

Software Installation and Tutorial for Universal Robots (UR)

[OS: Windows10 64bit]

Aug 1, 2021

by Soojin Kim and Youngshik Kim



Intelligent Control & Robot System (ICRS) lab.

Department of Mechanical Engineering

Hanbat National University

<http://robot.hanbat.ac.kr/>

Listed software

Windows10 64bit

Ubuntu 16.04 desktop

Python 3.7.7

Python libraries: math3d, urx, numpy, sympy

VirtualBox 6.1

PyCharm community

URsim 3.12.1.90940

MATLAB

URToolbox(1.1.4)(15,Mar,2018) - Dr.Kutzer

All rights reserved

List of Contents

1. 프로그램 설치

가. 설명

1) 필수, 선택 프로그램 설명 -----	1
-------------------------	---

나. 필수

1) Python 설치 -----	2
--------------------	---

다. 선택 – Pycharm, Ubuntu

1) PyCharm 설치 -----	5
2) PyCharm 설정 -----	6
3) VirtualBox 설치 -----	10
4) 가상 Ubuntu OS 설치 -----	11
5) 가상 Ubuntu URsim 설치,세팅 -----	20

라. 선택 – MATLAB

1) MATLAB URToolbox 설치 -----	31
------------------------------	----

List of Contents

2. 예제

마. PyCharm

1) 예제 파일 확인 및 실행 -----	34
2) 명령어 설명 -----	37
3) Robotiq 2-finger gripper 명령어 설명 -----	39
4) 예제에 사용된 변수명 설명 -----	40
5) 예제 결과 (잘 된 경우) -----	41
6) 예제 결과 (오류 현상) -----	43

바. MATLAB

1) 예제 파일 확인 및 실행 -----	47
2) 명령어 설명 -----	49
3) Robotiq 2-finger gripper 명령어 설명 -----	50
4) 예제에 사용된 변수명 설명 -----	51
5) 예제에 표시된 작업공간(변수값) 설명 -----	52
6) 예제 결과 (잘 된 경우) -----	53
7) 예제 결과 (오류 현상) -----	54

프로그램 설치 / 설명 / 필수, 선택 프로그램 설명

UR 로봇팔을 제어함에 있어 필수적으로 필요한 파일은 Python이다.

필요에 따라 선택적으로 설치할 수 있도록 본 설명집에서는 PyCharm과 MATLAB URToolbox의 설치 방법을 제공한다.

PyCharm과 MATLAB URToolbox 모두 실제 로봇과의 통신을 통해 구동할 수 있지만, 시뮬레이션의 경우 PyCharm(Python)은 자체적인 기능이 없다. 그래서 PyCharm(Python)은 별도로 VirtualBox(컴퓨터 가상화 프로그램)에서 Ubuntu OS, URsim을 통해 시뮬레이션을 확인할 수 있다.

[본 저작물의 무단 수정, 게재를 금합니다]

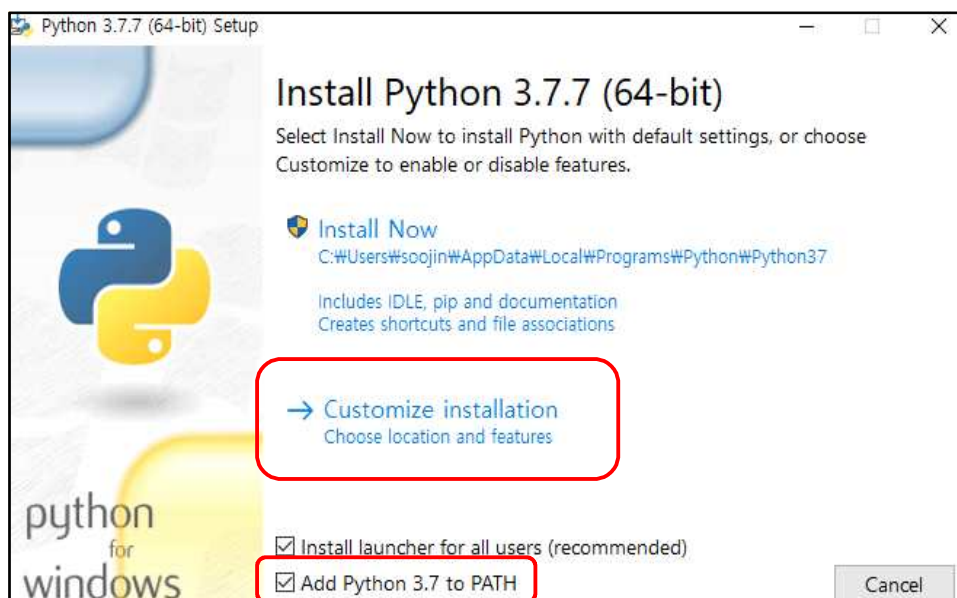
프로그램 설치 / 필수 / Python 설치

1. <https://www.python.org/downloads/release/python-377/> 에 들어가서 페이지 하단에 'Windows x86-64 executable installer'를 클릭, 다운로드 한다. (혹은 구글 'python3.7.7 download' 검색)

Files

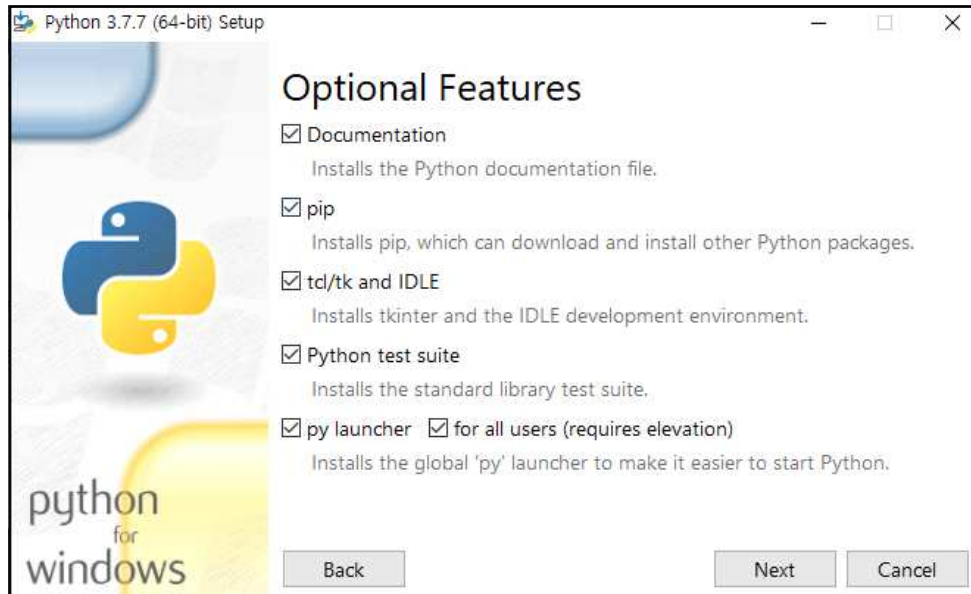
Version	Operating System
Gzipped source tarball	Source release
XZ compressed source tarball	Source release
macOS 64-bit installer	Mac OS X
Windows help file	Windows
Windows x86-64 embeddable zip file	Windows
Windows x86-64 executable installer	Windows
Windows x86-64 web-based installer	Windows
Windows x86 embeddable zip file	Windows
Windows x86 executable installer	Windows
Windows x86 web-based installer	Windows

2. 다운로드한 파일을 클릭하여 창 하단에 'Add Python 3.7 to PATH'를 **반드시 선택 후** 'Customize installation'을 클릭한다.

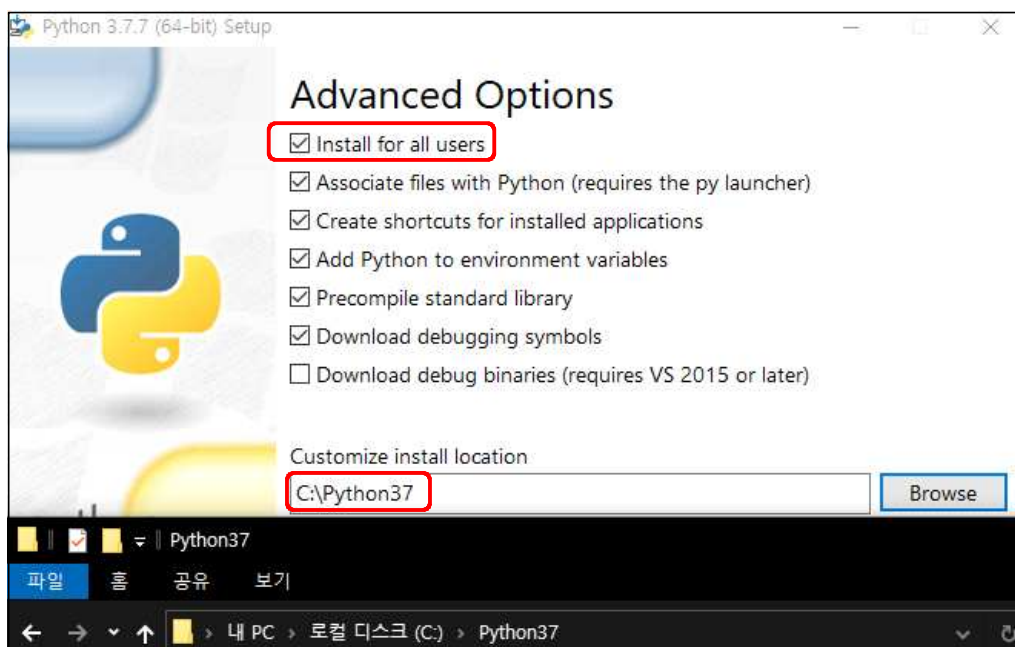


프로그램 설치 / 필수 / Python 설치

3. 모든 체크박스에 체크가 되어있는 것을 확인 한 후, 'Next'를 누른다.



4. 컴퓨터 폴더 'C:\w'에 들어가서 새 폴더 'Python37'을 만들고 경로를 복사하여 설치화면 'Customize install location'에 붙여 넣는다. 체크박스 항목 'Install for all users'를 체크한 후 'Install' 버튼을 눌러 설치한다.



프로그램 설치 / 필수 / Python 설치

5. 키보드 '시작'키를 누른 후, '명령 프롬프트' 입력 후 엔터키를 누른다.
생성된 명령 프롬프트 창에서
'pip install numpy --user' 입력 후 엔터키를 누르고 기다린다.



```
명령 프롬프트
Microsoft Windows [Version 10.0.18363.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\soojin>pip install numpy --user
Collecting numpy
```

6. 5번과 같은 방법으로 아래 두 명령어도 반복한다.
'pip install math3d --user'
'pip install urx --user'
'pip install sympy --user'

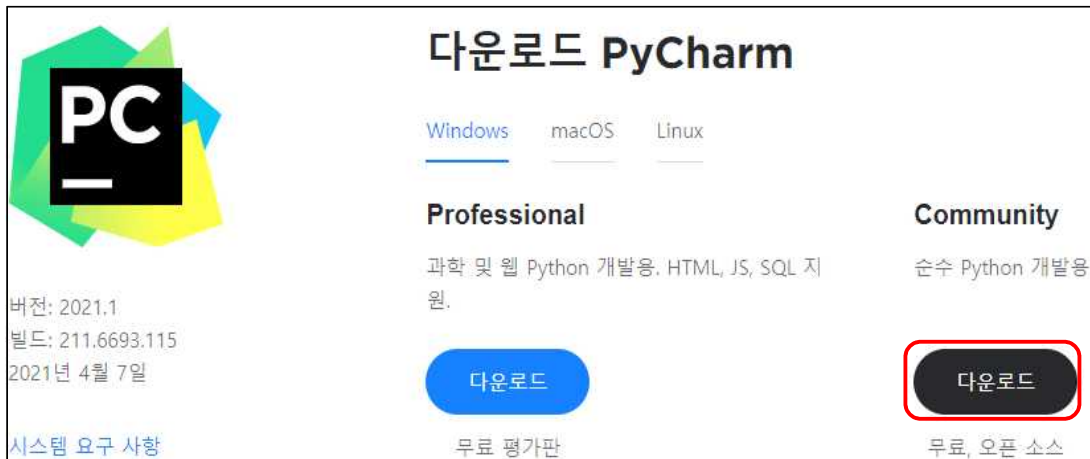
※ 설치 확인 방법

명령 프롬프트 창에서 'pip list'를 입력했을 때, 아래 그림처럼 표시가 된다면 설치가 잘 된 것이다.
(버전은 조금씩 다를 수 있음)

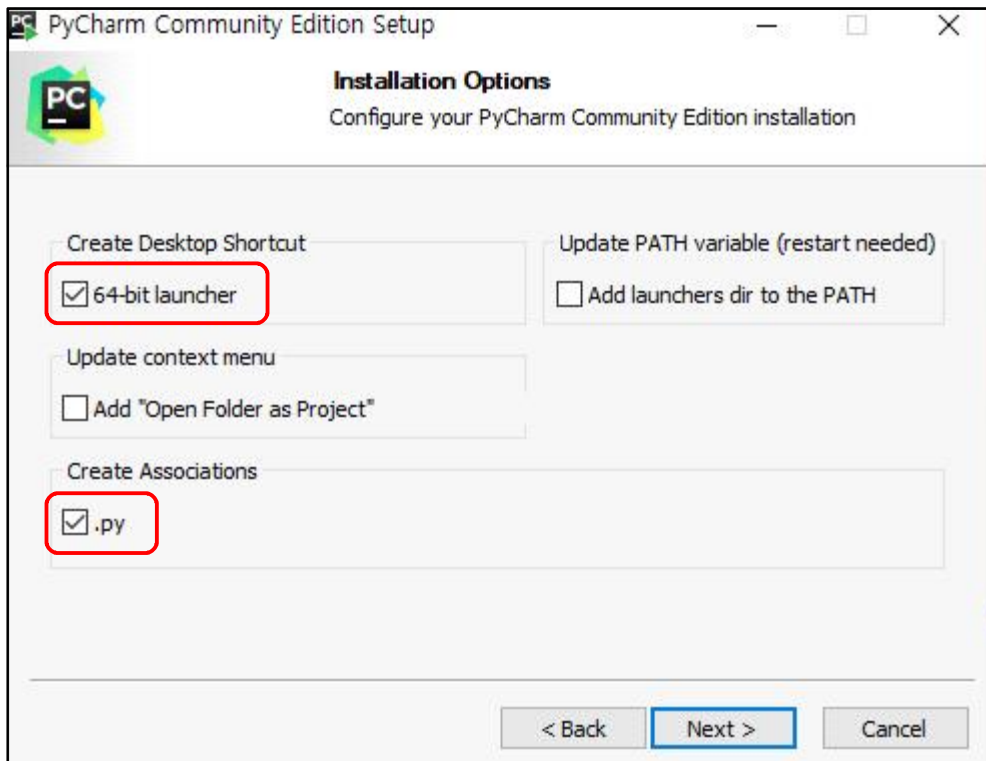
Package	Version
math3d	3.4.1
mpmath	1.2.1
numpy	1.20.2
pip	19.2.3
setuptools	41.2.0
sympy	1.7.1
urx	0.11.0

프로그램 설치 / 선택 / PyCharm 설치

1. <https://www.jetbrains.com/ko-kr/pycharm/download/#section=windows>
(혹은 구글 'pycharm' 검색)
위 주소에서 Community 아래에 있는 다운로드를 클릭한다.

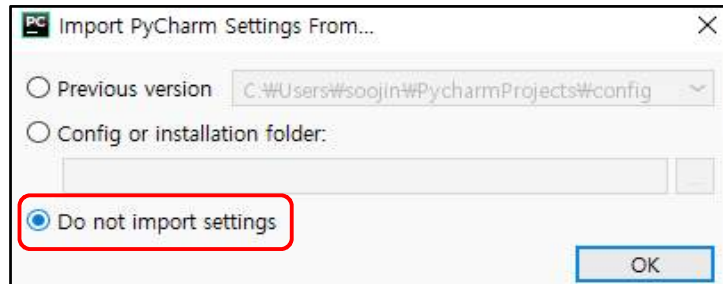


2. 다운로드 한 파일을 실행, 'Next' 버튼을 눌러 설치를 진행하다가 아래와 같은 화면이 나오게 되면 '64-bit launcher', '.py' 체크박스에 **체크하고** 설치를 진행한다.

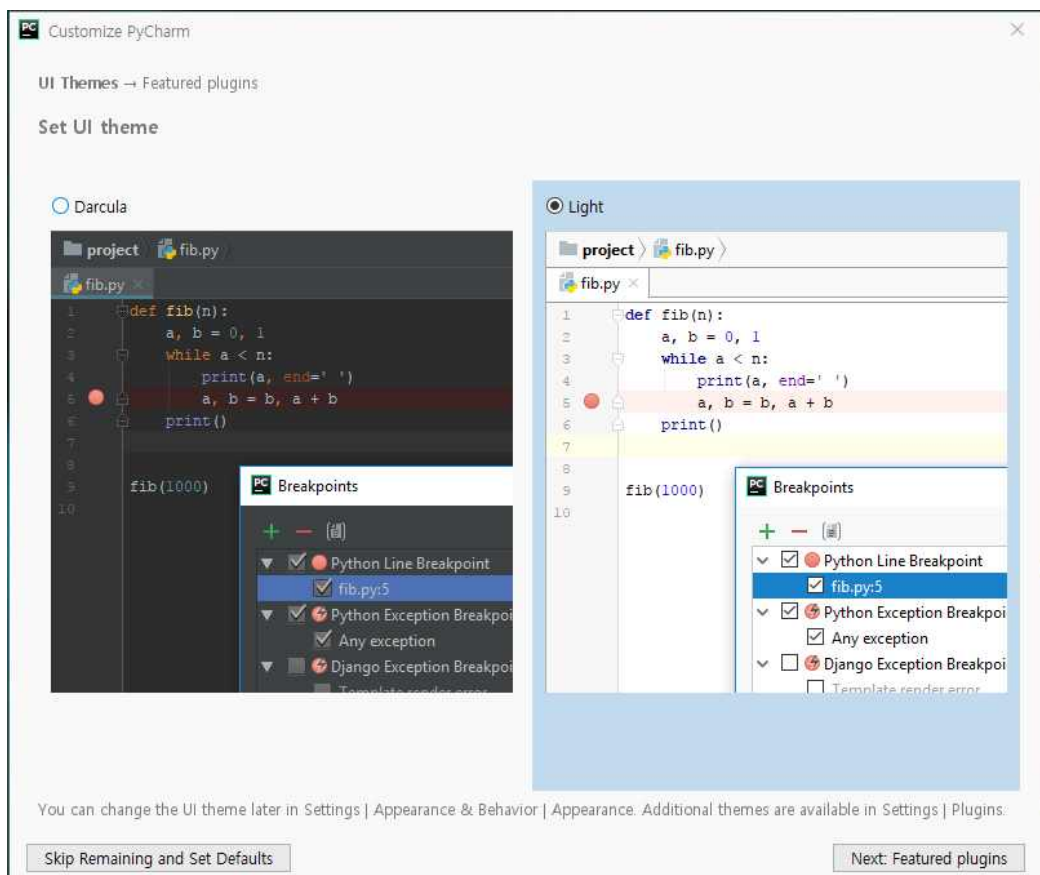


프로그램 설치 / 선택 / PyCharm 설정

※ 설치 후 PyCharm을 실행했을 때 아래와 같은 창이 뜬다면
'Do not import settings'를 선택 후 OK 버튼을 누른다.



