505 React

Meetup 2

Before we begin

Make sure you have the code from our last meet up and have followed all the set up directions in the slides from the last meetup

https://github.com/samanthaandrews/505-React-Meetup

Welcome to 505 React

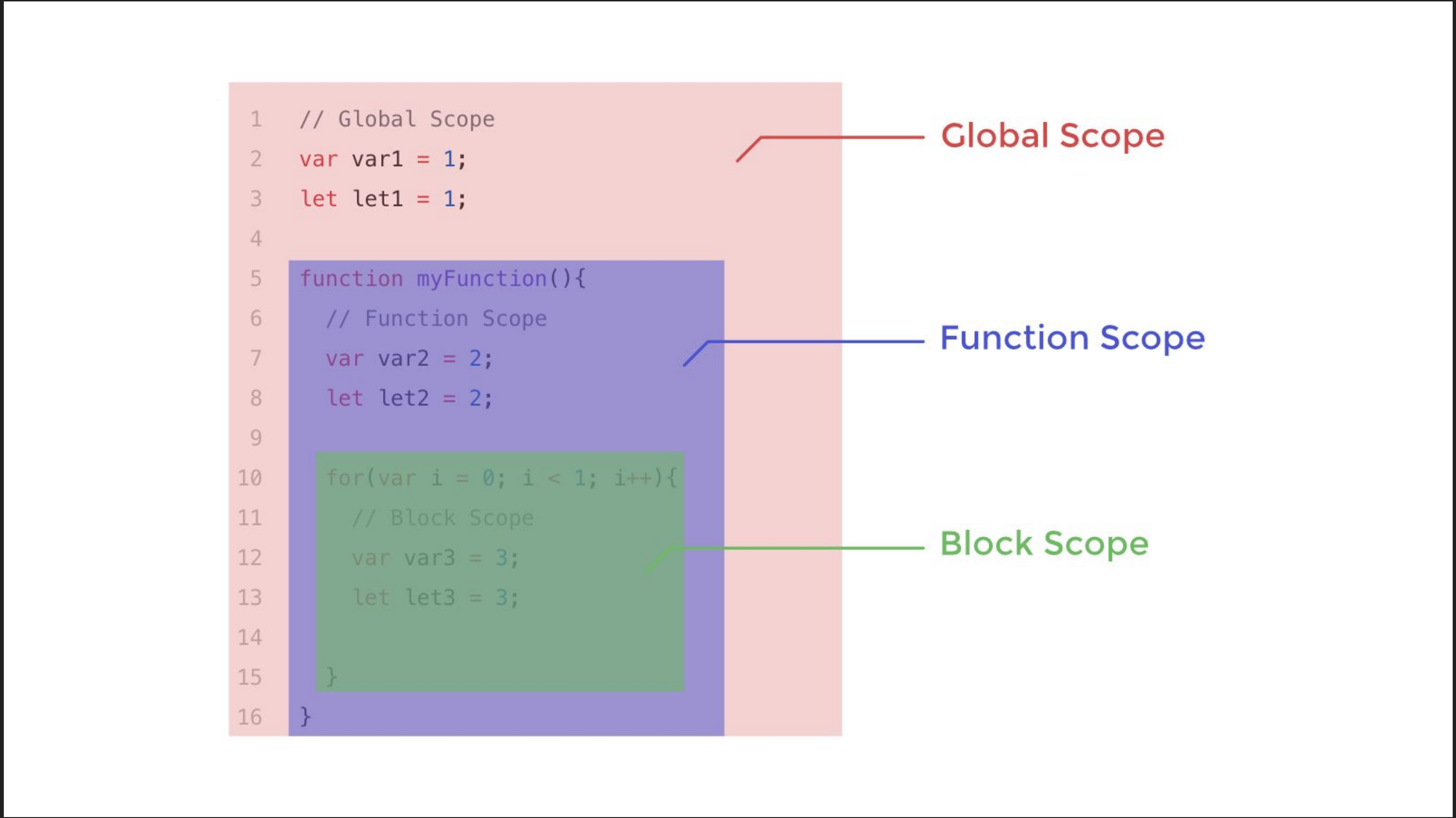
Tell us a little about yourself.

