React Native 1

CS571: Building User Interfaces

Cole Nelson

Download Expo

- Download for iOS
- Download for Android
- Don't have a smart phone? You can use an emulator like AVD or XCode

Mid-Semester Feedback

40% response rate! 🥳 Can we get 60% for the final evaluation? Takeaways...

1. **Go slower.**
2. More time for examples.
3. Get lecture notes out earlier -- I'll try!
4. More seats -> two sections of CS571 in F23

Midterm Exam

1. Great job!
2. Scores will be released in the next two weeks.
3. Final Exam will be 40 multiple-choice worth 20 pts over 90 minutes with a double-sided notesheet.
4. Free bonus +0.5 points for rounding error.

Midterm Exam Review

See Canvas

What will we learn today?

- What is mobile development?
- How does React Native fit into mobile development?
- Programming w/ React Native & Expo

Mobile Development

Native Development and its Alternatives

What is "True Native" Development?

Building specifically for the device (e.g. Android or iOS) that you want to support.

iOS: Objective-C or Swift w/ Cocoapods
Android: Java or Kotlin w/ Maven or Gradle

Pros and Cons of True Native

Pros

- Organic User Experience
- Optimized Apps
- Fine-Grained Control

Cons

- Expensive
- Little Code Reuse
- Less Sense of Abstraction

Alternatives to True Native

No mobile app! Do we really need an app? Could a responsive webpage be just as effective?

WebView! Can we take our existing code and just slap it into a WebView? e.g. Apache Cordova

Cross-Platform! Can we use a library or framework that will make our code work natively on Android *and* iOS? e.g. React Native

Who is using React Native?

- Facebook
- Microsoft
- Shopify
- Coinbase
- Discord
- Dave

... among many others. Other companies may be doing pure-native or hybrid development.

What is React Native?

A JS framework for building native, cross-platform mobile applications using React, developed by Facebook in 2015.

Unlike ReactJS, which was a library, React Native is a framework that includes everything* that we will need to build mobile applications.

React Native supports iOS and Android development.

