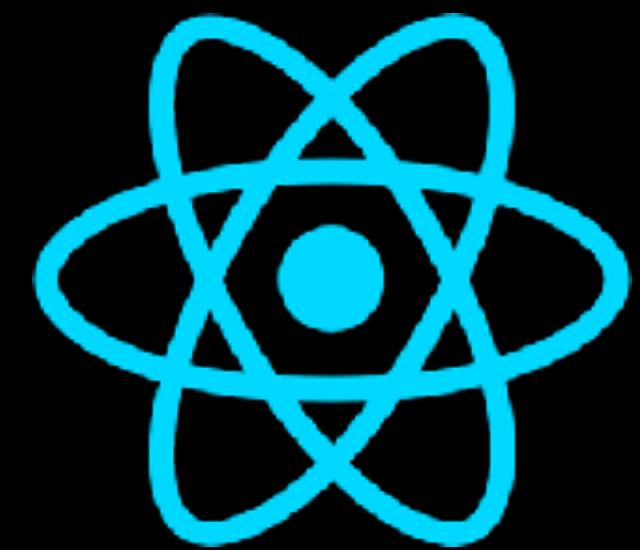


# React && React-native

A basic introduction



# Installation

- Node.js (& npm)
- Android SDK
- React Native

All of our code will run on both PC  
& Mac. But iOS can only be  
developed on a Mac.

# Android Development

- Install Java 8 or higher
- Install Android Studio
- Install Android SDK 25 (Nougat)
- Install Intel HAXM (x86 Android Emulator)



kurtesy

# Mac Installation

- brew install node
- brew install watchman\*
- npm i -g react-native-cli

# Windows Installation

- choco install nodejs.install
- choco install python2
- choco install jdk8



# React-Native

- `react-native init <app name>`
- `cd <app name>`
- `react-native run-android`



# React Native + NativeBase

- `react-native init <app-name>`
- `cd <app-name>`
- `yarn add native-base react-native-vector-icons`
- `react-native link react-native-vector-icons`
- `yarn remove babel-preset-react-native`
- `yarn add babel-preset-react-native@2.1.0`



kurtesy

# Mac Fix

- Close packager window
- watchman watch-del-all
- rm -rf node\_modules && npm install
- react-native run-android



# Windows Fix

- Close packager window
- Delete the node\_modules directory
- npm install
- react-native run-android



# react-native CLI

Command	Purpose
init [app name]	creates a react-native app
run-android [options]	builds app, starts on Android
run-ios [options]	builds app, starts on iOS
start [options]	starts webserver
new-library [options]	generates a native library bridge
bundle [options]	builds offline javascript bundle
unbundle [options]	builds unbundle javascript



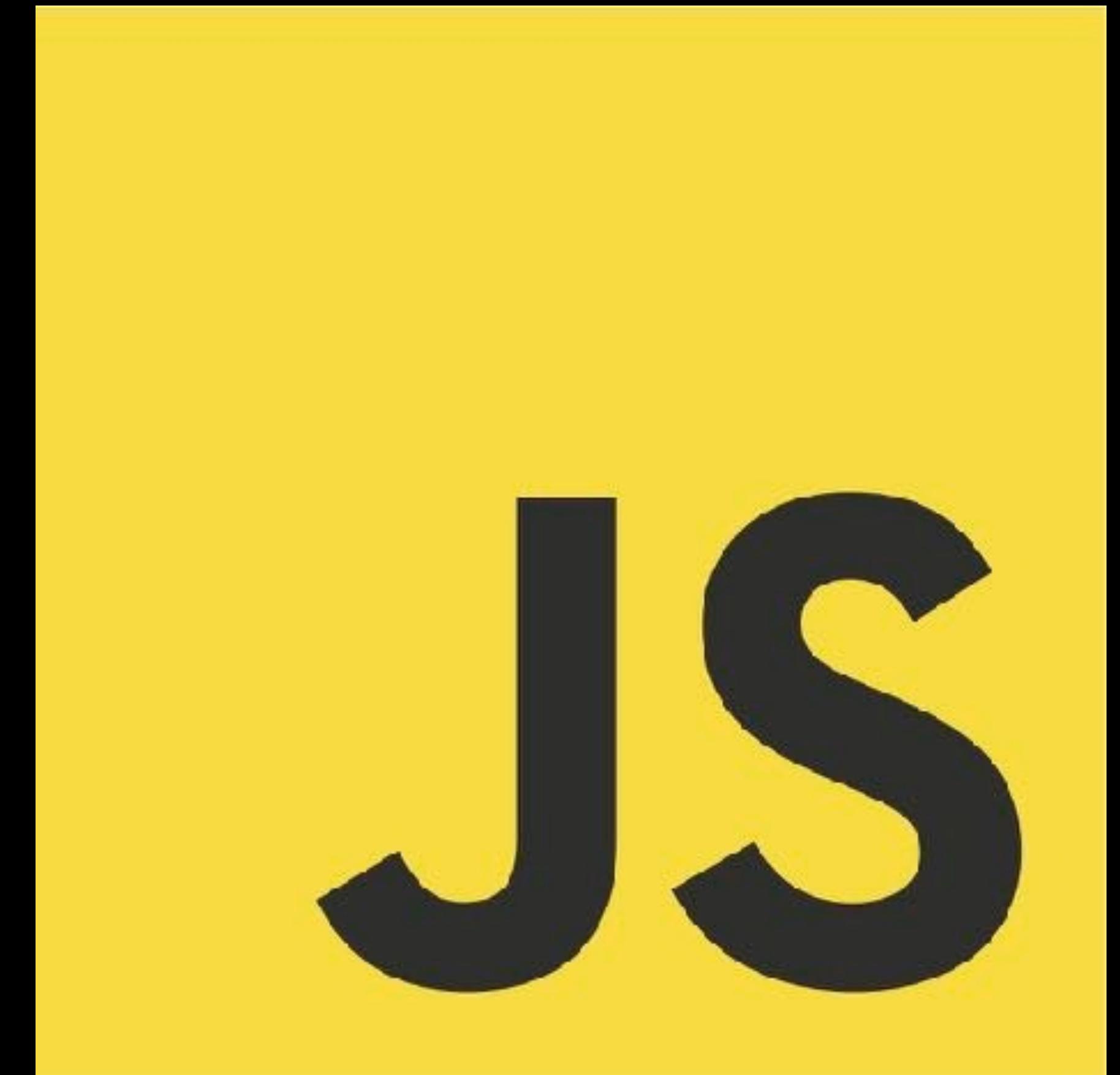
# react-native CLI

Command	Purpose
eject [options]	takes the shackles off
link [options] <packageName>	links all native dependencies
unlink [options] <packageName>	unlink native dependency
install [options] <packageName>	install+link native dependencies
uninstall [options] <packageName>	uninstall+unlink native
upgrade [options]	upgrade app's template files
log-android [options]	starts adb logcat
log-ios [options]	starts iOS device syslog tail

# Development Environment

- Command/Terminal window for CLI Commands
- The Packager window
- Android simulator or device
- Chrome browser
- Your IDE





kurtesy

# ECMAScript Versions

Version	Date
ES1	June 1997
ES2	June 1998
ES3	December 1999
ES4	DOA 2006
ES5	December 2009
ES2015 / ES6	June 2015
ES2016 / ES7	2016



# Collection Operators

- `.isArray()`
- `.every()`
- `.forEach()`
- `.indexOf()`
- `.lastIndexOf()`
- `.some()`
- `.map()`
- `.reduce()`
- `.filter()`



# map

```
let junk = [1, 2, 3, 4, 'Alpha', 5, {name: 'Jason'}];
```

```
let letters = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K'];
let nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20];
console.log(nums);
```

```
// map iterates over all of the elements and returns a new array with the same
// number of elements
let nums2 = nums.map((elem) => elem * 2);
console.log(nums2);
/// [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40]
```

# filter

```
let junk = [1, 2, 3, 4, 'Alpha', 5, {name: 'Jason'}];
let letters = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K'];
let nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20];
console.log(nums);

// filter iterates over the array and returns a new array with only the elements
// that pass the test
let nums3 = nums.filter((elem) => !(elem % 2));
console.log(nums3);
/// [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```

# reduce

