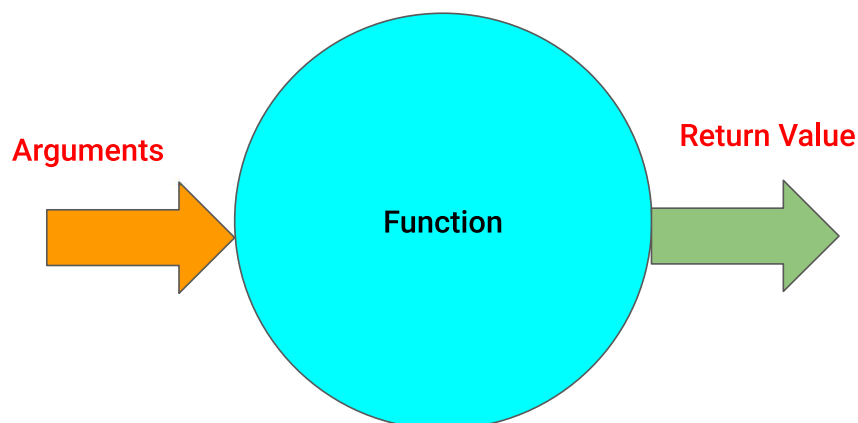


UI Component and React

Dr Harshad Prajapati
30 Nov 2023

1

Function: Let's Understand What a Function Does



Output will change as input changes

Example:

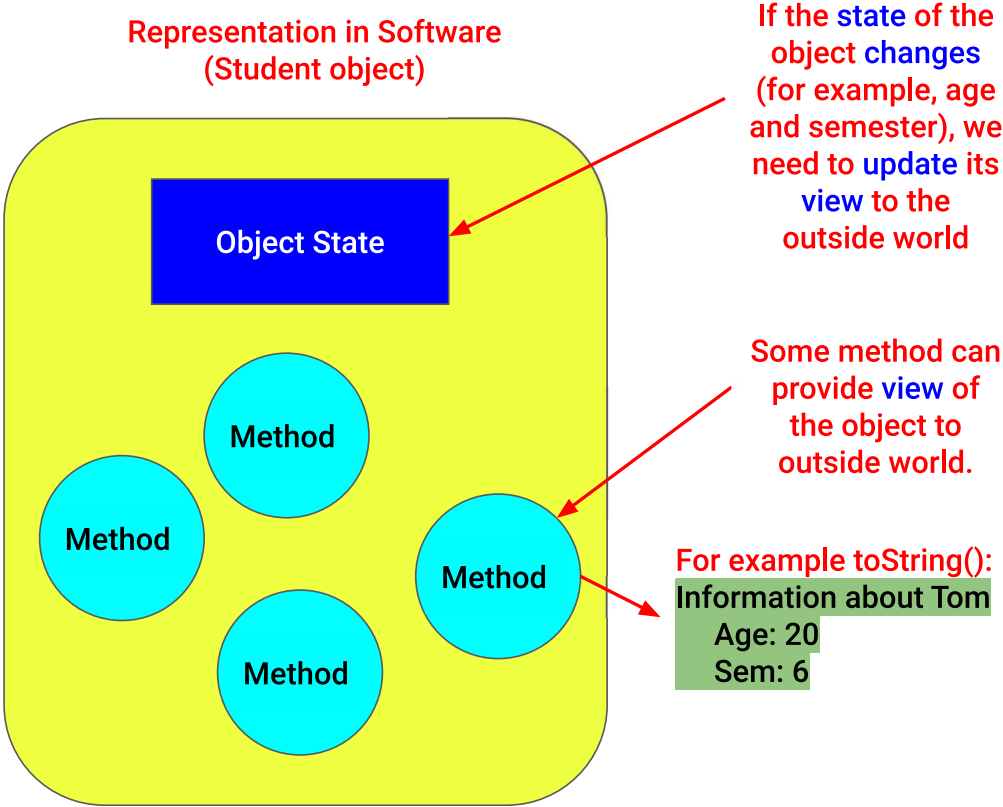
```
sortedStudentList = sort(studentList)
```