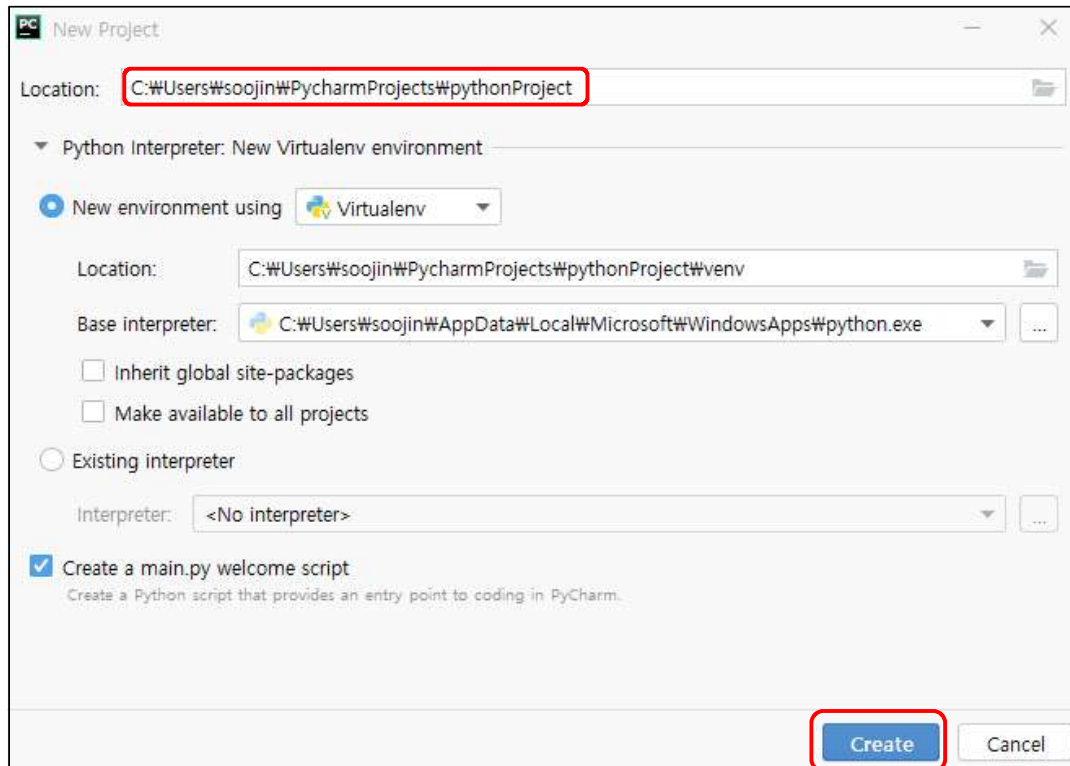
※ 아래 화면의 창에서 원하는 테마를 선택 후 'X'를 눌러 창을 나온다.



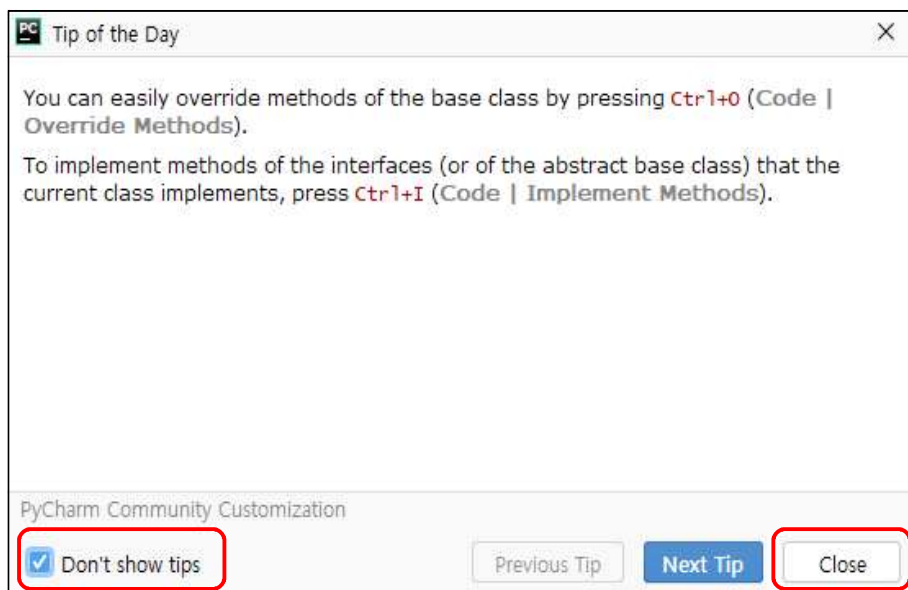
프로그램 설치 / 선택 / PyCharm 설정

1. PyCharm 실행 후 'New Project'를 선택한다.

Location에 빨간색 네모가 그려진 부분은 원하는 프로젝트 생성 경로를 입력해준다. 이후 Create 버튼을 누른다.

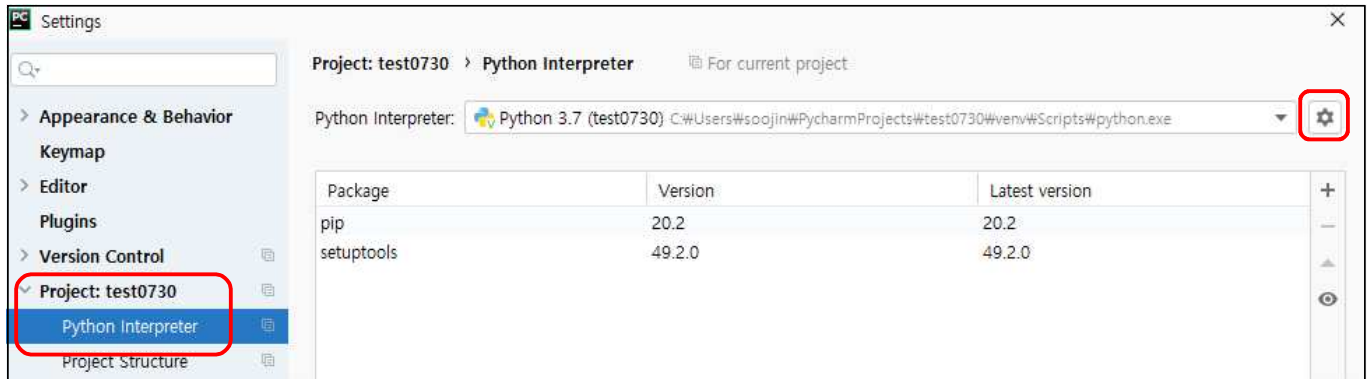




※ New Project 창에서 Create 버튼을 누른 후 뜨는 팝업은 Don't show tips를 체크하고 Close 버튼을 누른다.

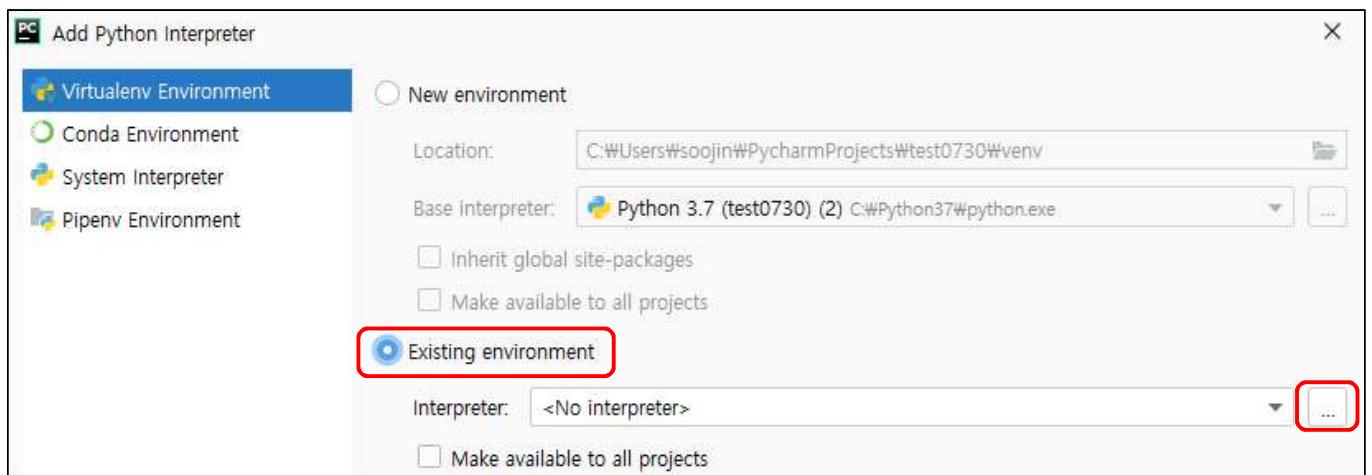


프로그램 설치 / 선택 / PyCharm 설정

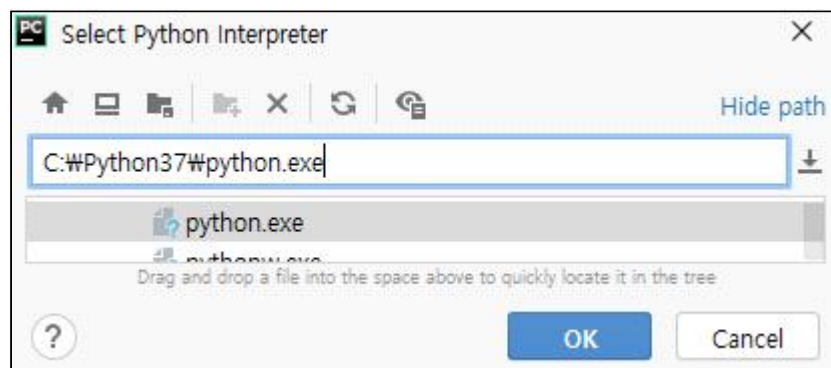
2. 키보드 'Ctrl' + 'Alt' + 'S' 눌러서 Settings 창을 열고
'Project: 프로젝트명' - 'Project Interpreter'를 누른다.



3. 창 우측 상단에  - 'Add' - 창 중반에 'Existing environment' 선택
'Interpreter:' 줄 맨 우측에  클릭



4. 생성되는 창에서 'C:\python37\python.exe' 선택 후
OK버튼을 누른다.

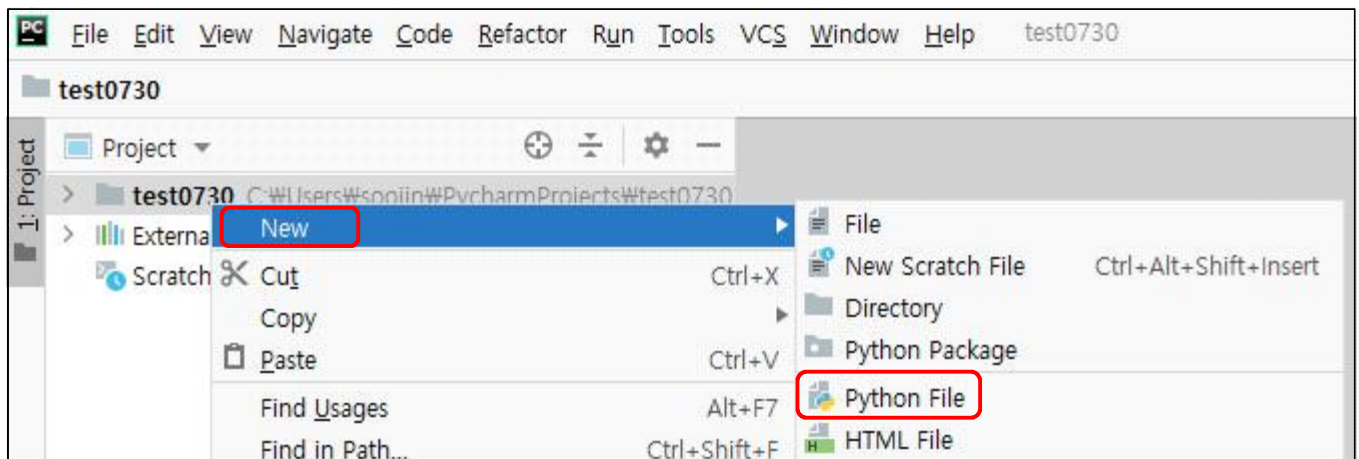


프로그램 설치 / 선택 / PyCharm 설정

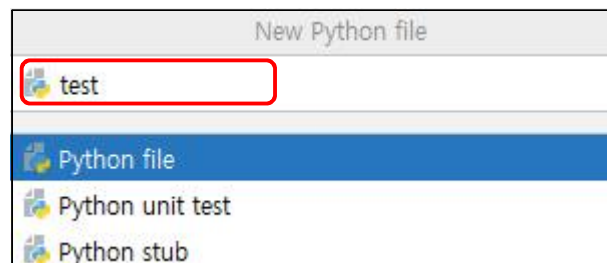
5. Settings 창에 아래 화면 빨간색 네모와 같이 있는지 확인한다.
(버전은 다를 수 있음)
동일하다면 OK버튼을 눌러 창을 빠져나온다.



6. 메인화면 좌측 상단 프로젝트 폴더 아이콘에 커서를 놓고
마우스 우클릭 -> New -> Python File을 누른다.



7. 생성할 파일명을 작성하고 엔터키를 누른다.



- ※ 외부 코드를 복사하여 해당 파일에 붙여넣어서 실행하는 것을 추천.
외부에서 PyCharm으로 파일 실행할 시, 설정 적용이 불가할 수 있음.

프로그램 설치 / 선택 / VirtualBox 설치

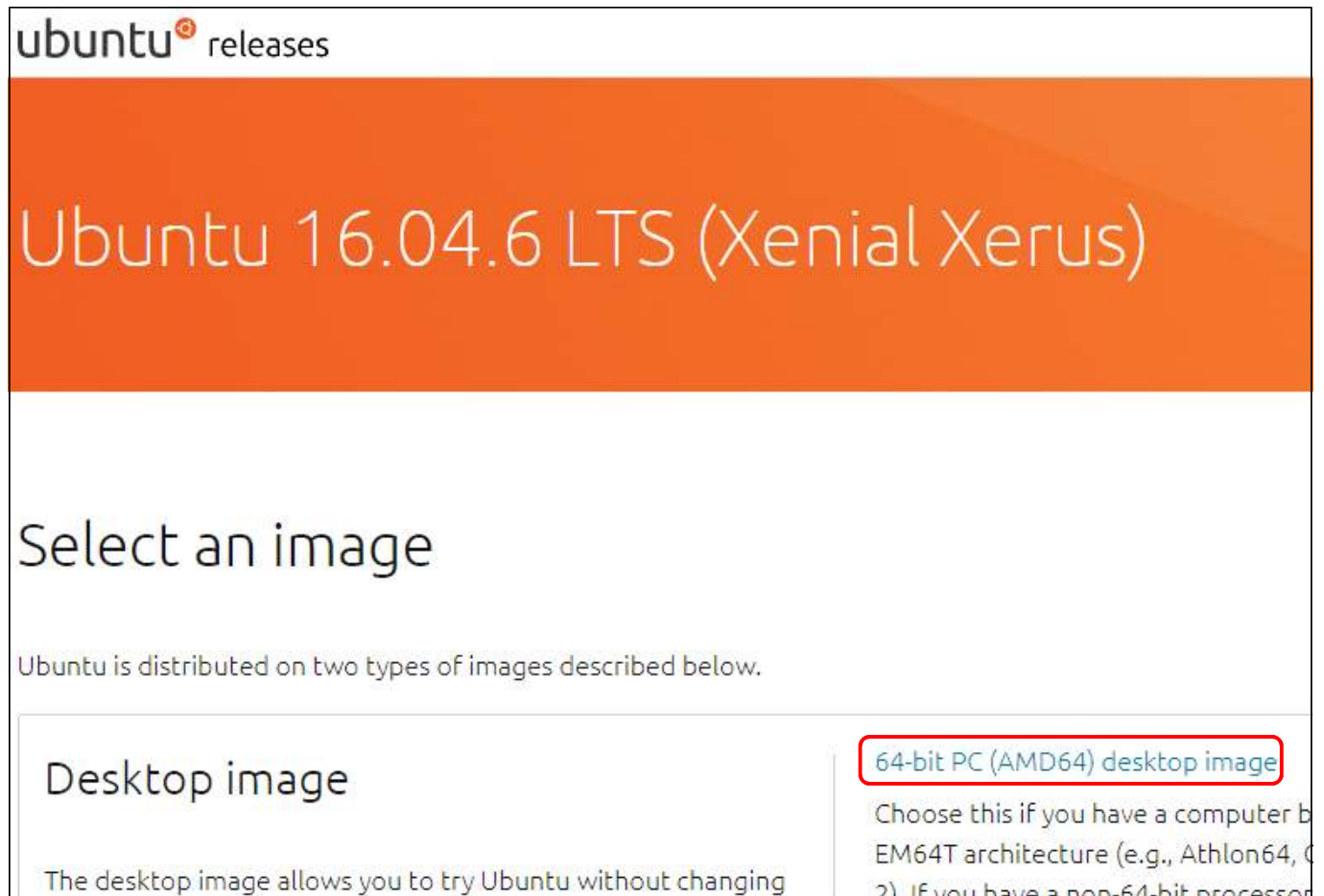
1. <https://www.virtualbox.org/wiki/Downloads> 에서
VirtualBox를 다운받는다. (혹은 구글에 virtualbox 검색)




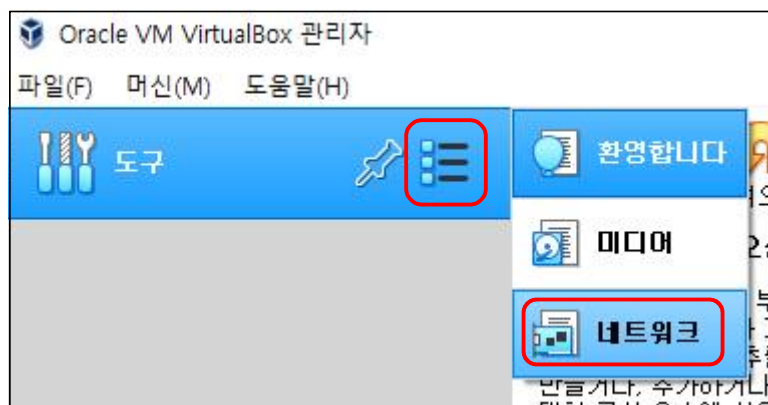
- ※ 다운받은 파일을 실행하여 설치를 완료하면 된다.
(설치를 진행할 때에는 특별히 건드릴 것 없이 Next를 눌러 진행)

프로그램 설치 / 선택 / 가상 Ubuntu OS 설치

1. <http://releases.ubuntu.com/16.04/> 에서 Desktop image를 다운받는다.
(혹은 구글에서 ubuntu 16.04 검색)

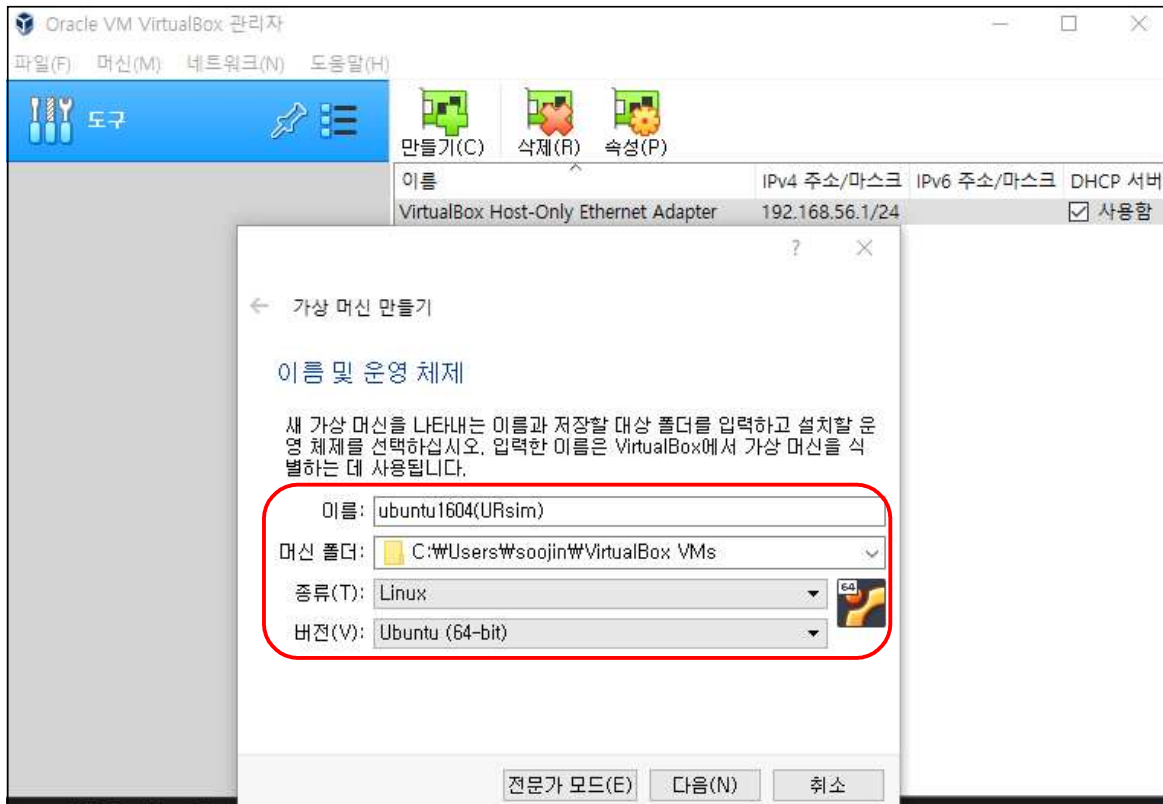


2. VirtualBox를 키고 화면 왼쪽 상단에  -> 네트워크 버튼을 누른다.



프로그램 설치 / 선택 / 가상 Ubuntu OS 설치

3. DHCP 서버에 체크 표시를 하고 키보드 'Ctrl' + N 키를 눌러 '가상 머신 만들기' 창을 띄운다. 이름에는 'ubuntu1604(URsim)', 머신폴더(설치경로)는 자신이 원하는 곳에 위치시킨다.



4. 이후 계속해서 다음, 만들기 버튼을 눌러 아래 그림과 같게 한다.
(진행하는 동안 특별히 건드릴 것이 없음)

