The Swift Programming Language

Extensions

osxdev (hothead)
성기평 (sungkipyung@gmail.com)

시작 전 공지

https://github.com/sungkipyung/Swift_Extension_osxdev

OSXDev.SwiftStudy.start()

Extension 이미지 검색 결과

Extension Syntax

```
class MyTableViewController: UIViewController {
    @IBOutlet weak var tableView: UITableView!

    fileprivate var items: [String]!
    fileprivate var selectedIndexPath: IndexPath?
    override func viewDidLoad() {
        super.viewDidLoad()

        items = ["1", "2", "3"]
    }
}

extension MyTableViewController: UITableViewDataSource {
    public func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return items.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "MyTableViewCell", for: indexPath) as!
MyTableViewCell

        cell.myTextLabel.text = items[indexPath.row]

        if selectedIndexPath == indexPath {
            cell.backgroundColor = UIColor.red
        } else {
            cell.backgroundColor = UIColor.white
        }

        return cell
    }
}

extension MyTableViewController: UITableViewDelegate {
    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        selectedIndexPath = indexPath
        tableView.reloadData()
    }
}
```

Computed Properties

```swift
import UIKit

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
print("One inch is \(oneInch) meters")
// Prints "One inch is 0.0254 meters"
let threeFeet = 3.ft
print("Three feet is \(threeFeet) meters")
// Prints "Three feet is 0.914399970739201 meters"

let aMarathon = 42.km + 195.m
print("A marathon is \(aMarathon) meters long")
// Prints "A marathon is 42195.0 meters long"
```

Initializers

```swift
struct Size {
    var width = 0.0, height = 0.0
}

struct Point {
    var x = 0.0, y = 0.0
}

struct Rect {
    var origin = Point()
    var size = Size()
}

let defaultRect = Rect()
let memberwiseRect = Rect(origin: Point(x: 2.0, y: 2.0),
                          size: Size(width: 5.0, height: 5.0))

// x, y 좌표 말고 center와 Size로 Rect를 만들고 싶으면
extension Rect {
    init(center: Point, size: Size) {
        let originX = center.x - (size.width / 2)
        let originY = center.y - (size.height / 2)
        self.init(origin: Point(x: originX, y: originY), size: size)
    }
}

let centerRect = Rect(center: Point(x: 4.0, y: 4.0),
                      size: Size(width: 3.0, height: 3.0))

// centerRect's origin is (2.5, 2.5) and its size is (3.0, 3.0)
```

Methods

```swift
extension Int {
    func repetitions(task: () -> Void) {
        for _ in 0..<self {
            task()
        }
    }
}

3.repetitions {
    print("Hello!")
}
// Hello!
// Hello!
// Hello!
```

Mutating Instance Methods

```swift
extension Int {
    mutating func square() {
        self = self * self
    }
}
var someInt = 3
someInt.square()
// someInt is now 9
```

Subscripts

```
extension Int {
    subscript(digitIndex: Int) -> Int {
        var decimalBase = 1
        for _ in 0..<digitIndex {
            decimalBase *= 10
        }
        return (self / decimalBase) % 10
    }
}
746381295[0]
// returns 5
746381295[1]
// returns 9
746381295[2]
// returns 2
746381295[8]
// returns 7

746381295[9]
// returns 0, as if you had requested:
0746381295[9]
```

Nested Types

```swift
extension Int {
    enum Kind {
        case negative, zero, positive
    }
    var kind: Kind {
        switch self {
        case 0:
            return .zero
        case let x where x > 0:
            return .positive
        default:
            return .negative
        }
    }
}

func printIntegerKinds(_ numbers: [Int]) {
    for number in numbers {
        switch number.kind {
        case .negative:
            print("- ", terminator: "")
        case .zero:
            print("0 ", terminator: "")
        case .positive:
            print("+ ", terminator: "")
        }
    }
    print("")
}
printIntegerKinds([3, 19, -27, 0, -6, 0, 7])
// Prints "+ + - 0 - 0 + "
```

강의를 날로 먹는다는 느낌이 든다면
기분 탓입니다.

Reference

- Extensions (apple document)
- https://github.com/sungkipyung/Swift_Extension_osxdev