LiberVance

목포대학교 RL 강의 (23.11)

Development Env. Setup Guide

(부제 : RL 실습 환경 셋업 가이드)

JuSung Kang, LiberVance Co. Ltd.,

jskang.libervance@gmail.com,

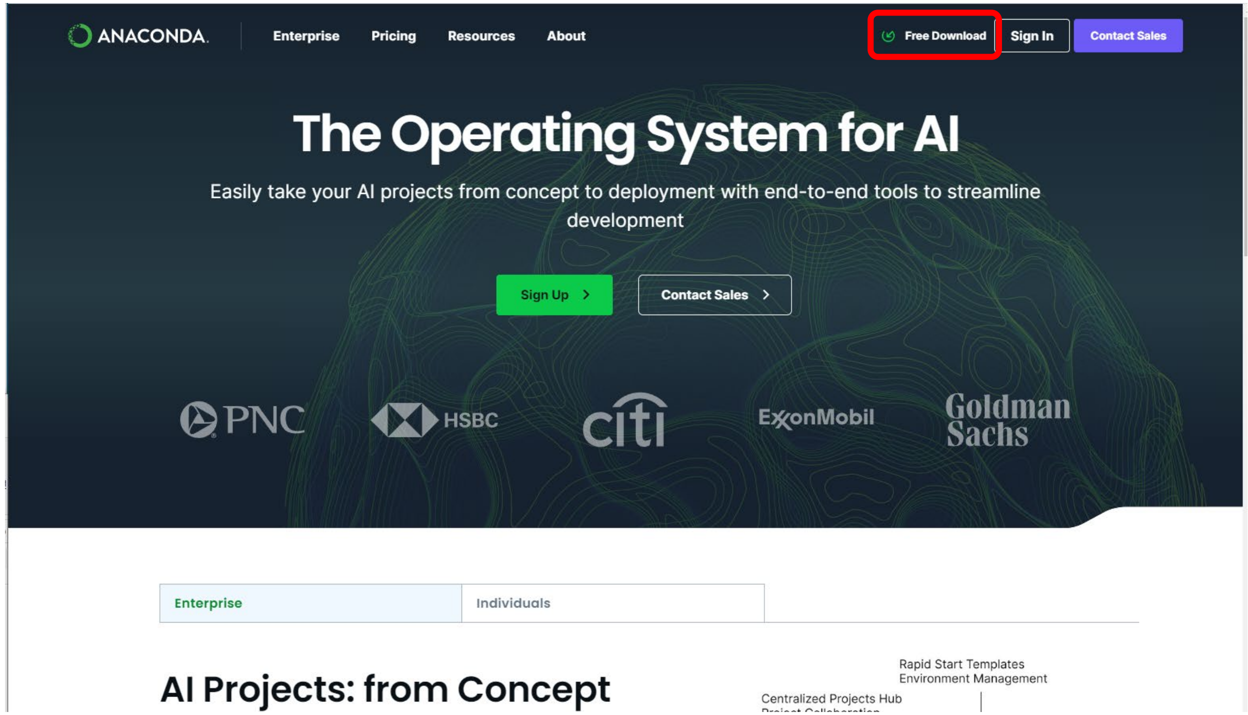
Development Env. Setup Guide

- **Requirements**
  - **OS**
    - Windows 10/11
  - **Packages**
    - Anaconda3 for Python 3.x
    - Pytorch
    - Pycharm
    - OpenAI Gymnasium

[출처] Jongseong Jang, Super Mario Bros. Leaned by Deep Q Network, https://youtu.be/YUYnGA0Gqv4

Development Env. Setup Guide – Anaconda3

- Anaconda3 (Download)
  - https://www.anaconda.com/
  - Free Download >> Download >> (files) Anaconda3-2023.09-0-Windows-x86_64.exe

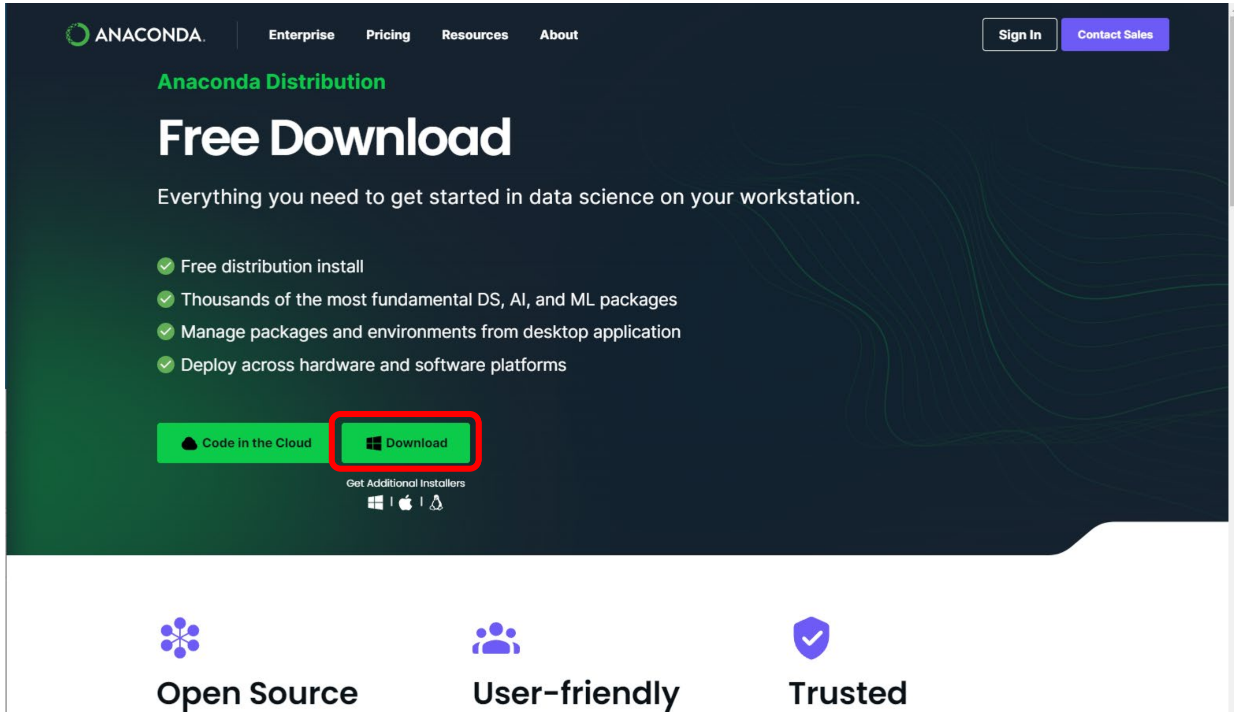


The Operating System for AI

Easily take your AI projects from concept to deployment with end-to-end tools to streamline development

AI Projects: from Concept

Rapid Start Templates
Environment Management
Centralized Projects Hub



Anaconda Distribution

Free Download

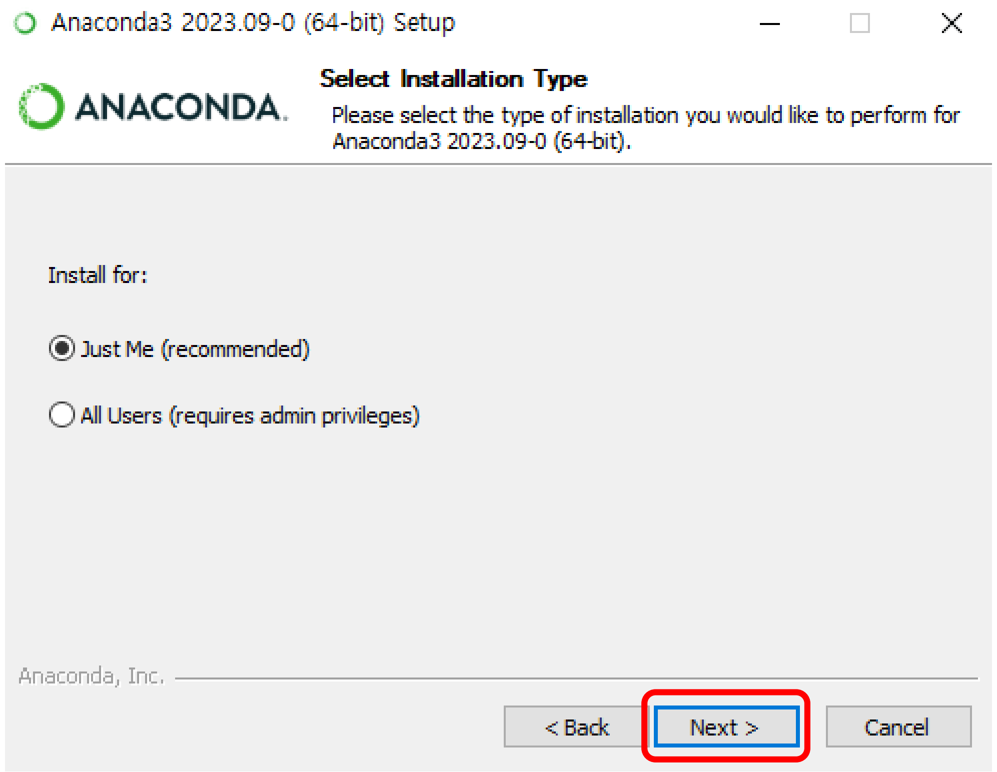
Everything you need to get started in data science on your workstation.

- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

Open Source   User-friendly   Trusted

Development Env. Setup Guide – Anaconda3

- **Anaconda3 (Install)**
  - **Next >> I Agree >> Next >> Next >> Install >> Next >> Finish**



Anaconda3 2023.09-0 (64-bit) Setup

Select Installation Type

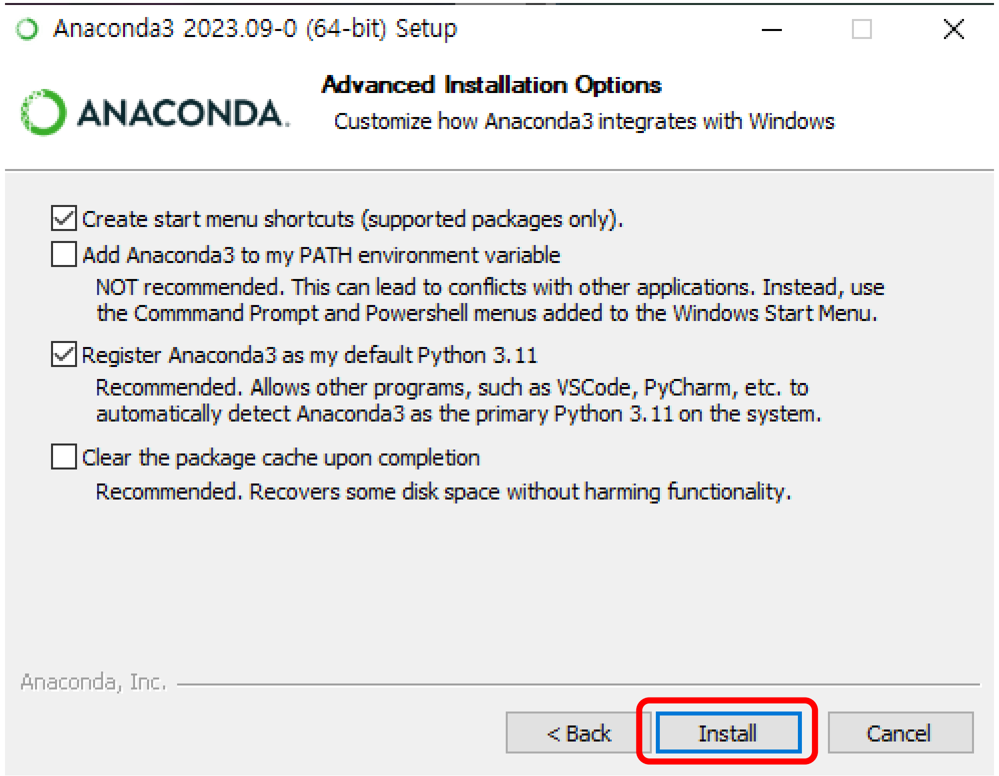
Please select the type of installation you would like to perform for Anaconda3 2023.09-0 (64-bit).

