

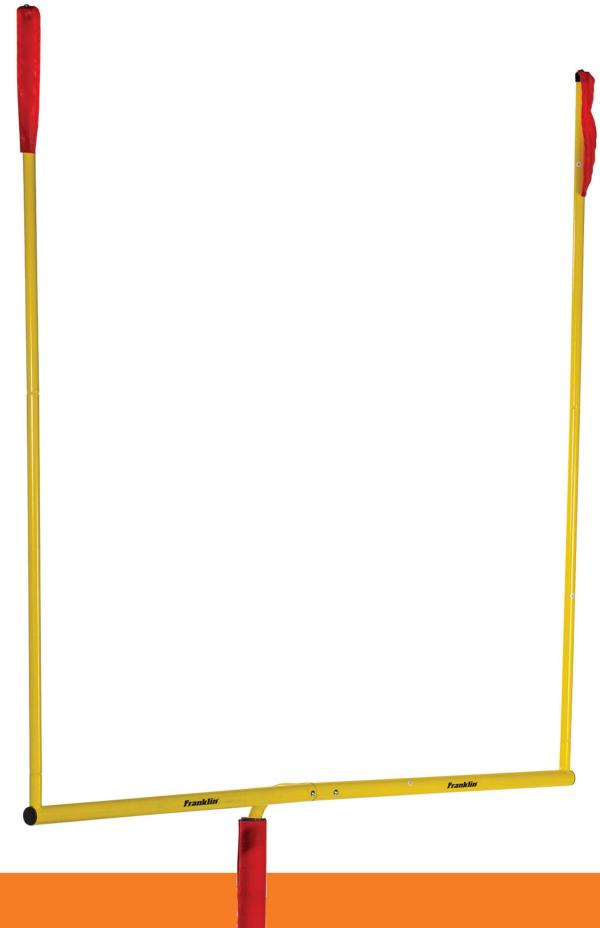
Screens





Goals

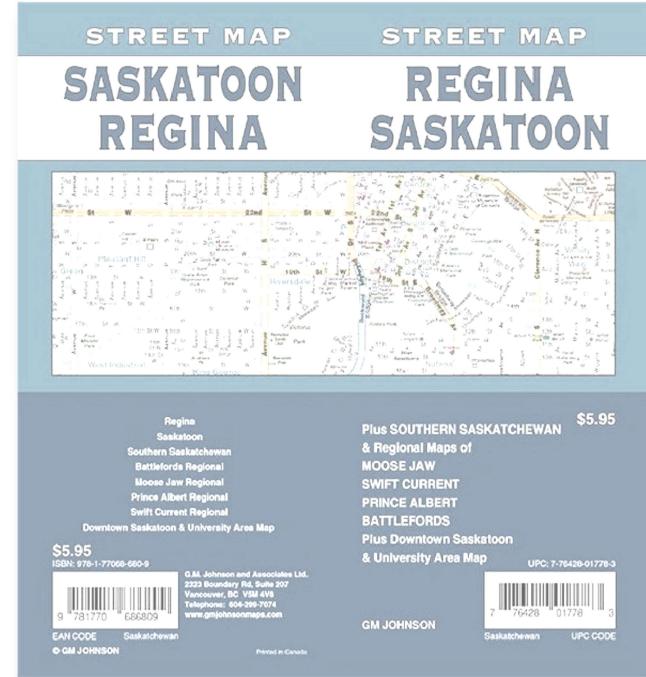
1. Compare Alert and <Modal>
2. List 2 flavors of **navigator**





