可选链

@M了个J

https://github.com/CoderMJLee

http://cnblogs.com/mjios

码拉松

可选链（Optional Chaining）

```
class Car { var price = 0 }
class Dog { var weight = 0 }
class Person {
    var name: String = ""
    var dog: Dog = Dog()
    var car: Car? = Car()
    func age() -> Int { 18 }
    func eat() { print("Person eat") }
    subscript(index: Int) -> Int { index }
}
```

```
var person: Person? = Person()
var age1 = person!.age() // Int
var age2 = person?.age() // Int?
var name = person?.name // String?
var index = person?[6] // Int?
```

```
func getName() -> String { "jack" }
// 如果person是nil，不会调用getName()
person?.name = getName()
```

- 如果可选项为nil，调用方法、下标、属性失败，结果为nil
- 如果可选项不为nil，调用方法、下标、属性成功，结果会被包装成可选项
  - 如果结果本来就是可选项，不会进行再次包装

```
if let _ = person?.eat() { // ()?
    print("eat调用成功")
} else {
    print("eat调用失败")
}
```

```
var dog = person?.dog // Dog?
var weight = person?.dog.weight // Int?
var price = person?.car?.price // Int?
```

- 多个?可以链接在一起
  - 如果链中任何一个节点是nil，那么整个链就会调用失败

小码哥教育 SEEMYGO

可选链

```
var scores = ["Jack": [86, 82, 84], "Rose": [79, 94, 81]]
scores["Jack"]?[0] = 100
scores["Rose"]?[2] += 10
scores["Kate"]?[0] = 88
```

```
var num1: Int? = 5
num1? = 10 // Optional(10)

var num2: Int? = nil
num2? = 10 // nil
```

```
var dict: [String : (Int, Int) -> Int] = [
    "sum" : (+),
    "difference" : (-)
]
var result = dict["sum"]?(10, 20) // Optional(30), Int?
```