

当前位置： 首页 > 编程社区 > Swift > Swift - 使用Image Literal实现代码中图片的智能提示（Xcode自带功能）

Swift – 使用Image Literal实现代码中图片的智能提示（Xcode自带功能）

发布：hangge 阅读：2610

2018-01-02

过去我们在代码中如果要使用一张资源库（Assets.xcassets）里的图片，通常都这么写：

```
1 | let image = UIImage(named: "email")
```

但它和上文提到的 UIColor 颜色一样，也存在同样的问题：

- 在代码中不能直观地看出图片的样子。
- 同时我们还要记对名字，如果把把 key 写错了，图片就不会显示。

一、Image Literal 介绍

为了解决上面这个问题，过去我们常常会装一些 UIImage 图片资源辅助插件（比如：KSIImageNamed）。这样在编写代码时，可以实时预览、或选择相应的图片。

而从 Xcode8 起，编辑器就自带了一个图片辅助功能：Image Literal。让我们不再需要再安装额外插件。

二、Image Literal 使用

Image Literal 提供如下两种使用方式。

方式1：

- (1) 假设我们要初始化一个 UIImage 对象，只要输入“Image Literal”（会有智能提示），然后回车。

```
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15
16         let image = imageLiteral
17
18         // String UIImagePickerControllerLivePhoto
19         // CFString kCGImageDestinationLossyCompressionQuality
20         // String kCIImageAutoAdjustLevel
21         // CFString kCGImagePropertyDNGLensInfo
22         // CFString kCGImagePropertyGPSLatitude
23         // CFString kCVImageBufferGammaLevelKey
24         // CFString kCGImagePropertyExifLensMake
25     }
}
```

- (2) 这时就会出现一个图片选择器小方块。

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        let image = 

    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }
}
```

文章类别

- 首页
- 编程社区
- SwiftFlex / FlutterHTML5 / CSS3React / RN其他
- 奇文共赏

本类相关

- Swift - CAGradientLayer渐变动画的实现2
- Swift - 字体图标的使用及样例（使用Font Awesome）
- Swift - RxSwift的使用详解40（UICollectionView）
- Swift - 网络抽象层库Moya的使用详解2（请
- Swift - 第三方加密库CryptoSwift使用详解2
- Swift - 模拟器路径、模拟器相簿照片路径位
- Swift - RxSwift的使用详解22（UI控件扩展）
- Swift - UIViewController生命周期详解（附：
- Swift - JPush极光推送的使用4（根据Tag标
- Swift - Framework的制作与使用教程3（与



(4) 选择完毕后那个小方块就会显示出图片的缩略图。

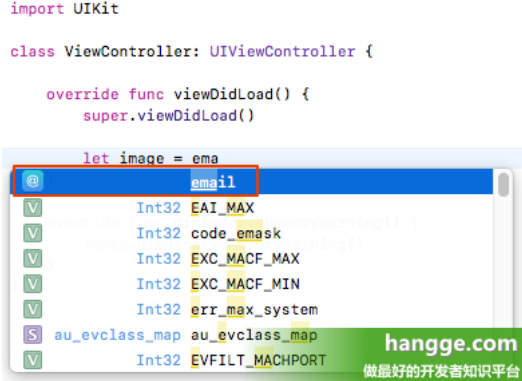


把这段代码复制出来，其实是长这样的：

```
let image = #imageLiteral(resourceName: "email")
```

方式2：

(1) 我们还可以直接输入图片名字，也会有智能提示和缩略图。



(2) 回车后，最终效果和上面的是一样的。