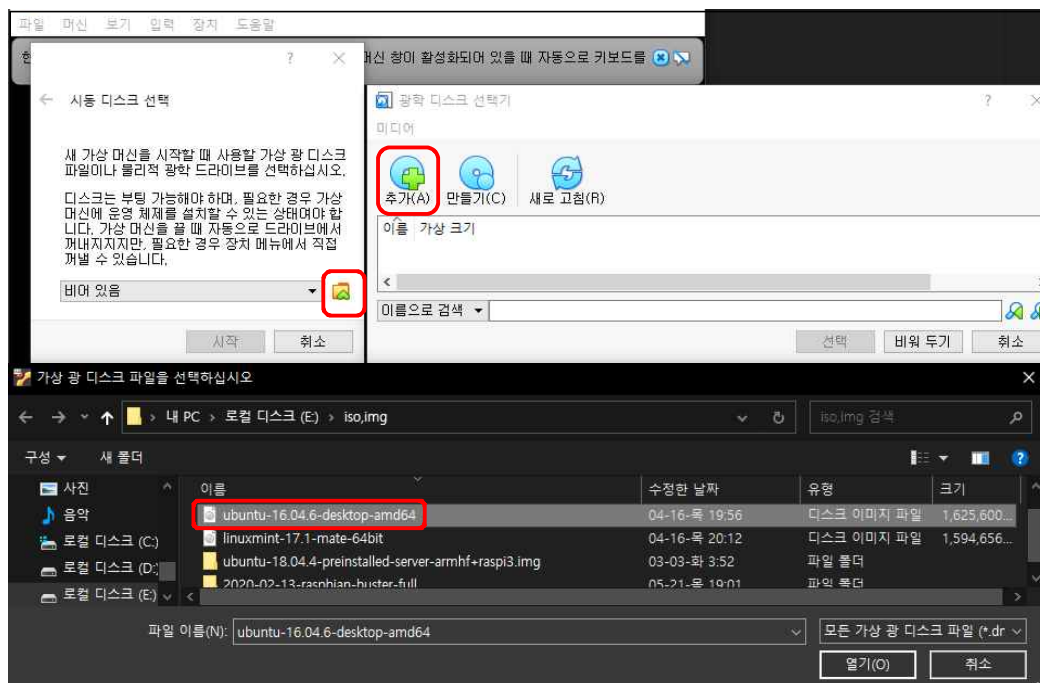


프로그램 설치 / 선택 / 가상 Ubuntu OS 설치

5. 위 그림에서 초록색 시작 버튼을 누른다.

새롭게 팝업되는 '시동 디스크 선택'이라는 창에서 폴더 모양의 버튼을 누르고, '광학 디스크 선택기' 창에서는 추가 버튼을 누른다.

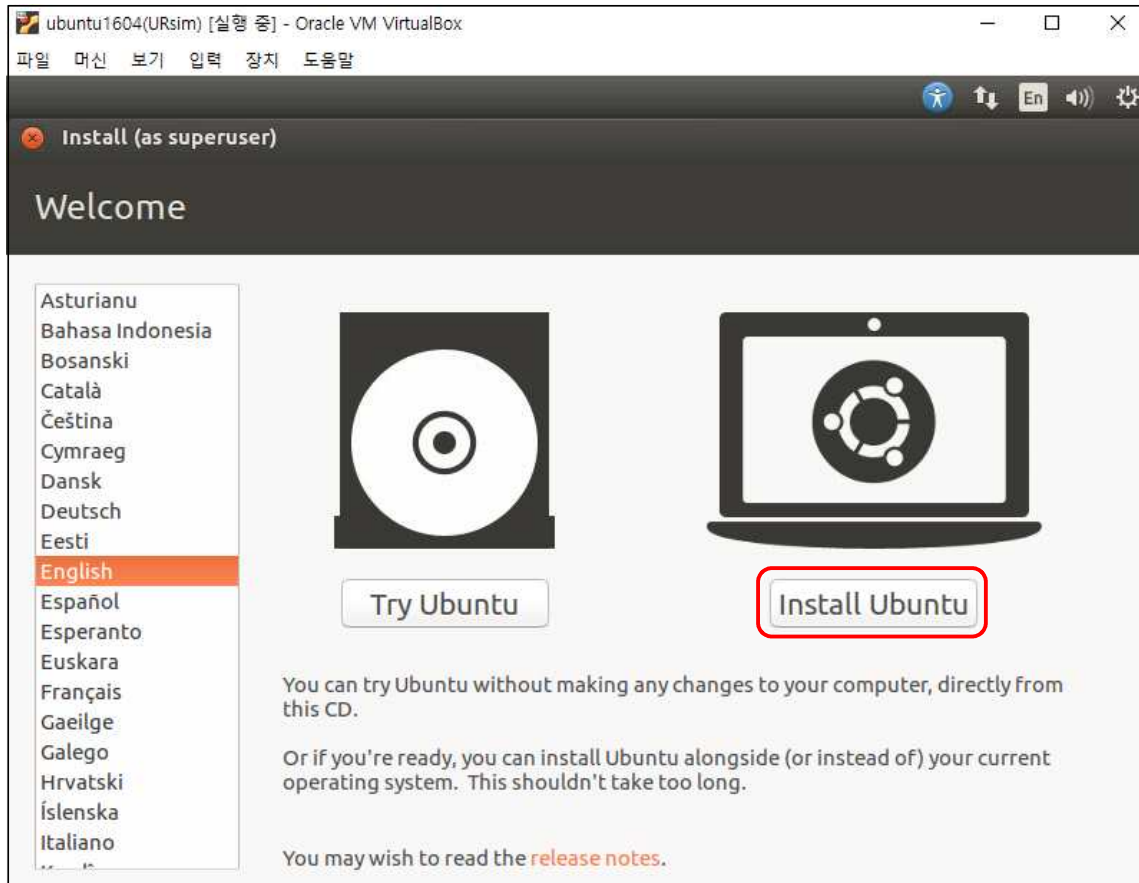
1.번에서 받은 파일(확장자 .iso)을 선택 후,
차례대로 열기, 선택, 시작 버튼을 눌러 창을 나온다.



프로그램 설치 / 선택 / 가상 Ubuntu OS 설치

6. 영어 버전으로 Install Ubuntu 버튼을 누른다.

이후 아래 그림을 따라 버튼을 누르며 설치를 진행한다.



프로그램 설치 / 선택 / 가상 Ubuntu OS 설치

Installation type

This computer currently has no detected operating systems. What would you like to do?

- ☒ **Erase disk and install Ubuntu**
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
- ☐ **Encrypt the new Ubuntu installation for security**
You will choose a security key in the next step.
- ☐ **Use LVM with the new Ubuntu installation**
This will set up Logical Volume Management. It allows taking snapshots and easier partition resizing.
- ☐ **Something else**
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

Write the changes to disks?

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

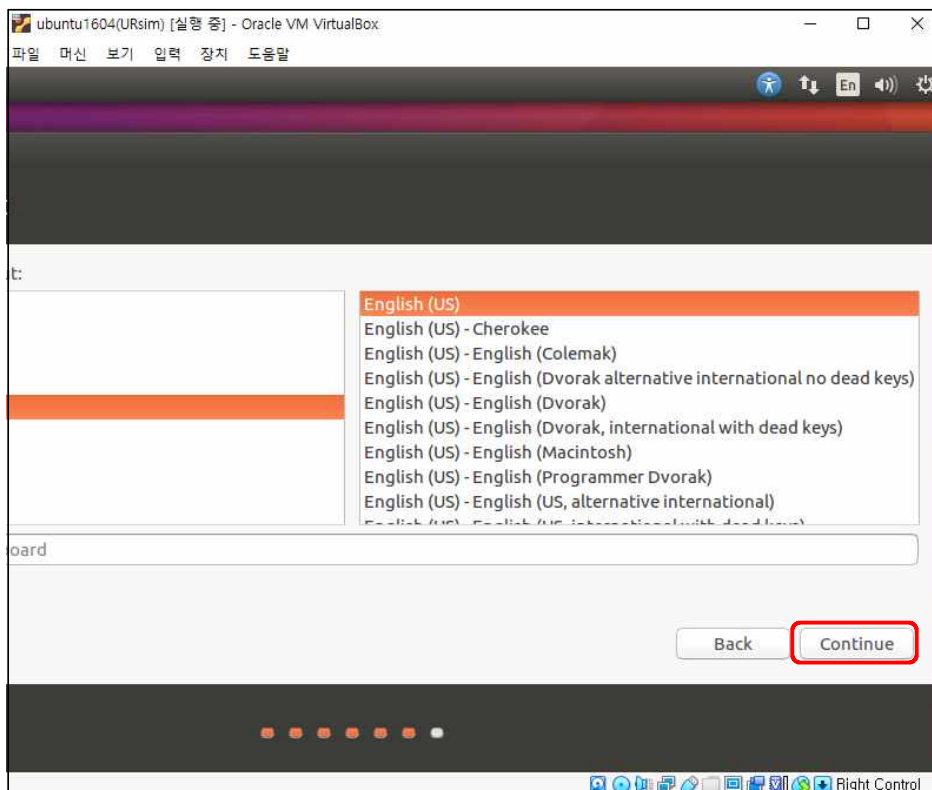
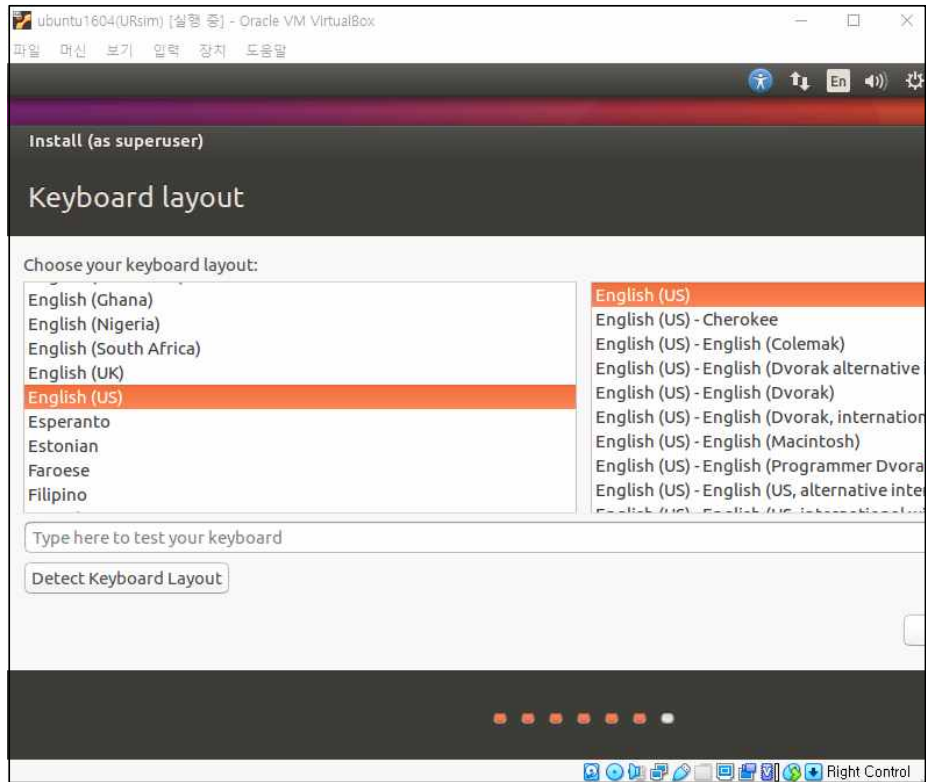
The partition tables of the following devices are changed:
SCSI3 (0,0,0) (sda)

The following partitions are going to be formatted:
partition #1 of SCSI3 (0,0,0) (sda) as ext4
partition #5 of SCSI3 (0,0,0) (sda) as swap

Seoul

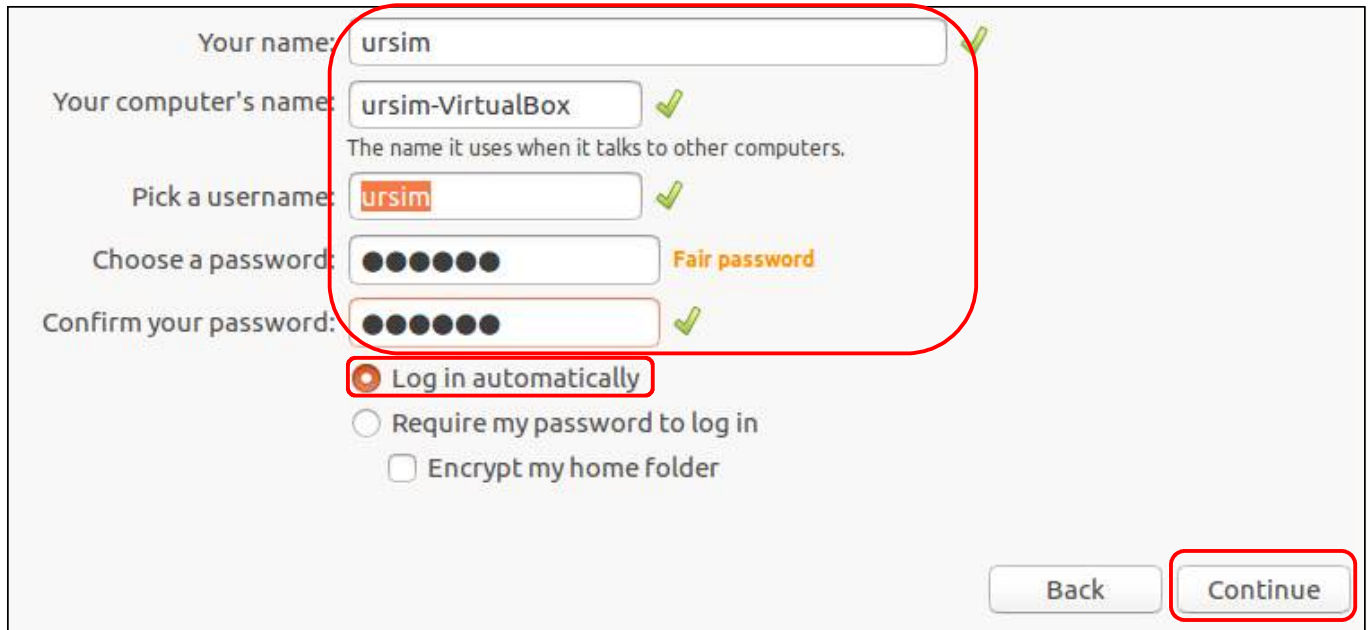
프로그램 설치 / 선택 / 가상 Ubuntu OS 설치

※ 해당 부분에서는 Install (as superuser) 부분을 클릭한채로 옆으로 창을 넘기면 보이는 Continue 부분을 클릭한다. (언어 변경 금지)



프로그램 설치 / 선택 / 가상 Ubuntu OS 설치

※ 비밀번호는 간단하게 1q2w3e로 설정하고 그림과 같이 작성, 선택한다.



The image shows the Ubuntu installer's user configuration screen. A red rounded rectangle highlights the input fields for 'Your name', 'Your computer's name', 'Pick a username', 'Choose a password', and 'Confirm your password'. The values entered are 'ursim', 'ursim-VirtualBox', 'ursim', and a masked password. The password is labeled 'Fair password'. Below the password fields, the 'Log in automatically' option is selected with a radio button. At the bottom right, the 'Continue' button is highlighted with a red rectangle.

Your name: ✓

Your computer's name: ✓
The name it uses when it talks to other computers.

Pick a username: ✓

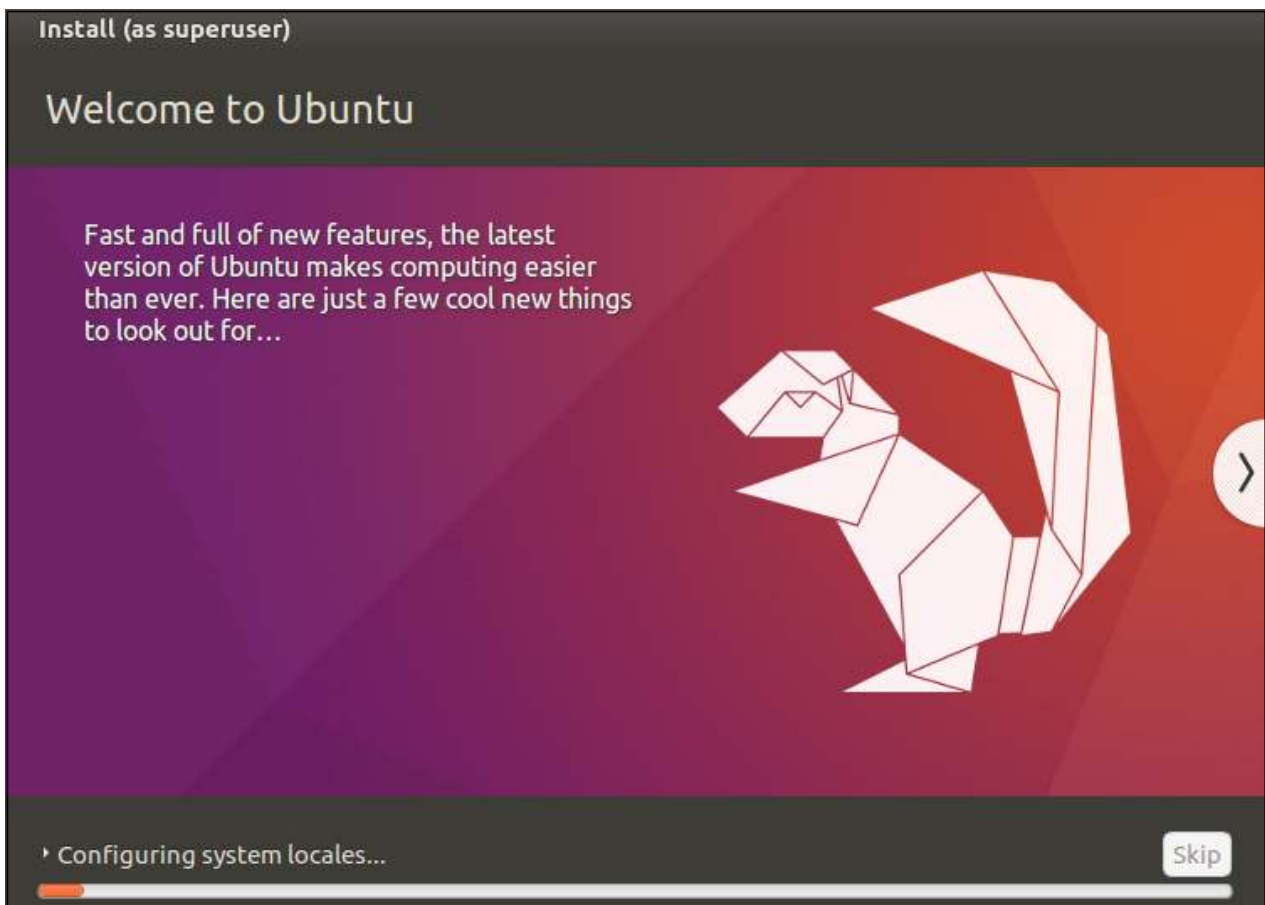
Choose a password: Fair password

Confirm your password: ✓

☒ Log in automatically
☐ Require my password to log in
☐ Encrypt my home folder

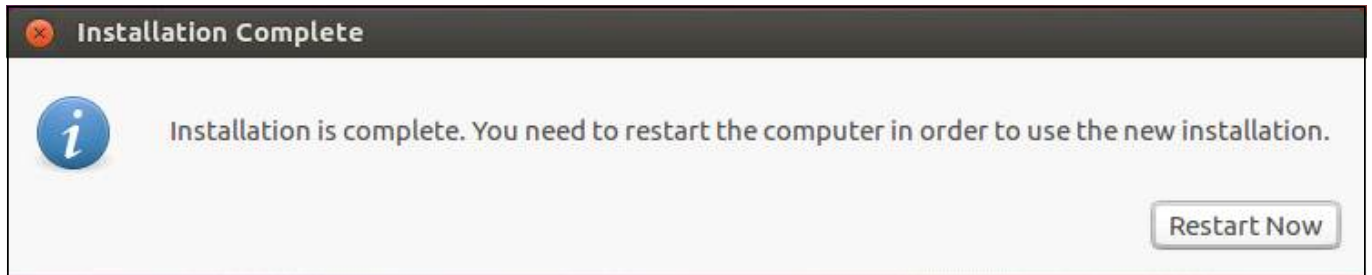
Back Continue

아래 그림부터는 약간의 시간이 소요된다.