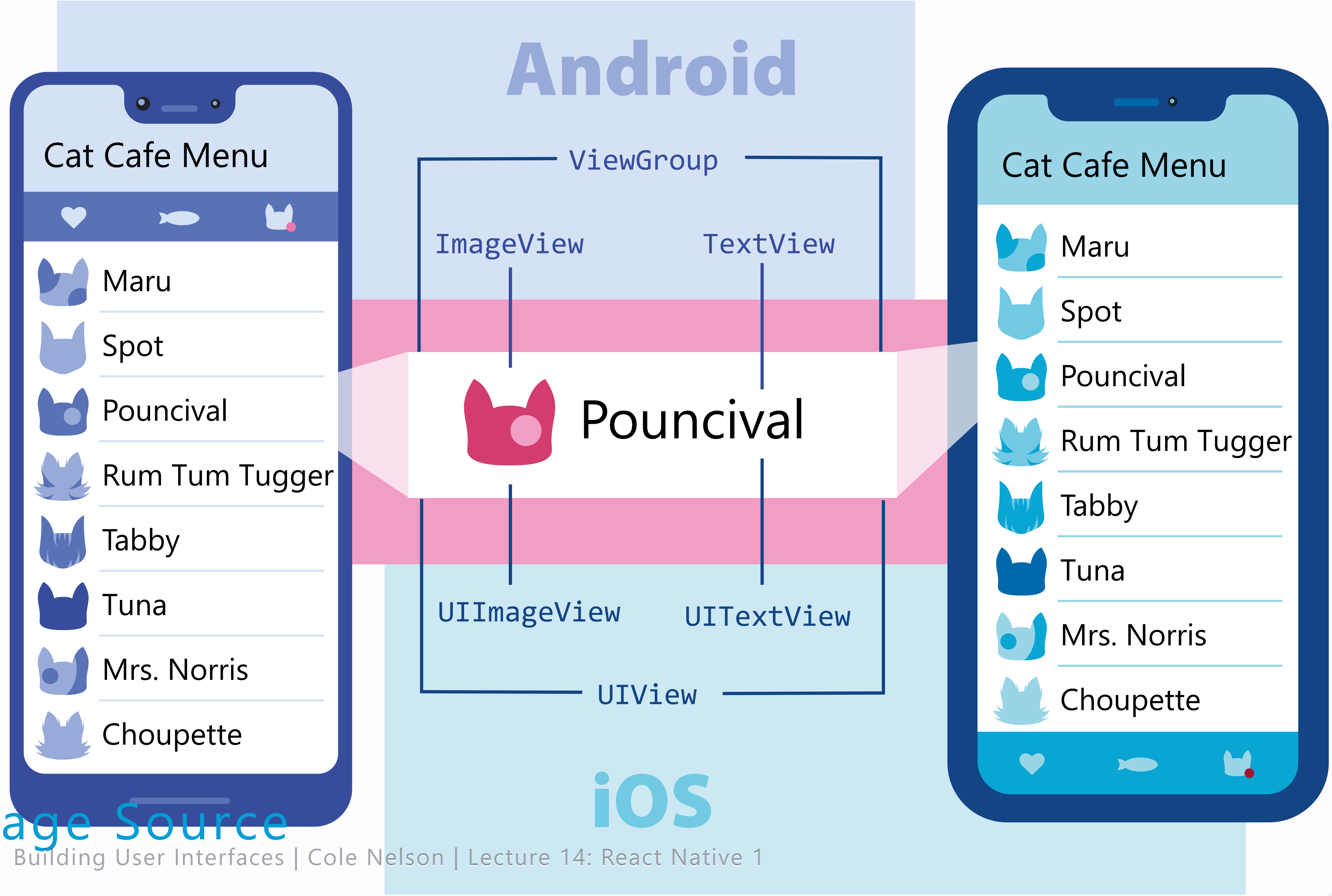
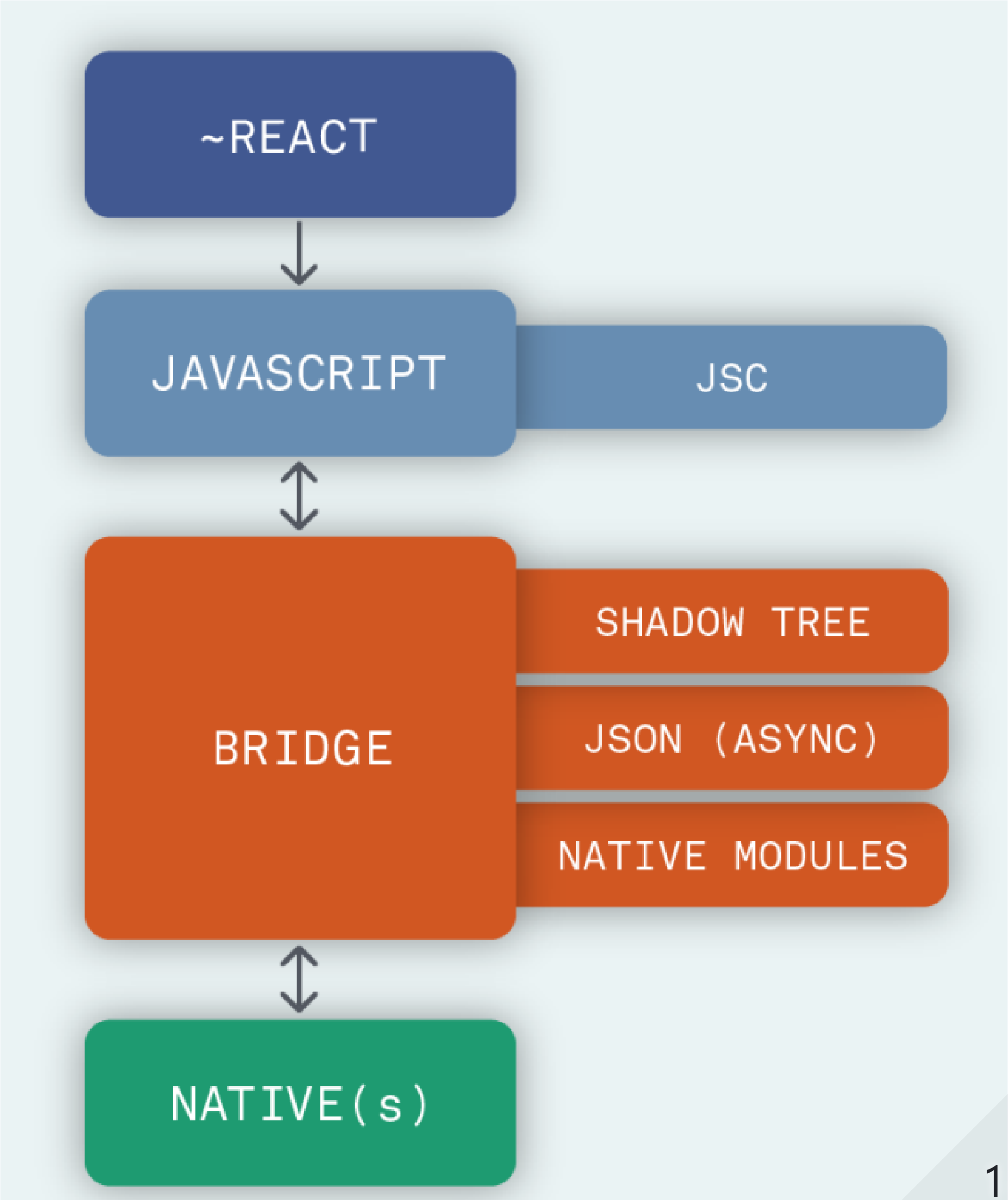


