

# Install from Anaconda to OpenAI Gym (Window Ver.)

이동민

삼성전자 서울대 공동연구소  
Jul 15, 2019

# Outline

---

- Anaconda
- Pytorch
- TensorboardX
- OpenAI Gym
- VSCode

# Anaconda

---



- 다양한 수학/과학, 시각화 라이브러리부터 머신러닝 라이브러리까지 쉽게 설치 가능하도록 지원하는 플랫폼
- 하나의 독립적인 환경을 만들어 라이브러리, 의존성 등을 쉽게 관리할 수 있음
- 윈도우, 맥, 리눅스 환경에서 모두 사용 가능
- <https://www.anaconda.com>

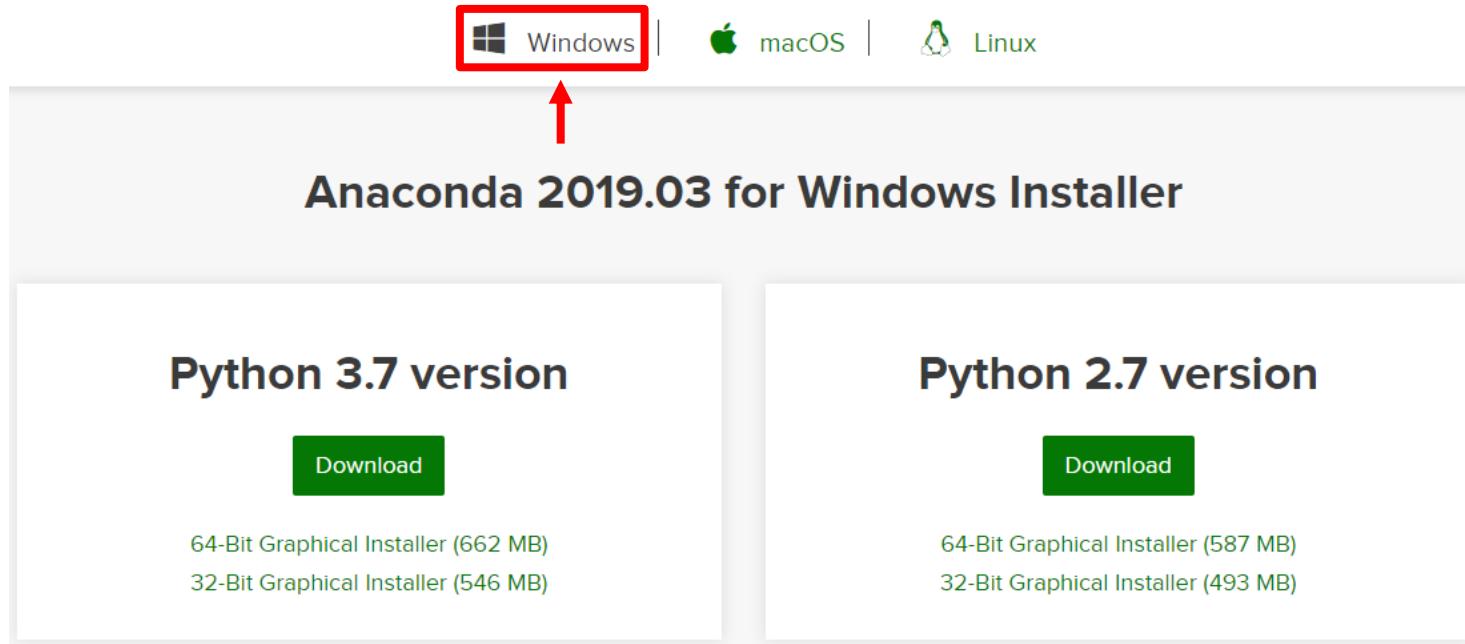
# Anaconda

- Anaconda 다운로드 (운영체제 및 CPU에 맞춰서 다운로드)
  - <https://www.anaconda.com>

The screenshot shows the Anaconda website homepage. At the top, there is a navigation bar with links for Products, Why Anaconda?, Solutions, Resources, Company, a prominent green 'Download' button, and a search icon. The main header features the Anaconda logo and the text '2019 STATE OF DATA SCIENCE REPORT'. Below the header, a paragraph describes a survey of the Anaconda community. A 'Download the Report' button is located below the text. At the bottom of the page, a green banner displays the news: 'Latest news: Anaconda named a May 2019 Gartner Peer Insights Customers' Choice for Data Science and Machine Learning Platforms.'

# Anaconda

- Anaconda 다운로드 (운영체제 및 CPU에 맞춰서 다운로드)



# Anaconda

- Anaconda 다운로드 (운영체제 및 CPU에 맞춰서 다운로드)

 Windows |  macOS |  Linux

### Anaconda 2019.03 for Windows Installer

**Python 3.7 version**

 Download ←

64-Bit Graphical Installer (662 MB)  
32-Bit Graphical Installer (546 MB)

**Python 2.7 version**

 Download

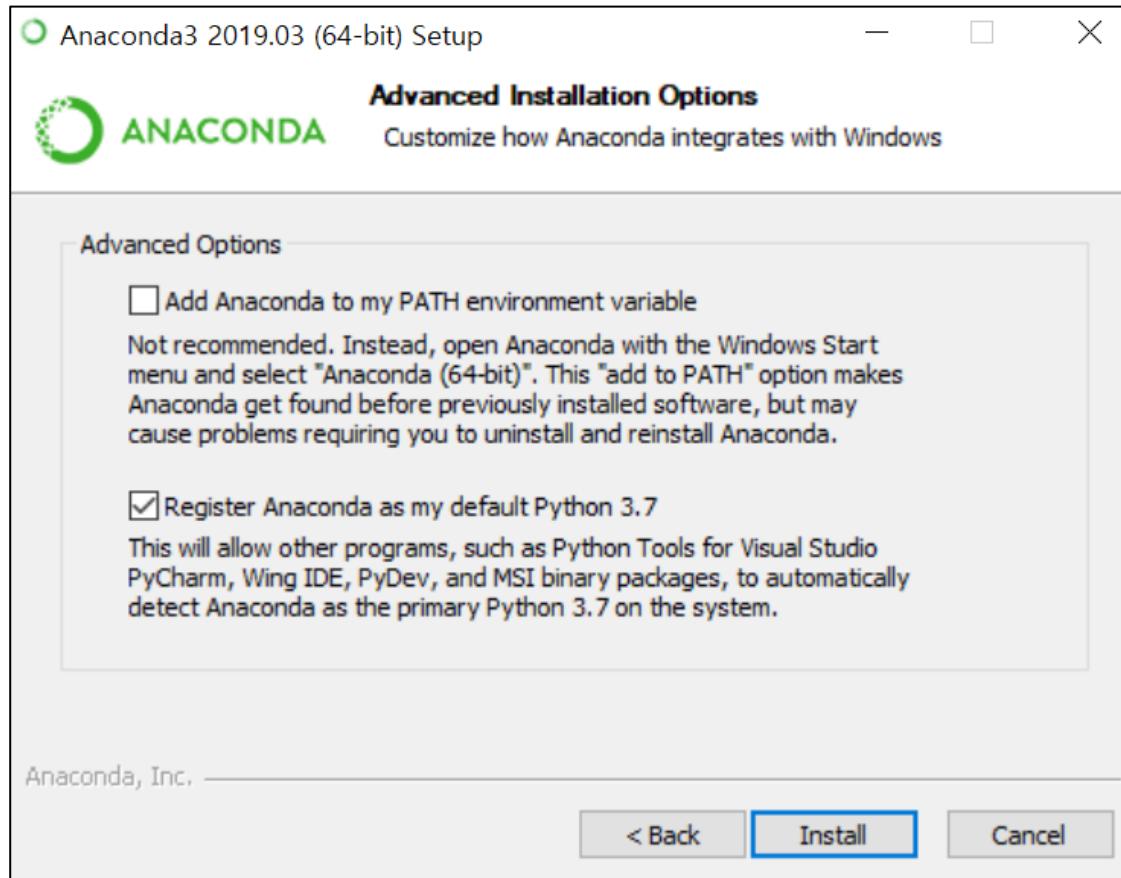
64-Bit Graphical Installer (587 MB)  
32-Bit Graphical Installer (493 MB)



