



## Piscine iOS Swift - Day 04

트윗

요약: 이 문서에는 iOS Swift piscine of [42](#) 의 Day 04에 대한 주제가 포함되어 있습니다.

# 내용물

나	서문 탐색 ..	2
I.1	.....	2
I.2	전시 통신 ... ..	삼
I.3	우편 운송 .	4
I.4	역사 . ...	4
II	위탁	5
III	소개	6
IV 연습 00: 트윗		7
V 연습 01: APITwitterDelegate		8
VI 연습 02: 요청		9
VII 연습 03: 테이블 보기		11
VIII 연습 04: 연구		12
IX 연습 05: 마무리 작업		13

# 1장

## 전문

다음은 귀환 비둘기의 Wikipedia 페이지입니다.



진정한 메신저 비둘기는 야생 바위비둘기에서 파생된 다양한 집비둘기(*Columba livia domestica*)로, 매우 먼 거리에서도 집으로 돌아가는 능력을 위해 선택적으로 사육됩니다. 바위비둘기는 타고난 귀환 능력을 가지고 있습니다. 즉, 일반적으로 자기 수용을 사용하여 둥지로 돌아갑니다(믿음). 이것은 장거리를 반복해서 집으로 돌아가는 새들로부터 번식을 비교적 쉽게 만들었습니다. 경쟁적인 비둘기 경주에서 새들이 1,800km(1,100마일)에 달하는 비행을 기록했습니다. 보통의 965km(600마일) 거리에서 평균 비행 속도는 약 97km/h(60마일)이며 최고 160km/h(100마일)의 속도는 단거리의 최고 레이서에서 관찰되었습니다.

### 1.1 탐색

비둘기가 운송된 후 한 번도 가본 적이 없는 먼 곳에서 어떻게 돌아올 수 있는지를 발견하려는 목적으로 연구가 수행되었습니다.

대부분의 연구자들은 귀환 능력이 "지도 및 나침반" 모델을 기반으로 하며, 나침반 기능으로 새가 방향을 잡을 수 있고 지도 기능으로 새가 방향을 잡을 수 있다고 믿습니다.

목표 위치(홈 로프트)를 기준으로 위치를 결정합니다. 나침반 메커니즘이 태양에 의존하는 것처럼 보이지만 지도 메커니즘에 대해서는 많은 논란이 있습니다. 일부 연구자들은 지도 메커니즘이 지구 자기장을 감지하는 새의 능력에 의존한다고 믿습니다. 저명한 이론은 새들이 집으로 가는 길을 찾도록 돕기 위해 자기장을 감지할 수 있다는 것입니다. 이전에 과학적 연구에 따르면 비둘기의 부리 위에는 인공 나침반처럼 북쪽으로 정렬된 상태로 남아 있는 많은 수의 철 입자가 발견되어 비둘기가 집을 결정하는 데 도움이 되는 나침반 역할을 한다고 제안했습니다. 그러나 2012년 연구는 이 이론을 반증하여 동물이 자기장을 감지하는 방법에 대한 설명을 찾기 위해 필드를 원래대로 되돌렸습니다. 눈을 포함하고 편축화되는 빛 매개 메커니즘이 다소 조사되었지만 개발은 자기 수용에서 삼차 신경과 관련이 있습니다. Flo riano Papi(이탈리아, 1970년대 초)의 연구와 주로 Hans Wallraff의 최근 연구에 따르면 비둘기도 후각 탐색으로 알려진 대기 냄새의 공간 분포를 사용하여 방향을 지정한다고 제안합니다. 다른 연구에 따르면 귀환 비둘기는 인간이 탐색하는 것과 거의 동일한 방식으로 친숙한 도로 및 기타 인공 지형을 따라 90도 회전하고 습관적인 경로를 따라 시각적 랜드마크를 통해 탐색합니다. US Geological Survey의 Jon Hagstrum의 연구에 따르면 귀환 비둘기는 저주파 초저주파를 사용하여 탐색합니다. 0.1Hz의 낮은 음파가 비둘기 탐색을 방해하거나 방향을 바꾸는 것으로 관찰되었습니다. 비둘기 귀는 너무 작아서 장파를 해석하기에는 너무 작아서 처음 공기를 들이마실 때 비둘기가 원을 그리며 날아가도록 하여 그러한 긴 초저주파를 정신적으로 지도화합니다. 다양한 실험에 따르면 다양한 유형의 귀환 비둘기가 서로 다른 신호에 서로 다른 정도로 의존합니다. 코넬 대학의 찰스 월콧은 한 로프트의 비둘기가 지구의 자기 이상으로 혼동되는 동안 1.6km(1마일) 떨어진 다른 로프트의 새에게는 영향을 미치지 않는다는 것을 증명할 수 있었습니다. 다른 실험에서는 인공 조명으로 하루 중 지각된 시간을 변경하거나 에어컨을 사용하여 비둘기 집의 냄새를 제거하는 것이 비둘기의 집으로 돌아가는 능력에 영향을 미치는 것으로 나타났습니다. GPS 추적 연구에 따르면 중력 이상 현상도 역할을 할 수 있습니다.