```
let junk = [1, 2, 3, 4, 'Alpha', 5, {name: 'Jason'}];
let letters = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K'];
let nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20];
console.log(nums);

// reduce iterates over the array passing the previous value and the current
// element it is up to you what the reduction does, let's concatenate the strings
let letters2 = letters.reduce((previous, current) => previous + current);
console.log(letters2);
/// ABCDEFGHIJK

// reduceRight does the same but goes from right to left
let letters3 = letters.reduceRight((previous, current) => previous + current);
console.log(letters3);
/// KJIHGFEDCBA
```

# let

- let allows us to create a block scoped variables
- they live and die within their curly braces
- var is considered deprecated
- best practice is to use let instead of var



# let

```
// let allows us to create block scoped variables
// they live and die within the curly braces
let val = 2;
console.info(`val = ${val}`);
{
  let val = 59;
  console.info(`val = ${val}`);
}
console.info(`val = ${val}`);
```

# const

- const creates a variable that can't be changed
- best practice is to make any variable that should not change a constant
- does not apply to object properties or array elements



# const

```
const name = 'Troy';
console.info(`My name is ${name}`);
// the line below triggers a type error
name = 'Miles';
```

# Template strings

- Defined by using opening & closing back ticks
- Templates defined by `$ {JavaScript value}`
- The value can be any simple JavaScript expression
- Allows multi-line strings  
(return is pass thru)



# Template strings

```
let state = 'California';
let city = 'Long Beach';
console.info(`This weekend's workshop is in ${city}, ${state}.`);

// template strings can run simple expressions like addition
let cup_coffee = 4.5;
let cup_tea = 2.5;
console.info(`coffee: $$\{cup_coffee\} + tea: $$\{cup_tea\} = $$\{cup_coffee + cup_tea\}.`);

// they can allow us to create multi-line strings
console.info(`This is line #1.
this is line #2.`);
```

# Arrow functions

- Succinct syntax
- Doesn't bind its own *this*, *arguments*, or *super*
- Facilitate a more functional style of coding
- Can't be used as constructors



# Arrow functions

- When only one parameter, parenthesis optional
- When zero or more than one parameter, parenthesis required



# Arrow function

```
let anon_func = function (num1, num2) {  
    return num1 + num2;  
};  
console.info(`Anonymous func: ${anon_func(1, 2)}`);  
  
let arrow_func = (num1, num2) => num1 + num2;  
console.info(`Arrow func: ${arrow_func(3, 4)}`);
```

# this

- this is handled different in arrow functions
- In anonymous function this is bound to the global object
- In arrow function this is what it was in the outer scope



# Spread syntax

Expands an expression in places where multiple arguments, elements, or variables are expected

```
Let obj = {'a': 1, 'b': 2}  
{...obj, 'c':3}
```

```
{'a': 1, 'b': 2, 'c':3}
```



# The spread operator

```
11
12
13 const myArray = ['Bob', 'Sue', 'Fido'];
14 function printFamily(person1, person2, pet) {
15   console.info(`Person 1: ${person1}, Person 2: ${person2}, and their pet: ${pet}`);
16 }
17 printFamily(...myArray);
18
19
```

# Generator Function\*

- function\* is called a generator
- A generator is a function which can be exited and re-entered
- statements are executed until a yield is encountered
- the generator then pauses execution until the yield returns
- if a return statement is encountered, it finishes the generator



# Yield

- `yield` is a keyword which pauses the execution of a generator function
- `yield` can return a value to the generator



# Class

- Syntactic sugar over JavaScript use of function constructors
- JavaScript uses proto-typical inheritance
- If a class **extends** another, it must include **super()** as first instruction in its **constructor**
- Only create a constructor if it does something



# Class

- Syntactic sugar over JavaScript use of function constructors
- JavaScript uses proto-typical inheritance
- If a class **extends** another, it must include **super()** as first instruction in its **constructor**
- Only create a constructor if it does something



# import

- Imports functions, objects, or primitives from other files
- `import <name> from "<module name>";`
- `import {name } from "<module name>";`
- `import * as Greetings from "<module name>";`
- relative path indicates not an npm package



# export

- `export <var a>`
- `export {a, b};`

# export default

- only one per file
- common pattern for libraries
- `const Greetings = {sayHi, sayBye};`
- `export default Greetings;`
- `export default {sayHi, sayBye};`

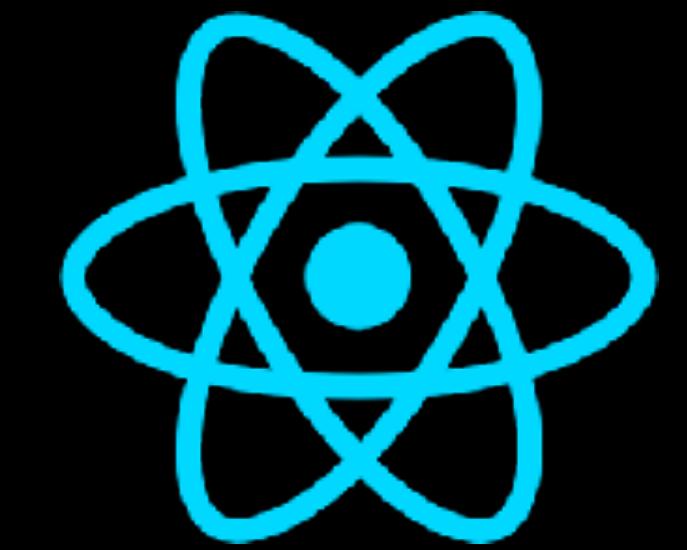
# Importing Old School JS

- Many JS libraries don't support the new syntax
- How do we use them?
- `import 'jquery';`



# ReactJS

A JS library for building user interfaces, open sourced by Facebook



kurtesy

# Motivation behind ReactJS

Previously for AngularJS...



kurtesy

# Motivation behind React

AngularJS uses **dirty checking**  
*scans* the scope for changes



Need to modify the DOM tree  
incessantly and a lot

# React Concepts

- Component based
- Virtual DOM
- JSX
- Immutability
- One way data flow



# React Concepts

- Component based
- Virtual DOM
- JSX
- Immutability
- One way data flow



# Components

<input type="text" value="Search..."/>	
<input type="checkbox"/> Only show products in stock	
Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

<input type="text" value="Search..."/>	
<input type="checkbox"/> Only show products in stock	
Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

# React Concepts

- Component based
- **Virtual DOM**
- JSX
- Immutability
- One way data flow



# Virtual DOM

It is lightweight JavaScript object which is a **copy** of Real DOM.



# How ReactJS works

Render an element

Every single virtual DOM object gets updated



kurtesy

# Inefficient?

Maybe not...



kurtesy

# Two Virtual Doms

Previous State  
VD

Current State  
VD

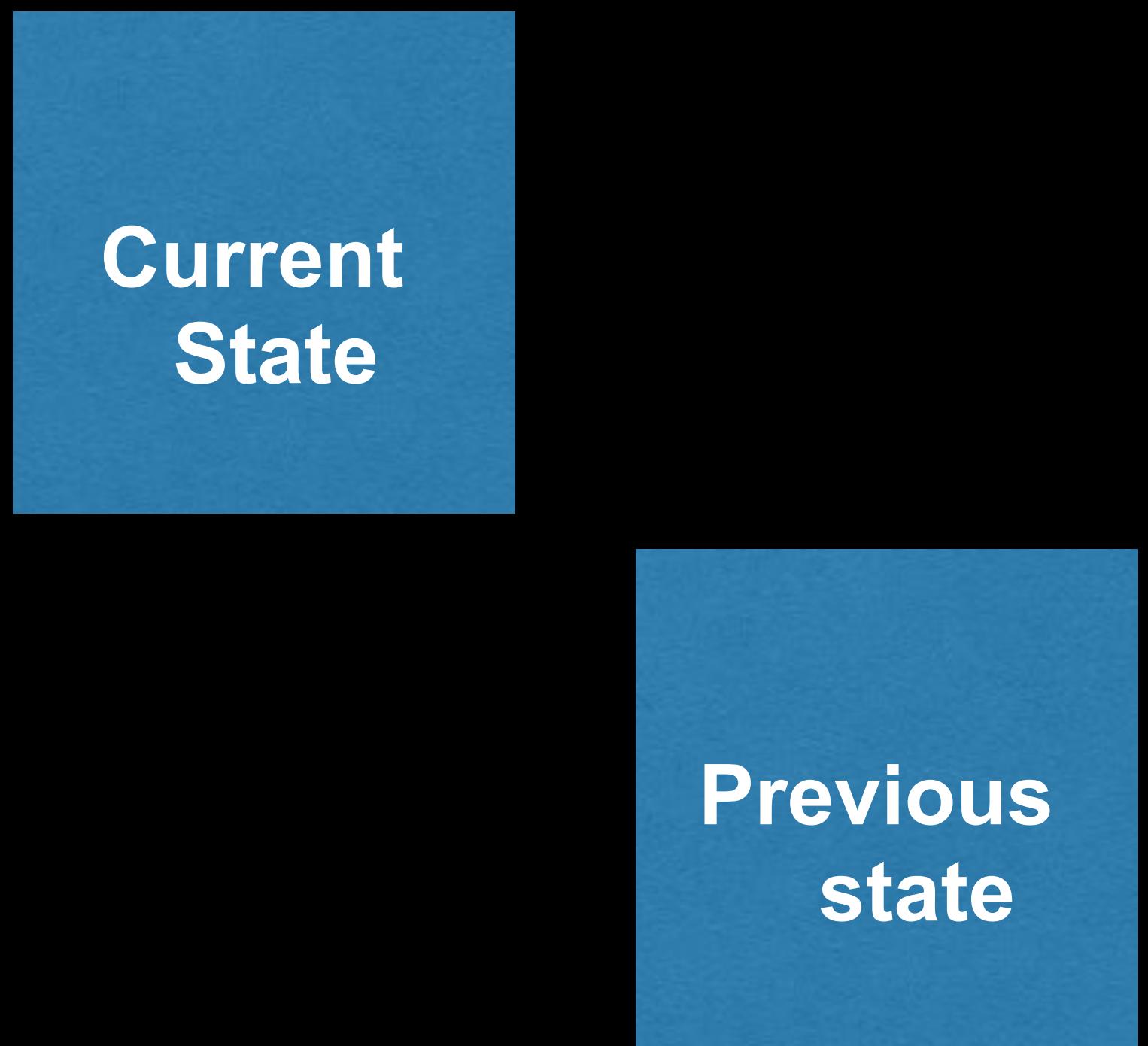
# “Diffing” algo

By comparing the new virtual DOM with the earlier version, React finds out exactly which virtual DOM objects have changed

Now, React smartly updates only those objects on the real HTML DOM.



# Virtual DOM



kurtjesy

Current  
State

Previous  
state

Updates only  
relevant  
changes

DOM



kurtesy

# Pretty smart eh?

# Lifecycle Events

Event	When
componentWillMount	invoked once before rendering
componentDidMount	invoked after component loaded
componentWillReceiveProps	invoked when receiving new props
shouldComponentUpdate	asks if component should update
componentWillUpdate	invoked before rendering new props
componentDidUpdate	invoked after rendered new props
componentWillUnmount	invoked before component removed



## Initialization

setup props and state

componentWillMount

render

componentDidMount

## Mounting

props

componentWillReceiveProps

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

## Updation

states

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

## Unmounting

componentWillUnmount



# JSX

- JSX stands for JavaScript XML
- A templating language
- Embed any javascript inside of a JSX template by wrapping it in curly braces (these: {})



# Babel

Compiles JSX

**React.createElement()** calls

Construct the DOM



