

JSON Parser 에서 String.Index 사용하기

Drake

목 차

- Parser 진행방법 비교
- 사용한 String.Index 소개
- 작업 진행 순서
- 코드 소개

Parser 진행 방법

- 보통의 방법 : 앞 글자부터 한글자씩 진행

{ "name" : "KIM JUNG", "level" : 5, "married" : true }



- String.Index 를 써보고 싶다....!

{ "name" : "KIM JUNG", "level" : 5, "married" : true }



- , 를 구분점으로 자름



String.Index

- `String.Index(of: target)` : 문자의 인덱스 리턴
- `Range<String.Index>` : 인덱스의 범위
- `String.startIndex ..< String.endIndex`
- `String.Index(before: Index)` : 인덱스의 앞 인덱스 리턴
- `String.Index(Index, offsetBy : Int)` : +Int 인덱스 리턴
- `String[String.Index]` : 해당 인덱스의 Character 리턴

작업 진행 순서

[{ \"level\" : 5, \"married\" : true }, 4, false, \"sd, true, {\"dsf\" : 4} f\"]

1. 배열인지 객체인지 체크 (맨 앞,뒤 글자로 체크)
2. “ 와 } 로 둘러쌓인 심표는 제외하고 심표로 나눠서 배열로 리턴
3. 각 항목이 알맞는 데이터 타입인지 체크
4. 객체생성 - 데이터화
5. 출력

코드소개

보너스

- 몰랐던 함수. String, Array 사용 가능
- `let A = String.removeLast()`
- 첫번째 항목을 기존값에서 삭제, 삭제된 값을 A에 입력
- 두가지 단계가 한줄에 동작됨

Swift 4 정규식 변경점

As of **Swift 4** (Xcode 9), the Swift standard library provides functions to convert between `Range<String.Index>` and `NSRange`.

```
func matches(for regex: String, in text: String) -> [String] {  
    do {  
        let regex = try NSRegularExpression(pattern: regex)  
        let results = regex.matches(in: text,  
                                    range: NSRange(text.startIndex..., in: text))  
        return results.map {  
            String(text[Range($0.range, in: text)!])  
        }  
    } catch let error {  
        print("invalid regex: \(error.localizedDescription)")  
        return []  
    }  
}
```

- https://stackoverflow.com/questions/27880650/swift-extract-regex-matches?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa