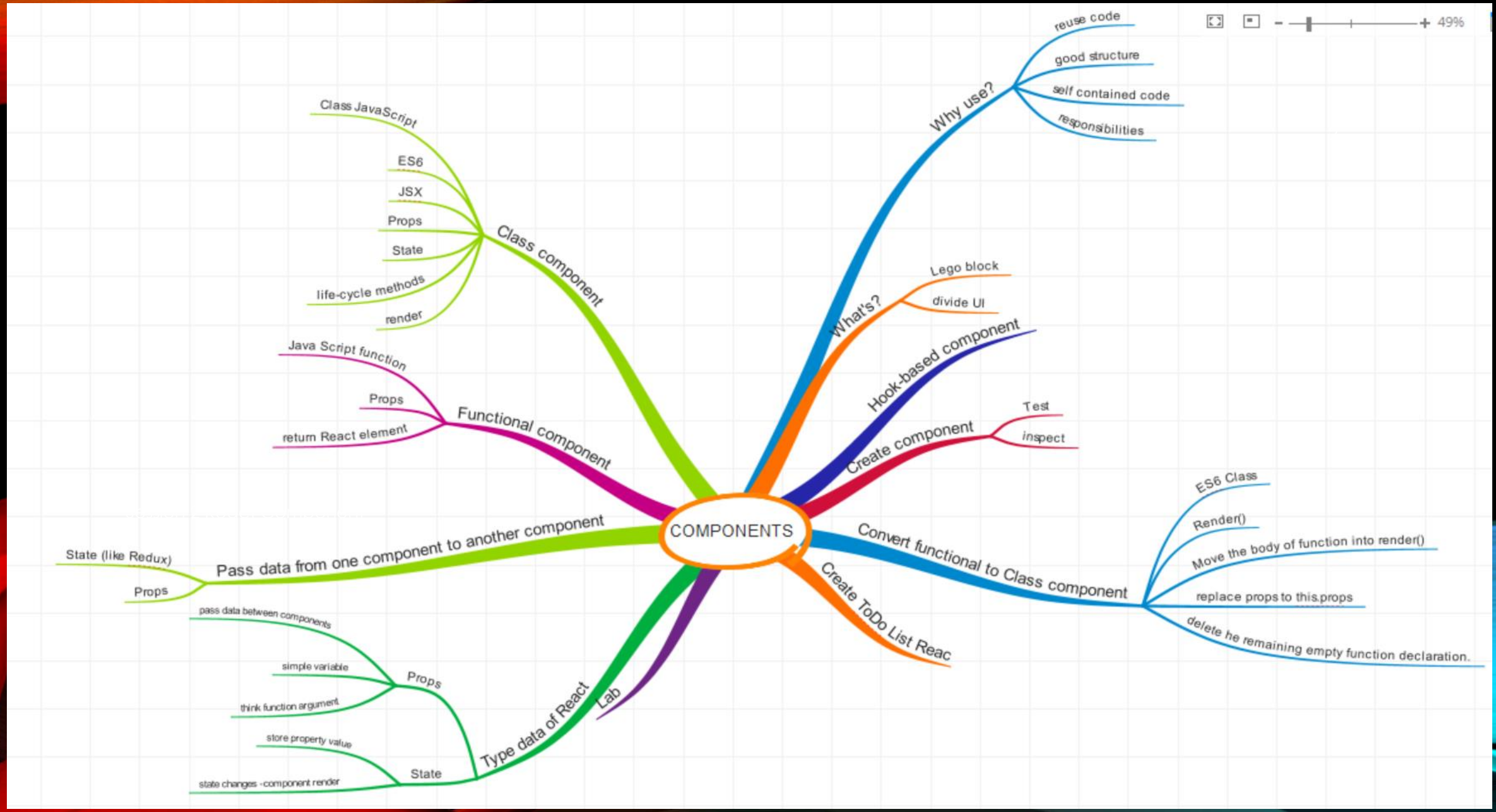
