

Objective-C 2.0 CheatSheet iOS 5+ (ARC)

This sheet (c) The iPhoneCoach 2012 - www.iphonecoach.com.au

Apple Docs: <http://developer.apple.com/library/ios>



Special & Commonly used Keywords:

```
@interface // declare a class
@implementation // implement a class
@property // declare a property
@synthesize // implement a property

#import // import a header file

void // a pseudo-data type introduced to
      // allow methods to 'return nothing'
id // a generic type indicating a
   // 'pointer to any kind of object'

self // the current instance of the class
super // current instance of the superclass
nil // equivalent to '0' for object pointers
```

Declaring variables:

```
int // 10,20,30, ...
float // 1.2345, 3.1415, ...
MyClass* // NSString*, NSArray*, ...
```

Class Declaration (.h):

```
@interface MyClass : SuperClass {
    // declare instance variables
}
// declare properties
// declare methods
@end
```

Class Implementation (.m):

```
#import "MyClass.h"

@implementation MyClass
// synthesize properties
// implement methods
@end
```

Method Declaration (in .h file):

```
-(type1)doit;
-(type2)doWithA:(type3)a;
-(type4)doWithA:(type5)a andB:(type6)b;
```

Method Implementation (in .m file):

```
-(type4)doWithA:(type5)a andB:(type6)b{
    // Do something with a and b
    return ret;
}
```

Declaring Properties (in .h file):

```
// Declare int property named 'age'
@property (assign) int age;

// Declare NSString property named 'name'
@property (strong) NSString *name;
```

Implementing Properties (in .m file):

```
// synthesize property named 'age'
@synthesize age;
// synthesize property named 'name'
@synthesize name;
```

Using Properties:

```
self.intProperty = 10;
NSString *tmp = self.stringProperty;
```

Many ways to instantiate an NSString object:

```
NSString *me = @"Patrick";
NSString *tmp = [[NSString alloc]
initWithFormat:@"Hello %@",me];
```

Instantiating a class using basic initializer:

```
// create new, empty instance of class MyClass
MyClass *obj = [[MyClass alloc] init];
```

Calling a method / Sending a message to an object:

```
[obj doit]; // send 'doit' message to obj
[obj doWithA:a]; // call 'doWithA' and pass 'a'
```

Special Keywords in iPhone Apps:

```
IBOutlet // a special compiler macro that
          // resolves to nothing and is only
          // used by Interface Builder to
          // indicate that this object is
          // connected to a User Interface
          // element such as a button
```

```
IBAction // same as 'void', but hinting to
          // Interface Builder that this
          // method is linked to an event
          // generated by a UI element, i.e.
          // a button being pressed
```

Creating an array of objects:

```
NSArray *people = [[NSArray alloc]
initWithObjects:@"Peter",@"Paul",@"Mary",nil];
```

Fast enumeration of an array:

```
for (NSString *name in people)
{
    NSLog(@"I have %@ in the team",name);
}
```