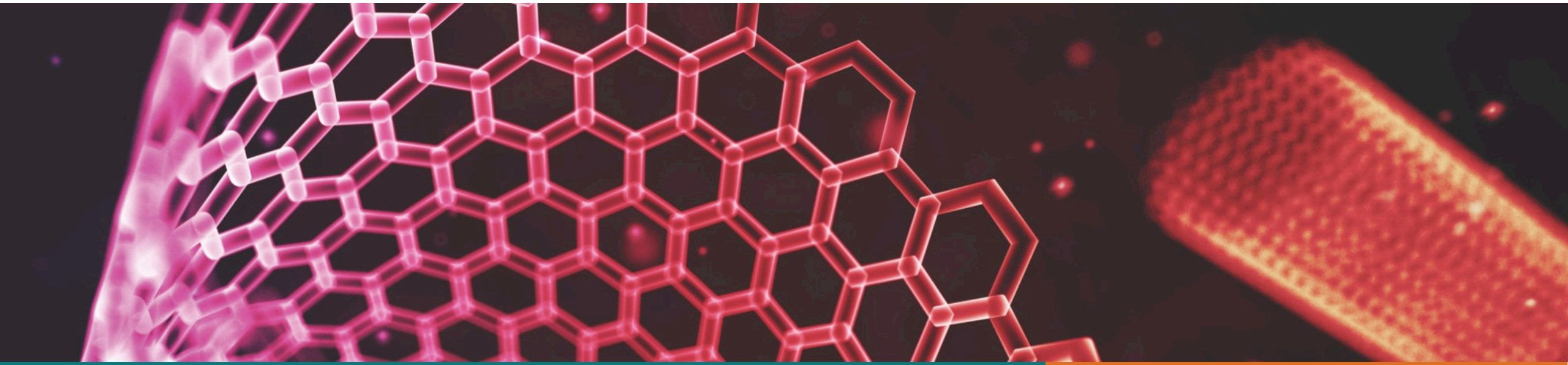


CS 554 – Web Programming II

React Native





What is React Native?

It's an open source cross platform mobile application framework by Facebook Inc. React Native subscribes to the philosophy of **"Write once and run anywhere"**. In essence, we are just writing JS & React components which talks to native platform api. Unlike most cross platform framework, React native doesn't use WebView but it rather negotiates with Native API



How React Native came into existence?

- Facebook in it's early days relied HTML5 for their mobile application
- HTML5 as a mobile application wasn't really a good idea though
- "The biggest mistake we made as a company was betting too much on [HTML](#) as opposed to native" ~ Mark Zuckerberg
- Jordan Walke from facebook found a way to generate Ui elements on ios from a background JavaScript thread
- Couple of hackathons later, React Native was born



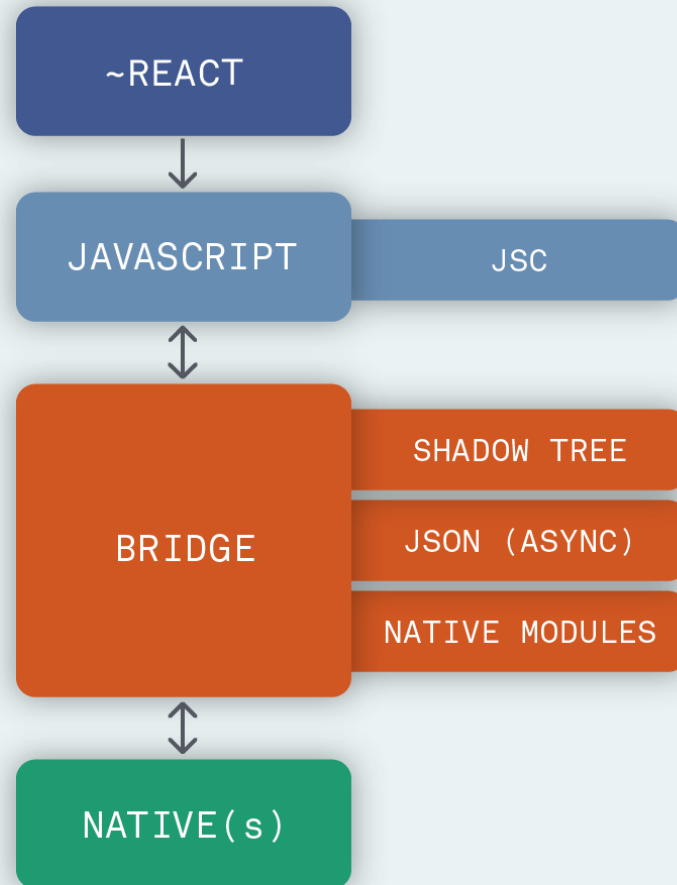
React Native out in the wild!