## 1.2 전시 통신

새는 제1차 세계 대전 중에 광범위하게 사용되었습니다. 귀환 비둘기 Cher Ami는 중상을 입었음에도 불구하고 12개의 중요한 메시지를 전달한 영웅적인 공로로 프랑스 Croix de guerre 상을 받았습니다. 총돌, 도랑 또는 무선 장애 발생 시 통신 수단으로 폭격기에 실린 귀환 비둘기를 든 승무원. 제2차 세계 대전 중에 아일랜드 패디, 미국 지리학자 조, 영국 엑서터의 메리가 모두 디킨 메달을 받았습니다. 그들은 행동으로 인명을 구한 용감함과 용감함으로 이 상을 받은 32마리의 비둘기 중 하나였습니다.

82마리의 귀환 비둘기는 제2차 세계 대전에서 마켓 가든 작전의 일환으로 제1 공수 사단 신호와 함께 네덜란드로 떨어졌습니다. 비둘기의 다락방은 런던에 위치하여 메시지를 전달하기 위해 390km(240마일)를 비행해야 했습니다.[33] 또한 제2차 세계 대전에서 Confidential Pigeon Service에 속한 수백 마리의 귀환 비둘기가 지역 저항 요원의 정보 매개체 역할을 하기 위해 북서부 유럽으로 공중 투하되었습니다. 새들은 노르망디 침공에서 중요한 역할을 했습니다. 중요한 정보가 적에게 도청될 것을 두려워하여 라디오를 사용할 수 없었기 때문입니다.

## 1.3 우편 운송

비둘기 포스트에서 캐리어 비둘기로 사용할 때 메시지는 얇고 가벼운 종이에 쓰여지고 새의 다리에 부착된 작은 튜브에 감겨집니다. 비둘기는 자신의 집으로 식별된 "정신적으로 표시된" 한 지점으로만 돌아갈 수 있으므로 "비둘기 메일"은 보낸 사람이 실제로 받는 사람의 비둘기를 잡고 있을 때만 작동할 수 있습니다. 흰색 귀환 비둘기는 결혼식, 장례식 및 일부 스포츠 행사에서 비둘기 방출 의식에 사용됩니다. 훈련을 받으면 비둘기는 등에 최대 75g(2.5oz)을 실을 수 있습니다. 독일의 약사인 Julius Neubronner는 긴급 약물을 전달하기 위해 운반비둘기를 사용했습니다. 1977년에는 두 영국 병원 간에 실험실 표본을 운송하기 위해 유사한 운반 비둘기 서비스가 설치되었습니다. 매일 아침 비둘기 바구니가 플리머스 종합 병원에서 데본포트 병원으로 옮겨졌습니다. 그런 다음 새들은 깨지지 않는 유리병을 필요에 따라 플리머스로 다시 배달했습니다. 30마리의 운반비둘기는 1983년 한 병원의 폐쇄로 인해 필요 없게 되었습니다. 1980년대에는 그랑빌(Granville)과 아브랑슈(Avranches)에 위치한 두 개의 프랑스 병원 간에 유사한 시스템이 존재했습니다.

## 1.4 역사

기원전 3000년까지 이집트는 집에서 멀리 떨어져 있을 때 특히 발달된 방향 감각으로 인해 집으로 돌아가는 길을 찾을 수 있는 이 새의 독특한 특성을 이용하여 귀환 비둘기를 비둘기 집으로 사용했습니다. 그런 다음 메시지를 비둘기의 다리에 묶었는데, 비둘기는 풀려났고 원래 동지에 도달할 수 있었습니다. 19세기까지 귀환 비둘기는 군사 통신에 광범위하게 사용되었습니다. 비행 메신저 비둘기의 스포츠는 이미 3000년 전에 확립되었습니다. 그들은 고대 올림픽의 승자를 선언하는 데 사용되었습니다. 메신저 비둘기는 일찍 1150년에 바그다드에서 사용되었으며 나중에는 징기스칸도 사용했습니다. 1167년까지 술탄 누르 앳 딘이 바그다드와 시리아 사이를 정기적으로 운행했습니다. 나일 강 어귀의 다미에타에서 스페인 여행자 Pedro Tafur는 1436년에 처음으로 운반비둘기를 보았지만 새들이 왔다 갔다 하는 것을 상상했습니다.[10]

제노바 공화국은 지중해의 감시탑 시스템에 비둘기 기동을 장착했습니다. 마이소르의 티푸 술탄(1750-1799)도 전령비둘기를 사용했습니다. 그들은 그의 본부였던 스리랑가파르타나에 있는 자미아 마스지드 모스크로 돌아갔다. 비둘기 구멍은 오늘날까지 모스크의 첨탑에서 볼 수 있습니다.

## 제2장

## 위탁

Sauf 모순 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

### 소개

오늘날 인터넷은 우리 삶의 어디에나 존재합니다. 휴대전화의 발명은 관찰을 더욱 의미 있게 만들었습니다.

많은 응용 프로그램에서 데이터를 보내거나 받기 위해 연결해야 합니다.


