

Speak React Native

React Native course by **U+.**

Overview

- About us + you
- Course structure
- Mobile apps in general
- Native vs React Native
- Expo.io

About us + you



- Helps companies to launch ideas into reality
- Based in Prague, San Francisco, New York
- Python, React, React Native, PHP,...
- Clients: Škoda, Home Credit, Creative Dock, Ataccama, Zonky, HoppyGo

Jan

- Czech Technical University, FIT
- 4 years of Ionic, React Native in U+
- Traveller & honey maker
- <http://linkedin.com/in/janvaclavik>



Dan

- Czech Technical University, FEE
- Web apps, Ionic and React Native
- Loves bizzare movies & photography
- <http://linkedin.com/in/rysdaniel>



Martin

- Czech Technical University, FIT
- In U+ since 2015
- React Native programmer
- NHL 2017 progamer
- <http://linkedin.com/in/martin-doubek>



You

- Introduction
- Your experience
- Why are you here?

Questions for you

- Do you have experience with
 - HTML
 - React
 - Native app development
 - React Native
 - Nothing above

Course structure

Course structure

- 11 lectures
- ~4 homeworks
- Team project (start to think about the topic)
- Final presentation

What you will learn

- Build iOS/Android apps
- Javascript
- Reactive programming
- Git basics

What you will learn (in detail)

- React Navigation
- Styling
- State management with Redux
- Using maps
- Forms
- REST API, GraphQL,...

Mobile apps in general

Mobile app vs. Web?

Mobile app



Using native device features (TouchID, inApp Purchase,...)