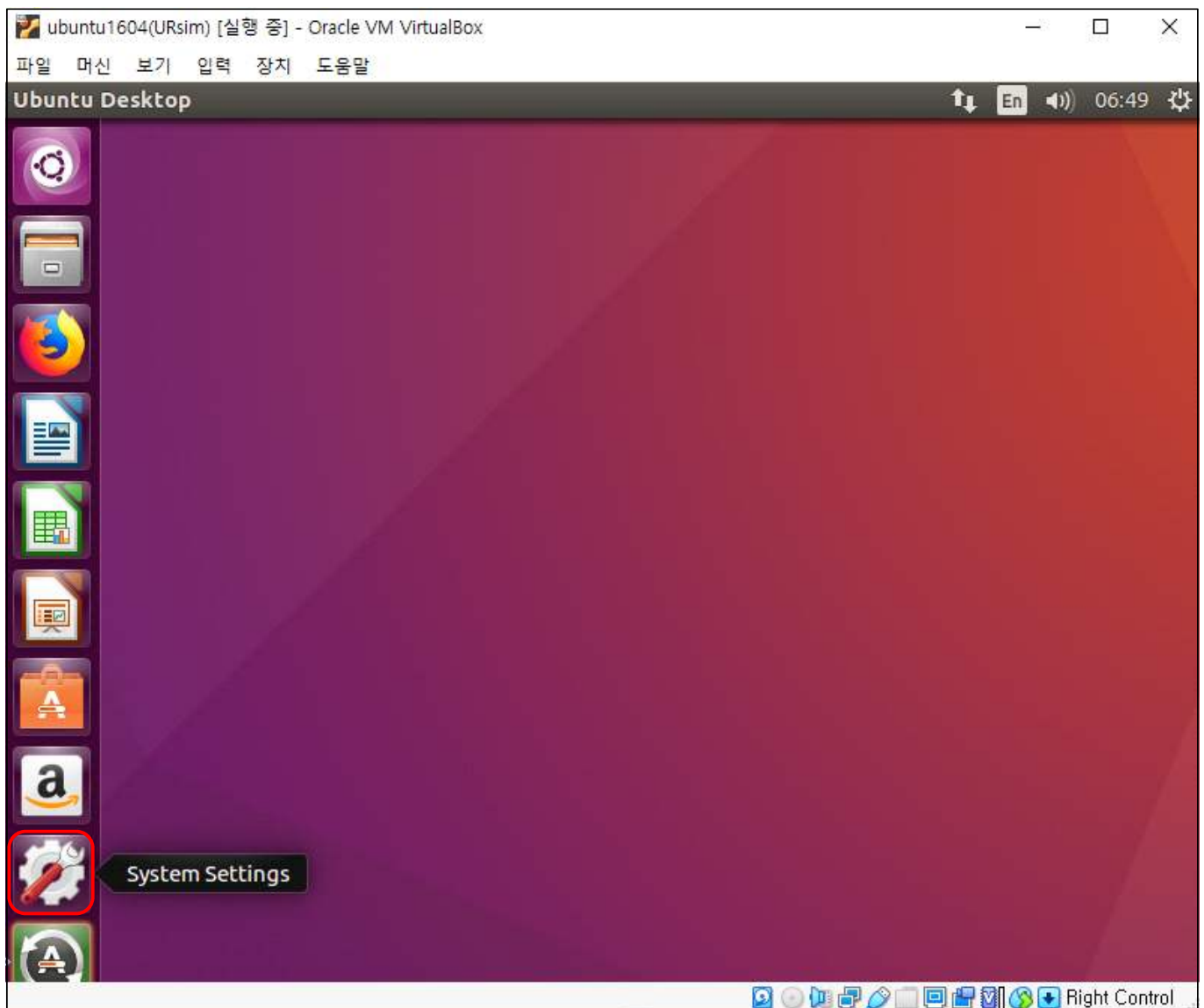


프로그램 설치 / 선택 / 가상 Ubuntu OS 설치

아래와 같은 창이 나타나면 엔터키를 두 번 누르고 대기한다.

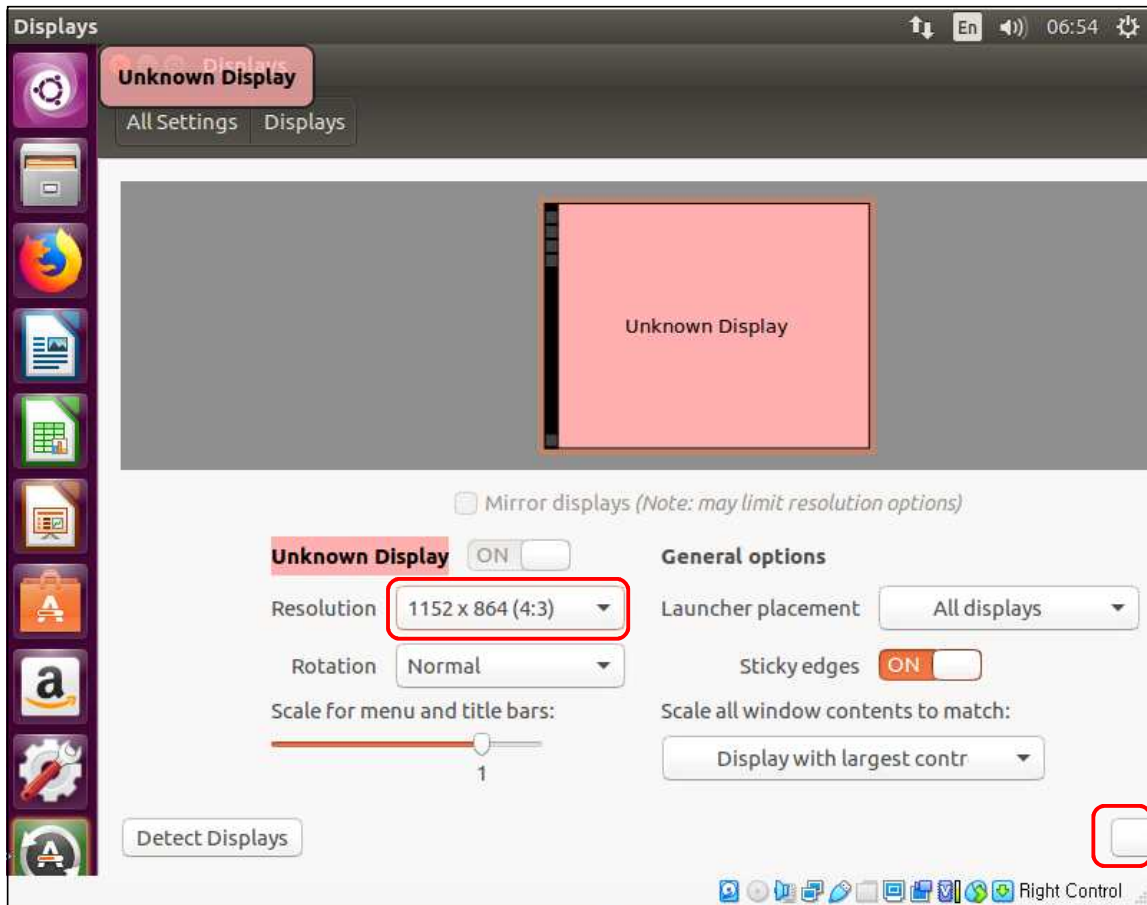


시간이 지나, 아래 그림처럼 된다면 화면 왼편에 Settings를 누른다.

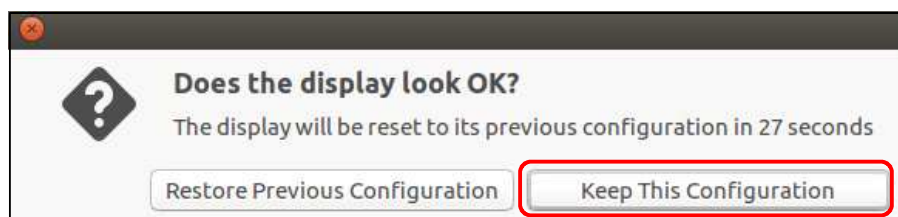


프로그램 설치 / 선택 / 가상 Ubuntu OS 설치

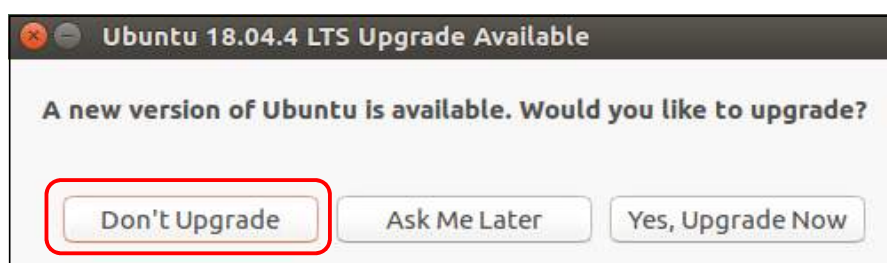
Hardware 탭 – Displays를 클릭, Resolution을 1152x864 이상으로 변경 후 우측 하단에 흰 버튼을 클릭한다.



아래와 같은 차에서는 Keep This Configuration 버튼을 누른다.



도중에 아래와 같은 알림이 뜨면, Don't Upgrade, OK를 선택한다.



프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅

화면 왼쪽 상단에 있는 firefox를 누른다.

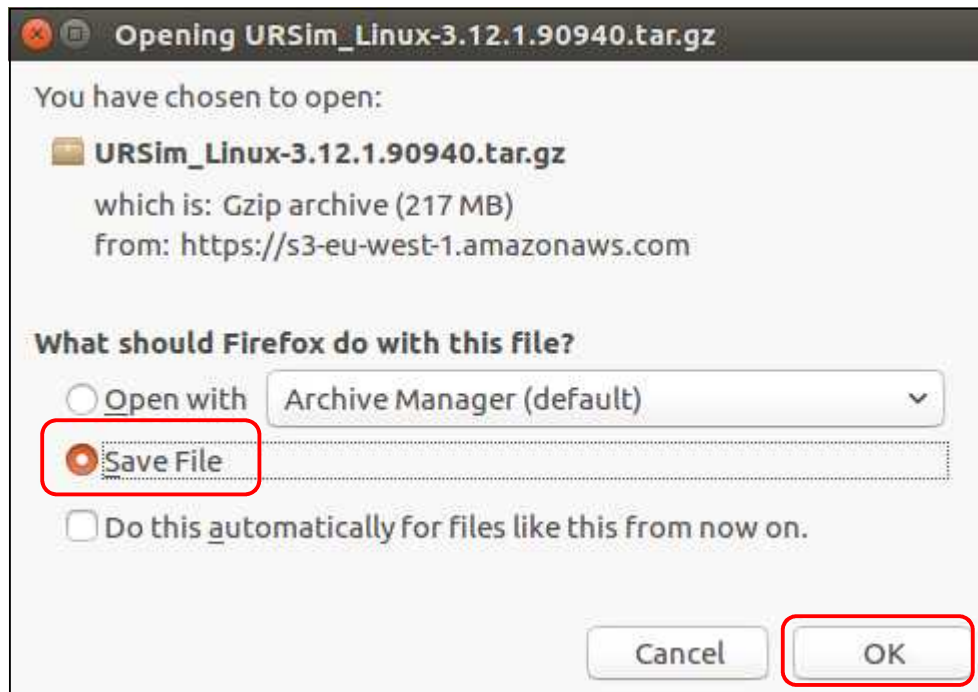


주소창에

<https://www.universal-robots.com/download/software-cb-series/simulator-linux/offline-simulator-cb-series-linux-ursim-3121/>를 입력하고 엔터를 누른다. (필요하다면 회원가입 후 다운로드 한다)

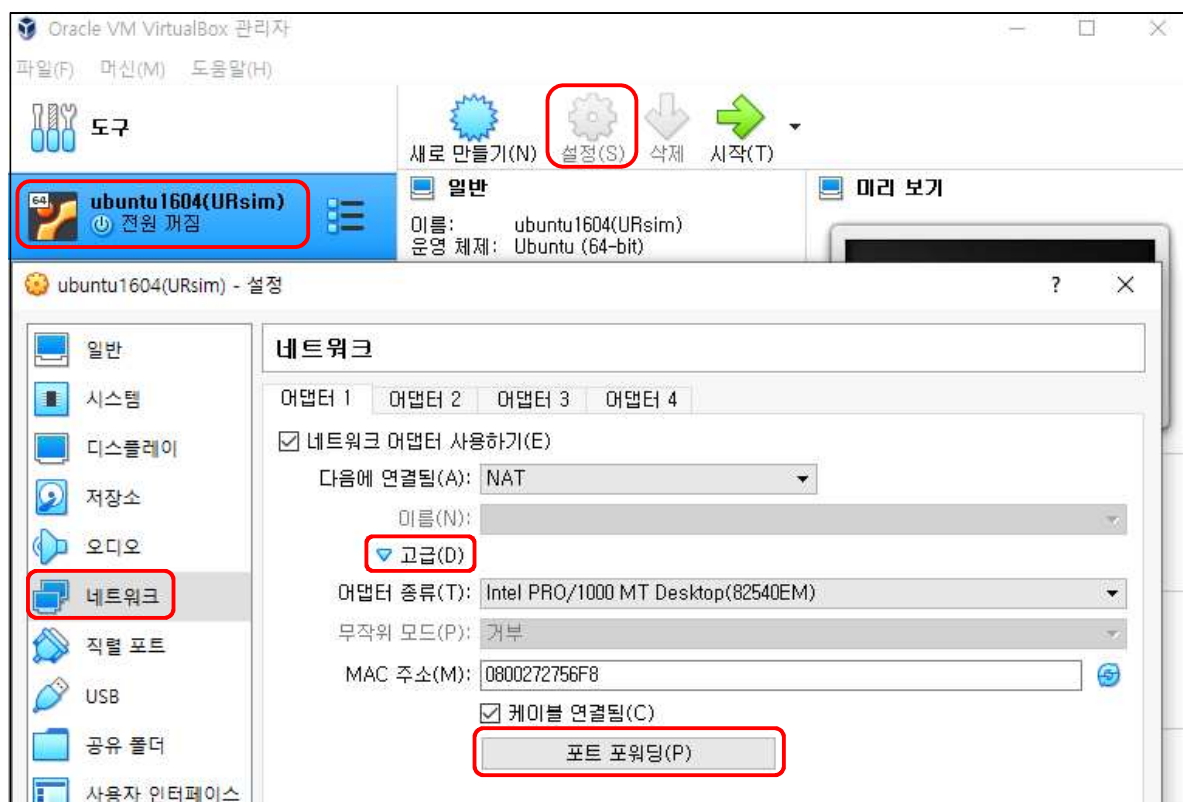
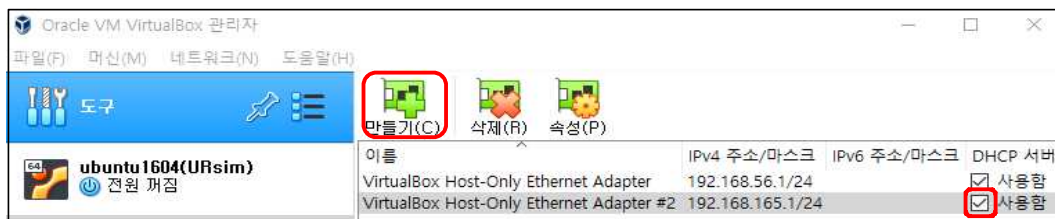
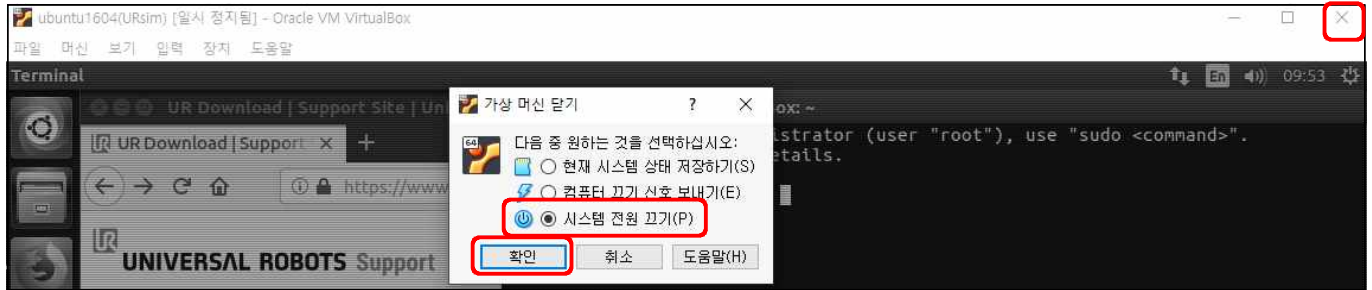


프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅

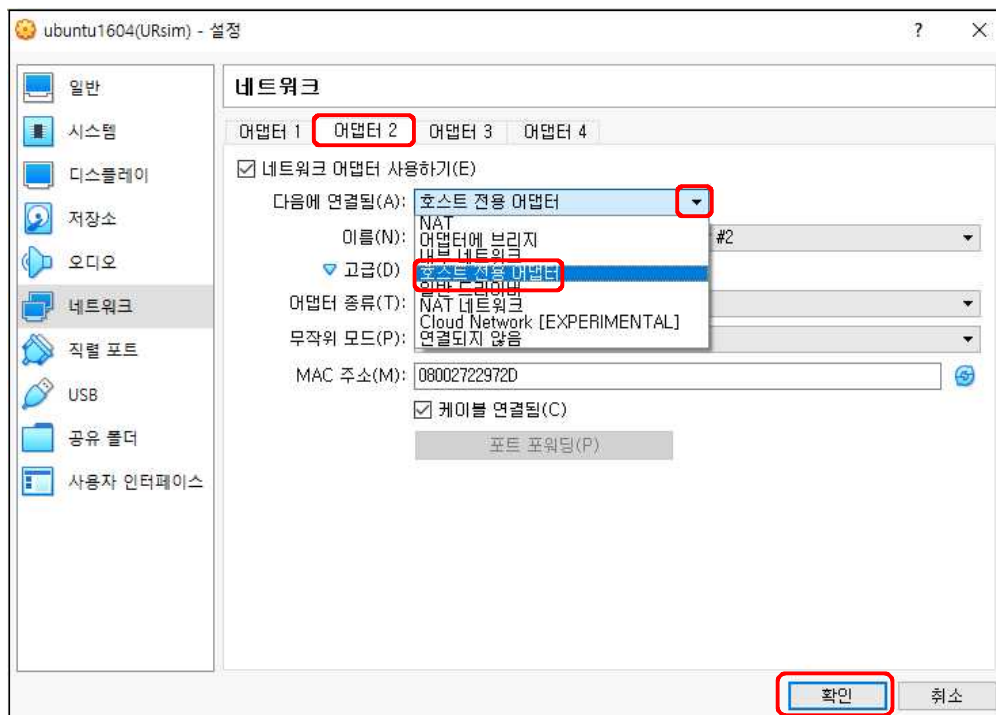


다운로드가 다 되면 다음 장으로 넘어간다.

프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅



프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅



프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅

키보드 'Ctrl' + 'Alt' + T를 눌러 터미널 창을 띄우고

sudo apt-get purge openssh-server 입력 후 엔터.

비밀번호 1q2w3e 입력 후 엔터.

```
ursim@ursim-VirtualBox: ~  
ursim@ursim-VirtualBox:~$ sudo apt-get purge openssh-server  
[sudo] password for ursim:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package 'openssh-server' is not installed, so not removed  
0 upgraded, 0 newly installed, 0 to remove and 382 not upgraded.  
ursim@ursim-VirtualBox:~$
```

※ 위 그림과 같이 나오지 않고 에러가 나오면
가상 Ubuntu를 종료 후 재시작 요망.

sudo apt-get install openssh-server 입력 후 엔터.

