

# **Bai 3**

# **React Component**

Module: BOOTCAMP WEB-FRONTEND



# Thảo luận

- React Component
- Class component & Function component
- Khái niệm và ý nghĩa của Props, Pure Function



# Mục tiêu

- Trình bày được khái niệm React Component.
- Triển khai được Class Component, Function Component.
- Trình bày được khái niệm Props.
- Trình bày được ý nghĩa của Pure function với React Component.

# React Component

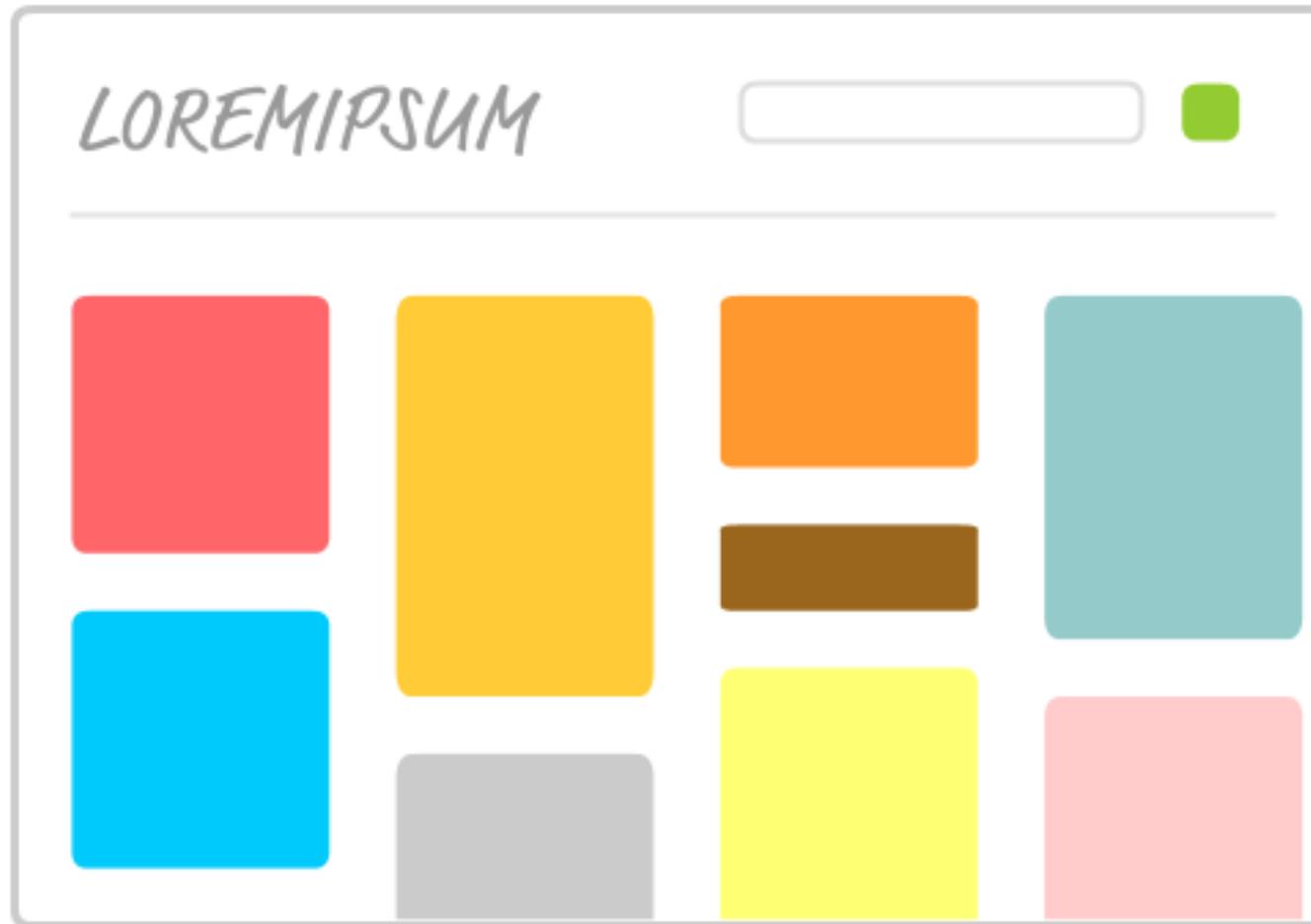


React Component?