Thousands of apps are already using React Native.

- Some of them are
 - Facebook Application
 - Tesla
 - Facebook Ads manager App
 - Facebook Analytics
 - Instagram
 - Oculus
 - Coinbase
 - Discord
 - Skype
 - Uber Eats
 - Walmart
 - Salesforce
 - Bloomberg



What is React Native?





Similarity with React

Here are few vital similarity to keep in mind before diving into React Native

- Components can be written in both Class based & Function based paradigms.
- All the default React hooks are applicable in React Native (e.g. useEffect, useState, etc)
- Syntax remains the same with some minor nuances (mostly semantic nuances – which we will talk about in a while)
- JS libraries can be imported (e.g. Axios)
- Uses JSX for UI components



Differences from React

- Doesn't use HTML or CSS
- Doesn't manipulate Virtual DOM since there is no WebView being rendered whatsoever.



Benefits of React Native

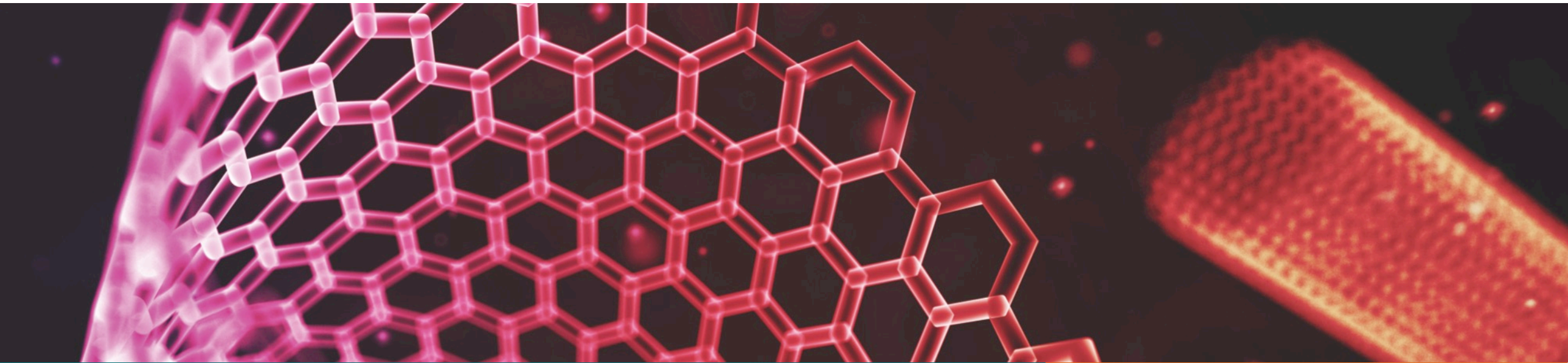
- ***Native like performance***—React Native ultimately gets translated into native code, therefore it is possible to achieve 60fps animations.
- ***Narrow Learning Curve***—If you have any experience with React & JS then it is very easy to get into React Native.
- ***Less code to write***—We can share component or code logic between IOS & Android applications. React Native has a very active community development going on. On top of that, We get the convenience of using thousands of javascript library / snippet right out of the box.



Disadvantages of React Native

- ***Not so Native*** —Although react native is fast but it is no where comparable to the real native performance provided by Kotlin or Swift
- ***Feature Coverage***—Not every feature is covered by React Native. You might find yourself writing native code in order to access specific features.
- ***Native Developers in reserve***—It may sound tempting to only use React-Native for your next mobile application. But as it turns out, You may need Native Developers to build a huge application. Maintaining bridging framework is really challenging. This is one of the main reasons why AirBnB switched back to native.

