

#8 기본적인 클래스 사용법

✔ 클래스

‘값’과 그 값을 사용하는 ‘기능’들을 묶어 놓은 것

속성(고유의 특징값) + 함수(기능의 구현)

클래스는 인스턴스를 만드는 틀이라는 것 이해하기

✔ 사람의 이름과 출생년도를 관리하는 클래스 만들기

```
fun main(){  
  
}  
class Person (var name: String, val birthYear: Int)  
    ▶ 이름과 출생년도를 심표로 구분하여 나열한다. 변수 선언과 형태가 같음
```

✔ 인스턴스 만들기

인스턴스 : 클래스를 이용해서 만들어 내는 서로 다른 속성의 객체를 지칭하는 용어

예) Person A instance - 박보영, 1990

Person B instance - 전정국, 1997

Person C instance - 장원영, 2004

```
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)
```

✔ 인스턴스 사용하기

변수명.속성명을 써주면 된다

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

이때 자주 사용하는 기능은 클래스 내에 함수로 넣어주게 된다.

📌 클래스에 함수를 추가하는 방법

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

    println("안녕하세요, ${a.birthYear}년생 ${a.name}입니다")삭제
    a.introduce()
    b.introduce()
    c.introduce()
}

class Person (var name: String, val birthYear: Int){
    fun introduce(){
        println("안녕하세요. ${birthYear}년생 ${name}입니다.")
    }
}
```

③ 인스턴스를 통해 이 함수를 참조하여 호출한다.

① 아까 선언했던 클래스에 중괄호를 추가한다

② 그 안에 4화에서 배운 형태대로 자기소개를 하는 함수를 하나 추가한다
▶ 클래스 내부에서는 속성 이름만 사용한다.

① 아까 선언했던 클래스에 중괄호를 추가한다

② 그 안에 4화에서 배운 형태대로 자기소개를 하는 함수를 하나 추가해주면 된다

③ 인스턴스를 통해 이 함수를 참조하여 호출한다.

코틀린은 객체지향 언어를 기반으로 함수형 언어의 장점을 흡수한 실용적인 언어이다. 클래스에 대한 이해가 높아야 한다