Smooth UX



You have to download app from store



Need to spend time on platform differences

Web



Access content instantly



Platform independent



Easy to develop

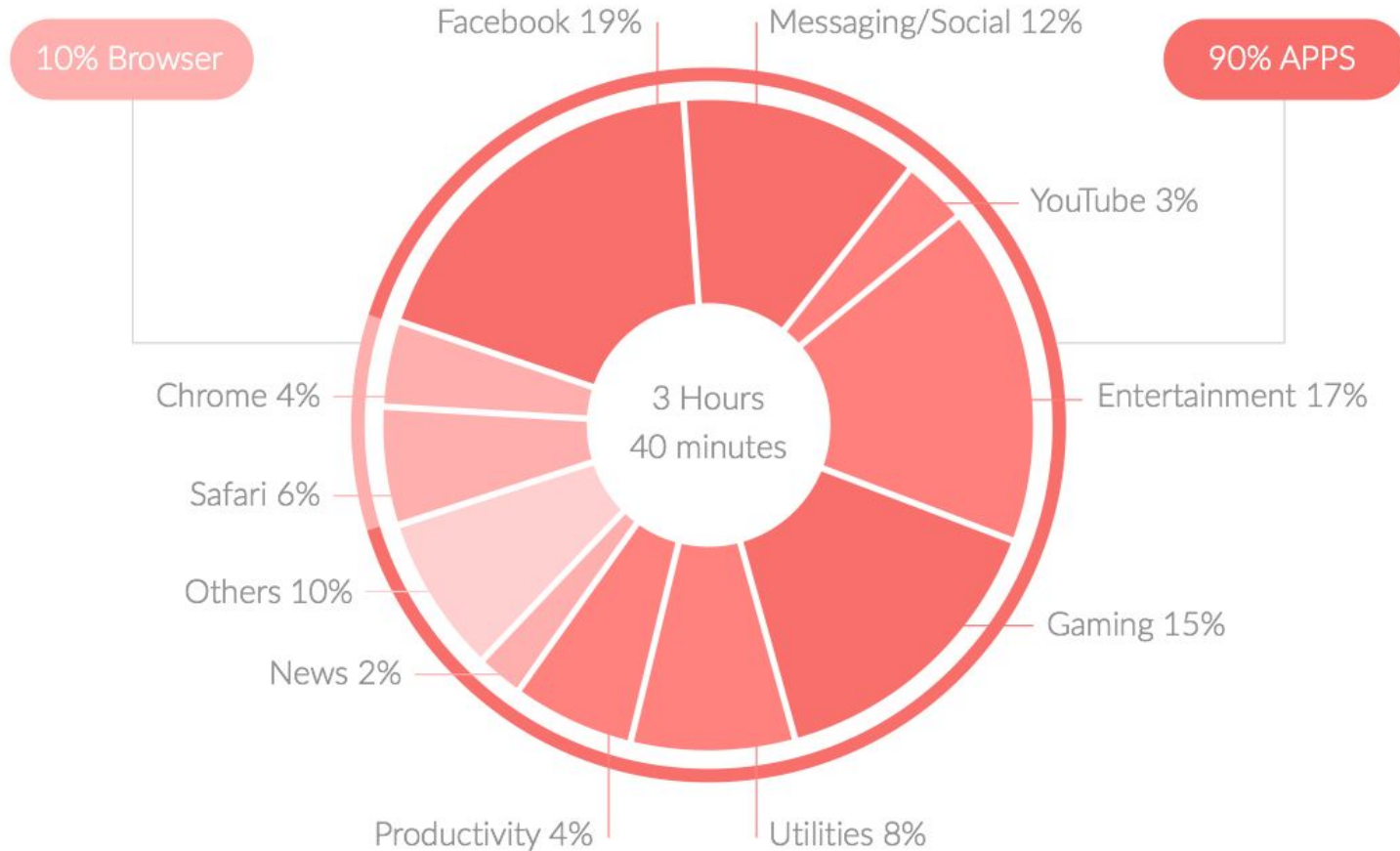


Poor performance



Worse access to native APIs

90% of Time on Mobile is Spent in Apps



Types of Mobile app development

- Native (Swift, Objective-C, Kotlin, Java)
- **React Native (Javascript)**
- NativeScript (Javascript)
- Cordova, Ionic (Javascript)
- Flutter (Dart)

Compare development types

Real-World Usage

Facebook



My First App

Accessing Native Device Features

Native-like

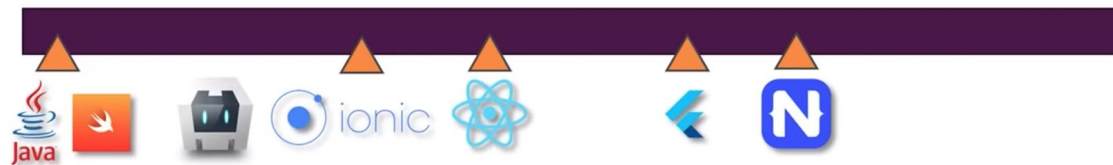


Wrapped-like

Compare development types

Ecosystem / Third-Party Libraries

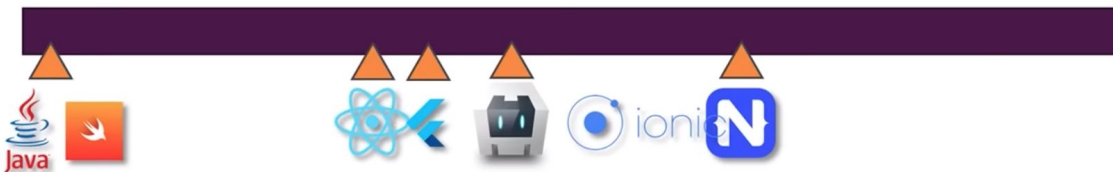
Plentiful



Scarce

Popularity / Coverage

Hot



Cold

Performance

Native-like



Wrapped-like

Native vs. React Native

What is React

- **Industry standard for building web apps**
- Javascript library for building User Interface
- Released 2013 by Facebook
- No MVC framework like Angular, Ember, etc.