```
const element = (
  <h1 className="greeting"> Hello, world!
</h1>
);
```

```
const element = React.createElement( 'h1',
{className: 'greeting'}, 'Hello, world!'
);
```

# React Concepts

- Component based
- Virtual DOM
- JSX
- **Immutability**
- One way data flow



# Immutability



# Example of Immutability

```
this.state=  
{ passengers:  
[ 'Simon' , 'Taylor' ] }
```

Suppose we want to add a passenger called **Vincent** to the passengers array...

# Example of Immutability

```
let updatedPassengers = this.state.passengers;
```

# Example of Immutability

```
let updatedPassengers = this.state.passengers;  
updatedPassengers.push('Vincent');
```

# But is it really correct?



kurtesy

**‘Push’ mutates the state directly**



kurtesy

```
let updatedPassengers =  
this.state.passengers.concat('Vincent');
```



kurtesy

# Example of Immutability

**Create copies** of the objects in this.state and manipulate the copies

**Map**, **filter** and **concat** => non-destructive array methods => methods that will return a new object or array with the desired mutations

# React Concepts

- Component based
- Virtual DOM
- JSX
- Immutability
- **One way data flow**



# Two way binding vs Unidirectional data flow

AngularJS

Change model -> **modifies view**

Change input field in view -> **modifies model**

Lead to cascading updates



kurtesy

# Props

**Properties** passed from the parent

```
<NewComponent prop={toBePassedIn}>
```



kurtesy

# State

**State == data that is going to change**



kurtesy

# Functional components vs Class components



kurtesy

# Functional

```
const App = () => {  
  return (<div>Hello World!</div>);  
};  
  
export default App;
```

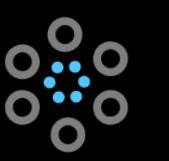
# Class

Local state is a feature available only to classes

# HelloWorld.js

```
import React, { Component } from 'react';
class HelloWorld extends Component {
  render() {
    return (
      <div>Hello {this.props.name}</div>
    );
  }
}
```

# How to change the state?



kurtesy

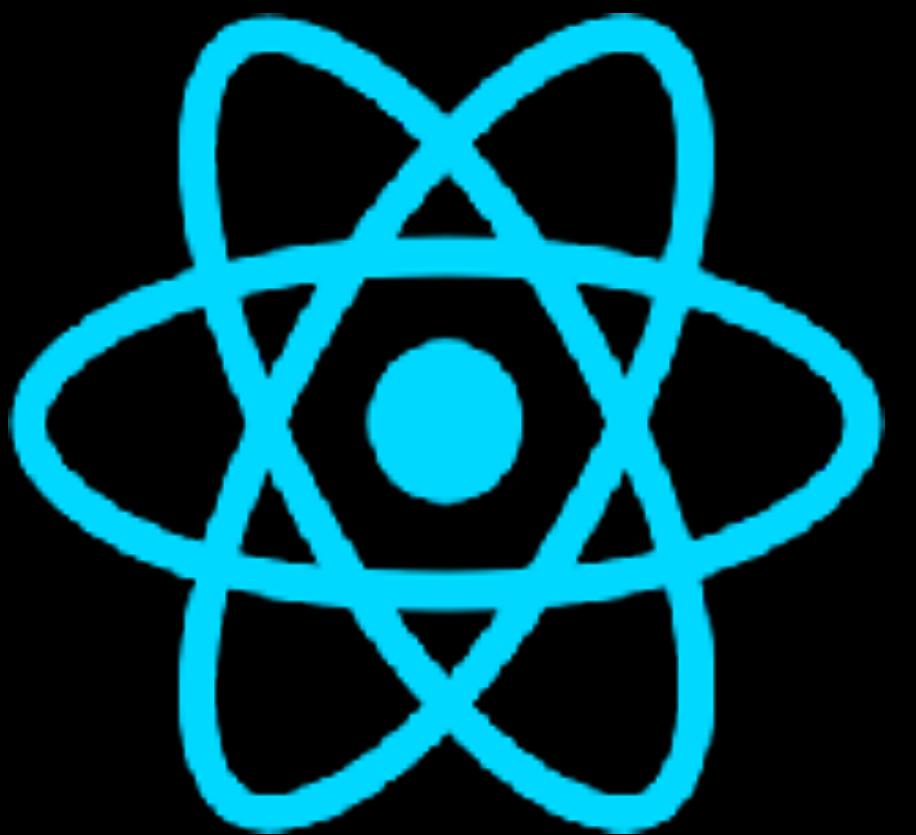
```
this.setState({  
  greeting: 'Bonjour'  
})
```



