Date : 01 - 07 - 2020
Morning Session : 9 am – 11.00 PM
By ~ Sundeep Charan Ramkumar Today

Topics: **React Day 3 (state and events)**

A guide to “ this “ in JavaScript

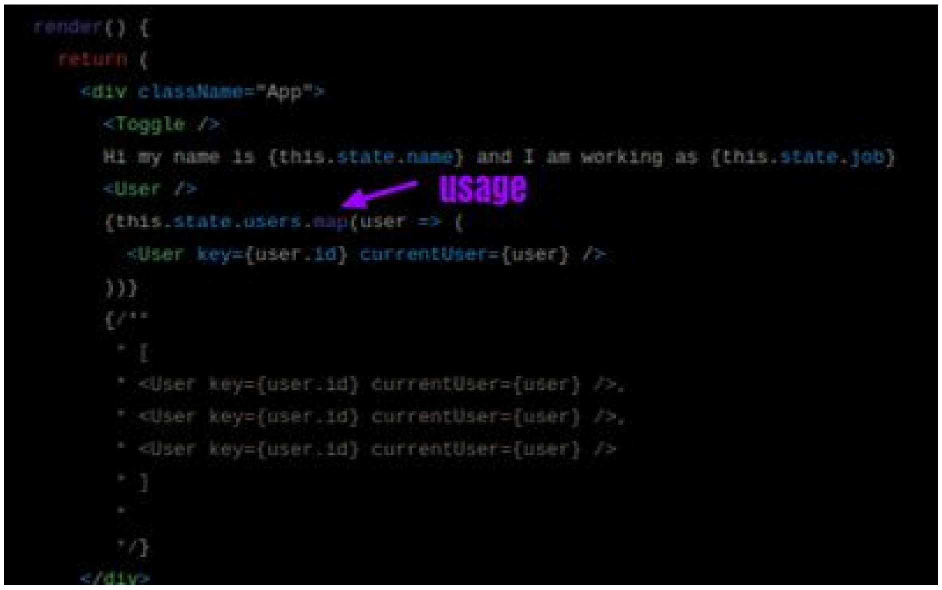
https://www.freecodecamp.org/news/a-guide-to-this-in-javascript-e3b9daef4df1/

Call,Apply, and Bind :

- **Use .bind() when you want that function to later be called with a certain context, useful in events.**
- **Use .call() or .apply() when you want to invoke the function immediately, and modify the context.**
- **Call/apply call the function immediately, whereas bind returns a function that, when later executed, will have the correct context set for calling the original function.**
- **This way you can maintain context in async callbacks and events.**

https://www.freecodecamp.org/news/how-to-use-the-apply-call-and-bind-methods-in-javascript-80a8e6096a90/

