Full-stack Application Development

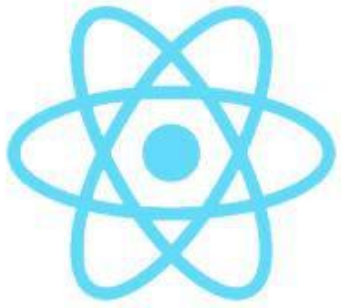
Introduction to React Native

Where to Find The Code and Materials?

https://github.com/iproduct/react-native-training

Why React Native?

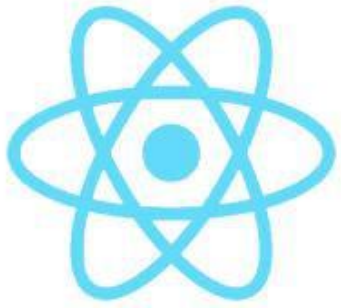
- Create native apps for Android and iOS using React
- React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces.
- You can add some RN components in your existing Android or iOS project, or can create a completely new app from scratch.
- Written in JavaScript - rendered with native code - your app uses the same native platform APIs, as native applications do.



React Native

Why React Native? - II

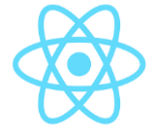
- Many platforms, one React. Create platform-specific versions of components so a single codebase can share code across platforms.
- React Native provides a core set of platform agnostic native components like View, Text, and Image that map directly to the platform's native UI building blocks.
- Fast page refresh – you can see your changes as soon as you save - no more waiting for native builds to finish. Save, see, repeat.



React Native

History and Perspectives

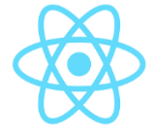
- Facebook released React Native in 2015 and has been maintaining it ever since.
- In 2018, React Native had the 2nd highest number of contributors for any repository in GitHub.
- Today, React Native is supported by contributions from individuals and companies around the world including Callstack, Expo, Infinite Red, Microsoft and Software Mansion.
- React Native community is always shipping exciting new projects and exploring platforms beyond Android and iOS with repos like React Native Windows, React Native macOS and React Native Web.
- Getting started: https://reactnative.dev/docs/getting-started



Setting Development Environment

- Using Expo CLI - Expo is a set of tools built around React Native and, while it has many useful features, including ability to start a React Native app in minutes. A recent version of Node.js and a phone or emulator are needed too.
- You can try Snack if you'd like to try out React Native directly in your web browser before installing any tools.
- Using React Native CLI - requires Xcode or Android Studio to get started. If you already have one of these tools installed, you should be able to get up and running within a few minutes. If they are not installed, you should expect to spend about an hour installing and configuring them.

- Getting started: https://reactnative.dev/docs/environment-setup



Expo CLI Setup

- npm install -g expo-cli / yarn global add expo-cli
- expo init MyProject
- cd MyProject
- yarn start # you can also use: expo start

OR

- yarn start --web # you can also use: expo start --web – for web client starting
- yarn add react-native-web – to add react-native-web dependencies needed for runnig RN web client
- Install Expo Go app for iOS and Android (from iOS App Store or Android Play Store)

Hello React Native Example App - Class

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

class HelloWorldClass extends Component {
    render() {
      return (
        <View style={{
            flex: 1,
            justifyContent: "center",
            alignItems: "center"
          }}>
          <Text>Hello, world Class Component!</Text>
        </View>
      );
    }
  }

