

# Unit A

# 참고자료





# 1. 실습환경 구성하기

## Python 설치 및 환경 개요

- Python 배포판 종류
- Python (여러 라이브러리를 설치하면 아나콘다와 동일한 환경)
  - Intel사의 인텔 최적화 배포판  
<https://www.intel.com/content/www/us/en/developer/tools/oneapi/distribution-for-python.html>
  - Anaconda Python 배포판에서는 conda command로 가상환경, Python 패키지 설치를 관리할 수 있음. Anaconda (과학, 수학 등 숫자와 관련된 라이브러리가 미리 설치되어 있음) 1GB ~ 수GB
  - Miniconda (아나콘다에서 라이브러리를 많이 제외한거) - 경량 버전 200MB ~ 수GB
    - Python 3.8
    - Miniconda3 Windows 64-bit
    - 57.0 MiB
  - Enthought Canopy
  - WinPython



# 1. 실습환경 구성하기

## Python 설치 및 환경 개요

- Python을 이용하는 방법은 다양하게 존재하며 사용하기 원하는 환경에 따라서 선택
- 개인 컴퓨터에서 Python 사용시에는 홈페이지 (<https://www.python.org>)에서 설치 파일을 다운로드하여 설치할 수 있으나, Python 기본 에디터보다는 다양한 통합개발환경 (IDE)을 활용하여 프로그래밍 가능
- 대표적인 통합개발환경(IDE): PyCharm(<https://www.jetbrains.com/pycharm>), Visual Studio Code(<https://code.visualstudio.com>), Eclipse에서 플러그인 형태로 제공되는 pydev(<http://www.pydev.org/>)



# 1. 실습환경 구성하기

## Python 설치 및 환경 개요

- IDE 설치

- Google Colaboratory (본인 PC 설정 필요 없음, Chrome 브라우저 사용)
- Jupyter Notebook
  1. colab 과 인터페이스가 매우 유사
  2. Python의 문법, 라이브러리 사용법을 익히기에 매우 편리 (명령문 한줄씩 실행 가능, 이전 실행 내역 기억)
- PyCharm, Visual Studio Code (VS Code)  
주피터 노트북은 전체 프로그램 소스 단위가 아닌 한 줄씩 수행이 되므로 하나의 프로그램으로 관리가 힘듦  
웹 애플리케이션을 만들려면 주피터 노트북으로는 어려움...
- 본 강의에서는 윈도 환경에서 아나콘다(Anaconda) 설치 없이 파이썬과 주피터 노트북(Jupyter Notebook), PyCharm 을 설치한다.



# 1. 실습환경 구성하기

## 왜 파이썬인가?

- 복잡한 데이터 가공과 자동화 작업에 인기 있는 언어는 파이썬이다. 파이썬은 많은 회사에서 데이터 과학뿐 아니라 C/C++, Java 등처럼 웹서버와 백엔드 시스템 프로그래밍에도 사용되는 ‘진짜’ 프로그래밍 언어다. 배우기 쉽고 가독성이 높은 코드 덕분에 프로그래밍 교육에도 많이 쓰인다.
- 데이터 취득, 데이터 가공(특히 큰 데이터!), 자동화 등에서 파이썬을 사용하면 훨씬 효율적이고 쉽게 작업할 수 있는 것들이 많다.

## 파이썬과 중요 패키지 설치

- 아나콘다(Anaconda)가 유료화 조짐을 보이는데 단순히 주피터 노트북만 사용하고자 한다면 가볍게 파이썬과 주피터 노트북을 설치하면 된다. 단, 데이터 분석 등 원하는 라이브러리는 직접 설치해야 한다.