Image Source

React Native

- No more DOM or browser capabilities!
- Connects with native components using a "bridge"

Image Source



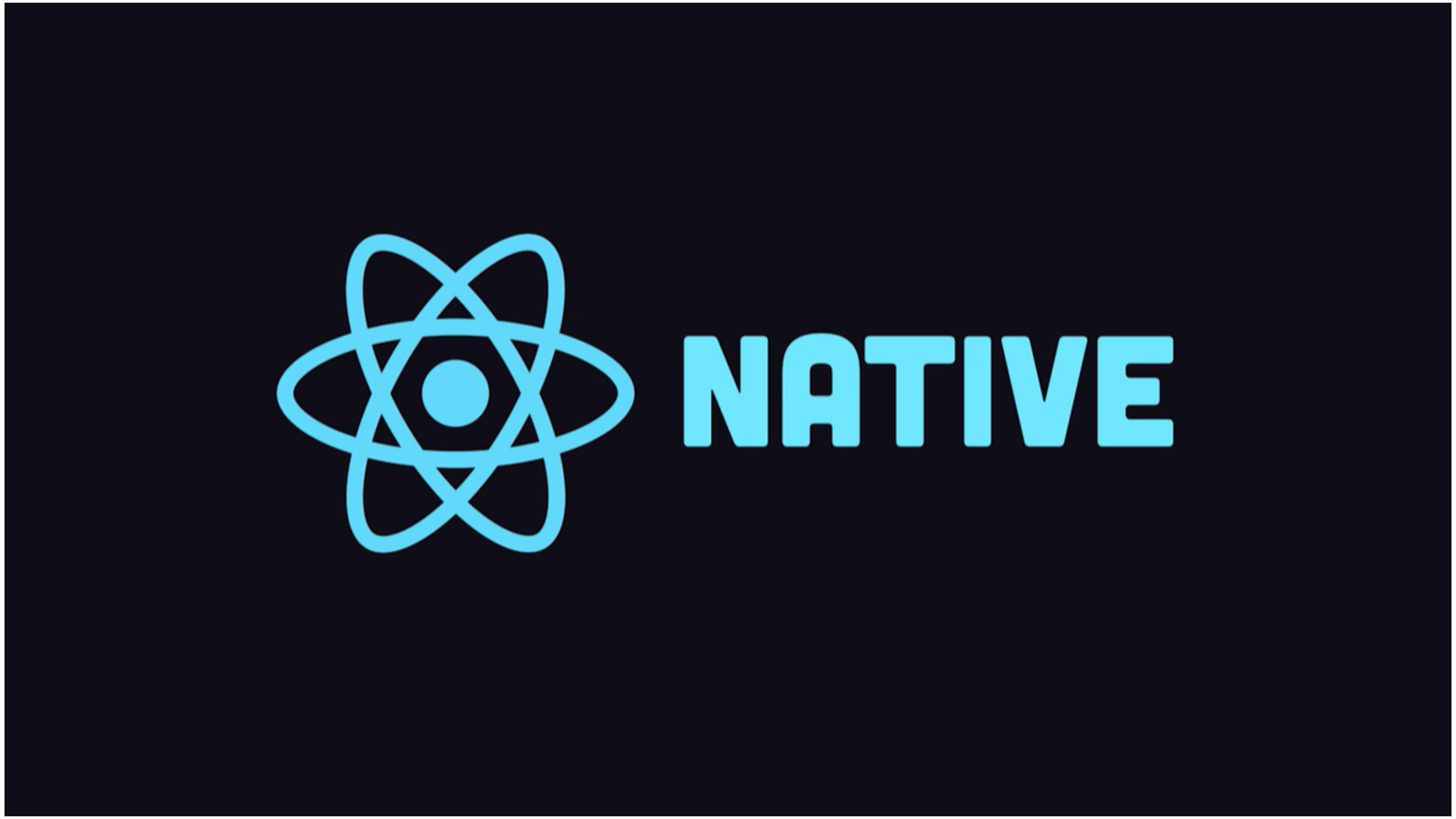
React Native

- The use of a bridge causes a slight hit to performance.
- Will soon (is already?) be remedied with "The New Architecture" and Hermes!

Image Source

React Native

React for Mobile Devices!



React Native in 100 seconds

A Review of *Implementation* So Far

Lecture	Takeaway