  export default HelloWorldClass;
```

Source: https://reactnative.dev/docs/getting-started

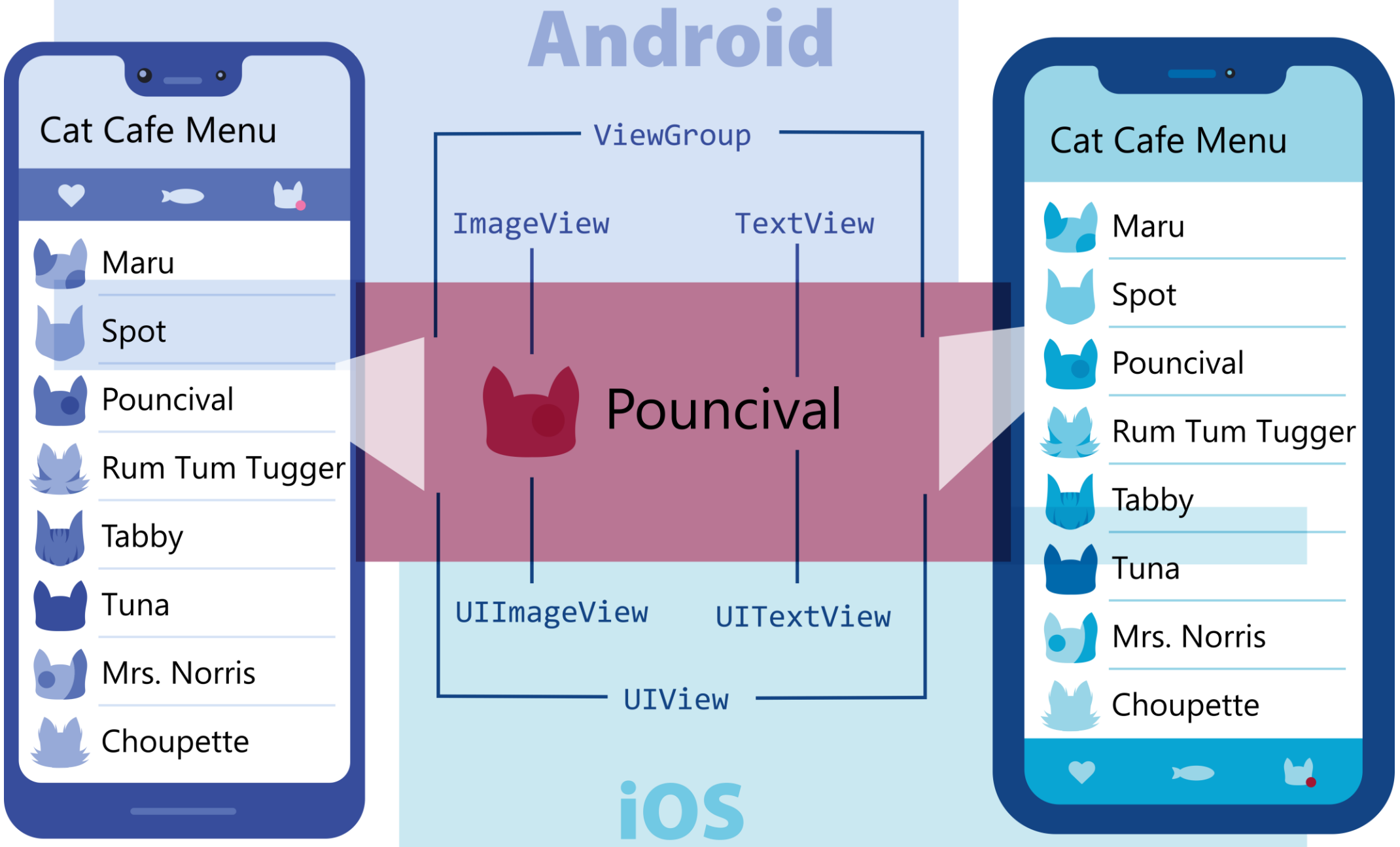
Hello React Native Example App - Function

```
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return (
    <View style={styles.container}>
      <Text>Hello from React Native!</Text>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
    fontSize: '48px',
  },
});
```

Source: https://reactnative.dev/docs/getting-started



Source: https://reactnative.dev/docs/intro-react-native-components

Core Components

React Native UI Component	Android View	iOS View	Web Analog	Description
<View>	<ViewGroup>	<UIView>	A non-scrolling	A container that supports layout with flexbox, style, some touch handling, and accessibility controls
<Text>	<TextView>	<UITextView>	<p>	Displays, styles, and nests strings of text and even handles touch events
<Image>	<ImageView>	<UIImageView>	<img>	Displays different types of images
<ScrollView>	<ScrollView>	<UIScrollView>		A generic scrolling container that can contain multiple components and views
<TextInput>	<EditText>	<UITextField>	<input type="text">	Allows the user to enter text

Source: https://github.com/facebook/react-native-website/blob/main/docs/intro-react-native-components.md

React Native Core Components Demo

```
import React from 'react';
import { View, Text, Image, ScrollView, TextInput } from 'react-native';
const App = () => {
  return (
    <ScrollView>
      <Text>Some text</Text>
      <View>
        <Text>Some more text</Text>
        <Image
          source={{
            uri: 'https://reactnative.dev/docs/assets/p_cat2.png',
          }}
          style={{ width: 200, height: 200 }}
        />
      </View>
      <TextInput
        style={{
          height: 40,
          borderColor: 'gray',
          borderWidth: 1
        }}
        defaultValue="You can type in me"
      />
    </ScrollView>
  );
}
export default App;
```

Thank's for Your Attention!

Trayan Iliev

IPT – Intellectual Products & Technologies

http://iproduct.org/

http://robolearn.org/

https://github.com/iproduct

https://twitter.com/trayaniliev

https://www.facebook.com/IPT.EACAD