# 1. 실습환경 구성하기

## 파이썬 다운로드

- 파이썬 공식 홈페이지에서 최신버전의 파이썬을 확인할 수 있는데 이를 다운로드 받으면 주피터 노트북 설치가 원활하게 되지 않을 가능성이 매우 높아 이전 버전을 다운로드 받아야 한다.
- 파이썬 설치와 주피터 노트북 설치를 안정적으로 하기 위해서는 minor 버전이 1~2 정도 낮은 파이썬 설치를 권장하는데 본 포스팅 작성 시점에서 파이썬의 최신 버전은 3.12으로 3.10 버전을 설치하고자 한다.
- 스크롤을 내리다 보면 3.10 버전 중 가장 최신의 다운로드 링크를 다음과 같이 볼 수 있는데 64bit 운영체제용 설치파일은 빨간 화살표가 가리키는 마지막 파일이다.



# 1. 실습환경 구성하기

## 파이썬 다운로드

- <https://www.python.org/downloads/windows/>에서  
Download Windows installer (64-bit)

Note that Python 3.8.17 cannot be used on Windows XP or earlier.

- No files for this release.

Python 3.9.17 - June 6, 2023

Note that Python 3.9.17 cannot be used on Windows 7 or earlier.

- No files for this release.

Python 3.10.11 - April 5, 2023

Note that Python 3.10.11 cannot be used on Windows 7 or earlier.

- Download Windows installer (64-bit)
- Download Windows help file
- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (32-bit)
- Download Windows installer (32-bit)

Python 3.11.3 - April 5, 2023

Note that Python 3.11.3 cannot be used on Windows 7 or earlier.

- Download Windows installer (64-bit)
- Download Windows installer (ARM64)
- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (ARM64)
- Download Windows installer (32-bit)

Python 3.10.10 - Feb. 8, 2023

Note that Python 3.10.10 cannot be used on Windows 7 or earlier.

- Download Windows installer (64-bit)
- Download Windows help file

Python 3.12.0a4 - Jan. 10, 2023

- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (ARM64)
- Download Windows installer (64-bit)
- Download Windows installer (32-bit)
- Download Windows installer (ARM64)
- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (ARM64)

Python 3.12.0a3 - Dec. 6, 2022

- Download Windows installer (64-bit)
- Download Windows installer (32-bit)
- Download Windows installer (ARM64)
- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (ARM64)

Python 3.12.0a2 - Nov. 15, 2022

- Download Windows installer (64-bit)
- Download Windows installer (32-bit)
- Download Windows installer (ARM64)
- Download Windows embeddable package (64-bit)
- Download Windows embeddable package (32-bit)
- Download Windows embeddable package (ARM64)