React Native Core Components

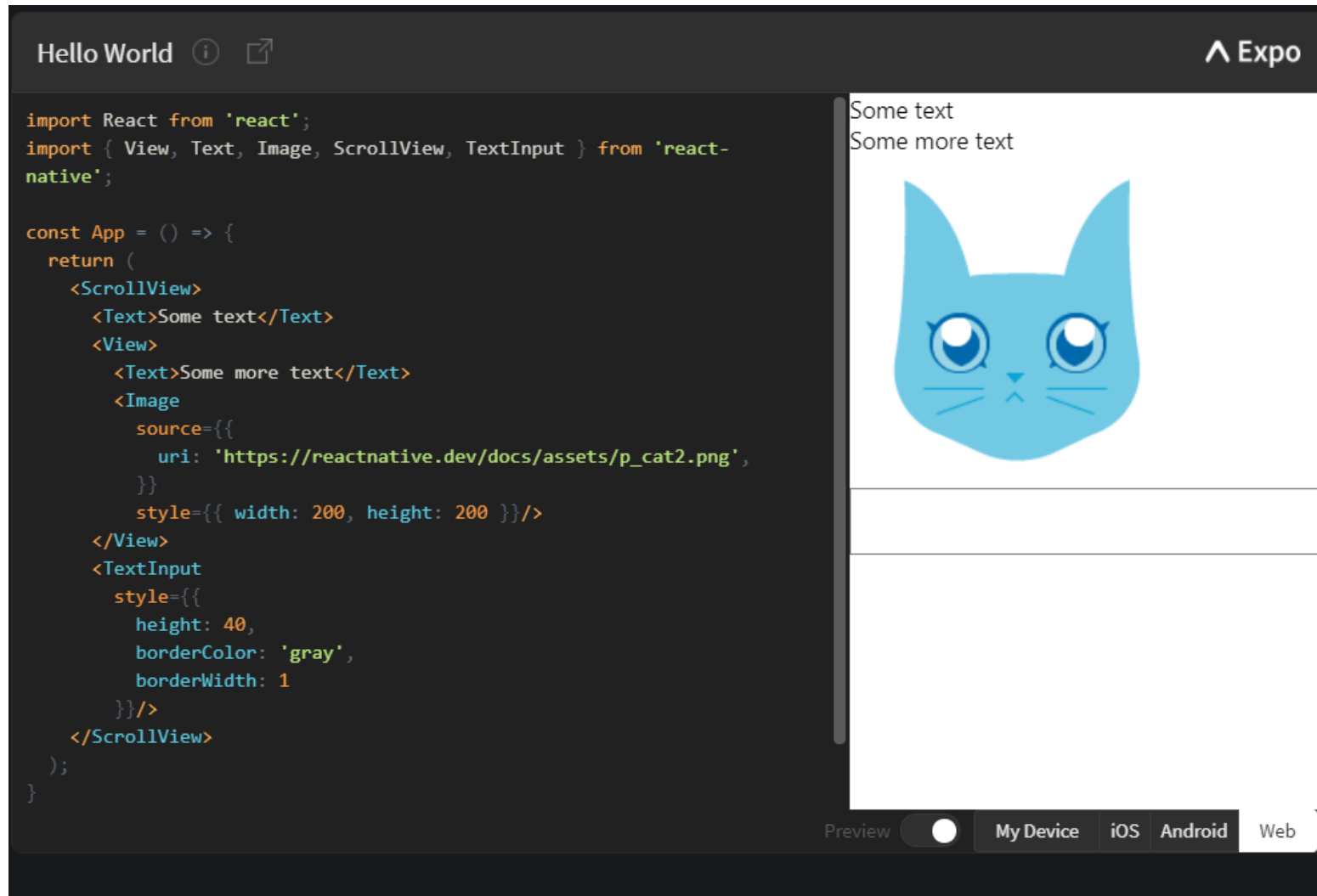




React Native : Core components

REACT NATIVE UI COMPONENT	ANDROID VIEW	IOS VIEW	WEB ANALOG	DESCRIPTION
<code><View></code>	<code><ViewGroup></code>	<code><UIView></code>	A non-scrolling <code><div></code>	A container that supports layout with flexbox, style, some touch handling, and accessibility controls
<code><Text></code>	<code><TextView></code>	<code><UITextView></code>	<code><p></code>	Displays, styles, and nests strings of text and even handles touch events
<code><Image></code>	<code><ImageView></code>	<code><UIImageView></code>	<code></code>	Displays different types of images
<code><ScrollView></code>	<code><ScrollView></code>	<code><UIScrollView></code>	<code><div></code>	A generic scrolling container that can contain multiple components and views
<code><TextInput></code>	<code><EditText></code>	<code><UITextField></code>	<code><input type="text"></code>	Allows the user to enter text

React Native : Core components





React Native : View, Text

- View component is essentially a normal **<div>** tag.
 - Every View tag has position set to relative
 - We can use Flex property to make it work like a flex box or flex item
- Text is a **<p>** tag
 - CSS styling such as **fontSize, fontWeight, textAlign, letterSpacing**, etc are relevant

```
<View style={{marginTop: 15}}>  
  <Text style={{fontWeight: 300, fontSize: 16}}> Some Text </Text>  
</View>
```

Some Text



React Native : Image, Scroll View

- Scroll View component just enables scrolling to your current screen.
- Image component is used to import image
 - Some important props are
 - **resizeMethod** – **cover**, **contain**, **stretch**, **repeat**, **center**
 - **Source** – used to define the source of the image.