- Name
- Where you work
- What you know about React / React Native already
- What you want to know about React / React Native

ES6: `var`, `let`, and `const`

- ES6 came with the addition of `let` and `const`, which can be used for variable declaration.
- What makes them different from `var`? Scope, use, and hoisting

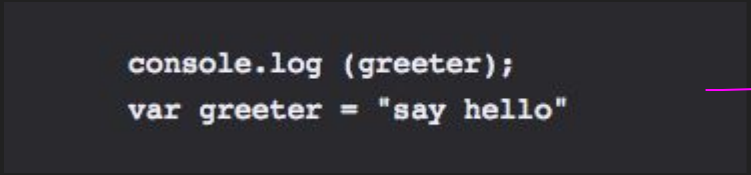
Scope



```
// Global Scope
var var1 = 1;
let let1 = 1;

function myFunction(){
  // Function Scope
  var var2 = 2;
  let let2 = 2;

  for(var i = 0; i < 1; i++){
    // Block Scope
    var var3 = 3;
    let let3 = 3;

  }
}
```

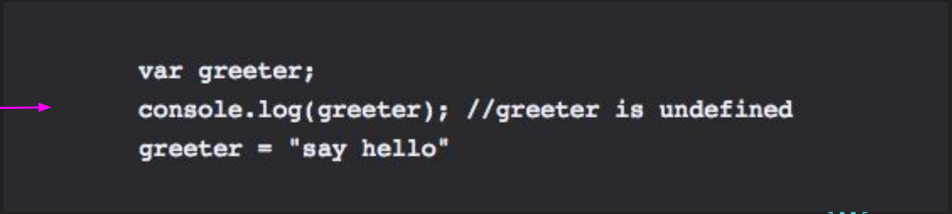
VAR

- Globally scoped OR function/locally scoped
- Can be redeclared and reassigned
- Hoisting of `var` – hoisting is a JS mechanism where variables and function declarations are moved to the top of their scope at code execution



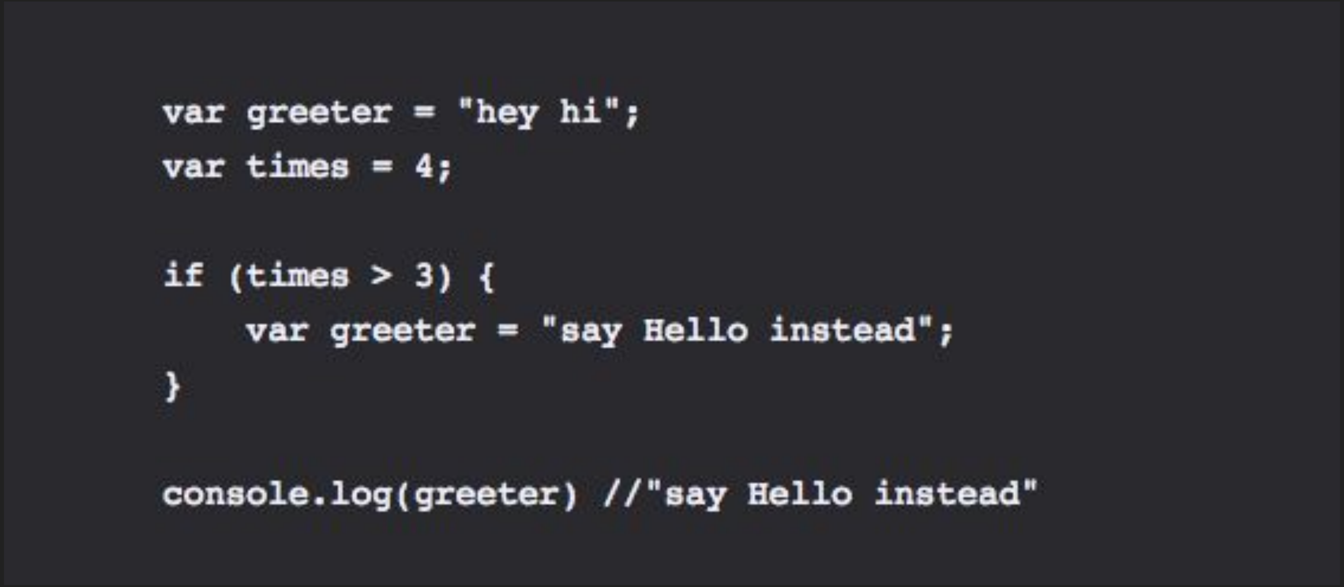
```
console.log (greeter);
var greeter = "say hello"
```

$\longrightarrow$



```
var greeter;
console.log(greeter); //greeter is undefined
greeter = "say hello"
```

