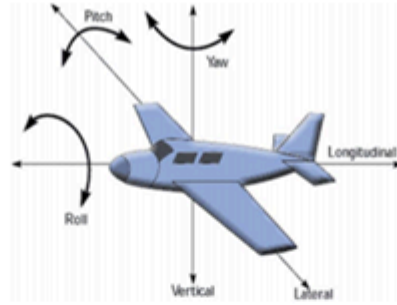


## Tello Drone Python 개발환경 구축

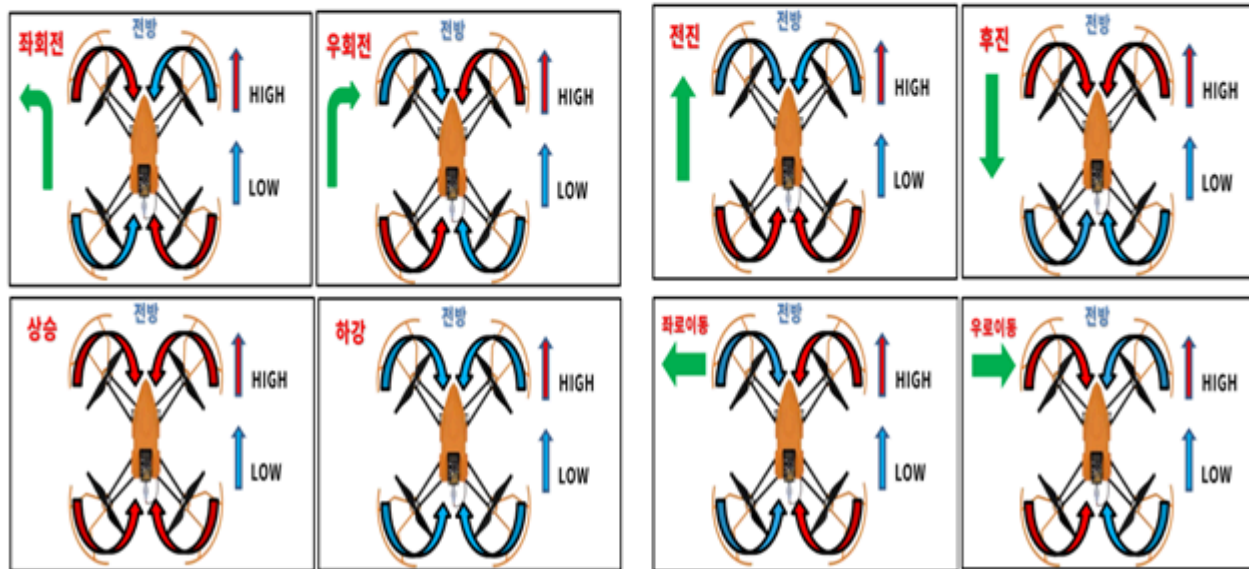
### ✓ 드론의 비행원리

- 드론의 회전과 관계된 역학의 기본 구성 요소 - 요, 피치, 롤

- 각각  $x, y, z$ 축 회전을 담당
- 요, 피치, 롤을 통해서 드론은 앞, 뒤, 좌, 우 이동과 회전을 할 수 있음
- 스로틀(throttle)은 모터의 출력을 결정하여 드론을 위, 아래로 움직일 수 있게 함



요, 피치, 롤



## ✓ 실습 전 준비 - 파이썬 텔로 드론 제어 프로그램 설치 & 텔로 드론

- PC용 와이파이 모듈
- DJI\_SDK - <https://github.com/dji-sdk/Tello-Python>

- 개발자에게 [다른 프로그램에 추가하거나 연결할 수 있는 커스텀 앱을 제작할 수 있는 기능을](#) 제공하는 도구 모음
  - 파이썬 인터프리터(2.7.15 버전)
  - [텔로/텔로 에듀/텔로 텔런트 로보마스터 준비](#)
    - 텔로는 DJI의 기술이 들어간 RYZE ROBOTICS 사의 미니드론
    - 기본적으로 스마트폰으로 조종을 하며, 조종기로도 조종할 수 있음
    - 텔로드론은 [스크래치만 사용가능](#)
    - [텔로 에듀 드론은 스크래치 / 파이썬의 두 가지 언어가 모두 사용가능](#)
-



 텔로	기체 구분	 텔로에듀
80g (프로펠러 및 배터리 포함)	무게	87g (프로펠러 및 배터리 포함)
Scratch	호환 코딩 프로그램	Scratch Python Swift
안됨	군집 비행	지원
Tello DroneBlocks	사용 앱	Tello Edu DroneBlocks
기체 x 1 배터리 x 1 프로펠러 2대 분량 프로펠러 가드 1대 분량	구성품	기체 x 1 배터리 x 1 프로펠러 2대 분량 프로펠러 가드 1대 분량 usb 충전 케이블 x 1 패드 x 4

## ✓ 파이썬과 모듈 설치 및 윈도우 환경 설정

- 파이썬 2.7.15 버전 다운로드(텔로 에두 드론 예제 파일은 2.7.15 버전에서 실행)

- <https://www.python.org/downloads/release/python-2715/>

- [파이썬 과학툴 패키지 Anaconda 설치 - 아나콘다 설치](#)

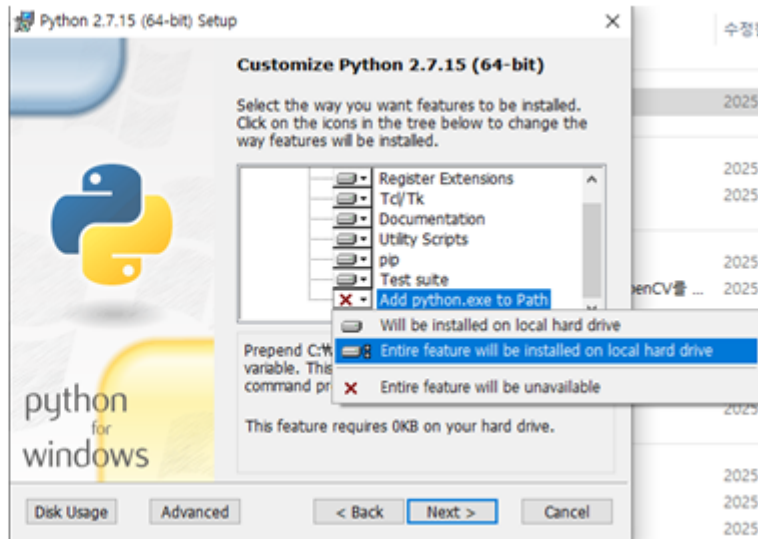
- <https://www.anaconda.com/products/individual-d>

- [파이썬 통합개발 환경 Pycharm 설치 - 파이참 설치](#)