React Native : Image, Scroll View



Function Component Example ⓘ ↗

^ Expo

```
width: 66,
height: 58,
},
});

const DisplayAnImage = () => {
  const img = require('@expo/snack-static/react-native-logo.png')
  return (
    <ScrollView style={{ flex: 1, alignItems: 'center' }}>
      <Image
        style={{ width: 50, height: 50, marginBottom: 10 }}
        source={img}
      />
      <Image
        style={styles.tinyLogo}
        source={{
          uri: 'https://reactnative.dev/img/tiny_logo.png',
        }}
      />
    </ScrollView>
  );
}

export default DisplayAnImage;
```





React Native : Text Input

- Text Input is a replacement for `<input type='text' />` element.
- To handle text change we have to use the Prop - **onChangeText**

React Native : Text Input

Handling Text Input ⓘ ↗


^ Expo

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

const PizzaTranslator = () => {
  const [text, setText] = useState('');
  return (
    <View style={{padding: 10}}>
      <TextInput
        style={{height: 40}}
        placeholder="Type here to translate!"
        onChangeText={text => setText(text)}
        defaultValue={text}
      />
      <Text style={{padding: 10, fontSize: 42}}>
        {text.split(' ').map((word) => word && '🍕').join(' ')}
      </Text>
    </View>
  );
};

export default PizzaTranslator;
```

1 2 3 4 5 6 7 8



Preview

☐

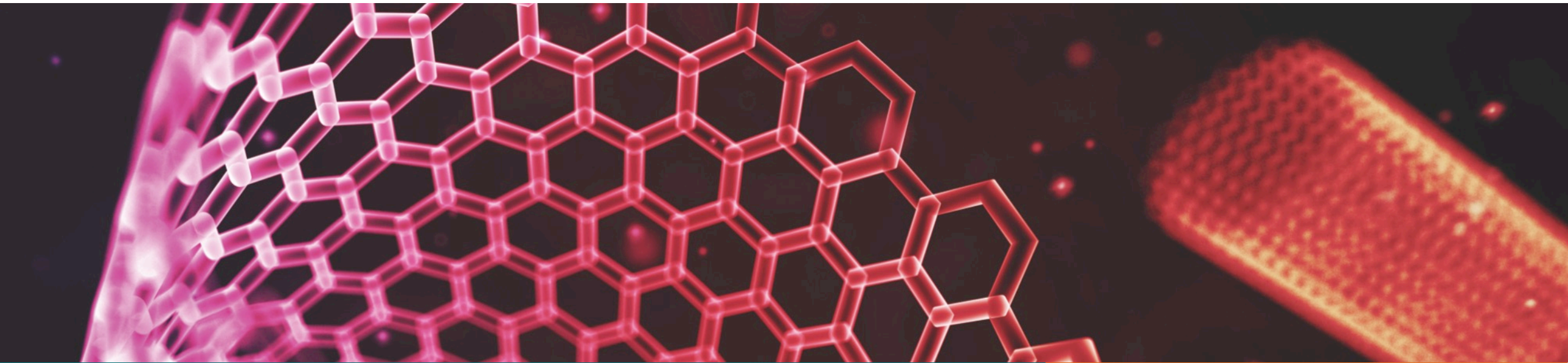
My Device

iOS

Android

Web

React Native Essential Components





React Native : Essential Component

Before we jump into our code, we need to know the following essential concepts

- **Stylesheet**
- **Platform**
- **Networking**
- **Flatlist**



React Native : Stylesheet

- It is used to generate Style in react native

