

# 3分钟实现iOS语言本地化/国际化（图文详解）



(w/93131dfba96a) + 关注

2016.06.20 18:14\* 字数 3601 阅读 34845 评论 79 喜欢 386 赞赏 4 阅读 34845 评论 79 喜欢 386 赞赏 4

## 前言

语言本地化，又叫做语言国际化。是指根据用户操作系统的语言设置，自动将应用程序的语言设置为和用户操作系统语言一致的语言。往往一些应用程序需要提供给多个国家的人群使用，或者一个国家有多种语言，这就要求应用程序所展示的文字、图片等信息，能够让讲不同语言的用户读懂、看懂。进而提出为同一个应用程序适配多种语言，也即是国际化。语言国际化之所以又叫做语言本地化，这是站在每个用户的角度而言的，是指能够让用户本地、本土人群能够看懂的语言信息，顾名思义，语言本地化。其实语言本地化 == 语言国际化！

本文将分如下7个主要章节一步一步讲解如何完全本地化一个App。

- 配置需要国际化的语言（国际化的准备工作）
- App名称本地化
- 代码中字符串本地化
- 多人开发情况下的字符串本地化
- 图片本地化（两种方式两种方式）
- 查看/切换本地语言
- storyboard/xib本地化

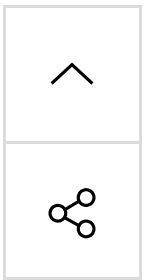
## 配置需要国际化的语言

配置需要国际化的语言，这也是国际化之前的准备工作，无论我们是国际化App名称、代码中的字符串、图片、还是storyboard和xib，都需要进行这一步的准备工作（一个项目中需要且仅需要配置一次）。

- 选中project->Info->Localizations，然后点击"+", 添加需要国际化/本地化的语言，如下图（默认需要勾选Use Base Internationalization）：

Snip20160616\_15.png

- 此处以添加法语为例，如下图：



Snip20160616\_18.png

---

- 弹出如下对话框，直接点击finish，如下图：

Snip20160616\_21.png

---

- 同理，添加简体中文、繁体中文、韩语，最终结果如下图：

备注：“zh-Hans”和“zh-Hant”是简体中文和繁体中文的缩写。这是标准的缩写。H可大写也可小写。“en”是英语的缩写。ko是韩语的缩写，fr是法语的缩写。其他语言请百度各国语言缩写即可查询。

## （一）应用名称本地化/国际化

应用名称本地化，是指同一个App的名称，在不同的语言环境下（也就是手机设备的语言设置）显示不同的名称。比如，微信在简体中文环境下App名称显示为“*微信*”，在英语环境下显示为“*weChat*”。下面就开始进行应用名称本地化。

1. 选中Info.plist，按下键盘上的command + N，选择**Strings File**（iOS->Resource->Strings File）

- 文件名字命名为**InfoPlist**，且必须是这个名字（因每个人电脑设置差异，此处本人电脑没有显示strings后缀名）：

Snip20160616\_11.png

---

- 点击create后，Xcode左侧导航列表就会出现名为InfoPlist.strings的文件，如下图：

Snip20160616\_13.png

---

- 选中InfoPlist.strings，在Xcode的File inspection（Xcode右侧文件检查器）中点击Localize，目的是选择我们需要本地化的语言，如下图：  
注意：在点击Localize之前，一定要保证我们已经添加了需要本地化的语言，也就是上面配置需要国际化的语言那一步（步骤：project->Info->Localizations，然后点击"+", 添加需要国际化/本地化的语言）。

Snip20160616\_14.png

---

- 点击Localize后，会弹出一个对话框，展开对话框列表，发现下拉列表所展示的语言正是我们在上面配置的需要国际化的语言，选择我们需要本地化的语言，然后点击对话框的Localize按钮，如下图：

Snip20160617\_10.png

---

Snip20160617\_12.png

---

注意：如果我们没有在 PROJECT 中配置需要国际化的语言（project->Info->Localizations，然后点击"+"），上图下拉列表中将只会出现"Base"和"English"选项，English语言是系统默认的语言，其他需要国际化的语言（例如中文简体、法语）必须通过上面的配置本地化语言那一步手动添加。

