

Coding Style & Project Setting

2016-10 明林清

- Coding Style
- Project Setting

Coding Style

Coding Style

- 便于交流
- 长期可维护

Coding Style

- 使用主流风格
- 与团队风格保持一致
- 尊重作者
- 据为己有

Coding Style

- 布局：缩进、空行、空格 ...
- 命名：名称、缩写 ...
- 注释
- 标准用法

Apple Style

布局

缩进

- XCode 自动缩进
- 4 个空格宽度

空行

- 适当位置空一行
- 最多空一行
- 文件末尾空一行

空格

- 适当位置空格，一般一个空一格
- 平衡的符号两边空格、或两边不空格

BOOL c = a && b;

- 不平衡的符号疏远的一边空格

BOOL a = x, b = !y;

空格

- 星号前空格、后不空格
NSString *headLine
- 属性的 @property 后空格
@property (nonatomic) BOOL ok;
- 方法的 +/- 后空格
- (void)makeSubviews;

特例

- . -> 两边不空格
it.delegate = self;
self->ivar = value;
- : 两边空格、两边不空格
@interface BJUser : NSObject
[self updateUser:user];
- ...

符号

- 与代码的关系体现在空格和空行上
- 需要换行时符号放在行首，便于看清关系

```
    BOOL isLive = (cur >= start  
                    && cur < end);  
    int result = ([self valueA]  
                  + [self valueB]  
                  - [self valueC]);
```

括号

- () [] 两边不换行
- { 后换行、} 前换行
 - (void)method {
 if (boolean) {
 }
}

括号

- 错误：非 Objective-C 风格

```
if (user.isHappy)
{
}
else
{
}
```


括号

- 错误

```
- (void)method;  
{  
    if (boolean) {  
    }  
}
```

括号

- 错误: else 与 if 代码混成一排、并且对 else 的注释只能写在 else 之后

```
if (user.isHappy) {  
}_else { // about else  
}
```

括号

- 允许

```
if (callback) callback();
```

- 禁止

```
if (callback)  
    callback();
```

- 禁止

```
if (callback) callback();  
else {  
    [delegate callback];  
}
```

命名

命名

- nameLabel:
变量名、属性名、方法名
- nameLabelFontSize (不推荐) :
私有, 常量、枚举、结构体等
- BJNameLabel:
全局, Class、Protocol、Category、常量、枚举、结构体等

命名

- `_BJLoginButton`:
成员变量、内部 Class
- `bj_doubleForKey`:
非项目内部类的 Category 方法、属性, 如
[dictionary bj_doubleForKey:]
- `BJ_USER_LOGIN_URL`、`user_name`:
禁止
- `kBJUserName`:
禁止

缩写

- 允许使用通用多个单词缩写
如 HTML、URL、HTTP、GSX
- 允许过长的单个单词缩写
如 i18n
- 禁止不必要的单个单词缩写
如 btn(button)、lbl(label)、tpl(template)
- 单字母变量只用于循环计数

is

- BOOL 类型属性的名称是形容词可省略 is，用于避免 setter 方法名为 setIsEditable:，但应 getter 方法名要保留 is

```
@property (getter=isEditable) BOOL  
editable;
```

```
// @property (setter=setEditable:)  
BOOL isEditable;
```


变量类型

- NSString *title vs ~~NSString *titleString~~
- NSString *titleHTML ~~NSString *title~~
- NSAttributedString *titleAttributedString vs ~~NSAttributedString *title~~
- NSDate *now vs ~~NSDate *nowDate~~
- NSTimeInterval nowTimeInterval vs ~~NSTimeInterval now~~
- NSDate *lastModifiedDate vs ~~NSDate *lastModified~~
- NSURL *URL, NSString *URLString vs ~~NSString *URL~~
- NSString *releaseDateString vs ~~NSString *releaseDate~~

标准用法

Variable Qualifiers

```
NSString * __weak text = nil;
```

```
__weak NSString *text = nil;
```