- <https://www.jetbrains.com/ko-kr/pycharm/download/#section=windows>

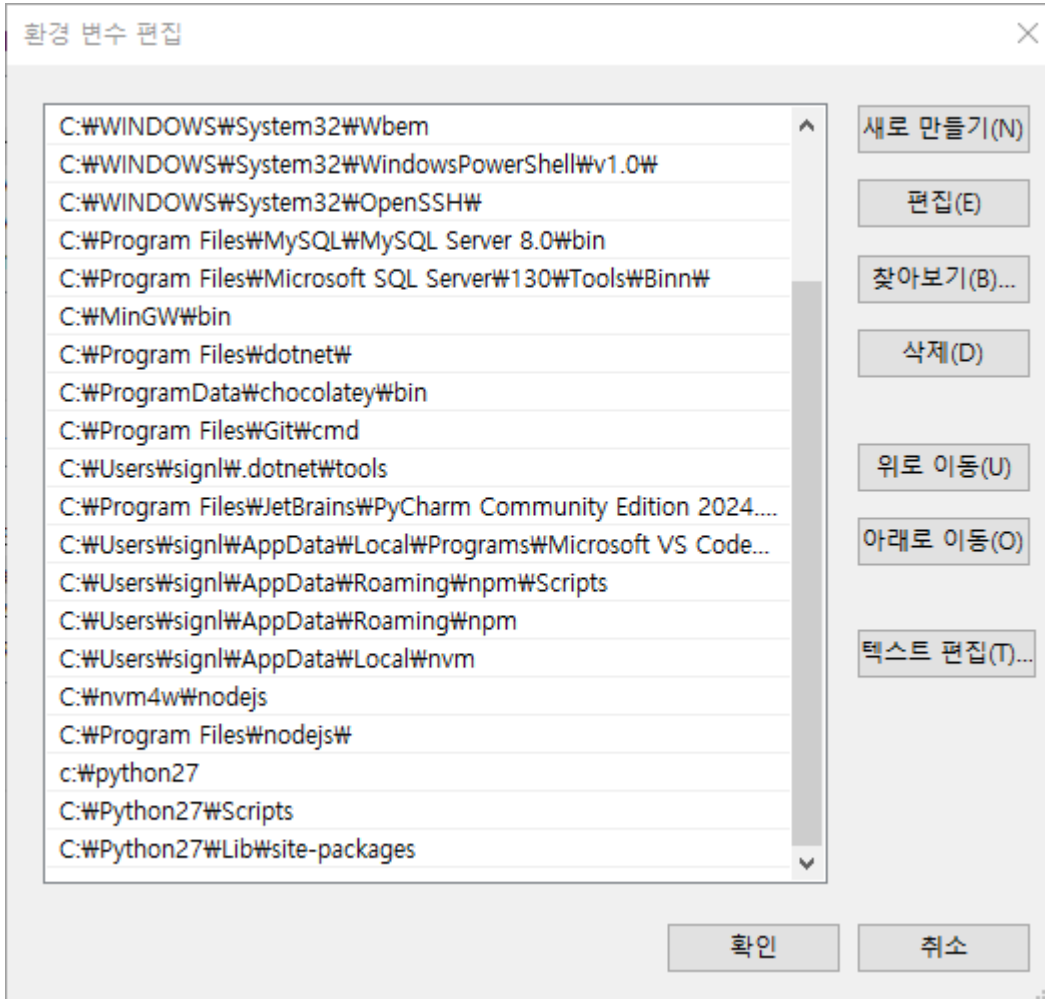
- 다운로드 후 설치

- 'Add python.exe to Path' 설치 경로 옵션에서 (Entire feature will be installed on local hard drive)를 선택하여 환경변수를 등록



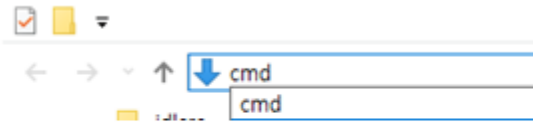
- 환경변수 등록 확인

- c:\python27
- c:\python27\Scripts
- c:\python27\Lib\site\_packages



- 탐색기에서 cmd를 입력한 후 Enter 키 입력 & cmd 창에서 'python'을 실행하여 파이썬 shell 환경이 작동하는지 확인하고 >>exit() 함수를 실행하여 파이썬 셸에서 빠져나옴





```
C:\Windows\System32\cmd.exe - python
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\signl\Downloads>python
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [M
Type "help", "copyright", "credits" or "license" for more in
>>> exit()
```

- [pip\(Python Package Index\)은 파이썬으로 작성된 패키지 소프트웨어를 설치 · 관리하는 패키지 관리 시스템\) 설치 확인 및 업그레이드](#)

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\signl\Downloads>python
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Users\signl\Downloads>pip list
DEPRECATION: The default format will switch to columns in the future. You can use --format=(legacy|columns) (or define a
format=(legacy|columns) in your pip.conf under the [list] section) to disable this warning.
pip (9.0.3)
setuptools (39.0.1)
You are using pip version 9.0.3, however version 25.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\signl\Downloads>python -m pip install --upgrade pip
Cache entry deserialization failed, entry ignored
Collecting pip
  Using cached https://files.pythonhosted.org/packages/27/79/8a850fe3496446ff0d584327ae44e7500daf6764ca1a382d2d02789accf
7/pip-20.3.4-py2.py3-none-any.whl
Installing collected packages: pip
  Found existing installation: pip 9.0.3
  Uninstalling pip-9.0.3:
    Successfully uninstalled pip-9.0.3
Successfully installed pip-20.3.4
You are using pip version 20.3.4, however version 25.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\signl\Downloads>pip list
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no
longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pi
p can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 will remove support
for this functionality.
Package      Version
-----
pip          20.3.4
setuptools   39.0.1

C:\Users\signl\Downloads>

```

- cmd 창을 열어 네트워크 인터페이스와 어드레스를 찾아내는 [netifaces와 netaddr 파이썬 라이브러리 모듈\(군집비행 구현 등 텔로 에듀를 공유기를 통해 컨트롤하고자 할 때 필수 요소\)형을 설치\(제대로 설치가 되면 'Successfully installed' 메시지가 나타남\)](#)

- c:\python -m pip install netifaces
- c:\python -m pip install netaddr