Hãy xem ví dụ sau...



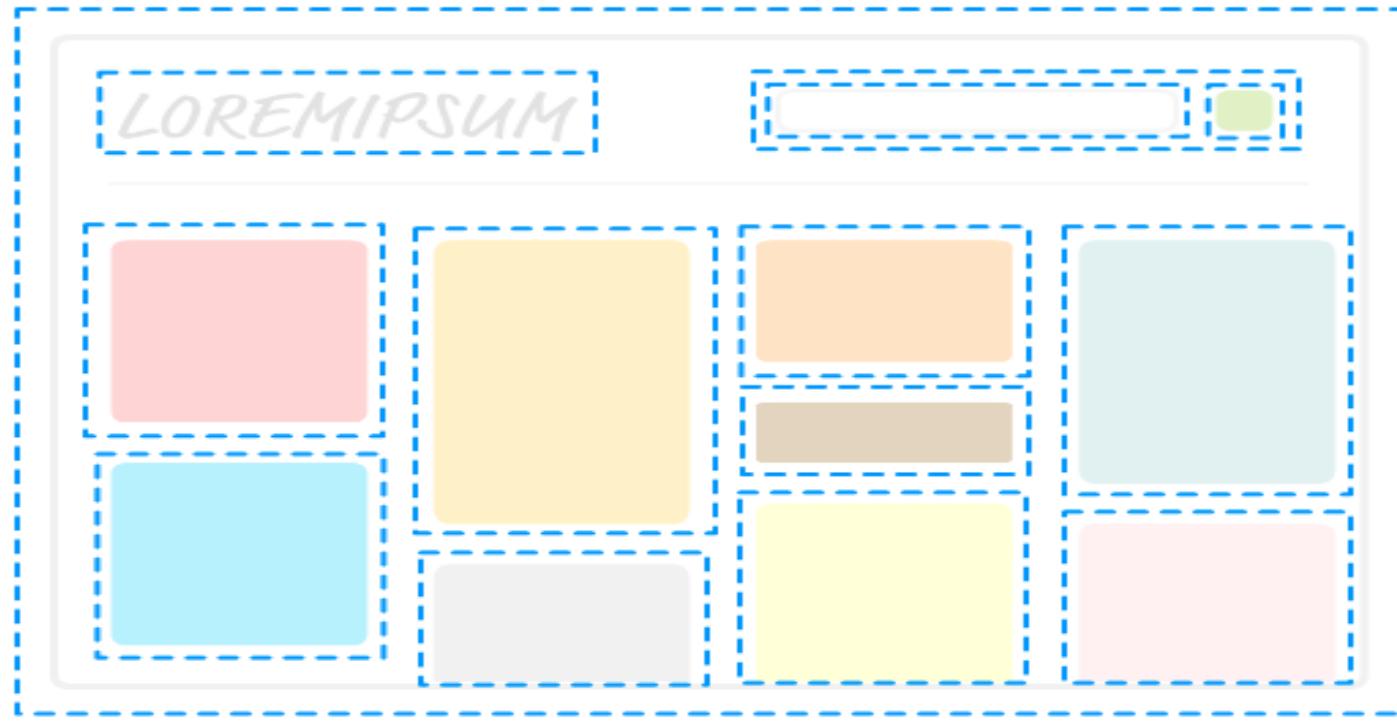
# React Component



Hình 1. Một trang web hoàn chỉnh



# React Component



↑  
*That's a lot of COMPONENTS!*

Hình 1.1 Một trang web hoàn chỉnh cấu thành từ các components

# React Component



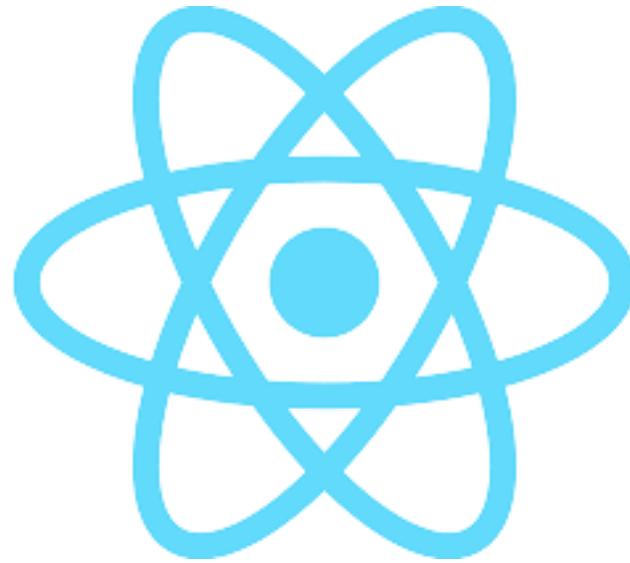
**VẬY, THẾ NÀO LÀ  
REACT COMPONENT?**



# React Component

- Khi ứng dụng phình to có hàng ngàn dòng code, việc phân tách thành component là hữu ích.
- Component là một block code độc lập, có thể tái sử dụng.
- Mỗi component có code JS và CSS riêng, chúng có thể tái sử dụng, dễ đọc, dễ viết, dễ test.
- React có 2 loại component: **Functional Component (Stateless)** và **Class Component(Stateful)**.

# React Component



**FUNCTIONAL COMPONENT**

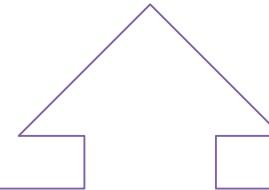


# Functional Component

```
var speed = 10;  
var time = 5;  
alert(speed * time);  
  
var speed1 = 85;  
var time1 = 1.5;  
alert(speed1 * time1);  
  
var speed2 = 12;  
var time2 = 9;  
alert(speed2 * time2);  
  
var speed3 = 42;  
var time3 = 21;  
alert(speed3 * time3);
```



```
function getDistance(speed, time) {  
    var result = speed * time;  
    alert(result);  
}  
  
getDistance(10, 5);  
getDistance(85, 1.5);  
getDistance(12, 9);  
getDistance(42, 21);
```



