

Head First Programming 을 위한 파이썬 트위터 예제 갱신

2014-03-29 강권학

먼저 『Head First Programming』을 구입해주신 독자 여러분께 감사드립니다. 이 책은 여러분이 재미있는 예제를 따라하며 프로그래밍에 익숙해지도록 도와주는 데 초점을 두고 있습니다. 그러나 아쉽게도 책이 출판된 이후 트위터가 인증 방식을 변경했기 때문에 3 장의 실습을 위한 새로운 예제 코드와 실행 방법을 안내하기 위해 작성되었습니다. 이 문서에서는 트위터에서 사용하는 새로운 인증 프로토콜인 OAuth 에 대해 설명하지 않으며, 공개 라이브러리를 이용해 차례대로 따라하면서 쉽게 트위터에 상태 메시지를 올리는 방법에 대해 설명합니다.

트위터에 올리기 위해서는 다음과 같이 3 단계로 진행합니다.

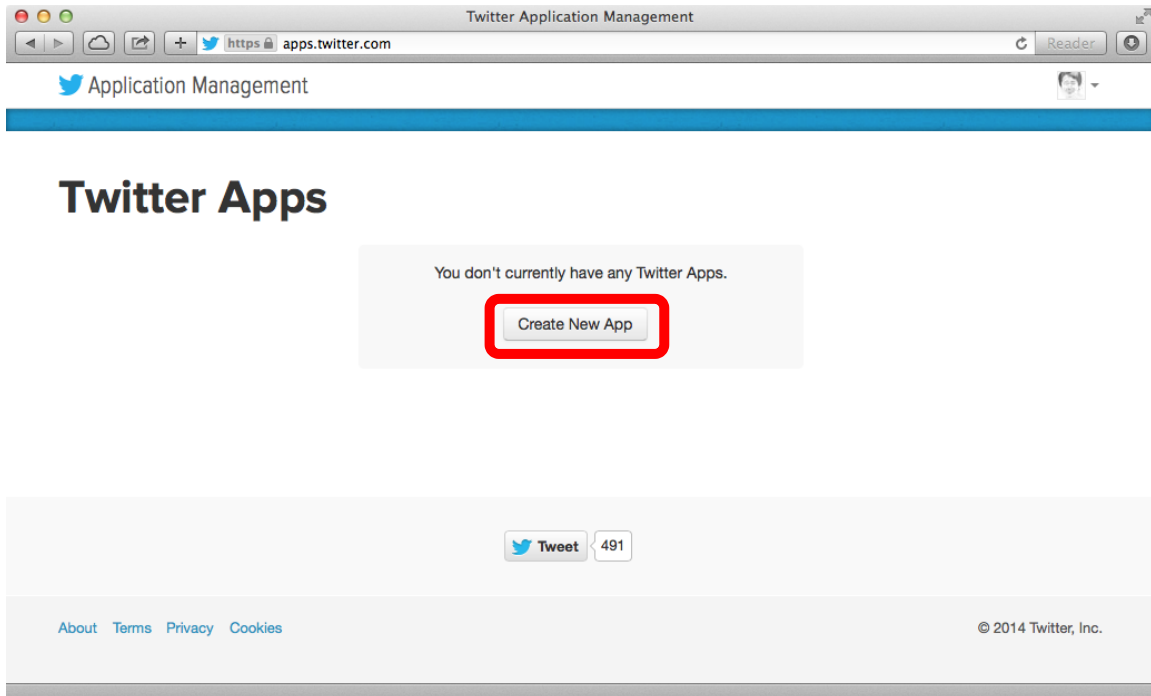
1. 트위터 애플리케이션 생성
2. 트위터 연동을 위한 공개 라이브러리 Twython 설치
3. Twython 을 사용한 Beans-R-Us 코드 작성

1 트위터 앱 생성

트위터의 타임라인을 읽거나 상태 메시지를 올리는 애플리케이션을 만들거나 여러분의 웹사이트에서 트위터와 연동하려면 트위터 개발자 페이지에서 애플리케이션을 생성해야 합니다. 다음의 과정대로 애플리케이션을 생성합니다.

1.1 트위터 개발자 페이지 접속

웹 브라우저를 열어 <https://apps.twitter.com> 에 접속합니다. 아직 로그인하지 않았다면 로그인하세요. 로그인하면 다음과 같이 기존에 등록한 앱들을 볼 수 있으며(그림에서는 등록된 앱이 없습니다), 새로운 앱을 등록할 수 있습니다. [Create New App] 버튼을 누르세요.