**CORE**  
Control + Optimization Research Lab

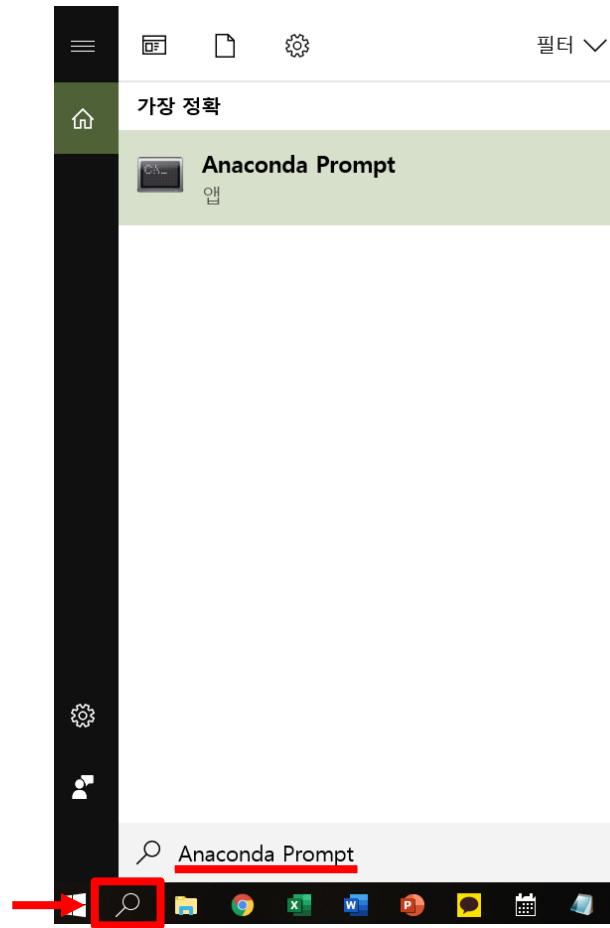
# Anaconda

- 다운로드에서 설치파일을 클릭하여 다운로드 받기



# Anaconda

- Window 검색에서 Anaconda Prompt 실행



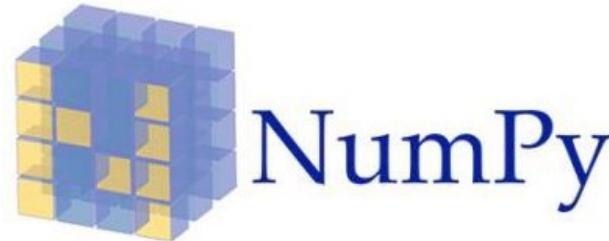
# Anaconda

- Anaconda version 확인
    - conda --version
  - 새로운 환경 생성
    - conda create -n env\_name python=3.6
    - y/n 나오면 y
  - 가상환경 목록보기
    - conda env list
  - 가상환경 활성화 / 비활성화 하기
    - 활성화 - conda activate env\_name
    - 비활성화 - conda deactivate
- (base) C:\Users\Lee2>conda activate rl-env  
(rl-env) C:\Users\Lee2>conda deactivate



**CORE**  
Control + Optimization Research Lab

# Numpy



NumPy

- Numerical Python
- 파이썬의 고성능 과학 계산용 패키지
- Vector와 Matrix와 같은 Array 연산의 표준
- Anaconda를 설치하면 자동적으로 설치되어 있음



