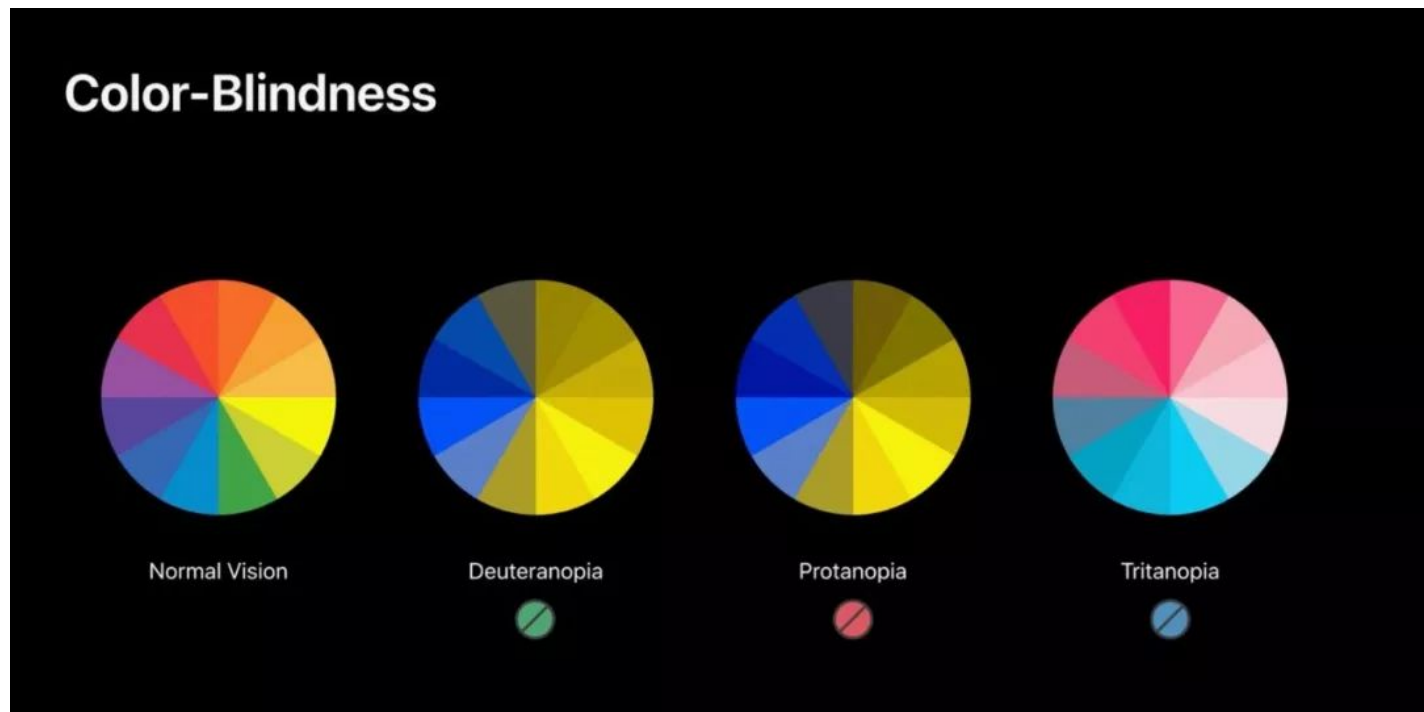


iOS 13 Dark Mode 适配的动态颜色 cheatsheet

原创：hite 知识小集 3天前



作者 | hite，目前在网易严选iOS 组，主要工作内容 webview 相关，业余时间会写一些胡思乱想产品策划稿，各类游戏云玩家

深色模式，作为 iOS 13 最大的新特性，从设计者角度，带来设计体系、颜色、材质、系统控件、SF Symbols 等若干方面的新的变化；对于开发者来讲，我们熟知的适配；

- 屏幕尺寸
- 屏幕方向
- View 渲染阶段（指 viewDidLoad 和 viewWillAppear 两个阶段，view 的尺寸是变化的）
- iOS SDK 适配

等适配维度，而后又多了一个维度，

- 外观模式（dark or light）

Dark mode 的适配，我们需要处理图片、背景、填充、文字和分割线的逻辑。首

先是设计师对两种外观，必须设计两套不同的颜色方案。当两套方案送到开发者手上的时候，我们也需要根据不同外观应用到不同的颜色。但幸好，开发者可以做一个封装把这层逻辑封装起来，给其他开发者使用时，屏蔽内部逻辑。为了这个目的，iOS 提供了被动式适配——动态颜色；

```
let backgroundColor = UIColor { (trainCollection) -> UIColor
in
    if trainCollection.userInterfaceStyle == .dark {
        return UIColor.black
    } else {
        return UIColor.white
    }
}
view.backgroundColor = backgroundColor
```

