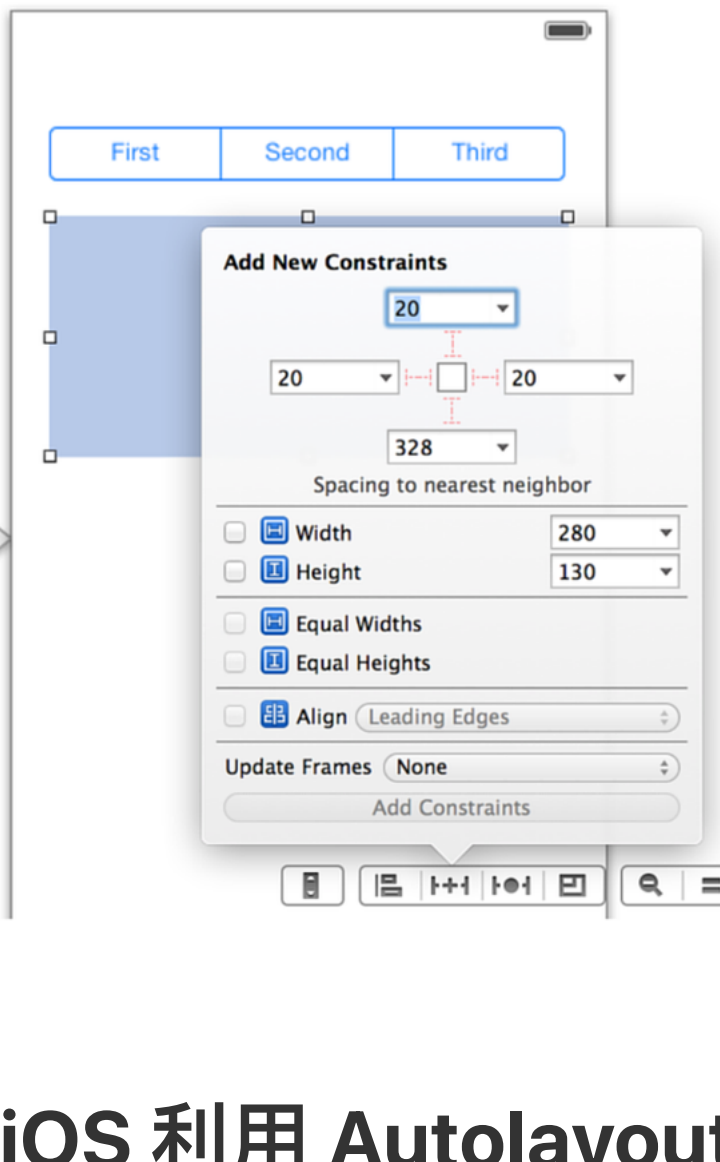


宇朋Look

2016年07月11日 阅读 2846

关注



Auto Layout

Tips and Tricks

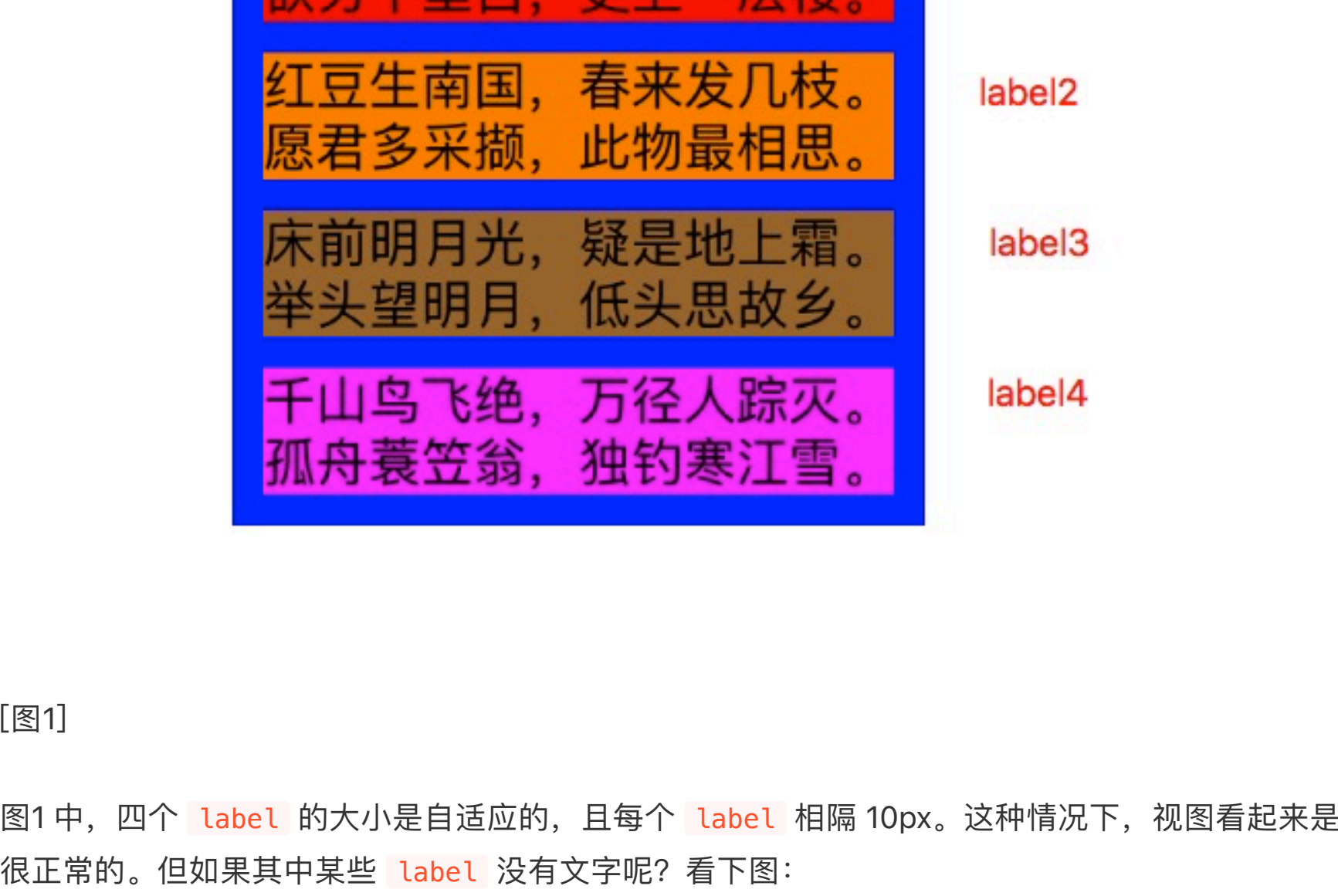
稀土掘金技术社区

iOS 利用 Autolayout 实现 view 间隔自动调整

原文链接：nathanli.cn

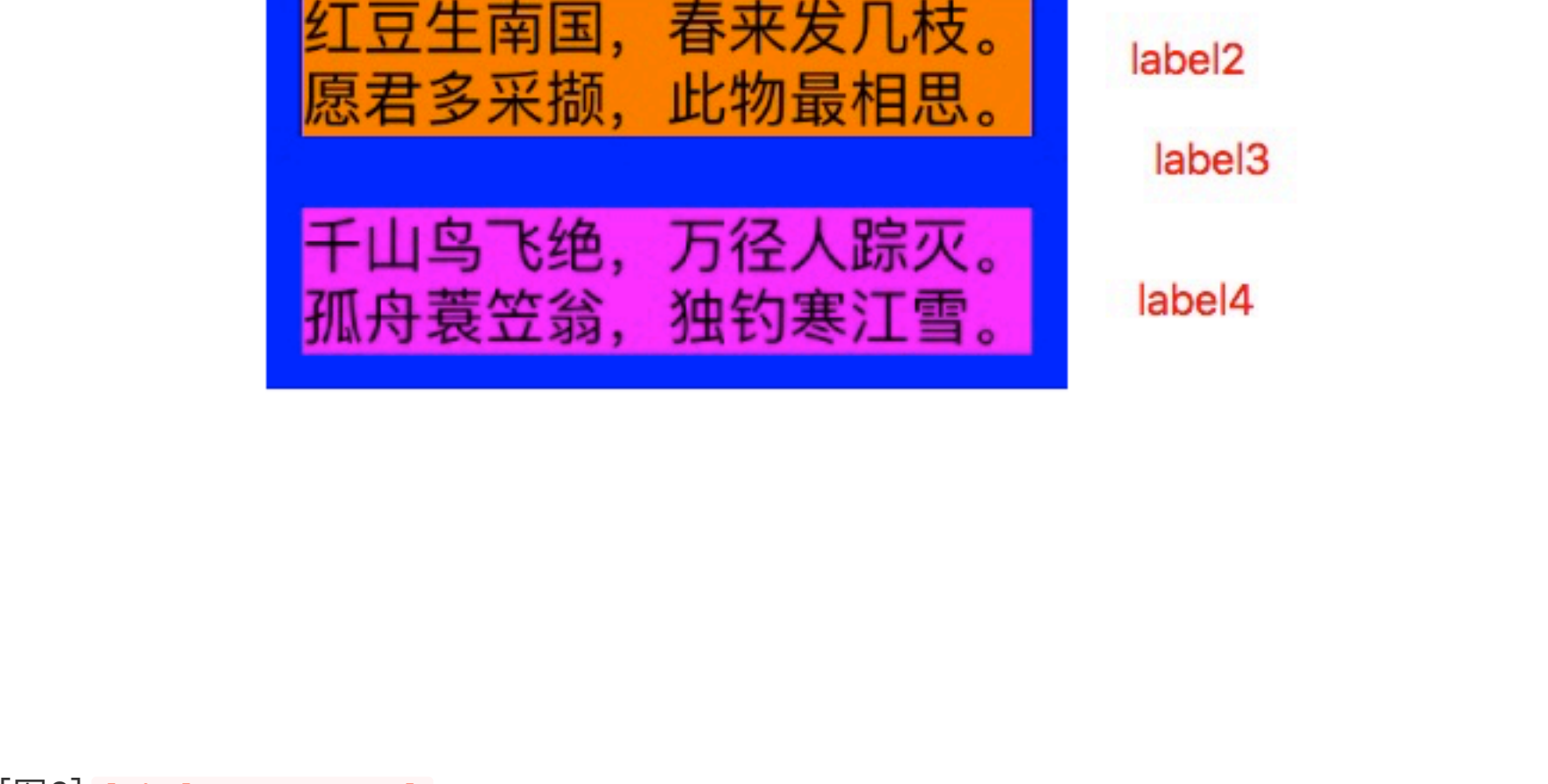
1、需求

不知道大家是否有这样的需求：一个界面中，有多个 view，每个 view 的大小由其内容决定。当一个 view 有内容时，下一个 view 与它之间会一个间隔。如果没有内容的话，下一个 view 就会紧挨着它，如下图所示：



[图1]

图1中，四个 `label` 的大小是自适应的，且每个 `label` 相隔 10px。这种情况下，视图看起来是很正常的。但如果其中某些 `label` 没有文字呢？看下图：



[图2] `label2.text = nil`



[图3] `label2.text = nil; label3.text= nil;`