- 然后我们发现Xcode右侧的File inspection变成了下图的样式：

Snip20160617\_13.png

---

- 接下来，勾选French、Chinese（zh-Hans）、Chinese（zh-Hant）、Korean，如下图：

Snip20160617\_14.png

---

- 此时，Xcode左侧的InfoPlist.strings左侧多了一个箭头，点击箭头可以展开，如下图所示：

Snip20160617\_15.png

---

从上图可以看出，InfoPlist.strings文件下包含了English、French、Chinese（Simplified）、Chinese（Traditional）、Korean这五种语言的文件。  
原理：程序启动时，会根据操作系统设置的语言，自动加载InfoPlist.strings文件下对应的语言文件，然后显示应用程序的名字。

- 接下来，我们分别用不同的语言给InfoPlist.strings下的文件设置对应的名字。

(1) 在InfoPlist.strings(english)中加入如下代码：

```
// Localizable App Name是App在英语环境环境下显示的名称
CFBundleDisplayName = "Localizable App Name";
```

备注：CFBundleDisplayName可以使用双引号，也可以不使用双引号！

Snip20160617\_21.png

---

(2) 在InfoPlist.strings(French)中加入如下代码：

```
CFBundleDisplayName = "Le nom de la localisation de l'App";
```

Snip20160617\_27.png

(3) 在InfoPlist.strings(Chinese(Simplified))中加入如下代码：

```
CFBundleDisplayName = "国际化App名称";
```

Snip20160617\_32.png

(4) 在InfoPlist.strings(Chinese(Traditional))中加入如下代码：

```
CFBundleDisplayName = "國際化App名稱";
```

Snip20160617\_30.png

(5) 在InfoPlist.strings(Korean)中加入如下代码：

```
CFBundleDisplayName = "현지화 앱 명칭";
```

修改模拟器语言环境为English。App名称如下图：

修改模拟器语言环境为Chinese(Simplified)。App名称如下图：



Snip20160618\_4.png

---

修改模拟器语言环境为Chinese(Traditional)。App名称如下图：

Snip20160618\_5.png

---

修改模拟器语言环境为Franch。App名称如下图：

Snip20160618\_6.png

---

修改模拟器语言环境为Korean。App名称如下图：

Snip20160618\_7.png

备注：过去本地化App名称，需要在Info.plist文件中增加一个名为“Application has localized display name”的BOOL类型的Key，并且需要将其值设置为YES（如下图）。目的是让App支持本地化App名称。但现在可以忽略这一步。

Snip20160618\_8.png

至此，本地化App名称已经演示完毕，其步骤就是：

- 在Project的设置中通过点击"+"添加需要本地化的语言。
- 然后在Xcode右侧的File inspection中点击Localize，选中需要本地化App名称的语言。
- 最后在每个语言对应的文件中以key = value(CFBundleDisplayName = "App名称");的形式设置App的名称。

## （二）代码中字符串的本地化

所谓字符串本地化，就是指App内的字符串在不同的语言环境下显示不同的内容。比如，“主页”这个字符串在中文语言环境下显示“主页”，在英语环境下显示“home”。下面就开始进行字符串本地化。

其实字符串本地化和App名称本地化过程如出一辙，只是创建的文件名成不一样（连同后缀一起，文件名必须是Localizable.strings），其他步骤完全相同。为了能够让大家彻底了解，此处还是会把步骤一一贴出来。

- 和应用名称本地化一样，首先需要command + N，选择iOS -> Resource -> Strings File

- 文件名必须命名为**Localizable**

备注：因本人电脑取消隐藏文件后缀名，所以会自动补全.strings后缀名。

Snip20160619\_8.png

---

- 文件创建成功，查看Xcode左侧导航列表，发现多了一个名为Localizable.strings的文件，如下图：

Snip20160619\_4.png

---

- 选中Localizable.strings文件，在Xcode的File inspection中点击Localize，目的是选择我们需要本地化的语言（和本地化App名称同理），如下图：

- 依次选择English->Localize，如下图：

- 然后我们发现Xcode右侧的File inspection变成了下图的样式：

- 然后勾选French、Chinese (zh-Hans) 、Chinese (zh-Hant) 、Korean，如下图：

- 此时，Xcode左侧的Localizable.strings左侧多了一个箭头，展开后，如下图所示：

Snip20160619\_6.png

---

- 然后我们只需要在Localizable.strings下对应的文件中，分别以Key-Value的形式，为代码中每一个需要本地化的字符串赋值，如下图：

Snip20160619\_9.png

---

Snip20160619\_10.png

---

Snip20160619\_12.png



Snip20160619\_13.png



Snip20160619\_14.png

- 本地化代码中的字符串，如下图：



- 我们只需要使用Foundation框架自带的NSLocalizedString(key, comment)这个宏根据Key获取对应的字符串，然后赋值给代码中的字符串。

```
// NSLocalizedString(key, comment) 本质
// NSLocalizedString 第一个参数是内容,根据第一个参数去对应语言的文件中取对应的字符串, 第二个参数将会转化为字符串文件里的注释, 可以传nil, 也可以传空字符串@""。
#define NSLocalizedString(key, comment) [[NSBundle mainBundle] localizedStringForKey:(key) value:@"" table:nil]
```