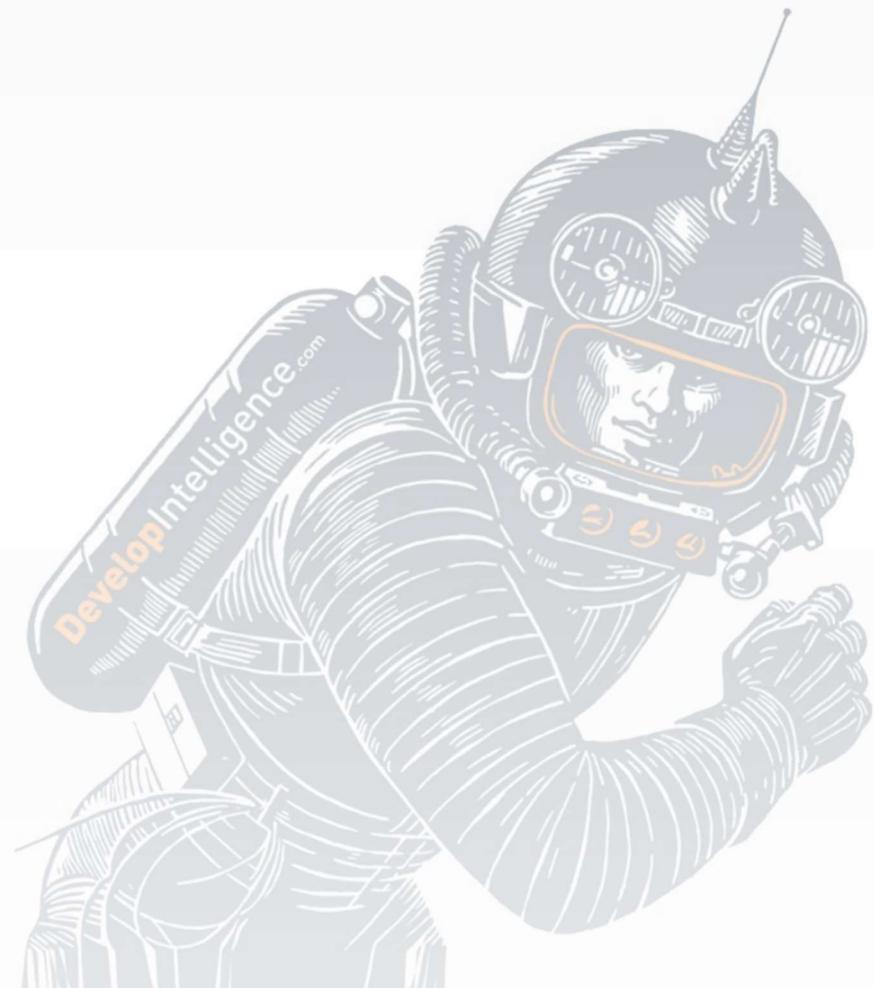
Roadmap

1. Alerts
2. Modals
3. Stacks
4. Drawers





Alerts





Overview

- Alert launches an actual (low-budget) dialog
 - On iOS, a user can enter information
- Better than a browser alert
- More like a browser confirm / prompt.



Vanilla Example

```
1 export default App = () => {
2   //Notice: Same as `window.alert`
3   const handlePress = ()=> alert('Everything Deleted!');
4
5   return <View style={{
6     flex:1,
7     justifyContent: 'space-around',
8     alignItems: 'center',
9   }}>
10    <Button
11      title='Delete everything!'
12      onPress={handlePress}
13      color='red'
14    />
15    </View>;
16};
```



More Buttons

```
1 export default App = () => {
2   const handlePress = ()=> Alert.alert(
3     'Confirm',
4     'Delete everything? Better be sure.',
5     [
6       {text: 'Cancel',onPress: () => alert('Cancelled.')},
7       {text: 'OK',  onPress: () => alert('Everything deleted!')},
8     ]
9   );
10
11  return <View style={{flex:1,justifyContent: 'space-around'}}>
12    <Button title='Delete everything!'
13      onPress={handlePress}
14      color='red'>
15    </View>;
16};
```



