

Cocoa Naming Conventions

Matthias Tretter, @myell0w

Clarity

Code	Commentary
<code>insertObjectAtIndex:</code>	Good!
<code>insert:at:</code>	Not clear: what is being inserted? what does “at” signify?
<code>removeObjectAtIndex:</code>	Good!
<code>removeObject:</code>	Good, because it removes object referred to in argument.
<code>remove:</code>	Not clear; what is being removed?

Abbreviations

Code	Commentary
destinationSelection	Good.
destSel	Not clear.
setBackgroundColor:	Good.
setBkgdColor:	Not clear.

Typography

- ⦿ Names composed of several words:
Camel Casing (`nameComposedOfSeveralWords`)
- ⦿ Method names start with **lowercase** (after each :)
`fileExistsAtPath:`
- ⦿ Class/Protocol/Struct/Function names start with **uppercase**

Naming Methods

- For methods that represent **actions** start the name with a **verb**
 - (void)invokeWithTarget:(id)target
 - (void)selectTabViewItem:(NSTabViewItem *)tabViewItem
- If the method returns an attribute of the receiver, name the method after the attribute

Code	Commentary
- (CGSize)cellSize	Right.
- (CGSize)calcCellSize	Wrong.
- (CGSize)getCellSize	Wrong.

Naming Methods

- ☛ Use keywords before all arguments

Code	Commentary
<pre>- (void)sendAction:(SEL)aSelector to:(id)anObject forAllCells:(BOOL)flag</pre>	Right.
<pre>- (void)sendAction:(SEL)aSelector :(id)anObject :(BOOL)flag</pre>	Wrong.

Naming Methods

- ⦿ Make the word before the argument describe the argument

Code	Commentary
- (id)viewWithTag:(NSInteger)tag	Right.
- (id)taggedView:(NSInteger)tag	Wrong.

Naming Methods

- Add new keywords to the end of an existing method when you create a method that is more specific than the inherited one

Code	Commentary
- (id)initWithFrame:(CGRect)frame	Defined in UIView
- (id)initWithFrame:(CGRect)frame labelColor:(UIColor *)color	Hypothetic more specific method

Naming Methods

- ➊ Don't use „and“ to link keywords that are attributes

Code	Commentary
<pre>- (int)runInDirectory:(NSString *)p file:(NSString *)name types:(NSArray *)fileTypes</pre>	Right.
<pre>- (int)runInDirectory:(NSString *)p andFile:(NSString *)name andTypes:(NSArray *)fileTypes</pre>	Wrong.

Other catchphrases

- ⦿ Consistency
- ⦿ No self-reference: `NSStringObject`, `NSInteger resultInteger`
- ⦿ Prefixes: NS, UI, your own: 3 Letters

Further Literature

- ⦿ Introduction to Coding Guidelines for Cocoa, Apple
<https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/CodingGuidelines/CodingGuidelines.html>
- ⦿ Cocoa Style for Objective-C, CocoaDevCentral
<http://cocoadevcentral.com/articles/000082.php>