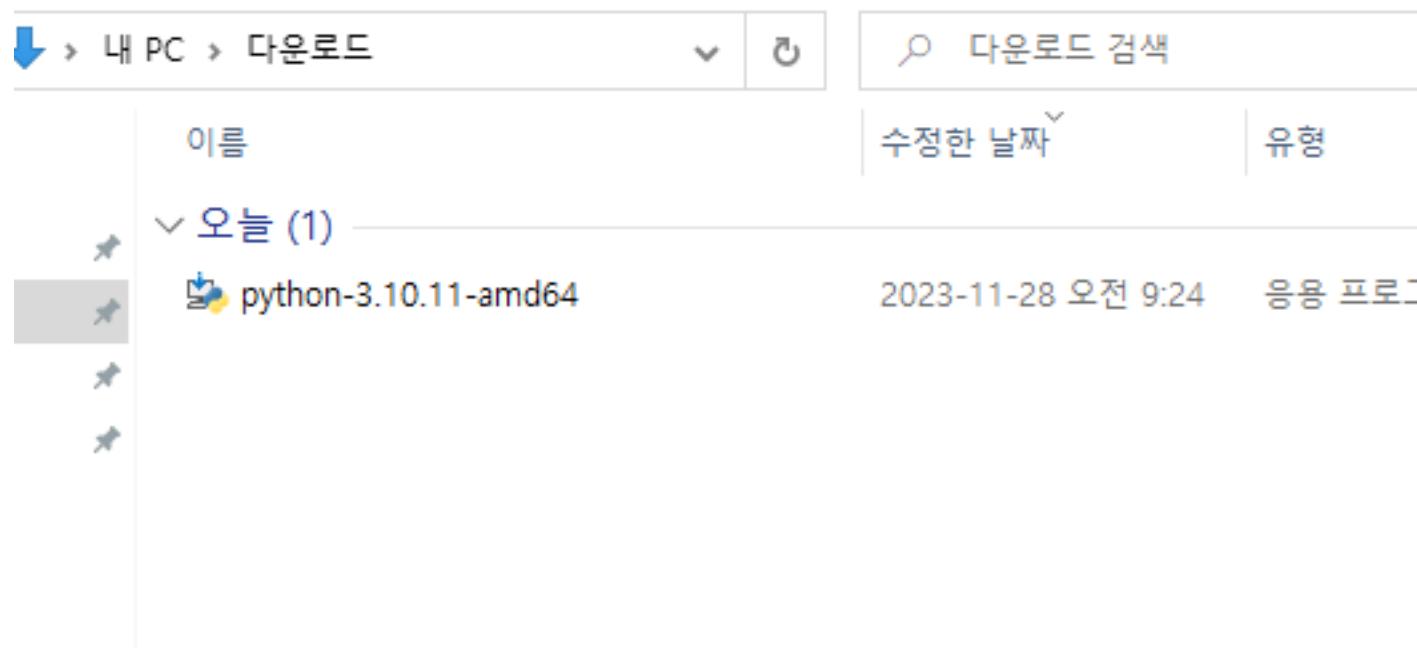
Python 3.12.0a1 - Oct. 25, 2022

- Download Windows installer (64-bit)
- Download Windows installer (32-bit)

