

Tóm tắt ngôn ngữ Swift 2.x

Cho anh em code Objective-C hay Swift 1.x

cuong@techmaster.vn
<http://techmaster.vn>

Swift x.x

- Read Evaluate Print Loop
- Playground
- Type inference
- UnsafePointer, UnsafeMutablePointer
- class
- protocol
- let, var
- private, public, internal

Objective-C

- No REPL
- No Playground
- Phải khai báo tường minh
- Pointer *, ->
- @interface
- @protocol
- @property
- @private, @protected, @public, @package

Swift String

Cú pháp đơn giản, dễ hiểu

```
var name = "Trinh"  
name += " Minh Cuong"  
print(name)
```

Objective-C String

NSString, C string...

Phải dùng method NSString xử lý

Không viết



```
NSMutableString* name = [NSMutableString stringWithString:@"Trinh"];  
[name appendString:@" Minh Cuong"];  
NSLog(@"%@", name);
```

Swift có Tuple

Giống Python, đóng gói nhiều biến

```
var a:(String,Int) = ("David",40)

func printAge(pa: (String, Int)) {
    print("\(pa.0) is \(pa.1)")
}

printAge(a)
```

Objective-C có NSDictionary

Không có tuple phải dùng
NSDictionary để thay thế

```
NSDictionary* a =
@{@"name": @"David", @"age": @40};
```

```
-(void) printAge: (NSDictionary*) pa {  
    NSLog(@"%@ is %@", pa[@"name"], pa[@"age"]);  
}  
  
-(void) demo {  
    NSDictionary* a = @{@"name": @"David", @"age": @40};  
  
    [self printAge: a];  
}
```



Swift x.x

- Function in function

```
func helloWorld() {  
    func hiWorld() {  
        print("Hello World")  
    }  
    hiWorld()  
}  
helloWorld()
```

Objective-C

- Không được phép
- Có thể khai báo closure trong method

```
- (void)helloWorld {  
    int num = 42;  
    void (^hiWorld)(void) = ^{  
        NSLog(@"%i", num);  
    };  
  
    hiWorld();  
}
```

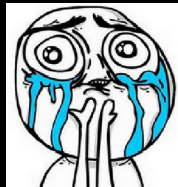
Swift variadic

```
func readIn(num: Int...)->Int {  
    var result: Int = 0  
    for i: Int in num {  
        result += i  
    }  
    return result  
}  
print(readIn(1, 2, 3, 4))
```

Objective-C variadic


```
- (int) addValues:(int) count, ... {  
    va_list args;  
    int result;  
  
    va_start(args, count);  
    for( int i = 0; i < count; i++ ) {  
        int value = va_arg(args, int);  
        result += value;  
    }  
    va_end(args);  
    return result;  
}
```

```
NSLog(@"%d", [self addValues:4, 1, 2, 3,  
4]);
```





Chồng đề toán tử



Swift Syntactic Sugar Syntax

Code gọn, đẹp, dễ hiểu

```
extension Double {  
    var km: Double { return self * 1_000.0 }  
    var m: Double { return self }  
    var cm: Double { return self / 100.0 }  
    var mm: Double { return self / 1_000.0 }  
    var ft: Double { return self / 3.28084 }  
}
```

```
let oneInch = 25.4.mm ←
```

```
let threeFeet = 3.ft ←
```

Syntatic Sugar

```
Async.background {  
    print("This is run on the background queue")  
}.main {  
    print("This is run on the main queue, after the previous block")  
}
```

No Sugar

```
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND, 0), {  
    print("This is run on the background queue")  
  
    dispatch_async(dispatch_get_main_queue(), {  
        print("This is run on the main queue, after the previous block")  
    })  
})
```



Tham khảo

- <http://sketchytech.blogspot.com/2015/06/whats-new-in-swift-20-repeat-while.html>
- <https://www.toptal.com/swift/from-objective-c-to-swift>
- <https://redwerk.com/blog/10-differences-objective-c-swift>
- <http://useyourloaf.com/blog/swift-string-cheat-sheet/>