

Objective-C程式設計



授課：宋文平

Email：perkinsung@hotmail.com

NSLog

表 2-3 資料型態表示範圍及其輸出格式

型態修飾詞	資料 型態	所佔 byte 數	表示範圍	使用 NSLog 函 式的輸出格式
	char	1	-128 ~ 127	%c
	int	4	-2,147,483,648 ~ 2,147,483,647	%i %d
	float	4	3.4E-38 ~ 3.4E+38	%f
	double	8	1.7E-308 ~ 1.7E+308	%f
short	int	2	-32,768 ~ 32767	%hi
long	int	8		%li
long	double	16		%Lf
★ unsigned	int	4	0 ~ 4,294,967,295	%u
unsigned short	int	2	0 ~ 65,535	%hu
unsigned long	int	8		%lu
	char[] or char *			%s

NSNumber

表 19-1 常用的數值物件初始化方法

資料型態	初始方法	自動配置與初始方法	對應的讀取方法
int	-initWithInt:	+numberWithInt:	-intValue
float	-initWithFloat:	+numberWithFloat:	-floatValue
double	-initWithDouble:	+numberWithDouble:	-doubleValue
char	-initWithChar:	+numberWithChar:	-charValue
BOOL	-initWithBool:	+numberWithBool:	-boolValue
unsigned int	-initWithUnsignedInt:	+numberWithUnsignedInt:	-unsignedIntValue
NSInteger	-initWithInteger:	+numberWithInteger:	-integerValue

表 19-2 常用的 NSNumber 方法

方法	說明
- (BOOL) isEqualToNumber: (NSNumber *) aNumber	若接收者的數值等於 <i>aNumber</i> ，則回傳 YES，否則回傳 NO
- (NSComparisonResult) compare: (NSNumber *) aNumber	若接收者的數值小於 <i>aNumber</i> ，則回傳 NSOrderedAscending；若等於 <i>aNumber</i> ，則回傳 NSOrderedSame；若大於 <i>aNumber</i> ，則回傳 NSOrderedDescending

NSComparisonResult

回傳值	列舉中代表的數值
NSOrderedAscending	-1
NSOrderedSame	0
NSOrderedDescending	1

NSString

表 19-3 常用的 NSString 方法

方法	說明
- (id)initWithString: (NSString *) <i>aString</i>	以 <i>aString</i> 初始化字串，需自行釋放記憶體
- (id)initWithUTF8String: (const char *) <i>bytes</i>	以 <i>bytes</i> 初始化字串。 <i>bytes</i> 是以 UTF-8 編碼的 C 字元陣列
+ (id)stringWithFormat: (NSString *) <i>format</i> , ...	將...對應至 <i>format</i> 格式後，再回傳此格式形成的字串
- (NSUInteger)length	回傳字串的長度
- (NSString *)stringByAppendingString: (NSString *) <i>aString</i>	將 <i>aString</i> 附加至接收者字串的尾端，然後回傳此新字串
- (unichar)characterAtIndex: (NSUInteger) <i>index</i>	回傳陣列索引值 <i>index</i> 代表的字元
- (NSString *)substringWithRange: (NSRange) <i>aRange</i>	只取接收者在 <i>aRange</i> 範圍內的字串後，將此字串回傳
- (NSString *) stringByTrimmingCharactersInSet: (NSCharacterSet *) <i>set</i>	若接收者左右邊符合 <i>set</i> 中任一字元，則回傳刪除這些字元的字串
- (NSArray *)componentsSeparatedByString: (NSString *) <i>separator</i>	以 <i>separator</i> 為基準，將字串切分為數個子字串後，以陣列儲存並回傳
- (NSString *) stringByReplacingOccurrencesOfString: (NSString *) <i>target</i> withString: (NSString *) <i>replacemen</i>	將接收者符合 <i>target</i> 的子字串，置換為 <i>replacemen</i>
- (BOOL)isEqualToString: (NSString *) <i>aString</i>	若 <i>aString</i> 與接收者相等，則回傳 YES；否則回傳 NO

NSString

方法	說明
- (NSComparisonResult)compare: (NSString *)aString	若接收者小於 <i>aString</i> ，則回傳 NSOrderedAscending；若接收者等於 <i>aString</i> ，則回傳 NSOrderedSame；若接收者大於 <i>aString</i> ，則回傳 NSOrderedDescending
- (NSString *)lowercaseString	將字串所有的字元轉換為小寫
- (NSString *)uppercaseString	將字串所有的字元轉換為大寫