```
C:\Users\signl\Downloads>python -m pip install netaddr
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no
longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pi
p can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 will remove support
for this functionality.
Collecting netaddr
  Using cached netaddr-0.10.1-py2.py3-none-any.whl (2.2 MB)
Collecting importlib-resources; python_version < "3.7"
  Using cached importlib_resources-3.3.1-py2.py3-none-any.whl (26 kB)
Collecting typing; python_version < "3.5"
  Using cached typing-3.10.0.0-py2-none-any.whl (26 kB)
Collecting singledispatch; python_version < "3.4"
  Using cached singledispatch-3.7.0-py2.py3-none-any.whl (9.2 kB)
Collecting contextlib2; python_version < "3"
  Using cached contextlib2-0.6.0.post1-py2.py3-none-any.whl (9.8 kB)
Collecting pathlib2; python_version < "3"
  Using cached pathlib2-2.3.7.post1-py2.py3-none-any.whl (18 kB)
Collecting zipp>=0.4; python_version < "3.8"
  Using cached zipp-1.2.0-py2.py3-none-any.whl (4.8 kB)
Collecting six
  Using cached six-1.17.0-py2.py3-none-any.whl (11 kB)
Collecting scandir; python_version < "3.5"
  Using cached scandir-1.10.0-cp27-cp27m-win_amd64.whl (20 kB)
Installing collected packages: typing, six, singledispatch, contextlib2, scandir, pathlib2, zipp, importlib-resources, n
etaddr
Successfully installed contextlib2-0.6.0.post1 importlib-resources-3.3.1 netaddr-0.10.1 pathlib2-2.3.7.post1 scandir-1.1
0.0 singledispatch-3.7.0 six-1.17.0 typing-3.10.0.0 zipp-1.2.0
C:\Users\signl\Downloads>
```

```

C:\Users\signl\Downloads>pip list
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support pip 21.0 will remove support for this functionality.
Package            Version
-----
contextlib2        0.6.0.post1
importlib-resources 3.3.1
netaddr             0.10.1
netifaces           0.11.0
pathlib2            2.3.7.post1
pip                 20.3.4
scandir             1.10.0
setuptools          39.0.1
singledispatch      3.7.0
six                 1.17.0
typing              3.10.0.0
zipp                1.2.0

```

- 'Successful installed' 메시지가 나오지 않으면 라이브러리 모듈을 수동으로 설치

- <https://www.lfd.uci.edu/~gohlke/pythonlibs>에서 netifaces와 netaddr 모듈을 검색하여 다운로드

- [https://gpc.ac.kr/119/page.do?mnu\\_uid=331&bod\\_uid=88682&cmd=258](https://gpc.ac.kr/119/page.do?mnu_uid=331&bod_uid=88682&cmd=258)
- <https://joyfulgenie.tistory.com/entry/>
- <https://repology.org/project/python:netifaces/informationPython%EC%9D%84-%EC%82%AC%EC%9A%A9%ED%95%98%EC%97%AC-%ED%98%84%EC%9E%AC-%EC%BB%B4%ED%93%A8%ED%84%B0%EC%9D%98-IP-%EC%A3%BC%EC%86%8C%EB%A5%BC-%EC%95%8C%EC%95%84%EB%82%B4%EB%8A%94-%EB%B0%A9%EB%B2%95>

- C:\telloedu 폴더를 생성하여 다운받은 모듈을 복사한 뒤 cmd 창에서 다음 명령문 같이 모듈을 설치

- [C:\telloedu>python -m pip install netifaces-0.10.9-cp27-cp27m-win\\_amd64.whl](#)

- [C:\telloedu>python -m pip install netaddr-0.7.19-py2.py3-none-any.whl](#)

명령 프롬프트

```
C:\Users\signl>python -m pip install netifaces-0.10.9-cp27-cp27m-win_amd64.whl
```

- cmd 창에서 파이썬을 실행하여 [import 명령으로 라이브러리 모듈을 로드하여도 에러가 발생하지 않고 실행되는지 확인](#)

명령 프롬프트 - python

```
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\signl>python
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import netifaces
>>> import netaddr
>>>
```

## ✓ [드론 연결 및 작동](#)

- 1. 드론의 전원을 작동시킨다
- 2. [Tello 드론을 와이파이에서 연결한다](#)
- 3. 드론을 작동시킨다

## ✓ DJI 로보마스터 TT 텔로 텔런트 파이썬 제어 실습

---

- [C:/telloedu](#) 폴더를 생성한 후 다음 사이트에서 [Tello3.py](#)를 다운로드한 후 복사
    - <https://dl-cdn.ryzerobotics.com/downloads/tello/20180222/Tello3.py>
- 

### ✓ *Tello3.py 파일*

```
1 # Tello Python3 Control Demo
2
3 import threading #스레딩을 사용하기 위해서는 'threading' 모듈을 import
4 import socket #소켓 통신을 하기 위해 socket이라는 모듈을 import
5 import sys #sys 라이브러리는 파이썬 인터프리터를 제어하는데 사용되는 기본 모듈
6 import time #시간관 관련된 time, sleep, localtime, strftime 등 다양한 함수를 제공하여 시간 측정, 딜레이, 시간 형식 변환 등에 사용
7
8
9 host = ''
10 port = 9000
11 locaddr = (host,port)
12
13 #Create a UDP socket
14 #AF_INET은 소켓이 통신할 수 있는 주소 유형(이 경우 Internet Protocol v4 주소)을 지정하는 데 사용되는 주소 체계
15 #SOCK_DGRAM을 전달하면 '비 연결 지향형 소켓'이 생성
16 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
17
18 tello_address = ('192.168.10.1', 8889)
```

```

19
20 sock.bind(locaddr)
21
22 def recv():
23     count = 0
24     while True:
25         try:
26             data, server = sock.recvfrom(1518) #UDP 소켓에서 데이터를 수신
27             print(data.decode(encoding="utf-8")) #바이트코드(가상머신이 이해할 수 있는 중간 코드로 컴파일한 것)를 유니코드로 변환
28         except Exception:
29             print ('WnExit . . .Wn')
30             break
31
32
33 print ('WrWnWrWnTello Python3 Demo.WrWn')
34
35 print ('Tello: command takeoff land flip forward back left right WrWn          up down cw ccw speed speed?WrWn')
36
37 print ('end -- quit demo.WrWn')
38
39
40 #recvThread create - Recv 스레드 생성
41 #Thread() : 스레드 생성 함수
42 #Thread(target=함수명, args=(매개변수))
43 #소켓에 연결된 각각의 클라이언트의 메시지를 받을 스레드
44
45 recvThread = threading.Thread(target=recv)
46 recvThread.start() #소켓 프로그래밍에서 데이터 수신과 쓰레드를 생성하는 데 사용
47
48 while True:
49
50     try:
51         msg = input(""); #input("문자열"). input 함수는 사용자로부터 입력을 받는 함수
52
53         # 2.x버전으로 저장하기 위해서는
54         # msg=raw_input("");로 변경후 Tello32.py로 저장
55

```

