

#9 인스턴스를 만드는 생성자의 이해

✔ 클래스의 생성자

```
class Person(var name:String, val birthYear:Int)
```

이것은 클래스의 '속성'들을 선언함과 동시에 '생성자' 역시 선언하는 방법이다

✔ 생성자 constructor

새로운 인스턴스를 만들기 위해 호출하는 특수한 함수

생성자를 호출하면 클래스의 인스턴스를 만들어 반환받을 수 있다.

```
var a = Person("박보영", 1990)
```

✔ 생성자의 기능

- ① 인스턴스의 속성을 초기화하는 기능
- ② 인스턴스 생성 시 구문을 수행하는 기능

하지만 지난 시간에서는(8강) 인스턴스 생성시 구문을 수행하는 기능은 넣을 수 없었다.

이는 **init**이라는 함수를 통해 수행이 가능하다.

✔ init 함수

패러미터나 반환형이 없는 특수한 함수

생성자를 통해 인스턴스가 만들어 질 때 호출되는 함수이다

✔ 지난 시간에 만들었던 Person 클래스에 Init 함수만 추가해 보자

```
fun main(){
    var a = Person("박보영", 1990)
    var b = Person("전정국", 1997)
    var c = Person("장원영", 2004)
}
class Person (var name: String, val birthYear: Int){
    init{
        println("${this.birthYear}년생 ${this.name}님이 생성되었습니다.")
    }
}
```

- ① 클래스 안에 init을 쓴 후 중괄호를 친다.
- ② init이 호출되는지 여부를 확인하기 위해 속성들을 출력하는 println 구문을 넣어보자
 - ▶ 이때, **this**는 인스턴스 자신의 속성이나 함수를 호출하기 위해 클래스 내부에서 사용되는 키워드이다.
- ③ main 함수에서 Person 인스턴스를 세 개 만들고 실행한다.
 - ▶ 인스턴스 생성을 통해 init이 호출되어 세 개의 인스턴스에 넣은 속성들이 출력되는 것을 볼 수 있다.

그런데 이러한 생성자를 사용할 때 항상 모든 속성을 수동으로 초기화하는 것이 비효율적인 경우도 있다.
예를 들어 Person 클래스로 100명의 사람이 가진 이름과 태어난 년도를 인스턴스로 만들 때 90% 정도가 1997년생인 경우 이를 **기본 값**으로 지정하는 것이 효율적일 것이다.

이럴 때는 **기본 생성자**(클래스를 만들 때 기본으로 선언) 외에 **보조 생성자**(필요에 따라 추가적으로 선언)를 만들 수 있다.

📌 보조 생성자

기본 생성자와는 다른 형태의 생성자를 제공하여 인스턴스 생성 시 편의를 제공하거나 추가적인 구문을 수행하는 기능을 제공하는 역할을 한다.

📌 **보조 생성자** 사용 방법 : **constructor** 라는 키워드를 사용함.

```
fun main(){
    var a = Person("박보영", 1990)
    var b = Person("전정국", 1997)
    var c = Person("장원영", 2004)

    var d = Person("이루다")
    var e = Person("차은우")
    var e = Person("류수정")    — ④
}

class Person (var name: String, val birthYear: Int){
    init{
        println("${this.birthYear}년생 ${this.name}님이 생성되었습니다.")
    }
    constructor(name:String) : this(name, 1997){    — ①②
        println("보조 생성자가 사용되었습니다.")    — ③
    }
}
```

- ① **init**처럼 **constructor**라는 이름의 함수를 만들고 괄호 안에 필요한 패러미터를 추가해준다.(이름만 받는 생성자)
- ② **보조 생성자를 만들 때는 반드시 기본 생성자를 통해 속성을 초기화해줘야 한다. ★**
보조 생성자가 기본 생성자를 호출하도록 하려면 **콜론(:)**을 붙인 후 **this**라는 키워드를 사용하고 기본 생성자가 필요로 하는 패러미터를 괄호 안에 넣어주면 된다.
▶ 여기서 이름은 받은 그대로 넘겨주고 년도는 1997년으로 고정하여 초기화하도록 만든다.
- ③ 중괄호 안에 보조 생성자가 수행되는지를 **println**을 통해 확인해보자
- ④ 그리고 년도 없이 이름만 넣어주면 되는 인스턴스를 생성해보자

기본 생성자와 보조 생성자는 클래스를 사용하는 사람에게 다양한 방법으로 인스턴스를 생성하는 방법을 제공하므로써 편의를 제공하는 기능이다