

## Piscine iOS Swift - Day 01

### 카드 게임

요약: iOS Swift Piscine's Day 01 de [42](#) 의 내용은 다음과 같습니다.

# 내용물

나	전문	2
II	위탁	삼
III	오늘의 구체적인 규칙	4
IV	소개	5
V	연습 00: 색상과 값	6
VI	연습 01: 카드	7
VII	연습 02: 데크	9
VIII	연습 03: 확장	10
IX	연습 04: 보드	11

# 1장

## 전문

" [e스포츠](#) 현재 프랑스에서 도박 행위로 간주됩니다. 따라서 온라인 도박 위원회에 책임이 있으며 해당 국가에서 대회가 금지될 수 있습니다.

하지만 e스포츠의 상대적 새로움에 대해서는 여전히 용인되고 있다"고 말했다 .

"2억 2,500만 명이 넘는 시청자를 보유한 e스포츠는 가장 큰 프로 스포츠 리그." [출처](#) 에서 번역

"이 대회는 팀 정신, 자기 통제 및 자기 우수성을 옹호합니다. 온라인 스트리밍 모드는 또한 통합 및 문화 간 교류를 지원합니다. 게다가 티켓 판매, 시청각 권리 및 간접적인 관광 혜택을 기반으로 한 대회의 경제적 전망이 중요합니다. 평가 금액은 8억 달러입니다. 2018년 유로.

따라서 경제적으로나 사회적으로 프랑스에 기회가 될 것입니다. "

"Axelle Lemaire가 제안한 디지털 법률 법안이 통과되었습니다. 정부는 법안의 주요 초점과 무엇보다도 eSports의 승인을 공개했습니다." [출처](#) 에서 번역됨

## 제2장

## 위탁

Sauf moins 명시적, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe.  
En aucun cas nous ne porterons Attention 9 ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- 주의 aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices. L'url de votre dépôt GIT는 인트라넷에 따라 결정됩니다.
- Vos exercices seront évalués par vos camarades de Piscine.
- En plus de vos camarades, vous pouvez être évalués par un program appelé la Moulinette. La Moulinette est très stricte dans sa notation car elle est totalement automatisée. Il est donc possible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les mauvaises 놀람.
- Les exercices shell doivent s'exécuter avec /bin/sh.
- Vous ne devez laisser aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices dans votre dépôt de rendu.
- Vous avez une 질문? Demandez à votre voisin de droite. 시논, 에세이즈 avec votre voisin de gauche.
- Toutes les reponses à vos Questions 기술 se trouvent dans les man ou sur 인터넷.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- 레플레키세. 파 피티에, 파 토르, 파 오딘!

## 제3장

# 오늘의 구체적인 규칙

오늘은 특별한 날입니다. 응용 프로그램을 제출하지는 않지만 Swift를 발견하기 위한 몇 가지 연습을 완료합니다.

- 리포지토리의 루트에 각 운동에 대한 폴더를 생성합니다: ex00 ex01 ex02...
- swiftc 로 연습을 컴파일합니다.
- 평가하는 동안 모든 것이 제대로 작동하는지 확인하기 위해 자체 게임 테스트를 제출해야 합니다. 철저해야 합니다. 테스트되지 않은 게임은 무효로 간주됩니다.
- 이전 연습의 파일을 사용할 수 있습니다.

# 제4장

## 소개

Swift는 프로토콜, 확장이라는 두 가지 태그 덕분에 프로토콜에 중점을 둔 다중 패러다임 프로그래밍 언어입니다. 이러한 개념을 이해하려면 객체 개발을 이해해야 합니다.

오늘은 다양한 연습을 통해 고전적인 52장 카드 게임을 만들어 보겠습니다.  
Swift에 익숙해지고 몇 가지 개념을 사용할 수 있습니다.

선언: var, let, type, weak, optional 따라서 var를 올바르게 입력할 수 있습니다.  
능력자.

제어 구조: 루프, 조건, if let을 사용하면 코드를 구조화할 수 있습니다.


클래스: 클래스, func, 오버로드, 재정의, 구조체, 열거형, 상속, 확장, 개체 개발을 위한 변경.

Algo: 익명 함수에 대한 클로저.

이 날은 나머지 물고기 자리를 준비해야 합니다. 최대한 멀리 가십시오.