2

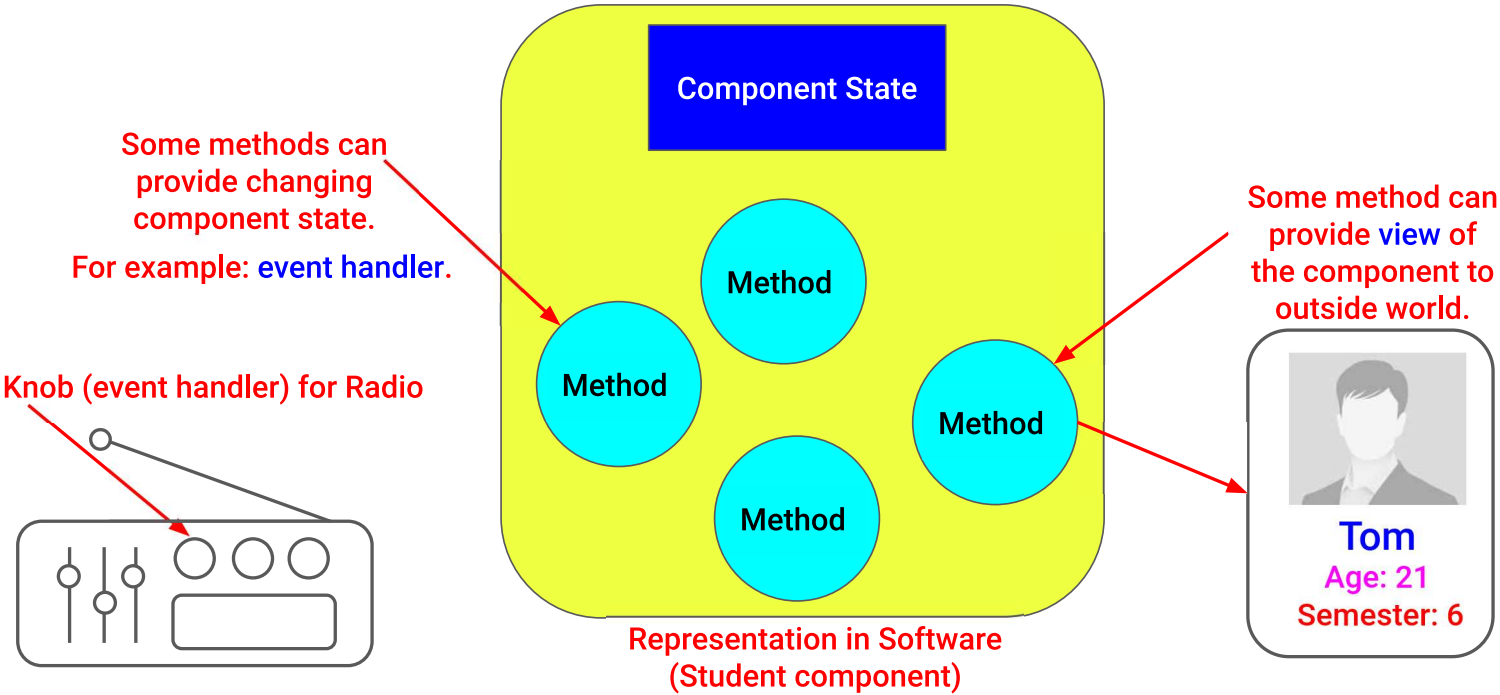
Object



