

# Property - By Dorayo

属性格式: `@property(参数1, 参数2) 类型 名字;`

`@property`: 新的编译器指令, 主要功能是为我们自动生成setter和getter方法的声明  
`@synthesize`: 与@property相对应的新的编译器功能。主要是为我们自动生成setter和getter方法的功能实现  
属性都是基于变量的, 编译器会自动为属性创建对应的实例变量

说明: Xcode4.5之后, 直接@property就可以生成声明与实现

## 点语法

“

使用了属性后, 我们可以在需要使用setter和getter方法的地方使用点语法  
点语法只是调用setter和getter方法的便捷方式而已, 无其他功能  
点语法出现在赋值符号(=)的左边表示调用setter方法, 右边表示调用getter方法

Demo

## 名称的使用

格式: `@synthesize name = variable;`

eg: .h中 `@property (copy) NSString *name;`  
.m中 `@synthesize name = appellation;` //appellation是成员变量

## 属性的参数

属性的参数主要分三类:

- 读写属性 (`readwrite/readonly`)
- setter语意 (`assign/retain/copy`)
- 原子性 (`atomic/nonatomic`)

## 所有者属性

属性值	关键字	所有权
strong	__strong	有
weak	__weak	无
unsafe_unretained	__unsafe_unretained	无
copy	__strong	有
assign	__unsafe_unretained	无
retain	__strong	有

strong是iOS5以后，retain是iOS4之前，都表示持有对象，没区别

weak比unsafe\_unretained多了对象销毁后指针置nil，表示不持有对象

assign用于基本数据类型

## 只读属性

可以修改属性参数中的读写属性值为readonly

这样将只生成一个getter方法而不会生成setter方法，但如果代码中调用了其setter方法，编译器会报错

## 重命名getter和setter方法

格式：@property (getter=isHidden) BOOL hidden;