[Y/n] 물어보는 구간이 나오면 y 입력 후 엔터.

sudo service sshd start 입력 후 엔터.

```
ursim@ursim-VirtualBox:~$ sudo service sshd start  
ursim@ursim-VirtualBox:~$
```

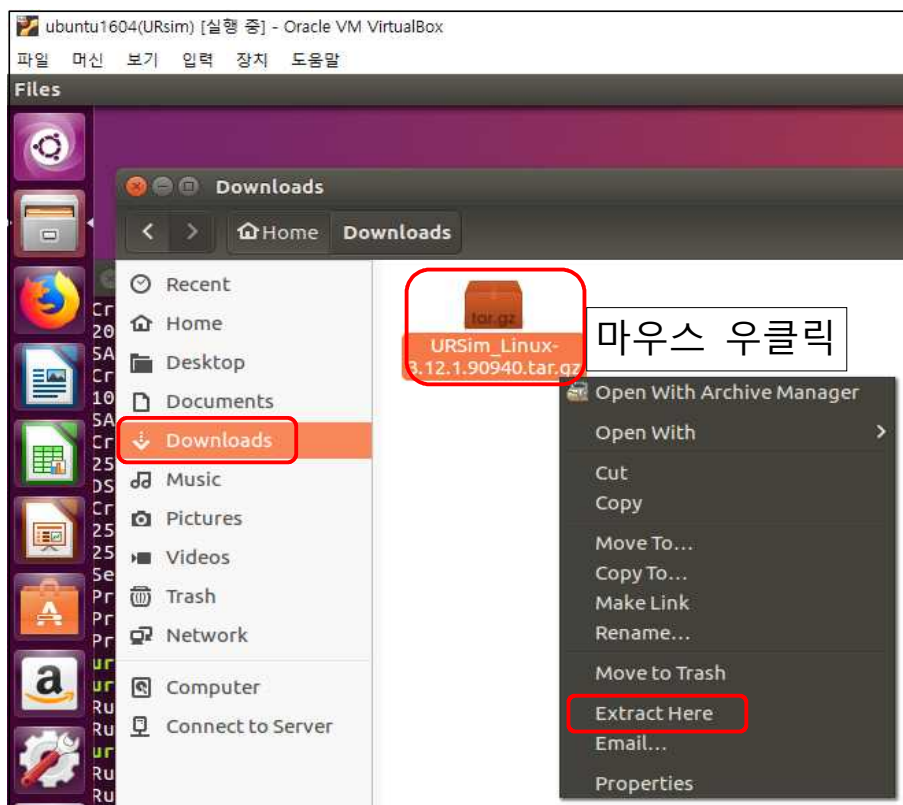
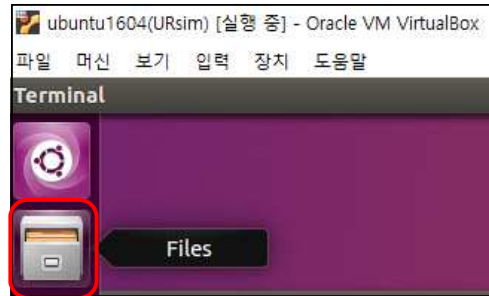
sudo ufw enable 입력 후 엔터.

sudo ufw allow 22 입력 후 엔터.

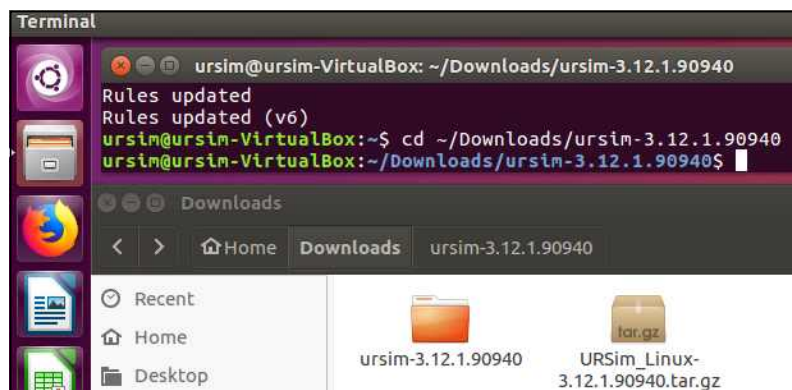
sudo ufw allow 30002 입력 후 엔터.

```
ursim@ursim-VirtualBox:~$ sudo ufw allow 22  
Rules updated  
Rules updated (v6)  
ursim@ursim-VirtualBox:~$ sudo ufw allow 30002  
Rules updated  
Rules updated (v6)  
ursim@ursim-VirtualBox:~$
```

프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅



다시 터미널창에서 `cd ~/Downloads/ursim-3.12.1.90940` 입력 후 엔터.



프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅

창을 닫고 새로운 터미널창 ('Ctrl' + 'Alt' + T)을 열고 ifconfig 입력 후 엔터를 누른다. 내용 중 192.168.의 내용을 찾고 숫자를 기억한다.

```
ursim@ursim-VirtualBox:~$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:27:56:f8
        inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::3282:96f4:a5b3:3820/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:88788 errors:0 dropped:0 overruns:0 frame:0
        TX packets:47837 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:73422339 (73.4 MB)  TX bytes:2884224 (2.8 MB)

enp0s8  Link encap:Ethernet  HWaddr 08:00:27:22:97:2d
        inet addr:192.168.165.3  Bcast:192.168.165.255  Mask:255.255.255.0
        inet6 addr: fe80::2e86:f46f:ad43:7e3b/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:48 errors:0 dropped:0 overruns:0 frame:0
        TX packets:104 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:8952 (8.9 KB)  TX bytes:14292 (14.2 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:236 errors:0 dropped:0 overruns:0 frame:0
        TX packets:236 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:18735 (18.7 KB)  TX bytes:18735 (18.7 KB)

ursim@ursim-VirtualBox:~$
```

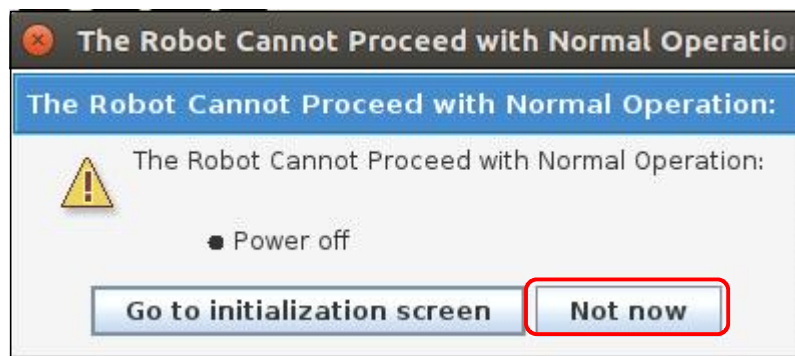
프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅

우분투 터미널 창('Ctrl' + 'Alt' + T)을 열고

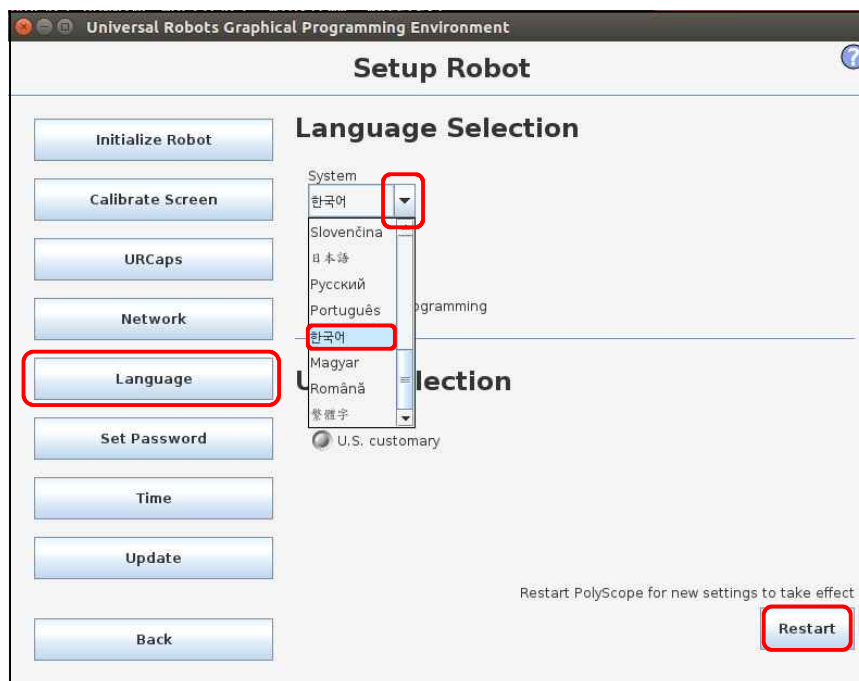
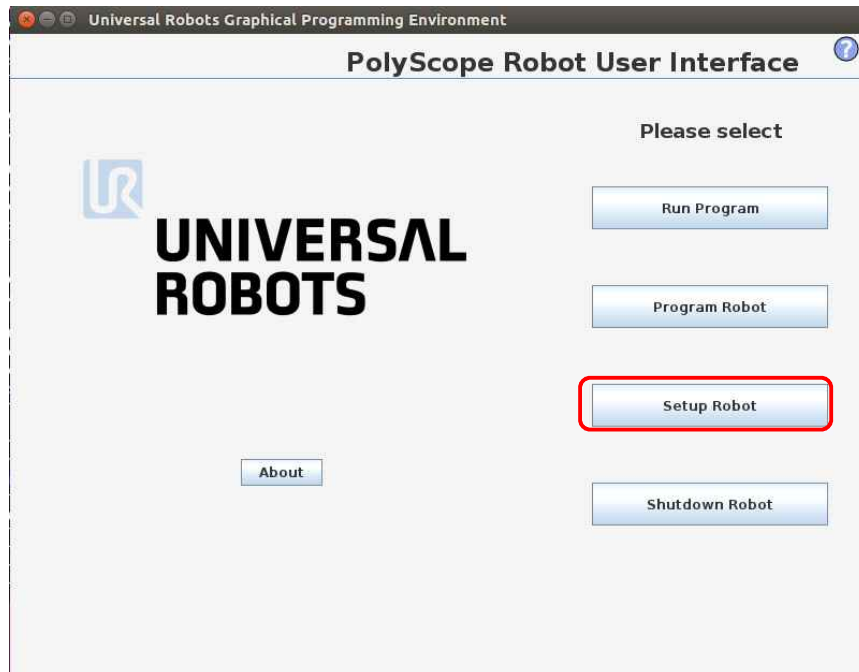
cd ~/Downloads/ursim-3.12.1.90940 입력 후 엔터를 누른다.

./start-ursim.sh UR3 혹은 ./start-ursim.sh UR5 (구동할 모델)입력 후 엔터를 누른다.

```
ursim@ursim-VirtualBox:~$ cd ~/Downloads/ursim-3.12.1.90940
ursim@ursim-VirtualBox:~/Downloads/ursim-3.12.1.90940$ ./start-ursim.sh UR5
```

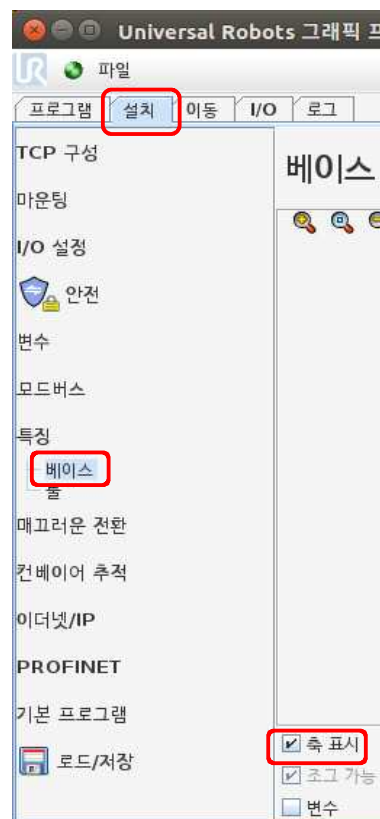
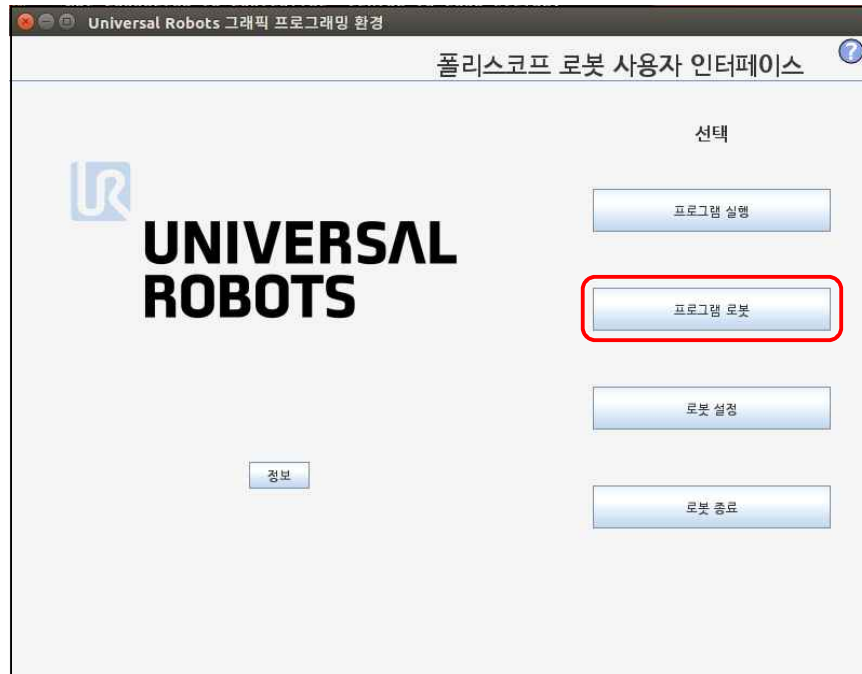


프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅



프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅

터미널 창에서 다시 ./start-ursim.sh UR3 혹은 ./start-ursim.sh UR5
(구동할 모델)입력 후 엔터를 누른다.



프로그램 설치 / 선택 / 가상 Ubuntu URsim 설치,세팅



홈 포지션(Home position)

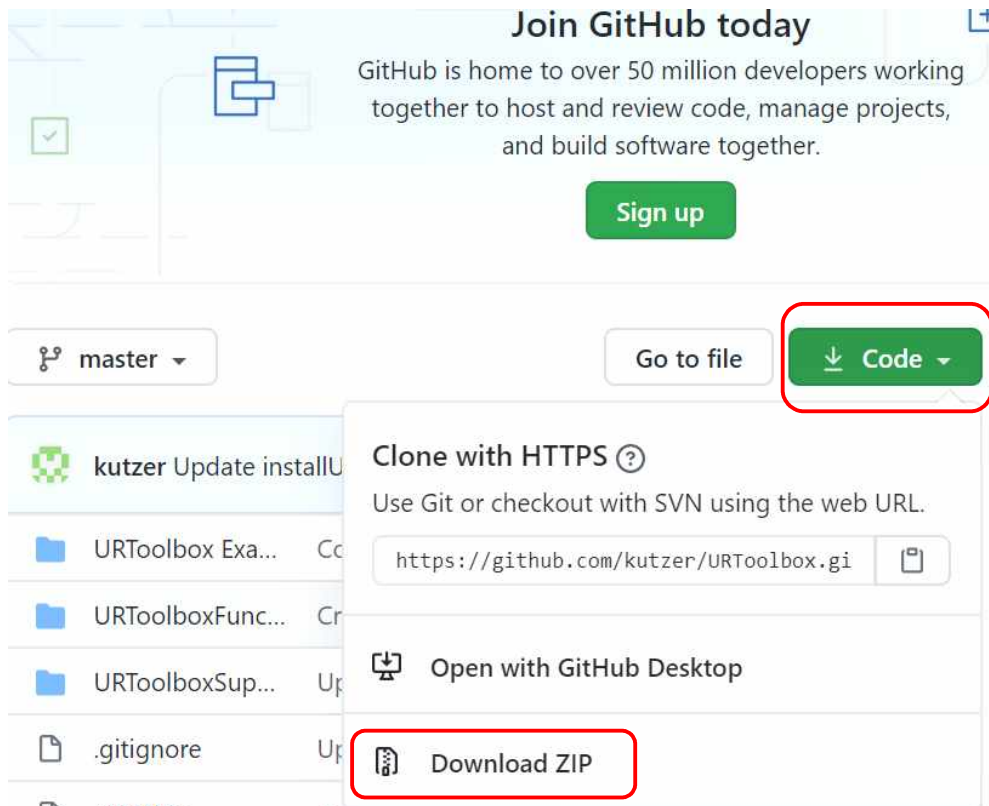
'자동' 버튼이 비활성화 될 때까지 꼭 누른다.

비활성화 되면 그때 취소 버튼을 누른다.

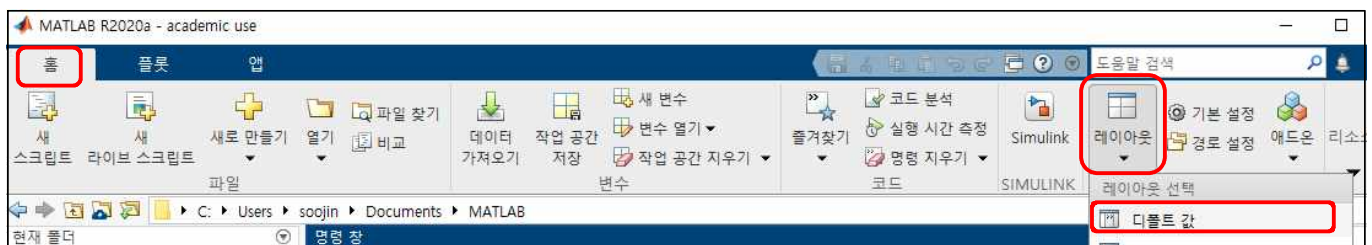


프로그램 설치 / 선택 / MATLAB URToolbox 설치

1. <https://github.com/kutzer/URToolbox/tree/f2eebcb153a3e6b171166774980c2777a1894c45> 에 접속하여 압축파일을 받는다.



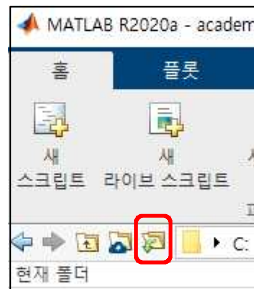
2. 받은 압축파일의 압축을 풀고 그대로 C:\Python37 폴더에 옮긴다.
3. MATLAB 아이콘에 마우스를 올려놓고 우클릭하여 관리자 권한으로 실행한다.
4. MATLAB 홈 탭 – 레이아웃 - (레이아웃 선택) 디폴트 값 을 선택한다.



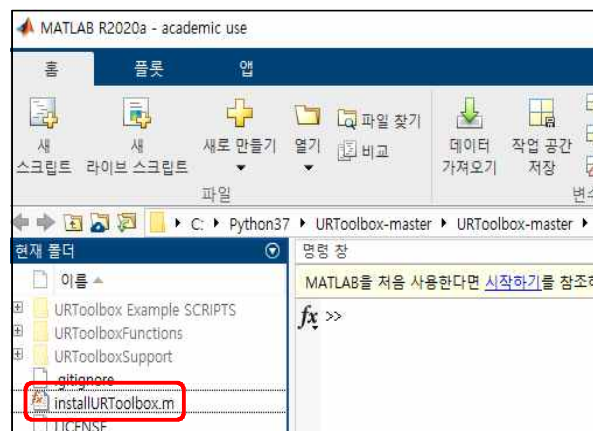
프로그램 설치 / 선택 / MATLAB URToolbox 설치

5. 아래 그림의 버튼을 눌러

C:\Python37\URToolbox-master\URToolbox-master
경로를 찾아 폴더선택 버튼을 누른다.



6. 창 왼편에 있는 installURToolbox.m을 더블클릭한다.



5. 편집기 창에 뜬 installURToolbox.m에 171,176줄에 있는
Python34의 문구를 Python37로 변경한다.