图2是 `label3.text` 为 `nil`，中间有一个明显过大的间隔。这是由于 `label3` 的高度虽然为 0，但由于它与 `label2` 的间隔为 10px，而 `label4` 与它的间隔又是 10px，所以造成了图中 `label2` 与 `label4` 的过大的间隔。

图3则更为惨不忍睹。

这时，咱们的需求就出来了：`label` 与 `label` 之间的间隔能随着它们自身内容的变化而变化。当有文本时，间隔存在；当没有文本时，则紧挨在一起。当然，这里的间隔希望不仅是垂直的，水平方向也应该是一样的。

2、解决方案

来看看各种解决方法。

2.1 动态更新约束

这是最直观、容易想到的办法，就是在 `label2` 等内容有变化时，去调整相关的间隔约束：更改 `constants` 或 `优先级` 等。

这种方法存在的问题是维护成本太高，这里只有四个 `label`，但是要维护这种间隔约束关系主就已经很累了。所以这种方法是比较初级的，不灵活。

2.2 自定义 `-(CGSize)intrinsicContentSize`

视图的 `内容` 在 `auto layout` 中，其与约束是 **同等重要的**。视图有一个方法：`-(CGSize)intrinsicContentSize`，用来返回展示完整视力内容的最小 `size`。

比如 `UILabel` 就是根据它的 `text`、`attributedText` 和 `preferredMaxLayoutWidth` 等来计算它的 `内容 size`。

当视图内容改变时，可以调用 `-(void)invalidateIntrinsicContentSize` 方法来让 `Auto Layout` 在下次布局时重新计算。

咱们可以将间隔当作 `内容` 的一部分，将其计算在内：

```
@interface NLabelIntervalView : UIView

@property (nonatomic, strong) UILabel *label;
@property (nonatomic, assign) CGSize intervalSize;

@end

@implementation NLabelIntervalView

- (CGSize)intrinsicContentSize {
    CGSize size = [self.label intrinsicContentSize];
    if ([self.label.text length] > 0) {
        size.width += self.intervalSize.width;
        size.height += self.intervalSize.height;
    }

    return size;
}

@end
```

