Brief Intro to Objective-C

5/12 ‘16

- Strict Superset of C.
  Objective-C is a thin layer on top of C, and moreover is a strict superset of C.

- Syntax derived from Smalltalk.

  ```
  'hello world' indexOf:'o'
  index('hello world', 'o')
  ```

- Why Objective-C?
  An Object-oriented language, extension of standard ANSI-C, simple syntax, and dynamic runtime.
  In the late 1980s, it was selected as the main language used by NeXT for its NeXTSTEP operating system, which OS X is derived from.

Objective-C Types

- BOOL
  Boolean, “YES” and “NO”.
- NSInteger, NSUInteger
  A typedef for integer data type. In 32-bit runtime, it’s int. In 64-bit, it’s long.
- id
  Some kind of object, Like `AnyObject` in Swift.
- nil
  Null pointer
- self
  A pointer to current object context.
- super
  A pointer used to access superclass of current object.

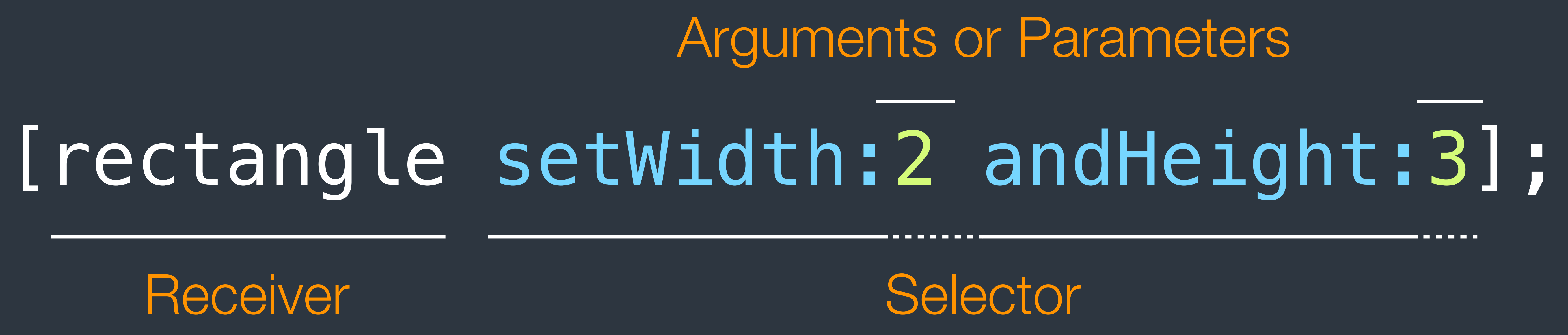
Traditional Function Call

```
Rectangle r = new Rectangle(10, 10, 20, 30);
Size s = r.getSize();
System.out.println(s.x);
```

Message Passing

```
Rectangle *r = [[Rectangle alloc]
                initWithOriginX:10
                originY:10
                width:1
                andHeight:2];
Size *s = [r size];
NSLog(@"%d", s.x);
```

Message Syntax



Arguments or Parameters

```
[rectangle setWidth:2 andHeight:3];
```

Receiver

Selector

- Receiver is “rectangle”
- Arguments are “2” and “3”

- Selector is “setWidth:andHeight:”

- No method overload. No default value for arguments.
  Due to its implementation and C compatibility

- It’s ok to send any message to “nil”, the null pointer in Objective-C.
  Nothing will happened, just a “NOP” if you send a message to “nil”.

- All classes must inherit from NSObject. All instances are references.

Class declaration (*.h)

```
@interface CLASS_NAME : SUPER_CLASS <PROTOCOL> {

    MEMBERS

}

PROPERTIES

+ CLASS METHODS DECLARATION

- INSTANCE METHODS DECLARATION

@end
```

Class declaration (*.h)

```
@interface IOFlight : NSObject {
    IOAirplane *plane;
}

@property (nonatomic, strong) IOAirport *origin;
@property (nonatomic, strong) IOAirport *destination;
@property (nonatomic, getter=isFlying) BOOL flying;

+ (int)countOfAllFlights;
- (BOOL)takeOff;
- (BOOL)land;

@end
```

“+” is static method.

“-” is instance method.

Class implementation (*.m)

```
@implementation CLASS_NAME

METHODS IMPLEMENTATION

@end
```

Class implementation (*.m)

```
@implementation IOFlight

+ (int)countOfAllFlights { ... }

- (BOOL)takeOff {
    [self.origin askForLeaving];
    ...
}

- (BOOL)land {
    [self.destination askForLanding];
    ...
}

@end
```