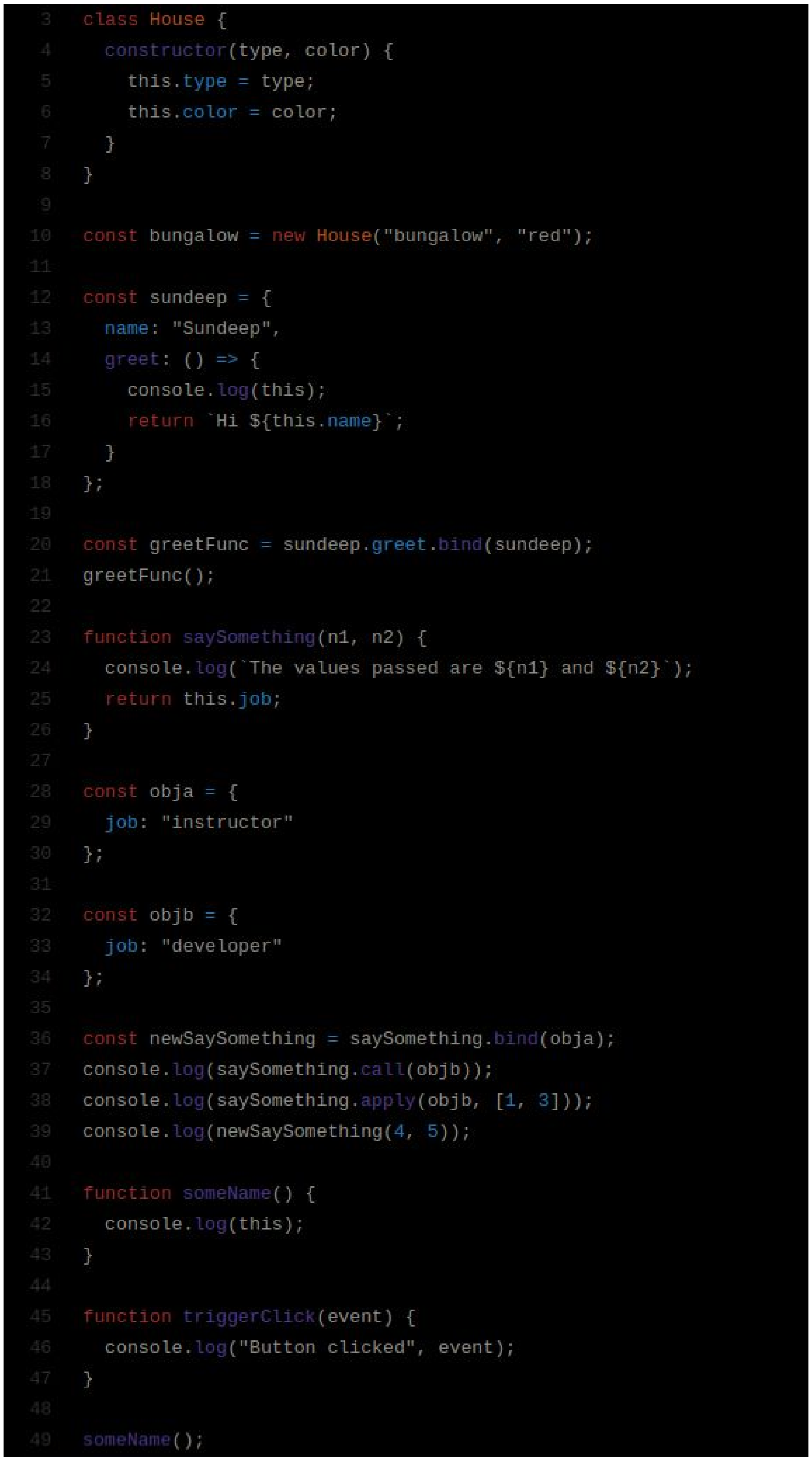
Map Function :



```
render() {
  return (
    <div className="App">
      <Toggle />
      Hi my name is {this.state.name} and I am working as {this.state.job}
      <User />
      {this.state.users.map(user => (
        <User key={user.id} currentUser={user} />
      ))}
      {/**
       * [
       * <User key={user.id} currentUser={user} />,
       * <User key={user.id} currentUser={user} />,
       * <User key={user.id} currentUser={user} />
       * ]
       *
       */}
    </div>
```

- **Keys help React identify which items have changed, are added, or are removed. Keys should be given to the elements inside the array to give the elements a stable identity**

This, bind, call and apply( ) code :