Install for:

● Just Me (recommended)

○ All Users (requires admin privileges)

Anaconda, Inc.



Anaconda3 2023.09-0 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda3 integrates with Windows

☑ Create start menu shortcuts (supported packages only).

☐ Add Anaconda3 to my PATH environment variable

NOT recommended. This can lead to conflicts with other applications. Instead, use the Commmand Prompt and Powershell menus added to the Windows Start Menu.

☑ Register Anaconda3 as my default Python 3.11

Recommended. Allows other programs, such as VSCode, PyCharm, etc. to automatically detect Anaconda3 as the primary Python 3.11 on the system.

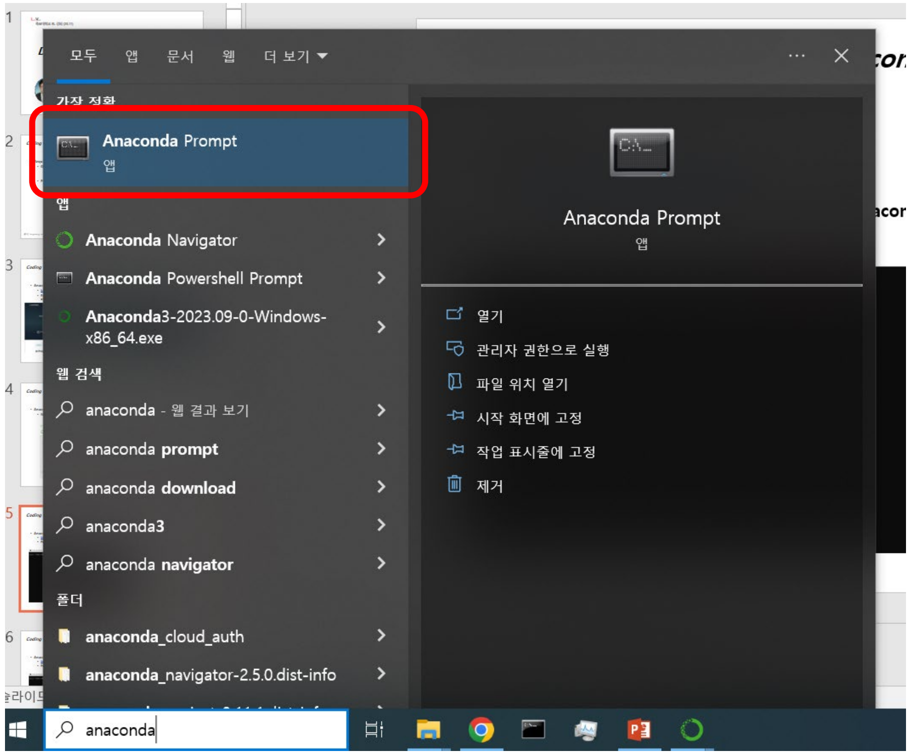
☐ Clear the package cache upon completion

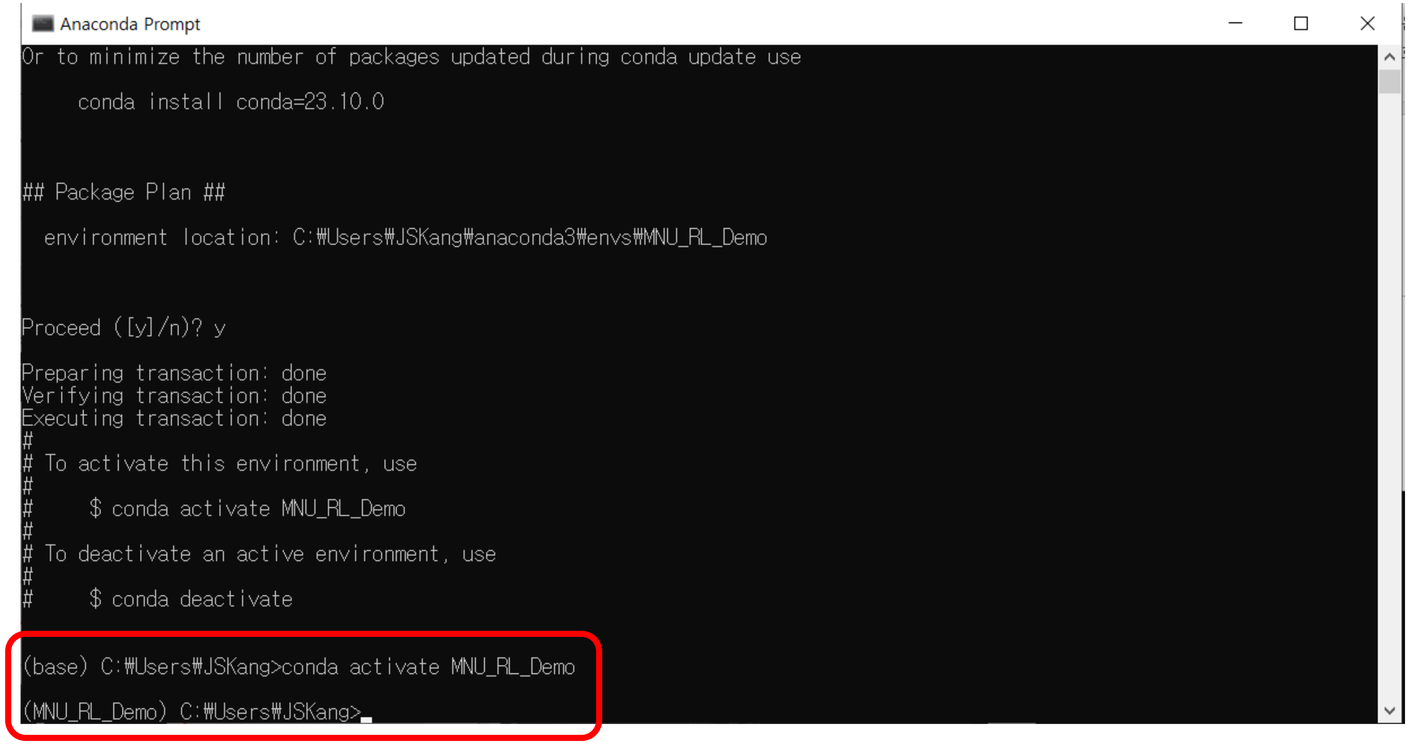
Recommended. Recovers some disk space without harming functionality.

Anaconda, Inc.

Development Env. Setup Guide – Anaconda3

- Anaconda3
  - 검색창 >> Anaconda Prompt 실행
  - >(base) conda create –n MNU_RL_Demo
  - >(base) conda activate MNU_RL_Demo

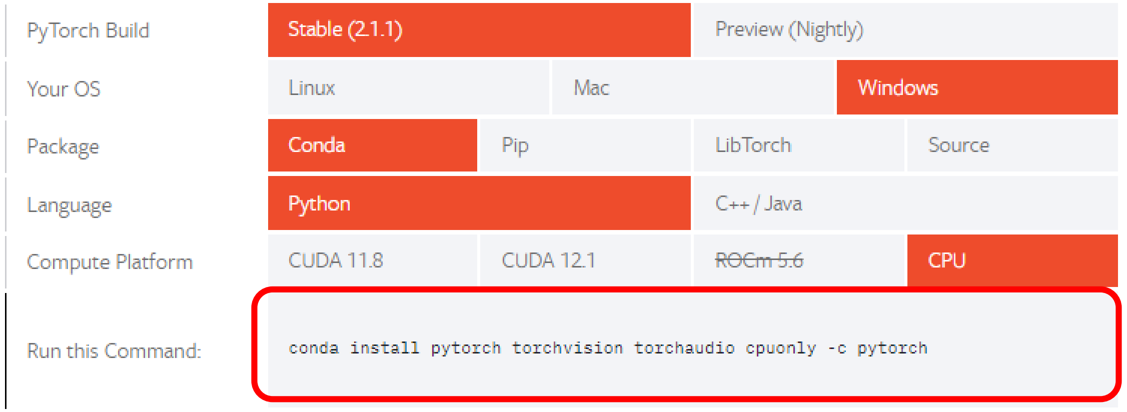




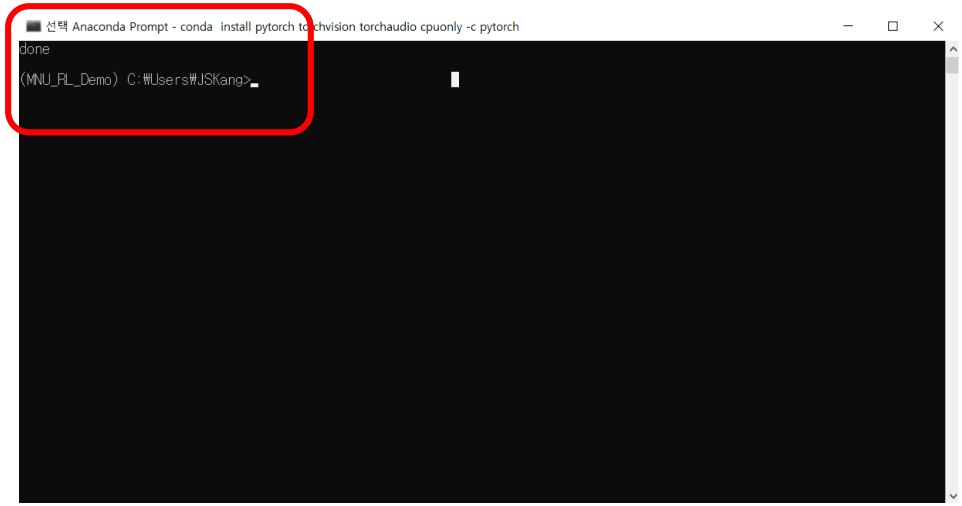
```
Or to minimize the number of packages updated during conda update use

     conda install conda=23.10.0



## Package Plan ##

  environment location: C:₩Users₩JSKang₩anaconda3₩envs₩MNU_RL_Demo



Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate MNU_RL_Demo
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:₩Users₩JSKang>conda activate MNU_RL_Demo

(MNU_RL_Demo) C:₩Users₩JSKang>
```

