

区别

类别(Category)

可以添加【方法】，不能添加【属性】和【成员变量】

形式

头文件

Person+eat.h

```
#import "Person.h"

@interface Person (eat)

- (void)eat;

@end
```

实现文件

Person+eat.m

```
#import "Person+eat.h"

@implementation Person (eat)

- (void)eat
{
    NSLog(@"我想吃饭了，好饿 0_0");
}

@end
```

特别需求

添加【属性】

形式

头文件

Person+eat.h

```
#import "Person.h"

@interface Person (eat)

@property (nonatomic, strong) NSString *name;
- (void)eat;

@end
```

实现文件

Person+eat.m

```
#import "Person+eat.h"

@implementation Person (eat)

- (void)eat
{
    NSLog(@"我想吃饭了，好饿 0_0");
}

/** 属性的get方法 */
- (NSString *)name
{
    return objc_getAssociatedObject(self, _cmd);
}

/** 属性的set方法 */
- (void)setName:(NSString *)name
{
    objc_setAssociatedObject(self, @selector(name), name,
    OBJC_ASSOCIATION_RETAIN);
}

@end
```

应用场景

1 对框架提供类的扩展(没有源码,不能修改);

2 不想生成一个新的子类的情况下,例如对NSArray的扩展;

3 方便做项目管理，可以将一份源码在多个地方共享或者做方法版本管理、多人协作开发、用本地版本替换公共版本实现。

注意事项

1 Category的方法不一定非要在@implementation中实现,也可以在其他位置中实现。但是当调用Category的方法时,依据继承树没有找到该方法的实现,程序则会崩溃;并且,Category的方法优先级较高,可以覆盖原类的方法;

2 在实际开发中要注意的是,Category 方法可能会覆盖于同一个类class 的其它 category 中的方法。但也可能被覆盖，因为不法预知他们的加载优先顺序，出现这种情况通常会在编译时出错。如果在一个开发的SDK中使用了类别，就最好保证类别名不同于使用者的类别名以及类别方法也不同于使用者的类别方法名，通常通过加前缀来做到。

3 Category的方法中,不可以调用super方法;

4 Category理论上不能添加【成员变量】,但是可以使用@dynamic(即运行时Runtime)来弥补这种不足。

拓展(Extension)

可以添加【方法】，可以添加【属性】和【成员变量】

形式

头文件

Student.h

```
#import "Person.h"

@interface Person (eat)

@end
```

实现文件

Student.m

```
#import "Student.h"

@interface Person (eat)

- (void)eat;

@end

@implementation Person (eat)

- (void)eat
{
    NSLog(@"放学了，好饿 0_0");
}

@end
```

应用场景

1 能为某个类添加【成员变量】，【属性】，【方法】；

2 当需要声明一个属性,它对外是只读的,但是在内部是可以修改的,这时可以通过Extension来实现;

3 extension一般用来隐藏类的私有信息，你必须有一个类的【源码】才能为一个类添加extension，所以你无法为系统的类比如NSString添加extension。

注意事项

1 声明的方法必须在@implementation中实现，不然编译器会报warning;

2 仅仅形式上，可以看做【匿名分类】：因为写法上就是【分类】的写法，只是()里面没有标识【分类名】

3 extension看起来很像一个匿名的category，但是extension和有名字的category几乎完全是两个东西。

小结

1 Category的小括号中有名字,而Extension没有;

2 Category只能扩充方法,不能扩充成员变量和属性;

3 如果Category的接口里声明了一个属性,那么Category只会生成这个属性的set,get方法的声明,也就不是会实现。

4 到底能不能在 category 加【成员变量】呢?结果是不可以! 除非用 runtime的Associated Objects添加【属性】

5 extension在【编译期决议】，它就是类的一部分，在编译期和头文件里的@interface以及实现文件里的@implementation一起形成一个完整的类，它伴随类的产生而产生，亦随之一起消亡。但是category则完全不一样，它是在【运行期决议】的。

6 extension可以添加【实例变量】，而category是无法添加【实例变量】的（因为在运行期，对象的内存布局已经确定，如果添加【实例变量】就会破坏类的内部布局，这对编译型语言来说是灾难性的）。

实际应用

Category

我们通常要封装一些公共方法的时候我们可以考虑使用Category的方式。

Extension

如果我们想在原始类上面增加一些不公开的方法、属性（私有方法、属性）时，可以新建一个Extension来解决问题。