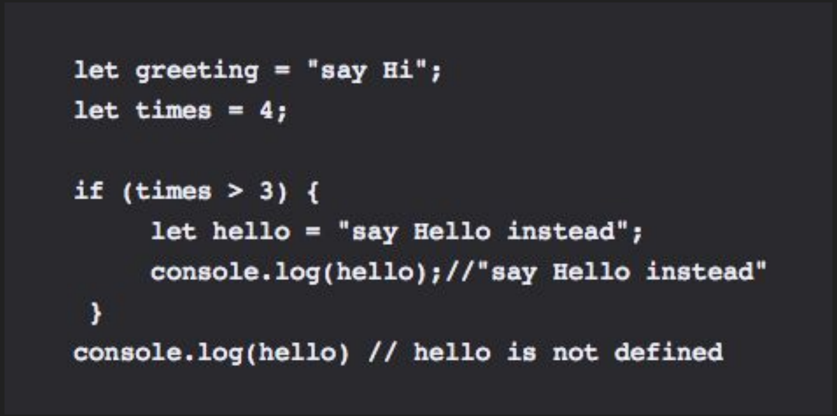
The problem with VAR



```
var greeter = "hey hi";
var times = 4;

if (times > 3) {
    var greeter = "say Hello instead";
}

console.log(greeter) //"say Hello instead"
```

LET

- Block scoped - a block is a chunk of code bounded by `{ }`
- Can be reassigned but not redeclared

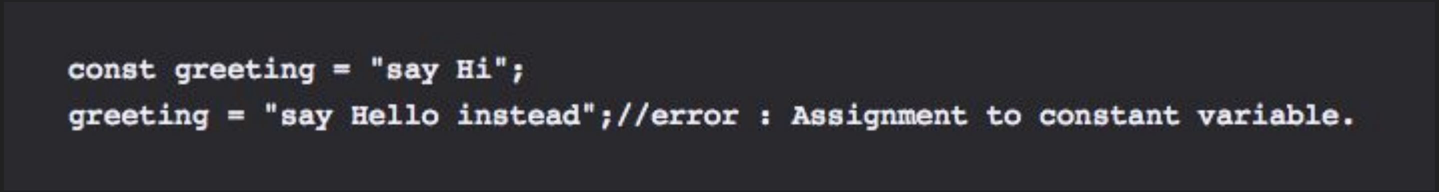


```
let greeting = "say Hi";
let times = 4;

if (times > 3) {
     let hello = "say Hello instead";
     console.log(hello);//"say Hello instead"
 }
console.log(hello) // hello is not defined
```

- Just like `var`, `let` declarations are hoisted to the top. But the let keyword is not initialized. So you will get a `Reference Error` instead of `undefined`.

CONST

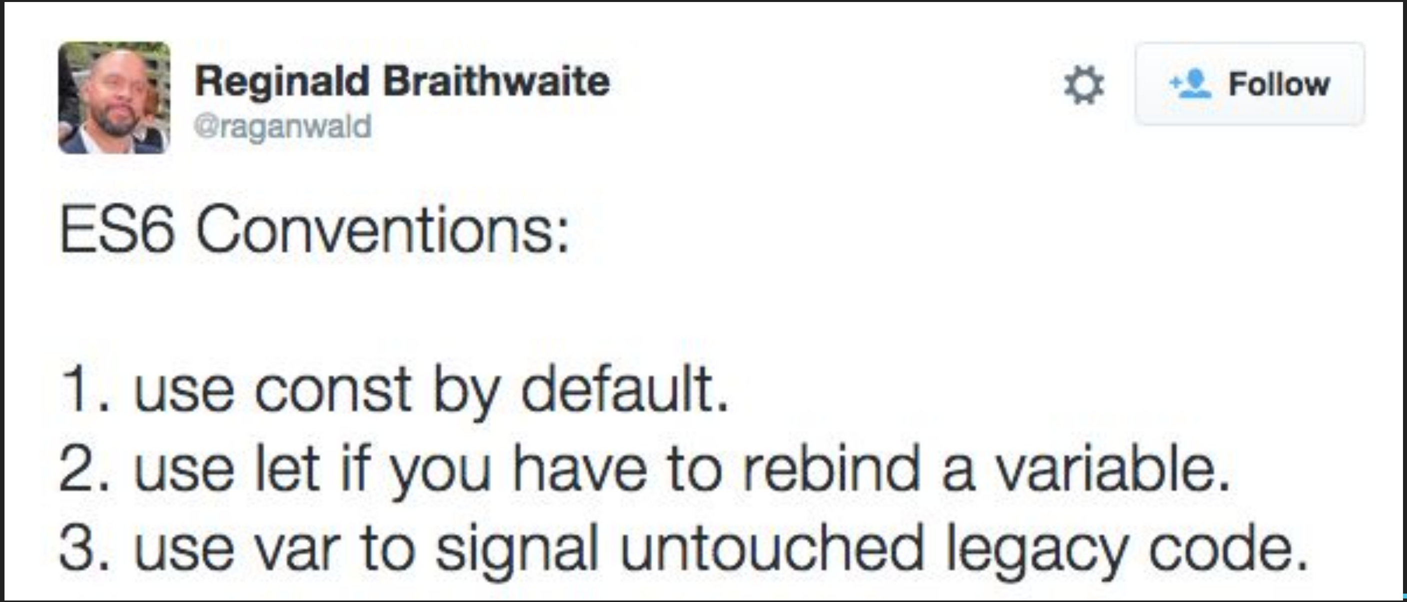
- Block scoped, just like `let`
- Cannot be redeclared or reassigned



```
const greeting = "say Hi";
greeting = "say Hello instead";//error : Assignment to constant variable.
```

- Just like `let,` declarations are hoisted to the top but are not initialized
- Disclosure: When you declare an object using const, its properties can be modified or added, but the variable itself cannot be redeclared

ES6: var, let, and const


Reginald Braithwaite
@raganwald

ES6 Conventions:

1. use const by default.
2. use let if you have to rebind a variable.
3. use var to signal untouched legacy code.

Ternary

- A ternary is a shorter way to do a conditional evaluation
- They implicitly return something regardless of the evaluation

```
let bar;
if (foo === true) {
  bar = ‘Hello World’
} else {
  bar = ‘Goodbye’
}

const bar = foo ? ‘Hello World’ : Goodbye’;
```

Ternary

- 3 parts: condition ? true code to execute : false code to execute;

```
const bar = foo ? ‘Hello World’ : ‘Goodbye’;
```

Firebase

- Firebase is a Backend-as-a-Service. It is your server, your API, and your datastore, all written so generically that you can modify it to suit most needs.
- We will be using it to store all of our chat messages and display them for everyone to see.

Firebase

Setup

- yarn add firebase / npm i --save firebase

src/firebase.js

```
import firebase from 'firebase';

const config = {
   apiKey: "AIzaSyD9HEJmF66X6Q5NR70U2dt7pFOIoQyw9ys" ,
   authDomain: "reactchat-5e8e9.firebaseapp.com" ,
   databaseURL: "https://reactchat-5e8e9.firebaseio.com" ,
   projectId: "reactchat-5e8e9",
   storageBucket: "reactchat-5e8e9.appspot.com" ,
   messagingSenderId: "888828995621"
};

firebase.initializeApp(config);

export default firebase;
```

App.js changes (componentDidMount)

```
componentDidMount = () => {
  const messagesRef = firebase.database().ref('messages');

  messagesRef.on('value', (snapshot) => {
    let messages = [];
    snapshot.forEach(element => {
      messages.push(`${element.val().username}: ${element.val().message}`);
    })
    this.setState({
      messages,
    })
  });
}
```

App.js changes (addNewMessage)

```
addNewMessage = message => {
   const messagesRef = firebase.database().ref('messages');
   messagesRef.push({
     username: 'jhonny#5',
     message,
   }, function(error) {
     if (error) {
       console.log(error)
     }else {
       console.log('success');
     }
   })
   //this.setState({ messages: [...this.state.messages , message] });
 }
```

Styled Components

- Styled Components are a way of writing CSS in JavaScript
- The syntax is similar to regular CSS
- https://www.styled-components.com/

Styled Components

- Styled Components are less verbose than normal CSS in JS
- The syntax is like normal CSS
- The allow you to conditionally render styles in the CSS declaration
- They are easy to test
- They allow you set themes

```
const NewStyledComponent = styled.div`
  color: #000000;
  margin: 25px;
`;
```

Styled Components

- They are created by declaring a new variable that extends an HTML element

```
const StyledImage = styled.img`
  width: 100%;
  margin: 50px;
`;
class App extends Component {
  render() {
    return (
      <StyledImage />
    );
  }
}
```

Styled Components

- They can also render the style conditionally

```
const mainColor = 'indianred'

const Title = styled.h1`
  color: ${props => props.color ||  'goldenrod'}
`;

class App extends Component {
  render() {
    return (
      <Title color={mainColor}>Mystagram</Title>
    );
  }
}
```

In terminal run the following command inside your app directory

```
npm install --save styled-components
```

Next import them into App.js

```
import styled from ‘styled-components`;
```