Development Env. Setup Guide – Pytorch

- **Pytorch**
  - **https://pytorch.org/get-started/locally/**
  - **> (MNU_RL_Demo) conda install pytorch torchvision torchaudio cpuonly -c pytorch**
    - * GPU computing - H/W 에 맞는 Cuda, cuDNN 설치 필요
      - * https://wannabenice.tistory.com/54



PyTorch Build	Stable (2.1.1)	Preview (Nightly)		
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python	C++ / Java		
Compute Platform	CUDA 11.8	CUDA 12.1	~~ROCm 5.6~~	CPU
Run this Command:	conda install pytorch torchvision torchaudio cpuonly -c pytorch			


선택 Anaconda Prompt - conda install pytorch to chvision torchaudio cpuonly -c pytorch

```
done

(MNU_RL_Demo) C:₩Users₩JSKang>_
```

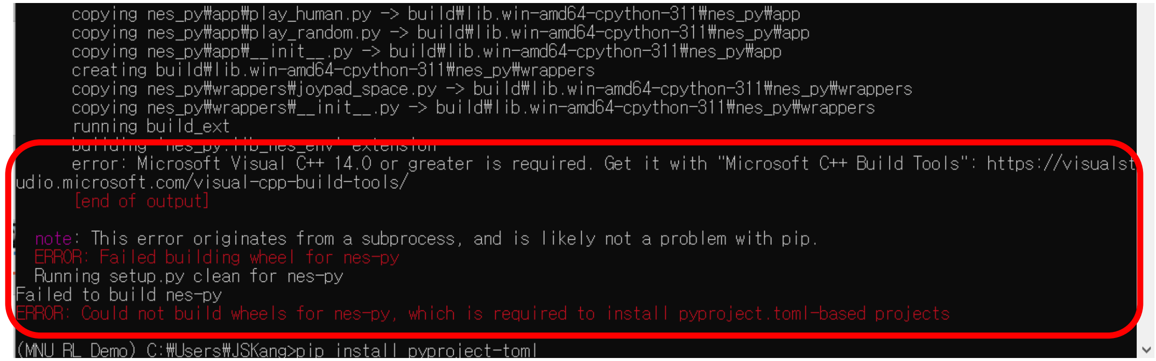
Development Env. Setup Guide – OpenAI Gym, Matplotlib

- **OpenAI Gym**
  - **> (MNU_RL_Demo) pip install nes-py**
    - If error message occurs, see the next page
      - “~ pyproject.toml-based projects”
  - **> (MNU_RL_Demo) pip install gymnasium[classic-control] gym-super-mario-bros==7.4.0**

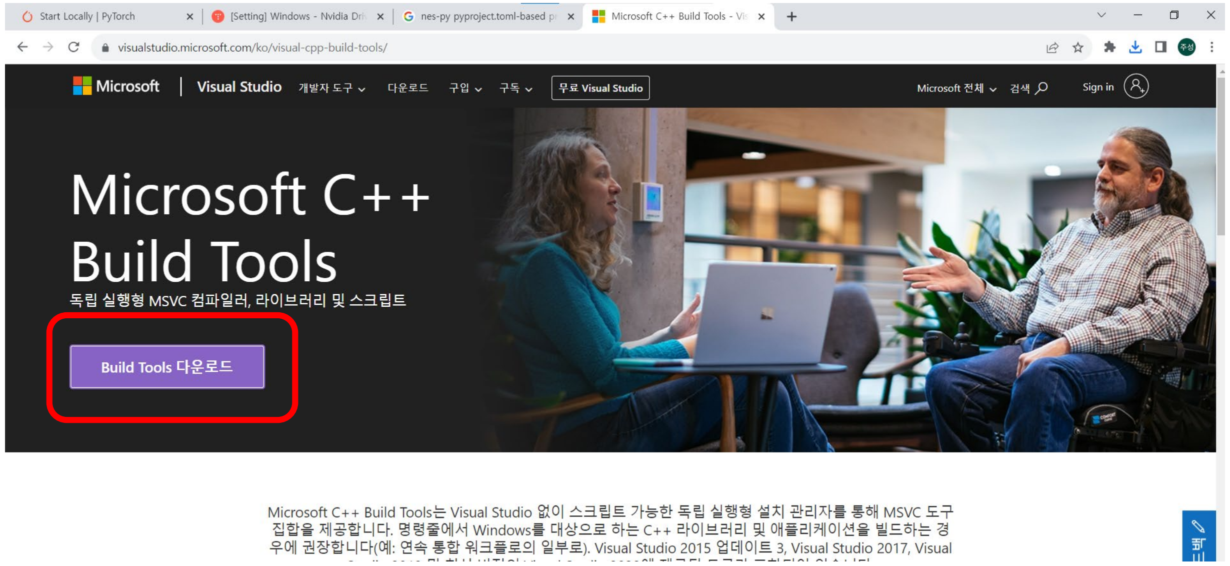
- **Matplotlib**
  - **> (MNU_RL_Demo) conda install -c conda-forge matplotlib**

Development Env. Setup Guide – OpenAI Gym (Opt.)

- **OpenAI Gym (Optional – For Debugging)**
  - **> (MNU_RL_Demo) pip install nes-py**
    - If error message occurs, go on the the MS site as bellow:
      - “~ pyproject.toml-based projects”
      - https://visualstudio.microsoft.com/ko/visual-cpp-build-tools/
      - vs_BuildTools.exe 실행



```
      copying nes_py₩app₩play_human.py -> build₩lib.win-amd64-cpython-311₩nes_py₩app
      copying nes_py₩app₩play_random.py -> build₩lib.win-amd64-cpython-311₩nes_py₩app
      copying nes_py₩app₩__init__.py -> build₩lib.win-amd64-cpython-311₩nes_py₩app
      creating build₩lib.win-amd64-cpython-311₩nes_py₩wrappers
      copying nes_py₩wrappers₩joypad_space.py -> build₩lib.win-amd64-cpython-311₩nes_py₩wrappers
      copying nes_py₩wrappers₩__init__.py -> build₩lib.win-amd64-cpython-311₩nes_py₩wrappers
      running build_ext
      building 'nes_py.lib_nes_env' extension
      error: Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools": https://visualst
udio.microsoft.com/visual-cpp-build-tools/
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
  ERROR: Failed building wheel for nes-py
  Running setup.py clean for nes-py
Failed to build nes-py
ERROR: Could not build wheels for nes-py, which is required to install pyproject.toml-based projects

(MNU_RL_Demo) C:₩Users₩JSKang>pip install pyproject-toml
```


Start Locally | PyTorch
[Setting] Windows - Nvidia Dri
nes-py pyproject.toml-based p
Microsoft C++ Build Tools - Vi
visualstudio.microsoft.com/ko/visual-cpp-build-tools/
Microsoft
Visual Studio
개발자 도구
다운로드
구입
구독
무료 Visual Studio
Microsoft 전체
검색
Sign in
Build Tools 다운로드

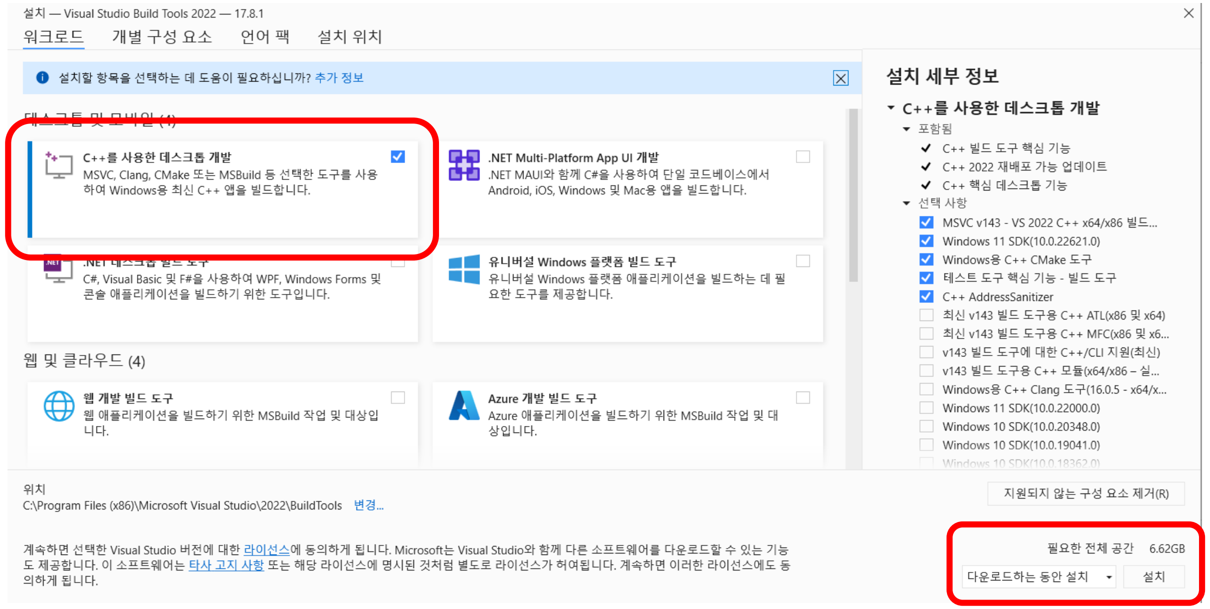
Microsoft C++ Build Tools

독립 실행형 MSVC 컴파일러, 라이브러리 및 스크립트

Microsoft C++ Build Tools는 Visual Studio 없이 스크립트 가능한 독립 실행형 설치 관리자를 통해 MSVC 도구 집합을 제공합니다. 명령줄에서 Windows를 대상으로 하는 C++ 라이브러리 및 애플리케이션을 빌드하는 경우에 권장합니다(예: 연속 통합 워크플로의 일부로). Visual Studio 2015 업데이트 3, Visual Studio 2017, Visual

Development Env. Setup Guide – OpenAI Gym (Opt.)

- OpenAI Gym (Optional – For Debugging)
  - > (MNU_RL_Demo) pip install nes-py
    - 계속 >> 'C++ 를 사용한 데스크톱 개발' 선택 및 설치 >> 설치 (few min.)
    - Nes-py 재설치 (Go back to page 7)



설치 — Visual Studio Build Tools 2022 — 17.8.1

설치할 항목을 선택하는 데 도움이 필요하십니까? 추가 정보

데스크톱 및 모바일 (4)

☑ C++를 사용한 데스크톱 개발
MSVC, Clang, CMake 또는 MSBuild 등 선택한 도구를 사용하여 Windows용 최신 C++ 앱을 빌드합니다.