What is React Native

- Used for building mobile apps
- React Native is 90 % similar to React
 - If you finish the course you will have experience for building web applications
- Open-sourced in 2015 by Facebook

Native app

- 👍 Native UI support
- 👍 Better optimization for advanced effects and performance
- 👍 Direct access to system APIs

- 🤡 Multiple codebases
- 🤡 Different teams and languages
- 🤡 Expensive

React Native app

- 👍 Native components
- 👍 Based on web technologies (JavaScript for both Android and iOS)
- 👍 One codebase (sharing more than 95 % of code)
- 👍 CSS-like styling with flexbox

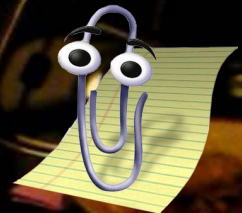
- 🤡 System API access relies on plugins
- 🤡 Not suitable for all types of applications

A man with glasses and a surprised expression is sitting at a desk. Behind him is a large monitor displaying a grid of many small app icons, resembling an Android home screen. A white thought bubble with a black outline is positioned in front of him, containing the text "BUT SUCCESSFUL APPS ARE NATIVE" in pink, all-caps, sans-serif font. The background is dark, and the room is dimly lit, with the primary light source being the monitor.

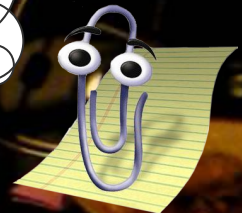
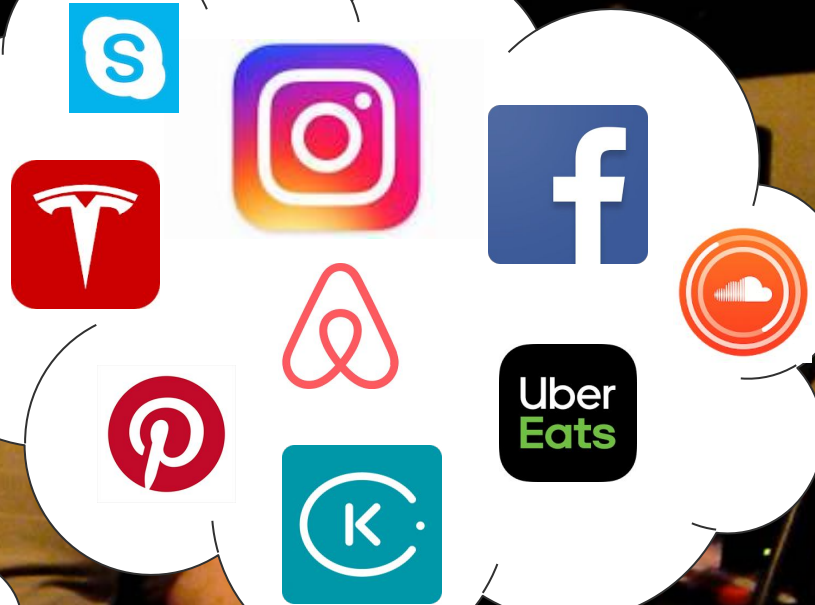
BUT SUCCESSFUL
APPS ARE NATIVE



BUT SUCCESSFUL
APPS ARE NATIVE



BUT SUCCESSFUL
APPS ARE NATIVE



Javascript intro

IDE

- “Text editor for development”
- Code formatting, syntax highlight, autocomplete
- Plugins, integrations (git, colors,...)
- Quick file navigation
- [VSCode](#), Atom, WebStorm,...

Yarn

- Package manager (mainly for Javascript)
- Script runner
- Just like [NPM](#), but better

JS basics

- Constants and variables
- Functions
- Classes
- Imports and exports
- Logical & Conditional statements

Constants and Variables

```
const iAmConstant = "I am a constant value"  
iAmConstant = "Wow, such constant" // Bad idea. You can't  
redefine constant. Use "let"
```

```
let iAmVariable = "I am variable"  
iAmVariable = "Redefined value"
```


Functions

```
const sum1 = (a, b) => {  
  return a + b  
}
```

// Call function

sum1(1, 2) // returns 3

```
class MathOperations {  
  sum3 = () => (  
    a + b  
  )  
  sum4 () {  
    return a + b  
  }  
}
```

