

🔥👤 Day 25

swift 에서 typealias 별칭 에 대해 한번 알아보았습니다.

```
01 protocol Naming {
02     func getName() -> String
03 }
04 protocol Aging {
05     func getAge() -> Int
06 }
07 // typealias 별칭을 통해 프로토콜 별명 설정이 가능
08 typealias Friendable = Naming & Aging
09
10 struct Friend : Friendable {
11     var name : String
12     var age : Int
13     func getName() -> String {
14         return self.name
15     }
16     func getAge() -> Int {
17         return self.age
18     }
19 }
20 // 자료형, 클래스, 스크립트, 클로저 등
21 // 모두 별명 설정이 가능
22 typealias FriendName = String
23
24 var friendName : FriendName = "정대리"
25
26 // 친구 배열을 별명으로 설정하였다.
27 typealias Friends = [Friend]
28
29 var myFriendsArray = Friends()
30
31 // 클로저를 StringBlock 이라는 별명으로 설정하였다.
32 typealias StringBlock = (String) -> Void
33
34 func sayHi(completion : StringBlock){
35     print("안녕하세요?!")
36     completion("오늘도 백코딩 하고 계신가요?!")
37 }
38 sayHi(completion: { result in
39     print("여기서 받을 : ",result)
40 })
```

→ 별칭 짓기.

typealias를 통해, Naming과 Aging를 함께 쓸게요.

→ struct Friend의

Friendable을 상속할 때,

Naming과 Aging가 모두 쓰인다.

- ① 프로토콜을 2개 지정하는 경우 } → typealias를 사용.
② 프로토콜의 이름이 긴지는 경우