- 不同语言环境下运行效果，如下图：
- 英语环境下：



- 法语环境下：



Snip20160619\_18.png

---

- 简体中文环境下：



Snip20160619\_19.png

---

- 繁体中文环境下：



Snip20160619\_20.png

---

- 韩语环境下：

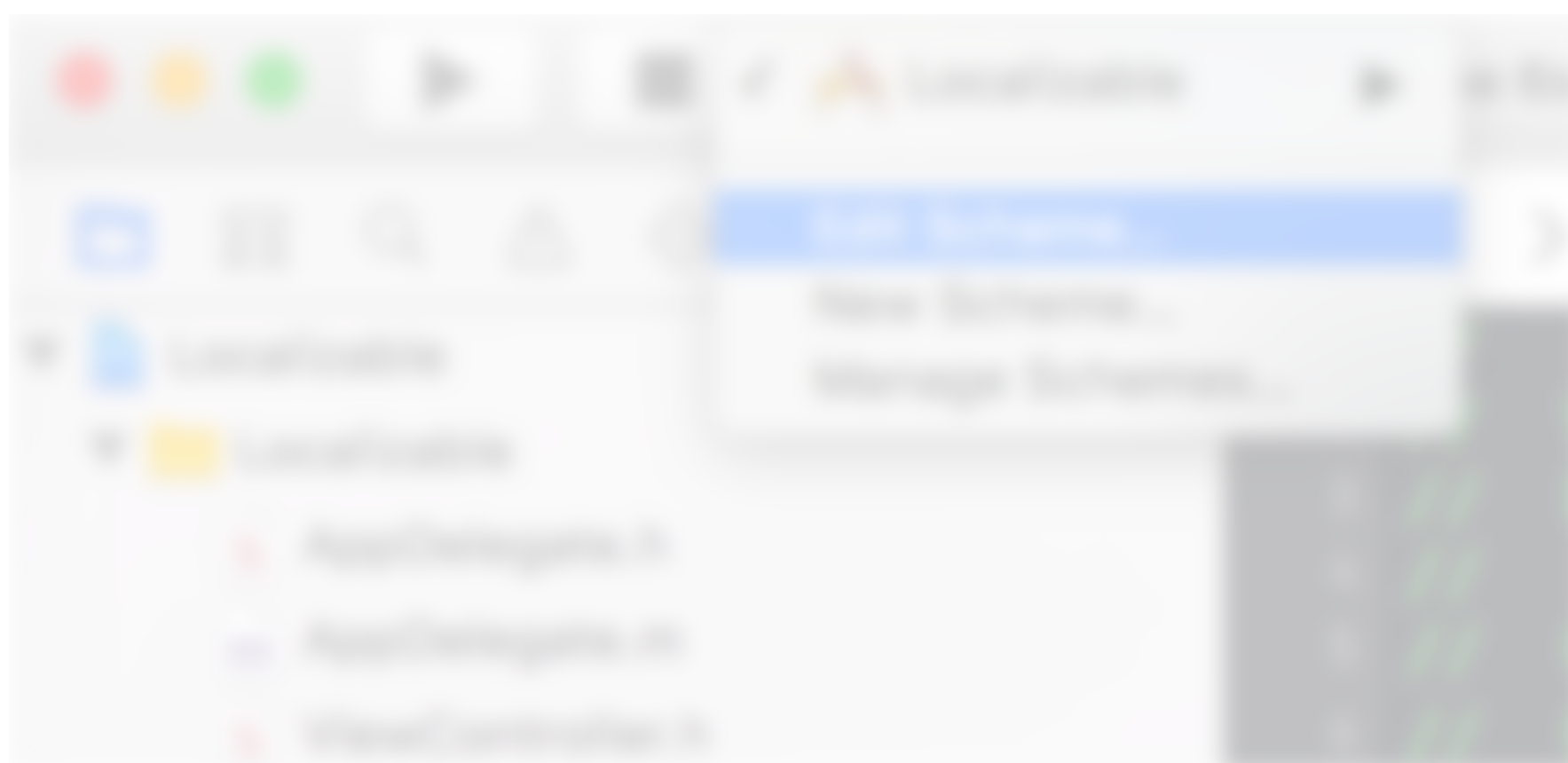


Snip20160619\_23.png

如此一来，我们就实现了代码中字符串的本地化。

## 技巧

- 其实，我们不需要给Localizable.strings(English)文件添加Key-Value。原因如下：系统根据某个key去获取对应的字符串时，如果没有找到，那么就会以key作为value返回。所以如果我们在Localizable.strings(English)文件中没有添加 `click = "hit";`那么在english环境下btn最终显示的title会是click。
- 切换语言无需在模拟器中设置，只需要在Xcode中进行如下设置：Edit->Scheme->Run->Arguments Passed On Launch ->-AppleLanguages (语言代码)。例如，我们模拟器此时虽然是韩语，如果通过上述步骤设置为zh-Hans，那么语言环境将会变为汉语，如下图：



Snip20160619\_26.png



Snip20160619\_28.png

- 运行效果：



Snip20160619\_29.png

- 如此一来，切换语言变得更加简单，无需在模拟器的设置中进行繁琐的语言切换。

### （三）多人开发情况下的字符串本地化

项目开发中，独立开发的还是少数。经常会有多人开发的情况，这种情况，如果多人同时操作本地化文件，极有可能会存在冲突。另一方面，我们又不希望自己的本地化文件受到对方的污染，也就是说，我们不希望对方操作我们的本地化文件。但是上面介绍的代码中字符串的本地化使用的是默认的文件名"Localizable",因为启动程序时，系统将根据语言加载相应的文件得到其对应的字符串文件，这个字符串可以通过系统将NSLocalizedString中的宏生成名为“Localizable.strings”的文件。那么如何让系统加载我们自己命名的本地化文件而非系统默认的Localizable.strings呢？这就是

NSStringFromClassTable(<#key#>, <#tbl#>, <#comment#>)的用处。也就是说，如果你的strings文件名字不是Localizable而是自定义的话，如VVS.strings，那么你就得使用NSStringFromClassTable这个宏来读取本地化字符串。

```
// key: 系统根据key取字符串
// tbl: 自定义strings文件的名字
// comment: 可以不传
    NSLocalizedStringFromTable(<#key#>, <#tbl#>, <#comment#>)
```

Snip20160619\_32.png

```
#import "ViewController.h"

@interface ViewController ()
@property (weak, nonatomic) IBOutlet UIButton *btn;