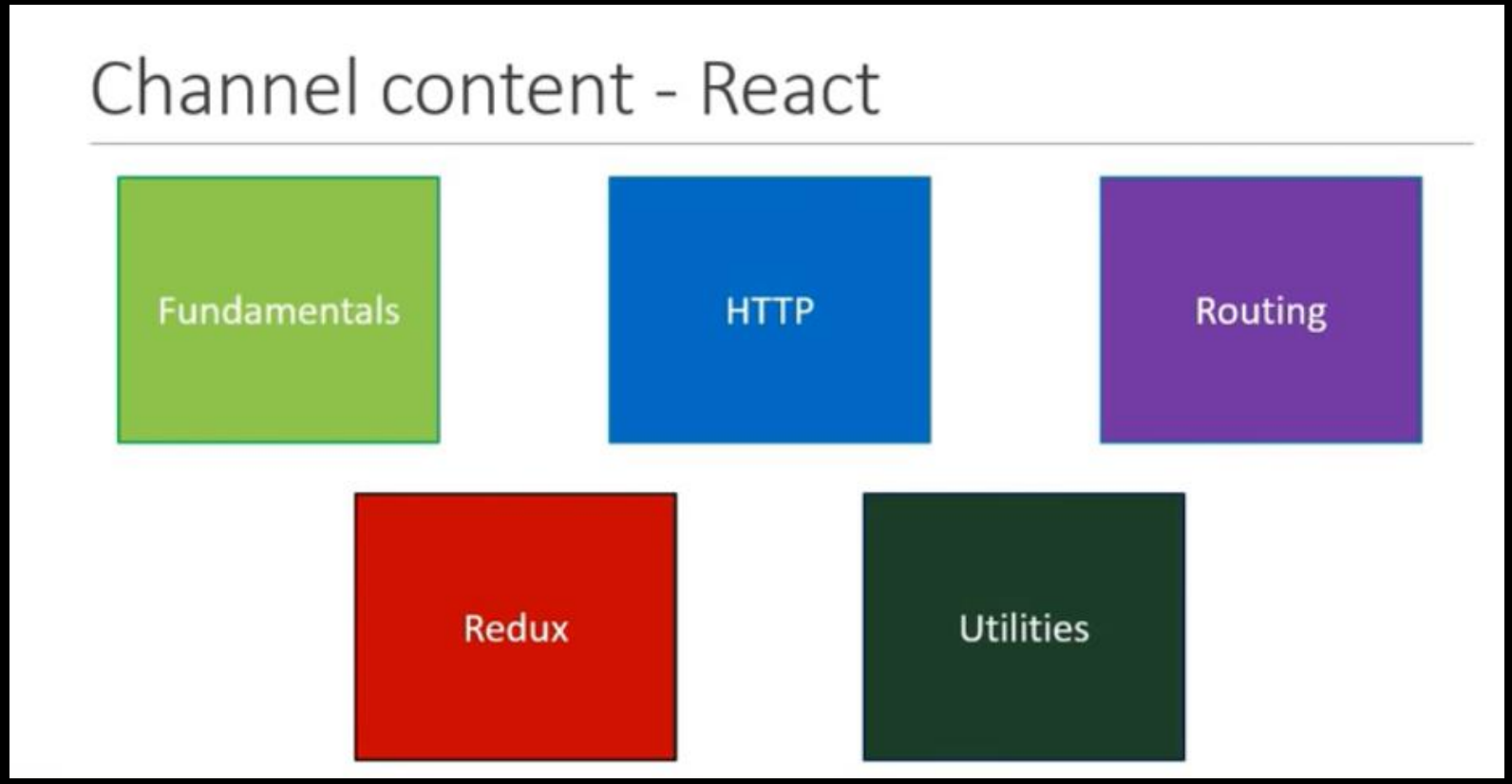
Session 2

