



# Objective-C

Witcha Feungchan  
2013





# Objective-C

Witcha Feungchan  
2013





# Objective-C

Witcha Feungchan  
2013



# Overview

- Method
- Instance
- Properties

# Language

- Objective C
- [obj method : argument]
- [hello drawAtPoint:location withFont:font];

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

- for instance method  
+ for class method

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

return type in (.....)  
can be void

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

Method name part I & II  
carAtPoint: withDamage:

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

Type of first argument

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;



First argument

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

Second argument

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

Semicolon at the end of statement

# Method

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

# Method

- (return\_type)instanceMethod2WithParameter:(param1\_type)param1\_varName andOtherParameter:(param2\_type)param2\_varName;

- (NSArray \*)carAtPoint:(CGPoint)crashLocation withDamage:(BOOL) damaged;

# Method

- (return\_type)instanceMethod2WithParameter:(param1\_type)param1\_varName andOtherParameter:(param2\_type)param2\_varName;
- **(void) splitViewController: (UISplitViewController\*) svc**  
**willHideViewController: (UIViewController \*) aViewController**  
**withBarButtonItem: (UIBarButtonItem \*) barButtonItem forPopoverController:**  
**(UIPopoverController \*) popoverController;**

# Instance Methods

- -' dash in front of method
- “Normal” methods in OOP
- Access local variables
- Send messages to self and super

# Objects

- Data (variables)
- Behavior (Methods)
- Dynamic typing
- id: data type for object identifier Memory management
- (reference counting, garbage collection)

# Objects Messaging

- Message Syntax
  - [receiver message]; ==> receiver.message
  - receiver = an object
  - message = what object should do?

```
[myRectangle display];  
[myRectangle setWidth: 20.0];  
[myRectangle setOriginx: 20.0 y: 30.0];  
[myRectangle setPrimaryColor:[otherRect primaryColor]];
```

# Accessing Properties

```
Graphic *graphic = [[Graphic alloc] init];  
  
NSColor *color = graphic.color;  
  
CGFloat xLoc = graphic.xLoc;  
  
BOOL hidden = graphic.hidden;  
  
int textCharacterLength = graphic.text.length;  
  
if (graphic.textHidden != YES) {  
    graphic.text = @"Hello";  
}  
  
graphic.bounds = NSMakeRect(10.0, 10.0, 20.0, 120.0);
```

```
Graphic *graphic = [[Graphic alloc] init];  
  
NSColor *color = [graphic color];  
  
CGFloat xLoc = [graphic xLoc];  
  
BOOL hidden = [graphic hidden];  
  
int textCharacterLength = [[graphic text] length];  
  
if ([graphic isTextHidden] != YES) {  
    [graphic setText:@"Hello"];  
}  
  
[graphic setBounds:NSMakeRect(10.0, 10.0, 20.0, 120.0)];
```

# Class

- NSObject is a root class
- Inheritance
- Overriding
- Interface .h
- Implementation .m

# Class Code

```
@interface ClassName : ItsSuperclass
```

```
{
```

```
    instance variable declarations
```

```
}
```

```
method declarations
```

```
@end
```

```
@implementation ClassName : ItsSuperclass
```

```
{
```

```
    instance variable declarations
```

```
}
```

```
method definitions
```

```
@end
```

# Foundation Framework

- NSObject : a root class for most of objects
- NSString : using Unicode (cannot be modified)
- NSMutableString : mutable of string
- NSNumber : wrapper primitive type int,float, etc.
- NSValue : wrapper for other non-object data types
- NSData : wrapper chunk of data for save/transmit
- NSDate : ask or store current date/time

# Read More

<https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/chapters/WriteObjective-CCode/WriteObjective-CCode/WriteObjective-CCode.html>

<https://developer.apple.com/library/ios/documentation/cocoa/conceptual/ProgrammingWithObjectiveC/ProgrammingWithObjectiveC.pdf>