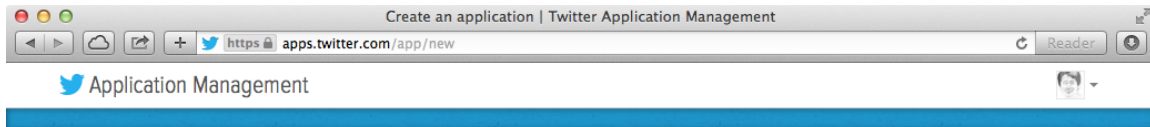
앱 생성 버튼을 누른 후에는 다음과 같이 앱 정보를 입력하는 화면이 나옵니다.

1.2 애플리케이션 기본 정보 입력

입력할 필드는 다음과 같습니다.

- **Name**(필수) : 애플리케이션의 이름을 입력합니다. 이 이름은 애플리케이션 이름으로 트위터 화면에 나오며, 최대 영문 32 자까지 사용할 수 있습니다. 그러나 이미 다른 개발자가 등록한 이름은 사용할 수 없습니다.
- **Description**(필수) : 애플리케이션에 대한 간략히 설명합니다.
- **Website**(필수) : 애플리케이션을 배포하거나 애플리케이션에 대한 정보를 제공하는 웹사이트를 입력합니다.
- **Callback URL** (선택) : 여러분의 웹 서버에서 트위터에 연동할 때 사용합니다. 우리는 애플리케이션을 만들고 있으므로 이 필드를 비워둬야 합니다.

'Yes, I agree' 박스에 체크함으로써 개발자 규칙을 준수한다고 명시한 후에 [Create your Twitter application] 버튼을 누르세요.



Create an application

Application details

Name *

HF-Programming-Twython

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

헤드퍼스트 프로그래밍 Twython 라이브러리를 이용한 트위터 클라이언트 예제

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

http://gonagi.blog.me/

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Rules of the Road

Last Update: July 2, 2013.

Twitter maintains an open platform that supports the millions of people around the world who are sharing and discovering what's happening now. We want to empower our ecosystem partners to build valuable businesses around the information flowing through Twitter. At the same time, we aim to strike a balance between encouraging interesting development and protecting both Twitter's and users' rights.

So, we've come up with a set of Developer Rules of the Road ("Rules") that describes the policies and philosophy around what type of innovation is permitted with the content and information shared on Twitter.

The Rules will evolve along with our ecosystem as developers continue to innovate and find new, creative ways to use the Twitter API, so please check back periodically to see the current version. Don't do anything prohibited by the Rules and talk to us if you think we should make a change or give you an exception.

If your application will eventually need more than 1 million user tokens, or you expect your [embedded Tweets](#) and [embedded timelines](#) to exceed 10 million daily impressions, you will need to talk to us directly about your access to the Twitter API as you may be subject to additional terms. Furthermore, applications that attempt to replicate Twitter's core user experience (as described in Section 1.5 below) will

☒ Yes, I agree

Create your Twitter application

1.3 애플리케이션 권한 설정


애플리케이션을 생성한 후에 다음과 같이 정상적으로 애플리케이션 생성되었음을 알려줍니다. 그런데 여기에서 'Access level' 필드를 보면, 애플리케이션의 권한이 '읽기-전용'으로 되어 있는 것을 알 수 있습니다. 우리는 트위터에서 타임라인을 읽는 것이 아니라, 트위터에 메시지를 포스팅해야 하므로 쓰기 권한이 필요합니다. 'Read-only'라고 되어 있는 부분 오른쪽의 [modify app permission] 링크를 클릭하세요.

The screenshot shows the Twitter Application Management interface in a web browser. The browser's address bar shows the URL `https://apps.twitter.com/app/5974802`. The page title is 'Application Management'. A green notification banner at the top states: 'Your application has been created. Please take a moment to review and adjust your application's settings.'

HF-Programming-Twython

Test OAuth

Details Settings API Keys Permissions

 헤드퍼스트 프로그래밍 Twython 라이브러리를 이용한 트위터 클라이언트 예제
<http://gonagi.blog.me/>

Organization

Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

Application settings

Your application's API keys are used to [authenticate](#) requests to the Twitter Platform.