表 19-4 將字串轉換為數值的方法

方法	回傳值的型態
intValue	int
boolValue	BOOL
longLongValue	long long
floatValue	float
doubleValue	double
integerValue	NSInteger

NSStringCharacterSet

表 19-5 常用 NSStringCharacterSet 方法

方法	說明
<code>+alphanumericCharacterSet</code>	回傳字母、標點符號與數字集
<code>+controlCharacterSet</code>	回傳控制字元集
<code>+newlineCharacterSet</code>	回傳換行字元集
<code>+whitespaceAndNewlineCharacterSet</code>	回傳空白與換行字元集
<code>+whitespaceCharacterSet</code>	回傳空白字元集

NSMutableString

表 19-6 常用的 NSMutableString 方法

方法	說明
+ (id)stringWithCapacity: (NSUInteger) <i>capacity</i>	建立一個儲存 <i>capacity</i> 個字元的可變更字串
- (void)appendString: (NSString *) <i>aString</i>	將字串 <i>aString</i> 附加於接收者之後
- (void)deleteCharactersInRange: (NSRange) <i>aRange</i>	將接收者在 <i>aRange</i> 之間的字串刪除
- (void)insertString: (NSString *) <i>aString</i> atIndex: (NSUInteger) <i>anIndex</i>	將字串 <i>aString</i> 插入接收者位於 <i>anIndex</i> 的位置

NSArray

表 19-6 常用的 NSArray 方法

方法	說明
+ (id)arrayWithObjects: (id) <i>firstObj</i> , ...	將多個物件當作陣列的初始值，最後以 <code>nil</code> 結束，例如： <code>str1, str2, str3, ..., strN, nil</code>
- (NSUInteger)count	回傳陣列儲存的物件數量
- (NSUInteger)indexOfObject: (id) <i>anObject</i>	若陣列中有物件 <i>anObject</i> ，則回傳此物件於陣列中的索引值，否則回傳 <code>NSNotFound</code>

方法	說明
- (id)objectAtIndex: (NSUInteger) <i>index</i>	回傳陣列索引值為 <i>index</i> 的物件
- (void)makeObjectsPerformSelector: (SEL) <i>aSelector</i>	對陣列內所有物件執行 <i>aSelector</i> 方法

NSMutableArray

表 19-8 常用的 NSMutableArray 方法

方法	說明
- (void)addObject: (id) <i>anObject</i>	將 <i>anObject</i> 加入陣列
- (void)removeObject: (id) <i>anObject</i>	將陣列內所有符合 <i>anObject</i> 的物件刪除
- (void)sortUsingSelector: (SEL) <i>comparator</i>	以 <i>comparator</i> 的方法，將陣列的元素加以排序
- (void)replaceObjectAtIndex: (NSUInteger) <i>index</i> withObject: (id) <i>anObject</i>	將陣列索引值為 <i>index</i> 的元素置換為 <i>anObject</i>
- (void)exchangeObjectAtIndex: (NSUInteger) <i>idx1</i> withObjectAtIndex: (NSUInteger) <i>idx2</i>	將陣列索引值為 <i>idx1</i> 與 <i>idx2</i> 的元素調換

NSObject

方法	功能說明
-(BOOL)isMemberOfClass:	判斷某一物件是否為某一類別的成員
-(BOOL)isKindOfClass:	判斷某一物件是否為某一類別或是其子孫類別

方法	功能說明
-respondsToSelector:@selector	此類別或是物件是否能回應 @selector 所指定的方法 由物件去檢查物件方法
+instancesRespondToSelector:@selector	某一類別所建立的物件是否能回應 @selector 所指定的實體方法 由類別去檢查物件方法
+isSubclassOfClass:	判斷某一類別是否為某一類別的子類別

NSDictionary

方法	功能說明
<code>respondsToSelector:@selector</code>	此類別或是物件是否能回應 <i>@selector</i> 所指定的方法
<code>instancesRespondToSelector:@selector</code>	某一類別所建立的物件是否能回應 <i>@selector</i> 所指定的實體方法
<code>isSubclassOfClass:</code>	判斷某一類別是否為某一類別的子類別

NSMutableDictionary

表 19-10 常用的 NSMutableDictionary 方法

方法	說明
+ (id)dictionaryWithCapacity: (NSUInteger) <i>numItems</i>	建立並初始一個具有 <i>numItems</i> 容量的物件
- (id)initWithCapacity: (NSUInteger) <i>numItems</i>	初始一個具有 <i>numItems</i> 容量的物件
- (void) setObject: (id) <i>anObject</i> forKey: (id) <i>aKey</i>	將 <i>anObject</i> 物件以鍵值 <i>aKey</i> 存入
- (void) removeObjectForKey: (id) <i>aKey</i>	將 <i>aKey</i> 鍵值對應的物件刪除