```
56     if not msg:
57         break
58     if 'end' in msg:
59         print ('...')
60         sock.close()
61         break
62
63     # Send data
64     msg = msg.encode(encoding="utf-8") #encoding 파라미터를 지정해서 어떤 인코딩 방식을 사용하는지 명시적으로 지정
65     sent = sock.sendto(msg, tello_address) # 소켓(접속 되었건 접속되지 않았건)에서 데이터를 접속된 상대방으로 전송하는데 사용되는 함수
66 except KeyboardInterrupt:
67     print ('Wn . . .Wn')
68     sock.close()
69     break
```

- 텔로 에듀의 오른쪽 전원 버튼을 눌러 전원을 켜고 [Wifi 접속창에서 텔로 에듀의 SSID에 접속](#)
- [커맨드 창을 열고 파이썬을 실행](#) 또는 [탐색기에서 파일을 직접 더블클릭](#)
- 커맨드 창에서 [SDK 명령어를 키보드로 입력받아 UDP 통신을 통해 신호를 전송하여 기체를 컨트롤](#)



command → SDK 모드진입, Ok 회신

battery? → 배터리 잔량 회신

sn? → 시리얼번호 회신

sdk? → sdk 버전 회신

wifi? → wifi 강도 회신

takeoff → 이륙, Ok 회신

cw 90 → 90도 우회전 (시계방향), Ok 회신

ccw 90 → 90도 좌회전 (반시계방향), Ok 회신

up 50 → 50cm 상승, Ok 회신

flip f → 전진 플립, Ok 회신

back 50 → 50cm 후진, Ok 회신

land → 착륙, Ok 회신

---

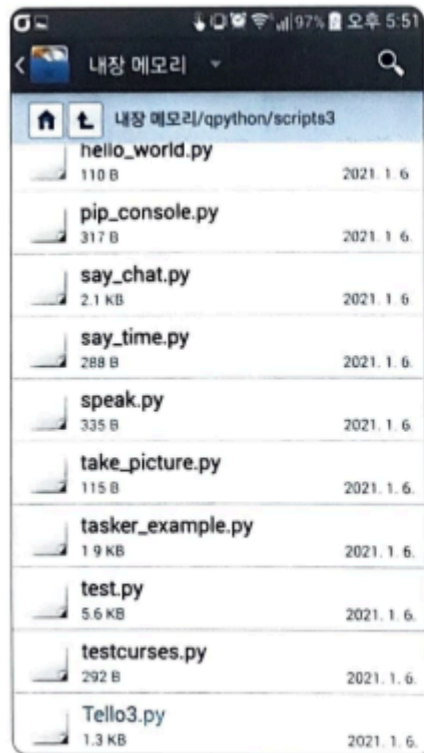
## ✓ Qpython3 앱을 활용한 텔로 에듀 컨트롤

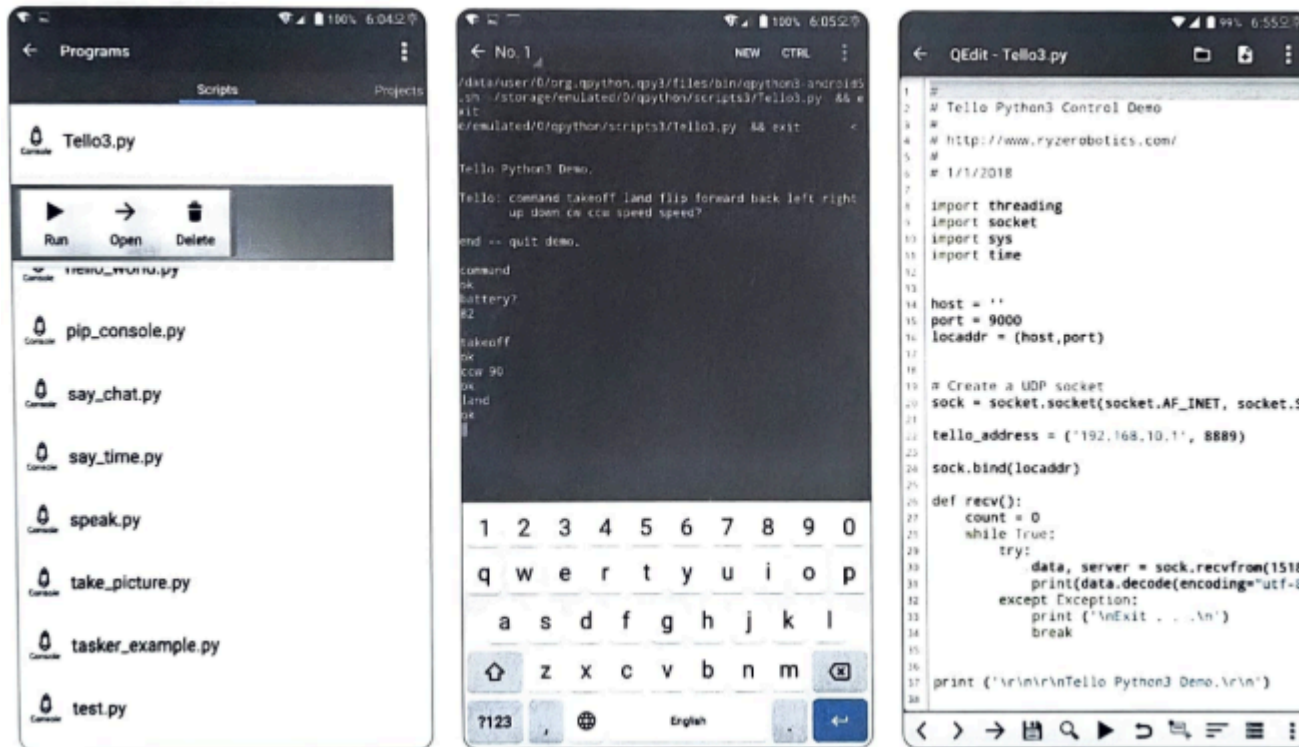


# QPython3 - Python for Android

QPythonLab

광고 포함





## ✓ 단일 텔로 자동비행 컨트롤 예제 다운로드

- [Tello-Python-master.zip](#)를 다운로드한 후 [C:/telloedu](#) 폴더에 복사
  - <https://github.com/dji-sdk/Tello-Python>

## ✓ 텔로 에듀의 자동 비행을 테스트하기 전에 기체의 상태정보 표시방법 파악하기

- 파이썬 라이브러리 제공 사이트에서 접속하여 [curses 모듈](#) 검색하여 [파이썬 2.7용 라이브러리](#)중 자신의 윈도우 버전에 맞는 파일을 선택하여 다운로드하여 [C:/telloedu](#) 폴더에 복사

- curses 모듈은 [이식성 있는 고급 터미널 처리를 위한 사실상의 표준인 curses 라이브러리](#)에 대한 인터페이스를 제공
- curses는 유닉스 환경에서 가장 널리 사용되지만, 윈도우, DOS 및 기타 시스템에서도 사용할 수 있는 버전이 있음
- 이 확장 모듈은 [리눅스와 유닉스의 BSD 변형에서 동작하는 오픈 소스 curses 라이브러리인 ncurses의 API와](#) 일치하도록 설계되었음

- 주소창에 [cmd](#)를 입력하여 커맨드 창을 열고 파이썬 명령어로 [모듈을 설치](#) - c:\telloedu>pip list

- <https://github.com/gtalarico/curses-win/blob/master/bin/curses-2.2%2Butf8-cp27-cp27m-win32.whl>
- <https://docs.python.org/ko/dev/howto/curses.html>
- <https://wikidocs.net/123586>

- [C:/telloedu>python -m pip install curses-2.2.1+utf8-cp27-cp27m-win\\_amd64.whl](#)
- Tello-Pytohn-Master 폴더에서 [tello\\_state.py](#)를 수정 후 실행

- INTERVAL=0.2 -> [INTERVAL=0.05](#)

## ✓ 명령창에서 실행할 것!!!1