□ .NET Multi-Platform App UI 개발
.NET MAUI와 함께 C#을 사용하여 단일 코드베이스에서 Android, iOS, Windows 및 Mac용 앱을 빌드합니다.

□ .NET 데스크톱 빌드 도구
C#, Visual Basic 및 F#을 사용하여 WPF, Windows Forms 및 콘솔 애플리케이션을 빌드하기 위한 도구입니다.

□ 유니버설 Windows 플랫폼 빌드 도구
유니버설 Windows 플랫폼 애플리케이션을 빌드하는 데 필요한 도구를 제공합니다.

웹 및 클라우드 (4)

□ 웹 개발 빌드 도구
웹 애플리케이션을 빌드하기 위한 MSBuild 작업 및 대상입니다.

□ Azure 개발 빌드 도구
Azure 애플리케이션을 빌드하기 위한 MSBuild 작업 및 대상입니다.

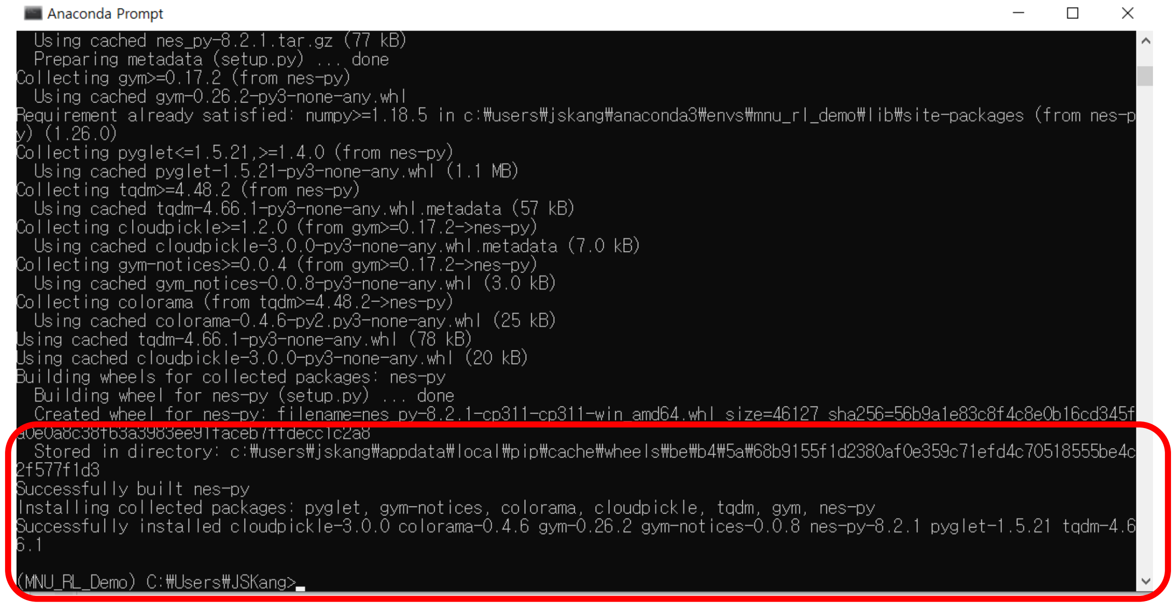
설치 세부 정보

- C++를 사용한 데스크톱 개발
  - 포함됨
    - ✓ C++ 빌드 도구 핵심 기능
    - ✓ C++ 2022 재배포 가능 업데이트
    - ✓ C++ 핵심 데스크톱 기능
  - 선택 사항
    - ☑ MSVC v143 - VS 2022 C++ x64/x86 빌드...
    - ☑ Windows 11 SDK(10.0.22621.0)
    - ☑ Windows용 C++ CMake 도구
    - ☑ 테스트 도구 핵심 기능 - 빌드 도구
    - ☑ C++ AddressSanitizer
    - □ 최신 v143 빌드 도구용 C++ ATL(x86 및 x64)
    - □ 최신 v143 빌드 도구용 C++ MFC(x86 및 x6...
    - □ v143 빌드 도구에 대한 C++/CLI 지원(최신)
    - □ v143 빌드 도구용 C++ 모듈(x64/x86 – 실...
    - □ Windows용 C++ Clang 도구(16.0.5 - x64/x...
    - □ Windows 11 SDK(10.0.22000.0)
    - □ Windows 10 SDK(10.0.20348.0)
    - □ Windows 10 SDK(10.0.19041.0)
    - □ Windows 10 SDK(10.0.18362.0)

위치
C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools 변경...

계속하면 선택한 Visual Studio 버전에 대한 라이선스에 동의하게 됩니다. Microsoft는 Visual Studio와 함께 다른 소프트웨어를 다운로드할 수 있는 기능도 제공합니다. 이 소프트웨어는 타사 고지 사항 또는 해당 라이선스에 명시된 것처럼 별도로 라이선스가 허여됩니다. 계속하면 이러한 라이선스에도 동의하게 됩니다.

필요한 전체 공간 6.62GB

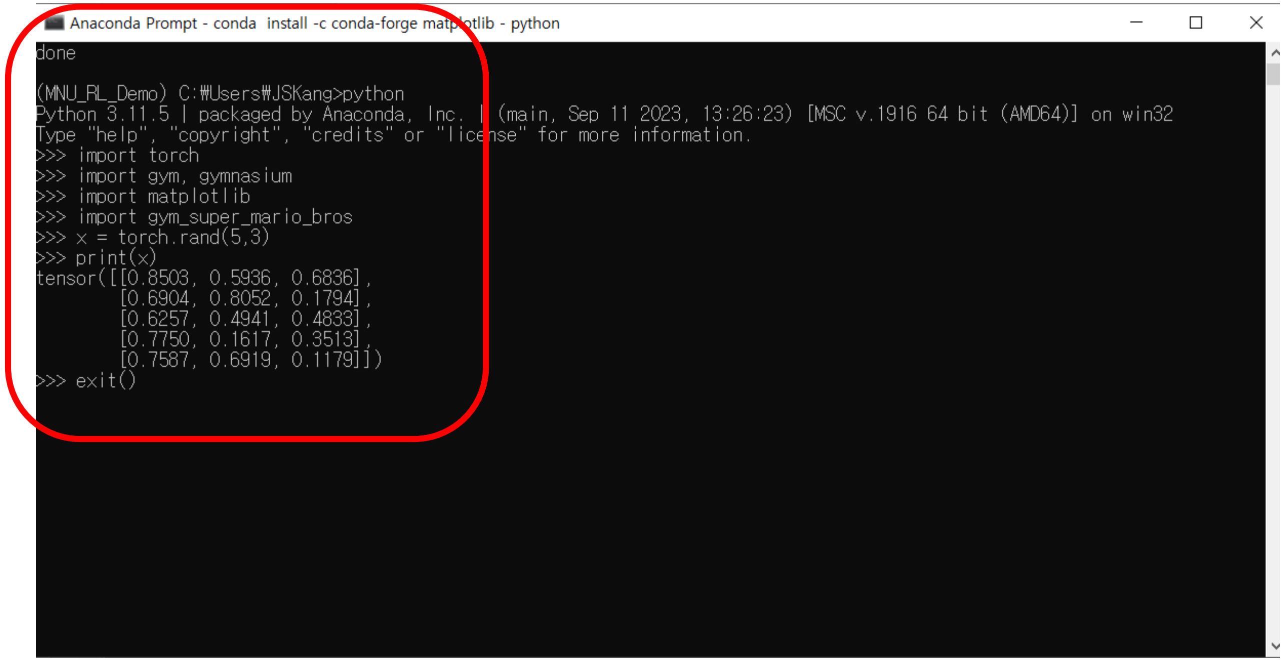


Anaconda Prompt

```
  Using cached nes_py-8.2.1.tar.gz (77 kB)
  Preparing metadata (setup.py) ... done
Collecting gym>=0.17.2 (from nes-py)
  Using cached gym-0.26.2-py3-none-any.whl
Requirement already satisfied: numpy>=1.18.5 in c:₩users₩jskang₩anaconda3₩envs₩mnu_rl_demo₩lib₩site-packages (from nes-p
y) (1.26.0)
Collecting pyglet<=1.5.21,>=1.4.0 (from nes-py)
  Using cached pyglet-1.5.21-py3-none-any.whl (1.1 MB)
Collecting tqdm>=4.48.2 (from nes-py)
  Using cached tqdm-4.66.1-py3-none-any.whl.metadata (57 kB)
Collecting cloudpickle>=1.2.0 (from gym>=0.17.2->nes-py)
  Using cached cloudpickle-3.0.0-py3-none-any.whl.metadata (7.0 kB)
Collecting gym-notices>=0.0.4 (from gym>=0.17.2->nes-py)
  Using cached gym_notices-0.0.8-py3-none-any.whl (3.0 kB)
Collecting colorama (from tqdm>=4.48.2->nes-py)
  Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Using cached tqdm-4.66.1-py3-none-any.whl (78 kB)
Using cached cloudpickle-3.0.0-py3-none-any.whl (20 kB)
Building wheels for collected packages: nes-py
  Building wheel for nes-py (setup.py) ... done
  Created wheel for nes-py: filename=nes_py-8.2.1-cp311-cp311-win_amd64.whl size=46127 sha256=56b9a1e83c8f4c8e0b16cd345f
a0e0a8c38f63a3983ee91faceb7ffdecc1c2a8
  Stored in directory: c:₩users₩jskang₩appdata₩local₩pip₩cache₩wheels₩be₩b4₩5a₩68b9155f1d2380af0e359c71efd4c70518555be4c
2f577f1d3
Successfully built nes-py
Installing collected packages: pyglet, gym-notices, colorama, cloudpickle, tqdm, gym, nes-py
Successfully installed cloudpickle-3.0.0 colorama-0.4.6 gym-0.26.2 gym-notices-0.0.8 nes-py-8.2.1 pyglet-1.5.21 tqdm-4.6
6.1

(MNU_RL_Demo) C:₩Users₩JSKang>_
```

Development Env. Setup Guide – Env. Setup Test

- Env. Setup Test
  - > (MNU_RL_Demo) python
  - >> import torch
  - >> import gym, gymnasium
  - >> import matplotlib
  - >> import gym_super_mario_bros
  - >> x = torch.rand(5,3)
  - >> print(x)