**CORE**  
Control + Optimization Research Lab

# Pytorch

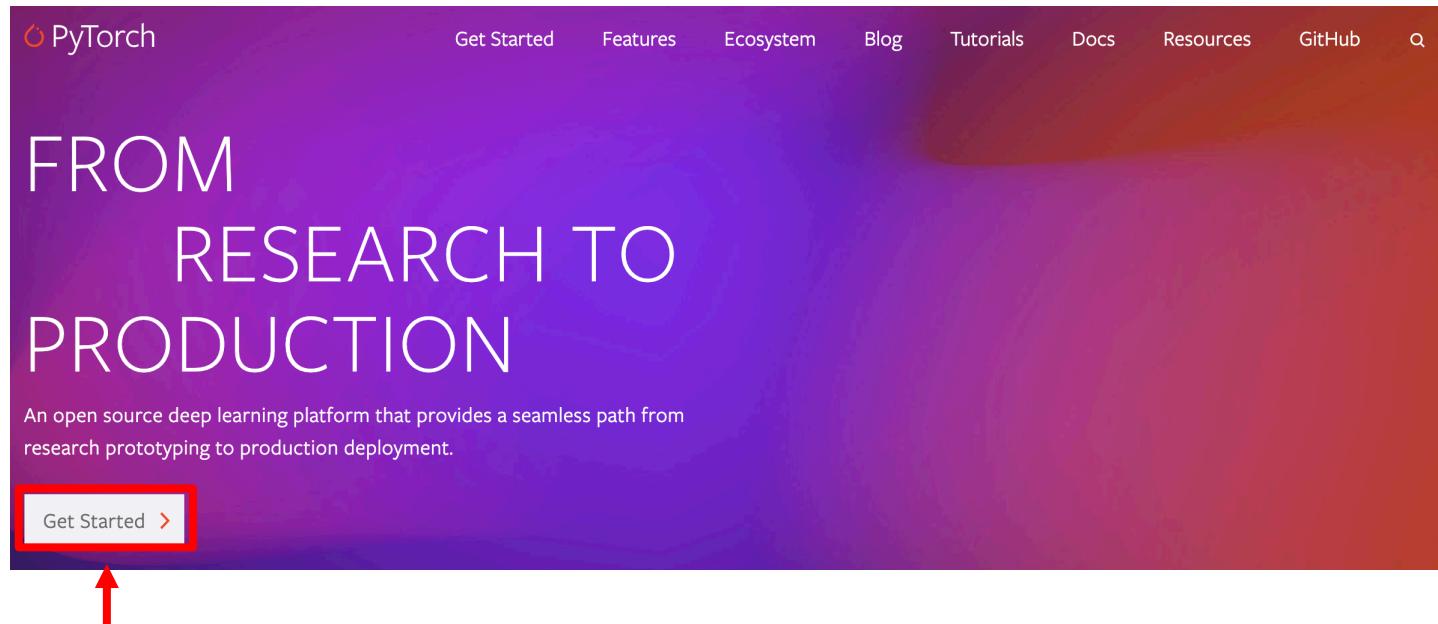
---



- Numpy를 대체하고 GPU를 쉽게 사용할 수 있음
- 최고의 유연성과 속도를 제공하는 딥러닝 연구 플랫폼
- 이해와 디버깅이 쉽고 직관적이며, 간결한 코드(Pythonic code)로 구성
- Define-by-Run 방식을 기반으로 한 실시간 결과값을 시각화
- 파이썬 라이브러리(Numpy, Matplotlib 등)와 높은 호환성을 가짐

# Pytorch

- Pytorch 설치
  - <https://pytorch.org>



# Pytorch

- Pytorch 설치
  - <https://pytorch.org>

START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch 1.1. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.1 builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also [install previous versions of PyTorch](#). Note that LibTorch is only available for C++.

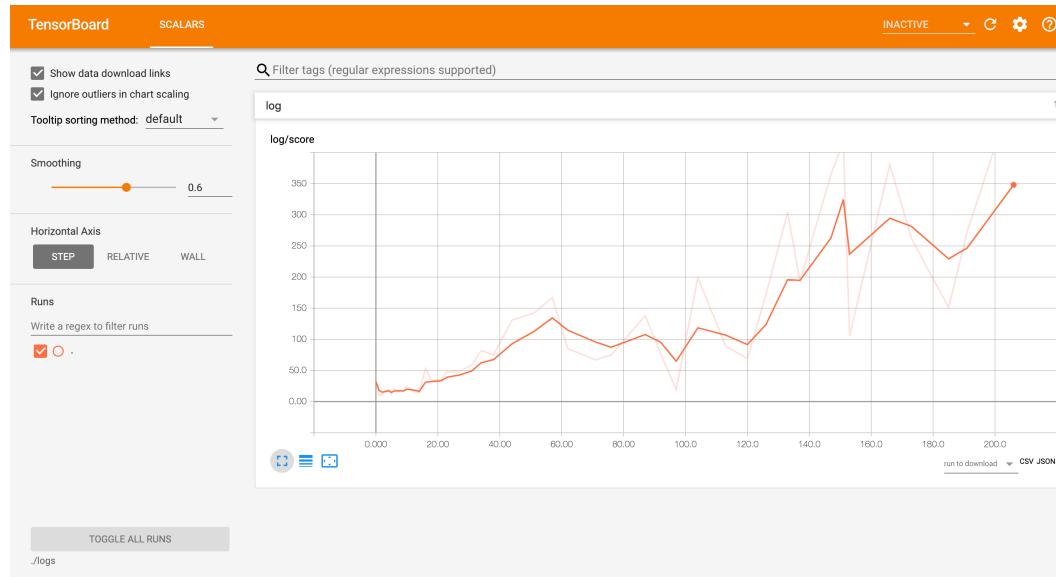
PyTorch Build	Stable (1.1)		Preview (Nightly)	
Your OS	Linux		Mac	Windows
Package	Conda	Pip	LibTorch	Source
Language	Python 2.7	Python 3.5	Python 3.6	Python 3.7 C++
CUDA	9.0	10.0	None	
Run this Command:	<code>conda install pytorch-cpu torchvision-cpu -c pytorch</code>			

# Pytorch

---

- Pytorch 설치
  - conda install pytorch-cpu torchvision-cpu -c pytorch

# TensorboardX



- Tensorflow의 Tensorboard처럼 Tensorflow 없이도 tensor들의 flow를 시각적으로 볼 수 있도록 만든 라이브러리 → 주로 Pytorch에서 많이 사용
- 강화학습 알고리즘 구현 시, episode마다 sum of reward가 얼마나 나오는지 보기 쉽게 볼 수 있음
- <https://tensorboardx.readthedocs.io/en/latest/tutorial.html>

# TensorboardX

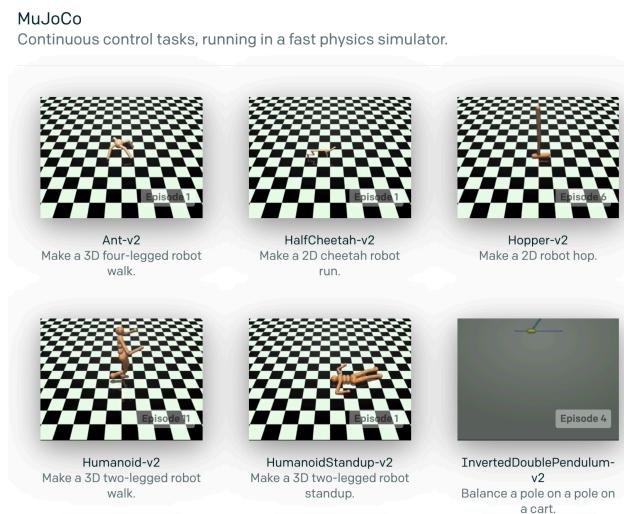
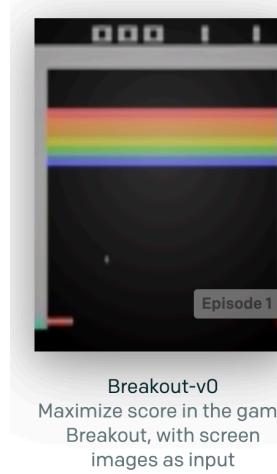
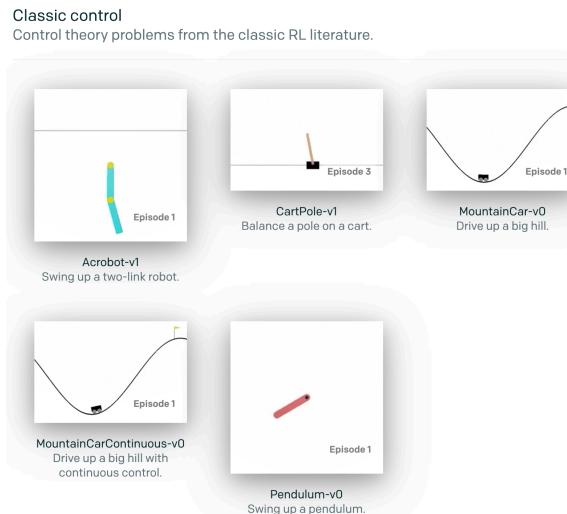
---