```
1 import curses
2 stdscr = curses.initscr()
3 # Clear screen
4 stdscr.clear()
```

```

5 stdscr.addstr(10, 5, "First Position(10, 5)")
6 stdscr.addstr(20, 5, "Second Position(20, 5)")
7 stdscr.addstr(5, 20, "Third Position(5, 20)")
8 stdscr.refresh()
9 stdscr.getkey()

```

```

1 import curses
2 import random
3 import time
4
5
6 def game_over():
7     screen = curses.initscr()
8     width = screen.getmaxyx()[1]
9     height = screen.getmaxyx()[0]
10    size = width * height
11    char = [" ", ".", ":", "^", "*", "x", "s", "S", "#", "$"]
12    b = []
13
14    curses.curs_set(0)
15    curses.start_color()
16    curses.init_pair(1, 0, 0)
17    curses.init_pair(4, 1, 0)
18    curses.init_pair(3, 6, 0)
19    curses.init_pair(2, 4, 0)
20    screen.clear()
21
22    for i in range(size + width + 1):
23        b.append(0)
24
25    while True:
26        for i in range(int(width / 9)):
27            b[int((random.random() * width) + width * (height - 1))] = 65
28        for i in range(size):
29            b[i] = int((b[i] + b[i + 1] + b[i + width] + b[i + width + 1]) / 4)
30            color = (4 if b[i] > 15 else (3 if b[i] > 9 else (2 if b[i] > 4 else 1)))
31            if i < size - 1:

```

```

32         screen.addstr(int(i / width), i % width, char[(9 if b[i] > 9 else b[i])],
33                       curses.color_pair(color) | curses.A_BOLD)
34
35     screen.refresh()
36     screen.timeout(30)
37     if screen.getch() != -1: break
38
39     curses.endwin()
40
41
42 def draw_title(stdscr, start_y, start_x):
43     title = [
44         "          ,----",
45         "        ,/  .`|",
46         "       ,`  .`|:",
47         "      ;    ;    /          ,--",
48         "     .`____,/    ,`____,--",
49         "    |____:    |,` / /|    |__, ,+-. / |",
50         "   ;    |,;    ;| | | ,--.--. `--' | | |",
51         "  `----' | | | ,` /      W ,` ,`| | | ,W" |",
52         "   '    : ;    : / .--. .- | | | | / | |",
53         "   | | | ' | | ' W_W/: . . | | : | | | |",
54         "   '    : |;    : | ,W" .--.; | ' : |__ | | | /",
55         "   ;    |,` | , ; / / ,. | | | ' | | | |__",
56         "   '____' ____ ; : .` W; : ; | | /",
57         "   | , .-./ | , / '----",
58         "   `--'____' ____`--' "
59     ]
60     for i, line in enumerate(title):
61         try:
62             stdscr.addstr(start_y + i, start_x, line)
63         except curses.error:
64             pass
65
66     return title
67
68

```

```
69 def draw_menu(stdscr, selected_row_idx, start_y, start_x):
70     menu_items = ['Continue', 'New Game', 'Settings', 'Quit']
71     menu_width = 20
72     for idx, item in enumerate(menu_items):
73         x = start_x + (menu_width - len(item)) // 2
74         y = start_y + idx
75         if idx == selected_row_idx:
76             stdscr.attron(curses.color_pair(1))
77             stdscr.addstr(y, x, item)
78             stdscr.attroff(curses.color_pair(1))
79         else:
80             stdscr.addstr(y, x, item)
81
82     # Draw menu border
83     stdscr.box()
84     stdscr.addch(start_y - 1, start_x - 1, curses.ACS_ULCORNER)
85     stdscr.addch(start_y - 1, start_x + 20, curses.ACS_URCORNER)
86     stdscr.addch(start_y + 4, start_x - 1, curses.ACS_LLCORNER)
87     stdscr.addch(start_y + 4, start_x + 20, curses.ACS_LRCORNER)
88     for i in range(4):
89         stdscr.addch(start_y + i, start_x - 1, curses.ACS_VLINE)
90         stdscr.addch(start_y + i, start_x + 20, curses.ACS_VLINE)
91     for i in range(menu_width):
92         stdscr.addch(start_y - 1, start_x + i, curses.ACS_HLINE)
93         stdscr.addch(start_y + 4, start_x + i, curses.ACS_HLINE)
94
95     # draw_road(stdscr, start_y + len(menu_items), start_x, menu_width)
96
97
98 def draw_road(stdscr, y, x, menu_width):
99     height, width = stdscr.getmaxyx()
100     line_length = int(width * 0.4)
101
102     road_start_x = x - (line_length - menu_width) // 2
103     road_end_x = road_start_x + line_length
104
105     for i in range(3):
```

```
106     stdscr.addch(y + i, road_start_x + i, 'WW')
107
108     for i in range(3):
109         stdscr.addch(y + i, road_end_x - i - 1, '/')
110
111
112 def new_game(stdscr, title, start_y, start_x):
113     height, width = stdscr.getmaxyx()
114     title_height = len(title)
115
116     for i in range(title_height + start_y + 20):
117         stdscr.clear()
118         stdscr.box()
119         for j, line in enumerate(title):
120             y = start_y + title_height - i + j
121             if 0 <= y < height:
122                 try:
123                     stdscr.addstr(y, start_x, line)
124                 except curses.error:
125                     pass
126         stdscr.refresh()
127         time.sleep(0.1)
128
129
130 def main(stdscr):
131     curses.curs_set(0)
132     curses.init_pair(1, curses.COLOR_BLACK, curses.COLOR_WHITE)
133
134     selected_row_idx = 0
135
136     while True:
137         stdscr.clear()
138         height, width = stdscr.getmaxyx()
139
140         title_start_y = height // 10
141         title_start_x = (width - 50) // 2 # 96 is the width of the title ASCII art
142
```