## 제5장

### 연습 00: 색상과 값

	운동 00
색상 및 가치	
제출 디렉토리 : ex00/ 제출 파일 :	
Color.swift, Value.swift, 테스트 파일 허용 기능 : Aucune	


우선, 우리는 카드의 색상과 가치를 정의할 것입니다.  
클래식 52 카드 게임.

무차별 값 으로 문자열 유형을 사용 하여 열거형 색상 을 만듭니다. 4가지 색상을 표현하게 됩니다. 카드의 가능한 모든 색상을 나타내는 [Color] 유형의 allColors 정적 상수를 추가합니다 .

이제 카드 값을 나타내는 Int 유형 의 brute 값으로 Value 열거형을 만듭니다. 카드의 가능한 모든 값을 나타내는 [Value] 유형의 allValues 정적 상수를 추가합니다 .

## 제6장

### 연습 01: 카드

	운동 01
카드	
제출 디렉토리 : ex01/ 제출 파일 :	
Color.swift, Value.swift, Card.swift, 테스트 파일 허용 기능 : Aucune	

다음을 사용하여 NSObject 를 상속 하는 Card 클래스를 만듭니다 .

- 속성 색상 및 값
- 색상 과 값 을 취하는 빌더
- 속성 var description 으로 재정의 : 문자열 을 쓸 수 있도록 허용합니다.  
카드.
- NSObject 의 isEqual 메서드 재정의

연산자 "=="를 오버로딩하여 isEqual 메소드 처럼 원격으로 작동 하는 2 Card 에서 작동합니다.


다음은 예입니다.



```
> let card1 = Card(c: Color.Spade, v: Value.Ace) card1: Card = { ObjectiveC.NSObject
= {
    isa = __lldb_expr_9.카드
}
색상 = 스페이드 값 = 에이
스
}
> 인쇄(카드1)
(1, Spade) > let
= Card(c: Color.Diamond, v: Value.Two) card2: Card = { ObjectiveC.NSObject =
{ isa = __lldb_expr_9.Card } color = Diamond
    값 = 2
}
> 인쇄(카드2)
(2, 다이아몬드) >
print(card1 == card2) false
```

## 제7장

### 연습 02: 데크

	운동 02
갑판	
제출 디렉토리 : ex02/ 제출 파일 :	
Color.swift, Value.swift, Card.swift, Deck.swift, 테스트 파일	
허용되는 기능 : 배열 인스턴스 메소드	

NSObject를 상속 하는 Deck 클래스 를 생성합니다 .  
[Card] 유형 의 정적 상수를 추가 합니다.

allSpades: 모든 스페이드를 나타냅니다 .

allDiamonds: 모든 다이아몬드를 나타냅니다 .


allHearts: 모든 마음을 나타냅니다 .

allClubs: 모든 클럽을 나타냅니다 .

[Card] 유형 을 가지며 다음이 될 allCards 정적 상수를 추가합니다 .  
52장의 카드 게임에서 가능한 모든 카드 목록.

## 제8장

### 연습 03: 확장


	운동 03
확대	
제출 디렉토리 : ex03/ 제출	
파일 : Color.swift, Value.swift, Card.swift, Deck.swift, 테스트 파일	
허용되는 기능 : arc4random_uniform	

확장은 이미 존재하는 일부 클래스 또는 구조에 코드를 추가하는 데 매우 유용합니다.

이 연습에서는 테이블을 무작위로 혼합하는 메서드를 추가하는 Deck.swift 파일에서 struct Array의 확장을 만들 것입니다.

## 제9장

### 연습 04: 보드

	운동 04
판자	
제출 디렉토리 : ex04/ 제출 파일 :	
Color.swift, Value.swift, Card.swift, Deck.swift, 테스트 파일	
허용되는 기능 : 모든 Array 메소드	

Deck 클래스 에 3개의 [Card] 유형 속성을 추가합니다 .

카드: 덱의 모든 카드를 나타냅니다.

폐기: 폐기 된 모든 카드를 나타냅니다.

outs: 더 이상 카드에 없고 아직 버리지 않은 모든 카드를 나타냅니다.

덱을 분류해야 하는지 또는 혼합해야 하는지를 보여주는 매개변수에 Bool 을 사용하는 빌더를 만듭니다 .

모든 카드 카드를 반환하는 var description: String 속성을 재정의합니다 .

추첨() -> 카드를 만드 시겠습니까? 카드 와 장소 의 첫 번째 카드를 뽑는 방법  
그것은 아웃.

outs 에 속하는 경우 c 카드를 폐기 하는 fold(c: Card) 메서드를 만듭니다 .