- Tensorflow + TensorboardX 설치
  - pip install tensorflow
  - pip install tensorboardX

# OpenAI Gym



- OpenAI에서 간단한 게임들을 통해 강화학습을 테스트 할 수 있는 Gym이라는 환경을 제공
- Tensorflow나 Pytorch와 같은 수치 계산 라이브러리와도 호환됨



**CORE**  
Control + Optimization Research Lab

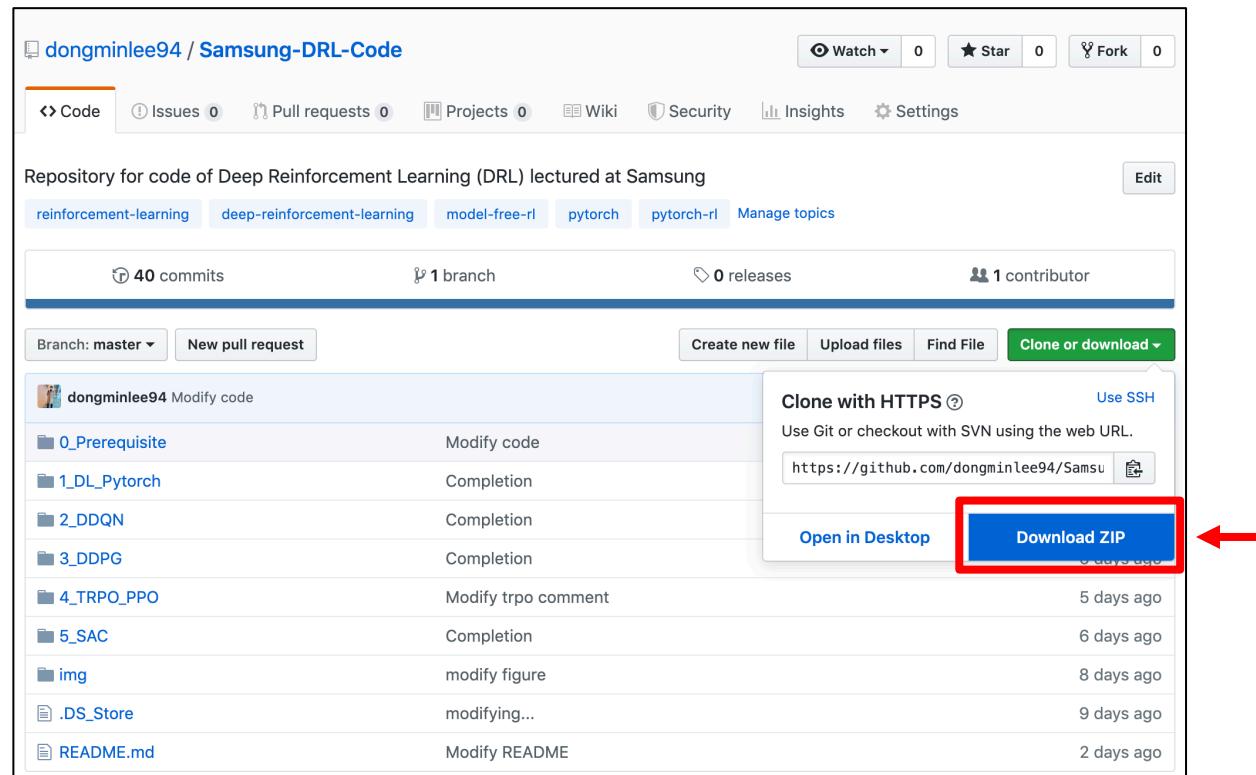
# OpenAI Gym

---

- Gym 설치
  - pip install gym

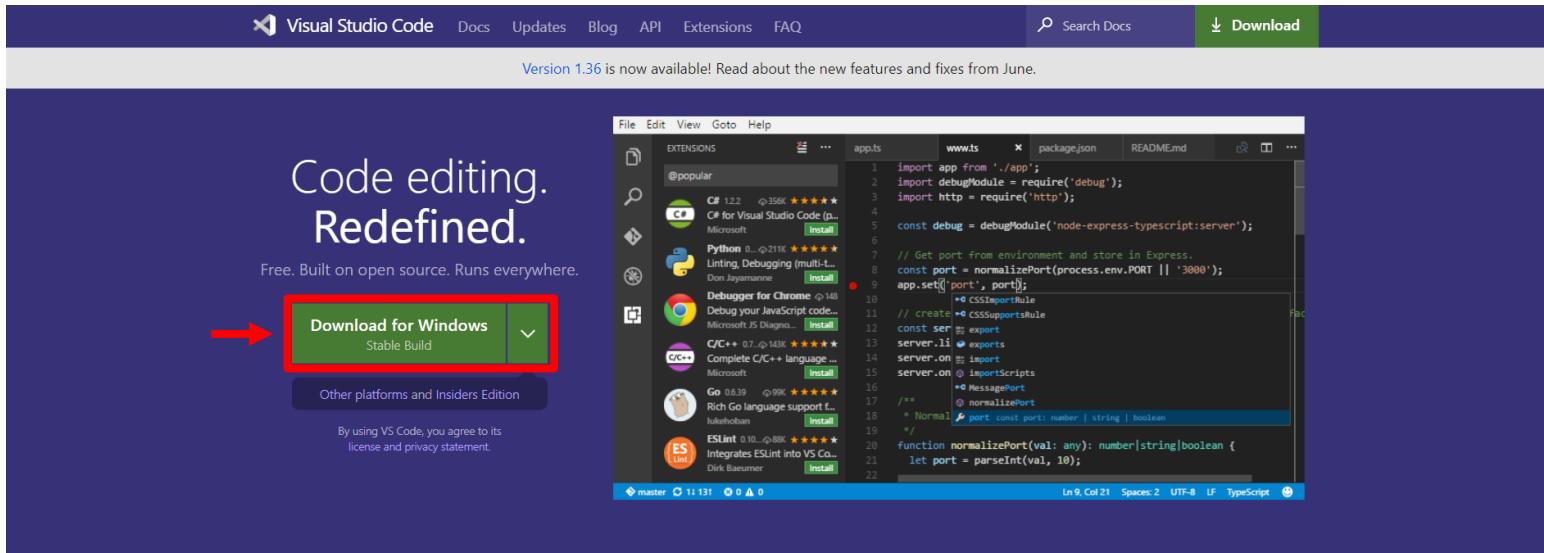
# DRL Code

- Code 다운로드
  - <https://github.com/dongminlee94/Samsung-DRL-Code>
  - 알집파일 압축 풀기



