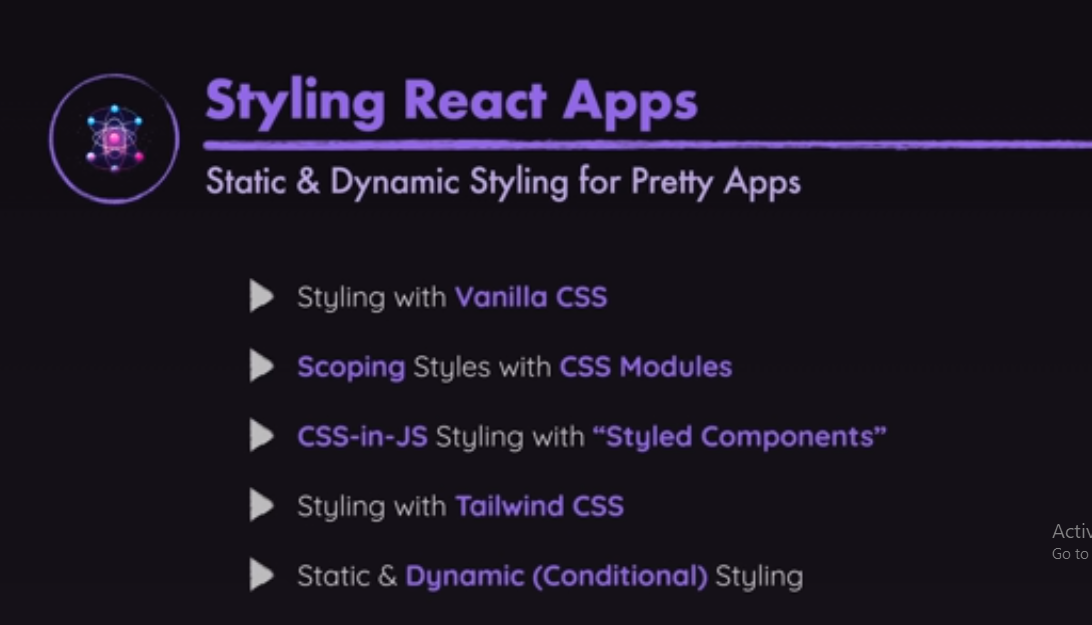
    </div>

  );

}

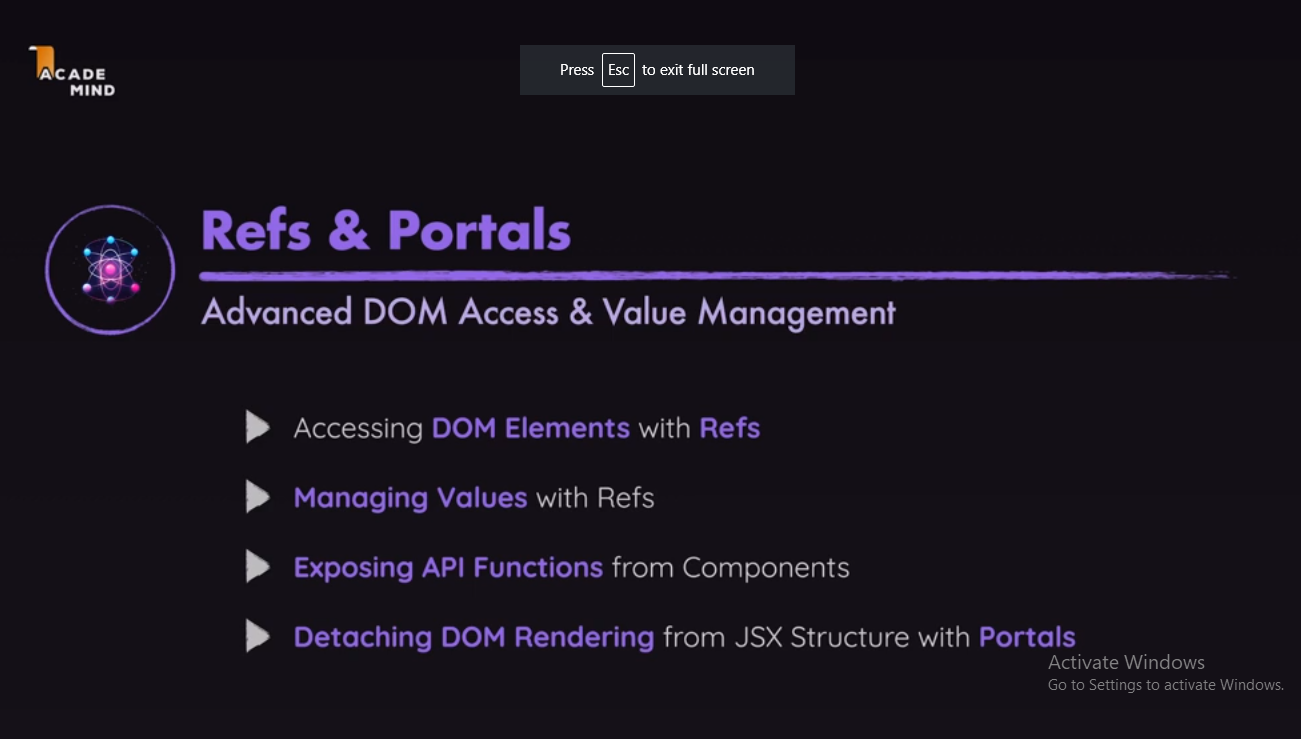
export default App;

**Section 6: Styling React components**



**Working with refs and portals:**

**139.**



What is the use of section in html

We can use section and div same time but if use section search engine looks for section values

49. Reacting to Events.

export default function TabButton({ children }) {

  function handleClick() {

    console.log('Hello World!');

  }

  return (

    <li>

      <button onClick={handleClick}>{children}</button>

    </li>

  );

}