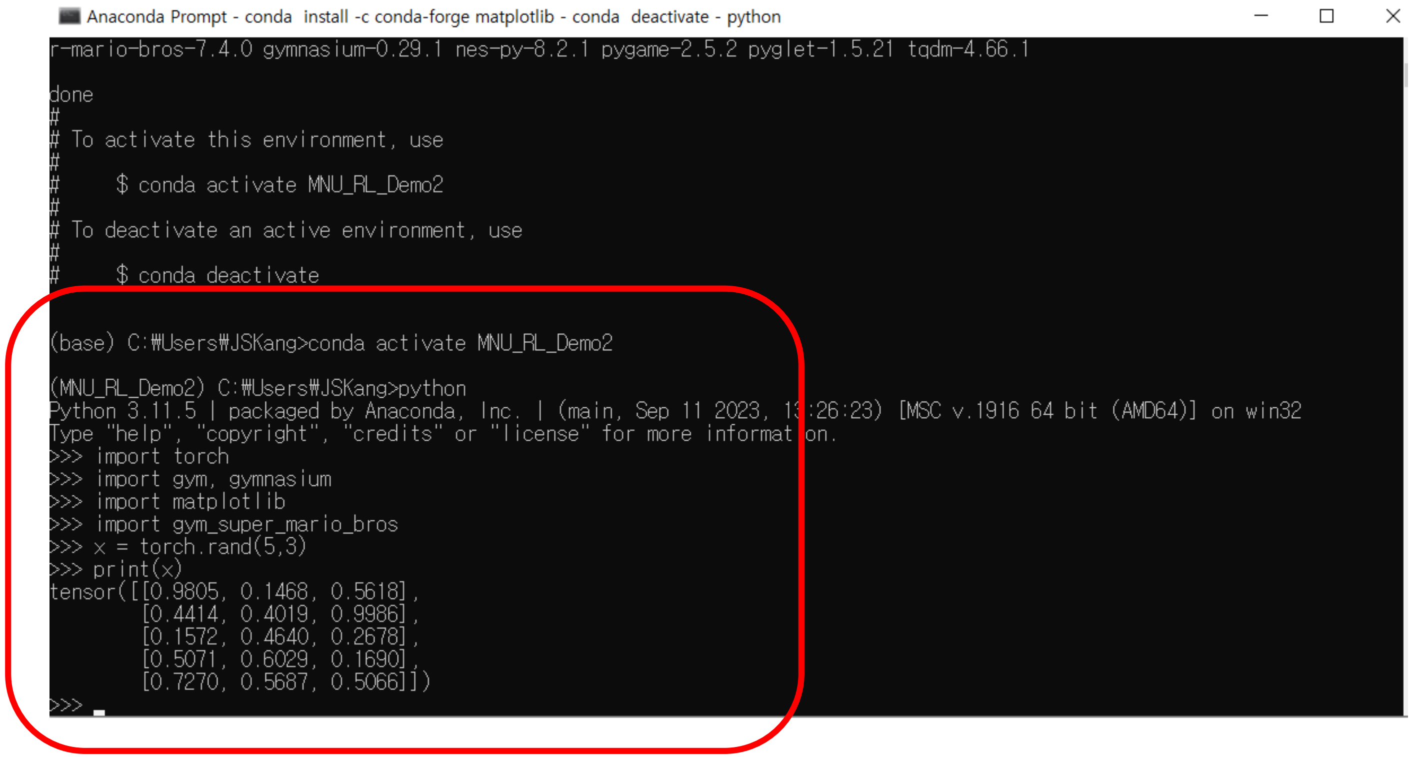


Anaconda Prompt - conda install -c conda-forge matplotlib - python

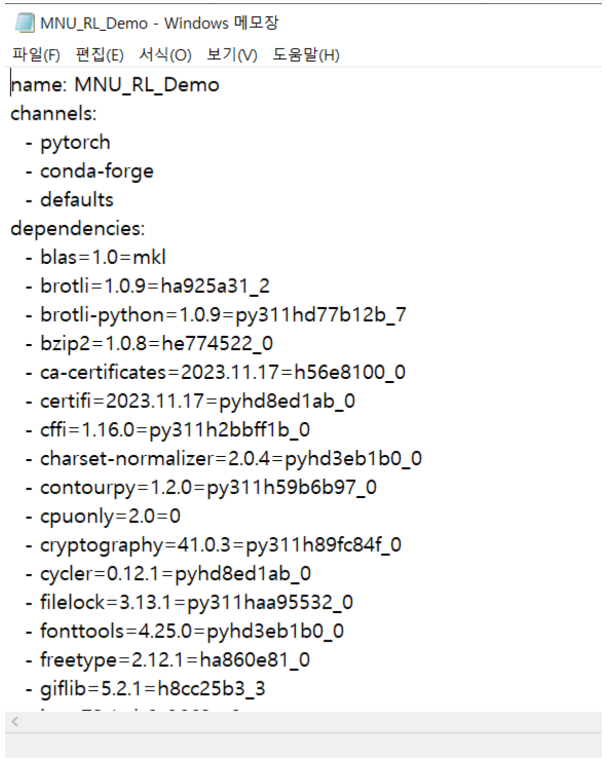
```
done

(MNU_RL_Demo) C:₩Users₩JSKang>python
Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> import gym, gymnasium
>>> import matplotlib
>>> import gym_super_mario_bros
>>> x = torch.rand(5,3)
>>> print(x)
tensor([[0.8503, 0.5936, 0.6836],
        [0.6904, 0.8052, 0.1794],
        [0.6257, 0.4941, 0.4833],
        [0.7750, 0.1617, 0.3513],
        [0.7587, 0.6919, 0.1179]])
>>> exit()
```

Development Env. Setup Guide – Env. Export and Import Method(opt)

- *Env. Export and Import method*
  - **> (MNU_RL_Demo) conda deactivate**
  - **> (base) conda env create -n MNU_RL_Demo2 -f MNU_RL_Demo.yaml**
    - Export env.) conda env export –n MNU_RL_Demo > MNU_RL_Demo.yaml


Anaconda Prompt - conda install -c conda-forge matplotlib - conda deactivate - python

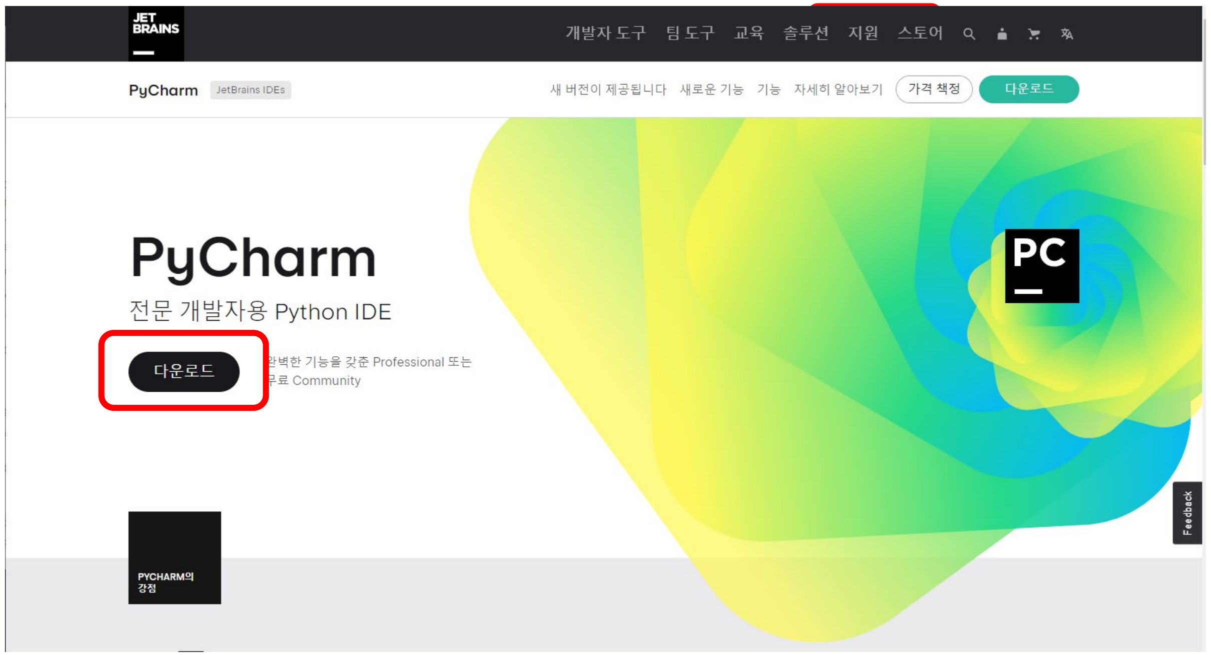
```
r-mario-bros-7.4.0 gymnasium-0.29.1 nes-py-8.2.1 pygame-2.5.2 pyglet-1.5.21 tqdm-4.66.1

done
#
# To activate this environment, use
#
#     $ conda activate MNU_RL_Demo2
#
# To deactivate an active environment, use
#
#     $ conda deactivate


(base) C:₩Users₩JSKang>conda activate MNU_RL_Demo2

(MNU_RL_Demo2) C:₩Users₩JSKang>python
Python 3.11.5 | packaged by Anaconda, Inc. | (main, Sep 11 2023, 13:26:23) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> import gym, gymnasium
>>> import matplotlib
>>> import gym_super_mario_bros
>>> x = torch.rand(5,3)
>>> print(x)
tensor([[0.9805, 0.1468, 0.5618],
        [0.4414, 0.4019, 0.9986],
        [0.1572, 0.4640, 0.2678],
        [0.5071, 0.6029, 0.1690],
        [0.7270, 0.5687, 0.5066]])
>>>
```