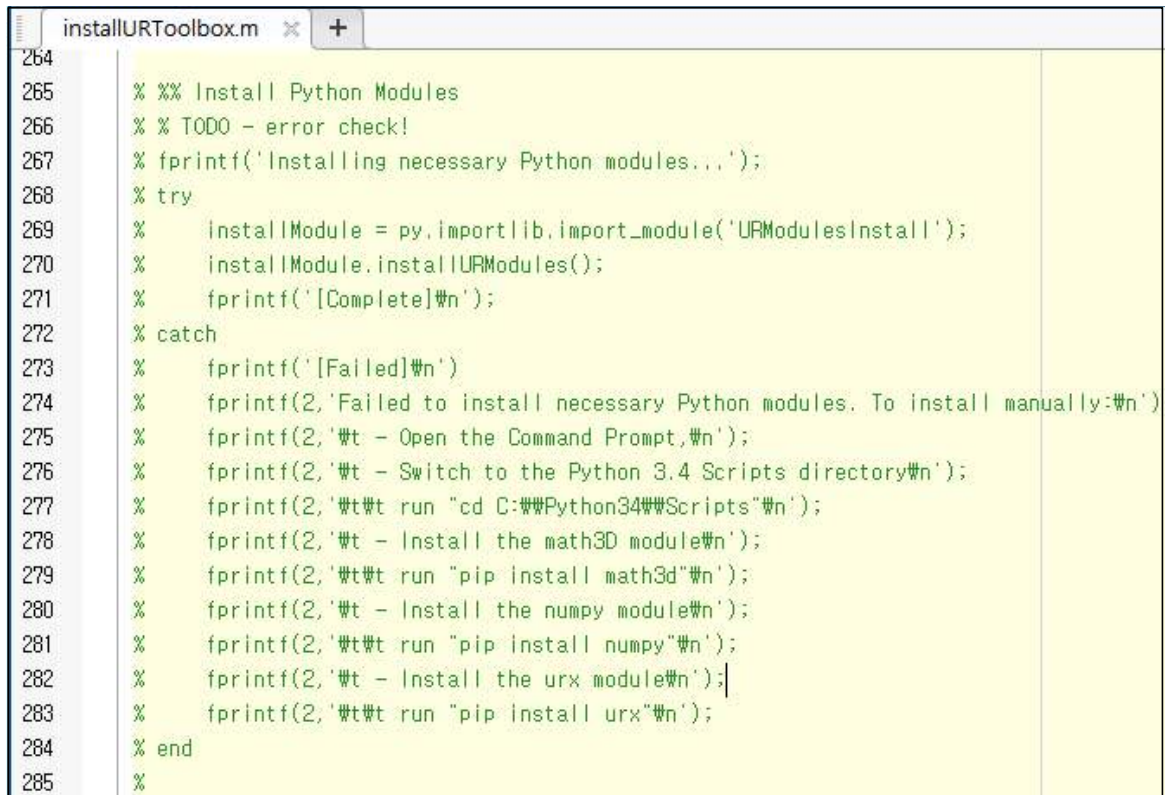
```
171 - if exist('C:\Python34\Lib', 'dir') ~= 7
172 -     % Python 3.4 is not installed in the stand
173 -     fprintf(2, 'Python 3.4.4 was not found in th
174 -     pythonRoot = uigetdir('', 'Select Python 3
175 - else
176 -     pythonRoot = fullfile('C:\Python34\Lib');
```

```
171 - if exist('C:\Python37\Lib', 'dir') ~= 7
172 -     % Python 3.4 is not installed in the stand
173 -     fprintf(2, 'Python 3.4.4 was not found in t
174 -     pythonRoot = uigetdir('', 'Select Python
175 - else
176 -     pythonRoot = fullfile('C:\Python37\Lib');
```

----->

프로그램 설치 / 선택 / MATLAB URToolbox 설치

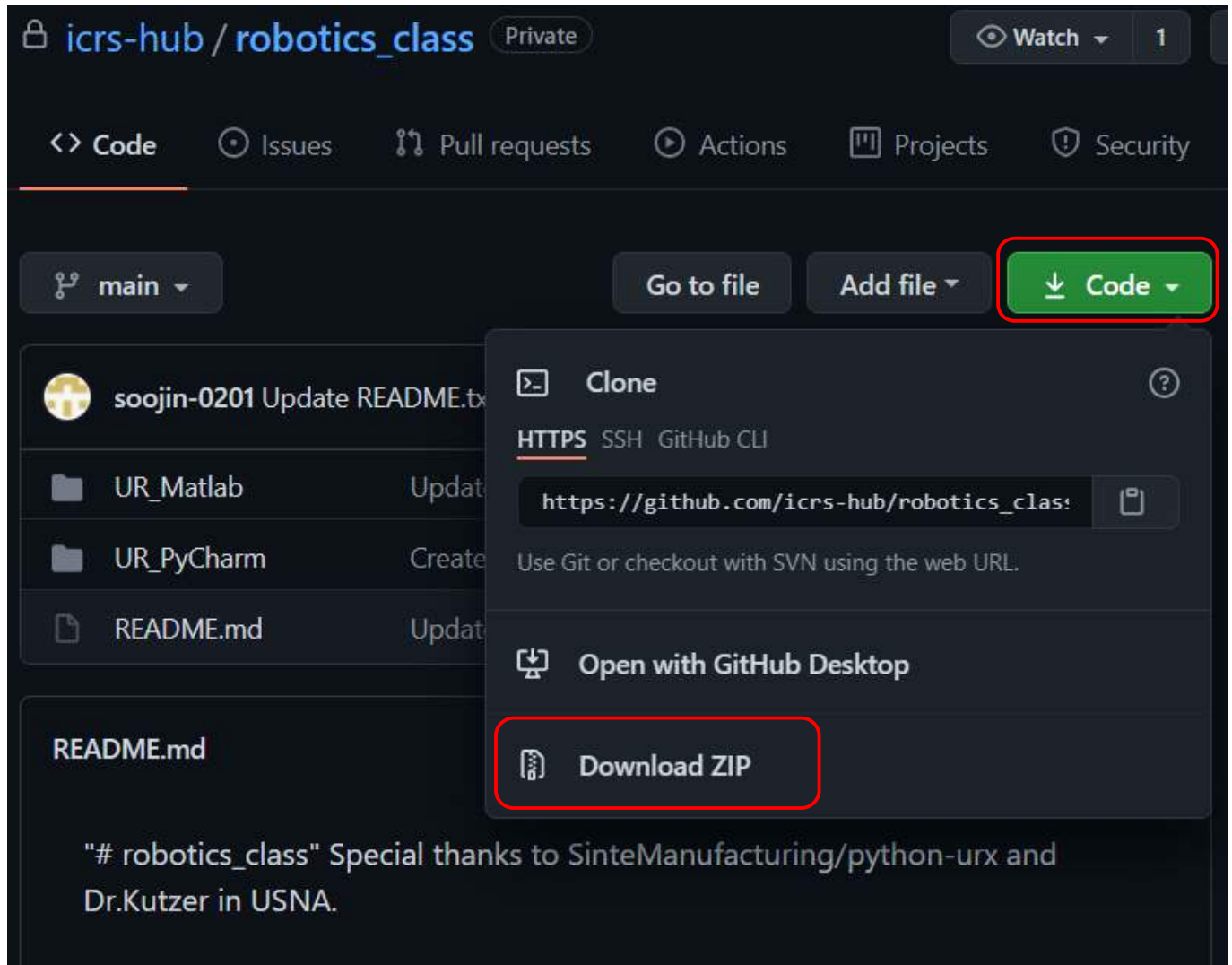
5. 편집기 창에 뜬 installURToolbox.m에 265줄부터 285줄까지 드래그 후 마우스 우클릭 - 주석을 클릭하고, F5키를 누른다.



```
264
265 % %% Install Python Modules
266 % % TODO - error check!
267 % fprintf('Installing necessary Python modules...');
268 % try
269 %     installModule = py.importlib.import_module('URModulesInstall');
270 %     installModule.installURModules();
271 %     fprintf('[Complete]\n');
272 % catch
273 %     fprintf('[Failed]\n')
274 %     fprintf(2,'Failed to install necessary Python modules. To install manually:\n');
275 %     fprintf(2,'%t - Open the Command Prompt,\n');
276 %     fprintf(2,'%t - Switch to the Python 3.4 Scripts directory\n');
277 %     fprintf(2,'%t%t run "cd C:\Python34\Scripts"\n');
278 %     fprintf(2,'%t - Install the math3D module\n');
279 %     fprintf(2,'%t%t run "pip install math3d"\n');
280 %     fprintf(2,'%t - Install the numpy module\n');
281 %     fprintf(2,'%t%t run "pip install numpy"\n');
282 %     fprintf(2,'%t - Install the urx module\n');
283 %     fprintf(2,'%t%t run "pip install urx"\n');
284 % end
285 %
```

예제 / PyCharm / 예제 파일 확인 및 실행

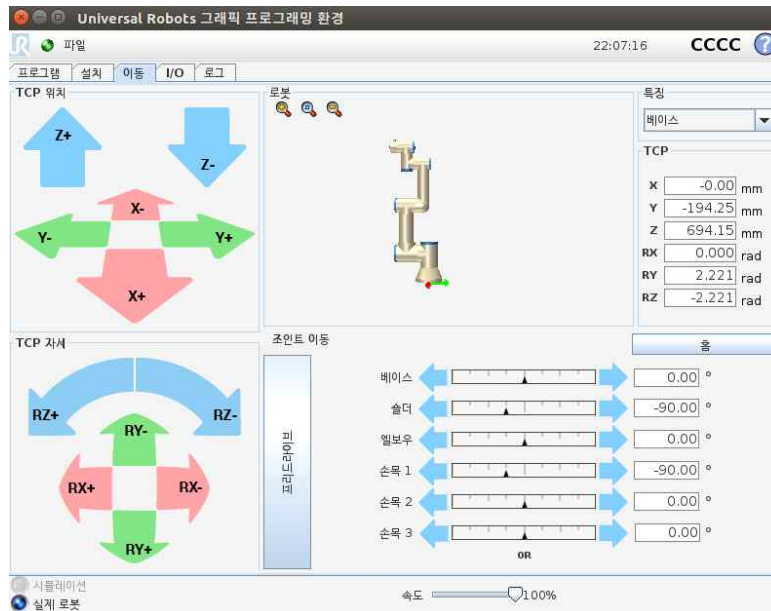
1. https://github.com/icrs-hub/robotics_class에 접속하여 압축파일을 받는다.



2. 받은 파일의 압축을 푼다.

예제 / PyCharm / 예제 파일 확인 및 실행

3. 페이지 27, 페이지 30를 따라하고 '프로그램 로봇' - '이동'탭을 눌러 아래 그림과 같은 창을 띄워놓는다.



4. 페이지 7부터 9까지 따라하고 압축을 해제한 폴더 내 UR_PyCharm 폴더에 있는 txt파일 내용을 복사하여 PyCharm에 붙여 넣는다.

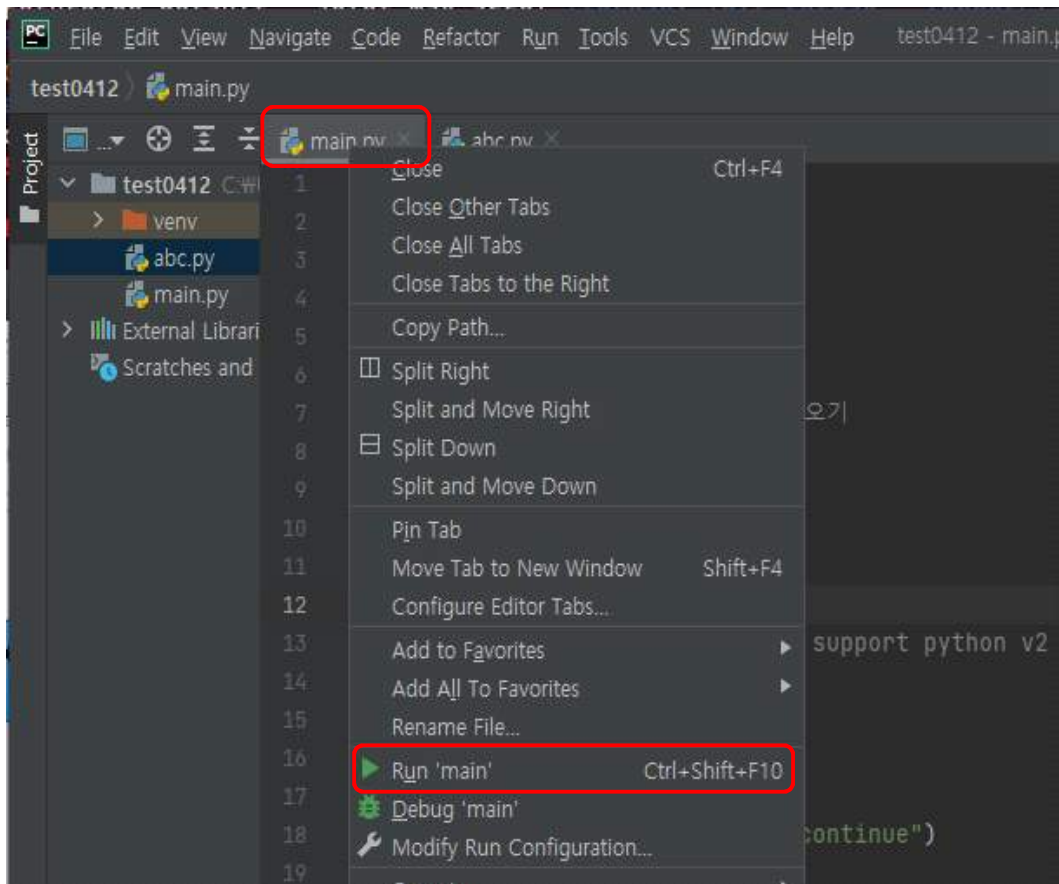
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help test0412 - main.py
test0412 main.py
Project
  test0412
    .venv
    main.py
    External Libraries
    Scratches and Snippets
  test0412
    main.py
1 from math import pi
2 import time
3 import sys
4 import csv
5 from typing import Union
6
7 import urx # urx 라이브러리 가져오기
8 import logging
9 import numpy as np
10 import sympy as sp
11 from urx import URRobot, Robot
12
13 if sys.version_info[0] < 3: # support python v2
14     input = raw_input
15
16 def wait():
17     if do_wait:
18         print("Click enter to continue")
19         input()
20
21
22 if __name__ == "__main__":
23     logging.basicConfig(level=logging.INFO)
24
25     do_wait = True
26     if len(sys.argv) > 1:
27         do_wait = False
28
29     rob: Union[Robot, URRobot] = urx.Robot("192.168.56.5") # 연결 설정 @@@@@@@@@ IP
30     # rob = urx.Robot("local host")
31     rob.set_tcp((0, 0, 0, 0, 0, 0)) # 로봇 플랜지를 공구 톱 변형으로 설정 TCP위치
32     rob.set_payload(0.98, (0, 0, 0)) # payload를 kg 단위로 설정
33     # Robot payload = 0.09kg(bracket) + 0.65kg(RG2) + ??? (work piece)
34     try:
35         # ----- 코딩 가능 구간 -----
36         f = open("D:\\samotina\\csv\\w\\", "w", newline="") # csv 파일 작성 가능하게 해주는
37         if __name__ == "__main__":
38             try:
39                 # csv 파일 작성 가능하게 해주는
40                 f.write("w")
41                 f.close()
42             except:
43                 pass
44     except:
45         pass
```

예제 / PyCharm / 예제 파일 확인 및 실행

5. 페이지 14에 3번을 따라하여 IP를 확인하고 PyCharm에 있는 예제 코드 중 `urx.Robot("~~~")`에 있는 IP를 수정해준다.

```
23 logging.basicConfig(level=logging.INFO)
24
25 do_wait = True
26 if len(sys.argv) > 1:
27     do_wait = False
28
29 rob: Union[Robot, URRobot] = urx.Robot("192.168.56.5") # 연결 설정
30 # rob = urx.Robot("local host")
31 rob.set_tcp((0, 0, 0, 0, 0, 0)) # 로봇 플랜지를 공구 팁 변형으로 설정
32 rob.set_payload(0.98, (0, 0, 0)) # payload를 kg 단위로 설정
33 # Robot payload = 0.09kg(bracket) + 0.65kg(RG2) + ??? (work piece
```

6. 좌측 상단에 `main.py` 혹은 현재 활성화 되어있는 탭 (예제가 담겨있는 탭) 위에 마우스를 놓고 우 클릭 - Run을 누른다. (활성화 여부는 탭 색깔이 조금 차이가 있음)



예제 / PyCharm / 명령어 설명

[로봇을 동작시키는 명령어]

```
rob.movej([0,0,0,0,0,0],acc=0.1,vel=0.5,wait=False)
```

TCP의 위치, 회전을 입력하여 움직이게 한다.

크게 나누어 배열, 가속도, 속도를 입력하는 구간으로 나뉜다.

배열 : [x,y,z,Rx,Ry,Rz] x,y,z는 미터 m, Rx,Ry,Rz는 라디안

가속도 : m/s^2

속도 : m/s

```
rob.movej([0,0,0,0,0,0],acc=0.1,vel=0.1,wait=False)
```

각 관절마다 라디안으로 값을 지정하여 움직이게 한다.

크게 나누어 배열, 가속도, 속도를 입력하는 구간으로 나뉜다.

배열 : [베이스, 숄더, 엘보우, 손목1,2,3] 단위는 모두 라디안

가속도 : rad/s^2

속도 : rad/s

[데이터를 파일로 저장(.csv)하는 것을 도와주는 명령어]

```
open('E:\UUBUNTU\sampling.csv', 'w', newline='')
```

E:\UUBUNTU 경로로 sampling.csv라는 파일을 생성한다.

```
makefile.writerow([항목1,항목2,~,항목7])
```

(기록하려는 단위가 1행일 때만 해당 명령어 사용)

내용 : 기록하고있는 파일에서 줄바꿈을 하여

항목1부터 항목 7사이에 ,를 친다.

(엑셀 파일의 경우에 .csv 확장자를 읽을 수 있어,

PyCharm실행을 마친 후 csv를 실행해보면

자동으로 엑셀로 창이 열림)

예제 / PyCharm / 명령어 설명

[로봇의 Joint 값을 읽어오는 명령어]

```
a = rob.getj()
```

a에 6칸 배열로 현재 로봇의 각 Joint값을 가져온다.

a배열 : [베이스, 숄더, 엘보우, 손목1,2,3] 단위는 모두 라디안

[변수, 대입 명령어]

```
x = sp.symbols('x')
```

x라는 미지수 변수를 사용할 것을 선언한다.

```
eq1_1.subs(x, t_now)
```

eq1_1이라는 식에 있는 미지수 x에 t_now라는 값을 대입한다.

x^{**2} 는 x 와 같은 역할을 한다. x^{**3} 은 x^3 와 같은 역할을 한다.