```
this.state.greeting = 'Bonjour';
```

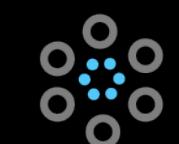


# REACT-NATIVE



kurtesy

# Who uses React Native?



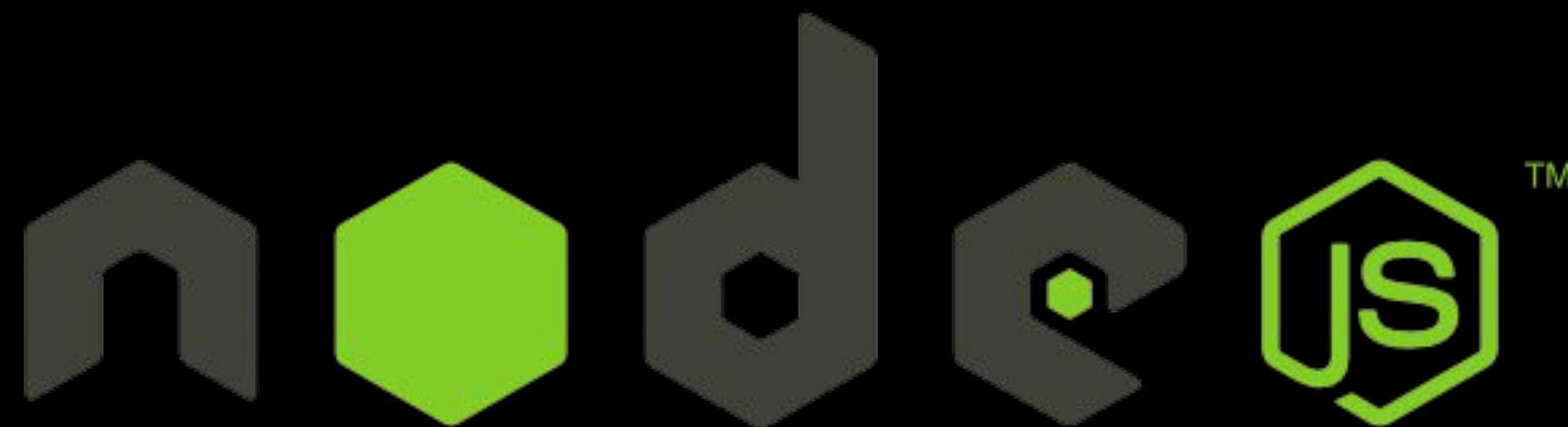
kurtesy

# React Native: Features

- Native apps (native UI components)
- No need to recompile every time
- Reusable code for iOS and Android (80%)
- Communication with native functions
  - Objective-C, Java



# Requirements

A screenshot of a terminal window titled "jordan — bash — 96x26". The window displays the output of the "ls -l" command, listing the contents of the user's home directory. The output shows various folders and their details, such as permissions, last modified date, and size.