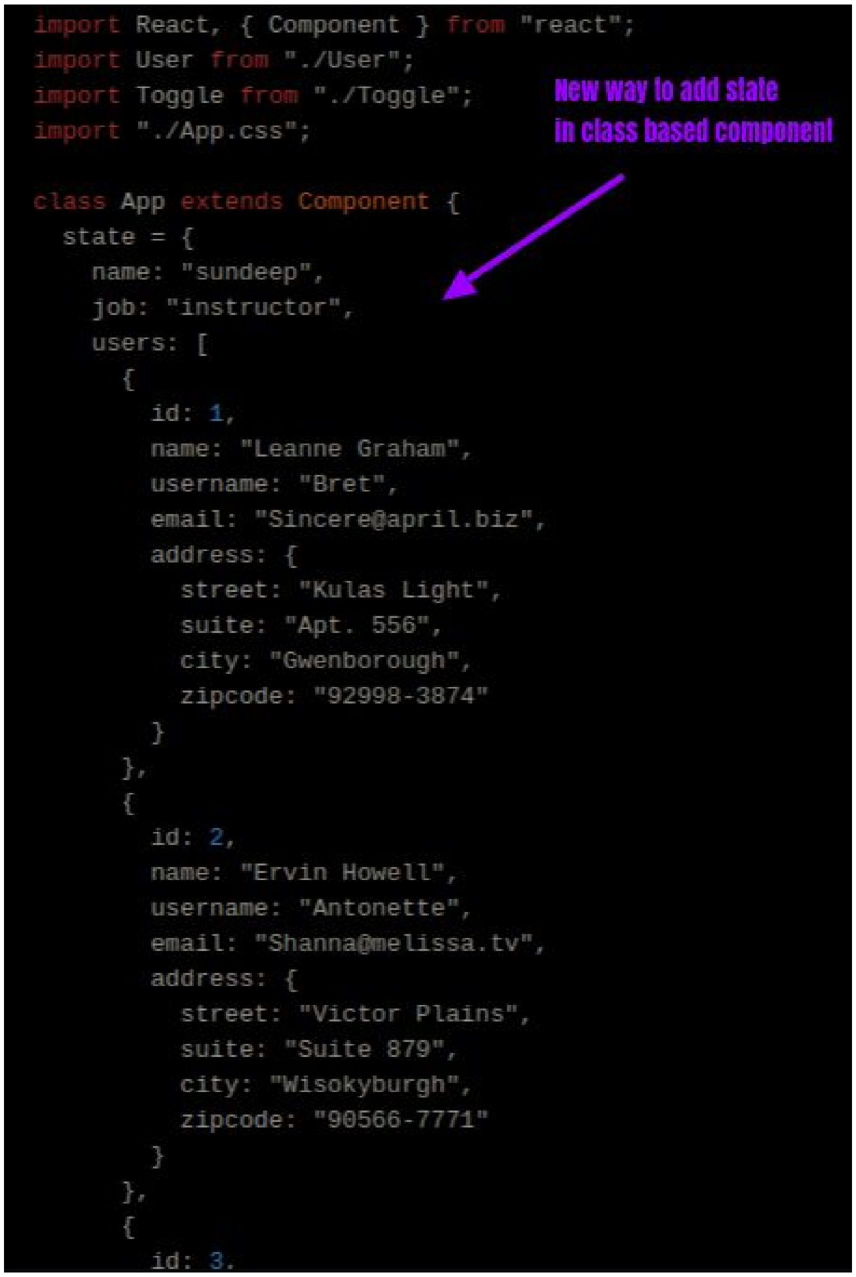
```javascript
class House {
  constructor(type, color) {
    this.type = type;
    this.color = color;
  }
}

const bungalow = new House("bungalow", "red");

const sundeep = {
  name: "Sundeep",
  greet: () => {
    console.log(this);
    return `Hi ${this.name}`;
  }
};

const greetFunc = sundeep.greet.bind(sundeep);
greetFunc();

function saySomething(n1, n2) {
  console.log(`The values passed are ${n1} and ${n2}`);
  return this.job;
}

const obja = {
  job: "instructor"
};

const objb = {
  job: "developer"
};

const newSaySomething = saySomething.bind(obja);
console.log(saySomething.call(objb));
console.log(saySomething.apply(objb, [1, 3]));
console.log(newSaySomething(4, 5));

function someName() {
  console.log(this);
}

function triggerClick(event) {
  console.log("Button clicked", event);
}

someName();
```

State in React:

- **It help is holding the “current data” of the Component, like at a particular instant.**

- **It is immutable (can't change it directly) → It is changed with setState, we will learn in events**
- **The state value should always be an object**
- **The state can be communicated via props.**
-

https://medium.com/the-andela-way/understanding-the-fundamentals-of-state-in-react-79c711be677f

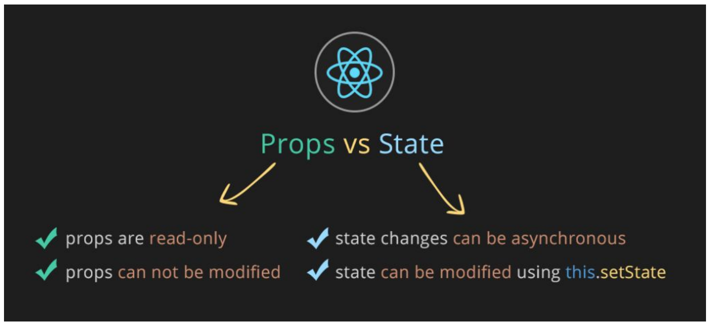


```
import React, { Component } from "react";
import User from "./User";
import Toggle from "./Toggle";
import "./App.css";

class App extends Component {
  state = {
    name: "sundeep",
    job: "instructor",
    users: [
      {
        id: 1,
        name: "Leanne Graham",
        username: "Bret",
        email: "Sincere@april.biz",
        address: {
          street: "Kulas Light",
          suite: "Apt. 556",
          city: "Gwenborough",
          zipcode: "92998-3874"
        }
      },
      {
        id: 2,
        name: "Ervin Howell",
        username: "Antonette",
        email: "Shanna@melissa.tv",
        address: {
          street: "Victor Plains",
          suite: "Suite 879",
          city: "Wisokyburgh",
          zipcode: "90566-7771"
        }
      },
      {
        id: 3,
```

https://reactjs.org/docs/state-and-lifecycle.html

https://reactjs.org/docs/lists-and-keys.html

State V/S Props:


Props vs State

✓ props are read-only
✓ props can not be modified
✓ state changes can be asynchronous
✓ state can be modified using this.setState

https://stackoverflow.com/questions/27991366/what-is-the-difference-between-state-and-props-in-react

Prop Drilling:

- Prop drilling is a technique of passing data at multiple levels
- prop drilling is used to pass data to lower components at multiple levels in the hierarchy

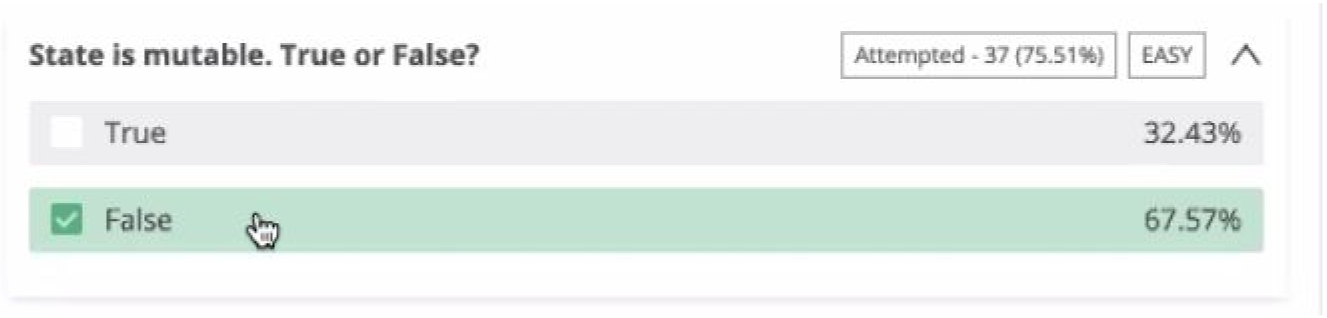
Events in React:

- **Just like HTML, React can perform actions based on user events.**
- **React has the same events as HTML: click, change, mouseover etc.**
- **React events are written in camelCase syntax:**
- **onClick instead of click.**
- **React event handlers are written inside curly braces:**
- **onClick={shoot} instead of onClick="shoot()".**
- **A good practice is to put the event handler as a method in the component class: → Event Handling**

What is a Synthetic Event?

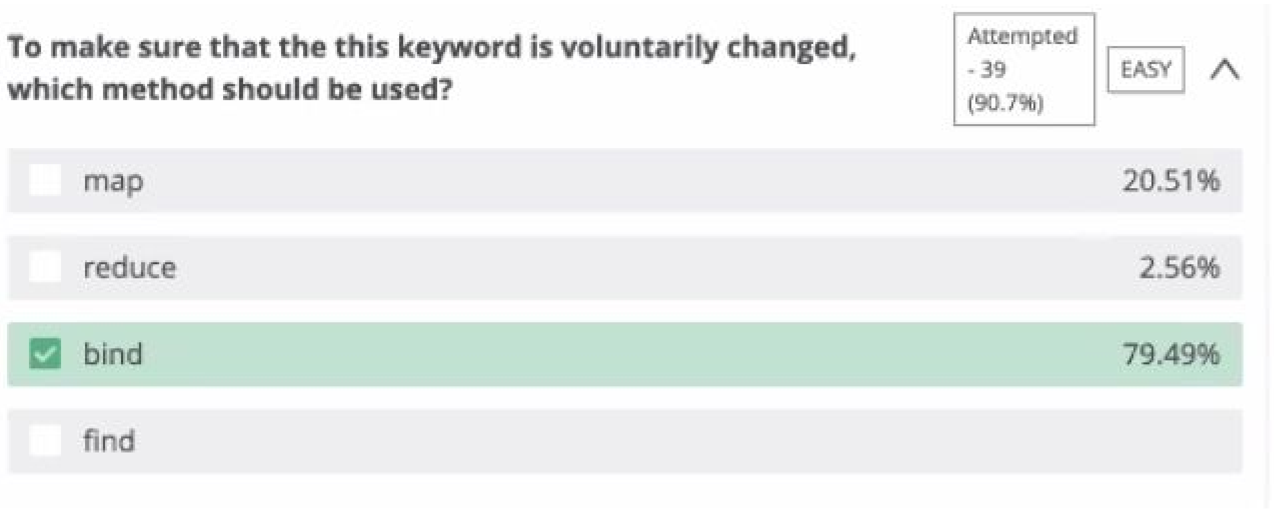
- **SyntheticEvent object will be reused and all properties will be nullified after the event callback has been invoked.**
- **This is for performance reasons.**