```
143     # menu_start_y = title_start_y + 10
144     menu_start_y = int(height // 1.3)
145     menu_start_x = (width - 20) // 2
146
147     # draw_title(stdscr, title_start_y, title_start_x)
148     title = draw_title(stdscr, title_start_y, title_start_x)
149     draw_menu(stdscr, selected_row_idx, menu_start_y, menu_start_x)
150
151     key = stdscr.getch()
152
153     if key == curses.KEY_UP and selected_row_idx > 0:
154         selected_row_idx -= 1
155     elif key == curses.KEY_DOWN and selected_row_idx < 3:
156         selected_row_idx += 1
157     elif key == 10: # Enter key
158         if selected_row_idx == 1:
159             new_game(stdscr, title, title_start_y + 20, title_start_x)
160         elif selected_row_idx == 2:
161             game_over()
162             break
163         elif selected_row_idx == 3: # Quit option
164             break
165         # Here you would typically handle other menu options
166
167     stdscr.refresh()
168
169
170 curses.wrapper(main)
```

 숨겨진 출력 표시

✓ *tello\_state.py 파일*

```
1 #tello_state.py
2
3 import socket
4 from time import sleep
5 import curses # curses 확장 모듈을 사용하여 텍스트 모드 디스플레이를 제어
6 #https://python.flowdas.com/howto/curses.html
7
8 INTERVAL = 0.2
9 #INTERVAL= 0.05
10
11 def report(str):
12     stdscr.addstr(0, 0, str) #0번째 줄 0번째 열부터 str라는 문자열을 출력하라는 의미
13     stdscr.refresh() #화면을 갱신하기 위해 창 객체의 refresh() 메서드를 호출
14
15 if __name__ == "__main__":
16     stdscr = curses.initscr() #curses로 터미널을 제어하려면 먼저 initscr()을 호출하여 터미널 객체 stdscr을 생성해야함
17     curses.noecho() #키보드 입력값이 화면에 보이지 않도록 설정
18
19     #응용 프로그램은 또한 일반적으로 Enter 키를 누르지 않아도 즉시 키에 반응해야 함
20     #이것을 일반적인 버퍼 입력 모드와 대비하여 cbreak 모드라고 함
21     curses.cbreak()
22
23     local_ip = ''
24     local_port = 8890
25     socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # socket for sending cmd
26     socket.bind((local_ip, local_port))
27
28     tello_ip = '192.168.10.1'
29     tello_port = 8889
30     tello_adderss = (tello_ip, tello_port)
31
32     socket.sendto('command'.encode('utf-8'), tello_adderss)
33
34     try:
35         index = 0
36         while True:
37             index += 1
```

```

38         #recvfrom() : 클라이언트로부터 데이터 수신(데이터 및 주소 정보 반환)
39         response, ip = socket.recvfrom(1024) #좌측에 콤마로 구분된 변수들을 나열하고 우측에 값을 순서대로 나열
40         if response == 'ok':
41             continue
42         out = response.replace(':', ';', '\n')
43         out = 'Tello State:\n' + out
44         report(out) #def report(str):
45         sleep(INTERVAL)
46     except KeyboardInterrupt:
47         curses.echo() #키보드 입력값이 화면에 출력되도록 설정
48         curses.nocbreak() #cbreak 모드를 해제
49         curses.endwin() # endwin() 함수를 호출하여 터미널을 원래 작동 모드로 복원

```

---

## ✓ PC 설정 및 코딩

- PC에서 WiFi 접근이 가능해야 함
    - Tello의 전원을 넣고 PC의 WiFi를 Tello와 연결시킴(PC에서 와이파이로 Tello 드론을 접속)
  - 파이썬 IDE(Pycharm/vscode 등)을 이용하여 코딩후 실행
- 

## ✓ *tello\_control\_test\_1.py 파일*

```

1 #tello_control_test_1.py - 현재 텔로 드론과 연결 상태 정보 출력
2 #드론과 와이파이 연결
3
4 import socket #소켓 통신을 하기 위해 socket이라는 모듈을 import

```

