

파이썬 실행시 sudo 유무의 차이

```
zero@pc: ~/catkin_ws/src/mini_model/point
zero@pc: ~/catkin_ws/src/mini_model/point 80x24
zero@pc:~/catkin_ws/src/mini_model/point$ sudo python3 rccarServer.py
Traceback (most recent call last):
  File "rccarServer.py", line 6, in <module>
    from point1_test import *
  File "/home/zero/catkin_ws/src/mini_model/point/point1_test.py", line 3, in <module>
    import rospy
ModuleNotFoundError: No module named 'rospy'
zero@pc:~/catkin_ws/src/mini_model/point$ python3 rccarServer.py
* Running on http://192.168.123.79:8080 (Press CTRL+C to quit)
192.168.123.79 - - [08/Aug/2022 12:01:34] "GET / HTTP/1.1" 200 -
192.168.123.79 - - [08/Aug/2022 12:01:36] "GET /working HTTP/1.1" 200 -
192.168.123.79 - - [08/Aug/2022 12:01:36] "GET /stream HTTP/1.1" 404 -
```

- sudo로 python3 파일을 실행시키면 import rospy 에러가 발생함
- sudo 없이 python3 파일을 실행시키면 서버 구동이 됨(솔루션)

혹시 몰라서 파이썬 버전마다 설치된 내역 확인

```
zero@pc: ~/catkin_ws/src/mini_model/point
zero@pc: ~/catkin_ws/src/mini_model/point 80x24
zero@pc:~/catkin_ws/src/mini_model/point$ pip3 freeze | grep "rosp*"
gazebo_ros==2.8.7
rosbag==1.14.13
rosclean==1.14.9
roscpp==1.14.9
roslaunch==1.14.13
roslib==1.14.9
roslint==0.11.2
roslz4==1.14.13
rosmake==1.14.9
rosmaster==1.14.13
rosmmsg==1.14.13
rostopic==1.14.13
roswtf==1.14.13
```

```
rospkg==1.4.0  
rospy==1.14.13  
rosservice==1.14.13  
rostest==1.14.13  
rostopic==1.14.13  
roswtf==1.14.13  
smach ros==2.0.1
```

```
zero@pc: ~/catkin_ws/src/mini_model/point  
zero@pc: ~/catkin_ws/src/mini_model/point$ pip freeze | grep "rosp*"  
gazebo_ros==2.8.7  
rosbag==1.14.13  
rosclean==1.14.9  
roscrc==1.14.9  
roslaunch==1.14.13  
roslib==1.14.9  
roslint==0.11.2  
roslz4==1.14.13  
rosmake==1.14.9  
rosmaster==1.14.13  
rosmmsg==1.14.13  
rostopic==1.14.13  
rospkg==1.4.0  
rospy==1.14.13  
rosservice==1.14.13  
rostest==1.14.13  
rostopic==1.14.13  
roswtf==1.14.13  
smach_ros==2.0.1
```

- python3, python 버전 둘다 rospy 설치되어 있음.

```
zero@pc: ~/catkin_ws/src/mini_model/point
zero@pc: ~/catkin_ws/src/mini_model/point 80x24
zero@pc:~/catkin_ws/src/mini_model/point$ sudo pip3 freeze | grep "rosp*"
WARNING: The directory '/home/zero/.cache/pip' or its parent directory is not owned or is not writable by the current user. The cache has been disabled. Check the permissions and owner of that directory. If executing pip with sudo, you should use sudo's -H flag.
rospkg==1.4.0
zero@pc:~/catkin_ws/src/mini_model/point$ sudo pip freeze | grep "rosp*"
WARNING: The directory '/home/zero/.cache/pip' or its parent directory is not owned or is not writable by the current user. The cache has been disabled. Check the permissions and owner of that directory. If executing pip with sudo, you should use sudo's -H flag.
rospkg==1.4.0
zero@pc:~/catkin_ws/src/mini_model/point$
```

- sudo로 pip3, pip에 설치된 목록에서는 rospy가 설치 되어 있지 않다.
- pip는 python패키지를 설치하고 관리하는 패키지매니저이다.
- **sudo로 python 파일을 실행시 rospy 모듈이 없다고 뜨는 이유가 바로 이것 때문이다.**

