

# Property 감시자

## 프로퍼티 감시자

프로퍼티 감시자를 사용하면 프로퍼티 값이 변경될 때 원하는 동작을 수행할 수 있습니다. 값이 변경되기 직전에 `willSet` 블록이, 값이 변경된 직후에 `didSet` 블록이 호출됩니다. 둘 중 필요한 하나만 구현해 주어도 무관합니다. 변경되려는 값이 기존 값과 똑같더라도 프로퍼티 감시자는 항상 동작합니다.

`willSet` 블록에서 암시적 매개변수 `newValue`를 사용할 수 있고, `didSet` 블록에서 암시적 매개변수 `oldValue`를 사용할 수 있습니다.

프로퍼티 감시자는 연산 프로퍼티에 사용할 수 없습니다.

```
struct Money {
    // 프로퍼티 감시자 사용 → 저장 property
    var currencyRate: Double = 1100 {
        willSet(newRate) {
            print("환율이 \(currencyRate)에서 \(newRate)으로 변경될 예정입니다")
        }
        didSet(oldRate) {
            print("환율이 \(oldRate)에서 \(currencyRate)으로 변경되었습니다")
        }
    }

    // 프로퍼티 감시자 사용
    var dollar: Double = 0 {
        // willSet의 암시적 매개변수 이름 newValue
        willSet{
            print("\(dollar)달러에서 \(newValue)달러로 변경될 예정입니다")
        }
        // didSet의 암시적 매개변수 이름 oldValue
        didSet {
            print("\(oldValue)달러에서 \(dollar)달러로 변경되었습니다")
        }
    }

    // 연산 프로퍼티
    var won: Double {
        get {
            return dollar * currencyRate
        }
        set {
            dollar = newValue / currencyRate
        }
    }

    /* 프로퍼티 감시자와 연산 프로퍼티 기능을 동시에 사용할 수 없습니다
    willSet {
        저장되는 값이 바뀔때 사용되는 것에 따른
    }
    */
}
```

```
var moneyInMyPocket: Money = Money()
```

```
// 환율이 1100.0에서 1150.0으로 변경될 예정입니다
```

```
moneyInMyPocket.currencyRate = 1150
```

```
// 환율이 1100.0에서 1150.0으로 변경되었습니다
```

```
// 0.0달러에서 10.0달러로 변경될 예정입니다
```

```
moneyInMyPocket.dollar = 10
```

```
// 0.0달러에서 10.0달러로 변경되었습니다
```

```
print(moneyInMyPocket.won)
```

```
// 11500.0
```

//프로퍼티 감시자의 기능은

//함수, 메서드, 클로저, 타입 등의 외부에 위치한

//지역/전역 변수에도 모두 사용 가능합니다

```
var a: Int = 100 {
    willSet {
        print("\(a)에서 \(newValue)로 변경될 예정입니다")
    }
    didSet {
        print("\(oldValue)에서 \(a)로 변경되었습니다")
    }
}
```

// 100에서 200로 변경될 예정입니다 → willSet

a = 200

// 100에서 200로 변경되었습니다 → didSet