```

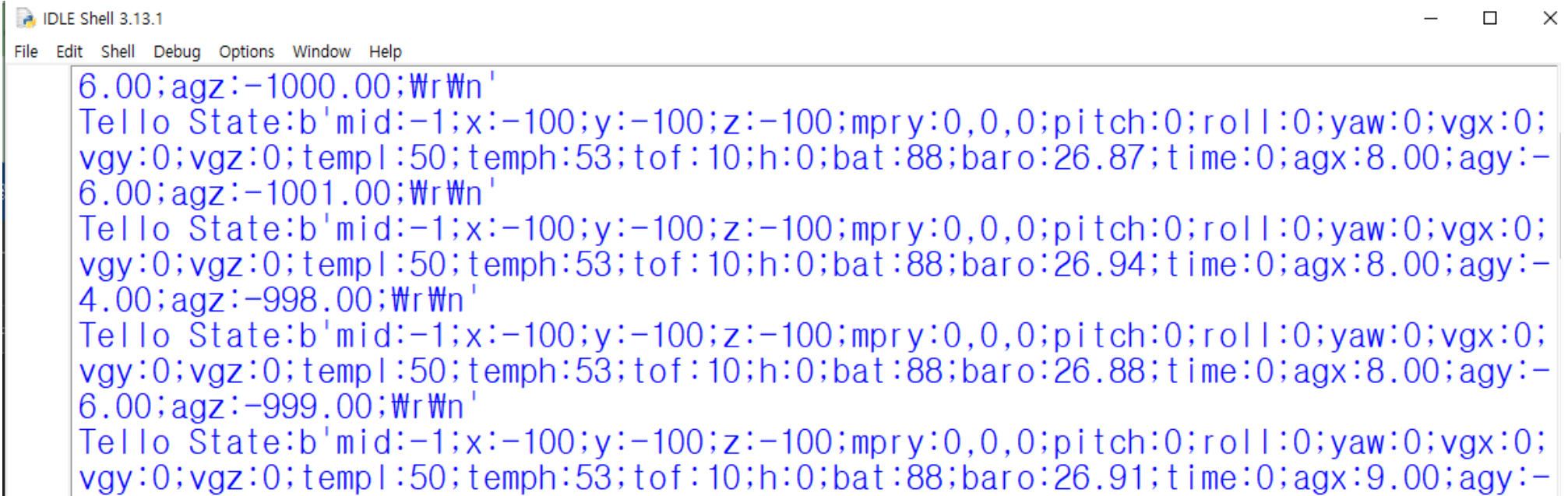
5 from time import sleep #time 라이브러리의 sleep 함수를 사용하면 일정 시간동안 프로세스를 일시정지
6
7 #__name__ 변수는 현재 모듈의 이름을 담고 있는 내장 변수
8 #모듈이 직접 실행되었는지(import 되었는지 아닌지) 판단할 때 __name__ 변수의 값을 사용
9 #일반적으로, 모듈은 직접 실행되거나 다른 모듈에서 import 되어 사용됨
10 #만약 모듈이 직접 실행되면, __name__ 변수는 문자열 "__main__"이 할당됨
11 #반대로, 모듈이 import 되어 사용될 때는, __name__ 변수는 해당 모듈의 이름(파일명)이 할당됨
12 #따라서, __name__ 변수의 값을 "__main__"과 비교하면 현재 모듈이 직접 실행되는지(import 되는지)를 판단할 수 있음
13 #따라서 코드를 if __name__ == "__main__"로 감싸면, 해당 파일이 모듈로 사용될 때는 실행되지 않고, 직접 실행될 때만 실행됨
14
15 if __name__ == "__main__":
16     local_ip = ''
17     local_port = 8890 # 입력 포트 저장
18
19     #파이썬에서 비연결성 UDP 소켓을 생성
20     #socket.AF_INET - IPv4 주소를 의미
21     #socket.SOCK_DGRAM - 비연결성 소켓 유형
22     #socket 모듈 - BSD 소켓 인터페이스에 대한 액세스를 제공
23     #socket() 함수 - 소켓 객체를 반환하고, 이 소켓 객체의 메서드는 다양한 소켓 시스템 호출을 구현
24
25     socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # socket for sending cmd
26     socket.bind((local_ip, local_port)) #소켓 주소 정보 할당
27
28     tello_ip = '192.168.10.1' # Tello는 AP(Access Point)이며 주소는 ip 192.168.10.1을 사용
29
30     #Send Command & Receive Response
31     #명령어는 UDP Port 8889를 통하여 보내줄 수 있으며 첫번째는 'command' 명령을 보내어 SDK 모드가 되도록 함
32     tello_port = 8889 # port 출력(저장)
33     tello_adderss = (tello_ip, tello_port)
34     socket.sendto('command'.encode('utf-8'), tello_adderss) # SDK Mode 진입
35
36     try:
37         index = 0
38         while True:
39             outStr=""
40             response, ip = socket.recvfrom(1024) #클라이언트로부터 데이터 수신(데이터 및 주소 정보 반환)
41             if response == 'ok':

```

```

42         continue
43         outStr = 'Tello State:' + str(response)
44         print(outStr)
45         sleep(0.2)
46     except KeyboardInterrupt:
47         pass # "아무것도 하지 않는" 명령어, 마치 #(코멘트) 처리되어 있는 라인과 같은 효과를 가짐

```



```

IDLE Shell 3.13.1
File Edit Shell Debug Options Window Help
6.00;agz:-1000.00;WrWn'
Tello State:b'mid:-1;x:-100;y:-100;z:-100;mpry:0,0,0;pitch:0;roll:0;yaw:0;vgx:0;
vgy:0;vgz:0;templ:50;temph:53;tof:10;h:0;bat:88;baro:26.87;time:0;agx:8.00;agy:-
6.00;agz:-1001.00;WrWn'
Tello State:b'mid:-1;x:-100;y:-100;z:-100;mpry:0,0,0;pitch:0;roll:0;yaw:0;vgx:0;
vgy:0;vgz:0;templ:50;temph:53;tof:10;h:0;bat:88;baro:26.94;time:0;agx:8.00;agy:-
4.00;agz:-998.00;WrWn'
Tello State:b'mid:-1;x:-100;y:-100;z:-100;mpry:0,0,0;pitch:0;roll:0;yaw:0;vgx:0;
vgy:0;vgz:0;templ:50;temph:53;tof:10;h:0;bat:88;baro:26.88;time:0;agx:8.00;agy:-
6.00;agz:-999.00;WrWn'
Tello State:b'mid:-1;x:-100;y:-100;z:-100;mpry:0,0,0;pitch:0;roll:0;yaw:0;vgx:0;
vgy:0;vgz:0;templ:50;temph:53;tof:10;h:0;bat:88;baro:26.91;time:0;agx:9.00;agy:-

```

- Send Command & Receive Response

- Tello는 AP(Access Point)이며 주소는 ip 192.168.10.1을 사용
- 명령어는 UDP Port 8889를 통하여 보내줄 수 있으며 첫번째는 'command' 명령을 보내어 SDK 모드가 되도록 함

- [Receive Tello State](#)

- PC에 [UDP Server 0.0.0.0 UDP port: 8890](#)를 설치하여 Tello로 부터 오는 메시지를 받을 수 있음

- [Receive Tello Video Stream](#)

- PC에 [UDP Server 0.0.0.0 UDP port: 11111](#)를 설치하여 Tello로 부터 오는 비디오를 받을 수 있음
- 비디오 스트림을 받기 위해서는 ["streamon" 명령어를 UDP Port 8889로 보내주면 됨](#)

## Control Commands

Command	Description	Possible Response
Command	Enter SDK mode.	ok / error
takeoff	Auto takeoff.	
land	Auto landing.	
streamon	Enable video stream.	
streamoff	Disable video stream.	
emergency	Stop motors immediately.	
up x	Ascend to "x" cm. x = 20-500	
down x	down "x" Descend to "x" cm. x = 20-500	
left x	Fly left for "x" cm. "x" = 20-500	
right x	Fly right for "x" cm. "x" = 20-500	
forward x	Fly forward for "x" cm. "x" = 20-500	
back x	Fly backward for "x" cm. "x" = 20-500	
cw x	Rotate "x" degrees clockwise. "x" = 1-360	

- SDK 라이브러리 사용

- 소켓 프로그램이 아닌 SDK에서 제공하는 Tello() 클래스를 사용하면 더욱 쉽게 프로그램 개발이 가능
    - 다운 받은 SDK에서 Tello() 클래스를 제공
    - 소켓프로그램을 클래스 안에 내장한 것으로 좀 더 빠르고 편하게 개발할 수 있음
- 

## ✓ Tello() 클래스