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    NSString *title = NSLocalizedStringFromTable(@"click", @"VVS", nil);
    [self.btn setTitle:title forState:UIControlStateNormal];
}

@end
```

- 运行效果，如下图：

Snip20160619\_33.png

如此一来，我们就可以挣脱别人的strings文件和系统默认的Localizable.strings文件，自己另起炉灶。

## （四）图片本地化

本地化图片，有两种方式，第一种方式和本地化代码中的字符串一样，通过 `NSLocalizedString(key,comment)` 来获取相应的字符串，然后根据这个字符串再获取图片。

### 方式一

```
NSString *imageName = NSLocalizedString(@"icon", nil);
UIImage *image = [UIImage imageNamed:imageName];
self.imageView.image = image;
```

### 方式二

首先需要添加需要本地化的语言，具体步骤参考第一章配置需要国际化的语言。因为我演示的demo中在本地化App名称时已经添加了需要国际化的语言。所以不需要再设置。

- 将图片拖入工程中，例如“icon.png”，然后选中icon.png，展开Xcode右侧的file Inspection，点击Localize，如下图：

Snip20160619\_36.png

- 然后，右击icon.png->show in Finder，我们发现在en.lproj文件中多了一个名为icon.png的图片，如下图：

Snip20160619\_38.png

- 当然，zh-Hans.lproj文件夹下并没有图片，如下图：

Snip20160619\_39.png

- 我们只需给zh-Hans.lproj添加一个名字也为icon.png的图片。如下图：



Snip20160619\_40.png

---

- 然后把zh-Hans.lproj中的icon.png拖到Xcode中，如下图：

Snip20160619\_43.png

---

- 然后发现，icon.png左边出现了一个可以展开的三角形。如下图：

Snip20160619\_44.png

---

- 展开后发现里面包含两张图片，如下图：

Snip20160619\_45.png

---

然后控制器中添加如下代码：

```
#import "ViewController.h"

@interface ViewController ()
@property (weak, nonatomic) IBOutlet UIImageView *imageView;

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    NSString *imageName = NSLocalizedString(@"icon", nil);
    UIImage *image = [UIImage imageNamed:imageName];
    self.imageView.image = image;
}

@end
```

