

Objective-C Memory Management (上課建議簡單整理版)

只整理建議方案，所以不列來龍去脈，來龍去脈在完成版在寫。基本的概念是自己透過alloc/init產生的物件，要自己處理記憶體回收。
一般使用狀況而言，以下三個狀況建議寫法是

1. Local variable

在function/method或block內透過alloc/init產生的物件建議寫法是寫了alloc/init之後，馬上在return前補上release。

Example:

我們有一個Song類別，在某個function使用的時候，我們透過alloc/init建立物件，在function結束之前就要釋放記憶體。

```
int test1() {  
    // 產生物件  
    Song *s1 = [[Song alloc] init];  
  
    [s1 setName: @"It's my life"];  
  
    [s1 play];  
  
    // 釋放記憶體  
    [s1 release];  
    return 0;  
}
```

2. Class member attribute

假設我們有一個Song的類別，裡面有一個attribute放的是Singer的物件。如果在下列程式碼使用狀況

```
int test2() {  
    Singer *singer = [[Singer alloc] init];  
    Song *song = [[Song alloc] init];  
  
    [song setSinger: singer];  
  
    [singer release];  
    // 如果只是一般的設定物件到song物件裡，在此時有可能singer  
    // 物件已經被release掉，所以下面這行有可能會出錯。  
    NSLog( @"Singer: %@", [song singer] );  
  
    [song release];  
    return 0;  
}
```

為了避免這個問題，Song類別的singer attribute的setter/getter要這樣寫

Song.h

```
#import <Foundation/Foundation.h>  
#import "Singer.h"
```

```
@interface Song : NSObject {  
    Singer* _singer;  
}  
  
@property (copy) NSString* name;  
@property (assign) CGFloat length;  
  
-(void)setSinger:(Singer *) singer;
```

```

-(Singer *)singer;

-(void)play;

@end

Song.m
#import "Song.h"

@implementation Song

-(void)setSinger:(Singer *)singer {
    [_singer release];
    _singer = singer;
    [_singer retain];
}

-(void)play {
    NSLog( @"Song %@ is playing", [self name] );
}

- (void)dealloc
{
    NSLog( @"Song %@ is released ...", self );
    // Song dealloc時也要釋放singer物件記憶體
    [_singer release];
    [super dealloc];
}

@end

```

兩個關鍵點

- a. 在setSinger時，要先呼叫之前singer object的release，設定之後再呼叫新的singer物件的retain
- b. 在Song類別的dealloc方法中要把singer釋放掉

3. Function return value

我們會有一些function會回傳自己寫的類別的物件，因為我們必須肩負釋放自己產生物件的責任，因此做法是使用autorelease來做。這個method會把release的動作留到最近一次的@autoreleasepool區塊結束時才釋放記憶體。

Example:

```

Singer* createSinger() {
    Singer* newSinger = [[[Singer alloc] init] autorelease];

    return newSinger;
}

```