MNU_RL_Demo - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
name: MNU_RL_Demo
channels:
  - pytorch
  - conda-forge
  - defaults
dependencies:
  - blas=1.0=mkl
  - brotli=1.0.9=ha925a31_2
  - brotli-python=1.0.9=py311hd77b12b_7
  - bzip2=1.0.8=he774522_0
  - ca-certificates=2023.11.17=h56e8100_0
  - certifi=2023.11.17=pyhd8ed1ab_0
  - cffi=1.16.0=py311h2bbff1b_0
  - charset-normalizer=2.0.4=pyhd3eb1b0_0
  - contourpy=1.2.0=py311h59b6b97_0
  - cpuonly=2.0=0
  - cryptography=41.0.3=py311h89fc84f_0
  - cycler=0.12.1=pyhd8ed1ab_0
  - filelock=3.13.1=py311haa95532_0
  - fonttools=4.25.0=pyhd3eb1b0_0
  - freetype=2.12.1=ha860e81_0
  - giflib=5.2.1=h8cc25b3_3
```

Development Env. Setup Guide – Pycharm

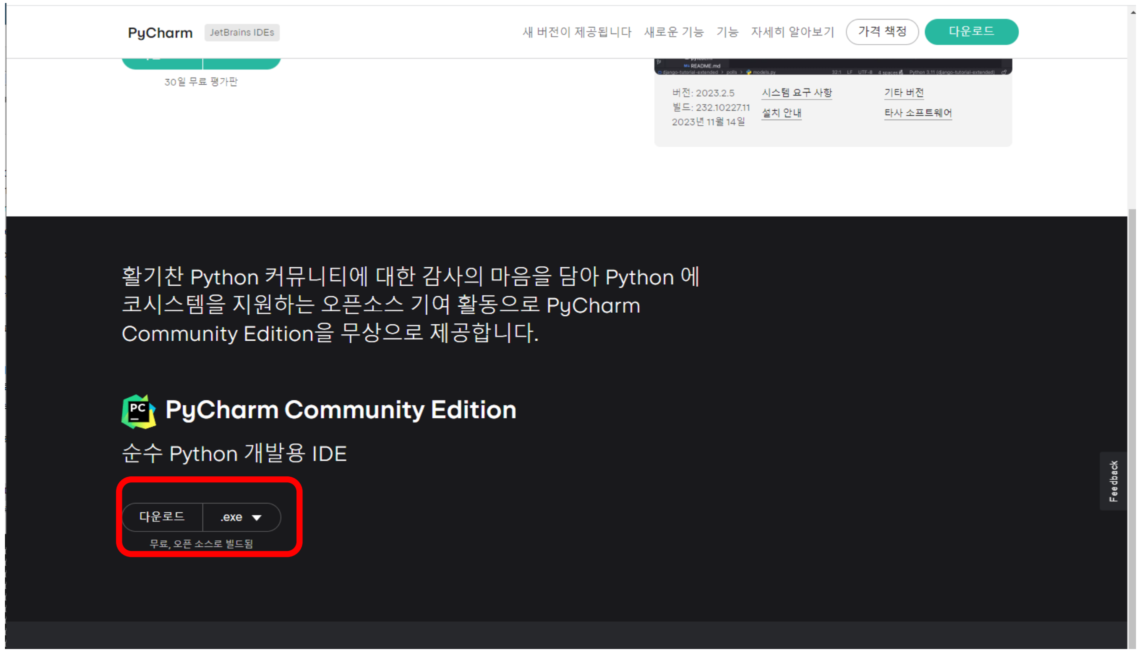
- **Pycharm**
  - **https://www.jetbrains.com/ko-kr/pycharm/**
  - **다운로드 > Pycharm Community Edition (다운로드) > (files) pycharm-commununity-2023.2.5.exe 실행**



PyCharm

전문 개발자용 Python IDE

완벽한 기능을 갖춘 Professional 또는 무료 Community



버전: 2023.2.5
빌드: 232.10227.11
2023년 11월 14일

활기찬 Python 커뮤니티에 대한 감사의 마음을 담아 Python 에코시스템을 지원하는 오픈소스 기여 활동으로 PyCharm Community Edition을 무상으로 제공합니다.

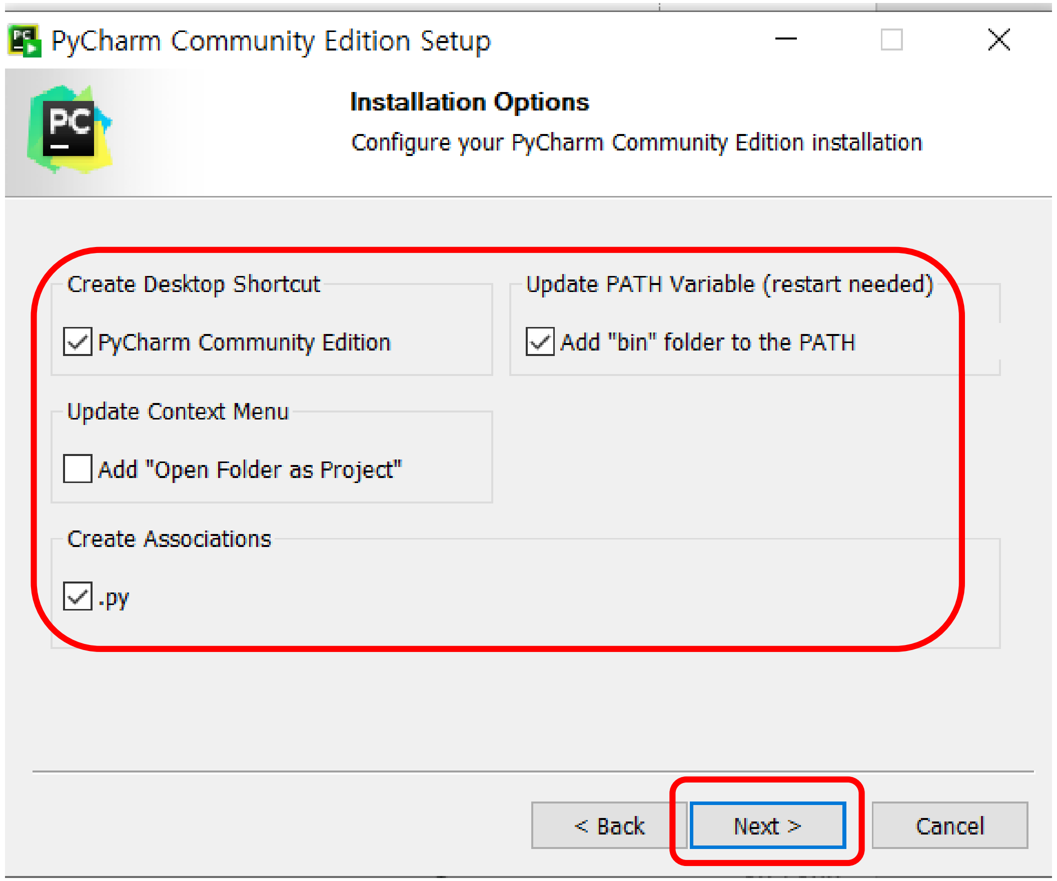
PyCharm Community Edition

순수 Python 개발용 IDE

무료, 오픈 소스로 빌드됨

Development Env. Setup Guide – Pycharm

- Pycharm
  - Next >> Next >> Check options and Next >> Install



PyCharm Community Edition Setup

Installation Options

Configure your PyCharm Community Edition installation

Create Desktop Shortcut

☑ PyCharm Community Edition

Update PATH Variable (restart needed)

☑ Add "bin" folder to the PATH

Update Context Menu

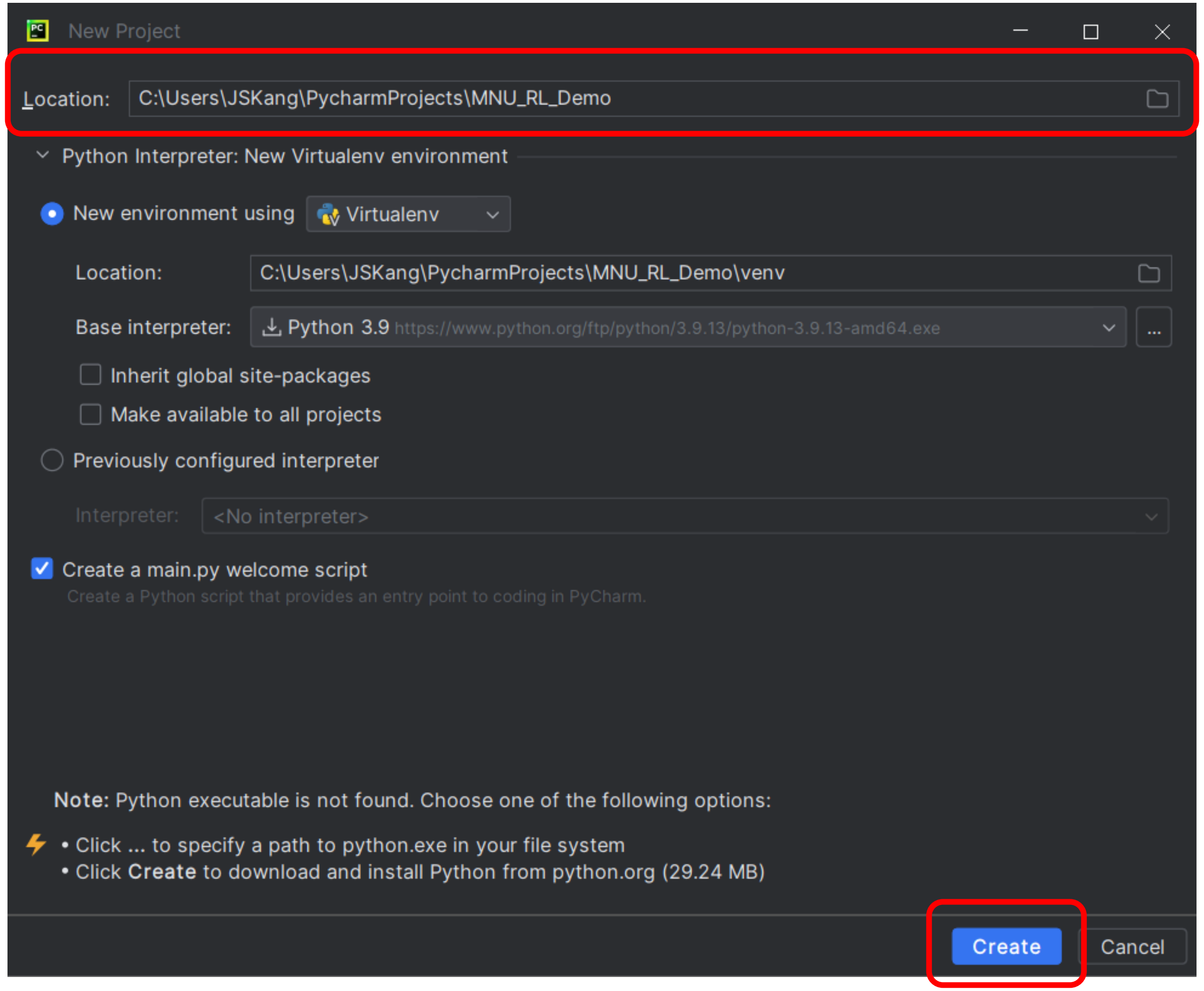
☐ Add "Open Folder as Project"

Create Associations

☑ .py

Development Env. Setup Guide – Pycharm

- **Pycharm Project Setting**
  - **검색창 >> Pycharm Community Edition 실행**
  - **Create New Project**
    - Location 지정 (Project Title)
    - Interpreter location
      - C:₩Users₩JSKang₩anaconda3₩envs₩MNU_RL_Demo₩python.exe
      - C:₩ProgramData₩envs₩MNU_RL_Demo₩python.exe



Location: C:\Users\JSKang\PycharmProjects\MNU_RL_Demo

Python Interpreter: New Virtualenv environment

● New environment using Virtualenv