오늘날 개발자는 일부 [API](#) 를 설계해야 합니다. 데이터를 중앙 집중화하기 위해 여러 장치와 통신할 수 있습니다. 이러한 API 중 일부는 모든 개발자에게 공개됩니다. Twitter의 것이 그 중 하나이며 오늘 우리의 초점이 될 것입니다.

오늘은 [HTTP](#) 를 만드는 법을 배웁니다. iOS에서 Twitter API에 요청하여 트위터 클라이언트 애플리케이션. 테이블 보기 에 트윗을 표시해야 합니다 .

다음은 [Twitter API](#) 에 대한 문서입니다. 이해하려면 이것이 필요합니다 . 트윗을 받는 방법.

## 제4장

### 연습 00: 트윗

	운동 : 00
트위터	
제출 파일 : Swift Standard Library, UIKit	
승인된 기능: 해당 없음	
참고: 해당 사항 없음	

시작하려면 트윗 을 나타내는 모델 이 필요합니다.

다음을 사용하여 구조 트윗 을 만듭니다 .


- let name : 트윗하는 사람의 이름인 String 속성. 트위터.
- let text : 트윗의 내용인 String 속성.

Tweet 구조체 는 CustomStringConvertible 프로토콜 을 구현해야 합니다 .



## 제5장

### 연습 01: APITwitterDelegate

	운동 : 01
APITwitterDelegate	
제출 파일 : Swift Standard Library, UIKit	
승인된 기능: 해당 없음	
참고: 해당 사항 없음	


오늘은 Twitter에 요청하는 데 도움이 되는 컨트롤러 가 필요합니다.  
이 컨트롤러 는 그렇게 하려면 ViewController 의 올바른 메서드를 호출해야 합니다 .  
이것이 프로토콜이 매우 중요한 이유입니다.

두 가지 방법으로 APITwitterDelegate 프로토콜 을 만듭니다 .

- [Tweet] 을 매개변수로 받을 수신된 트윗 을 관리할 것입니다.
- 오류가 발생했을 때 호출되는 또 다른 매개변수로 NSError 를 매개변수로 사용합니다.

## 제6장

### 연습 02: 요청

	운동 : 02
	요청
제출 파일 : Swift Standard Library, UIKit	
승인된 기능: 해당 없음	
참고: 해당 사항 없음	

이제 이전 연습에서 지정한 컨트롤러 를 만들 차례 입니다.

최소한 다음을 갖춘 ApiController 클래스 를 만드십시오 .

- 약한 var 대리자 : APITwitterDelegate? 나중에 될 재산  
우리의 ViewController.
- let 토큰 : Twitter에 대한 연결 토큰이 될 String 속성.
- APITwitterDelegate 를 사용하는 빌더 ? 대리자 에 대한 매개변수로  
및 토큰 에 대한 문자열 입니다 .
- String 을 포함하는 마지막 100개의 프랑스어 트윗을 검색하고 대리자의 일치 메서드 를 호출 하도록 Twitter에 요청해  
야 하는 매개 변수로 String 을 사용하는 메서드.




현재로서는 Twitter에서 "ecole 42"라는 단어가 포함된 마지막 100개의 프랑스어 트윗을 검색  
하고 수신된 모든 트윗을 디버그에 표시할 수 있어야 합니다.



요청의 URL 인코딩을 확인하십시오!

## 제7장

### 연습 03: 테이블 보기

	운동 : 03
	테이블 보기
	제출 파일 : Swift Standard Library, UIKit
	승인된 기능: 해당 없음
	참고: 해당 사항 없음

이제 애플리케이션의 UI를 디자인할 것입니다.


애플리케이션에 탐색 모음 과 테이블 보기 를 추가합니다 . 물론 테이블 은 view 는 "ecole 42"라는 단어가 포함된 마지막 100개의 프랑스어 트윗을 표시해야 합니다.

그렇게 하려면 테이블 뷰 에 연결된 컨트롤러 가 APITwitterDelegate 프로토콜 을 구현해야 합니다 .



## 제8장

### 연습 04: 연구

	운동 : 04
	연구
제출 파일 : Swift Standard Library, UIKit	
승인된 기능: 해당 없음	
참고: 해당 사항 없음	

이 미니 트위터 클라이언트는 검색 필드를 놓치고 있다고 생각하지 않습니까?


"ecole 42"가 아닌 다른 단어를 검색하는 데 도움이 되도록 테이블 보기 에 텍스트 필드 를 추가합니다 .

"Enter"를 누르면 키보드가 사라지고 요청이 실행되어야 합니다.



## 제9장

### 연습 05: 마무리 작업

	운동 : 05
마무리 손질	
제출 파일 : Swift Standard Library, UIKit	
승인된 기능: 해당 없음	
참고: 해당 사항 없음	

구현 방법을 이미 알고 있는 기능으로 이 클라이언트를 보다 효율적으로 만들 때입니다.

- 사용자 지정 셀 을 추가하여 디스플레이를 더 멋지게 만듭니다.
- 콘텐츠 크기에 따른 동적 셀.
- 오류의 경우 경고 를 추가합니다.
- 트윗 날짜를 표시합니다.