

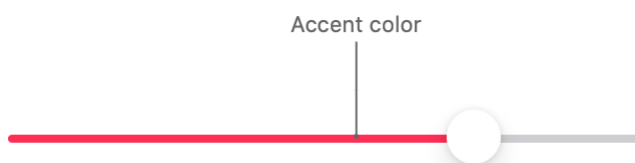
## Article

# Specifying your app's color scheme

Set a global accent color for your app by using asset catalogs.

## Overview

An *accent color*, or *tint color*, is a broad theme color that applies to views and controls in your app. Use an accent color to quickly create a unifying color scheme for your app. You can set an accent for your app by specifying an accent color in your asset catalog.



## Create an accent color set

When you create your project from a template, it automatically includes a default asset catalog (`Assets.xcassets`) with an `AccentColor` color set. Xcode applies the color you specify in this color set as your app's accent color.

If your app doesn't have an `AccentColor` color set, create a color set manually.

1. In the Project navigator, select an asset catalog.
2. Click the Add button (+) at the bottom of the outline view.
3. In the pop-up menu, choose Color Set. A new color set appears in the outline view, and opens in the detail area.
4. Double-click the color set name in the outline view to rename the color set with a descriptive name, and press the Return key.

5. In Build Settings, find the build setting for “Global Accent Color Name”. Double-click the build setting, type in the name of your accent color set, and press Return.

## Specify accent color variations

When you select an accent color, choose a color that works well in light and dark appearances. In your accent color set, you can specify different color values for light and dark appearances if necessary.

1. In the Project navigator, select an asset catalog.
2. In the outline view, select the accent color set.
3. Open the Attributes inspector. In the Appearances field, choose the appearances for which you want to specify color values. Additional wells appear in the detail area for the appearance options you specify.
4. Select a well, and set a color by using the Content field in the Attributes inspector. Use the Any Appearance well to specify the color value the app uses on systems that don’t differentiate between light and dark appearances.

You can also specify high-contrast versions of your colors by selecting the High Contrast checkbox.

## Access the accent color from your code

By default, your accent color applies to all views and controls in your app that use a tint color, unless you override the color for a specific subset of the view hierarchy. However, you might want to incorporate the accent color into other parts of your user interface that don’t rely on a tint color, like static text elements.

To use the accent color value from an asset catalog in code, load the color like this:

```
// SwiftUI
Text("Accent Color")
    .foregroundColor(Color.accentColor)

// UIKit
label.textColor = UIColor.tintColor
```

## See Also

# Colors

## Supporting Dark Mode in your interface

Update colors, images, and behaviors so that your app adapts automatically when Dark Mode is active.