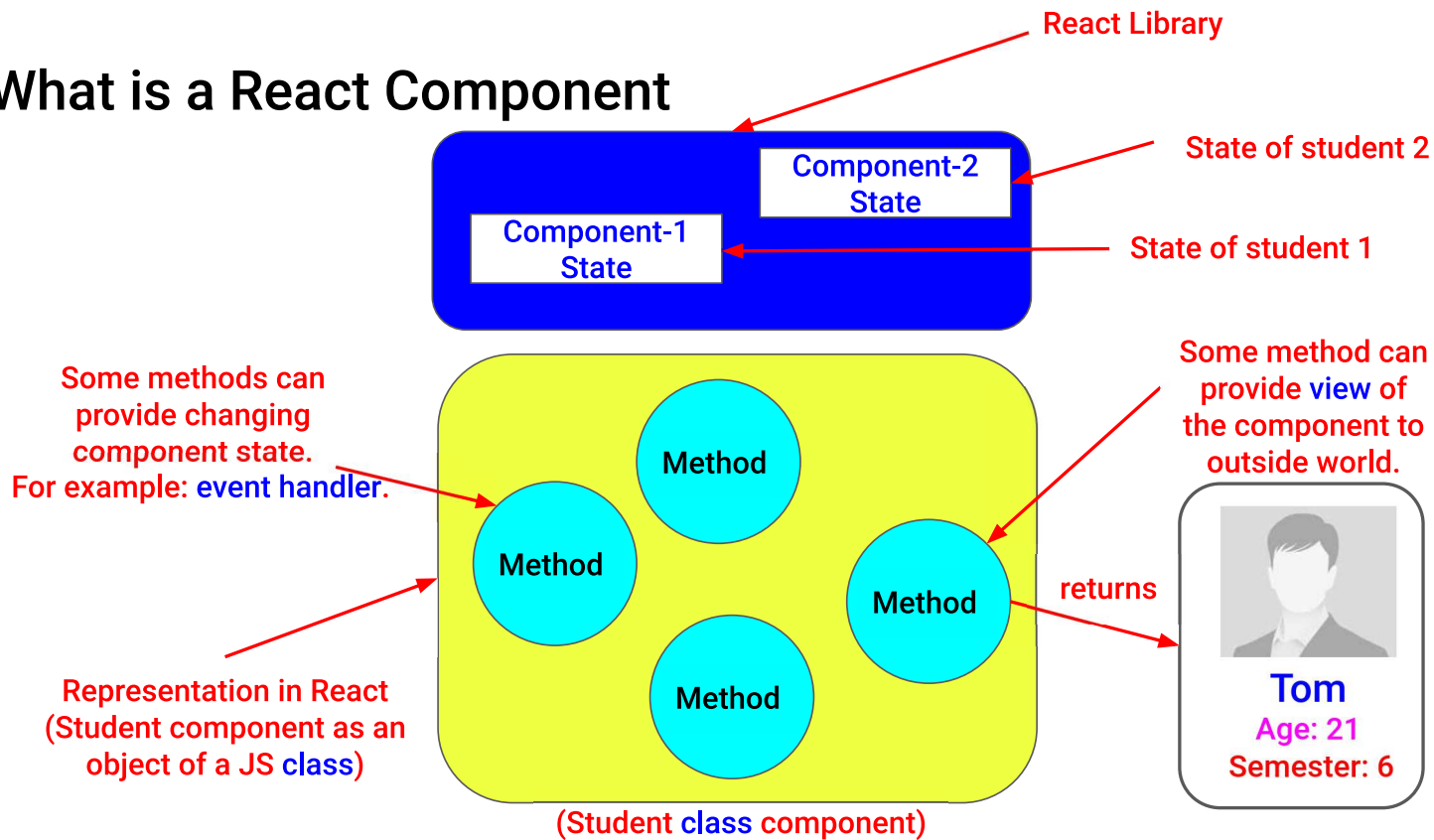
Real world object
(Student)