NSSet

表 19-11 常用的 NSSet 方法

方法	說明
+ (id)setWithObjects: (id) <i>anObject</i> ...	以 <i>anObject</i> ..., nil 等數個物件建立
- (id)initWithObjects: (id) <i>firstObj</i> ...	以 <i>firstObj</i> ..., nil 等數個物件初始化
- (BOOL)containsObject: (id) <i>anObject</i>	若接收者儲存 <i>anObject</i> ，則回傳 YES，否則回傳 NO
- (BOOL)isSubsetOfSet: (NSSet *) <i>otherSet</i>	若 <i>otherSet</i> 是接收者的子集合，則回傳 YES，否則回傳 NO
- (BOOL)isEqualToSet: (NSSet *) <i>otherSet</i>	若 <i>otherSet</i> 等於接收者，則回傳 YES，否則回傳 NO
- (BOOL)intersectSet: (NSSet *) <i>otherSet</i>	若 <i>otherSet</i> 儲存的物件，有一個以上相同於 <i>otherSet</i> 內的物件，則回傳 YES，否則回傳 NO

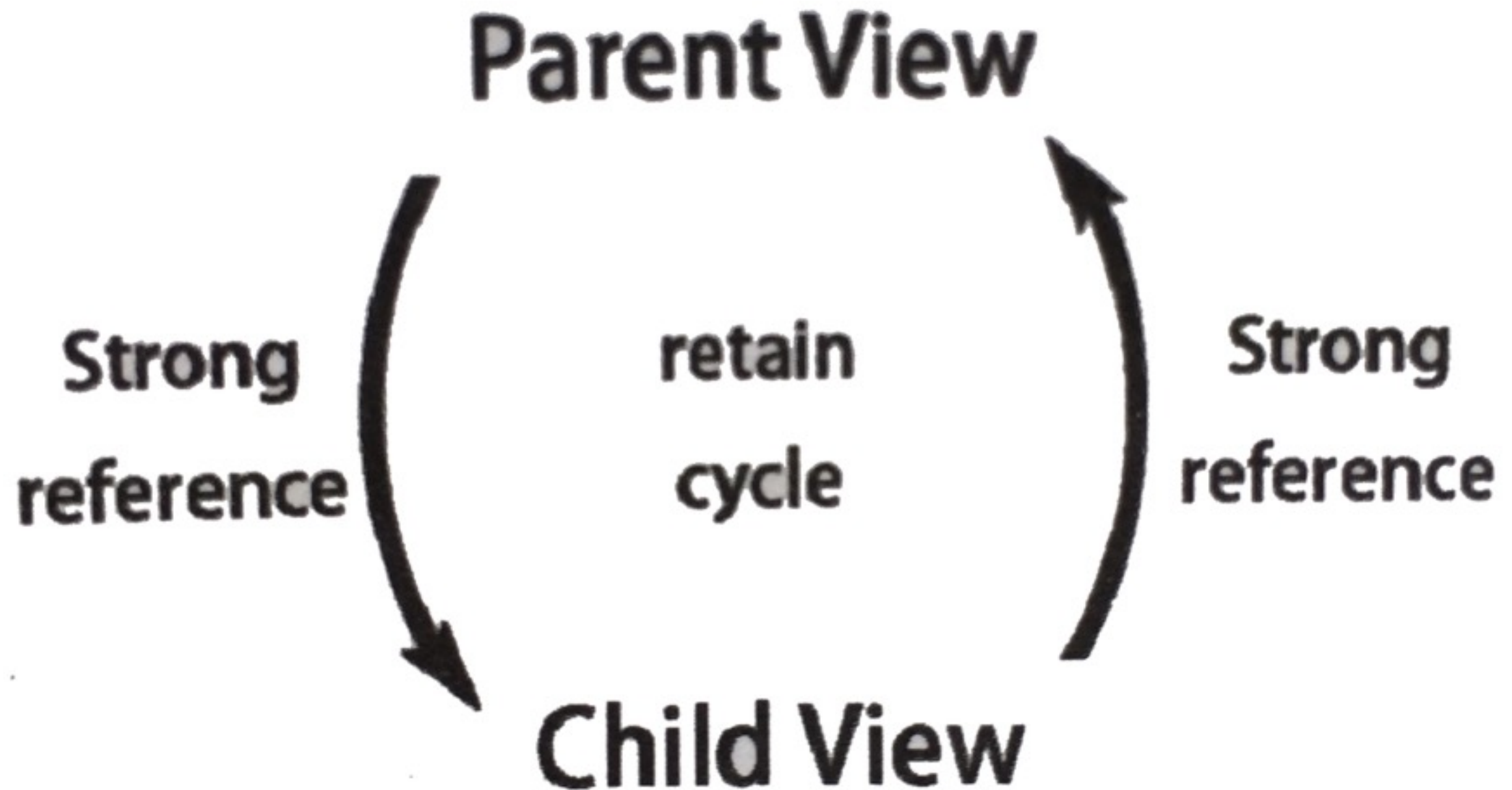
NSMutableSet

表 19-12 常用的 NSMutableSet 方法

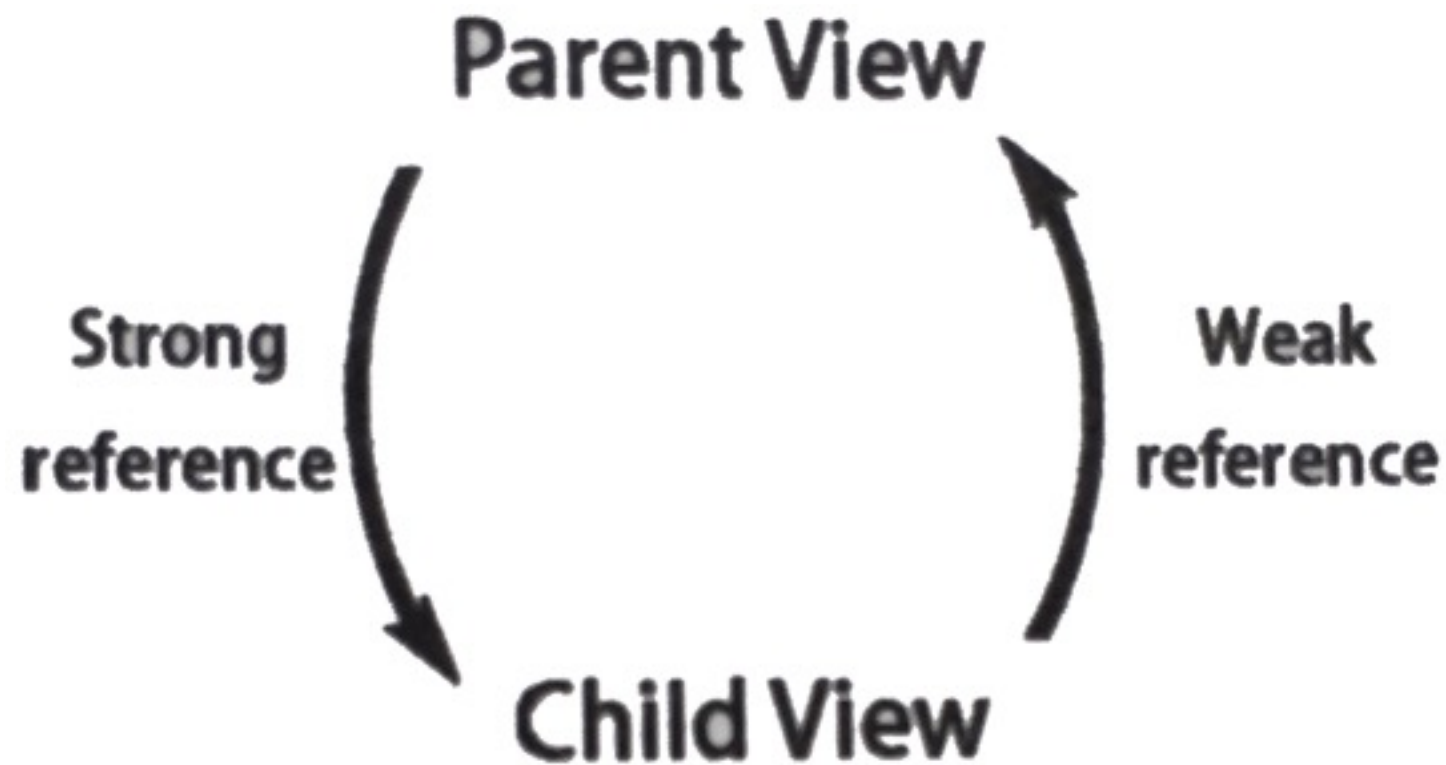
方法	說明
+ (id)setWithCapacity: (NSUInteger) <i>numItems</i>	建立並初始為具有 <i>numItems</i> 容量的集合
- (id)initWithCapacity: (NSUInteger) <i>numItems</i>	初始為具有 <i>numItems</i> 容量的集合
- (void)addObject: (id) <i>anObject</i>	將 <i>anObject</i> 加入接收者
- (void)removeObject: (id) <i>anObject</i>	將 <i>anObject</i> 從接收者中移除