React Component

OBJECTIVES

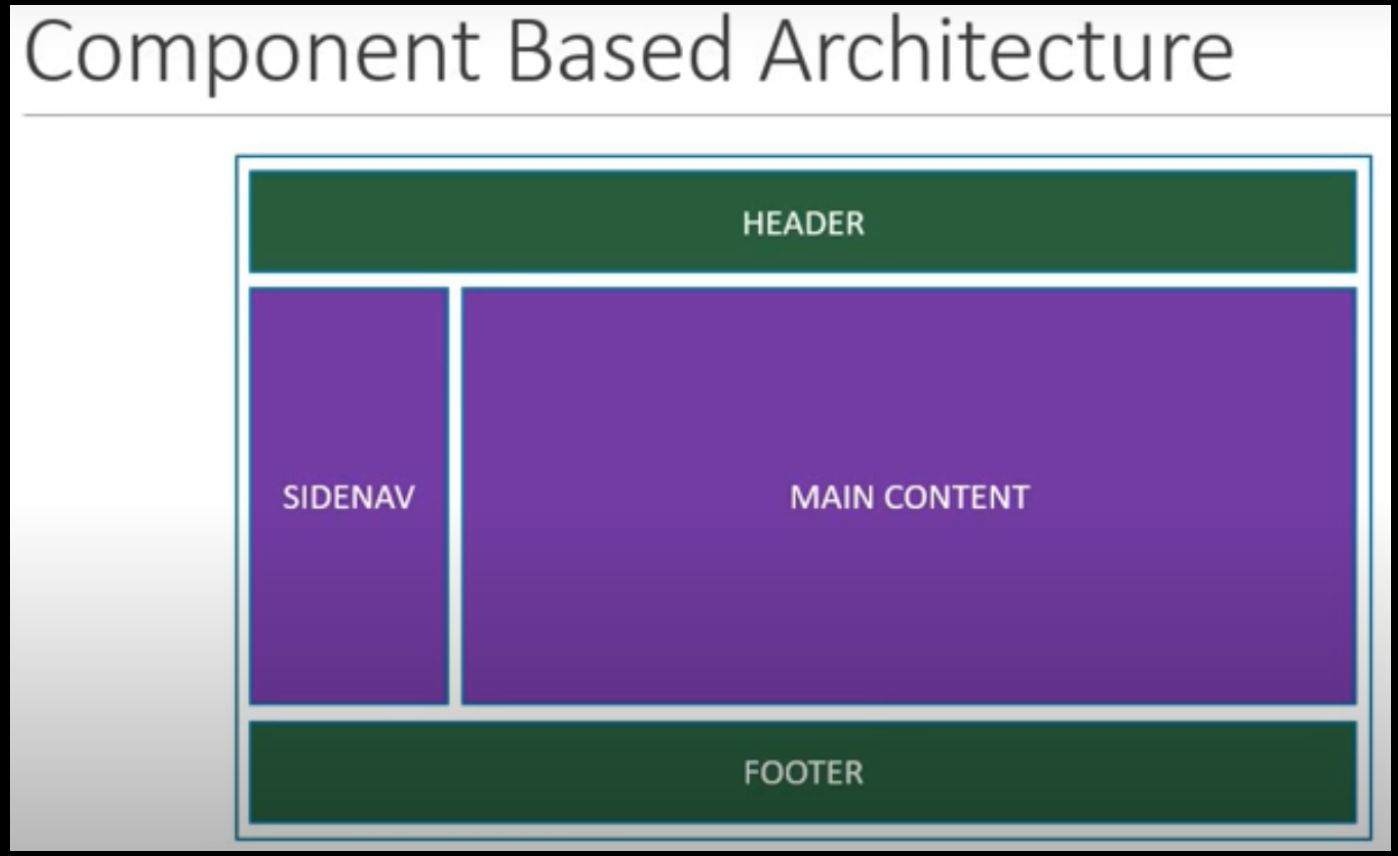

COMPONENTS
Class component
Class JavaScript
ES6
JSX
Props
State
life-cycle methods
render
Functional component
Java Script function
Props
return React element
Pass data from one component to another component
State (like Redux)
Props
Type data of React
Lab
Props
pass data between components
simple variable
think function argument
State
store property value
state changes -component render
Why use?
reuse code
good structure
self contained code
responsibilities
What's?
Lego block
divide UI
Hook-based component
Create component
Test
inspect
Convert functional to Class component
ES6 Class
Render()
Move the body of function into render()
replace props to this.props
delete he remaining empty function declaration.
Create ToDo List Reac
49%

WHAT'S A REACT COMPONENT?