Intro	The web runs on HTTP
JS1	Basics of HTML, CSS, and JS.
JS2	APIs and Declarative Programming
JS3	Data Copying and Bootstrap

A Review of *Implementation* So Far

Lecture	Takeaway
React 1	Intro, useState, and useEffect
React 2	Review, Uncontrolled vs Controlled, NPM
React 3	State Management, Context, and Routing
React 4	Complex APIs and Secret Management
React 5	Memoization and Deployment

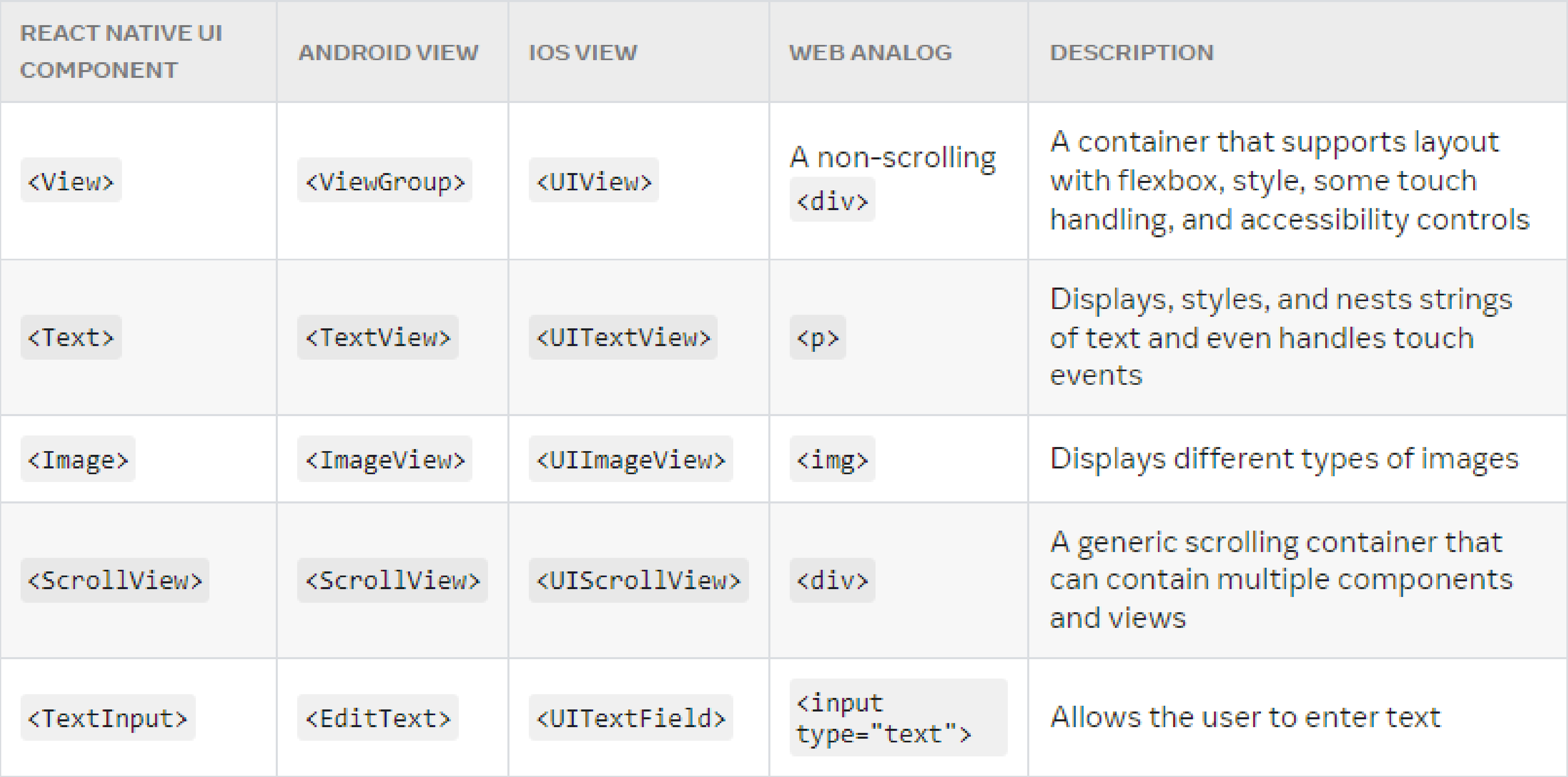
What stays the same?