方法	說明
- (void)unionSet: (NSSet *) <i>otherSet</i>	將 <i>otherSet</i> 的所有物件存入接收者
- (void)minusSet: (NSSet *) <i>otherSet</i>	將接收者內與 <i>otherSet</i> 相同的物件刪除
- (void)intersectSet: (NSSet *) <i>otherSet</i>	只留下接收者中，與 <i>otherSet</i> 相同的物件

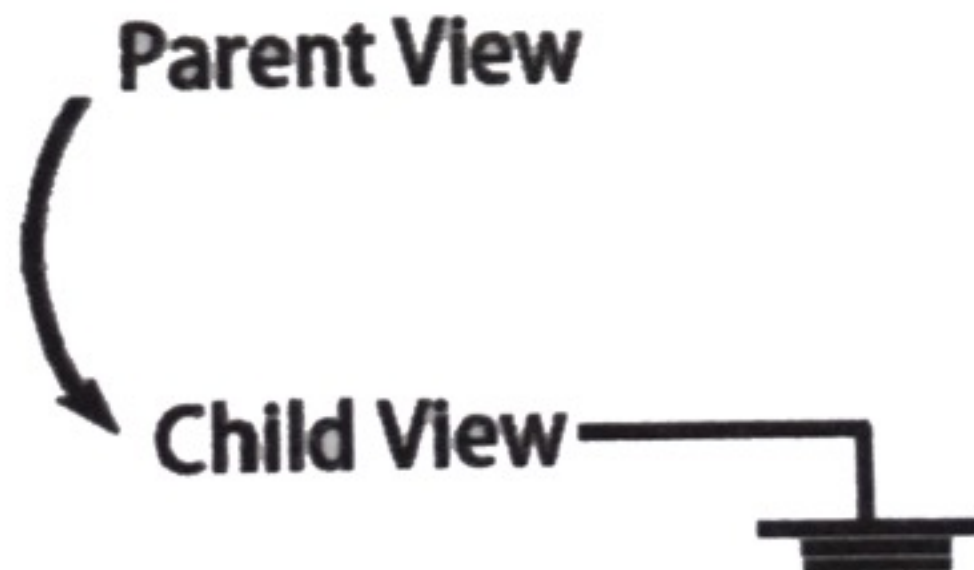
記憶體管理



記憶體管理



當 **Child View** 物件被釋放後，
Child View 會指向 nil



@property

表 22-1 @property 的一些屬性

屬性(attribute)	說明
assign	直接將參數指定給類別成員
retain	先將類別成員參考的物件 release，再將 retain 後的參數指定給類別成員
copy	先將類別成員參考的物件 release，再將複製的參數指定給類別成員
atomic	若在多執行緒的環境下，setter 與 getter 功能會受到保護，同一時間內只能讓一個執行緒存取。(atomic 關鍵字並不存在，在這裡列出僅是告知讀者，這是預設的模式)
nonatomic	系統不會啟用單一讀寫限制，換言之，如果確定使用 getter 與 setter 的時機，不是以多執行緒的方法設計，即可使用此模式增加執行效率

常用數學函式

表 12-1 常用的數學函式

函式	功能說明
<code>pow(x, y)</code>	x 的 y 次方，亦即 x^y
<code>sqrt(x)</code>	x 的平方根
<code>floor(x)</code>	小於 x 的最大正整數 小數部分無條件捨去
<code>ceil(x)</code>	大於 x 的最小正整數 小數部分無條件進位
<code>exp(x)</code>	e^x
<code>log(x)</code>	以 e 為底 x 的對數值，亦即 $\log_e(x)$
<code>log10(x)</code>	以 10 為底 x 的對數值，亦即 $\log_{10}(x)$
<code>abs(x)</code>	整數 x 的絕對值
<code>fabs(x)</code>	浮點數 x 的絕對值
<code>rand()</code>	產生 0~最大整數之間的亂數
<code>srand(x)</code>	產生亂數種子<需配合rand()使用>
<code>arc4random()</code>	產生 0~最大整數之間的亂數<不需亂數種子>

物件導向基礎名詞釋要

- ★ 類別 class → 物件 object（類別實體 instance）
- ★ 函式 function → 方法 method<定義於類別之內>
- ★ 變數 variable → 屬性 property<定義於類別之內>