可以看到，这种方法在一定程度上可以解决间隔问题，但它有很大的不足：它将间隔 **嵌入** 到内容中；需要 **包装** 目标视图，这个代价却实在有点大。虽然利用 **继承** 可以一部分 **包装** 问题，但类似于这里的 `UILabel`，由于它内容的绘制方法（文字垂直居中），**继承** 是无法做到 **间隔** 的。不过在自定义视图时，如果就间隔考虑进去的话，问题倒是不大。

所以，这种方案适用于自定义视图中，对系统定义的视图帮助有限。

2.3 利用对齐矩形(alignment rect)

你可能会直观的认为 `Auto Layout` 中，约束是使用 `frame` 来确定视图的大小和位置的，但实际上，它使用的是 **对齐矩形(alignment rect)** 这个几何元素。不过在大多数情况下，`frame` 和 `alignment rect` 是相等的，所以你这么理解也没什么不对。

系统有 `frame` 不用，为啥要用 `alignment rect` 呢？

有时候，咱们在创建复杂的视图时，可能会添加各种装饰元素，如阴影、外边框、角标等等。但考虑到开发这样的视图所需时间成本，或者为了避免画面污染等原因，会找设计师直接切相应的成品图给咱们。如下图：



(图片来源：iOS Auto Layout Demystified)

上图中，(a)是咱们拿到的图，(c)是这个图的 `frame`。显然，咱们在布局的时候，不想将阴影和角标考虑进去（视图的 `center` 和 底边、右边都发生了偏移），而是只考虑中间的核心部分，如图 (b) 中框出的矩形所示。

对齐矩形就是用来处理这种情况的。

`UIView` 提供了方法，由 `frame` 得到 `alignment rect`：

```
// The alignment rectangle for the specified frame.
- (CGRect)alignmentRectForFrame:(CGRect)frame;
```

它得可逆，也就是说得能从 `alignment rect` 反过来得到 `frame`：

```
// The frame for the specified alignment rectangle.
- (CGRect)frameForAlignmentRect:(CGRect)alignmentRect;
```

考虑到每次重写这两个方法比较烦，系统也提供了一个简便方法，由 `inset` 来指定 `frame` 与 `alignment rect` 的关系：

```
// The insets from the view's frame that define its alignment rectangle.
- (UIEdgeInsets)alignmentRectInsets;
```

回到间隔问题，咱们可以将间隔当作上面提到的装饰，让 `UILabel` 的 `alignment rect` 比 `frame` 多个 `10 point` 间隔就好了：

```
@interface NLabel : UILabel
@end

@implementation NLabel
- (UIEdgeInsets)alignmentRectInsets {
    return UIEdgeInsetsMake(0, -10, 0, 0);
}
}
```

不过让人迷惑的是，在 iOS 中，`frameForAlignmentRect` 和 `alignmentRectForFrame`：重写之后，并没有起到预期的作用，OS X 中倒是正常。所以在 iOS 中，还是使用 `alignmentRectInsets` 的好。对于这个现象，还希望有了解的同学帮忙解释一下。

当然，每次都得要继承才能使用对齐矩形，毕竟不太方便，也许 **关联对象** 和 **method swizzled** 组合起来是个可行方案：

```
#import
@interface UIView (n_labelRectInsets)
@property (nonatomic, copy) UIEdgeInsets (^n_labelRectInsets)() UIEdgeInsets originInsets;
@end
...
```

3 总结

对齐矩形是个好玩意呢~

- 参考：
1. [Apple Developer](#)
 2. [Advanced Auto Layout Toolbox](#)
 3. [iOS Auto Layout Demystified](#)

文章分类 `iOS` 文章标签 `iOS` `Swift`

宇朋Look iOS Dev @ 昆仑万维
发布了 627 篇文章 · 获得点赞 25,265 · 获得阅读 957,404

安装掘金浏览器插件
多内容聚合浏览、多引擎快捷搜索、多工具便捷提效、多模式随心畅享，你想要的，这里都有！

输入评论（Enter换行，Shift+Enter发选）

发表评论

全部评论 (1)

tanpengsccc iOS · 2年前
好文，收藏

相关推荐

hejunm 3天前 | iOS
iOS AutoLayout还可以这样玩
网上关于Auto Layout的文章繁多，但是总觉得有些内容并没有说清楚。所以在这篇博文，我将通过一个demo...

可复来宝宝 13天前 | iOS · Xcode
Xcode13.0和iOS15.0适配
小知识，大挑战！本文正在参与“程序员必备小知识”创作活动 看了朋友的文章后，今天才升级xcode13，跑了...