1. Đẹp + ngắn gọn
2. Tái sử dụng nhiều lần
3. Có thể xây thêm func khác lồng nhau

# React Component



**LIÊN QUAN GÌ ĐẾN  
FUNCTIONAL COMPONENT?**



# Functional Component

- Một function Javascript / ES6 function
- Phải trả về 1 React element
- Nhận props làm tham số nếu cần



# Function Component

Ví dụ sau demo cho function component

```
import React from 'react';

const App =()=>{
  const greeting = 'Hello Function Component!';
  return <Headline value={greeting} />;
}

const Headline = ({ value }) => {
  return <h1>{value}</h1>;
};

export default App;
```

Component App gọi đến component Headline và truyền vào một props là value với giá trị là “Hello Function Component!”.

Giá trị này được gán vào biến greeting. Component Headline trả về (return) giá trị của props truyền vào.



# Function Component

Xây dựng demo functional component

```
// demo without function
ReactDOM.render(
  (
    <div className='hello'>
      <p>Hello, Batman!</p>
      <p>Hello, Iron Man!</p>
      <p>Hello, Nicholas Cage!</p>
      <p>Hello, Mega Man!</p>
      <p>Hello, Bono!</p>
      <p>Hello, Catwoman!</p>
    </div>
  ),
  document.getElementById('root')
);
```



Hello, Batman!  
Hello, Iron Man!  
Hello, Nicolas Cage!  
Hello, Mega Man!  
Hello, Bono!  
Hello, Catwoman!