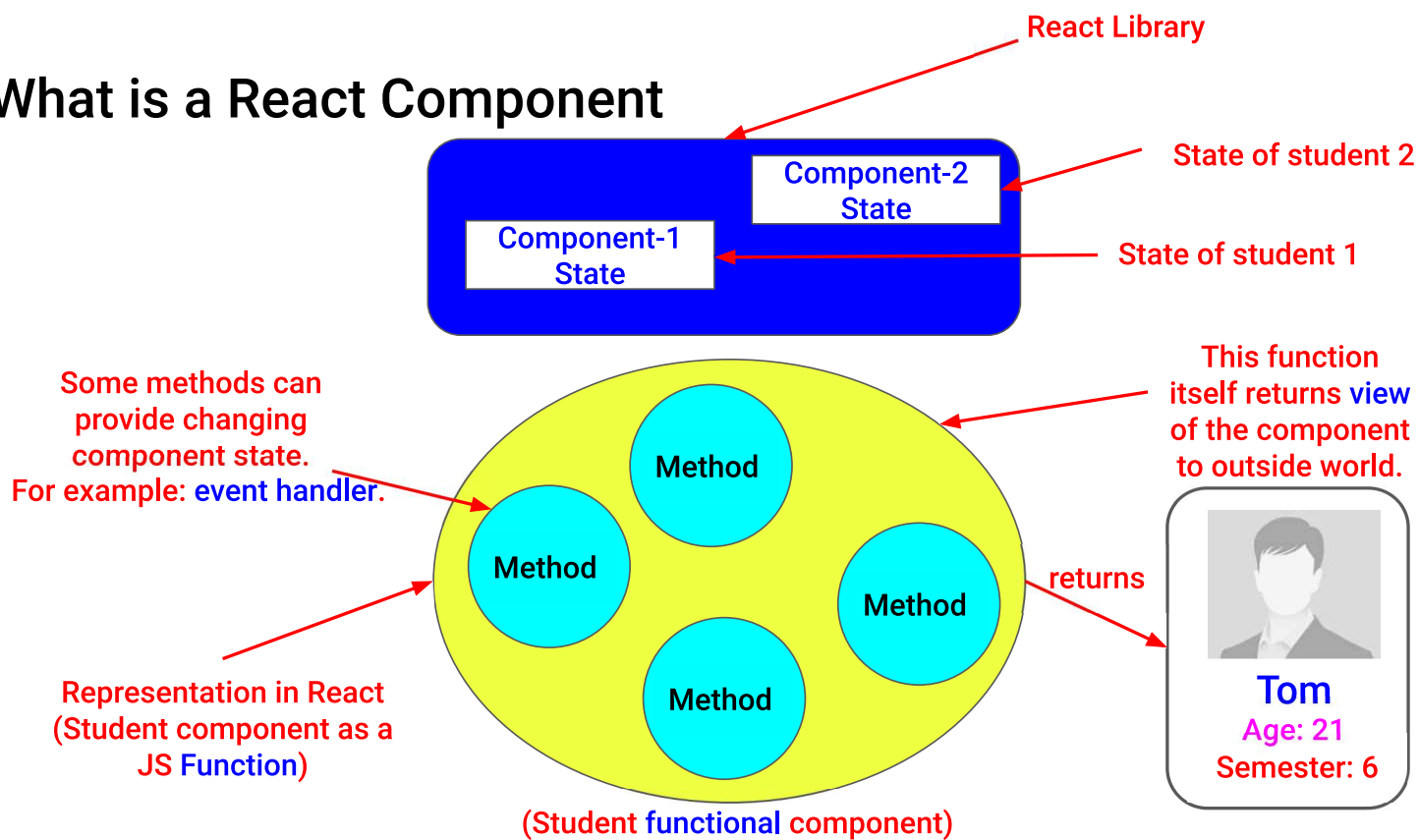
Component in General



What is a React Component



What is a React Component



Instantiate Object versus Component

If there is a class named **Student**, we instantiate it using the following:

```
student = new Student("Tom", 21, 6);
```

When we want to display student object, we use its `toString()` method.

If there is a component named **Student**, we instantiate it using the following:

```
<Student name="Tom" age=21 sem=6 />
```

When we want to display student component, we use `render()` method of class component or return value of functional component.