Platform Details

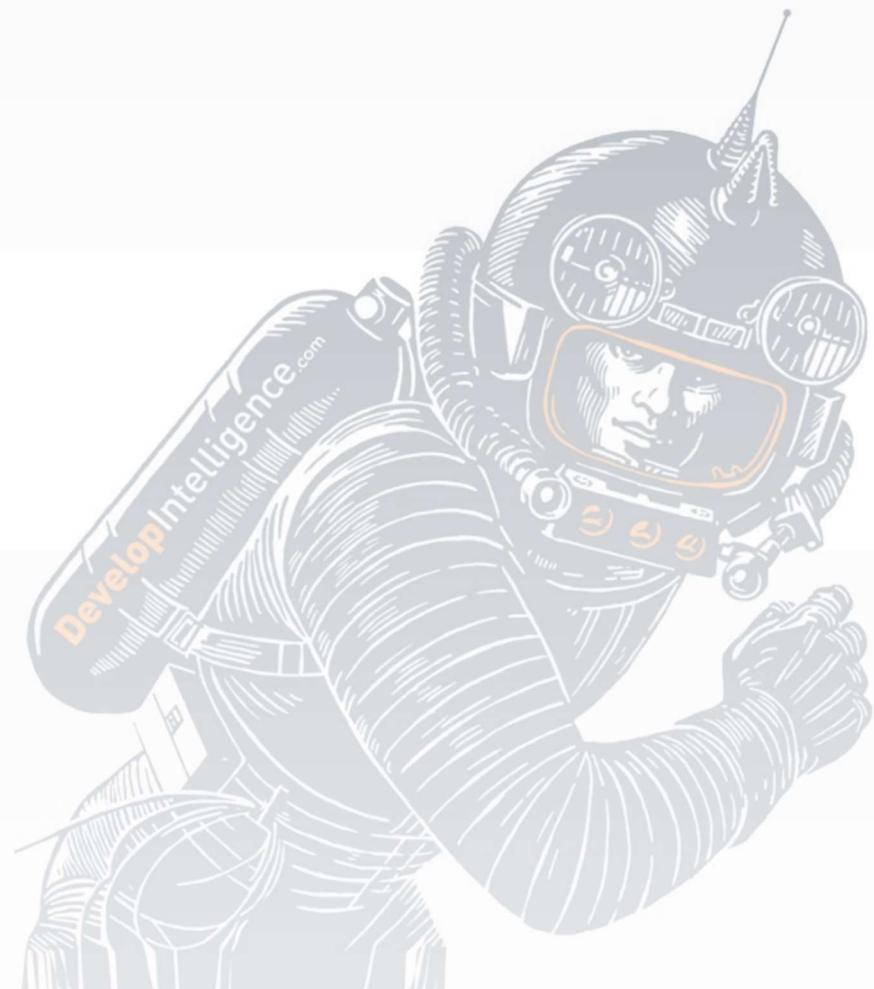


- **iOS**
 - As many buttons as you like
 - Each is stylable
- **Android**
 - Maximum of 3 buttons
 - Buttons styled differently based on number (neutral, negative, positive)
- **Web**
 - Not supported!



Takeaways

- Alerts kind of suck
- Suited only for the simplest of use cases



Modals





Overview

- Modal is a *basic way to present content above an enclosing view.*
- Better than alert
- Mobile equivalent of a dialog