REACT COMPONENT

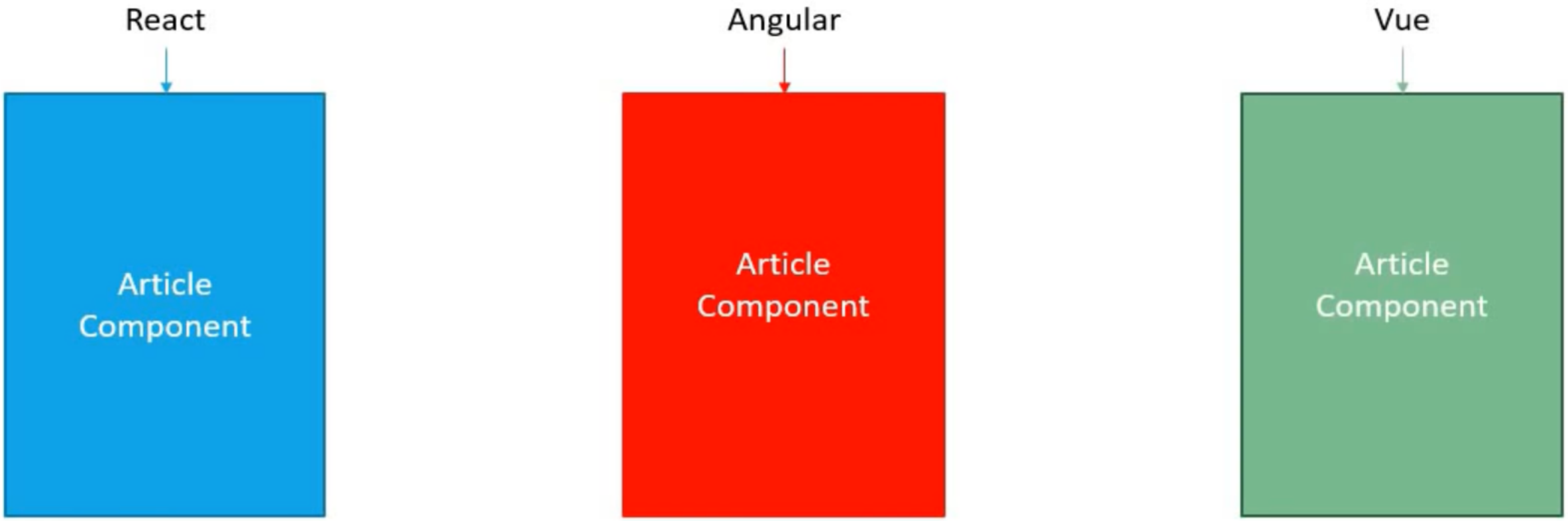
COMPONENTS


Component Based Architecture
HEADER
SIDENAV
MAIN CONTENT
FOOTER

COMPONENTS

Aptech COMPUTER EDUCATION Unleash your potential
Alliance with FPT Education

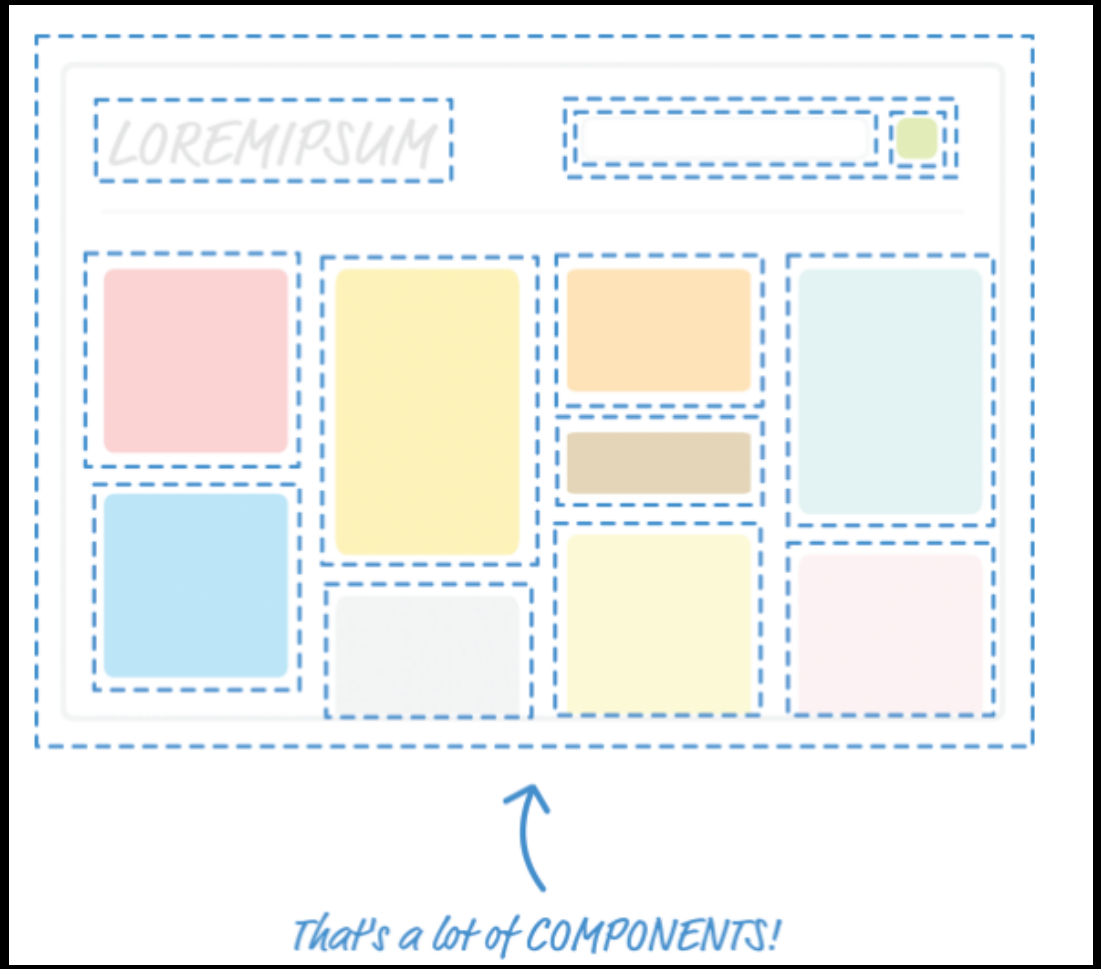
Reusable code



COMPONENTS

Components

COMPONENTS



WHAT IS A REACT COMPONENT?

- A component is an independent, reusable code block, which divides the UI into smaller pieces.
- In other words, we can think of components as LEGO blocks. Likewise we create a LEGO structure from many little LEGO blocks, we create a webpage or UI from many little code blocks (components).

FUNCTIONAL(STATELESS) COMPONENT

- React has 2 types of components: Functional (Stateless) and Class (Stateful).

```
function App() {
  return (
    <div>
      <Clothes name="Quần jean" type="Skinny" color ="Đen" size = "L">Jean</Clothes>
      <Clothes name="Váy" type="váy Dạ hội" color ="Trắng" size = "M">Đầm</Clothes>
    </div>
  );
}
```

functional component

You, a few seconds ago • Uncommitted changes

React Functional Component:

- is a JavaScript / ES6 function
- must return a React element
- take props as parameter if necessary

FUNCTIONAL(STATELESS) COMPONENT

- React has 2 types of components:

  Functional (Stateless) and Class (Stateful).

The simplest way to define a component in React is to write a JavaScript function

```
function Welcome(props) {
      return <h1>Hello,   {props.name}</h1>;
}
```

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello,
      {this.props.name}</h1>;
  }
}
```

REACT CLASS COMPONENT

A React class component

- Is **an ES6 class**, will be a component once it **'extends' React component**.
- can accept **props** (in the constructor) if needed
- can maintain **its own data with state**
- must have a **render( )** method which **returns a React element (JSX),** or null.

CLASS (STATEFUL) COMPONENT

- Class components are ES6 classes.
- They are more complex than functional components including constructors, life-cycle methods, render( ) function and state (data) management.

```
import React, { Component } from 'react';

class ExampleComponent extends Component {
  render() {
    return (
      <div>This is an example component.</div>
    );
  }
}

export default ExampleComponent;
```

FUNCTIONAL(STATELESS) COMPONENT

Functional component is basically a JavaScript (or ES6) function which returns a React element.

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';

function UserInfo(props) {
}

ReactDOM.render(

  document.getElementById('root')
);
```

component as function

WHY COMPONENT?

- We don’t really want to have thousands of lines of code together in one single file. Maintenance of the code gets more and more complex as the project gets bigger.
- Solution : dividing the source code into components helps us a lot.
- Each component has its own JS and CSS code, they are reusable, easier to read, write and test.