# VSCode

- VSCode 설치
  - <https://code.visualstudio.com>



IntelliSense



Debugging



Built-in Git



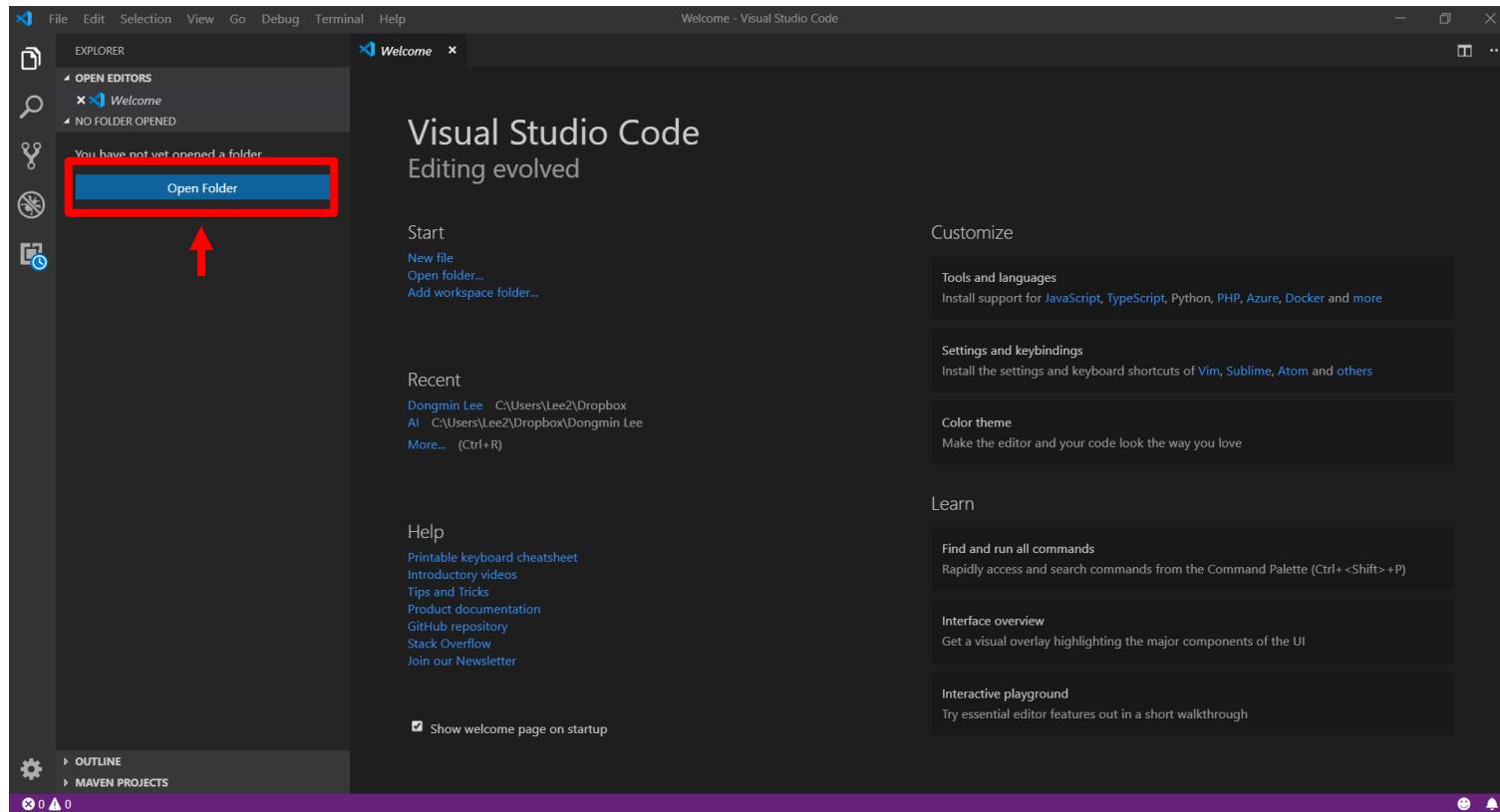
Extensions



**CORE**  
Control + Optimization Research Lab

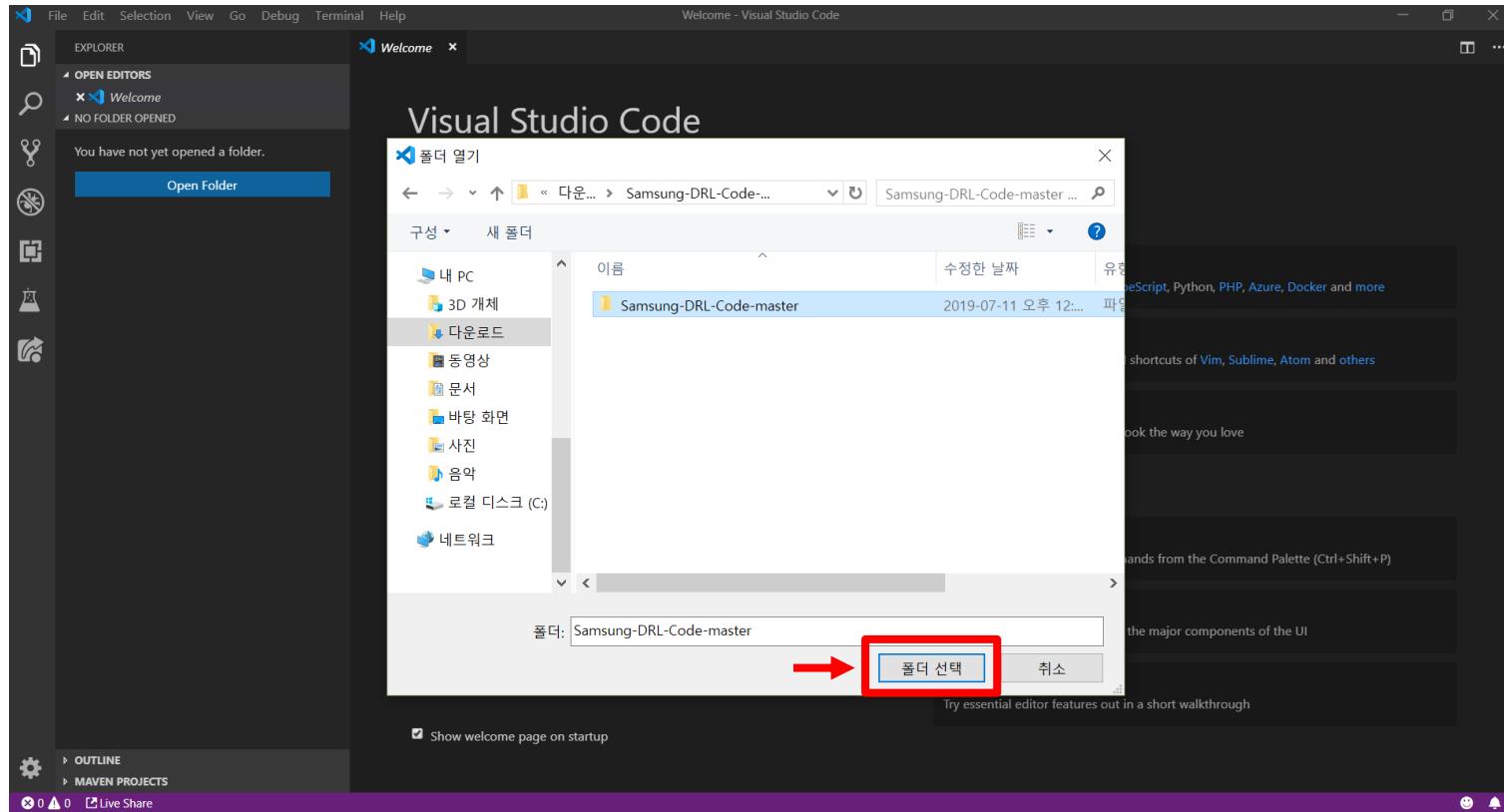
# VSCode

- VSCode에서 DRL code 불러오기



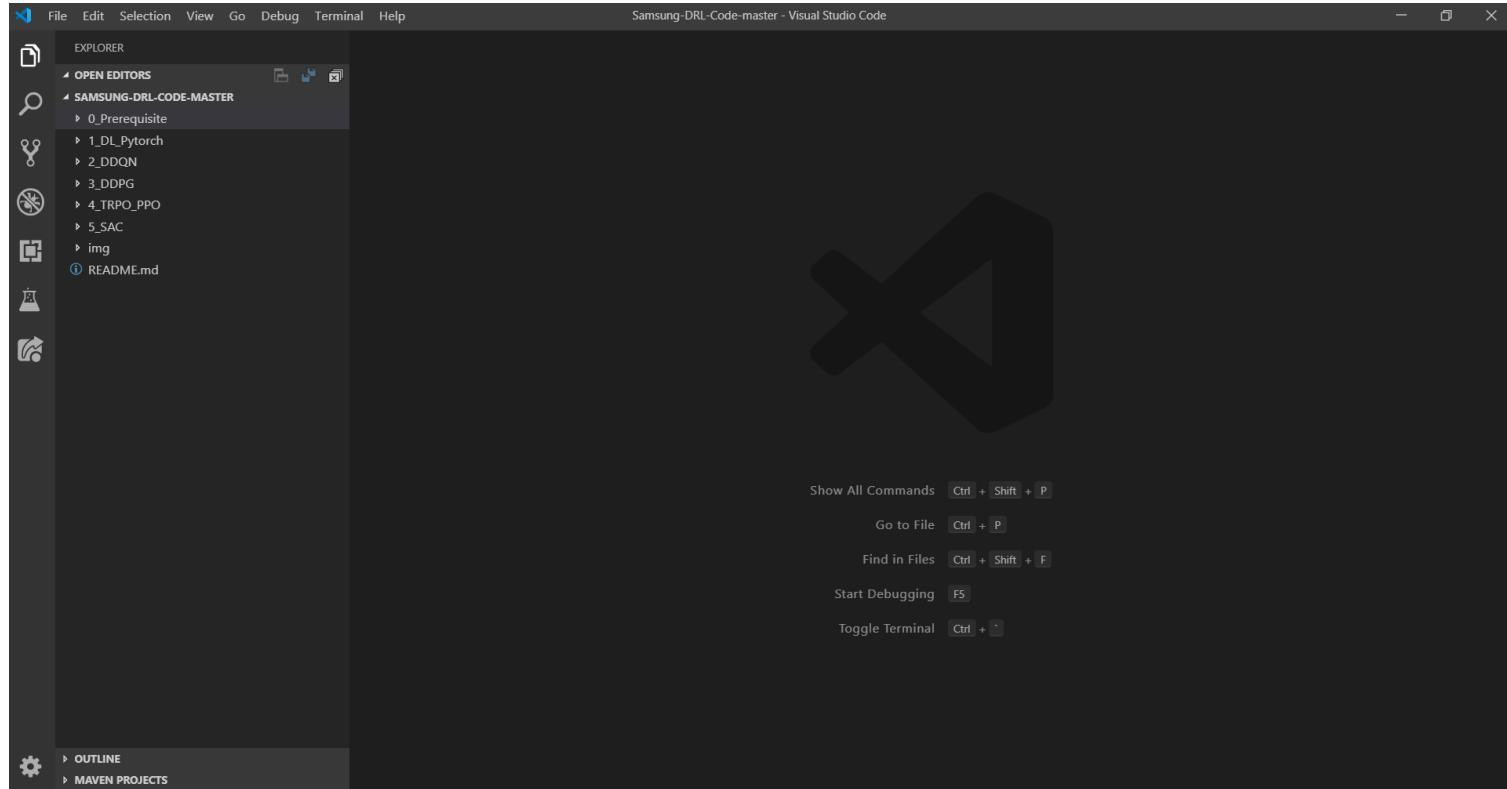
# VSCode

- VSCode에서 DRL code 불러오기



# VSCode

- VSCode에서 DRL code 불러오기



# VSCode

- DRL code → 0\_Prerequisite → 01\_Install → test.py 불러오기

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure:
  - OPEN EDITORS:** test.py
  - SAMSUNG-DRL-CODE-MASTER:** 0\_Prerequisite, 01\_Install, logs, Install.pdf, test.py
  - Others:** 02\_Numpy, 1\_DL\_Pytorch, 2\_DDQN, 3\_DDPG, 4\_TRPO\_PPO, 5\_SAC, img, README.mdA red box highlights the 01\_Install folder and its contents.
- Code Editor (Right):** The test.py file is open, displaying Python code to check imports, tensorboardX, gym, and environment steps.

```
test.py - Samsung-DRL-Code-master - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