- english环境下运行效果如下：

Snip20160620\_46.png

---

- 中文环境改下运行效果如下：

Snip20160620\_47.png

- 其他语言环境，例如法语、中文繁体、韩语，操作步骤完全和上面相同。都是在对应的.lproj文件夹下添加同名的图片，然后把图片拖放到Xcode中，不再赘述。

## （五）查看/切换本地语言

原理：应用启动时，首先会读取NSUserDefaults中的key为AppleLanguages对应的value，该value是一个String数组，也就是说，我们访问这个名为AppleLanguages的key可以返回一个string数组，该数组存储着APP支持的语言列表，数组的第一项为APP当前默认的语言。

```
#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    NSArray *languages = [[NSUserDefaults standardUserDefaults] valueForKey:@"AppleLanguages"];
    NSString *currentLanguage = languages.firstObject;
    NSLog(@"模拟器当前语言: %@", currentLanguage);
}

@end
```

-控制台打印结果：

同理，既然我们可以通过AppleLanguages这个key从NSUserDefaults中取出语言数组，那么我们也可以给AppleLanguages这个key赋值来达到切换本地语言的效果，从此以后，我们就无需频繁的去模拟器的设置->通用->语言与地区 中切换语言。

```
#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    // 切换语言前
    NSArray *langArr1 = [[NSUserDefaults standardUserDefaults] valueForKey:@"AppleLanguages"];
    NSString *language1 = langArr1.firstObject;
    NSLog(@"模拟器语言切换之前: %@", language1);

    // 切换语言
    NSArray *langs = @[@"en"];
    [[NSUserDefaults standardUserDefaults] setObject:langs forKey:@"AppleLanguages"];

    // 切换语言后
    NSArray *langArr2 = [[NSUserDefaults standardUserDefaults] valueForKey:@"AppleLanguages"];
    NSString *language2 = langArr2.firstObject;
    NSLog(@"模拟器语言切换之后: %@", language2);
}

@end
```

- 回想一下，我在上面的技巧一节中提过：切换语言无需在模拟器中设置，只需要在Xcode中进行如下设置：Edit->Scheme->Run->Arguments Passed On Launch ->-AppleLanguages (语言代码)。其实本质上就是给NSUserDefaults中名为AppleLanguages的key赋值。

未完待续...

文 / VV木公子 (简书作者)

**PS:**如非特别说明，所有文章均为原创作品，著作权归作者所有，转载转载请联系作者获得授权，并注明出处，所有打赏均归本人所有！

如果您是iOS开发者，或者对本篇文章感兴趣，请关注本人，后续会更新更多相关文章！敬请期待！

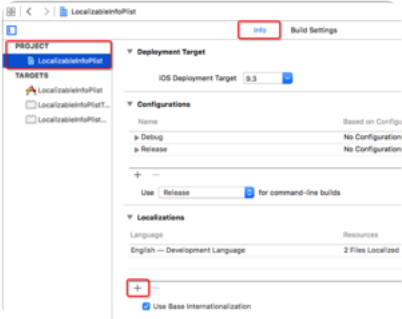
小礼物走一走，来简书关注我

赞赏支持



utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/1c3d69c75137?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

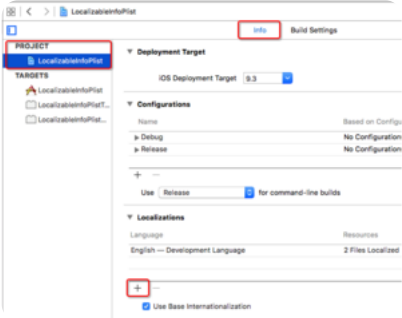
### 【转】3分钟实现iOS语言本地化/国际化（图文详解）（/p/1c3d69c75137...