HOW TO CREATE AND CALL A COMPONENT

- React functional component as a **JS Function**

```
function Example() {
  return ( <h1>I'm a functional component!</h1> );
};
```

or as an ES6 arrow function

```
const Example = () => {
  return ( <h1>I'm a functional component!</h1> );
};
```

Both of the functions are valid React components.

REACT COMPONENT

COMPONENT BASED

BUILDING BLOCK OF REACT

React helps us break up applications into reusable, self-contained components.

DATA + UI

Components combine view logic, user interface, and data.

COMPOSITION

We can combine multiple components to form complex apps.

REACT COMPONENTS

CLASS COMPONENTS

Component written using a **JavaScript Class**.....

Until recently, many important React features only worked in class component

FUNCTION COMPONENTS

Component written using a **JavaScript Function**....

Until recently, function component were reserved for very simple components.

CLASS COMPONENTS


```
class CounterClassBased extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
    this.decrement = this.decrement.bind(this);
    this.increment = this.increment.bind(this);
  }

  decrement() {
    this.setState({ count: this.state.count - 1 });
  }

  increment() {
    this.setState({ count: this.state.count + 1 });
  }

  render() {
    return (
      <div>
        <div>Current count: {this.state.count}</div>
        <div>
          <button onClick={this.decrement}>-</button>
          <button onClick={this.increment}>+</button>
        </div>
      </div>
    );
  }
}
```

HOOKS-BASED COMPONENT


```
function Counter() {
  const [ count, setCount ] = useState(0);

  return (
    <div>
      <p>count: {count}</p>
      <button onClick={() => setCount(count+1)}>Add</button>
      <button onClick={() => setCount(count-1)}>Subtract</button>
    </div>
  );
}
```

TYPE OF DATA IN REACT

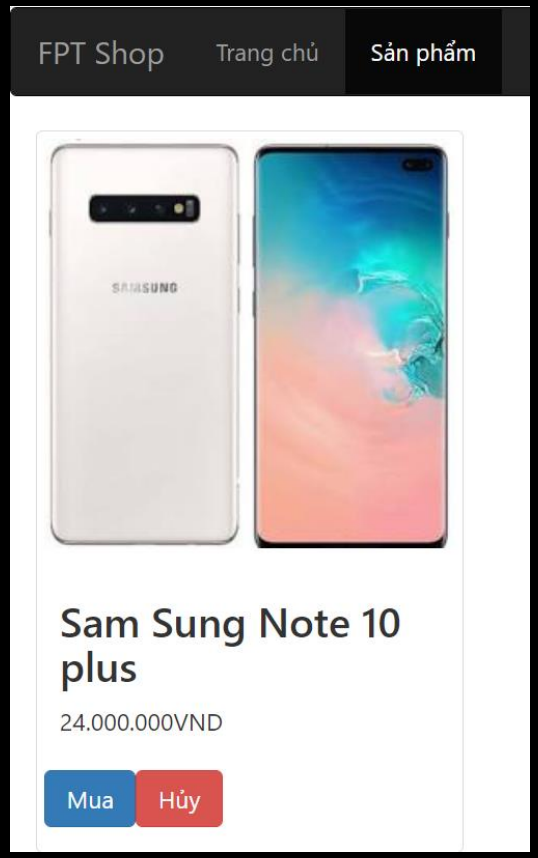
State :

- A variable that can be updated.
- Used to implement “Reactivity”
- Used to trigger a Re-render.
- Responsible for all the interactivity of the app.

DEMO: CREATE REACT COMPONENT

- STEP 1 : I will create 1 **class** component, as **parent** component
- STEP 2: And 1 **functional** component as **child** component
- STEP 3: call the child inside the parent component
- STEP 4: Finally, call the parent component on the **root file (app.js)**

ACTIVITY


FPT Shop
Trang chủ
Sản phẩm
SAMSUNG
Sam Sung Note 10 plus
24.000.000VND
Mua
Hủy

DEMO COMPONENT

Thế Giới Di Động | Trang chủ | Danh sách sản phẩm

Sam Sung Galaxy Note 10 Plus

24000000 VND

Mua Hủy

Sam Sung S10

20000000 VND

Mua Hủy

Iphone 11X Mars

26000000 VND

Mua Hủy

THE END