7

View of **Object** versus UI **Component** for Outside World

Return value of method (`toString()`):

```
"Information about" + "Tom" + "\n" +  
"\tAge:" + 20 + "\tSem:" + 6
```

This is what
method
returns

```
Information about Tom  
Age: 20  
Sem: 6
```

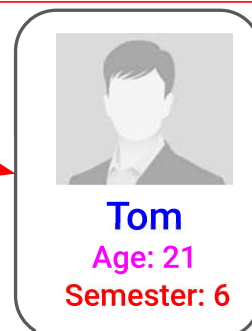
This is what
is printed

Return value of component:

```
<div ...>  
    
  <div ...>Tom</div>  
  <div ...>Age: 21</div>  
  <div ...>Semester: 6</div>  
</div>
```

This is what
component
returns

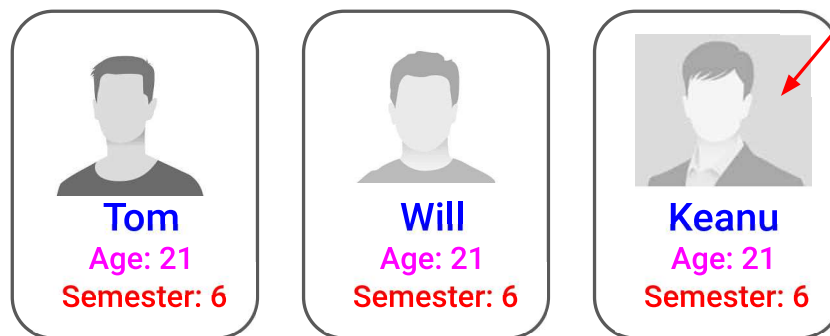
This is what
component
looks like



8

What else a React Component can do?

- A component is a self contained **reusable** UI part.



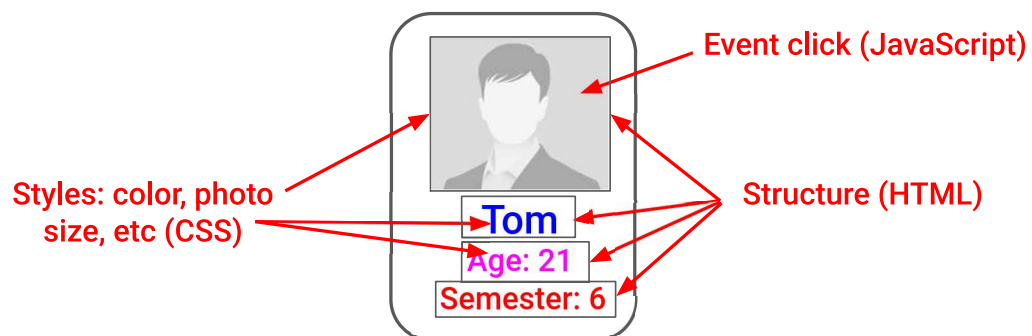
Component can allow a user to **interact** with itself.
For example, change profile picture by interacting with it using keyboard /touch/mouse

We define a Student component once and then can create multiple instances of the component for different students.