```
Shard:~ jordan$ ls -l
total 0
drwx-----+ 10 jordan staff 340 12 Jun 17:00 Desktop
drwx-----+ 13 jordan staff 442 27 May 15:03 Documents
drwx-----+ 172 jordan staff 5848 12 Jun 17:16 Downloads
drwx-----@ 27 jordan staff 918 11 Jun 23:14 Dropbox
drwx-----@ 75 jordan staff 2550 11 Jun 23:14 Library
drwx-----+ 8 jordan staff 272 17 Apr 17:20 Movies
drwx-----+ 8 jordan staff 272 12 Jun 10:56 Music
drwx-----+ 33 jordan staff 1122 9 May 10:48 Pictures
drwxr-xr-x+ 5 jordan staff 170 23 Mar 12:17 Public
drwxr-xr-x  3 jordan staff 102 11 Jun 17:03 Sites
Shard:~ jordan$
```

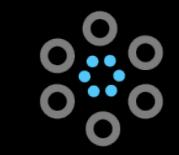
# React Native Advantages

- **JavaScript** – Existing JavaScript knowledge
- **Code sharing** – Share most code on different platforms
- **Community** – Community around React and React Native is large
- **Native Look and Feel**

# React Native Limitations

- **Native Components** – might need to write some platform specific code
- **Not even version 1 yet**





kurtesy

# What's the difference?



kurtesy

# Answer:

- **Cordova(with ionic)**: create HTML, style with CSS, and write all the logic with JavaScript
- **React-Native**: Visual components (reusable UI elements) are rendered as a native UI

# 3 threads

UI Thread

JS Thread

Native  
Modules  
Thread



kurtesy

# Environment for React Native

**Method 1:** create-react-native-app through Expo

(to use use custom native modules, you have to eject)

**Method 2:** react-native cli

(for the braver souls and those who never die before)