Imports and exports

```
// external module - default export
```

```
export default React
```

```
// external module - named export
```

```
export const Text = { /* something */ }
```

```
// our module - default export
```

```
export default MyComponent
```

```
// external module - default import
```

```
import React from "react"
```

```
// external module - named import
```

```
import { Text } from "react-native"
```

```
// our module - default import
```

```
import MyComponent from "./MyComponent"
```

Logical and conditional statements

```
// Conditional statement
```

```
if (exists && !hasError) {  
    color = 'green'  
} else {  
    color = 'red'  
}
```

```
// Ternary operator
```

```
const color = hasError ? 'red' : 'green'  
const color = !exists || hasError ? 'red' : 'green'
```

Class

```
import { Component } from "react"
import { View, Text } from "react-native"
```

```
class WelcomeText extends Component {
  render () {
    return (
      <View>
        <Text>Let's speak React Native</Text>
      </View>
    )
  }
}
```

```
// Render (show) component
<WelcomeText />
```

Class

```
import { Component } from "react"
import { View, Text } from "react-native"

class WelcomeText extends Component {
  render () {
    return (
      <View>
        <Text>Let's speak React Native</Text>
      </View>
    )
  }
}
```

JSX

JSX (JavaScript XML)

- Looks like HTML
- Syntactic sugar (not a markup language, not a templating engine)
- Makes code more readable

JSX vs. JS

```
class WelcomeText extends Component {  
  render () {  
    return (  
      <View>  
        <Text>Let's React</Text>  
      </View>  
    )  
  }  
}
```



BABEL

```
_createClass(WelcomeText, [{  
  key: "render",  
  value: function render() {  
    return React.createElement(  
      _reactNative.View,  
      null,  
      React.createElement(  
        _reactNative.Text,  
        null,  
        "Let's React"  
      )  
    );  
  }  
}]);
```

Expo.io

Expo.io



- Zero configuration tool for building React Native apps
- Minimize the need to use Android Studio (~~💩~~) or XCode (~~💩~~) or install native plugins (~~💩~~)
- Makes using of system APIs easier

Installation

Get NodeJS (<http://nodejs.org>)

```
$ npm install -g expo-cli
```

Create your app

<code>\$ expo init hello-world</code>	<code># create app</code>
<code>\$ cd hello-world</code>	<code># enter to directory</code>
<code>\$ yarn</code>	<code># install dependencies</code>
<code>\$ expo start</code>	<code># start expo server</code>

Test on device

(Your notebook and device have to be on the same network)

1. Download Expo client: <https://expo.io/tools#client>
2. Use your mobile device to scan QR code

Hello world!

- Let's code!

More complex example

- Add MapView
 - <https://docs.expo.io/versions/v30.0.0/sdk/map-view>
- Pin marker to “Paralelní Polis”
 - <https://github.com/react-community/react-native-maps/blob/master/docs/marker.md>

Expo docs

- <https://docs.expo.io/>

Add MapView

```
import { MapView } from 'expo';

export default class App extends React.Component {
  render() {
    return (
      <MapView
        style={{ flex: 1 }}
        initialRegion={{
          latitude: 50.1033533,
          longitude: 14.4504872,
          latitudeDelta: 0.09,
          longitudeDelta: 0.04,
        }}
      />
    )
  }
}
```


Place marker

```
<MapView
  ...
>
  <MapView.Marker coordinate={{
    latitude: 50.1033533,
    longitude: 14.4504872
  }}
/>
</MapView>
```

Homework

- Install [Git](#)
- For Mac users:
 - [Install XCode](#)
- For Windows/Linux users:
 - [Install Android SDK](#) + [Android emulator](#)

Questions?



Sources

- <https://ipure.cz/archiv/magazin/ne-nepotrebuje-aplikaci/>
- <https://rubygarage.org/blog/mobile-app-vs-mobile-website>
- <https://www.youtube.com/watch?v=bnYJRYFsrSw>