```
time.time()
```

1970년 1월 1일 자정 이후로 누적된 초를 가져온다.

[기타 명령어 설명]

```
wait()
```

PyCharm 실행 시 창 하단에 올라오는 조그마한 창에 엔터키를 누르기 전까지 대기하는 명령어다.

시뮬레이션의 로봇이 Home Position까지 도달했을 때 확인하고 누르면 된다.

예제 / PyCharm / Robotiq 2-finger gripper 명령어 설명

[Robotiq 2-finger gripper의 tool 너비를 조절하는 명령어]

robotiqgrip.gripper_action(x)

x에는 0에서부터 255까지의 숫자가 들어갈 수 있다.

0은 최대로 tool을 벌린 상태, 255는 최소로 닫은 상태이다.

예제 / PyCharm / 예제에 사용된 변수명 설명

[각 Joint별 수식 입력]

eq1_1, eq1_2, ~~ , eq1_6

각 스텝별로 수식을 구분하기 위해 예제와 같이
eq 바로 뒤에있는 숫자를 수정할 수 있도록 한다.

(수정하며 그 아래에 있는 rob.movej(~~)에
있는 숫자도 필히 바꾸어야 한다)

언더바_ 뒤에 있는 숫자는 1번부터 베이스, 2번은 솔더 ~~
6번은 손목3을 의미한다.

(그러므로 언더바 뒤에있는 숫자는 변경하지 말 것)

[시간 변수명]

t_begin time.time()의 값을 저장하고 있다.

t_now 현재시간(time.time())으로부터 이미 저장한 t_begin값을 뺀다.
이렇게 값을 갱신해주면 처음 시작으로부터 현재까지의
시간을 알 수 있다.

time_while_2

t_now(시작부터 현재까지 걸린 시간) - 이전 루프 시간이다.

t_now를 통해 얻는 시작부터 누적시간이 아닌

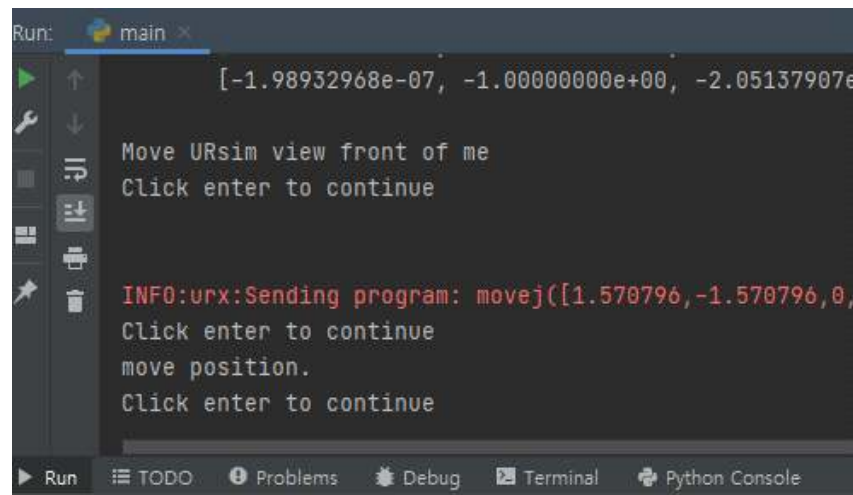
해당 루프 내부의 시간을 사용하고자 할 때 사용한다.

while문을 늘려갈 때 필히 뒤에 숫자를 변경하며
사용하기를 바라며 사용유무는 자유롭게 선택한다.

예제 / PyCharm / 예제 결과 (잘 된 경우)

[PyCharm]

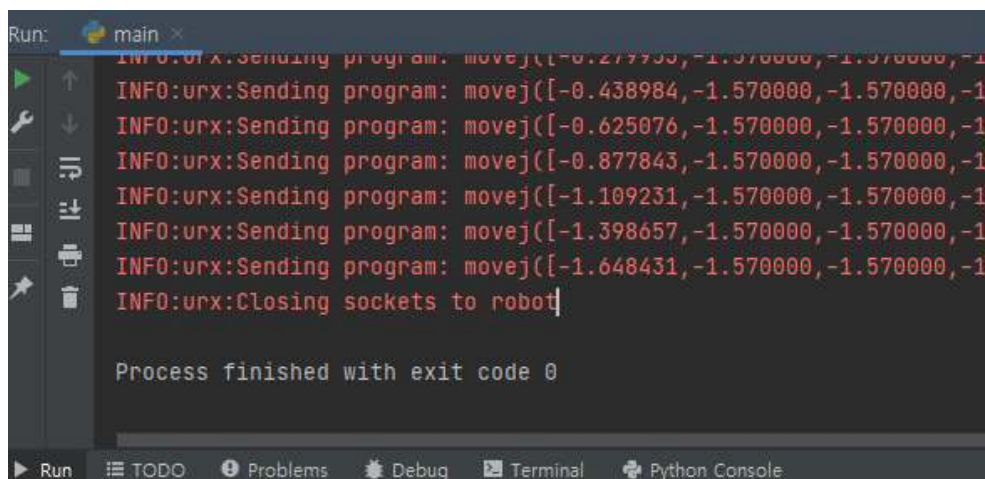
아래 그림에 쓰인 것과 같이 'Click enter to continue'라는 문구가 PyCharm 하단 창에 생성되며, 하단 창을 한번 클릭 후 엔터를 누르면 코드에 의해 시뮬레이션이 진행된다.



```
Run: main x
[-1.98932968e-07, -1.00000000e+00, -2.05137907e
Move URsim view front of me
Click enter to continue

INFO:urx:Sending program: movej([1.570796,-1.570796,0,
Click enter to continue
move position.
Click enter to continue
```

코드 실행을 모두 마치면 아래 그림과 같이 '~~code 0'이 생성된다.
그리고 앞에서 설명한 csv파일이 생성된다.
(절대 프로그램 실행 중 csv파일을 열지 말 것)



```
Run: main x
INFO:urx:Sending program: movej([-0.279733,-1.570000,-1.570000,-1
INFO:urx:Sending program: movej([-0.438984,-1.570000,-1.570000,-1
INFO:urx:Sending program: movej([-0.625076,-1.570000,-1.570000,-1
INFO:urx:Sending program: movej([-0.877843,-1.570000,-1.570000,-1
INFO:urx:Sending program: movej([-1.109231,-1.570000,-1.570000,-1
INFO:urx:Sending program: movej([-1.398657,-1.570000,-1.570000,-1
INFO:urx:Sending program: movej([-1.648431,-1.570000,-1.570000,-1
INFO:urx:Closing sockets to robot

Process finished with exit code 0
```

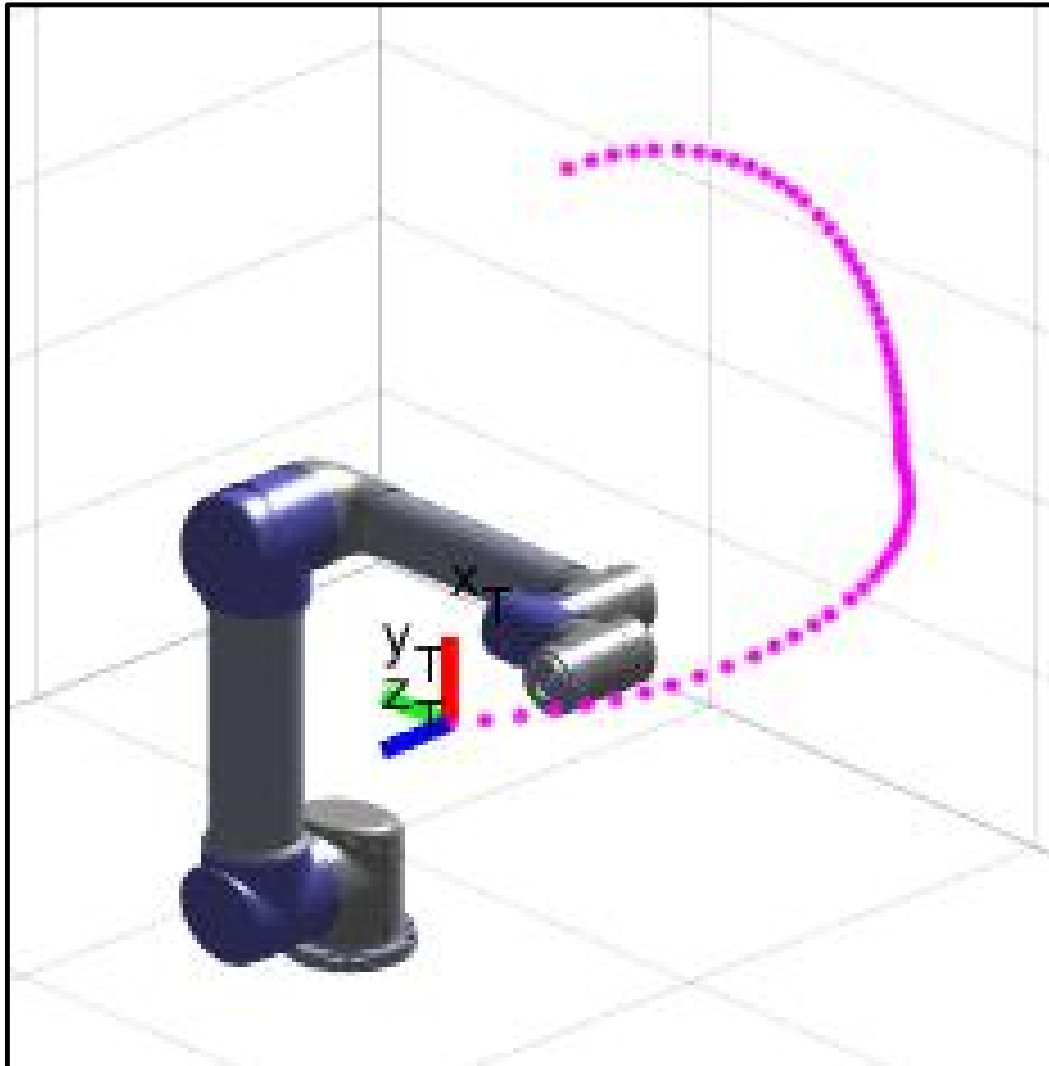
[URsim]

PyCharm에서 엔터키를 누르고 나면 그에 따라 동작한다.

URsim 창 우측상단에는 TCP의 x, y, z, Rx, Ry, Rz값을 보여준다.

예제 / PyCharm / 예제 결과 (잘 된 경우)

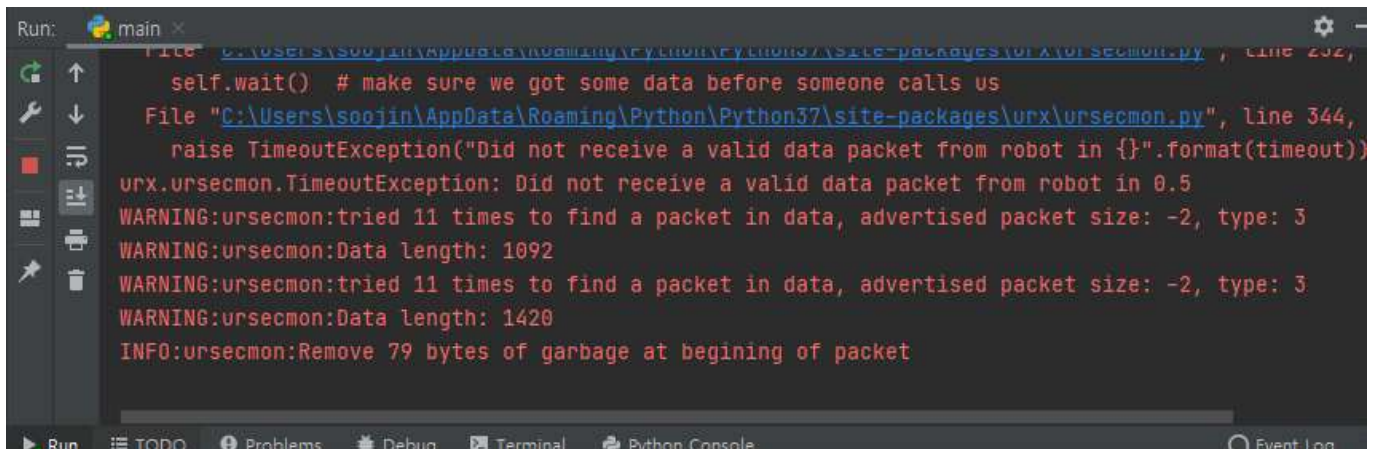
[이상적인 로봇의 움직임]



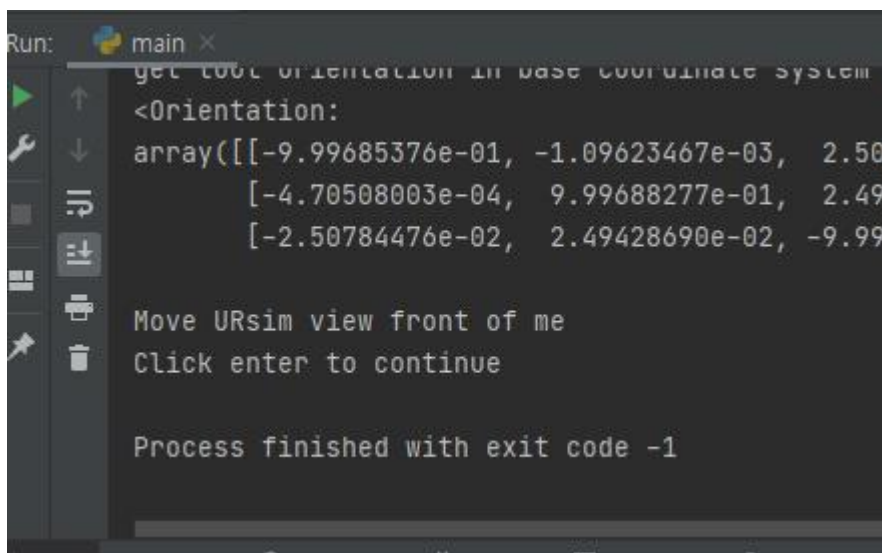
예제 / PyCharm / 예제 결과 (오류 현상)

[PyCharm]

위 튜토리얼을 모두 따라하더라도 시뮬레이션이 움직이지 않고
아래 두 그림 혹은 'Process finished with exit code 1' 등
'~~ code 숫자'가 나온다면

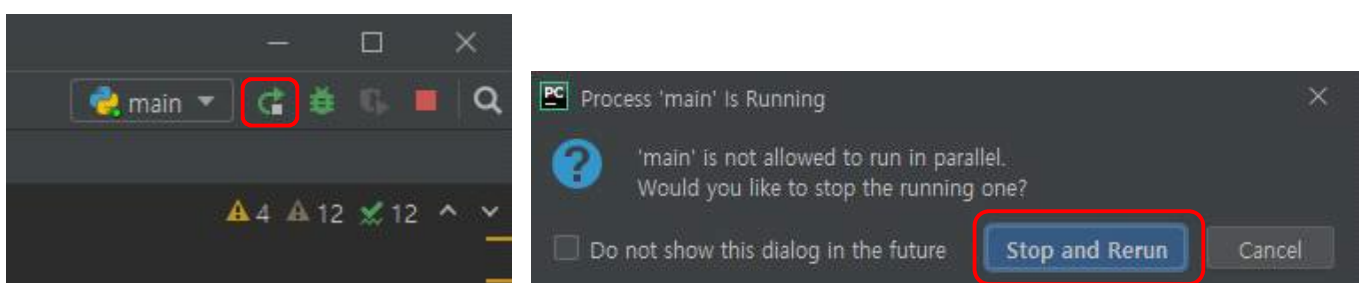


```
Run: main x
File: C:\Users\soojin\AppData\Roaming\Python\Python37\site-packages\urx\ursecmon.py, line 402,
self.wait() # make sure we got some data before someone calls us
File "C:\Users\soojin\AppData\Roaming\Python\Python37\site-packages\urx\ursecmon.py", line 344,
raise TimeoutException("Did not receive a valid data packet from robot in {}".format(timeout))
urx.ursecmon.TimeoutException: Did not receive a valid data packet from robot in 0.5
WARNING:ursecmon:tried 11 times to find a packet in data, advertised packet size: -2, type: 3
WARNING:ursecmon:Data length: 1092
WARNING:ursecmon:tried 11 times to find a packet in data, advertised packet size: -2, type: 3
WARNING:ursecmon:Data length: 1420
INFO:ursecmon:Remove 79 bytes of garbage at beginning of packet
```



```
Run: main x
get tool orientation in base coordinate system
<Orientation:
array([[ -9.99685376e-01, -1.09623467e-03,  2.509
        [-4.70508003e-04,  9.99688277e-01,  2.49
        [-2.50784476e-02,  2.49428690e-02, -9.99
Move URsim view front of me
Click enter to continue
Process finished with exit code -1
```

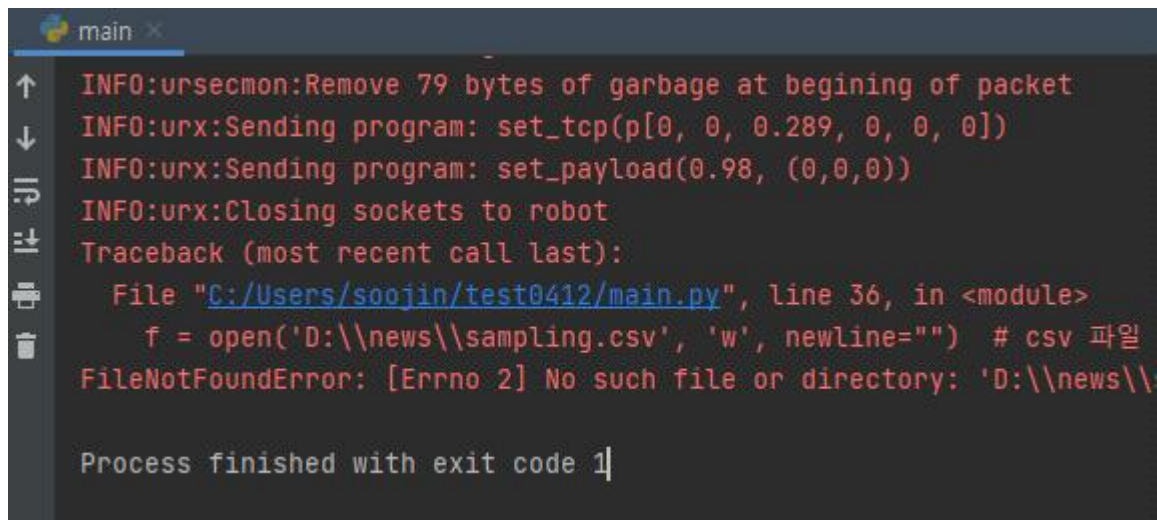
PyCharm 우측 상단에 있는 초록색 삼각형 'run'버튼을 다시 눌러보거나
아래 그림을 따라한다.



예제 / PyCharm / 예제 결과 (오류 현상)

[PyCharm]

아래 그림과 같이 오류 내용 중 `f = open('파일경로 ~~~')`에 대한 내용이 나와서 '~~code 1'이 나온다면 [아래 그림 파일을 예시로] `sampling.csv` 파일을 만들고자 하는 `D:\news` (D드라이브에 `news`라고 하는 폴더)가 없는 것이므로 폴더 명을 수정하거나 새롭게 폴더를 생성해주면 된다.



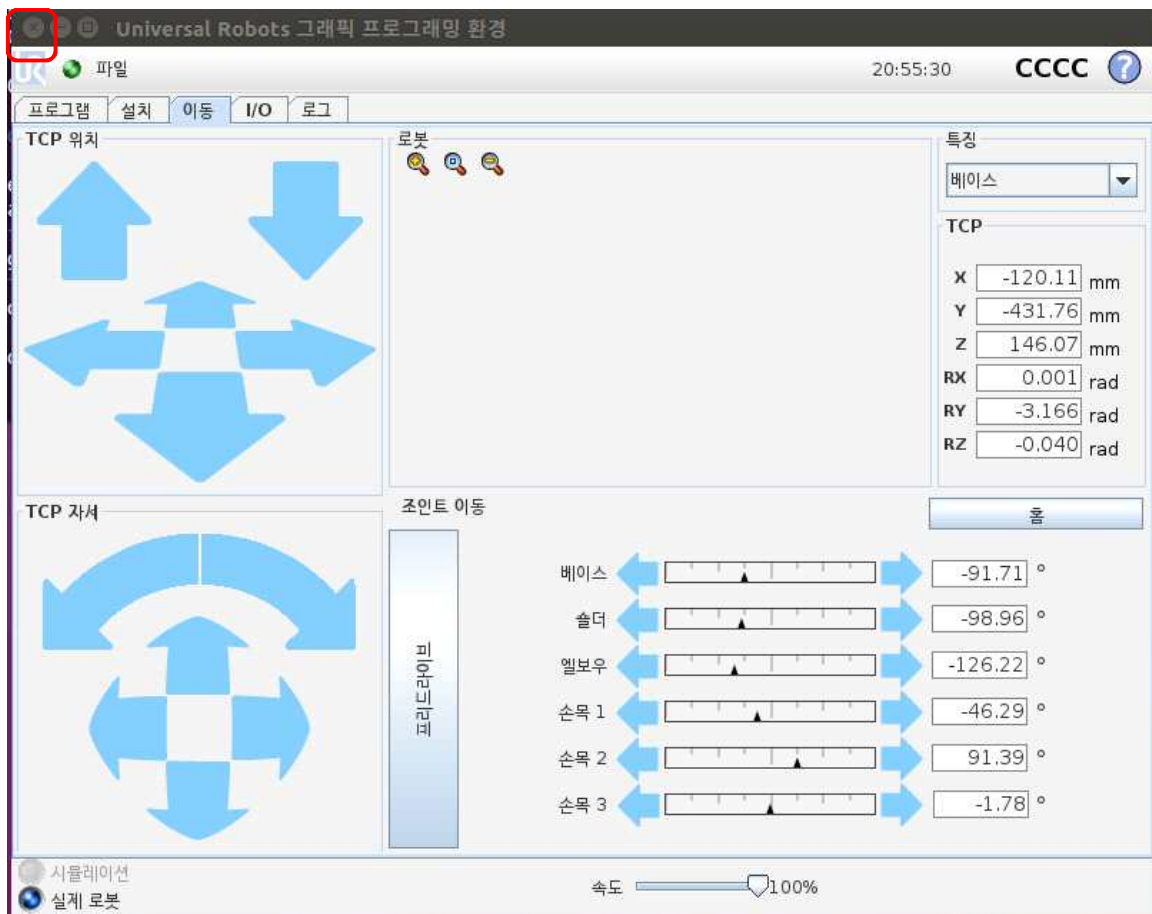
```
main x
↑ INFO:ursecmon:Remove 79 bytes of garbage at begining of packet
↓ INFO:urx:Sending program: set_tcp(p[0, 0, 0.289, 0, 0, 0])
: INFO:urx:Sending program: set_payload(0.98, (0,0,0))
: INFO:urx:Closing sockets to robot
: Traceback (most recent call last):
:   File "C:/Users/soojin/test0412/main.py", line 36, in <module>
:     f = open('D:\\news\\sampling.csv', 'w', newline="") # csv 파일
:   FileNotFoundError: [Errno 2] No such file or directory: 'D:\\news\\