Access level	Read-only (modify app permissions)
API key	OCfad9PWRxvhxp9vfw9vQ (manage API keys)
Callback URL	None
Sign in with Twitter	No
App-only authentication	https://api.twitter.com/oauth2/token
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token

Application actions

[Delete application](#)

About Terms Privacy Cookies

© 2014 Twitter, Inc.

그러면 다음과 같이 권한 설정 변경 화면으로 들어갑니다. 권한 설정 화면에서 다음과 같이 'Read and Write' 라디오 버튼을 선택하고 [Update settings] 버튼을 누릅니다.

The screenshot shows a web browser window with the address bar displaying 'https://apps.twitter.com/app/5974802/permissions'. The page title is 'HF-Programming-Twython | Twitter Application Management'. The main heading is 'HF-Programming-Twython' with a 'Test OAuth' button. Below the heading are tabs for 'Details', 'Settings', 'API Keys', and 'Permissions'. The 'Permissions' tab is active. The section 'What type of access does your application need?' has a link 'Read more about our Application Permission Model.' and three radio button options: 'Read only', 'Read and Write' (which is selected and highlighted with a red box), and 'Read, Write and Access direct messages'. Below this is a 'Note:' section with text explaining that changes to the permission model will only reflect in access tokens obtained after the change is saved. At the bottom of this section is an 'Update settings' button, also highlighted with a red box. The footer contains links for 'About', 'Terms', 'Privacy', and 'Cookies', and a copyright notice '© 2014 Twitter, Inc.'

*** 주의 ***

권한은 즉시 변경되지 않고 약간의 시간이 소요됩니다. 1~2 분 정도 후에 화면을 다시 갱신하면 권한이 변경된 것을 볼 수 있습니다.

권한이 변경된 후에는 화면에서 [API Keys] 탭을 누릅니다. 다음과 같이 API 토큰과 API 토큰의 비밀번호가 보입니다.

HF-Programming-Twython | Twitter Application Management

apps.twitter.com/app/5974802/keys

Application Management

HF-Programming-Twython

Test OAuth

Details Settings **API Keys** Permissions

Application settings

Keep the "API secret" a secret. This key should never be human-readable in your application.

API key	OCfad9PWRxvhxp9vfw9vQ
API secret	Vc4UMJG39tzjsBD0eoDJWzM4DsBtlfkZXJCxz3wu8
Access level	Read and write (modify app permissions)
Owner	gonagi
Owner ID	30607478

Application actions

[Regenerate API keys](#) [Change App Permissions](#)

Your access token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.

Token actions

[Create my access token](#)

[About](#) [Terms](#) [Privacy](#) [Cookies](#)

© 2014 Twitter, Inc.

그러나 애플리케이션이 트위터에 접근하려면 접근 토큰(Access token)이 필요합니다. [Create my access token] 버튼을 클릭해 접근 토큰을 생성하세요. 접근 토큰이 바로 생성되지 않고 약간의

시간이 지연됩니다. 1~2 분 정도 후에 화면을 갱신해 다음과 같이 접근 토큰이 생성되었음을 확인하세요.

The screenshot shows the Twitter Application Management interface in a web browser. The browser's address bar displays the URL `https://apps.twitter.com/app/5974802/keys`. The page title is "HF-Programming-Twython | Twitter Application Management". The main heading is "HF-Programming-Twython", with a "Test OAuth" button to its right. Below the heading are four tabs: "Details", "Settings", "API Keys", and "Permissions", with "API Keys" being the active tab. The "Application settings" section contains a table with the following data:

API key	OCfad9PWRxvhxp9vfws9vQ
API secret	Vc4UMJG39tzjsBD0eoDJWzM4DsBtlfkZXJCxz3wu8
Access level	Read and write (modify app permissions)
Owner	gonagi
Owner ID	30607478

Below the settings table is the "Application actions" section, which includes two buttons: "Regenerate API keys" and "Change App Permissions".

The "Your access token" section contains a table with the following data:

Access token	30607478-jVOwwjU565mJrXXolkuV9B8fQWH89GKdoUmGVbDQ
Access token secret	QOY9HnZoyKLQctW7kg4fN1vSETIQ7ksn8NJ0FP0XpxUA9
Access level	Read and write
Owner	gonagi
Owner ID	30607478

Below the access token table is the "Token actions" section, which includes two buttons: "Regenerate my access token" and "Revoke token access".