和主动式适配，我们可以在 UIViewController 或 UIView 中调用 `traitCollection.userInterfaceStyle` 来获取当前视图的样式；

```
if traitCollection.userInterfaceStyle == .dark {
    // Dark
} else {
    // Light
}
```

或者使用 `name assets` 技术，提前配置，让系统自动切换。

实际使用下来，还是有点麻烦。为了进一步简化适配的代码量，Apple 特意提供了所谓的 `DynamicColorProvider` 机制，内置的一些动态颜色。

即一个「字体颜色」，在深色模式当中可以指代白色，而在浅色模式当中可以指代黑色。

```
textLabel.textColor = UIColor.labelColor
```

在 iOS Design System 里，对界面的文字、背景、图标做了信息分层，相应的需要不同配色。我们以文本颜色为例，iOS 13 内置了 4 层；

- UIColor.labelColor,
- UIColor.secondaryLabelColor,
- UIColor.tertiaryLabelColor,
- UIColor.quaternaryLabelColor

「文本色」 (*LabelColor*) 是一级字色，可以提供最高的对比度，主要用于最为重要的内容元素，如内容的主标题。而「二级文本色」 (*SecondaryLabelColor*) 可以用于副标题，「三级文本色」 (*TertiaryLabelColor*) 用于输入框占位符文本，「四级文本色」 (*QuaternaryLabelColor*) 用于禁用状态的文本。

使用动态颜色是最简洁适配 dark 模式的方法。但是它有问题: labelColor 的色值是多少？dark 是多少，light 是多少？
















labelColor 的答案很简单，但是 iOS 一共引入了 24 动态颜色，48 种色值。真正正在和我们 App 自身的样式系统合并的时候，设计师和开发者都不知道，内置的动态色和 App 自身相协调？

当我为《9 星浏览器》做适配的时候，我的头都是大，我知道 UIColor.separatorColor 可以自动变色，但是色值和我 App 的样式是否和谐，没有色值让我对比。我尝试从网上找 DynamicColor 的色值大全的时候，一时半会没有找到。所以我自己来做一份。


深色模式下的配色















颜色	#hex	rgba
















<div></div> labelColor	#FFFFFFF	rgba(255,255,255,1.00)
<div></div> secondaryLabelColor	#EBEBF59	rgba(235,235,245,0.60)
<div></div> tertiaryLabelColor	#EBEBF54C	rgba(235,235,245,0.30)
<div></div> quaternaryLabelColor	#EBEBF52D	rgba(235,235,245,0.18)
<div></div> linkColor	#0984FFF	rgba(9,132,255,1.00)
<div></div> placeholderTextColor	#EBEBF54C	rgba(235,235,245,0.30)
<div></div> separatorColor	#5454589	rgba(84,84,88,0.60)
<div></div> opaqueSeparatorColor	#38383AF	rgba(56,56,58,1.00)
<div></div> systemBackgroundColor	#000000F	rgba(0,0,0,1.00)
<div></div> secondarySystemBackgroundColor	#1C1C1EF	rgba(28,28,30,1.00)
<div></div> tertiarySystemBackgroundColor	#2C2C2EF	rgba(44,44,46,1.00)
<div></div> systemGroupedBackgroundColor	#000000F	rgba(0,0,0,1.00)
<div></div> secondarySystemGroupedBackground Color	#1C1C1EF	rgba(28,28,30,1.00)
<div></div> tertiarySystemGroupedBackgroundCol or	#2C2C2EF	rgba(44,44,46,1.00)
<div></div> systemFillColor	#7878805B	rgba(120,120,128,0.36)
<div></div> secondarySystemGroupedBackground Color	#1C1C1EF	rgba(28,28,30,1.00)
<div></div> tertiarySystemGroupedBackgroundCol or	#2C2C2EF	rgba(44,44,46,1.00)
<div></div> systemFillColor	#7878805	rgba(120,120,128,0.3