test.py - Samsung-DRL-Code-master - Visual Studio Code
EXPLORER
OPEN EDITORS
x test.py 0_Prerequisite\01_Install
SAMSUNG-DRL-CODE-MASTER
0_Prerequisite
01_Install
logs
Install.pdf
test.py
02_Numpy
1_DL_Pytorch
2_DDQN
3_DDPG
4_TRPO_PPO
5_SAC
img
README.md
test.py
#####
1. Check import numpy, torch #####
#####
import numpy
import torch
print('numpy' + numpy.__version__)
print('torch' + torch.__version__)
#####
2. Check tensorboardX #####
#####
from tensorboardX import SummaryWriter
writer = SummaryWriter('./logs')
#####
3. Check gym #####
#####
# import gym
# env = gym.make('CartPole-v1')
#
# for episode in range(1000):
#     done = False
#     state = env.reset()
#     while not done:
#         env.render()
#         action = env.action_space.sample()
#         next_state, reward, done, _ = env.step(action)
# 
```

# VSCode

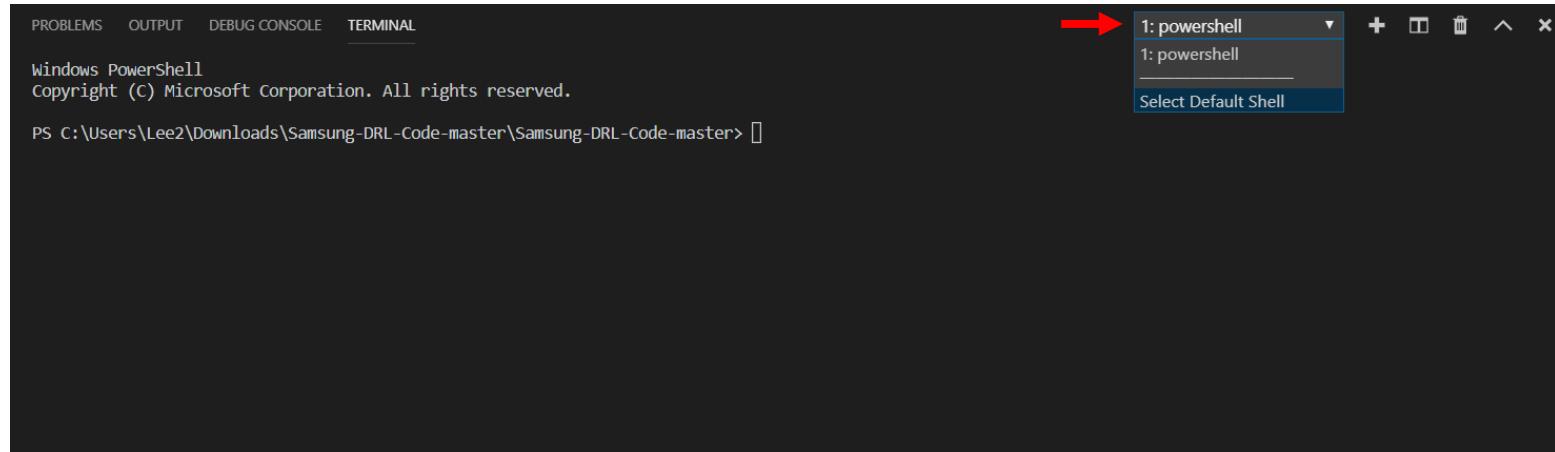
- Terminal 불러오기
  - Ctrl + ~ 로 터미널창 띄우기

A screenshot of the Visual Studio Code interface. The title bar reads "test.py - Samsung-DRL-Code-master - Visual Studio Code". The left sidebar shows the "EXPLORER" view with a tree structure of files and folders, including "test.py" which is currently selected. The main editor area displays Python code for testing numpy and torch versions. Below the editor are tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL". The "TERMINAL" tab is active, showing a Windows PowerShell window with the command "PS C:\Users\Lee2\Downloads\Samsung-DRL-Code-master\Samsung-DRL-Code-master>". The status bar at the bottom shows "Python 3.6.1 64-bit", "Ln 26, Col 13", "Spaces: 4", "UTF-8", "LF", "Python", and a bell icon.

```
1 ##### 1. Check import numpy, torch #####
2 #####
3 #####
4
5 import numpy
6 import torch
7
8 print('numpy' + numpy.__version__)
9 print('torch' + torch.__version__)
10 #####
11 #####
12 #####
13 #####
14 ##### 2. Check tensorboardX #####
15 #####
16
17 from tensorboardX import SummaryWriter
18 writer = SummaryWriter('./logs')
19 #####
20 #####
```