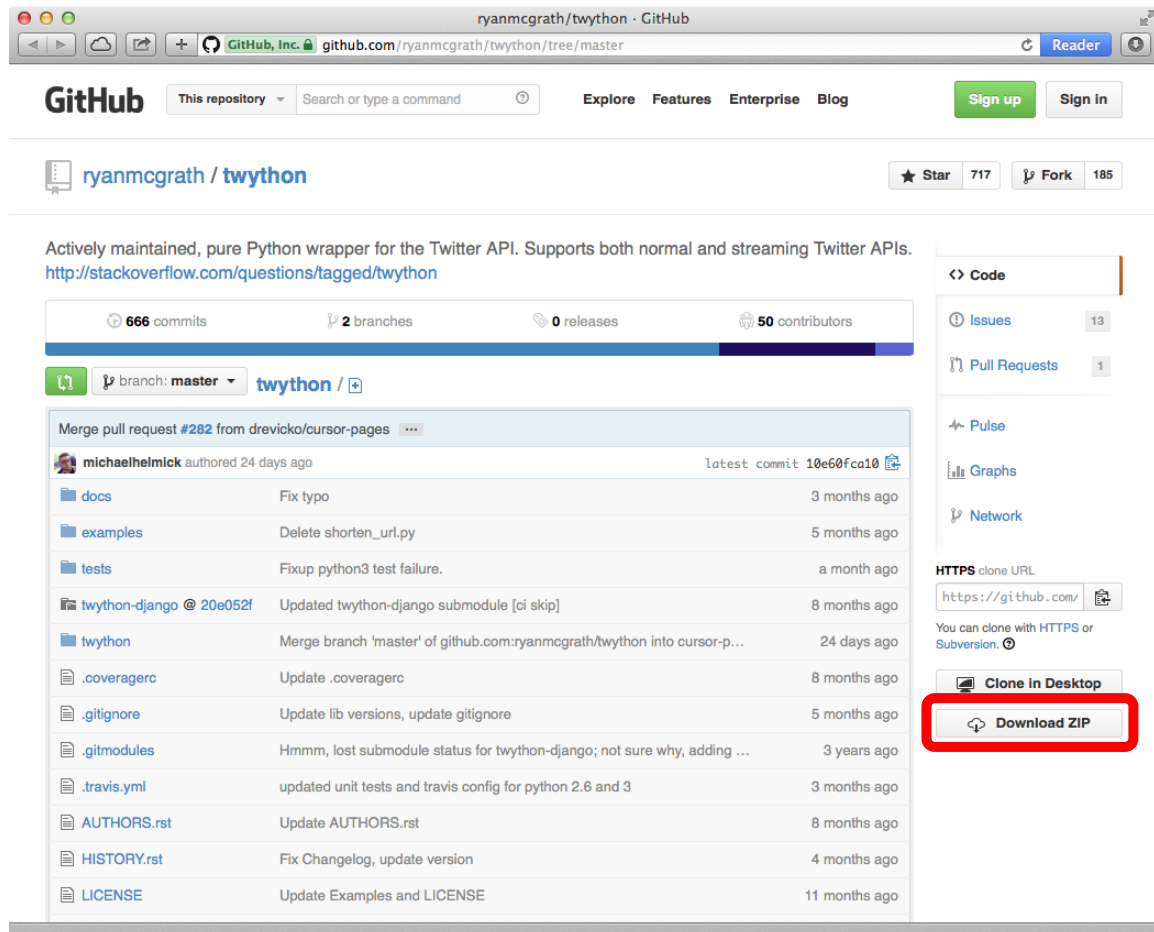
The footer of the page contains links for "About", "Terms", "Privacy", and "Cookies", along with the copyright notice "© 2014 Twitter, Inc."

이제 트위터 홈페이지에서 애플리케이션 생성을 끝냈습니다. 여기에서 만들어진 비밀번호(Secret)은 포털 등에 로그인하기 위해 사용하는 패스워드와 마찬가지로 중요한 비밀이며, 다른 사람과 공유하면 안 됩니다. 애플리케이션을 작성할 때 여기에서 만들어진 API 키, API 키 비밀번호, 접근 토큰, 접근 토큰 비밀번호가 필요합니다.

이제 애플리케이션을 생성했으니, 우리가 사용할 공개 라이브러리 트와이썬(Twython)을 설치하겠습니다.

2 Twython 설치

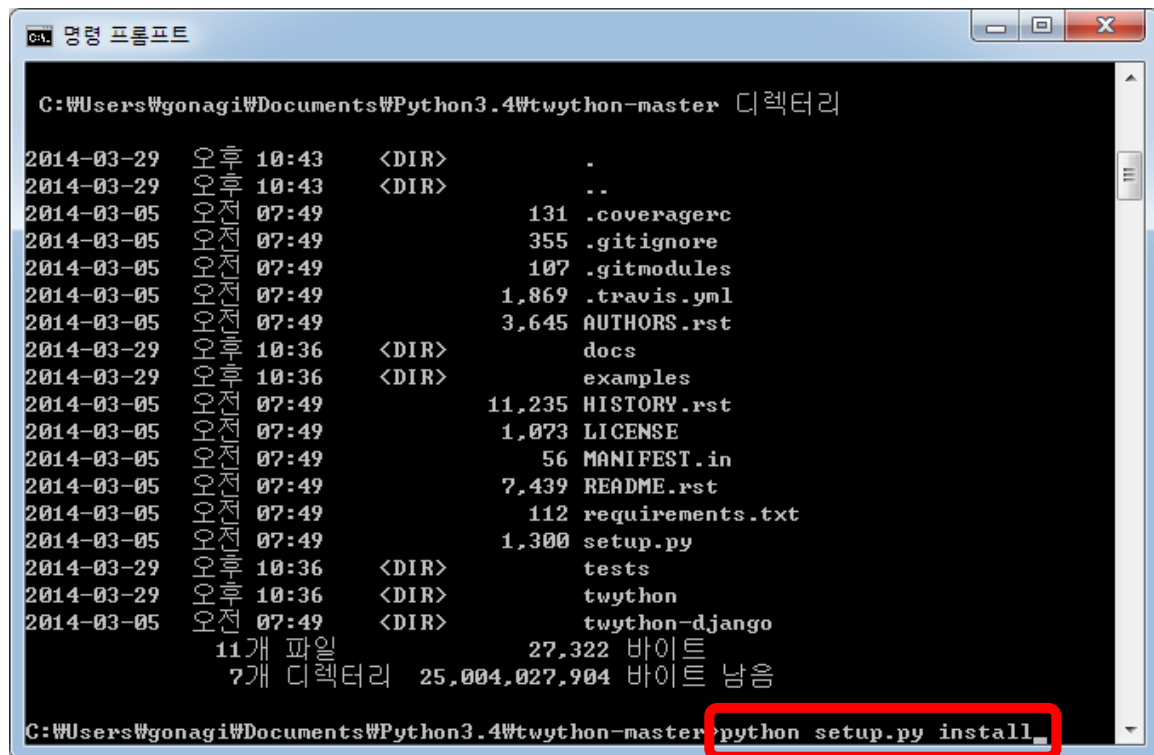
『Head First Programming』에서는 공개라이브러리 tweepy 를 사용했지만, 이 글을 쓰고 있는 현재 파이썬 3.x 를 지원하고 있지 않습니다. 그래서 여기에서는 Twython 을 사용하겠습니다. 웹 브라우저에서 새로 창을 열어 Twython 을 내려받을 수 있는 깃허브 페이지 <https://github.com/ryanmcgrath/twython/tree/master> 로 접속해주세요.



PIP 나 git 을 이용하면 쉽게 설치할 수 있지만 여기에서는 소스 코드 전체를 ZIP 파일로 내려받아 설치하겠습니다. [Download ZIP] 버튼을 클릭해서 소스 코드를 예제 프로젝트를 만들고 있는 작업 디렉터리에 **내려받고 압축을 해제**하세요.

이제 Twython 디렉터리로 이동해 라이브러리를 설치합니다. setup.py 파일이 있는 디렉터리에서 다음과 같이 입력합니다(여기에서는 파이썬 디렉터리가 환경변수 PATH 에 들어 있지만, 그렇지 않은 경우에는 파이썬이 설치된 전체 경로를 써주면 됩니다.).

```
> python setup.py install
```



```
C:\Users\gonagi\Documents\Python3.4\twython-master 디렉터리

2014-03-29 오후 10:43 <DIR> .
2014-03-29 오후 10:43 <DIR> ..
2014-03-05 오전 07:49      131 .coveragerc
2014-03-05 오전 07:49     355 .gitignore
2014-03-05 오전 07:49     107 .gitmodules
2014-03-05 오전 07:49    1,869 .travis.yml
2014-03-05 오전 07:49    3,645 AUTHORS.rst
2014-03-29 오후 10:36 <DIR> docs
2014-03-29 오후 10:36 <DIR> examples
2014-03-05 오전 07:49   11,235 HISTORY.rst
2014-03-05 오전 07:49    1,073 LICENSE
2014-03-05 오전 07:49     56 MANIFEST.in
2014-03-05 오전 07:49    7,439 README.rst
2014-03-05 오전 07:49    112 requirements.txt
2014-03-05 오전 07:49    1,300 setup.py
2014-03-29 오후 10:36 <DIR> tests
2014-03-29 오후 10:36 <DIR> twython
2014-03-05 오전 07:49 <DIR> twython-django
      11개 파일      27,322 바이트
      7개 디렉터리 25,004,027,904 바이트 남음

C:\Users\gonagi\Documents\Python3.4\twython-master>python setup.py install
```

설치가 끝나면 다음과 같이 "Finished ..." 메시지가 나옵니다.

```
CA 명령 프롬프트
Processing requests-2.1.0.tar.gz
Writing C:\Users\gonagi\AppData\Local\Temp\easy_install-s8asjkl7\requests-2.1.0\
setup.cfg
Running requests-2.1.0\setup.py -q bdist_egg --dist-dir C:\Users\gonagi\AppData\
Local\Temp\easy_install-s8asjkl7\requests-2.1.0\egg-dist-tmp-swr79rbz
Adding requests 2.1.0 to easy-install.pth file

Installed c:\python34\lib\site-packages\requests-2.1.0-py3.4.egg
Searching for oauthlib>=0.4.2
Reading https://pypi.python.org/simple/oauthlib/
Best match: oauthlib 0.6.1
Downloading https://pypi.python.org/packages/source/o/oauthlib/oauthlib-0.6.1.ta
r.gz#md5=18f8524c85606380c1ef8a7ff50d69cc
Processing oauthlib-0.6.1.tar.gz
Writing C:\Users\gonagi\AppData\Local\Temp\easy_install-vg3c3rsc\oauthlib-0.6.1\
setup.cfg
Running oauthlib-0.6.1\setup.py -q bdist_egg --dist-dir C:\Users\gonagi\AppData\
Local\Temp\easy_install-vg3c3rsc\oauthlib-0.6.1\egg-dist-tmp-r48k1l3k
zip_safe flag not set; analyzing archive contents...
Adding oauthlib 0.6.1 to easy-install.pth file

Installed c:\python34\lib\site-packages\oauthlib-0.6.1-py3.4.egg
Finished processing dependencies for twython==3.1.2
C:\Users\gonagi\Documents\Python3.4\twython-master>
```

자, 이제 모든 준비가 끝났습니다. 이제 Twython 라이브러리를 이용해 트위터에 포스팅하는 코드를 만들겠습니다.

