

## #03. Property

전미정

### 1. Property

property 는 객체외부에서 접근할 수 있는 인스턴스의 속성으로 객체를 캡슐화 하는데 중요한 요소이다. 이는 객체지향 프로그래밍 패러다임에 관련된 중요한 개념으로 객체 내부 변수에 접근할 수 있는 통로를 제공한다.

(접근자 getter, 설정자 setter). Objective-c 는 ARC 에 의해 메모리 관리가 되는데, property 의 속성을 잘 이해하고 사용해야 strong reference cycle, app crash 등의 문제를 피할 수 있다.

Property 에서 변수 something 을 선언하면 자동으로 변수 \_something, setter, getter 코드를 생성해준다. 이때 생성되는 setter, getter 등의 재정의의를 통해 새로운 값 출력/입력 및 사용 확인이 가능하다(custom). Getter/setter/내부변수 선언을 통해 property 와 동일하게 사용가능하다. 하지만 만약 getter method 이름에 "get" 단어를 붙이면 아래와 같은 점 연산자 접근이 불가능하다.(method 호출은 가능).  
"."연산자로 변수에 접근 가능하다.

```
[myTest something] = myTest.something
```

```
[myTest setSomething] = myTest.something = 100;
```

\*self: getter/setter 역할

## 2. Property Option

**readonly** 외부에서 접근가능하지만 값 변경은 불가능(getter 만 구현)

**readwrite** 외부에서 접근, 값 수정 가능(getter, setter 구현, 기본값)

**strong** 강한 참조를 걸겠다. 이 객체가 참조하는 대상 객체의 retain ++(기본값)

**weak** 약한 참조를 걸겠다. 이 객체가 참조하는 대상 객체의 retain count 변화 없음.

앱 크래쉬 방지 가능

**nonatomic** 여러개의 thread 가 해당 객체에 동시에 접근 가능함(안전하지 않을 수 있지만 끊임 없이 감시를 해야하기 때문에 성능저하가 발생할 수 있음)

**atomic** 여러개의 thread 가 하나의 객체에 동시에 접근해 충돌이 일어나는 것을 막기위한 것. Thread lock(디폴트)

**getter/setter** = 자동으로 만들어지지만 method 이름을 변경할 때 활용

**nullable/nonnull** \*something 이라는 객체가 null(비어있음)일 수 있다/없다

