

SWIFT 정규표현식

검색부터 사용까지

정규식이란?

- 다 아시죠?
- 스위프트의 정규식 사용을 검색

검색어 : 스위프트 정규식

먼저 `NSRange` 를 이용하여 부분 문자열을 구하게 하는 문자열 확장은 다음과 같다.

code

```
extension String {  
    public func range(with r: NSRange) -> String.Index {  
        let a = index(startIndex, offsetBy: r.location)  
        let b = index(startIndex, offsetBy: r.location + r.length)  
        return a..  
    }  
  
    public subscript(range: NSRange) -> String {  
        return self[self.range(with: range)]  
    }  
}
```

또, 흔히 문자열 전체 범위에 대해서 탐색을 수행하는 경우가 많기 때문에 다음과 같은 확장을 하나 주는 것도 좋다.

code

```
extension String {  
    public var fullRange: NSRange {  
        return NSRange(location: 0, length: characters.count)  
    }  
}
```

검색어 : 스위프트 정규식

Package.swift

즐거 사용하는 텍스트 에디터로 아래와 같은 내용의 `Package.swift` 파일을 작성합니다.

```
import PackageDescription

let package = Package(
    name: "SampleRegex",
    dependencies: [
        .Package(url: "https://github.com/sharplet/Regex.git", majorVersion: 0, minor: 3),
    ]
)
```

이 파일은 예제로 작성할 SampleRegex 프로그램의 패키지 매니저 설정 파일입니다. 이제부터 작성할 우리의 예제 프로그램 패키지는 이름이 "SampleRegex"이고 Regex 버전 0.3이 필요하다는 것을 기술하고 있습니다.

main.swift

예제 스위프트 프로그램을 작성하기 위해 `main.swift` 파일을 Sources 디렉터리를 아래에 만듭니다.

```
$ mkdir Sources
$ touch Sources/main.swift
```

이제 `main.swift` 에 코드를 작성합니다. 패키지 매니저를 사용하는데 필요한 `Package.swift` 와 디렉터리 구조 및 소스 파일의 위치에 관한 자세한 사항은 이전 글 "[스위프트 패키지 매니저](#)"을 참고하시기 바랍니다.

```
import Regex

let greeting = Regex("hello (world|universe)")

print("regex for greeting: \(greeting)")
```

빌드하기

너무 어렵다

- 기본에 충실한 예제를 찾으러 출발

정규식에 사용되는 클래스

사용 : NSRegularExpression

```
NSRegularExpression *regex = [
    NSRegularExpression regularExpressionWithPattern:@"\\b(a|b)(c|d)\\b"
    options:NSRegularExpressionCaseInsensitive
    error:&error];
```

```
let regexForm = try! NSRegularExpression(pattern: regexTry, options: [])
```

```
NSArray *matches = [regex matchesInString:string
                        options:0
                        range:NSMakeRange(0, [string length])];
```

- 정규식 패턴작성 및 매치까지는 어렵지 않다. 매치된 다음이 문제

문제는 형변환

- 정규표현식을 통과한 String 은 NSArray 형태로 변환다
- 정확히는 NSString.Range 의 배열형태
- 결과값을 추출하기 위해서 후처리가 필요

눈에 띄는 첫번째 방식

```
for match in matches {  
    for n in 0..  
match.numberOfRanges {  
        let range = match.rangeAtIndex(n)  
        let r = testStr.startIndex.advancedBy(range.location)..  
testStr.startIndex.advancedBy(range.location+range.length)  
        testStr.substringWithRange(r)  
    }  
}
```

각 매치의 range 를 받고
range의 시작점~range의 길이 를 추출해서
substringWithRange 함수를 사용해 줍니다.

- 구동 방식이 아주 자세히 잘 나와있음. 초보자가 이해하기 좋음
- 결격사유 : 2중 for문 사용

최종 결정 방식

```
let string = self as NSString
return regex.matches(in: self,
                      options: [],
                      range: NSRange(location: 0, length: string.length)).map {
    string.substring(with: $0.range)
}
```

- NSString 의 결과값을 변환하는게 아니라 range 에 사용할 NSString 형태를 선언해줌
- range 값을 쓸때 해당 NSString 형태를 사용하면 결과값이 String 으로 출력됨
- for문 없이도 가능