9

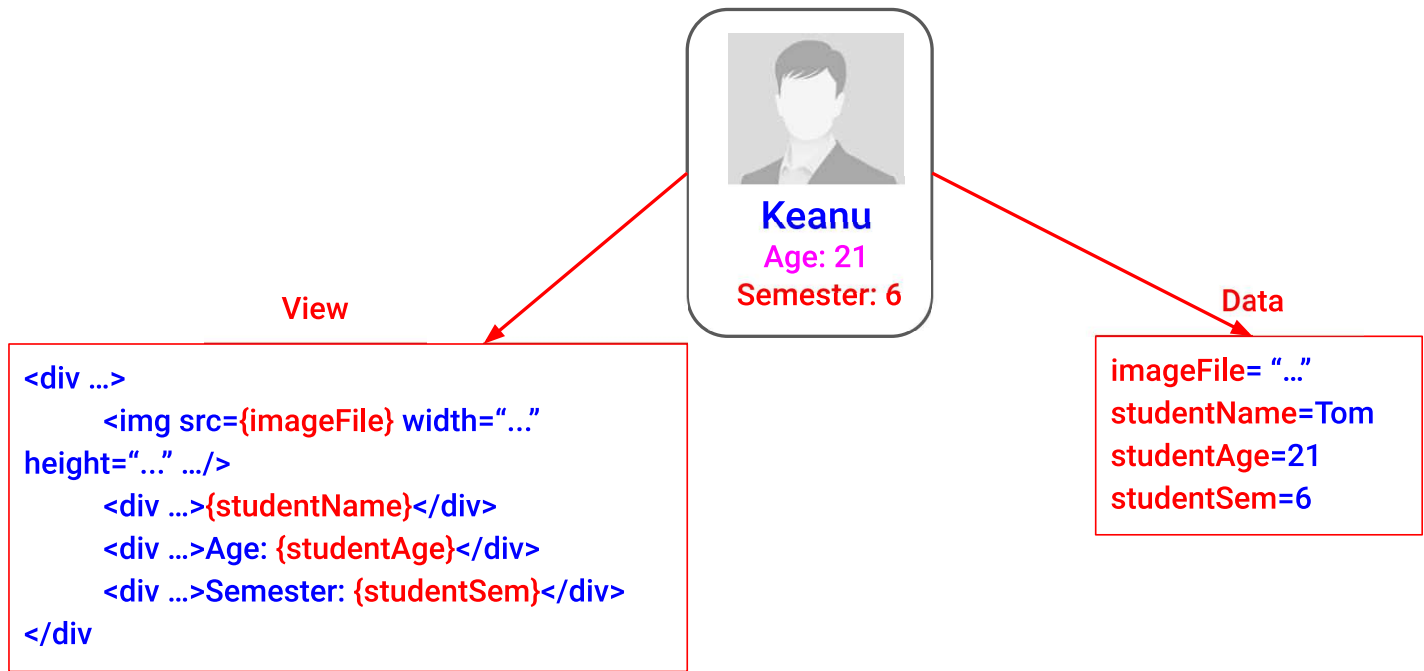
React Component

- A React component is composed of:
 - A template (HTML)
 - Interactivity (JavaScript)
 - Styles (CSS)



10

Separation of Data and View



11

Data Related to Component

- A React component can have **dynamic information** in **two forms**:
 - **props** (Properties): Initialization values.
 - **state**: State is within the component.
- **Props**:
 - **Props** are **arguments** passed into **React component**, similar to arguments passed to a function.
 - Props are **immutable**.
- **State**:
 - State is **mutable**.
 - The **state object** can contain **multiple properties**.

12

Props (Initialization values)

- **Information** that is **passed** from **one component to another** is called as props in React.
- A component's **props** is an **object** and it holds **initialization information** about the component.
- These **props** are **transferred** to the **component** through the **constructor** of the **class component** and through **function arguments** in **functional component**.
- We can access props object inside a component using **this.props.<propName>** in **class component** or **props.<propName>** in a **functional component**.

13

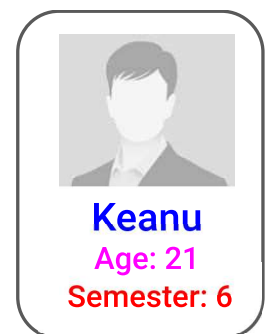
What is a React Component with Props

- We can pass data as props to the component.
- Suppose, the student component is defined (class or function) and is named as **Student**.
- And Student component is used in **StudentList** component.

The parent component (StudentList) can pass values to three props (name, age, and sem)

```
<div ... >  
  <Student name="Keanu" age=21 sem=6 />  
  <Student name="Tom" age=21 sem=6 />  
</div>
```

Student component

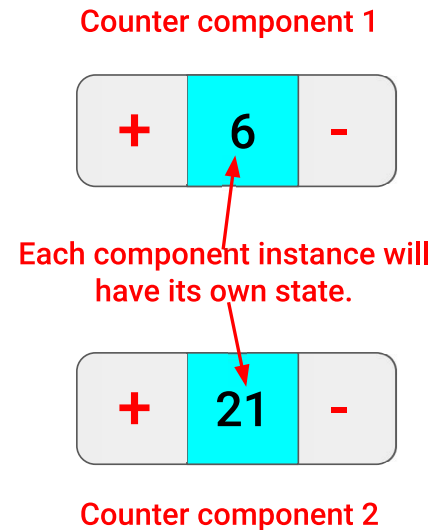


Each component instance will have its own props.