前言 语言本地化，又叫做语言国际化。是指根据用户操作系统的语言设置，自动将应用程序的语言设置为和用户操作系统语言一致的语言。往往一些应用程序需要提供给多个国家的人群使用，或者一个国家有多

 一只不会飞的小艳子 (/u/3c3fd109eb28?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/8770d67268ce?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

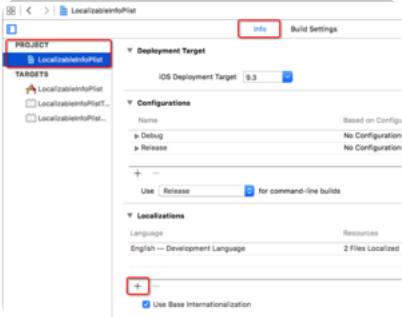
### iOS语言本地化/国际化（图文详解）（/p/8770d67268ce?utm\_campaign=...

3分钟实现iOS语言本地化/国际化(图文详解)前言语言本地化，又叫做语言国际化。是指根据用户操作系统的语言设置，自动将应用程序的语言设置为和用户操作系统语言一致的语言。往往一些应用程序需要提供

 CoderDwang (/u/75afc06dcc73?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/d7f2aabc9194?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

### iOS 应用的国际化与本地化 (/p/d7f2aabc9194?utm\_campaign=maleski...


前言 语言本地化，又叫做语言国际化。是指根据用户操作系统的语言设置，自动将应用程序的语言设置为和用户操作系统语言一致的语言。往往一些应用程序需要提供给多个国家的人群使用，或者一个国家有多

 卡尔特斯 (/u/887d1fc86fe6?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

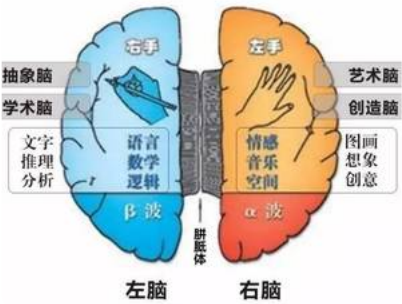
### 3分钟实现iOS语言本地化/国际化（图文详解）（/p/ed56b921edd7?utm\_...

iOS语言本地化/国际化

 Silence\_广 (/u/6313f5c2a6cd?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/5e32be8226b6?



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

### 像最强大脑一样记忆（3）（/p/5e32be8226b6?utm\_campaign=maleski...

今天我们来介绍按顺序记住一系列东西的方法。顺序，就涉及序号。最常用的是数字序号和字母序号。我们常见的按姓氏笔画排列和按音序排列，说到底其实就是数字序号和字母序号，前者是姓氏笔画数的数

 楚人长铗 (/u/e946c5e327ef?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)



林宥嘉：余生和你一起浪费，这就是最好的晚安。 (/p/c5d5f7b4d193?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

1. 早上还在睡觉，好友Z给我打来电话激动的说：林宥嘉，林、、、宥、、、嘉，在微博上求婚了。我顿时瞬间睡意全无，赶紧拿出手机打开微博。对，这无疑肯定就是微博热搜的话题。 第一次知道林宥嘉这个

 zzy毅哥 (/u/4c5fbcb17c27?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/f6b69e119bab?utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)  
寂寞支付宝：如果我不这样做，我可能会死 (/p/f6b69e119bab?utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

「我们用女白领和女大学生来打第一波UGC」 「嫖客的资质怎么控制呢？」 「用总资产？用流水？」  
「感觉维度有点单一，不如用芝麻信用分吧」 支付宝的PM围坐在一起商定好策略之后，决定把他们的新功

 季源 (/u/07e6388ccd62?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/9c69989cdc5e?utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)  
花开在眼前，每日分享 (/p/9c69989cdc5e?utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

2017年2月28日 今天向大家分享：“虽然”换框法。 在任何一句含有“但是”的话语当中，强迫自己简单的用“虽然”代替“但是”，就会起到神奇的效果。 例如： 1.我找到了问题的解决办法，但是问题以后可能还会

 无限爸爸 (/u/80a3db50034f?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/e76fa02de5ad?utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)  
今天下午，一张快递小哥的照片，看哭了十万快递人..... (/p/e76fa02de5ad?utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

据台媒报道，前天（3月12日）台湾一快递员，因送货迟到了两个小时，遭到收货方的拒签，最后被逼下跪道歉，精神崩溃，痛哭不止。当我第一眼瞥到这张照片的时候，我的心是紧紧揪在一起的。 据报道，这位

 陈大贤 (/u/fc3a1fd2d192?utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)