- Using NPM for our library management
- Using complex APIs
- Core React features
  - React Hooks (useEffect, useState, etc.)
  - Passing props and state management
  - Controlled vs Uncontrolled Inputs
  - Memoization

What changes?

- This isn't a browser!
  - No more DOM!
  - No more CSS!
    - No more Bootstrap!
  - No more sessionStorage, localStorage, or cookies.
- Wider variety of inputs
  - Sensors
  - Gestures
- React Navigation vs React Router

Conversions to Know



REACT NATIVE UI COMPONENT	ANDROID VIEW	IOS VIEW	WEB ANALOG	DESCRIPTION
`<View>`	`<ViewGroup>`	`<UIView>`	A non-scrolling `<div>`	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>`	`<div>`	A generic scrolling container that can contain multiple components and views
`<TextInput>`	`<EditText>`	`<UITextField>`	`<input type="text">`	Allows the user to enter text

Image Source

Other Good Questions to Ask…

- Can we declaratively program using RN? **YES**
- Can we use JSX with RN? **YES**
- Can we use React hooks in RN? **YES**
- Can we do styling in RN? **YES**-ish
- Is it *truly* cross-platform? **MAYBE**-ish

Hello World!

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

function MyApp() {
  return (
    <View style={{ flex: 1, justifyContent: "center", alignItems: "center" }}>
      <Text>
        Try editing me! 🎉
      </Text>
    </View>
  );
}

export default MyApp;
```

Expo Snack

React Native for React Devs

How can we write our mobile apps with React Native?

Getting Started

Using Expo, similar to create-react-app!

Run one-time...

```
npm install -g create-expo-app
```

Run for each project...

```
create-expo-app my-new-project
cd my-new-project
npm start
```

Getting Started: A Special Note

By default, expo uses "lan" to host your app. This may cause issues on certain networks. Try using "localhost" or "tunnel" by modifying scripts of `package.json`...

```json
"scripts": {
  "start": "expo start --localhost",
  "android": "expo start --android",
  "ios": "expo start --ios",
  "web": "expo start --web"
}
```

You will need to be wired in to your computer!

Expo Demo

Tackling HW7...

What did we learn today?

- What is mobile development?
- How does React Native fit into mobile development?
- Programming w/ React Native & Expo

Questions?