



Objective-C Cheat Sheet and Quick Reference

Class Header (.h)

```
#import "AnyHeaderFile.h"

@interface ClassName : SuperClass

// define public properties
// define public methods

@end
```

Class Implementation (.m)

```
#import "YourClassName.h"

@interface ClassName ()
// define private properties
// define private methods
@end

@implementation ClassName {
// define private instance variables
}

// implement methods

@end
```

Defining Methods

```
- (type)doIt;
- (type)doItWithA:(type)a;
- (type)doItWithA:(type)a
  b:(type)b;
```

Implementing Methods

```
- (type)doItWithA:(type)a
  b:(type)b {
    // Do something with a and b...
    return retVal;
}
```

Creating an Object

```
ClassName * myObject =
    [[ClassName alloc] init];
```

Calling a Method

```
[myObject doIt];
[myObject doItWithA:a];
[myObject doItWithA:a b:b];
```

Declaring Variables

```
type myVariable;
```

Variable types	
int	1, 2, 500, 10000
float double	1.5, 3.14, 578.234
BOOL	YES, NO
ClassName *	NSString *, NSArray *, etc.
id	Can hold reference to any object

Defining Properties

```
@property (attribute1, attribute2)
    type propertyName;
```

strong	Adds reference to keep object alive
weak	Object can disappear, become nil
assign	Normal assign, no reference
copy	Make copy on assign
nonatomic	Make not threadsafe, increase perf
readwrite	Create getter&setter (default)
readonly	Create just getter

Using Properties

```
[myObject setPropertyName:a];
myObject.propertyName = a; // alt

a = [myObject propertyName];
a = myObject.propertyName; // alt
```

What is a Property?

1) Automatically defines a private instance variable:

```
type _propertyName;
```

2) Automatically creates a getter and setter:

```
- (type)propertyName;
- (void)setPropertyName:(type)name;
```

Using `_propertyName` uses the private instance variable directly. Using `self.propertyName` uses the getter/setter.

Custom_INITIALIZER_Example

```
- (id)initWithParam:(type)param {
    if ((self = [super init])) {
        _propertyName = param;
    }
    return self;
}
```

NSString Quick Examples

```
NSString *personOne = @"Ray";
NSString *personTwo = @"Shawn";
NSString *combinedString =
    [NSString stringWithFormat:
        @"%@: Hello, %@!",
        personOne, personTwo];
NSLog(@"%@", combinedString);
NSString *tipString = @"24.99";
float tipFloat = [tipString floatValue];
```

NSArray Quick Examples

```
NSMutableArray *array =
    [[person1, person2] mutableCopy];
[array addObject:@"Waldo"];
NSLog(@"%d items!", [array count]);
for (NSString *person in array) {
    NSLog(@"Person: %@", person);
}
NSString *waldo = array[2];
```