Location: C:\Users\JSKang\PycharmProjects\MNU_RL_Demo\venv

Base interpreter: Python 3.9 https://www.python.org/ftp/python/3.9.13/python-3.9.13-amd64.exe

□ Inherit global site-packages

□ Make available to all projects

○ Previously configured interpreter

Interpreter: <No interpreter>

☑ Create a main.py welcome script

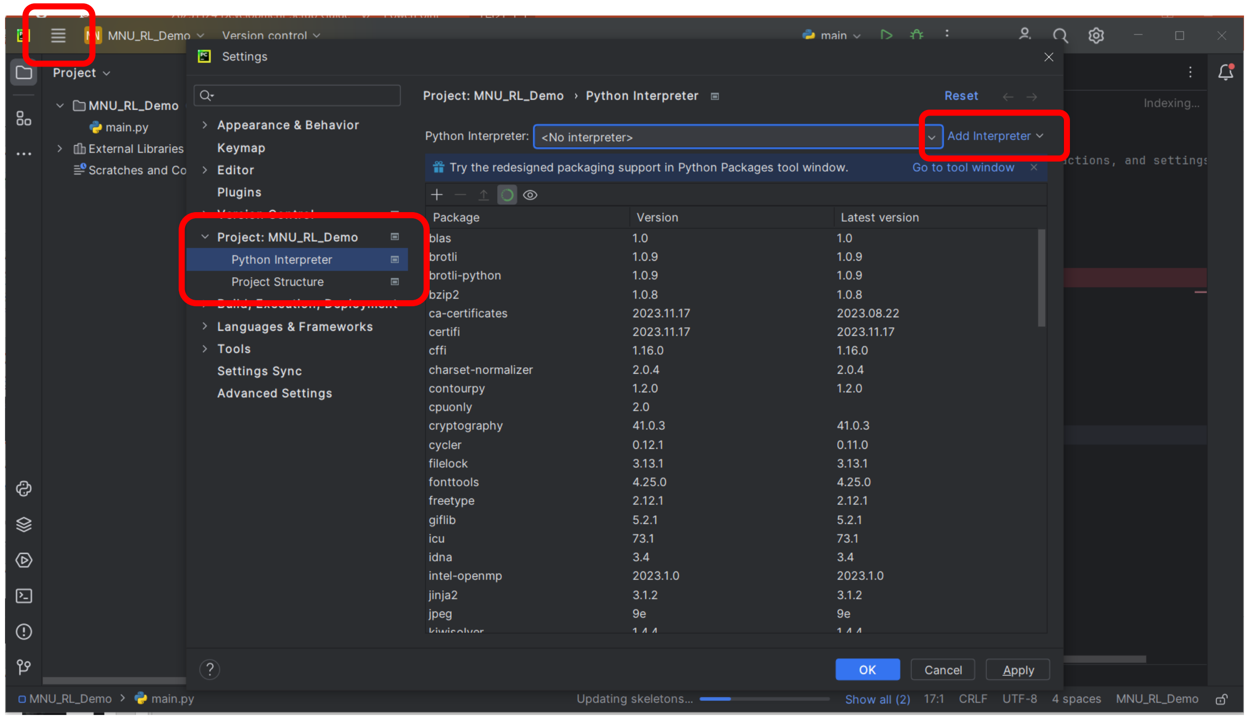
Create a Python script that provides an entry point to coding in PyCharm.

Note: Python executable is not found. Choose one of the following options:

- Click ... to specify a path to python.exe in your file system
- Click **Create** to download and install Python from python.org (29.24 MB)

Development Env. Setup Guide – Pycharm

- Pycharm Project Setting
  - Project Interpreter 설정방법
    - File >> Settings >> Project >> Project Interpreter >> Add Interpreter >> Add local interpreter >> Conda Environment >> Select conda env. >> ok >> ok

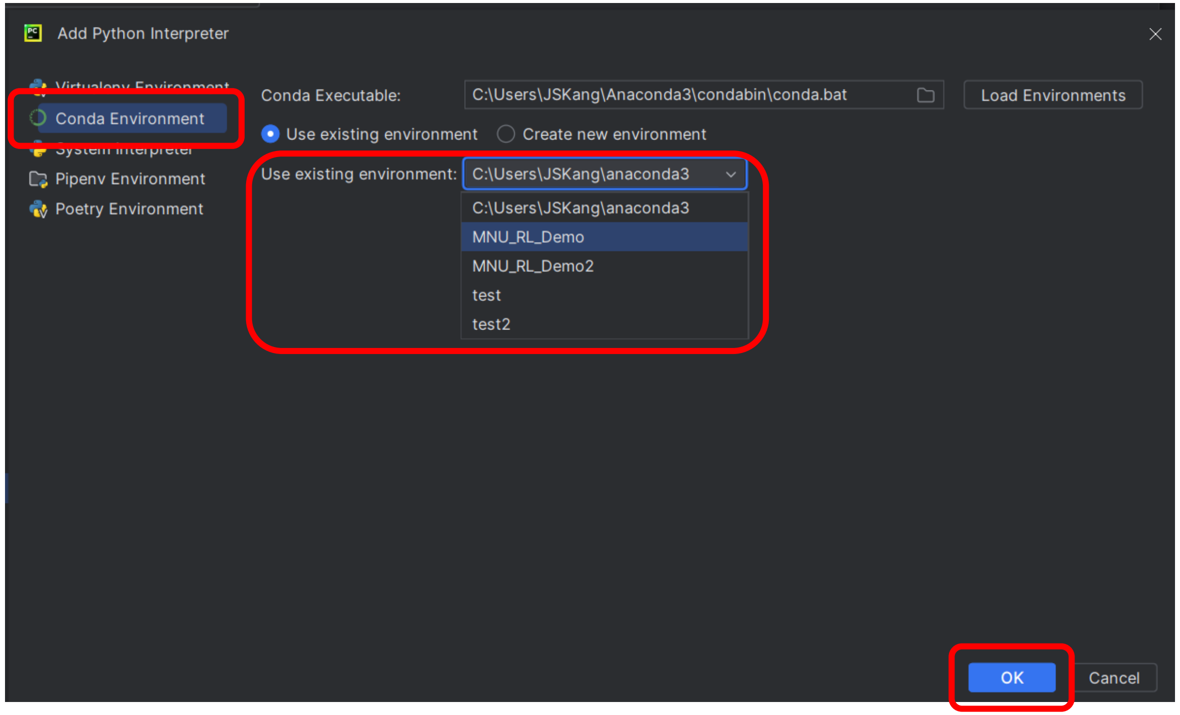


Project: MNU_RL_Demo › Python Interpreter

Python Interpreter: <No interpreter>

Try the redesigned packaging support in Python Packages tool window.

Package	Version	Latest version
blas	1.0	1.0
brotli	1.0.9	1.0.9
brotli-python	1.0.9	1.0.9
bzip2	1.0.8	1.0.8
ca-certificates	2023.11.17	2023.08.22
certifi	2023.11.17	2023.11.17
cffi	1.16.0	1.16.0
charset-normalizer	2.0.4	2.0.4
contourpy	1.2.0	1.2.0
cpuonly	2.0	
cryptography	41.0.3	41.0.3
cycler	0.12.1	0.11.0
filelock	3.13.1	3.13.1
fonttools	4.25.0	4.25.0
freetype	2.12.1	2.12.1
giflib	5.2.1	5.2.1
icu	73.1	73.1
idna	3.4	3.4
intel-openmp	2023.1.0	2023.1.0
jinja2	3.1.2	3.1.2
jpeg	9e	9e
kiwisolver	1.4.4	1.4.4



Conda Executable: C:\Users\JSKang\Anaconda3\condabin\conda.bat

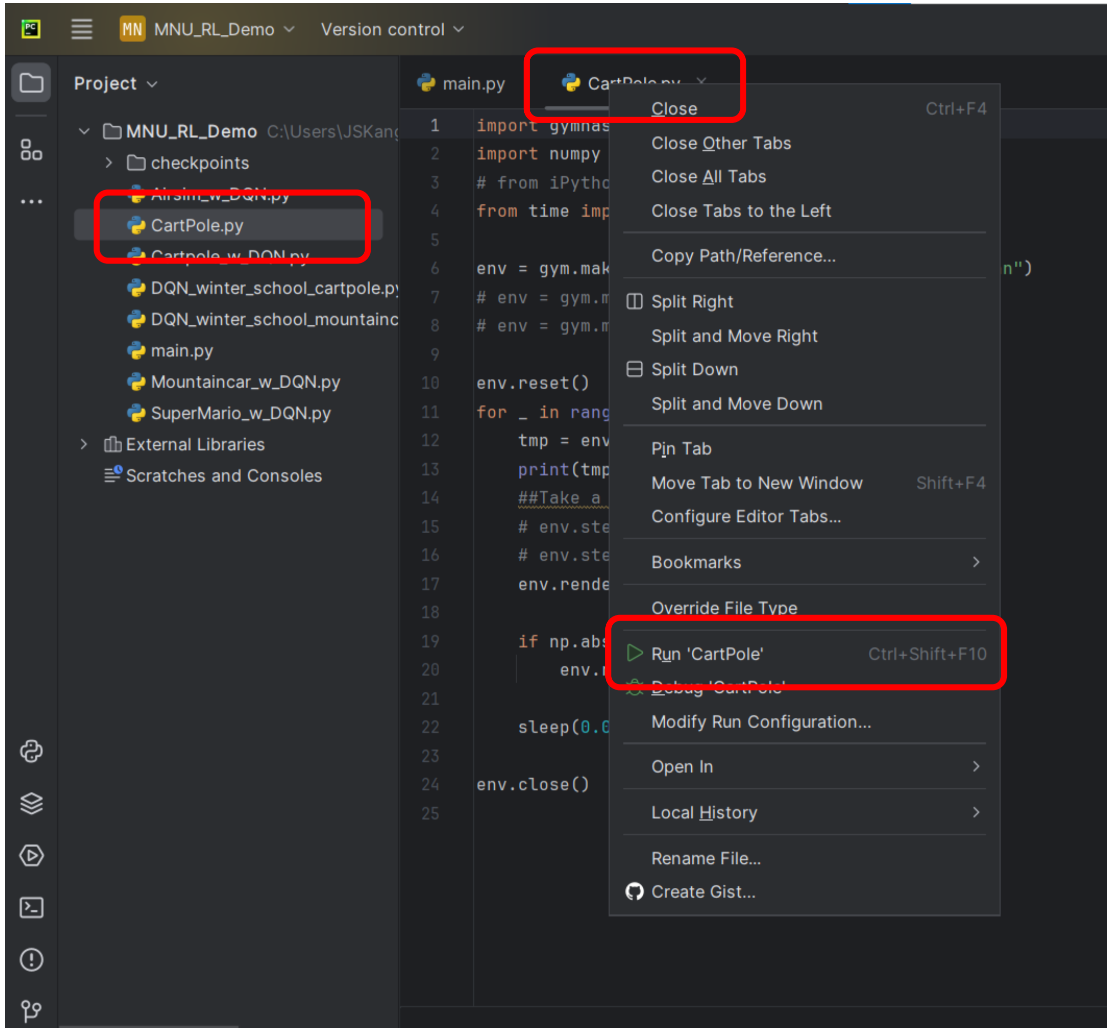
● Use existing environment ○ Create new environment

Use existing environment: C:\Users\JSKang\anaconda3

C:\Users\JSKang\anaconda3
MNU_RL_Demo
MNU_RL_Demo2
test
test2

Development Env. Setup Guide – Pycharm Test code

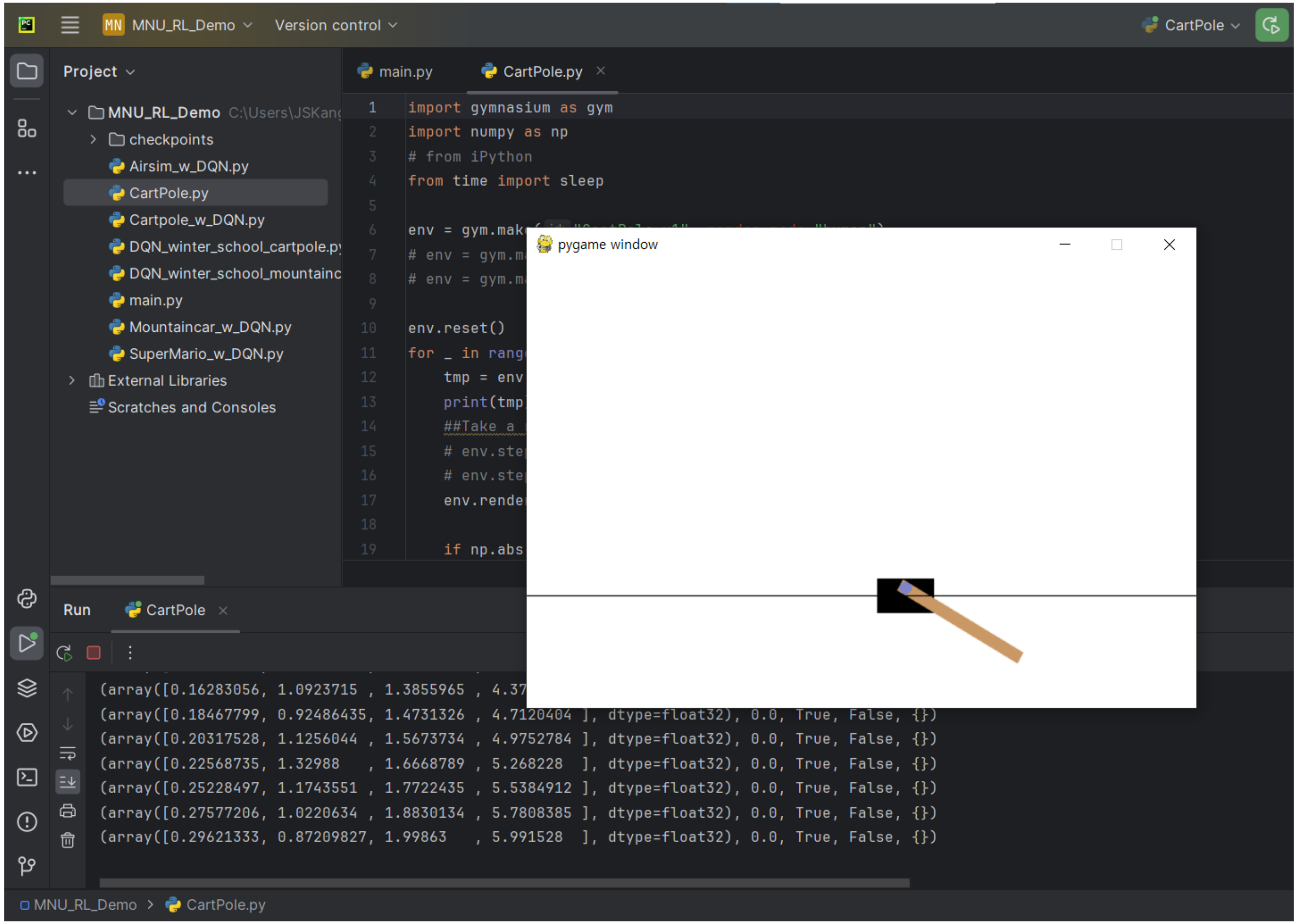
- **Pycharm Setup test code**
  - **Copy all files to project location**
    - C:₩Users₩JSKang₩PycharmProjects ₩MNU_RL_Demo
  - **‘CartPole.py’ file**
    - Right click >> Run ‘CartPole”