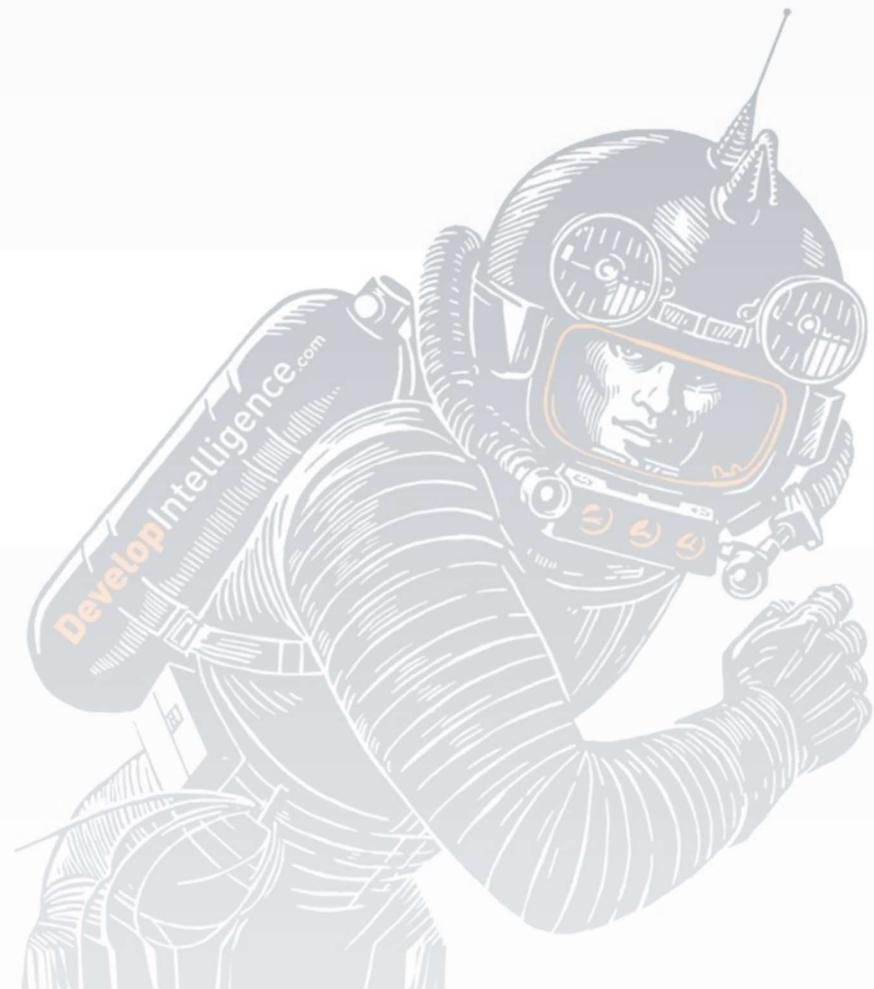
Example

```
1 export default App = () => {
2   const [modalVisible, setModalVisible] = useState(false);
3   return <View style={{flex:1,justifyContent: 'space-around'}}>
4     <Modal
5       animationType="fade"
6       visible={modalVisible}
7       onRequestClose={() => setModalVisible(!modalVisible)}
8     >
9       <Text>Hello World!</Text>
10      <Button title = 'Hide Modal'
11        onPress={() => setModalVisible(!modalVisible)}
12      />
13    </Modal>
14    <Button title='Show Modal'
15      onPress={() => setModalVisible(true)}
16    />
17  </View>
```



Takeaways

- Modals are pretty good
 - Arbitrarily customizable
- But they're still not great for representing a page (or a screen)



Stacks





Background

- Mobile apps have
 - No address bar
 - No back/forward buttons
- Varieties of navigation
 - Stack
 - Drawer
 - Tabs



About Stack Navigation



- **Good for** navigating between related screen
- **Example** Delivery app
 - Order Summary > Edit Order > Edit item #2
 - Order Summary > Edit driver instructions



React Navigation



- Open source, community library
- Works well with Expo & React CLI

```
1 | npm install @react-navigation/native --Save  
2 | npx expo install react-native-screens react-native-safe-area-context @react-na
```



Key Players

- Screens
- Navigator
- Container



Step 1: Create Navigator



```
1 import { createStackNavigator } from "react-navigation-stack";
2 import { ReviewOrder } from "@/screens/review-order";
3 import { OrderDetail } from "@/screens/OrderDetail";
4
5 const OrderStack = createNativeStackNavigator();
6
7 function App() {
8     return (
9         <NavigationContainer>
10         <OrderStack.Navigator>
11             </OrderStack.Navigator>
12         </NavigationContainer>
13     );
14 }
```



