

클래스

구조체의 인스턴스는 값 타입

클래스의 인스턴스는 참조타입

```
struct BasicInformation {  
    var name: String  
    var age: Int  
}
```

```
var yagomInfo: BasicInformation = BasicInformation(name: "yagom", age: 99)  
yagomInfo.age = 100 // 변경 가능  
yagomInfo.name = "Seba" // 변경 가능
```

```
let sebaInfo: BasicInformation = BasicInformation(name: "Seba", age: 99)
```

```
sebaInfo.age = 100 // 변경 불가 오류
```

❌ Cannot assign to property: 'sebaInfo' is a 'let' constant

```
class Person {  
    var height: Float = 0.0  
    var weight: Float = 0.0  
}
```

```
var yagom: Person = Person()  
yagom.height = 123.4  
yagom.weight = 123.4
```

```
let jenny: Person = Person()  
jenny.height = 123.4  
jenny.weight = 123.4
```

구조체와는 다르게 클래스의 인스턴스는
참조타입이므로 클래스의 인스턴스를
상수 let으로 선언해도 내부 프로퍼티 값을 변경할 수 있다

클래스의 인스턴스를 더는 참조할 필요가 없을때
메모리에서 해제되는 과정을 소멸이라고 하는데
소멸되기 직전 deinit라는 메서드가 호출된다.
이렇게 호출되는 deinit 메서드는
디이니셜라이저(Deinitializer)라고 부른다

```
class Person {  
    var height: Float = 0.0  
    var weight: Float = 0.0  
  
    deinit {  
        print("Person 클래스의 인스턴스가 소멸됩니다")  
    }  
}
```

스위프트의 기본 데이터 타입은 모두 구조체
기본 데이터 타입은 모두 값 타입이다

구조체와 클래스는 생긴것은 비슷하지만 용도는 다르다.
프로젝트의 성격에 따라, 데이터의 활용도에 따라,
특정 타입을 구현할 때 구조체와 클래스
둘 중 하나를 선택해서 사용해야 한다

애플 가이드라인에서 다음 조건 중 하나 이상에 해당한다면 구조체를 사용하는 것을 권장한다

- 연관된 간단한 값의 집합을 캡슐화하는 것만이 목적일 때
- 캡슐화한 값을 참조하는 것보다 복사하는 것이 합당할 때
- 구조체에 저장된 프로퍼티가 값 타입이며 참조하는 것보다 복사하는 것이 합당할 때
- 다른 타입으로부터 상속받거나 자신을 상속할 필요가 없을 때