

🔥👤 Day 24

swift 에서 프로토콜 제네릭 associatedtype 에 대해 한번 알아보았습니다.

```
1 // 애완동물 관련 프로토콜
2 protocol PetHaving {
3     // associatedtype 을 통해
4     // 제네릭으로 어떠한 자료형이든 가질수 있다
5     associatedtype T
6     // 애완동물 배열
7     var pets: [T] { get set }
8     // mutating 을 통해 스트럭트에서 멤버 변수 값변경
9     mutating func gotNewPet(newPet: T)
10 }
11 // 익스텐션으로 프로토콜의 메소드 로직구성
12 extension PetHaving {
13     mutating func gotNewPet(newPet: T){
14         self.pets.append(newPet)
15     }
16 }
17 enum Animal{
18     case cat, dog, bird
19 }
20 // 애완동물 프로토콜을 implements 하는 친구 스트럭트
21 struct Friend : PetHaving{
22     // 배열의 자료형이 Animal 이넘으로 사용가능
23     var pets = [Animal]()
24 }
25 // 애완동물 프로토콜을 implements 하는 친구 스트럭트
26 struct Family : PetHaving{
27     // 배열의 자료형이 문자열
28     var pets = [String]()
29 }
30 var myFriend = Friend()
31 myFriend.gotNewPet(newPet: Animal.cat)
32 myFriend.gotNewPet(newPet: Animal.dog)
33 print(myFriend.pets)
34 var myFamily = Family()
35 myFamily.gotNewPet(newPet: "거북이")
36 myFamily.gotNewPet(newPet: "토끼")
37 print(myFamily.pets)
```

→ 제네릭 형태로
값을 갖게 한다.

→ my friend 이 animal 이 들어감