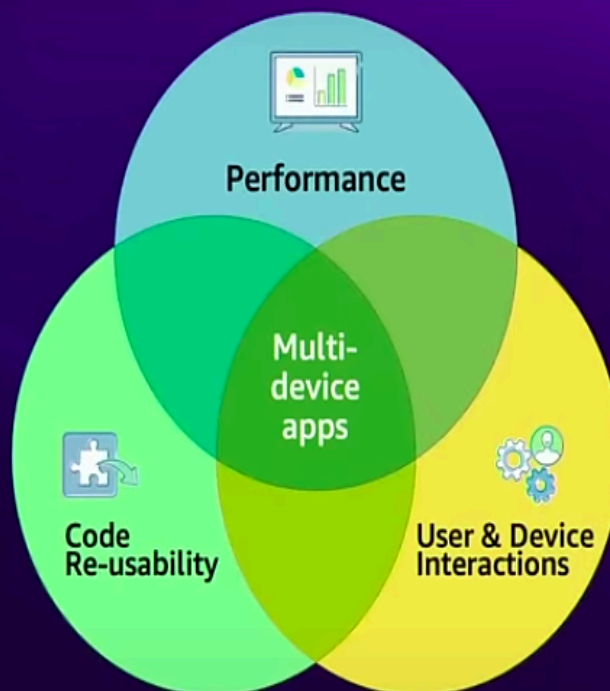




Extending React Native Beyond Mobile & Desktop Apps

Learn once, write anywhere... watch everywhere



React Native...for TVs?

- **2017:** Support for Apple TV features [v0.49](#)
- **2018:** Android TV [v0.55](#)
- **2020:** TV support to `react-native-tvos` [v0.62](#)
- **2023:** Support for Hermes on Apple TV [v0.71](#)
- **2024:** Build tvOS & Android TV apps in [Expo SDK 50](#)

```
> npm i react-native-tvos
```

React Native

Learn once, write anywhere.
Build mobile apps with React.

License MIT · Dockerfile Dockerfile · npm package npm package · PRs PRs · X Follow Breachnotifier

[Getting Started](#) · [Learn the Basics](#) · [Showcase](#) · [Contribute](#) · [Community](#) · [Support](#)

React Native brings [React's](#) declarative UI framework to iOS and Android. With React Native, you use native UI controls and have full access to the native platform.

- **Declarative.** React makes it painless to create interactive UIs. Declarative views make your code more predictable and easier to debug.
- **Component-Based.** Build encapsulated components that manage their state, then compose them to make complex UIs.
- **Developer Velocity.** See local changes in seconds. Changes to JavaScript code can be live reloaded without rebuilding the native app.
- **Portability.** Reuse code across iOS, Android, and [other platforms](#).

React Native is developed and supported by many companies and individual core contributors. Find out more in our [ecosystem overview](#).

Contents

- [Requirements](#)
- [Building your first React Native app](#)
- [Documentation](#)
- [Upgrading](#)
- [How to Contribute](#)
- [Code of Conduct](#)
- [License](#)

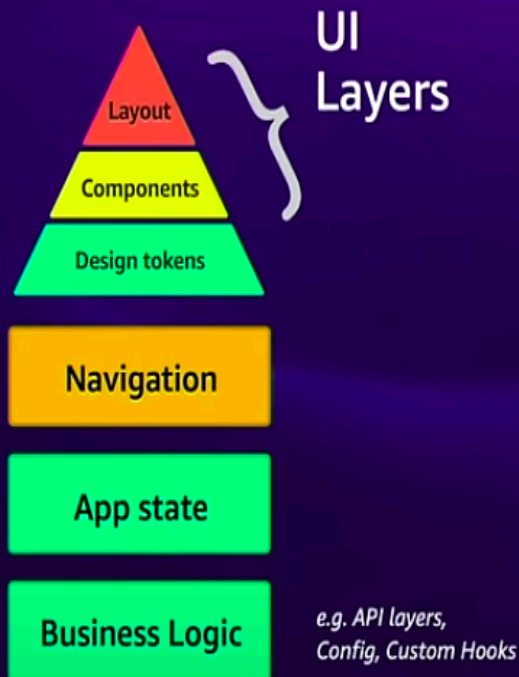
Requirements

React Native apps may target iOS 13.4 and Android 6.0 (API 23) or newer. You may use Windows, macOS, or Linux as your development operating system, though building and running iOS apps is limited to macOS. Tools like [Expo](#) can be used to work around this.

🔥 Building your first React Native app

Follow the [Getting Started guide](#). The recommended way to install React Native depends on your project. Here you can find short guides for the most common scenarios:

Code Reusability



Building with React Native for Amazon devices



User inputs



App Navigation



Focus management

TVEventHandler

```
<View style={styles.container}>
  <TVFocusGuideView autoFocus style={{ backgroundColor: 'blue' }}>
    <View style={styles.sectionTitleContainer}>
      <View>
        <Text style={styles.sectionTitle}>{selectedTitle}</Text>
      </View>
      <View style={styles.heroContainer}>
        <TouchableOpacity style={styles.heroButton} onFocus={() => {}}>
          <Text style={styles.heroButtonTitle}>Play Now</Text>
        </TouchableOpacity>
      </View>
    </TVFocusGuideView>
    <TVFocusGuideView autoFocus style={{ backgroundColor: 'red' }}>
      {data.map(renderCategory)}
    </TVFocusGuideView>
  </View>
```



bamlab/react-tv-space-navigation

```
const Rabbit = ({ onSelect }) => {  
  <SpatialNavigationFocusableView onSelect={onSelect}>  
    {{{ isFocused }}} <=>  
    <RabbitLayout isFocused={isFocused} />  
  </SpatialNavigationFocusableView>  
};  
  
const RabbitRow = () => {  
  <SpatialNavigationScrollView horizontal>  
    <SpatialNavigationView direction="horizontal">  
      {rabbits.map((_, index) => {  
        <Rabbit onSelect={() =>  
          console.log('selected rabbit ', index)} />  
      })}  
    </SpatialNavigationView>  
  </SpatialNavigationScrollView>  
};  
  
const Page = () => {  
  <SpatialNavigationRoot>  
    <SpatialNavigationScrollView>  
      <RabbitRow />  
      <RabbitRow />  
      <RabbitRow />  
      <RabbitRow />  
      <RabbitRow />  
    </SpatialNavigationScrollView>  
  </SpatialNavigationRoot>  
};
```

