

명명 규칙 (Naming)

- **클래스 및 프로토콜:** `UpperMixedCase`를 사용하며, 공유 라이브러리의 경우 적절한 접두사(3자 이상, 예: `GTM`)를 붙입니다.
- **메서드:** `LowerMixedCase`를 사용합니다. 문장처럼 읽히도록 명명하며, `get` 접두사는 지양합니다.
 - ▶ 예: `- (void)addTarget:(id)target action:(SEL)action;`
- **변수:** `LowerMixedCase`를 사용합니다.
 - ▶ **인스턴스 변수:** 앞부분에 언더스코어를 붙입니다 (예: `_myInstanceVariable`).
 - ▶ **파일 범위/전역 변수:** `g` 접두사를 붙입니다 (예: `gMyGlobalVariable`).
- **상수:** `UpperMixedCase`를 사용하고 접두사를 붙입니다. 구형 파일 내 정적 상수는 `k` 접두사를 사용할 수 있습니다.
- **속성 (Properties):** 불리언 속성의 경우 getter 이름에 `is`를 붙이지만 속성 이름 자체에서는 생략합니다.

포매팅 및 구조 (Formatting)

- **들여쓰기:** 탭을 사용하지 않고 2개의 공백을 사용합니다.
- **줄 길이:** 최대 80자를 권장합니다 (C++ 스타일 가이드와 일관성 유지).
- **메서드 선언:** 속성, 클래스 메서드, 초기화 메서드, 인스턴스 메서드 순으로 배치합니다.
- **가시성:** 인터페이스 헤더의 `ivar`는 `@protected` 또는 `@private`로 명시합니다.
- **NSObject 메서드:** `init`, `description`, `isEqual:`, `hash`, `dealloc` 등 재정의된 메서드는 구현부 상단에 배치합니다.

프로그래밍 관행 (Programming Practices)

- **초기화 (Initialization):** 지정된 초기화 메서드 (`NS_DESIGNATED_INITIALIZER`)를 명확히 식별하고, 부모 클래스의 지정된 초기화 메서드를 재정의합니다.
- **초기화 및 dealloc 내 메시징:** 초기화 메서드와 `dealloc` 내에서 인스턴스 메서드 호출(특히 접근자)을 피하고 `ivar`에 직접 할당합니다.
- **속성 접근:** 중복된 속성 접근을 피하고, 여러 번 사용될 경우 로컬 변수에 할당하여 사용합니다.

- **메모리 관리:** ARC를 사용하더라도 `CoreFoundation` 등 비-Objective-C 객체는 명시적으로 관리해야 합니다.
- **복사 (Copy):** 변경 가능한(`mutable`) 변종이 있는 클래스(`NSString`, `NSArray` 등)를 속성으로 가질 때는 `copy` 속성을 사용하여 방어적 복사를 수행합니다.
- **예외 (Exceptions):** Objective-C 예외를 `@throw` 하지 않습니다. 에러 처리는 `NSError`를 사용합니다.

C 언어 기능 사용

- **매크로:** 가능한 지양하며, 대신 `const` 변수, 열거형(`enum`), C 함수를 사용합니다. 이름은 `SHOUTY_SNAKE_CASE`를 사용합니다.
- **타입 추론:** `__auto_type`은 블록 및 함수 포인터 타입의 로컬 변수에만 제한적으로 사용합니다.
- **포인터:** `nil` 체크를 단순히 메시지 전송을 방지하기 위해 수행하지 않습니다 (Objective-C에서는 `nil`에 메시지를 보내도 안전함).

임포트 (Imports)

- Objective-C 헤더는 `#import`를, C/C++ 헤더는 `#include`를 사용합니다.
- **순서:** 관련 헤더, 운영체제 헤더, 언어 라이브러리 헤더, 기타 의존성 헤더 순으로 정렬합니다.
- **프레임워크 임포트 시 개별 파일이 아닌 우산 (Umbrella) 헤더를 사용합니다** (예: `#import <Foundation/Foundation.h>`).