```
StyleSheet ⓘ ↗ Expo

import React from "react";
import { StyleSheet, Text, View } from "react-native";

const App = () => (
  <View style={styles.container}>
    <Text >React Native</Text>
  </View>
);

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 24,
    backgroundColor: "#eaeaea",
    borderWidth: 5,
    borderColor: 'red'
  }
});

export default App;
```

React Native



React Native : Platform

- Platform module let's us divide our code base based on the mobile platform
- We can identify the mobile platform in two ways
 - By using Platform module
 - Platform specific file extensions



React Native : Platform Module

- Platform Module provides us with the following properties
 - Platform.OS
 - Platform.select
 - Platform.Version

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

const styles = StyleSheet.create({
  height: Platform.OS === 'ios' ? 200 : 100
});
```

```
const Component = Platform.select({
  ios: () => require('ComponentIOS'),
  android: () => require('ComponentAndroid')
})();

<Component />;
```



React Native : Platform Module

- You may find yourself in the position where a particular feature isn't supported by the current version of the mobile platform.
- In that case we can use Platform.version to detect the version of the current platform

```
import { Platform } from 'react-native';

if (Platform.Version === 25) {
  console.log('Running on Nougat!');
}
```

```
import { Platform } from 'react-native';

const majorVersionIOS = parseInt(Platform.Version, 10);
if (majorVersionIOS <= 9) {
  console.log('Work around a change in behavior');
}
```



React Native : Platform Specific file ext.

- Another way to write platform specific code is using file extensions
- All we must do is add `.android.js` or `.ios.js` to the file and react native bundler will choose appropriate modules while building the app for respective platform

```
JS Navigation.android.js
```

```
JS Navigation.ios.js
```

```
import Nav from './components/Navigation'

const App = () => {
  return (
    <NavigationContainer>
      <Nav/>
    </NavigationContainer>
  );
};
```




React Native : Networking

- Although React native provides a fetch method for all the networking needs.
- We can still use axios since JS will run on a background thread on your smartphone

```
utils > JS queries.js > ...
1  import axios from 'axios'
2
3  const client = axios.create()
4  const host = `https://api.tvmaze.com`
5
6  const grabTvShows = () =>{
7    |   return client.get(`${host}/shows`)
8  }
9
```



React Native : Flatlist

- Flatlist is essentially a performant scrollable list component
- It will only load item when a particular item is visible on the screen. Hence you might end up saving whole lot of network calls & rendering.

```
const Item = ({ title }) => (  
  <View style={styles.item}>  
    <Text style={styles.title}>{title}</Text>  
  </View>  
);  
  
const App = () => {  
  const renderItem = ({ item }) => (  
    <Item title={item.title} />  
  );  
  
  return (  
    <SafeAreaView style={styles.container}>  
      <FlatList  
        data={DATA}  
        renderItem={renderItem}  
        keyExtractor={item => item.id}  
      />  
    </SafeAreaView>  
  );  
}
```

First Item

Second Item

Third Item



Further Reading

- [Async Storage Module](#)
 - Provides idiomatic api to handle native storage
- [React Navigation](#)
 - Provides a router like feature to your mobile applications.
- UI library

Questions?