3 Beans-R-Us 코드 작성

Beans-R-Us 코드는 기존 코드와 동일하지만 트위터에 포스팅하는 `send_to_twitter()` 메서드만 Twython 을 사용하도록 변경하면 됩니다. 이때 다음 코드에서 **API_KEY, API_SECRET, ACCESS_TOKEN, ACCESS_TOKEN_SECRET** 은 여러분이 이 문서의 1 장에서 생성한 애플리케이션의 API 키, API 비밀번호, 접근 토큰, 접근 토큰 비밀번호로 변경해야 합니다. 이 코드는 함께 제공되는 **Ch3-BeansRUs-Twython.py** 파일에 소스코드로 제공됩니다.

```
import sys
from twython import Twython
import time
import urllib

def send_to_twitter(msg):
    API_KEY          = '여러분이 생성한 애플리케이션의 API 키로 변경하세요'
    API_SECRET       = '여러분이 생성한 애플리케이션의 API 비밀번호로 변경하세요'
    ACCESS_TOKEN     = '여러분이 생성한 애플리케이션의 접근 토큰으로 변경하세요'
    ACCESS_TOKEN_SECRET = '여러분이 생성한 애플리케이션의 접근 토큰 비밀번호로 변경하세요'

    twitter = Twython(API_KEY, API_SECRET, ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
```

```

        twitter.update_status(status=msg)

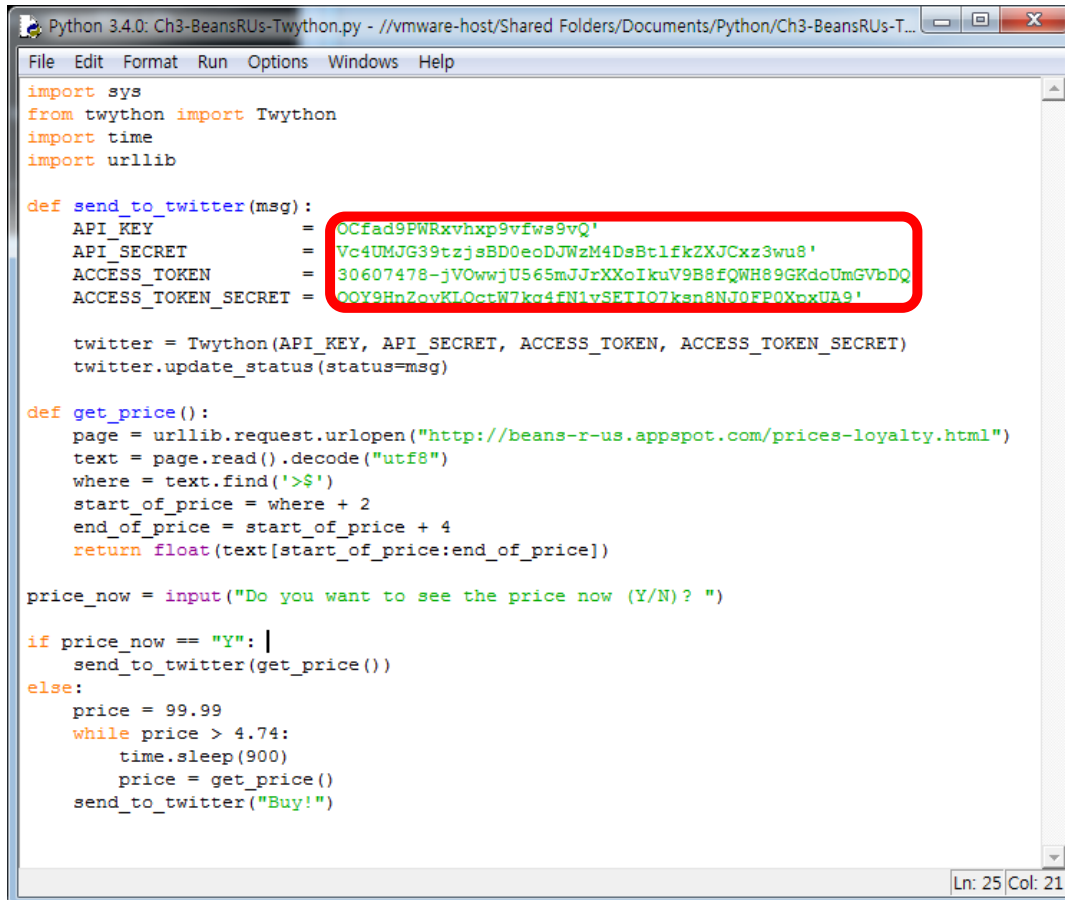
def get_price():
    page = urllib.request.urlopen("http://beans-r-us.appspot.com/prices-loyalty.html")
    text = page.read().decode("utf8")
    where = text.find('>$')
    start_of_price = where + 2
    end_of_price = start_of_price + 4
    return float(text[start_of_price:end_of_price])

price_now = input("Do you want to see the price now (Y/N)? ")

if price_now == "Y":
    send_to_twitter(get_price())
else:
    price = 99.99
    while price > 4.74:
        time.sleep(900)
        price = get_price()
    send_to_twitter("Buy!")