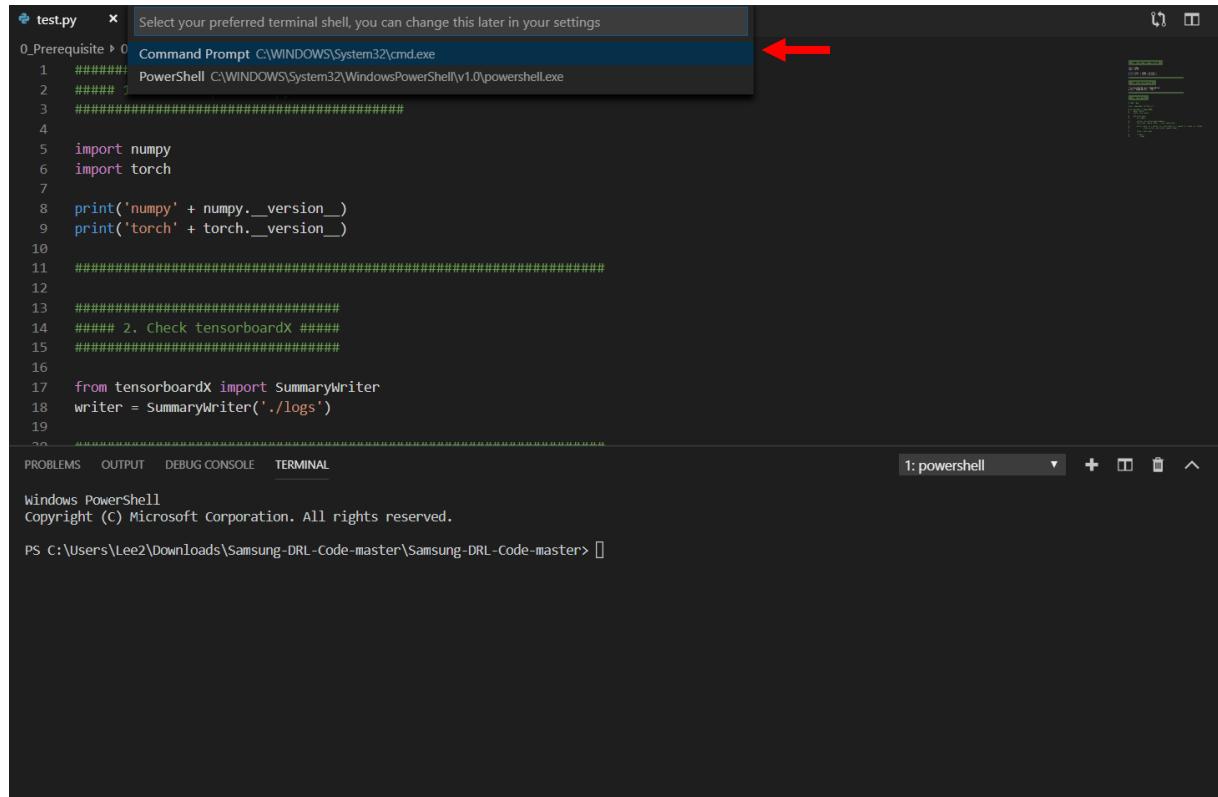
# VSCode

- Terminal 불러오기
  - Ctrl + ~ 로 터미널창 띄우기
  - PowerShell에서 Command Prompt로 바꾸기



# VSCode

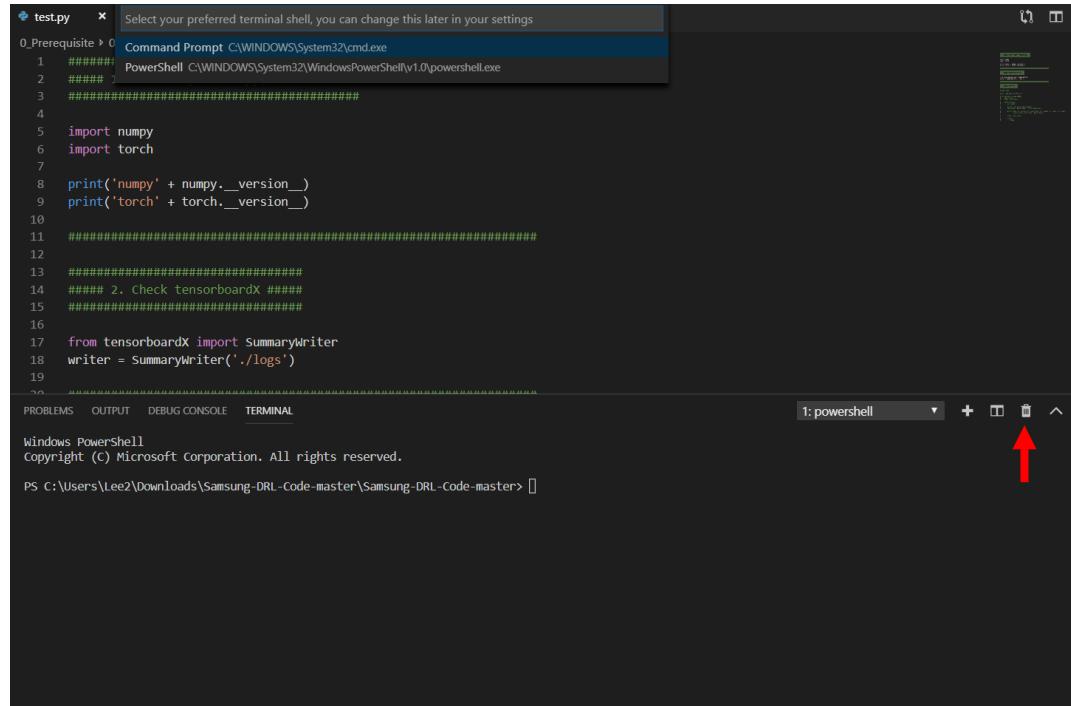
- Terminal 불러오기
  - Ctrl + ~ 로 터미널창 띄우기
  - PowerShell에서 Command Prompt로 바꾸기



**CORE**  
Control + Optimization Research Lab

# VSCode

- Terminal 불러오기
  - Ctrl + ~ 로 터미널창 띄우기
  - PowerShell에서 Command Prompt로 바꾸기
  - PowerShell 종료하고 다시 Ctrl + ~로 터미널창 띄우기



# VSCode

- Terminal 불러오기 (Terminal setting 완료)

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "SAMSUNG-DRL-CODE-MASTER". The "test.py" file is selected.
- Code Editor:** Displays the content of "test.py". The code prints the versions of numpy and torch.
- Terminal:** Shows the command line output:

```
Microsoft Windows [Version 10.0.17134.829]
(c) 2018 Microsoft corporation. All rights reserved.

C:\Users\Lee2\Downloads\Samsung-DRL-Code-master\Samsung-DRL-Code-master>C:/Users/Lee2/Anaconda3/Scripts/activate
(base) C:\Users\Lee2\Downloads\Samsung-DRL-Code-master\Samsung-DRL-Code-master>conda activate base
(base) C:\Users\Lee2\Downloads\Samsung-DRL-Code-master\Samsung-DRL-Code-master>]
```
- Status Bar:** Shows "Python 3.7.3 64-bit ('base': conda)" and other status indicators.

# VSCode

- Numpy, Pytorch, TensorboardX, OpenAI Gym이 제대로 설치 되었는지 확인
  - conda activate env\_name
  - cd 0\_Prerequisite
  - cd 01\_Install
  - python test.py

```
import gym

env = gym.make('CartPole-v1')

for episode in range(10000):
    done = False
    state = env.reset()

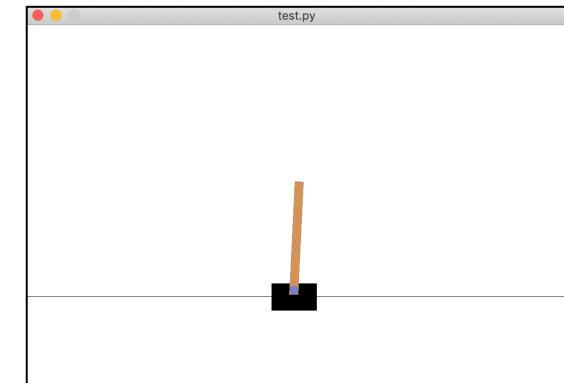
    while not done:
        env.render()

        action = env.action_space.sample()
        next_state, reward, done, _ = env.step(action)

        print('state: {} | action: {} | next_state: {} | reward: {} | done: {}'.format(
            state, action, next_state, reward, done))

        state = next_state

    if done:
        break
```



# Thank you



**CORE**  
Control + Optimization Research Lab