14

What is a React Component with **State**

- The state is **not passed from the outside**. A component decides **its own state**.
- To create state, we need to assign an object (called **state object**) to **this.state** property in **class component** OR using hook (**useState**) in a **functional component**.
- For example,
 - When we **click + or - button**, the **counter** gets **incremented or decremented**.
 - This **count value** is the **state** of the component.



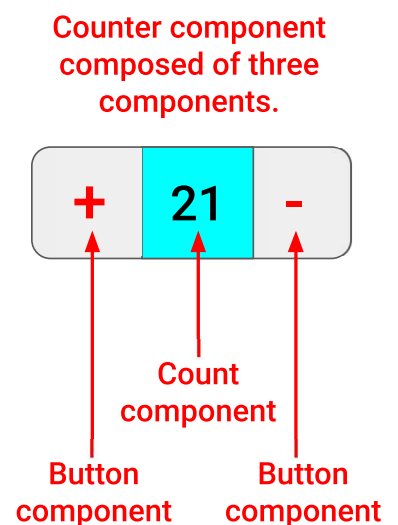
15

React Component with **Composition**

- We can **compose existing components** to create a **new component**.
- For example, We can implement Counter using **existing component Button and Count**.

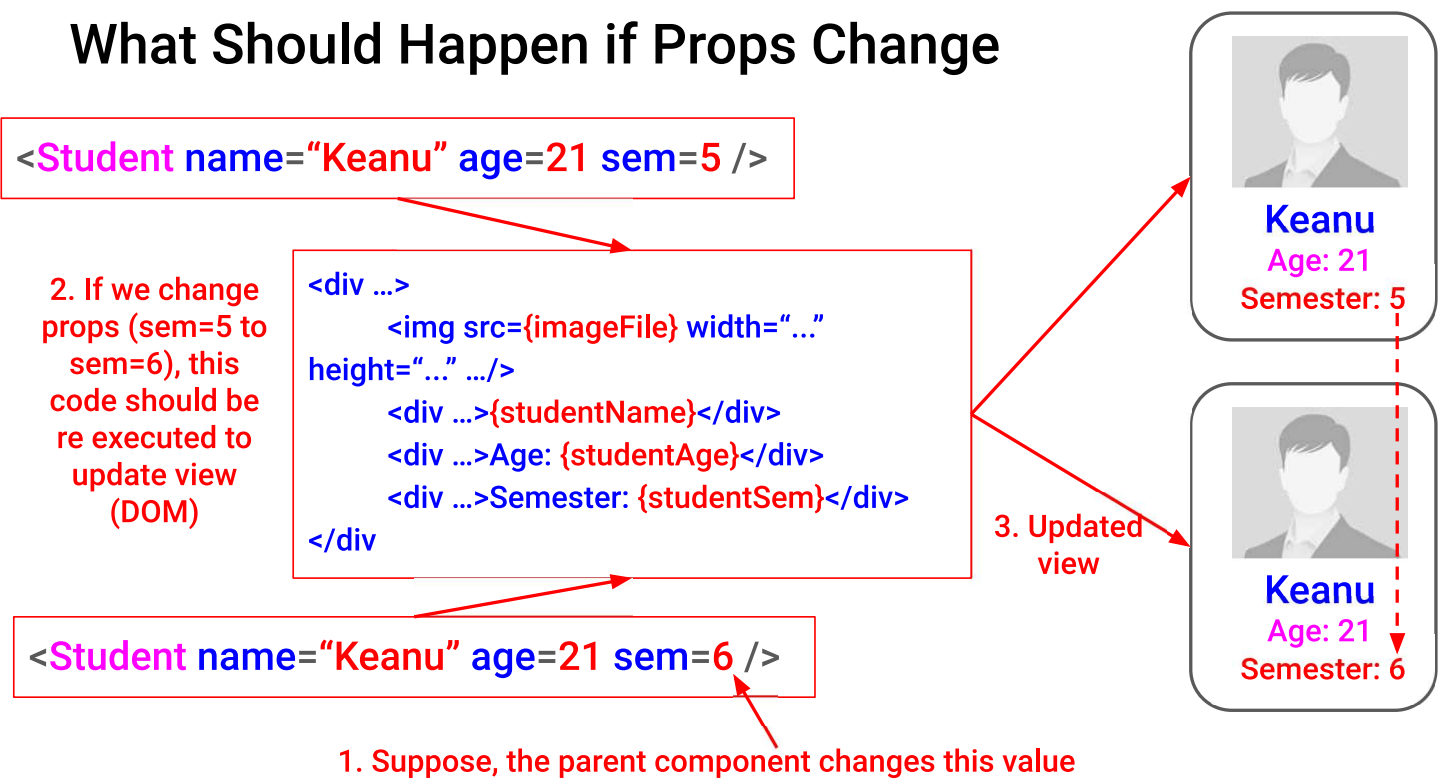
```
<div ...>  
  <Button ... />  
  <Count .../>  
  <Button ... />  
</div>
```