Step 2: Add Screens



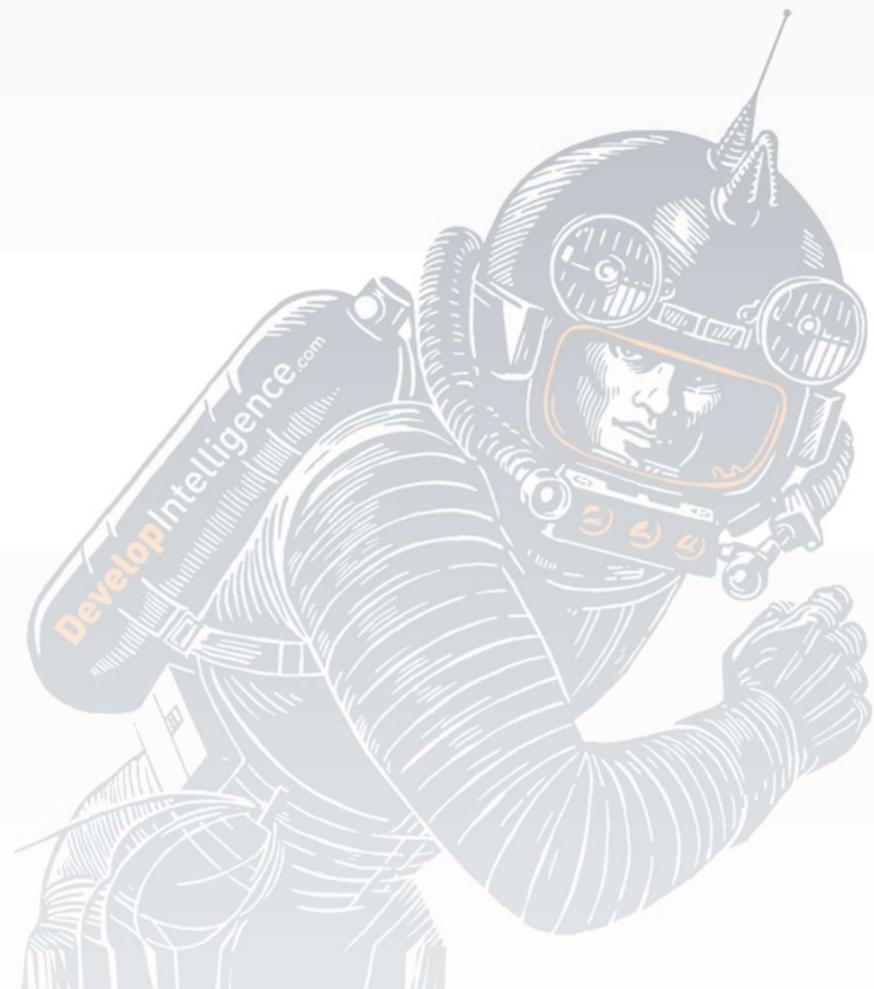
- An app container is a renderable component

```
1 const VillainStack = createNativeStackNavigator();
2
3 function App() {
4   return (
5     <NavigationContainer>
6       <VillainStack.Navigator>
7         <VillainStack.Screen name="Review" component={ReviewScreen} />
8         <VillainStack.Screen name="Details" component={DetailsScreen} />
9       </VillainStack.Navigator>
10      </NavigationContainer>
11    );
12 }
```