Enum Types

```
typedef NS_ENUM(NSInteger, BJState) {  
    NYTStateInactive,  
    NYTStateLoading  
};
```

```
typedef enum {  
    NYTStateInactive,  
    NYTStateLoading  
} NYTState;
```

Bitmasks

```
typedef NS_OPTIONS(NSUInteger, BJMask) {  
    BJStateMaskA = 1 << 0,  
    BJStateMaskB = 1 << 1  
};
```

Constants

// 不要使用宏代替常量

```
static const CGFloat BJValueA = 1.0;
```

```
static NSString * const BJKey = @"";
```

Constants

```
const CGFloat BJValueA = 1.0,      BJValueB = 2.0;  
NSString      * const BJKeyA = @"" , * const BJKeyB = @"";
```

Literals

```
NSNumber *shouldUseLiterals = @YES;  
NSNumber *buildingZipCode = @10018;  
NSArray *names = @[@"Brian",  
                  @"Matt"];  
NSDictionary *owners = @{@"iPhone": @"Kate",  
                          @"iPad":   @"Kamal"};
```


CGRect Functions

```
CGFloat x = CGRectGetMinX(frame);  
CGFloat y = CGRectGetMinY(frame);  
CGFloat width = CGRectGetWidth(frame);  
CGFloat height = CGRectGetHeight(frame);
```

```
CGFloat x = frame.origin.x;  
CGFloat y = frame.origin.y;  
CGFloat width = frame.size.width;  
CGFloat height = frame.size.height;
```

Imports

```
// header
#import "BJManagerRoomViewController.h"
// categories
#import "BJManagerRoomViewController+protected.h"

// frameworks
#import <BJHL-Foundation-iOS/BJHL-Foundation-iOS.h>
#import <YYModel/YYModel.h>

// models
#import "BJRoom.h"

// views
#import "BJManagerRoomViewCell.h"
```

Private Properties

```
@interface NYTAdvertisement ()
```

```
// 尽量避免使用成员变量
```

```
@property (nonatomic) UIWebView *adWebView;
```

```
@end
```

Protocols

@optional

– (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView;

@required

– (NSInteger)tableView:(UITableView *)tableView
 numberOfRowsInSection:(NSInteger)section;

~~– (NSInteger)numberOfSectionsInTableView;~~

~~– (NSInteger)numberOfRowsInSection:(NSInteger)section;~~

if

```
if (!someObject)  
if (someObject == nil)  
if (isAwesome)  
if (!someNumber.boolValue)  
if (someNumber.boolValue == NO)
```

```
if (nil == someObject)  
if (isAwesome == YES)
```

error

```
NSError *error;
```

```
if (![self trySomethingWithError:&error]) {  
    // Handle Error  
}
```

```
[self trySomethingWithError:&error];  
if (error) {  
    // Handle Error  
}
```

注释

- 文档注释
- TODO 注释
- 解释性注释

文档注释

```
/**  
 * Apple Style  
 */
```

```
/**  
 * Common Style  
 */
```


TODO 注释

// TODO: NOT ~~TODO~~, ~~todo~~, ~~to-do~~, ~~ToDo~~

// ???: not sure

// !!!: important

TODO:、????: 警告脚本

```
// Project > Target > Build Phases > New Run Script Phase
# http://rocry.com/2012/12/17/xcode-tips/
# NO '|\\!\\!\\!:|WARN:|WARNING:'
KEYWORDS="TODO:|FIXME:|\\?\\?\\?:"
find ${SRCROOT}/../your-dir \( -name "*.h" -or -name "*.m" \) -print0 | \
xargs -0 egrep --with-filename --line-number --only-matching "($KEYWORDS).*\\$" | \
perl -p -e "s/($KEYWORDS)/ warning: \\$1/"
```

解释性注释

- 尽量让代码本身就能说明它在做什么
- 仅用于可能引发疑问的代码、解释原因

代码分隔

- 空行

- #pragma

```
#pragma mark -  
#pragma mark - life cycle  
#pragma mark - subviews  
#pragma mark labels  
#pragma mark buttons  
#pragma mark - <ProtocolA>  
#pragma mark - <ProtocolB>
```