```
import gymnas
import numpy
# from iPytho
from time imp

env = gym.mak                                        n")
# env = gym.m
# env = gym.m

env.reset()
for _ in rang
    tmp = env
    print(tmp
    ##Take a
    # env.ste
    # env.ste
    env.rende

    if np.abs
        env.r

    sleep(0.0

env.close()
```

Development Env. Setup Guide – Pycharm Test code

- **Pycharm Setup test code**
  - **Setup Done!**
    - **Enjoy your RL**

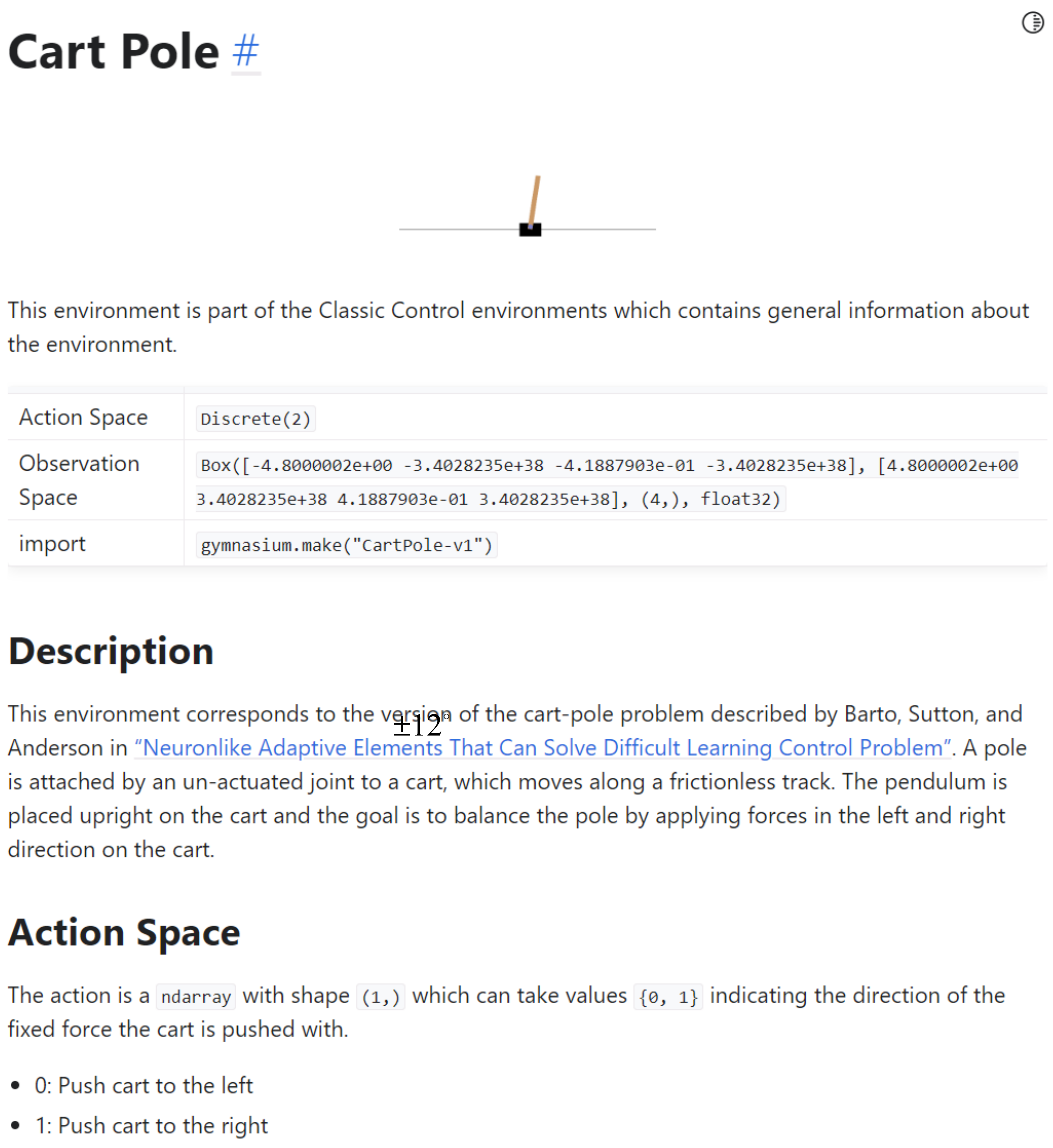


```python
import gymnasium as gym
import numpy as np
# from iPython
from time import sleep

env = gym.mak
# env = gym.m
# env = gym.m

env.reset()
for _ in rang
    tmp = env
    print(tmp
    ##Take a
    # env.ste
    # env.ste
    env.rende

    if np.abs
```

```
(array([0.16283056, 1.0923715 , 1.3855965 , 4.37
(array([0.18467799, 0.92486435, 1.4731326 , 4.7120404 ], dtype=float32), 0.0, True, False, {})
(array([0.20317528, 1.1256044 , 1.5673734 , 4.9752784 ], dtype=float32), 0.0, True, False, {})
(array([0.22568735, 1.32988   , 1.6668789 , 5.268228  ], dtype=float32), 0.0, True, False, {})
(array([0.25228497, 1.1743551 , 1.7722435 , 5.5384912 ], dtype=float32), 0.0, True, False, {})
(array([0.27577206, 1.0220634 , 1.8830134 , 5.7808385 ], dtype=float32), 0.0, True, False, {})
(array([0.29621333, 0.87209827, 1.99863   , 5.991528  ], dtype=float32), 0.0, True, False, {})
```

Development Env. Setup Guide – Pycharm Test code

- **Test Sample Code**
  - https://gymnasium.farama.org/environments/classic_control/cart_pole/
  - **Action Space**
    - 0 : Push Cart to the left
    - 1 : Push Cart to the right
  - **Observation Space**
    - {(Position, Velocity, Angle, Angular Velocity)}
  - **Rewards**
    - +1 for every step taken.
  - **Terminates**
    - Angle greater than ±12°
    - Position greater than ±2.4
    - Step length greater than 500



Cart Pole

This environment is part of the Classic Control environments which contains general information about the environment.

Action Space	`Discrete(2)`
Observation Space	`Box([-4.8000002e+00 -3.4028235e+38 -4.1887903e-01 -3.4028235e+38], [4.8000002e+00 3.4028235e+38 4.1887903e-01 3.4028235e+38], (4,), float32)`
import	`gymnasium.make("CartPole-v1")`

Description

This environment corresponds to the version of the cart-pole problem described by Barto, Sutton, and Anderson in "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problem". A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart.

±12°

Action Space

The action is a `ndarray` with shape `(1,)` which can take values `{0, 1}` indicating the direction of the fixed force the cart is pushed with.

- 0: Push cart to the left
- 1: Push cart to the right