

## 🔥 Day 07

swift 에서 property observer(프로퍼티 옵저버)에 대해서 한번 알아보았습니다.

```
Ready to continue propertyObserver

3 var myAge = 0 {
4     willSet{
5         print("값이 설정될 예정이다. / myAge: \(myAge)")
6     }
7     didSet{
8         print("값이 설정되었다. / myAge: \(myAge)")
9     }
10 }
11
12 print("myAge: \(myAge)")
13
14 myAge = 20
15
16 my Age = 30
17
```

myAge: 0  
값이 설정될 예정이다. / myAge: 0  
값이 설정되었다. / myAge: 20

<호재>

값이 설정될 예정이다 : 20

값이 변경 되었다 : 30

## <참고 자료>

Apple 왈 : Property Observers(프로퍼티 옵저버)를 정의해서 프로퍼티 값의 변경을 모니터링 할 수 있습니다. 프로퍼티 옵저버는 자신이 정의한 "저장 프로퍼티"에 추가 할 수 있으며, super class(부모클래스)를 상속한 프로퍼티에도 추가 할 수 있습니다.

- willSet - 값이 저장되기 직전에 호출됩니다.
- didSet - 새로운 값이 저장된 직후에 호출됩니다.

### ❖ 프로퍼티 옵저버는 언제 사용? 🔗

Swift의 프로퍼티 옵저버를 사용하면 속성이 변경되기 전이나 후에 실행될 기능을 각각 willSet 및 didSet 를 사용하여 연결 할 수 있다.. 하지만 대부분의 경우 프로퍼티 옵저버는 쓰지 않는다 . 그냥 가지고 있으면 좋을 뿐!

일반적으로 속성을 업데이트 한 다음 코드에서 직접 함수를 호출할 수 있다. 그럼 왜 귀찮게 그것을 사용해야 하고, 언제 그것을 사용할까?

프로퍼티 옵저버를 이용하면 가장 좋은 점은 편의성이다. 프로퍼티 옵저버를 사용하면 속성이 변경될 때마다 기능이 실행된다. 속성이 변경될 때마다 함수를 호출해야 한다는 것을 기억해야 한다. 그리고 만약 잊어버리면 코드 안에 찾기 어려운 버그가 있을 것이다.

반면에, didSet 을 이용하여 기능을 속성에 직접 붙이면, 언제든 잘 실행된다!

프로퍼티 옵저버를 이용하면 안 좋은 점은 , 느린 작업을 수행하는 didSet 속성 관찰자를 첨부하면 갑자기 단일 정수를 변경하는데 예상보다 시간이 오래 걸릴 수 있으며 문제가 발생할 수 있다.