早起年轻人 1年前 | Objective-C
iOS获取最顶层ViewController
UIApplication的核心作用是提供了iOS程序运行期间的控制和协作工作，每一个程序在运行期必须有且仅有一个...

season_zhu 15天前 | iOS · Xcode
酸了我的手写这篇文章：用Xcode编写适合自己的代码块
代码块是我写这篇文章的利器，本文从系统代码块出发，通过分析配置项，然后再来实自...

Study_Min 2月前 | 前端 · iOS
AutoLayout
一、自动布局的原理 1、Constraints Change 激活、失活 创建约束、优先级 添加、移除视图 检测到改变后系统...

edwardyk 29天前 | 前端 · iOS
iOS 15适配总结
iOS 15适配总结---tabBar及navigationBar的背景颜色问题及tableView的新属性

公众号iOS进阶 2月前 | 前端 · iOS
iOS导航条BarBottomItem与titleLabel的间距
1、原理 titleLabel的起点位置和尺寸依赖于leftBarButtonitem和rightBarButtonitem...

_GodIsCoder 2年前 | Swift
如何使用 AutoLayout 自动计算 ScrollView 的 contentSize
在日常开发中，令人看了就想头秃三连的代码，莫过于手动计算各种数字的坐标。如下图，看了真的是头发疯...

Singularity_Lee 4月前 | iOS
iOS 视频播放有声音无画面
项目中播放个别MP4视频文件播放时有声音但无画面的情况，使用 invector 导出视频文件获取相关信息，...

jerryfans 15天前 | iOS · Swift
一个优雅的Swift弹窗组件-JFPopup
一个优雅的Swift弹窗组件，让你能快速的弹出各种风格的自定义View，目前支持Drawer,BottomSheet,Dialog...

LEE1994107904 1年前 | Swift
iOS15优雅的拆分UIViewController与View
对于iOS开发而言 始终无法离开UIKit这个框架，加之SwiftUI并不成熟，所以懂懂的，而UIKit框架就是基于的MVC...

leeluanxin 1年前 | iOS
autolayout中的线性规划算法——simplex
auto layout是苹果公司提供的一个基于约束布局、动态计算视图大小和位置的库，并且已经集成到xcode开发环...

Tioks0 3年前 | iOS · 面试 · Swift
《iOS面试之道》算法基础学习(上)
道长和唐巧的面试之道,刚出来第一时间就入手了,也是趁着公司目前不是很忙,能好好静下心来细读这本书.笔者...

编程怪才_凌南画 11月前 | iOS
iOS布局进阶：你真的会用 autolayout 么？
前言：iOS系统已经迎来了10.3.2版本，iOS开发发展至今已相当成熟了。布局的方式从frame、size、center到...

season_zhu 14天前 | iOS · 产品
别关我小黑屋，聊一聊掘金iOS端App
本文主要从App的功能需求角度与使用角度谈掘金iOS端App，我也可能会顺带说说我...

G9YH 4年前 | Android
解决ScrollView嵌套RecyclerView的显示及滑动问题
项目中常常需要实现ScrollView中嵌入一个或多个RecyclerView，这一做法通常会导致页面滑动卡顿、...

奥报的小画家Sure 2年前 | iOS · Objective-C
自定义View都不好还做什么iOS开发
前言：对于接触业务开发的童鞋，自定义View的开发是进行最频繁的工作了。但发现一些童鞋还是没以一个...

没故事的卓同学 2年前 | iOS · Apple · Swift
理解 Swift 中的元类型：Type 与 self
元类型就是类型的类型。比如我们说 5 是 Int 类型，此时 5 是 Int 类型的一个值，但是如果我问 Int 类型占用多...

Ivym 1年前 | iOS
子类化 UIButton 支持 spacing 并完美兼容 AutoLayout
众所周知，UIButton 不支持设置图像和文字问题的。网上的文章大多数都是通过调节 titleLabelsets 和 ...

Weiyi17178 2年前 | iOS
详解iOS的presentViewController

稀土掘金浏览器插件——你的一站式工作台

多内容聚合浏览、多引擎快捷搜索、多工具便捷提效、多模式随心畅享，你想要的，这里都有。

安装浏览器插件

关于作者

宇朋Look
iOS Dev @ 昆仑万维

获得点赞 25,265

文章被阅读 957,404

掘金开发者大会

程序员必懂小知识

听雨落计划

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货

硬核干货