# Function Component

Xây dựng demo functional component

```
// demo with function
const names = ['Batman', 'Iron Man', 'Nicholas Cage', 'Mega Man', 'Bono', 'Catwoman'];

function Hello() { // note: this function name should named as Pascal case
  return (
    names.map(helloFunction)
  );
}

function helloFunction(name) {
  return (<p>Hello, {name}!</p>)
}

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



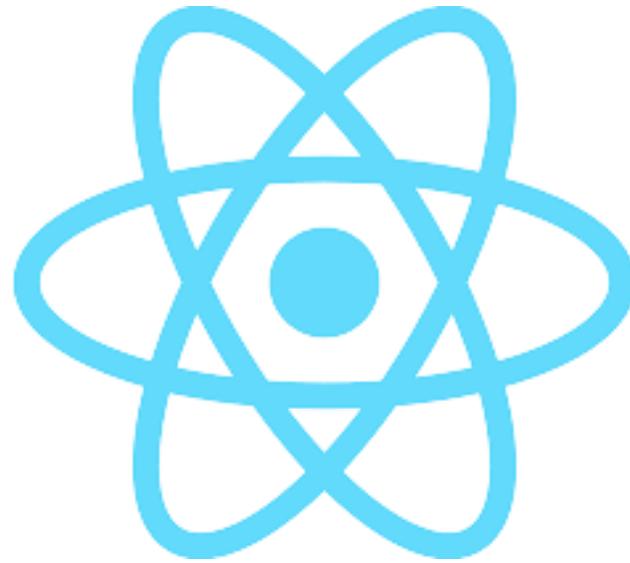
Hello, Batman!  
Hello, Iron Man!  
Hello, Nicolas Cage!  
Hello, Mega Man!  
Hello, Bono!  
Hello, Catwoman!



# Function Component

Thực hành: Xây dựng function Component

# React Component



**CLASS COMPONENT**



# Class Component

- Là một class ES6, nó sẽ là một component khi nó "kết thừa" React component.
- Có thể nhận props (trong hàm khởi tạo) nếu cần.
- Có thể maintain data của nó với state
- Phải có 1 method render() trả về 1 React element (JSX), hoặc null



# Class Component

Ví dụ sau demo cho class component

```
class AppComponent extends React.Component{  
    greeting = “Hello Class Component!”;  
    render(){  
        return <Headline value={greeting} />  
    }  
}  
  
class Headline extends React.Component {  
    render() {  
        return <h1>{this.props.value}</h1>;  
    }  
}
```

Component App gọi đến component Headline và truyền vào một props là value với giá trị là “Hello Class Component!”.

Giá trị này được gán vào biến greeting. Component Headline trả về (return) giá trị của props truyền vào.



# Class Component

Thực hành: Xây dựng Class component



# Props

- Chúng ta sử dụng props để truyền dữ liệu đến các component
- Mọi component được gọi là hàm javascript thuần khiết (Pure function).
- Props tương ứng với tham số của pure function javascript
- Không thể thay đổi được dữ liệu của Props.

# Props



Thực hành: Sử dụng Props trong component



# Pure function

- Hàm luôn trả về cùng một kết quả nếu được truyền vào các tham số không đổi. Nó không hề phụ thuộc vào bất kỳ trạng thái hoặc dữ liệu nào.
- Hàm này không tạo ra bất kỳ ảnh hưởng nào đến các đối tượng khác , chẳng hạn như các request, input, output hoặc data mutation.

# Tóm tắt bài học



- Components giúp phân chia các UI (giao diện người dùng) thành các phần nhỏ để dễ dàng quản lý và tái sử dụng.
- Function component là một hàm Javascript (hoặc ES6) trả về 1 phần tử/1 element React
- Các Class components là những class ES6. Chúng phức tạp hơn functional components ở chỗ nó còn có: phương thức khởi tạo, life-cycle, hàm render() và quản lý state (data)
- Props là 1 đối tượng, nó lưu trữ các giá trị của các attribute (thuộc tính) của một thẻ (Tag).
- Pure function là Hàm luôn trả về cùng một kết quả nếu được truyền vào các tham số không đổi, Hàm này không tạo ra bất kỳ ảnh hưởng nào đến các đối tượng khác.