kurtesy

# Method 2

- `npm install -g react-native-cli`
- `react-native init reactTutorialApp`
- `cd reactTutorialApp`
- `react-native start`
- (start in a new terminal) `react-native run-ios`
- `react-native run-android`



# Method 2

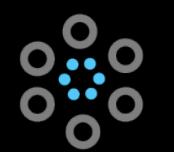
- node/npm
- HomeBrew
- Watchman
- react-native-cli
- XCode/Android Studio



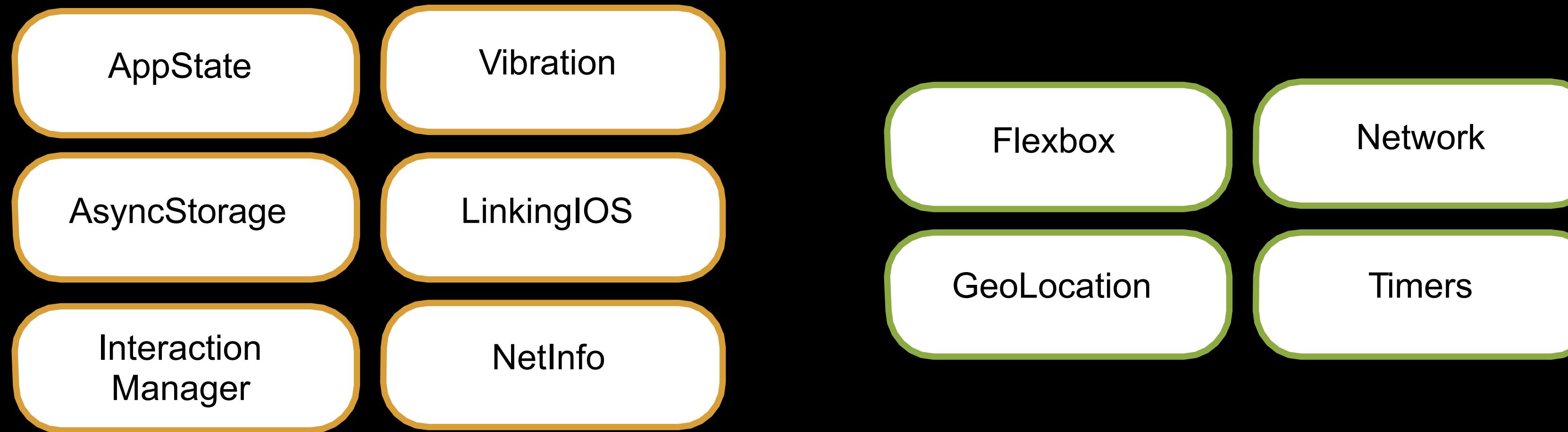
kurtesy

# UI Component Library

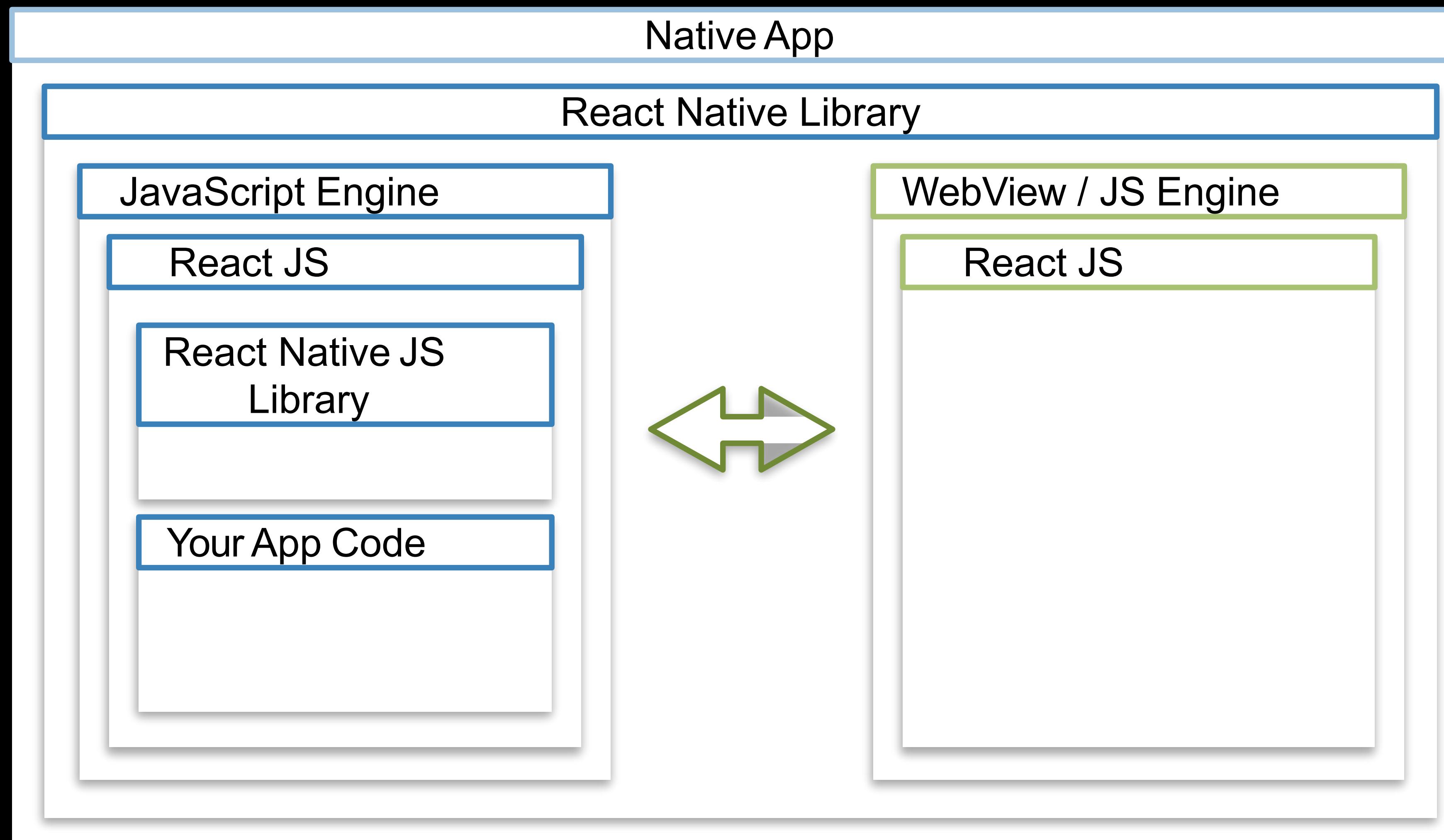
ActivityIndicator	Picker	Touchable	ActionSheet
Date Picker	ScrollView	TouchableOpacity	Alert
Image	Slider	Touchable Highlight	Border
ListView	TabBar	Touchable WithoutFeedback	StatusBarIOS
MapView	Text	View	StatusBarIOS
Navigator	TextInput	WebView	CameraRoll



# Device APIs & PolyFills



# React Native can work with existing apps



## Initialization

setup props and state

componentWillMount

render

componentDidMount

## Mounting

props

componentWillReceiveProps

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

## Updation

states

shouldComponentUpdate

componentWillUpdate

render

componentDidUpdate

## Unmounting

componentWillUnmount

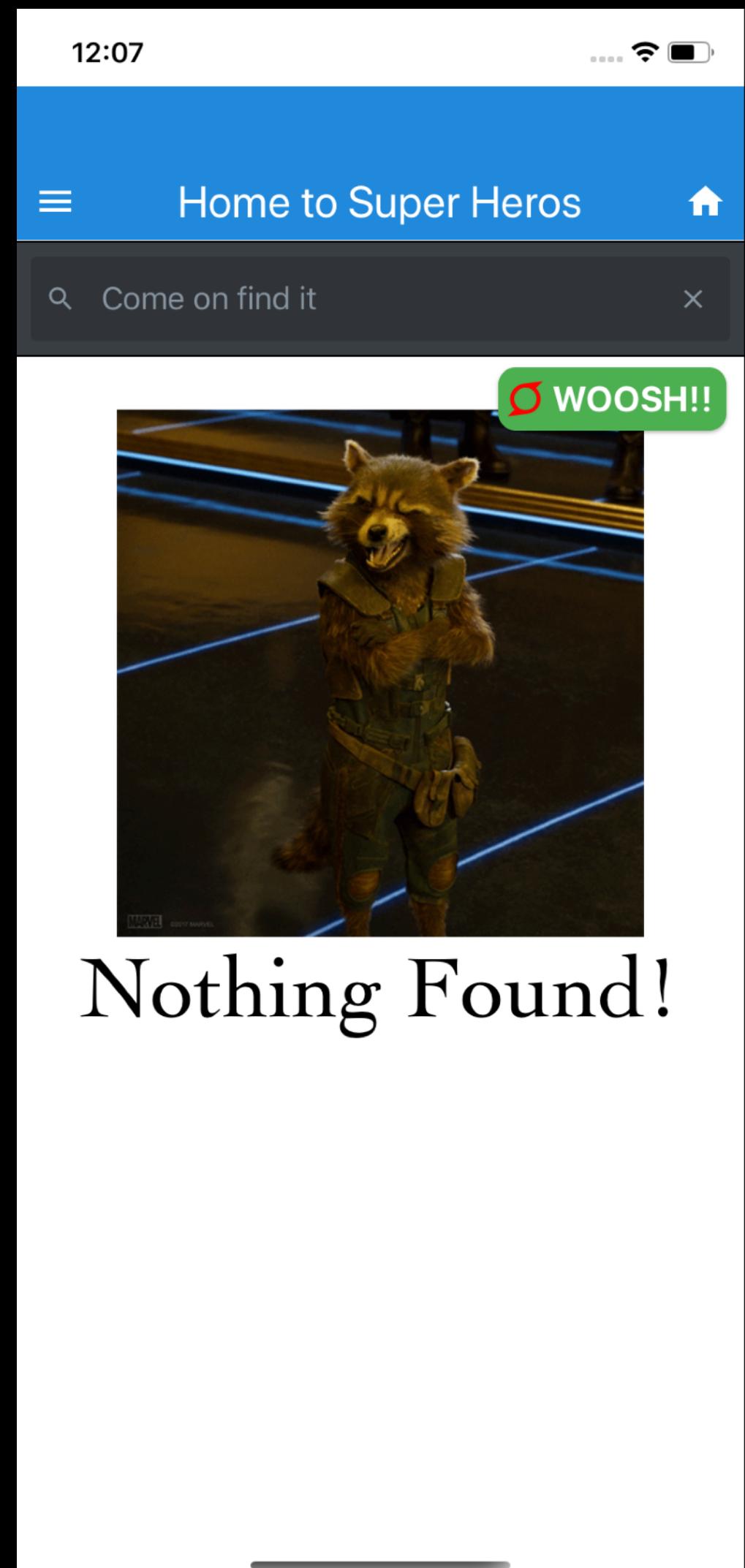
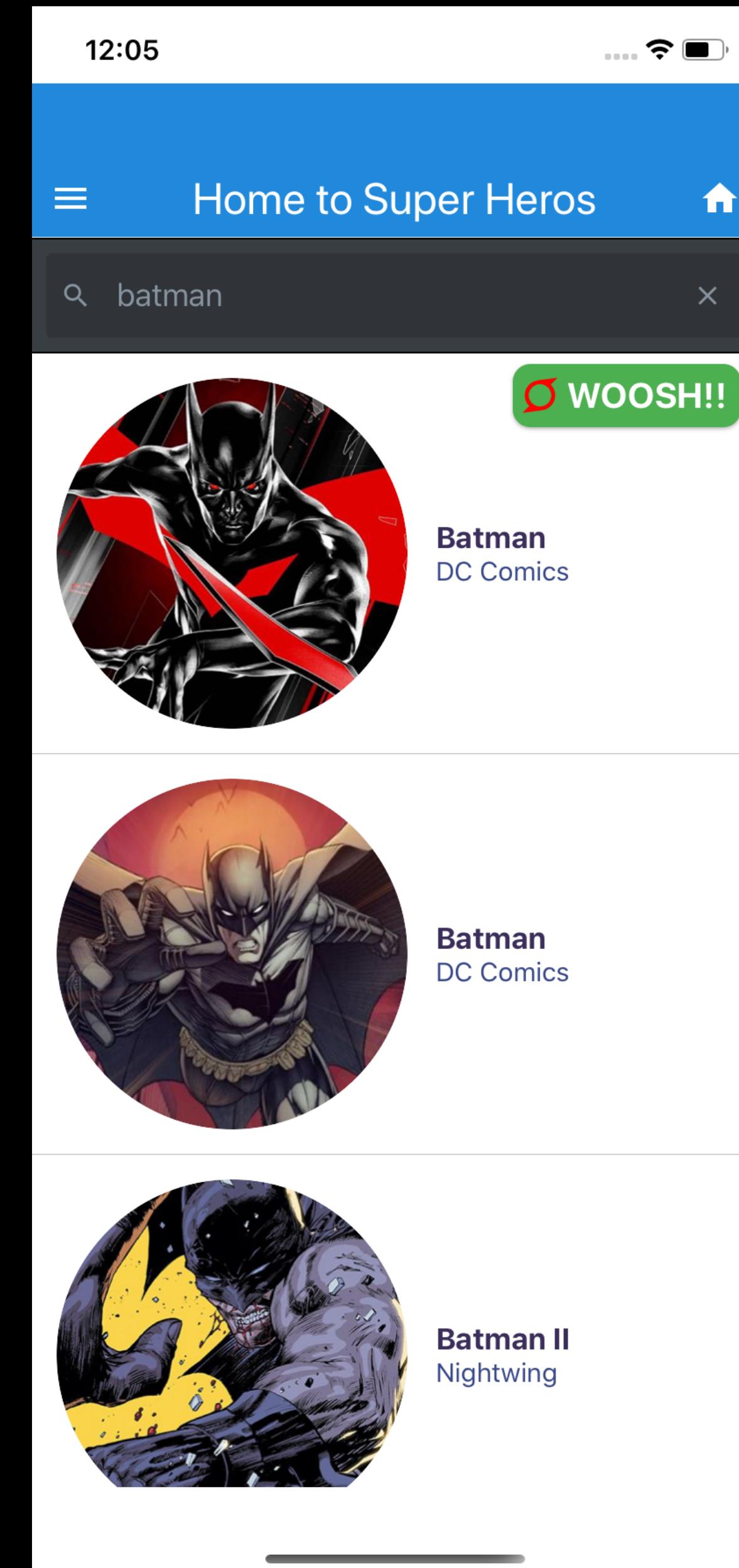
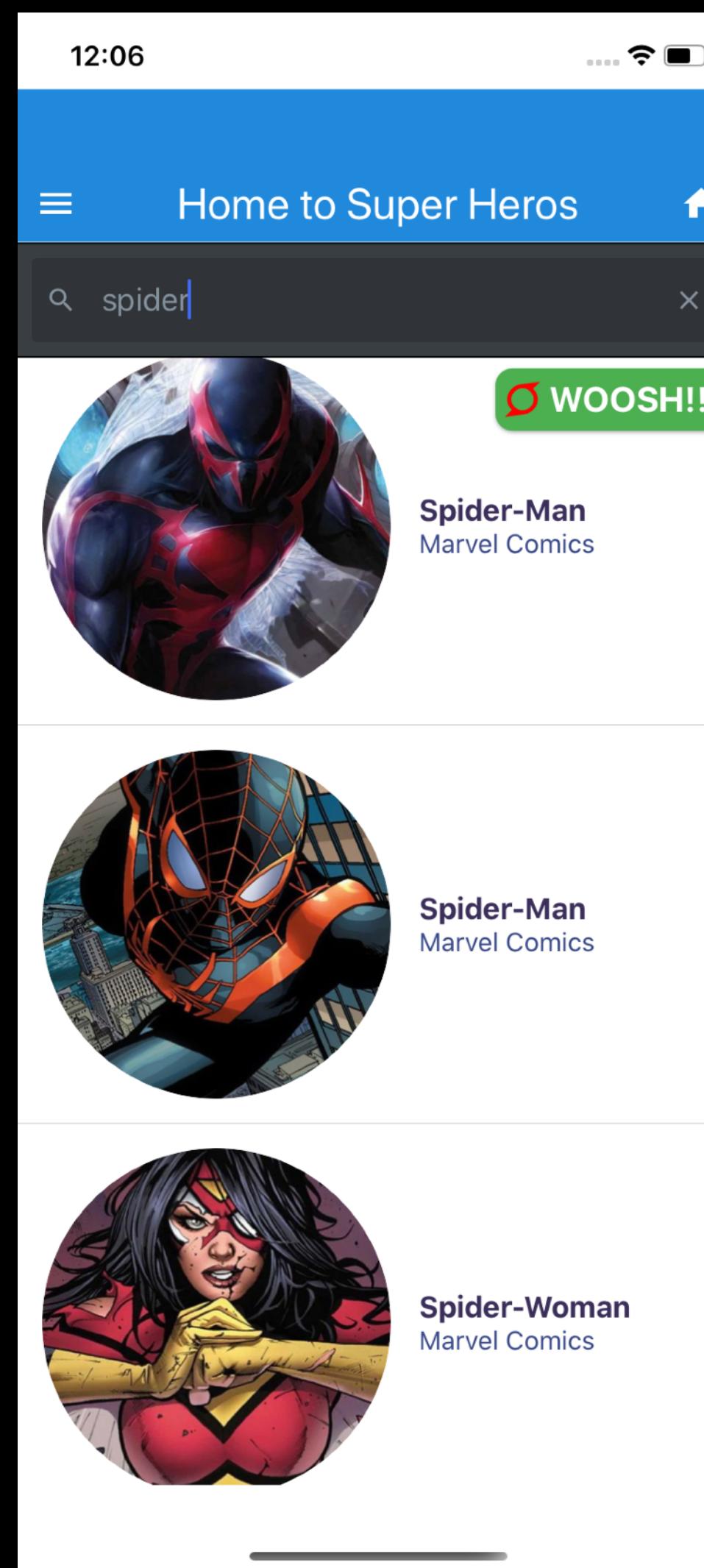


# Debugging



kurtesy

# A short exercise...



<https://github.com/kurtesy/HomeToSuperheros>

# Core Components

- View
- Stylesheet
- Text
- Flexbox
- Image
- TextInput
- Fetch or Use Axios



kurtesy

# View

`<View></View> == <div></div>`



kurtesy

# Styling

