

---

# Delegate

---

# Protocol

---

- 프로토콜은 원하는 작업이나 기능을 구현되도록 메서드, 프로퍼티 및 기타 요구 사항의 청사진을 정의합니다.
- 클래스, 구조체, 열거형은 프로토콜을 채택하여, 정의된 요구사항을 구현해야 됩니다.

# Protocol 문법

---

```
protocol SomeProtocol {  
    // protocol definition goes here  
}  
  
struct SomeStructure: SomeProtocol, AnotherProtocol {  
    // structure definition goes here  
}  
  
class SomeClass: SomeSuperclass, SomeProtocol, AnotherProtocol {  
    // class definition goes here  
}
```

# Protocol 문법

---

```
protocol SomeProtocol {  
    // protocol definition goes here  
    func someMethod(_ inputStr:String);  
}
```

```
class SomeClass: SomeSuperclass, SomeProtocol {  
    // class definition goes here  
    func someMethod(_ inputStr:String)  
    {  
        print(inputStr)  
    }  
}
```

# Delegate Design pattern

---

- Delegation is a design pattern that enables a class or structure to hand off (or delegate ) some of its responsibilities to an instance of another type.
- 델리게이트는 클래스나 구조체에서의 일부분의 할 일을 다른 인스턴스에게 대신 하게 하는 디자인 패턴!