```

코드를 입력한 창 화면은 다음과 같습니다.



```

Python 3.4.0: Ch3-BeansRUs-Twython.py - //vmware-host/Shared Folders/Documents/Python/Ch3-BeansRUs-T...
File Edit Format Run Options Windows Help

import sys
from twython import Twython
import time
import urllib

def send_to_twitter(msg):
    API_KEY = 'oCfad9PWRxvhxp9vfw9vQ'
    API_SECRET = 'Vc4UMJG39tzjsBD0eoDJWzM4DsBtlfkZXJCxz3wu8'
    ACCESS_TOKEN = '30607478-jVOwwjU565mJJrXXoIkuV9B8fQWH89GKdoUmGVbDQ'
    ACCESS_TOKEN_SECRET = 'QQY9HnZovKLOctW7ka4fN1ySETIO7kpn8NJOFP0XpxUA9'

    twitter = Twython(API_KEY, API_SECRET, ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
    twitter.update_status(status=msg)

def get_price():
    page = urllib.request.urlopen("http://beans-r-us.appspot.com/prices-loyalty.html")
    text = page.read().decode("utf8")
    where = text.find('>$')
    start_of_price = where + 2
    end_of_price = start_of_price + 4
    return float(text[start_of_price:end_of_price])

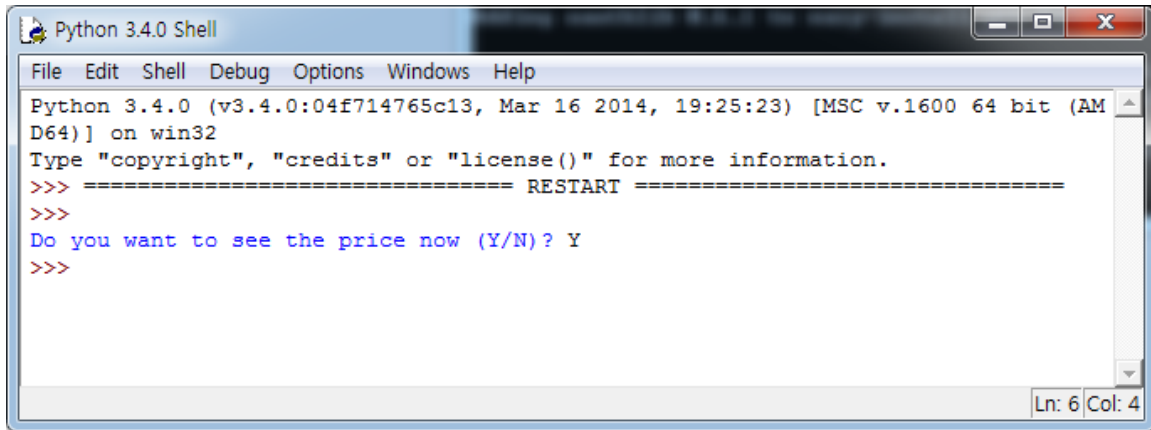
price_now = input("Do you want to see the price now (Y/N)? ")

if price_now == "Y":
    send_to_twitter(get_price())
else:
    price = 99.99
    while price > 4.74:
        time.sleep(900)
        price = get_price()
    send_to_twitter("Buy!")

```

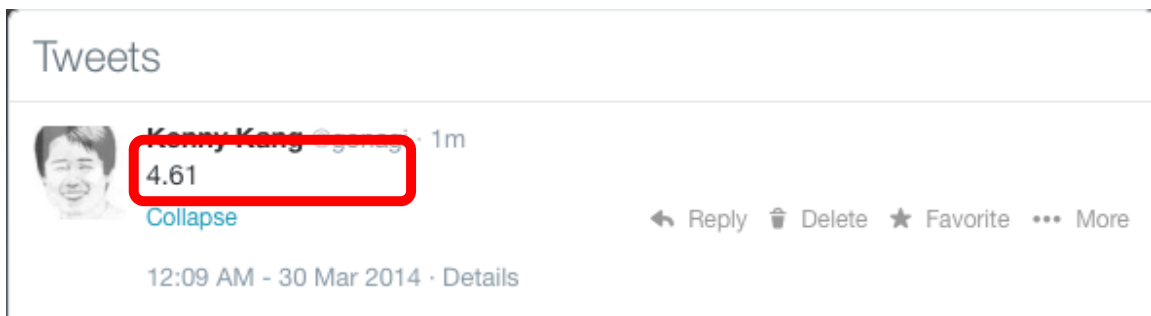
Ln: 25 Col: 21

이제 [F5] 키를 눌러서 실행하면 다음과 같이 물어봅니다.



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:25:23) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Do you want to see the price now (Y/N)? Y
>>>
```

여기에서 Y 를 입력하고 [Enter] 키를 누릅니다. 그리고 나서 여러분의 트위터 페이지를 보면 다음과 같이 메시지를 포스팅한 것을 볼 수 있습니다.



지금까지 Twython 라이브러리를 이용해 트위터에 메시지 포스팅하는 예제를 모두 마칩니다. 감사합니다.