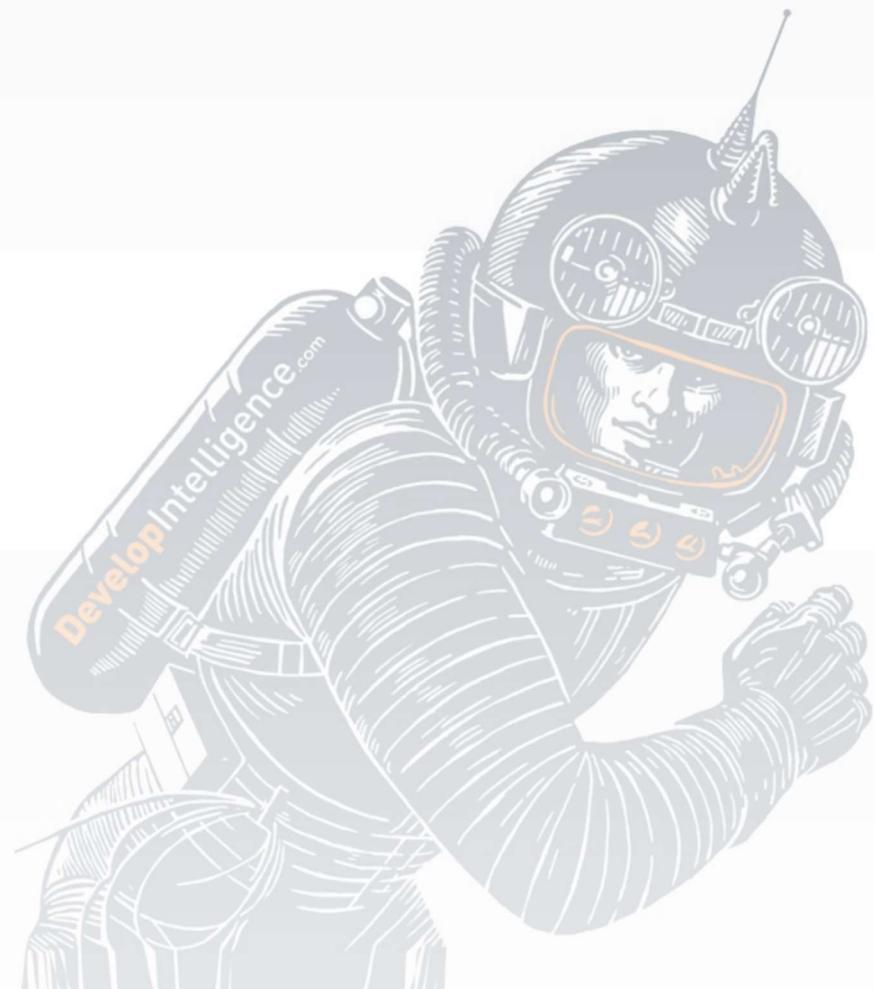


Step 3: Navigate

```
1 interface Props{
2     navigation:{navigate:(s:string)=>void}
3 }
4
5 export const Home = ({navigation}:Props) => {
6
7     const onPress = ()=>
8         navigation.navigate('VillainDetail');
9
10    return <View>
11        {/* & etc....*/}
12        <Button title={"Details"} onPress={onPress}/>
13    </View>;
14};
```



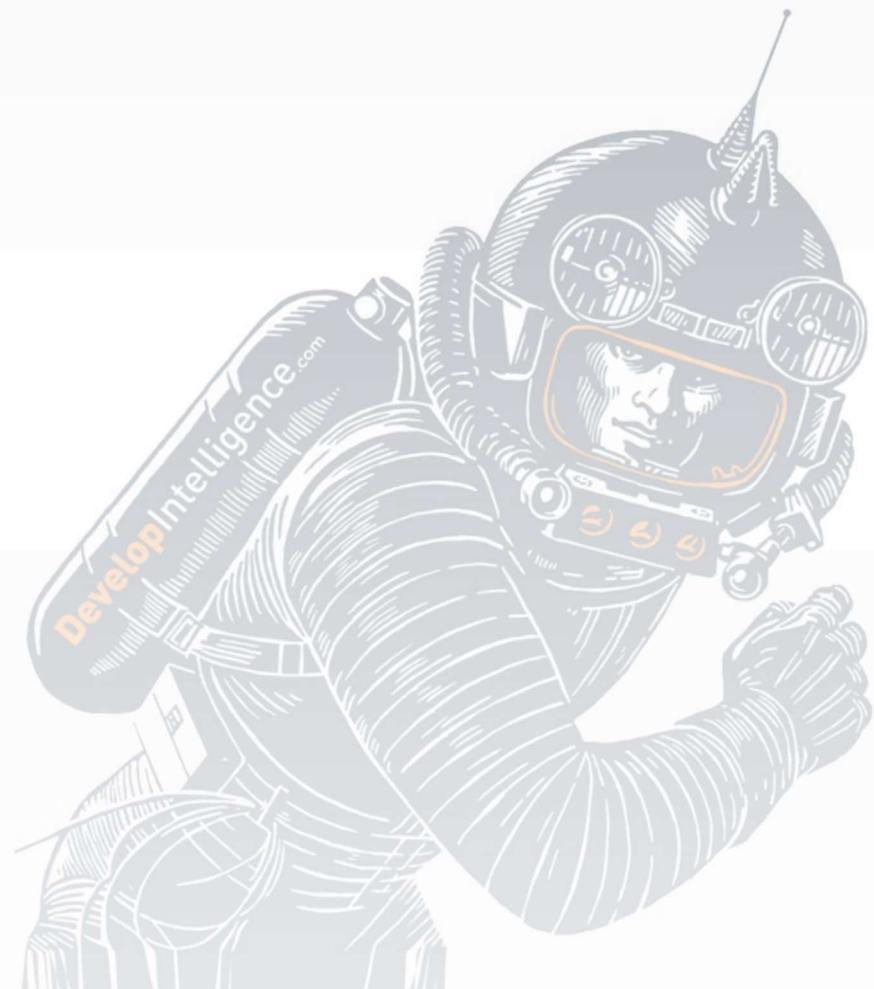
Drawers





Overview

- Navigate to arbitrary screens
- **Good for** top-level screens unrelated to eachother





Review

1. Compare Alert and <Modal>
2. List 2 flavors of **navigator**