https://www.w3schools.com/react/react_events.asp



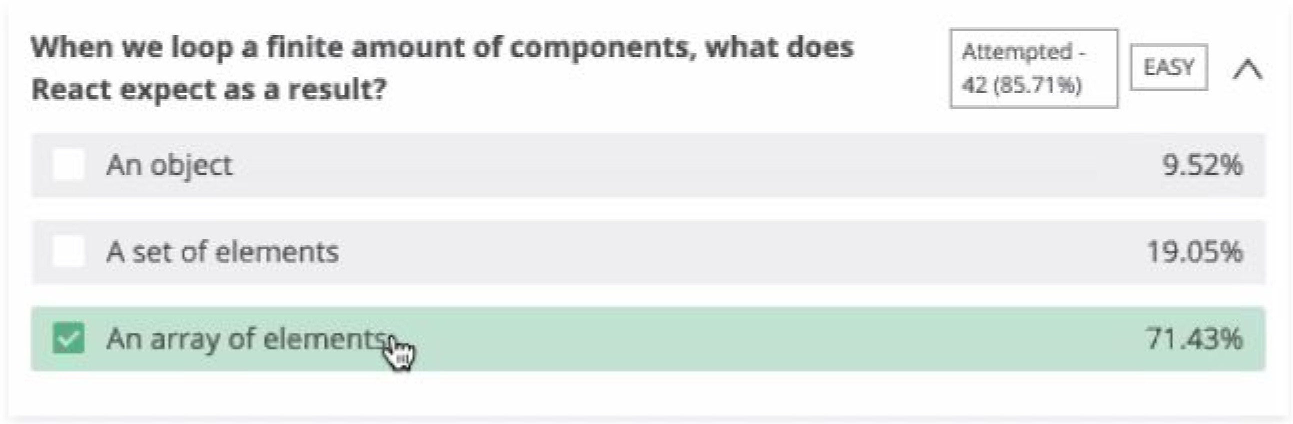
State is mutable. True or False? Attempted - 37 (75.51%) EASY

True	32.43%
[✓ False]	67.57%



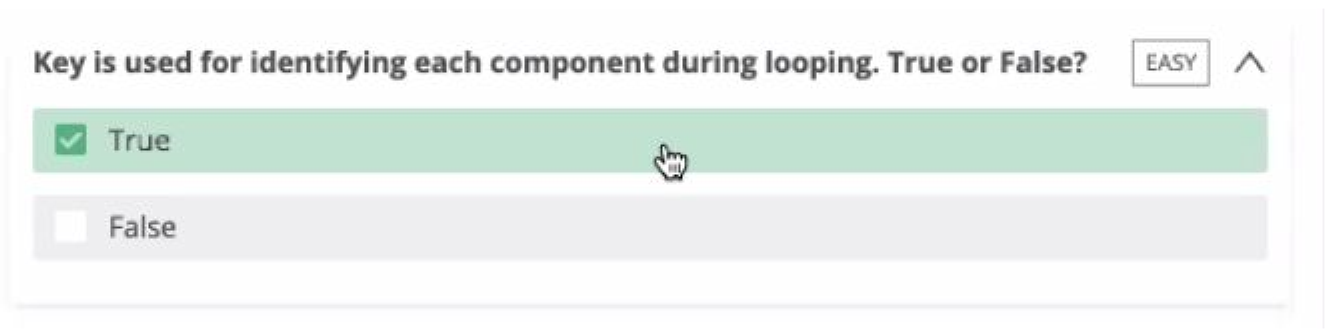
To make sure that the this keyword is voluntarily changed, which method should be used? Attempted - 39 (90.7%) EASY

map	20.51%
reduce	2.56%
[✓ bind]	79.49%
find	



When we loop a finite amount of components, what does React expect as a result? Attempted - 42 (85.71%) EASY

An object	9.52%
A set of elements	19.05%
[✓ An array of elements]	71.43%



Key is used for identifying each component during looping. True or False? EASY

[✓ True]

False