**Swift API Design Guidelines**

**Fundamentals**

* 명쾌함이 핵심이다.
* 짧은 것보다 명쾌함이 중요하다.
* 주석을 사용해라.

디테일

* 가능한한 엔티티에 관해 모든 것을 알 수 있도록 서머리를 작성해라.
* 서머리는 가장 중요한 부분이다. 훌륭하게 문서화는 훌륭한 서머리와 다르지 않다.
* 가능하면 완전한 문장보다 온점으로 끝나는 구를 활용하라.
* 함수나 메소드가 무슨 일을 ‘**하는지’**와 ‘**반환하는지’**를표현해라.
* 섭스크립트가 **무엇**에 접근하는지 표현해라.
* 생성자가 **무엇**을 생성하는지 표현해라.

**Naming**

명쾌한 사용성을 지향해라.

* 가능한한 모든 모호함을 피해라.

예를 들어 콜렉션 안에서 특정 인자를 삭제하는 함수를 선언한다면

extension List {

public mutating func remove(at position: Index) -> Element

}

employees.remove(at: x) // 좋음

employees.remove(x) // 나쁨

간단한 사례이지만 레이블이 달린 코드는 x의 위치에 해당하는 인자를 삭제하라는 맥락을 전달한다. 하지만 후자는 캐릭터 ‘x’와 일치하는 값을 삭제하라는 건지 인덱스를 삭제하라는 건지 중의적으로 해석될 수 있다.

따라서 레이블을 활용해 메소드가 정확히 무슨 일을 하는지 구체적으로알려주는 게 좋다.

* 쓸데없는 말은 빼라

간혹 메소드 이름이 해당 메소드가 무엇을 대하는지를 중복적으로 드러내는 경우가 있다. 자바에서는 그런 긴 메소드 이름이 일반적이지만 스위프트에는 아규먼트 레이블과 파라미터가 존재한다. 따라서 메소드 이름이 표현할 영역과 레이블이 표현할 영역을 구분하는 게 좋다. 레이블이 인자 타입의 성격을 충분히 드러낸다면, 메소드에서는 타입에 관한 설명을 하지 않는 게 좋다.