```
const styles = StyleSheet.create( {  
    container: {  
        flex: 1,  
        justifyContent: 'center',  
        alignItems: 'center',  
        backgroundColor: '#F5FCFF',  
    },  
    welcome: {  
        fontSize:  
            20,  
        textAlign: 'center',  
        margin: 10  
    }  
} );
```

# Text

<Text></Text> == <p></p>

# Flexbox

- `flexDirection: column, row`
- `justifyContent: flex-start, center, flex-end, space-around, and space-between`
- `alignItems: flex-start, center, flex-end`



# Image

```
<Image source={require('./test.png')} />
```

```
<Image source={{uri: 'https://  
facebook.github.io/react/img/  
logo_og.png'}} {}
```

# TextInput

Attributes:

- `onChangeText: function`
- `value: string`

# Button

Attributes:

- `onPress: function`
- `title: string`



kurtesy

# Fetch

```
fetch(url, {  
  method:  
  
  headers:  
  
  body:  
} ) .then( () => {  
  
  ...  
  
}) )
```

# Differences between React vs React Native?

# Answer:

- React-Native doesn't use HTML to render the app
- Slightly different syntax e.g. `onPress` vs `onClick`
- Publishing

# Android RN

1. Generate debug.keystore

```
keytool -genkey -v -keystore ./android/app/debug.keystore -alias androiddebugkey -keyalg RSA -keysize 2048 -validity 10000
```

2. Change below in this file: android/gradle/wrapper/gradle-wrapper.properties

```
distributionUrl=https\://services.gradle.org/distributions/gradle-6.0.1-all.zip  
to  
distributionUrl=https\://services.gradle.org/distributions/gradle-6.5-bin.zip
```

3. add some optional modules in android/app/build.gradle,

```
dependencies {  
    // For animated GIF support  
    implementation 'com.facebook.fresco:fresco:2.0.0'  
    implementation 'com.facebook.fresco:animated-gif:2.0.0'  
}
```

4. use GIF with ProGuard, you will need to add this rule in proguard-rules.pro :

```
-keep class com.facebook.imagepipeline.animated.factory.AnimatedFactoryImpl {  
    public AnimatedFactoryImpl(com.facebook.imagepipeline.bitmaps.PlatformBitmapFactory, com.facebook.imagepipeline.core.ExecutorSupplier);  
}
```

5. Yarn android-bundle

6. Yarn android

7. To Clean the android build run this

```
cd android &&./gradlew clean
```

# Some RN Utilities and IOS setup

1. Install node: brew install node
2. Install watchman: brew install watchman
3. Install XCode from the App Store or Apple's Developer Portal
4. Install cocoapods: sudo gem install cocoapods
5. Install yarn: npm install -g yarn
6. Create the project: react-native init SuperHeroHome
7. Change folder and kickstart the simulator:  
cd SuperHeroHome && react-native run-ios

List all devices

```
xcrun simctl list devices
```

Npm I -g react-devtools

For windows user

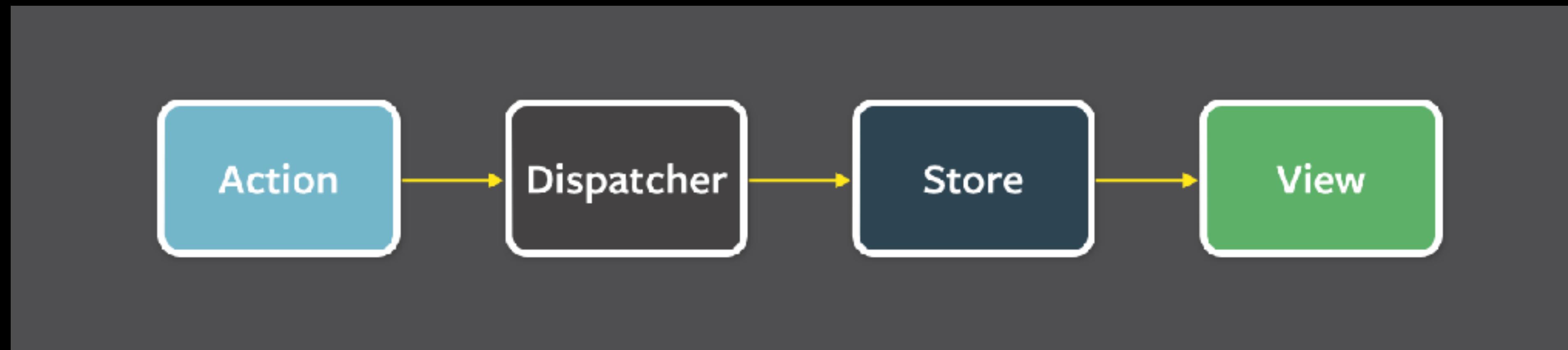
Clean Up:

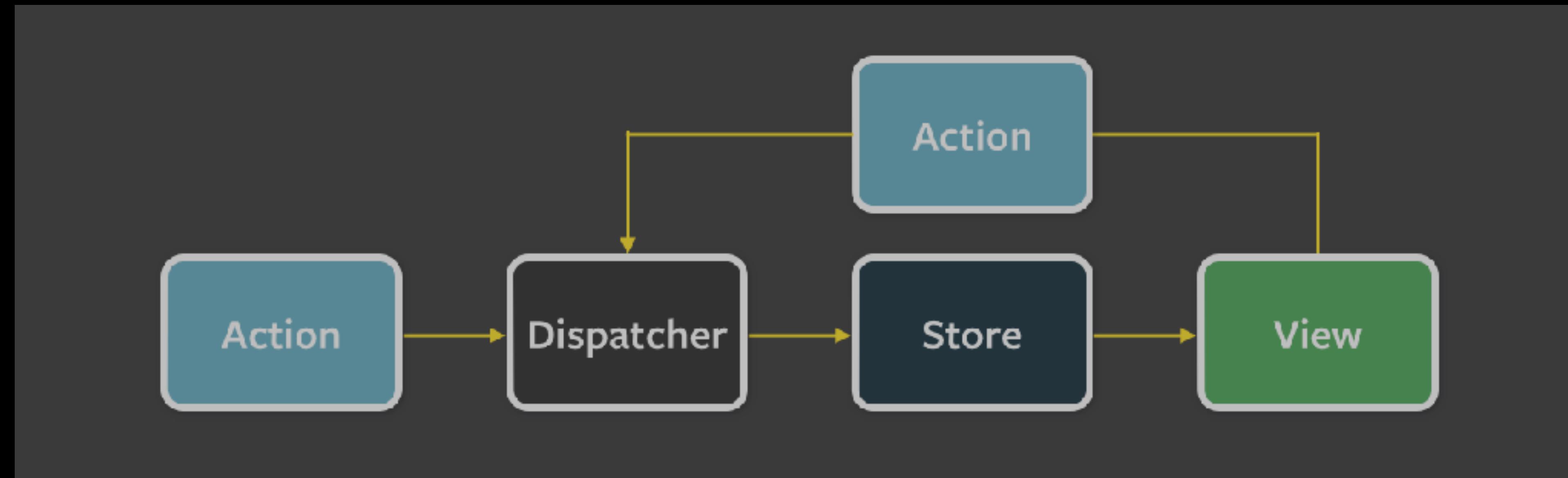
```
npx react-native clean-project-auto  
rm -rf ~/Library/Developer/Xcode/DerivedData  
rm -rf node_modules && rm -rf ios/build/* && rm -rf android/build/* && rm -rf $TMPDIR/react-* && yarn cache clean && yarn
```

Debugging:

```
npm install -g react-devtools  
react-devtools
```

# Add ons: Redux, Navigation and Hooks







# Redux



kurtesy

# Additional exercises:

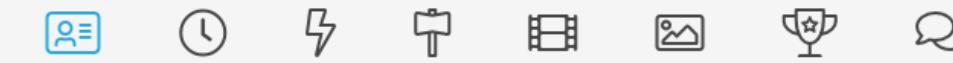
Add a component in our app to press a list item and see more details about our superhero

Reference - <https://www.superherodb.com/>



## Origin

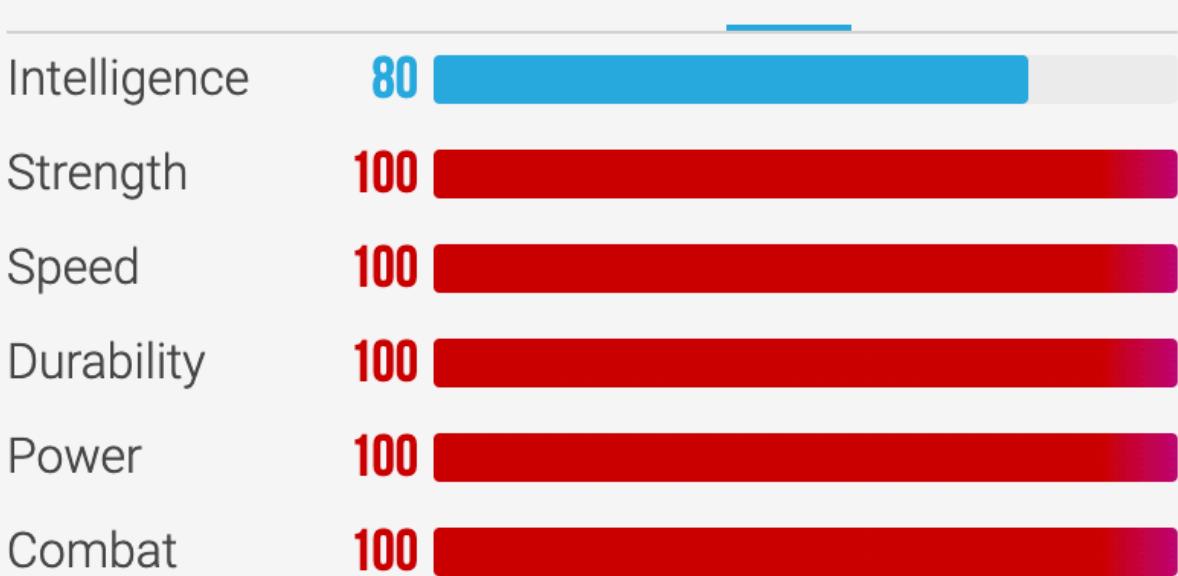
Full name	Thor Odinson
Alter Egos	Cosmic King Thor Fat Thor (MCU) Gladiator Thor (MCU) Golden Thor  Lord Thor Necro King Thor Old King Phoenix Thor Old King Thor Rune King Thor Thor (Mangaverse) Thor (MCU) Thor (Odin Force) Thor (Pre-Ragnarok) (MCU) Thor (Ultimate) Thor (Warrior Madness) Th Thorr  Unworthy Thor
Aliases	Donald Blake, Sigurd Jarlson, Jake Olsen, Donar the Mighty
Place of birth	Asgard
First appearance	Venus #11
Creator	Marvel Comics
Alignment	Good
Connections	
Occupation	King of Asgard; formerly EMS Technician; Physician



## History

Descended from Norse gods, Thor had many adventures in Earth's past. Around the year 1000 CE, he faced Hercules and time travelling Greek warriors, the vampire Varnae, the Annunaki deity Marduk, the troll Grylak the Greater, giant rats in an Atlantean outpost, visited the sons of one of the men killed by the rats, and battled Dromedan. Thor also lived as the humans Sigurd and Siegfried, slaying the... [read more >](#)

## Power Stats



Official Superhero Database stats.

## Top Battles

Superman Thor

Team Batman Team Iron Man

Wonder Woman Thor

## Appearance

Gender	Male
Type / Race	Asgardian
Height	6'6 • 198 cm
Weight	640 lb • 288 kg
Eye color	Blue
Hair color	Blond

## Collections

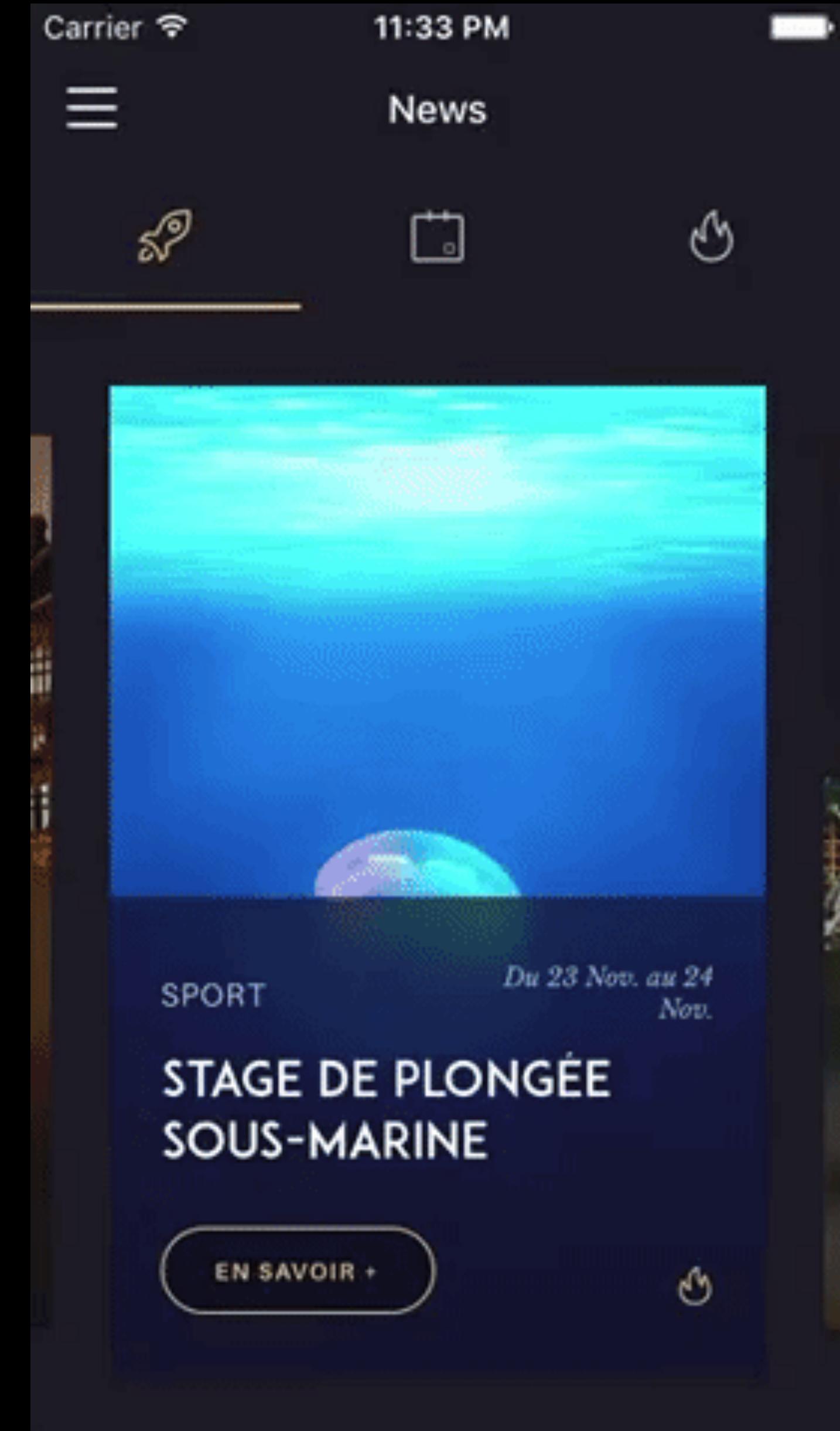
- 20 Most Powerful Marvel Characters
- Aquaman's Marvel Enemies Avengers Assemble
- Avengers : Earth's Mightiest Heroes
- Avengers : Earth's Mightiest Heroes
- Avengers : Earth's Mightiest Heroes
- Avengers Assemble (2013 - ) Avengers Best
- Avengers Nba Team Avengers Of All Time
- Avx Civil War Avengers Vs X-Men Civil War
- Batman Vs Thor Biggest Tempers
- Cosmic Level Dream Team
- Element Based Characters Elemental Characters
- Elemental Thunder/Lightning Characters
- Enemies Of The Mardc
- Every Superhero/Villain From Every Marvel Movie
- Favorite Characters (Blotsky)
- Favorite Characters (Mrjaeger07)
- Favorite Characters (Spectral\_supernova)
- Favorite Characters (Thor)



# Additional Exercise

Add Image snap carousel feature

<https://www.npmjs.com/package/react-native-snap-carousel>



# What's next?

- **React Navigation** ([https://reactnavigation.org/  
docs/intro/](https://reactnavigation.org/docs/intro/))



kurtesy

# Additional resources:

1. <https://reactnative.dev/docs/getting-started>
2. <https://reactnative.dev/docs/>
3. <https://oblador.github.io/react-native-vector-icons/>
4. <https://react-native-elements.github.io/react-native-elements/>
5. <https://github.com/xotahal/react-native-material-ui/blob/master/docs/Components.md>

