

Swift 数据结构

struct

- https://www.tutorialspoint.com/swift/swift_structures.htm
- var/let 声明属性
- 属性在定义时给了初始值, 相当于有了缺省值
- func定义方法
- 如果在方法里修改属性的值要给func加mutating声明
-

enum

- https://www.tutorialspoint.com/swift/swift_enumerations.htm
- case 声明不同的 cases
- enum 只支持 == 和 != 运算符
- func定义方法
- 如果在方法里修改变量(self)的值要给func加mutating声明
-
- 如果使用switch来判断一个enum的value时, 必须覆盖所有的case
-

```

enum Chess{
    case chessA
    case chessB
    case noChess

    func getColor()->String{
        switch self{
            case .chessA:
                return "red"
            case .chessB:
                return "green"
            default:
                return "clear"
        }
    }

    func getChessTitle()->String{
        switch self{
            case .chessA:
                return "A"
            case .chessB:
                return "B"
            default:
                return "N"
        }
    }
}

```

```

    func getShowString()->String{
        return "\(\getChessTitle())^\(\getColor())"
    }

    mutating func next(){
        switch self{
            case .chessA:
                self = .chessB
            case .chessB:
                self = .noChess
            default:
                self = .chessA
        }
    }
}

var chess = Chess.chessA
print(chess.getShowString())

```