```
1 import socket
2 import threading
3 import time
4 from stats import Stats #stats 클래스
5
6 class Tello:
7     #초기화 initialize 메서드 - 객체가 생성될 때, 자동으로 호출되는 메서드
8     #객체의 초기값을 설정해야 할 때 사용
9     #self는 객체의 인스턴스 그 자체를 말하며 객체 자기 자신을 참조하는 매개변수
10    def __init__(self):
11        self.local_ip = ''
12        self.local_port = 8889
13        self.socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # socket for sending cmd
14        self.socket.bind((self.local_ip, self.local_port))
15
16        # thread for receiving cmd ack
17        self.receive_thread = threading.Thread(target=self._receive_thread)
18        self.receive_thread.daemon = True
19        self.receive_thread.start()
20
21        self.tello_ip = '192.168.10.1'
22        self.tello_port = 8889
23        self.tello_addrs = (self.tello_ip, self.tello_port)
24        self.log = []
25
```



```
26         self.MAX_TIME_OUT = 15.0
27
28     def send_command(self, command):
29         """
30         Send a command to the ip address. Will be blocked until
31         the last command receives an 'OK'.
32         If the command fails (either b/c time out or error),
33         will try to resend the command
34         :param command: (str) the command to send
35         :param ip: (str) the ip of Tello
36         :return: The latest command response
37         """
38         self.log.append(Stats(command, len(self.log)))
39
40         self.socket.sendto(command.encode('utf-8'), self.tello_adderss)
41         print('sending command: %s to %s' % (command, self.tello_ip))
42
43         start = time.time()
44         while not self.log[-1].got_response():
45             now = time.time()
46             diff = now - start
47             if diff > self.MAX_TIME_OUT:
48                 print('Max timeout exceeded... command %s') % command
49                 # TODO: is timeout considered failure or next command still get executed
50                 # now, next one got executed
51                 return
52         print('Done!!! sent command: %s to %s' % (command, self.tello_ip))
53
54     def _receive_thread(self):
55         """Listen to responses from the Tello.
56
57         Runs as a thread, sets self.response to whatever the Tello last returned.
58
59         """
60         while True:
61             try:
62                 self.response, ip = self.socket.recvfrom(1024)
```

```
63         print('from %s: %s' % (ip, self.response))
64
65         self.log[-1].add_response(self.response)
66     except (socket.error, exc):
67         print("Caught exception socket.error : %s" % exc)
68
69     def on_close(self):
70         pass
71         # for ip in self.tello_ip_list:
72         #     self.socket.sendto('land'.encode('utf-8'), (ip, 8889))
73         # self.socket.close()
74
75     def get_log(self):
76         return self.log
```

---

## ▼ Stats() 클래스

```
1 from datetime import datetime
2
3 class Stats:
4     def __init__(self, command, id):
5         self.command = command
6         self.response = None
7         self.id = id
8
9         self.start_time = datetime.now()
10        self.end_time = None
11        self.duration = None
12
13    def add_response(self, response):
14        self.response = response
15        self.end_time = datetime.now()
16        self.duration = self.get_duration()
```

```
17         # self.print_stats()
18
19     def get_duration(self):
20         diff = self.end_time - self.start_time
21         return diff.total_seconds()
22
23     def print_stats(self):
24         print('Wnid: %s' % self.id)
25         print('command: %s' % self.command)
26         print('response: %s' % self.response)
27         print('start time: %s' % self.start_time)
28         print('end_time: %s' % self.end_time)
29         print('duration: %sWn' % self.duration)
30
31     def got_response(self):
32         if self.response is None:
33             return False
34         else:
35             return True
36
37     def return_stats(self):
38         str = ''
39         str += 'Wnid: %sWn' % self.id
40         str += 'command: %sWn' % self.command
41         str += 'response: %sWn' % self.response
42         str += 'start time: %sWn' % self.start_time
43         str += 'end_time: %sWn' % self.end_time
44         str += 'duration: %sWn' % self.duration
45         return str
```


---

▼ *tello\_control\_test\_2.py 파일*

```
1 #tello_control_test_2.py - #자동 이착륙 코드
2
3 # 반드시 드론을 멀리 떨어져서 실행할 것
4 # 두 개의 파일 - 폴더 Single_Tello_Test에 있는 stats와 tello를 같은 폴더에 위치하고 실행할 것
5 # command 파일 작성할 것
6
7 # 드론과 와이파이 연결
8
9 from tello import Tello
10 from datetime import datetime
11 import time #time 모듈은 Python에서 시간 관련 작업을 수행하기 위한 표준 라이브러리
12
13 start_time = str(datetime.now())
14 file_name = "command.txt"
15 f = open(file_name, "r")
16 commands = f.readlines() # readlines() : 파일 내 텍스트에서 각 줄을 element로 하는 리스트로 반환
17 tello = Tello()
18
19 for command in commands:
20     if command != '' and command != '\n':
21         command = command.rstrip() #인자로 전달된 문자를 String의 오른쪽에서 제거
22         if command.find('delay') != -1:
23             sec = float(command.partition('delay')[2])
24             print('delay %s' % sec)
25             time.sleep(sec)
26             pass
27         else:
28             tello.send_command(command)
29
30 log = tello.get_log() #get_log method is used to get the log for a given log type
31 outFile = open('log.txt', 'w+')
32 for stat in log:
33     stat.print_stats()
34     str = stat.return_stats()
35     outFile.write(str)
36
37
```

```
38 #command.txt의 내용
39 #takeoff
40 #delay 5
41 #land
```

---

 command - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
command
takeoff
delay 5
land
```

---

## ✓ [DJI TelloPy API Reference](#)

---

## ✓ [기본동작제어](#)

- takeoff/land - takeoff(), land()
- move up/down - move\_up(), move\_down()
- move left/right - move\_left(), move\_right()
- rotate\_CW\_CCW - rotate\_clockwise(), rotate\_counter\_clockwise()
- send\_rc\_control\_async - send\_rc\_control(self, left\_right\_velocity, forward\_backward\_velocity, up\_down\_velocity, yaw\_velocity)

---

### ✓ 실습(1)

- takeoff -> 3회 up(40) -> down(40) 반복 -> landing
- 

### ✓ 실습(2)

- takeoff -> fwd(40) -> fwd(40) -> fwd(40) -> cw(180) -> fwd(40) -> fwd(40) -> fwd(40) -> land
- 

## 참고사항

### ✓ DJITelloPy 패키지 설치

- DJI Tello drone python interface using the official Tello SDK and Tello EDU SDK. This library has the following features:

- implementation of all tello commands
- easily retrieve a video stream
- receive and parse state packets
- control a swarm of drones
- [support for python >= 3.6](#)

- [pip install djitellopy](#)

- [pip install opencv-python](#)

- [from djitellopy import tello](#)

```
1 #기본적인 동작 제어
2 from djitellopy import Tello
3 tello = Tello()
4 tello.connect()
5 tello.takeoff()
6 tello.move_left(100)
7 tello.rotate_clockwise(90)
8 tello.move_forward(100)
9 tello.land()
```

```
1 #반복문 이용하기 : for in range()
2 from djitellopy import Tello
3 myTello = Tello()
```