

PlatformColor

```
PlatformColor(color1, [color2, ...colorN]);
```

You can use the `PlatformColor` function to access native colors on the target platform by supplying the native color's corresponding string value. You pass a string to the `PlatformColor` function and, provided it exists on that platform, it will return the corresponding native color, which you can apply in any part of your application.

If you pass more than one string value to the `PlatformColor` function, it will treat the first value as the default and the rest as fallback.

```
PlatformColor('bogusName', 'linkColor');
```

Since native colors can be sensitive to themes and/or high contrast, this platform specific logic also translates inside your components.

Supported colors

For a full list of the types of system colors supported, see:

- Android:
 - R.attr - `?attr` prefix
 - R.color - `@android:color` prefix
- iOS (Objective-C and Swift notations):
 - UIColor Standard Colors
 - UIColor UI Element Colors

Developer notes

Web

If you're familiar with design systems, another way of thinking about this is that PlatformColor lets you tap into the local design system's color tokens so your app can blend right in!

Example

PlatformColor Example

^ Expo

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

const App = () => (
  <View style={styles.container}>
    <Text style={styles.label}>I am a special label color!</Text>
  </View>
);

const styles = StyleSheet.create({
  label: {
    padding: 16,
    ...Platform.select({
      ios: {
        color: PlatformColor('label'),
        backgroundColor: PlatformColor('systemTealColor'),
      },
      android: {
        color: PlatformColor('?android:attr/textColor'),
        backgroundColor:
PlatformColor('@android:color/holo_blue_bright'),
      },
      default: {color: 'black'},
    }),
  },
});
```

Preview

My Device

iOS


Android

The string value provided to the PlatformColor function must match the string as it exists on the native platform where the app is running. In order to avoid runtime errors, the function should be wrapped in a platform check, either through a Platform.OS === 'platform' or a Platform.select(), as shown on the example above.

Note: You can find a complete example that demonstrates proper, intended use of

PlatformColor in PlatformColorExample.js.

Is this page useful?  

 Edit this page

Last updated on **Jun 21, 2023**