Code fragment of
Counter component

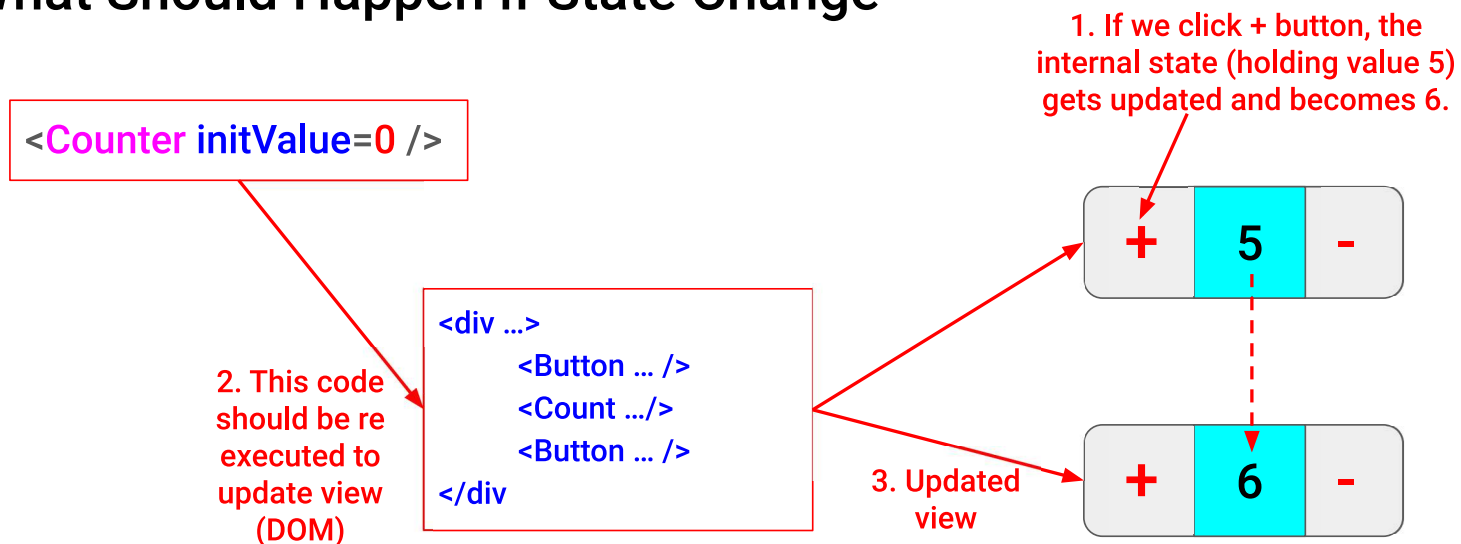


16

What Should Happen if Props Change



What Should Happen if State Change



What React is Responsible for

- React is responsible for:
 - **Rendering** and **re-rendering** component (creating **virtual DOM**).
 - **Updating** virtual DOM into **actual browser DOM** efficiently.