Process finished with exit code 1
```

예제 / PyCharm / 예제 결과 (오류 현상)

[URsim]

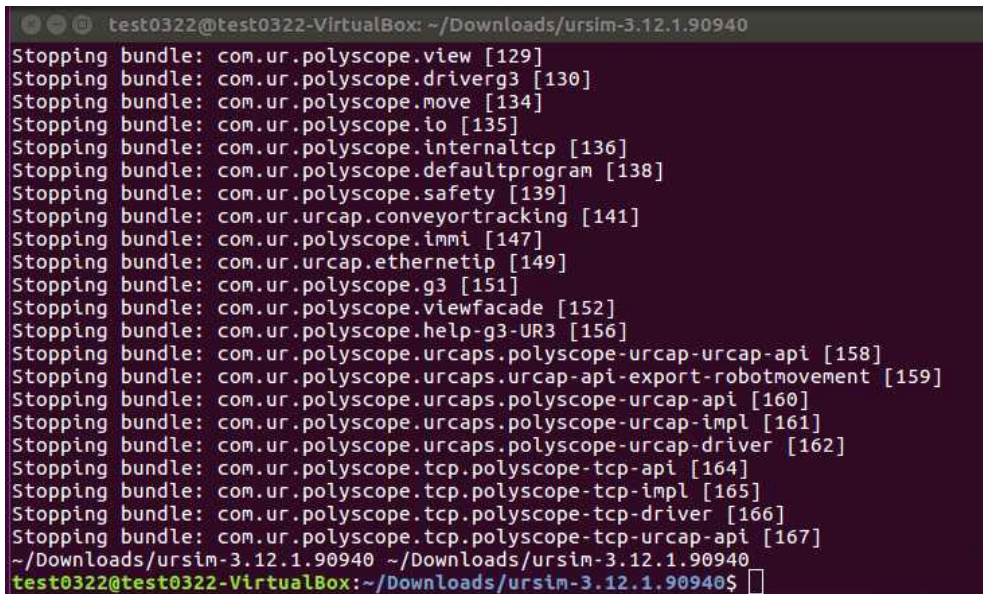
아래 화면까지 접근했을 때, 그림과 같이 창 중간 '로봇'이라고 하는 탭에 아무런 그림이 없다면 창 왼쪽 상단에 X를 눌러 창을 없앤다.



예제 / PyCharm / 예제 결과 (오류 현상)

[URsim]

그러면 ./start-ursim.sh UR5를 입력했던 창이 아래 그림처럼 명령어를 입력할 수 있게 바뀐다.

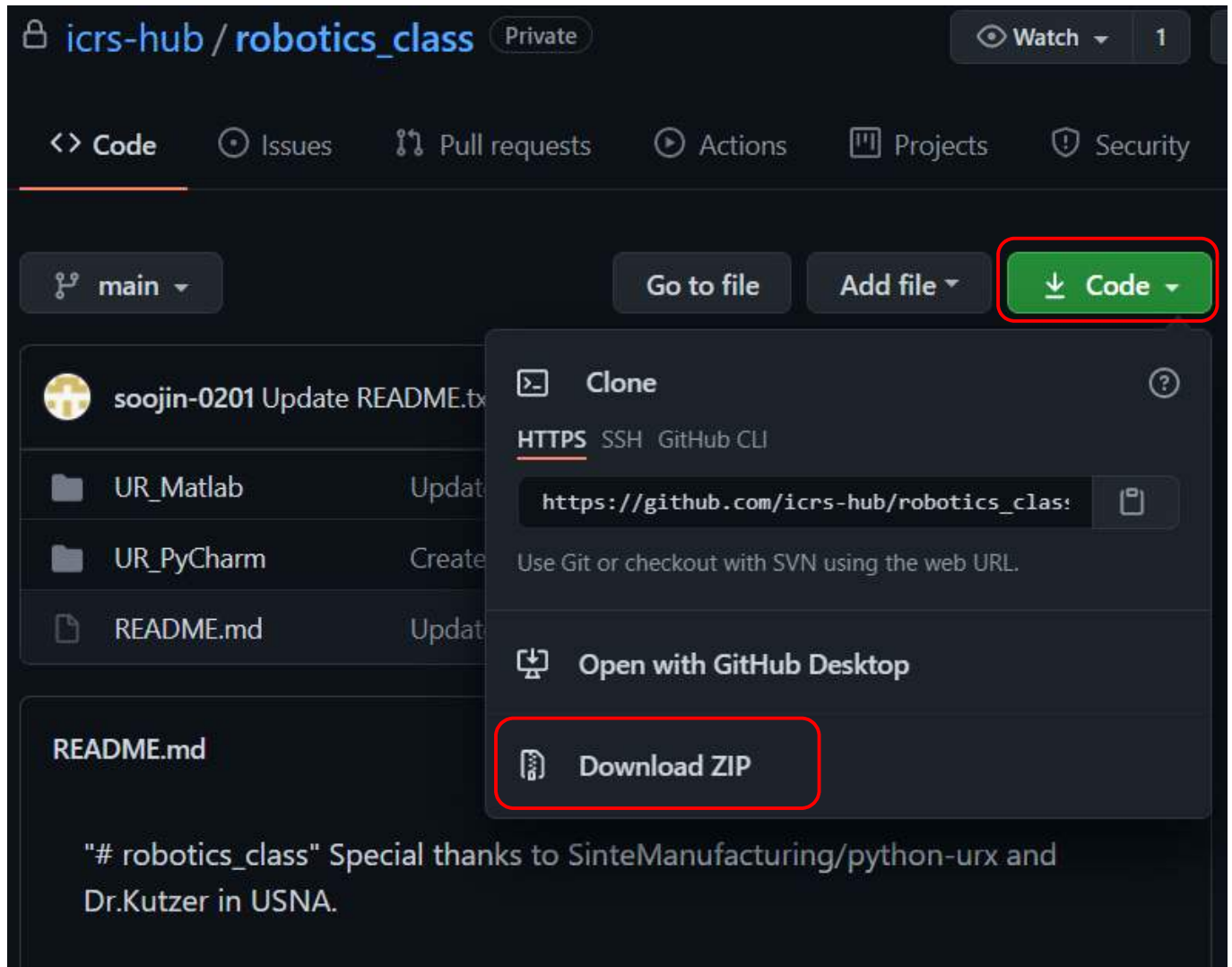


```
test0322@test0322-VirtualBox: ~/Downloads/ursim-3.12.1.90940
Stopping bundle: com.ur.polyscope.view [129]
Stopping bundle: com.ur.polyscope.driverg3 [130]
Stopping bundle: com.ur.polyscope.move [134]
Stopping bundle: com.ur.polyscope.io [135]
Stopping bundle: com.ur.polyscope.internaltcp [136]
Stopping bundle: com.ur.polyscope.defaultprogram [138]
Stopping bundle: com.ur.polyscope.safety [139]
Stopping bundle: com.ur.urcap.conveyortracking [141]
Stopping bundle: com.ur.polyscope.immi [147]
Stopping bundle: com.ur.urcap.ethernetip [149]
Stopping bundle: com.ur.polyscope.g3 [151]
Stopping bundle: com.ur.polyscope.viewfacade [152]
Stopping bundle: com.ur.polyscope.help-g3-UR3 [156]
Stopping bundle: com.ur.polyscope.urcaps.polyscope-urcap-urcap-api [158]
Stopping bundle: com.ur.polyscope.urcaps.urcap-api-export-robotmovement [159]
Stopping bundle: com.ur.polyscope.urcaps.polyscope-urcap-api [160]
Stopping bundle: com.ur.polyscope.urcaps.polyscope-urcap-impl [161]
Stopping bundle: com.ur.polyscope.urcaps.polyscope-urcap-driver [162]
Stopping bundle: com.ur.polyscope.tcp.polyscope-tcp-api [164]
Stopping bundle: com.ur.polyscope.tcp.polyscope-tcp-impl [165]
Stopping bundle: com.ur.polyscope.tcp.polyscope-tcp-driver [166]
Stopping bundle: com.ur.polyscope.tcp.polyscope-tcp-urcap-api [167]
~/Downloads/ursim-3.12.1.90940 ~/Downloads/ursim-3.12.1.90940
test0322@test0322-VirtualBox:~/Downloads/ursim-3.12.1.90940$
```

해당 터미널 창을 클릭한 상태로 키보드 위 방향키를 눌러 이전에 사용한 명령어 './start-ursim.sh UR5'를 다시 불러오고 엔터키를 눌러 실행한다.

예제 / MATLAB / 예제 파일 확인 및 실행

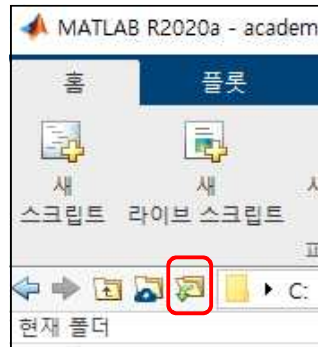
1. https://github.com/icrs-hub/robotics_class에 접속하여 압축파일을 받는다.



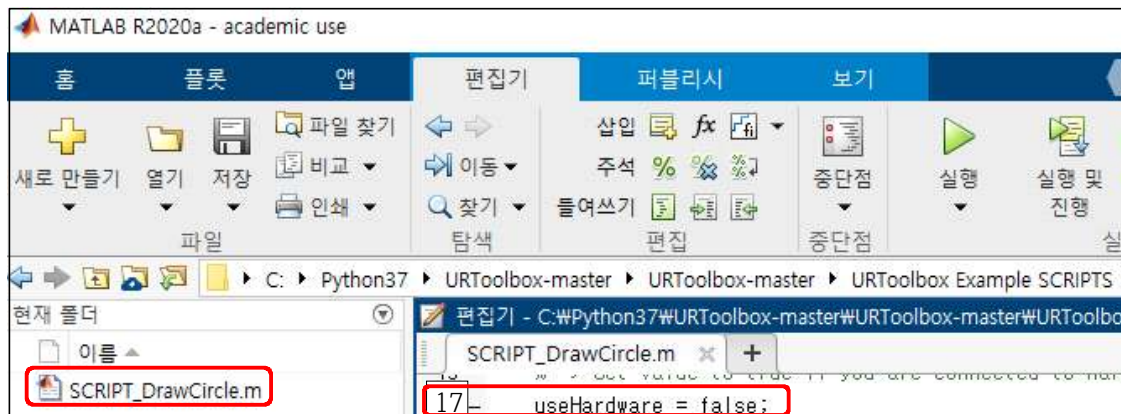
2. 받은 파일의 압축을 푼다.
3. MATLAB을 마우스 오른쪽 클릭 -> 관리자 모드로 실행한다.

예제 / MATLAB / 예제 파일 확인 및 실행

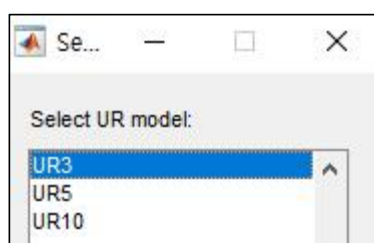
4. 아래 그림의 버튼을 눌러 이전페이지 2번에서 압축 해제한 폴더 안에 UR_Matlab->students를 찾아 폴더선택 버튼을 누른다.



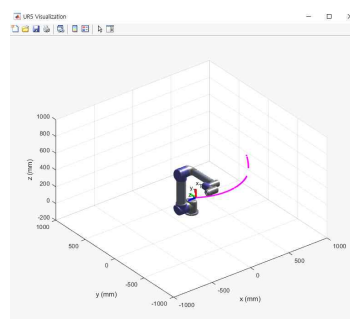
5. 창 왼편에 있는 ~~.m (확장자 .m) 파일을 더블클릭 후 편집기 17번째 줄에 `useHardware = false;` 인 것을 확인한다. 이후에 F5키를 눌러 파일을 실행시킨다.
(그림과 폴더, 파일명은 다를 수 있음)



6. 그림과 같은 창에서 실행할 로봇 모델을 선택하고 확인버튼을 누른다. 이후 잠시 기다리면 로봇이 원형을 그리는 시뮬레이션을 보여준다.



-->



예제 / MATLAB / 명령어 설명

[구동 시간을 알려주는 명령어]

`etime(t2,t1)`

(t2에 저장된 시각 - t1에 저장된 시각)을 알려주는 명령어다.
단위는 초이다.

[미지수 x에 대입하는 명령어]

`subs(eq1_1,sum_t1(i))`

eq1_1에 있는 미지수에 배열 sum_t1의 i번째 값을 대입한다.

[로봇의 Joint 값을 지정하는 명령어]

`simObj.Joints = [eq1(1,i) eq1(2,i) eq1(3,i) eq1(4,i) eq1(5,i) eq1(6,i)];`

a에 6칸 배열로 현재 로봇의 각 Joint값을 지정한다.

단위는 모두 라디안이다.

(시뮬레이션은 지정하는 Joint의 값과

시뮬레이션의 결과 Joint의 값이 같기 때문에 별도로 읽어오지 않는다.)

[실물로봇의 Joint 값을 지정하는 명령어]

`msg(hwObj,sprintf('(1,%f,%f,%f,%f,%f,%f)',q));`

실제 로봇이 Joint의 값으로 움직일 수 있도록 데이터를

보낼 때 맨처음에 '1'을 보내고, 뒤에 Joint값 6개를 보낸다.

[MATLAB 명령 창에 내용을 보여주는 명령어]

`fprintf('COMPLETE\n')`

MATLAB 명령 창에 작은따옴표 안에 있는 내용을 보여준다.

그리고 \n을 입력하면 줄바꿈(엔터)효과가 적용된다.

예제 / MATLAB / Robotiq 2-finger gripper 명령어 설명

[Robotiq 2-finger gripper의 tool 너비를 조절하는 명령어]

```
pause(1)
if useHardware
    msg(hwObj,sprintf('(5,%f,%f,%f,%f,%f,%f)',[0,0,0,0,0,0]));
end
pause(1)
```

예제 17번째줄을 true로 했다면 위 예시msg(~~)를 실행한다.
4는 최대로 tool을 벌린 상태, 4는 최소로 닫은 상태이다.

예제 / MATLAB / 예제에 사용된 변수명 설명

[각 Joint별 수식 입력]

eq1_1, eq1_2, ~~ , eq1_6

각 단계별로 수식을 구분하기 위해 예제와 같이

eq 바로 뒤에있는 숫자를 수정할 수 있도록 한다.

(수정하며 그 아래에 있는 simObj.Joints(eqx(~~),eqx(~~)~)에
있는 숫자, joint_velx도 필히 바꾸어야 한다)

언더바_ 뒤에 있는 숫자는 1번부터 베이스, 2번은 솔더 ~~
6번은 손목3을 의미한다.

(그러므로 언더바 뒤에있는 숫자는 변경하지 말 것)

[시간 변수명]

t1 로봇이 홈포지션에 도달한 시각을 저장한다

t2=clock 현재시각을 t2에 계속하여 갱신해준다.

루프를 돌며 t2-t1(처음 시각-현재시각)을 계산하여
구동 시간을 알려준다.

이렇게 값을 갱신해주면 처음 시작으로부터 현재까지의
시간을 알 수 있다.

예제 / MATLAB / 예제에 표시된 작업공간(변수값) 설명

[각 단계별 계산된 Joint값]

eq1, eq2

각 단계별로 작성한 수식에서 나온 Joint값들이 기재되었다.
생각한 경로와 구동되는 경로가 다를 경우,
해당 값들을 살펴보고 제대로 수식이 계산되었는지 확인한다.

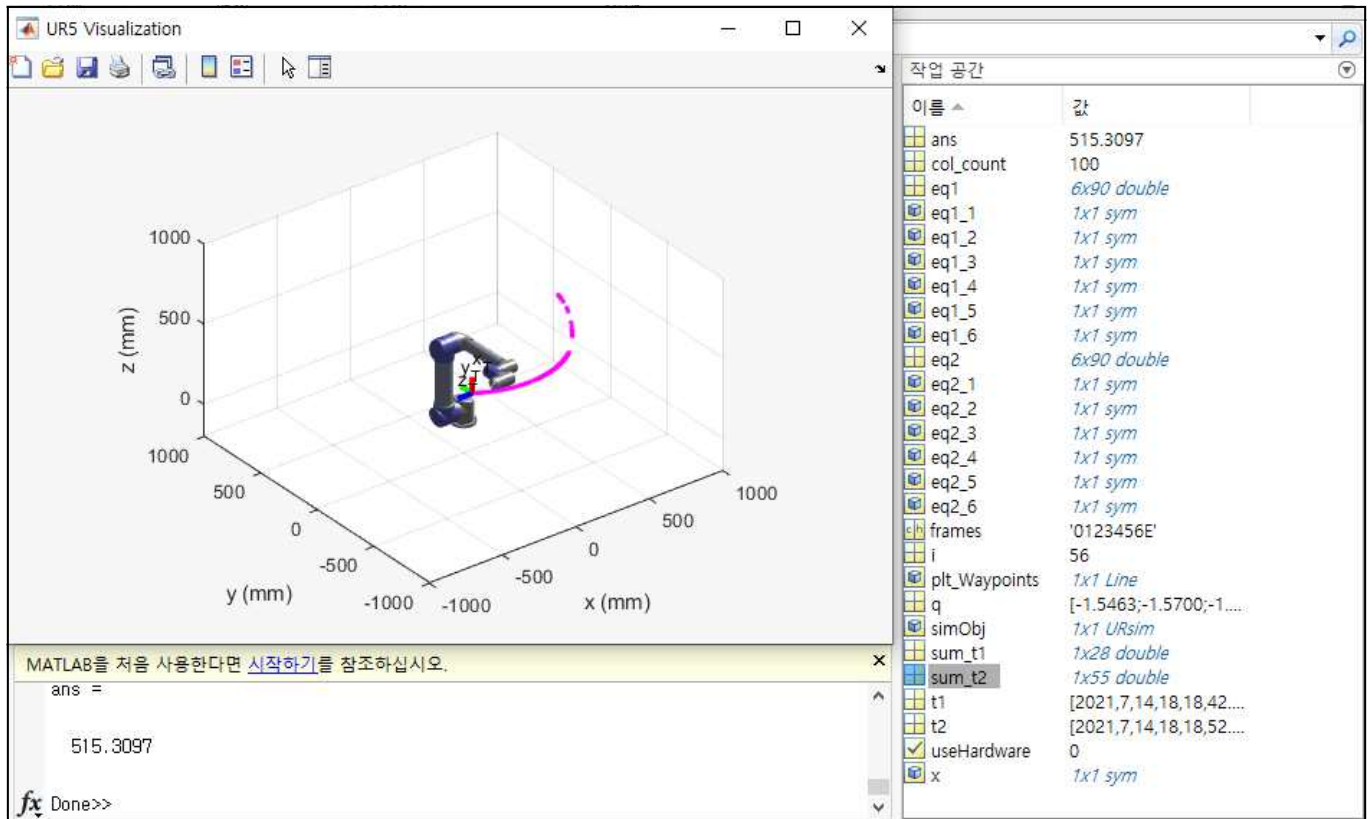
[실물로봇의 Joint 값]

joint_vel1, joint_vel2

실제 로봇이 알려주는 Joint의 값이다.
속도와 시간을 고려하지 않았다면,
생각한 값과 크게 다를 수 있다.

예제 / MATLAB / 예제 결과 (잘 된 경우)

MATLAB 명령 창에 Done이라는 글과 함께 왼쪽에 fx가 보이며 오른쪽 작업공간과 같이 데이터들이 보이게 된다.



예제 / MATLAB / 예제 결과 (오류 현상)

[시뮬레이션을 돌릴 때 `useHardware = true`로 한 경우]

`hwObj = UR;`이라는 오류가 발생하게 되며, 이는 예제 17번째줄에 `useHardware = true`를 `useHardware = false;`로 고치면 된다.