파이썬 패키지 설치시 sudo 유무의 차이

파이썬 패키지 설치시 pip나 pip3로 설치하게 되는데,
pip는 Pip Install Packages의 약자로 Python 패키지를 설치하고 관리하는 패키지 매니저이다.

pip와 pip3의 차이

pip	Python2 버전 패키지 매니저
pip3	Python3 버전 패키지 매니저

pip로 패키지를 설치할 때 sudo는 위험하니 절대 쓰면 안된다고 한다.
대신 가상환경 패키지 매니저를 설치하고 가상환경을 활용하자고 함
pip 개발자도 특권이 필요한 경로를 디폴트 경로로 설정한 것에 대해 후회한다고 함

처음 사용자는 pip를 통해서 패키지 설치를 하게 되면 주로 다음과 같은 흐름을 거쳐서 패키지 설치를 진행하게 된다.

ex)

1. 패키지 설치
 - pip3 install opencv
2. 설치 중 에러 발생
 - PermissionError: [Errno 13] Permission denied: '설치_경로'
3. 구글링 결과 sudo 해결책이 대다수
 - sudo pip3 install opencv 명령으로 설치

하지만 sudo를 가급적이면 사용하지 말라는 글이 있다.

왜냐하면 위에 설치 시 뜬 에러를 디버깅해보면

pip3 install의 default 설치 경로는 다음과 같다.

/usr/local/bin/python3.7

문제는 해킹 경로에 파일을 쓰거나(write) 변경할(edit) 권리(privilege)가 현재 사용자에게 없는 점이다.

즉, pip3가 현재 사용자에게 허락되지 않는 경로에 패키지를 쓰려고 했기 때문에(설치) PermissionError가 발생한 것이다.

이때, sudo = Super User Do로 현재 사용자에게 슈퍼 유저 권한을 부여하게 되면 PermissionError를 회피할 수 있게 된다.

왜 sudo를 쓰면 안될까?

1. 원리적으로 하면 안되는 것을 할 이유가 없다.
 - 현 프로그래머들은 파일 권한 시스템을 만들어서, 시스템의 계층 구조를 합리적으로 만들어두었다. 계층상 상위에 속하는 파일들이 지워지거나 수정되면 문제가 되기 때문에 시스템 안정성을 위해 몇몇 경로에는 일반 권한으로는 쓸수도 지울수도 없게 만들었기 때문이다.
2. sudo pip3 install을 함부로 쓰면, 해킹에 노출된다.
 - 어떤 해커가 한 파이썬 파일에 악성 코드를 숨겨 놓고 sudo 권한을 가진 파일을 악성코드가 숨겨진 파일을 실행하게 하면 root 시스템이 무너져 해커가 root 권한을 얻게 되기 때문이다.

Solution

1. local user home 경로에 패키지 설치하기
 - 현 사용자에게 허락된 경로에 설치한다.
 - pip3 install opencv --user
 - 현 사용자 내부 경로에 설치하는 것으로 설치경로는 다음과 같다.
 - '/home/zero/.local/lib/python3.7/site-packages'
 - 하지만 이 방법은 버저닝에 대한 고려가 없다.

예를 들어 [a.py](#)에는 opencv version을 1로 해야하고, [b.py](#)에는 opencv version을 2로 해야할 때 동시에 서로 다른 버전을 활용해야하는 경우 불가능해진다.
2. 위에 버저닝에 대한 대안으로 Virtualenv과 같은 가상 환경 패키지를 설치해서 이용하는 방법이다.
 - 가상환경 1(opencv version 1설치)을 활성화하고 [a.py](#)를 실행하고 다 끝났으면 비활성화한 후 가상환경 2(opencv version 2설치)을 활성화 하고 [b.py](#)를 실행한다.

Versioning = 인쇄물을 여러 버전으로 작성하는 것을 뜻함.

즉 python 파일마다 요구하는 프로그램의 버전이 다를때 그게 맞게 버전도 달라야 됨