Project Setting

Project

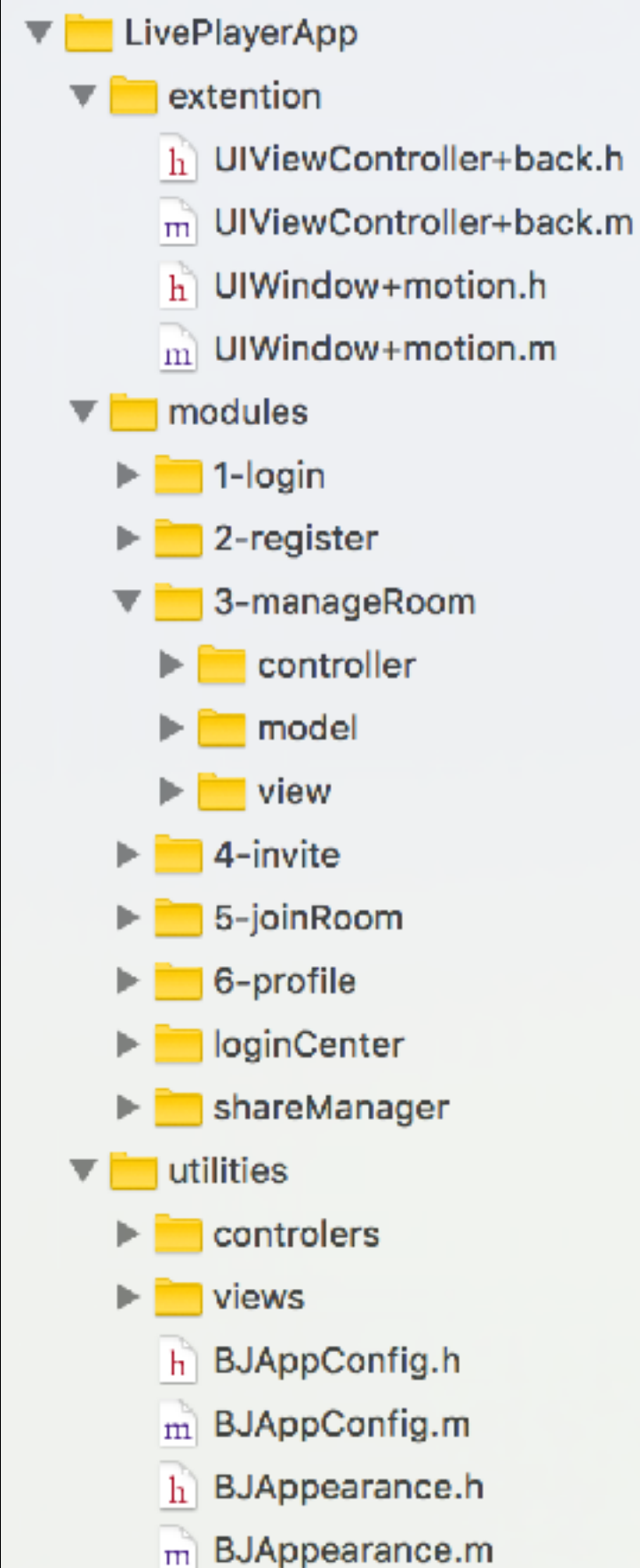
- Automatically manage signing
- Organization & Class Prefix
- Indent Using Space, Tab Widths 4, Indent Widths 4, NOT Wrap lines

Project

- 开启 Treat Warnings as Errors
- 必须忽略警告时使用 Clang 的 pragma
- 项目目录、文件与物理目录、文件一一对应

目录结构

- 架构
- 模块
- 类



NYTimes objective-c-style-guide

<https://github.com/NYTimes/objective-c-style-guide>

git

<http://git.baijiahulian.com/iOS/training>

the end