	B	6)
 secondarySystemFillColor	#78788051	rgba(120,120,128,0.32)
 tertiarySystemFillColor	#7676803D	rgba(118,118,128,0.24)
 quaternarySystemFillColor	#7676802D	rgba(118,118,128,0.18)
 systemRedColor	#FF453AF	rgba(255,69,58,1.00)
 systemGreenColor	#30D158F	rgba(48,209,88,1.00)
 systemBlueColor	#0A84FFF	rgba(10,132,255,1.00)
 systemOrangeColor	#FF9F0AF	rgba(255,159,10,1.00)
 systemYellowColor	#FFD60AF	rgba(255,214,10,1.00)
 systemPinkColor	#FF375FF	rgba(255,55,95,1.00)
 systemPurpleColor	#BF5AF2F	rgba(191,90,242,1.00)
 systemTealColor	#64D2FFF	rgba(100,210,255,1.00)
 systemIndigoColor	#5E5CE6F	rgba(94,92,230,1.00)
 systemGrayColor	#8E8E93F	rgba(142,142,147,1.00)
 systemGray2Color	#636366F	rgba(99,99,102,1.00)
 systemGray3Color	#48484AF	rgba(72,72,74,1.00)

浅色模式下的配色

颜色	#hex	rgba
	#8E8E93F	rgba(142,142,147,1.00)

 labelColor	#000000FF	rgba(0,0,0,1.00)
 secondaryLabelColor	#3C3C4399	rgba(60,60,67,0.60)
 tertiaryLabelColor	#3C3C434C	rgba(60,60,67,0.30)
 quaternaryLabelColor	#3C3C432D	rgba(60,60,67,0.18)
 linkColor	#007AFFFF	rgba(0,122,255,1.00)
 placeholderTextColor	#3C3C434C	rgba(60,60,67,0.30)
 separatorColor	#3C3C4349	rgba(60,60,67,0.29)
 opaqueSeparatorColor	#C6C6C8FF	rgba(198,198,200,1.00)
systemBackgroundColor	#FFFFFFFF	rgba(255,255,255,1.00)
 secondarySystemBackgroundColor	#F2F2F7FF	rgba(242,242,247,1.00)
tertiarySystemBackgroundColor	#FFFFFFFF	rgba(255,255,255,1.00)
 systemGroupedBackgroundColor	#F2F2F7FF	rgba(242,242,247,1.00)
secondarySystemGroupedBackgroundColor	#FFFFFFFF	rgba(255,255,255,1.00)
 tertiarySystemGroupedBackgroundColor	#F2F2F7FF	rgba(242,242,247,1.00)
 systemFillColor	#78788033	rgba(120,120,128,0.20)
secondarySystemGroupedBackgroundColor	#FFFFFFFF	rgba(255,255,255,1.00)
 tertiarySystemGroupedBackgroundColor	#F2F2F7FF	rgba(242,242,247,1.00)
 systemFillColor	#78788033	rgba(120,120,128,0.20)

 systemFillColor		
 secondarySystemFillColor	#78788028	rgba(120,120,128,0.16)
 tertiarySystemFillColor	#7676801E	rgba(118,118,128,0.12)
 quaternarySystemFillColor	#74748014	rgba(116,116,128,0.08)
 systemRedColor	#FF3B30FF	rgba(255,59,48,1.00)
 systemGreenColor	#34C759FF	rgba(52,199,89,1.00)
 systemBlueColor	#007AFFFF	rgba(0,122,255,1.00)
 systemOrangeColor	#FF9500FF	rgba(255,149,0,1.00)
 systemYellowColor	#FFCC00FF	rgba(255,204,0,1.00)
 systemPinkColor	#FF2D55FF	rgba(255,45,85,1.00)
 systemPurpleColor	#AF52DEF	rgba(175,82,222,1.00)
 systemTealColor	#5AC8FAFF	rgba(90,200,250,1.00)
 systemIndigoColor	#5856D6FF	rgba(88,86,214,1.00)
 systemGrayColor	#8E8E93FF	rgba(142,142,147,1.00)
 systemGray2Color	#AEAEB2FF	rgba(174,174,178,1.00)

从上面表格里，我们有一些发现；

- 动态颜色分为 3 大类：文本颜色，用于最基本的信息前景色；填充色，指小块的区域和文字配合传达信息；大面积的背景色，这些颜色是对比色，辅助主题色的显示，如 tableview 的背景色。
- 对于文本颜色，4 级颜色，第二级到第四级的颜色包含了透明度，但命名中没有

体现；separatorColor 则有明确的不透明版本 opaqueSeparatorColor；

- 相同颜色的不同模式颜色并不是取相反值（如 labelColor），大部分颜色都是细调，典型的如 linkColor。浅色：rgba(0,122,255,255)，深色：rgba(9,132,255,255)
- systemRedColor 系列的颜色不受外观模式影响。

结论

我用系统动态颜色适配了一个版本的《9 星浏览器》和远程键盘，隐约感觉比之前的配色那里少了点东西，虽然大部分还是相似的。凌晨 1 点提交审核，躺在床上睡不着，心里一直不爽，最后还是爬起来，把全部内置动态颜色改为自定义，凌晨 3 点重新提交；

```
if #available(iOS 13.0, *) {
    let backgroundColor = UIColor { (trainCollection)
-> UIColor in
        if trainCollection.userInterfaceStyle ==
.light {
            return UIColor(hex: "#f3f3f3ff")!
        } else {
            return UIColor(hex: "#2e2f30ff")!
        }
    }
    tableView.separatorColor = backgroundColor
} else {
    // Fallback on earlier versions
    tableView.separatorColor = UIColor(hex:
"#e3e3e3ff")
}
```


是的，对于大部分应用，你需要同时处理 dark mode 适配和 iOS 版本适配；当然你可以写 category 或 extension，把上面代码简化为

```
tableView.separatorColor = UIColor(hex: "#e3e3e3ff",
darkHex: "#2e2f30ff")
```

结论：扔掉 iOS 内置的动态颜色，他们毫无用处，上面的表格只是为了证明 动态颜色是多么鸡肋。

特别提醒

- 使用 Any, Dark 的 imageassets 在设置导航栏的自定义 backButton 图标时不好用（可能是我姿势不对），改为

```
if #available(iOS 13.0, *) {
    // 先设置，不设置后面的 tint 设置不上
    let navBarAppearance = UINavigationBarAppearance()
    navBarAppearance.setBackIndicatorImage(backImage,
transitionMaskImage: backImage)
    navBarAppearance.titleTextAttributes = [.font:
UIFont.systemFont(ofSize: 16, weight: .regular)]
    navBarAppearance.shadowImage = UIImage()
    navBarAppearance.shadowColor = .clear

    navigationController?.navigationBar.standardAppearance =
navBarAppearance
    // 改文字和 backimage 图片的颜色
    let backgroundColor = UIColor { (trainCollection)
-> UIColor in
        if trainCollection.userInterfaceStyle ==
.light {
```

```

        return UIColor.black
    } else {
        return UIColor.white
    }
}

navigationController?.navigationBar.tintColor =
backgroundColor
}

```

- 如何启用 dark 模式。对于主 App 而言，只要用户启用了 dark 模式，你的 App 就是 dark 模式；但是对于像《9 星浏览器》的远程键盘、财务键盘的 dark 模式，则需要宿主 App 也启用 dark 模式适配，如远程键盘在微信里没有深色模式，在 Safari 里有的；
- 设置里或者在 xcode 切换外观模式？App 如何响应？对于 ViewController，我们知道 iOS 13 提供了一个监听

```

override func traitCollectionDidChange(_
previousTraitCollection: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraitCollection)
    if traitCollection.hasDifferentColorAppearance(comparedTo:
previousTraitCollection) {
        // 适配代码
    }
}

```

但是对于动态颜色是如何作用的呢？如

```

text.textColor = UIColor.label

```

原因：系统触发了 `updateDisplay`，怎么触发还没搞明白，肯定不是

```
self.view.setNeedDispay()
```

触发的。不会触发 `layoutSubviews`，更不会触发 `viewDidLoad`。

- 谨慎适配 dark mode，尤其是引用了第三方 UI 组件的 App，他们没升级，你在适配时，会遇到千奇百怪的界面。

- 上面的表格是自动生成的，如果想要添加新的色值或者改格式，你可以下 `ColorForDarkMode`^[3] 自行修改

参考

[1]<https://juejin.im/post/5cf6276be51d455a68490b26>

[2]<https://www.uisdc.com/ios-13-design>

[3]<https://github.com/hite/ColorForDarkMode>

[4]<https://noahgilmore.com/blog/dark-mode-uicolor-compatibility/>

推荐阅读

- [5 个有趣的 iOS 库](#)
- [iOS 获取任意线程调用栈](#)
- [在 GPUImage 中检测人脸关键点](#)
- [全面深度解析iOS端URL编码和解码过程](#)

[阅读原文](#)