# 1. 실습환경 구성하기

## 파이썬 다운로드

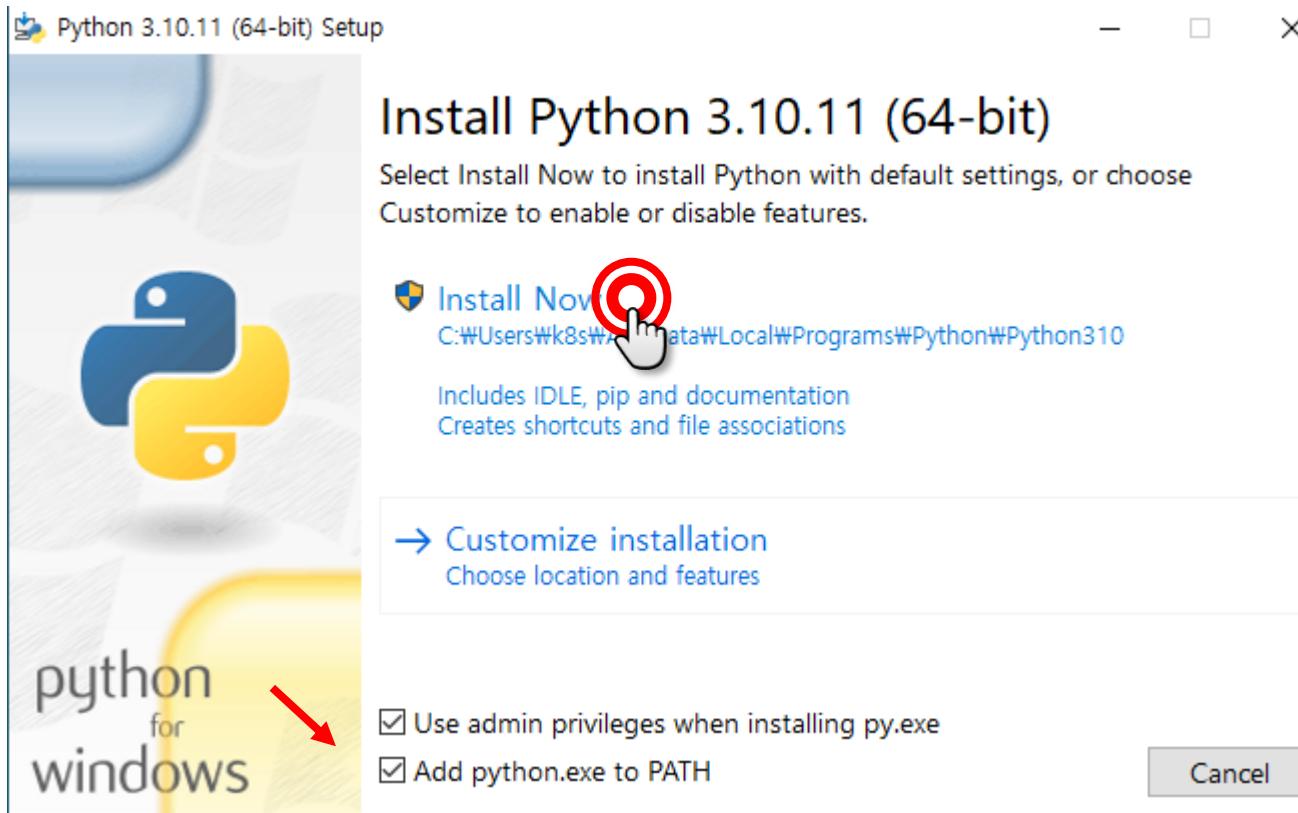
- python-3.10.11-amd64.exe



# 1. 실습환경 구성하기

## 파이썬 설치

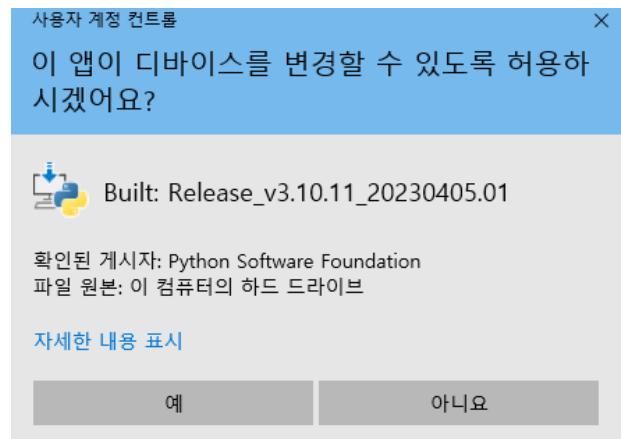
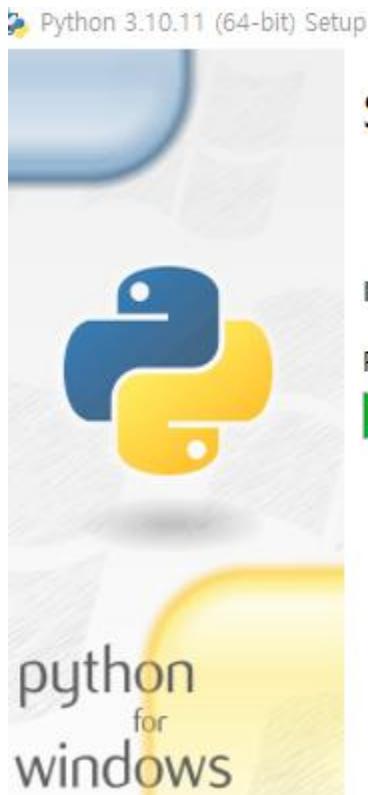
- 다운로드 받은 설치파일을 실행하면 다음과 같은 설치 화면을 볼 수 있다. 가장 아래쪽에 “Add Python.exe to PATH”를 체크해야 보다 손쉽게 파이썬 및 주피터 노트북을 설치할 수 있다.



# 1. 실습환경 구성하기

## 파이썬 설치

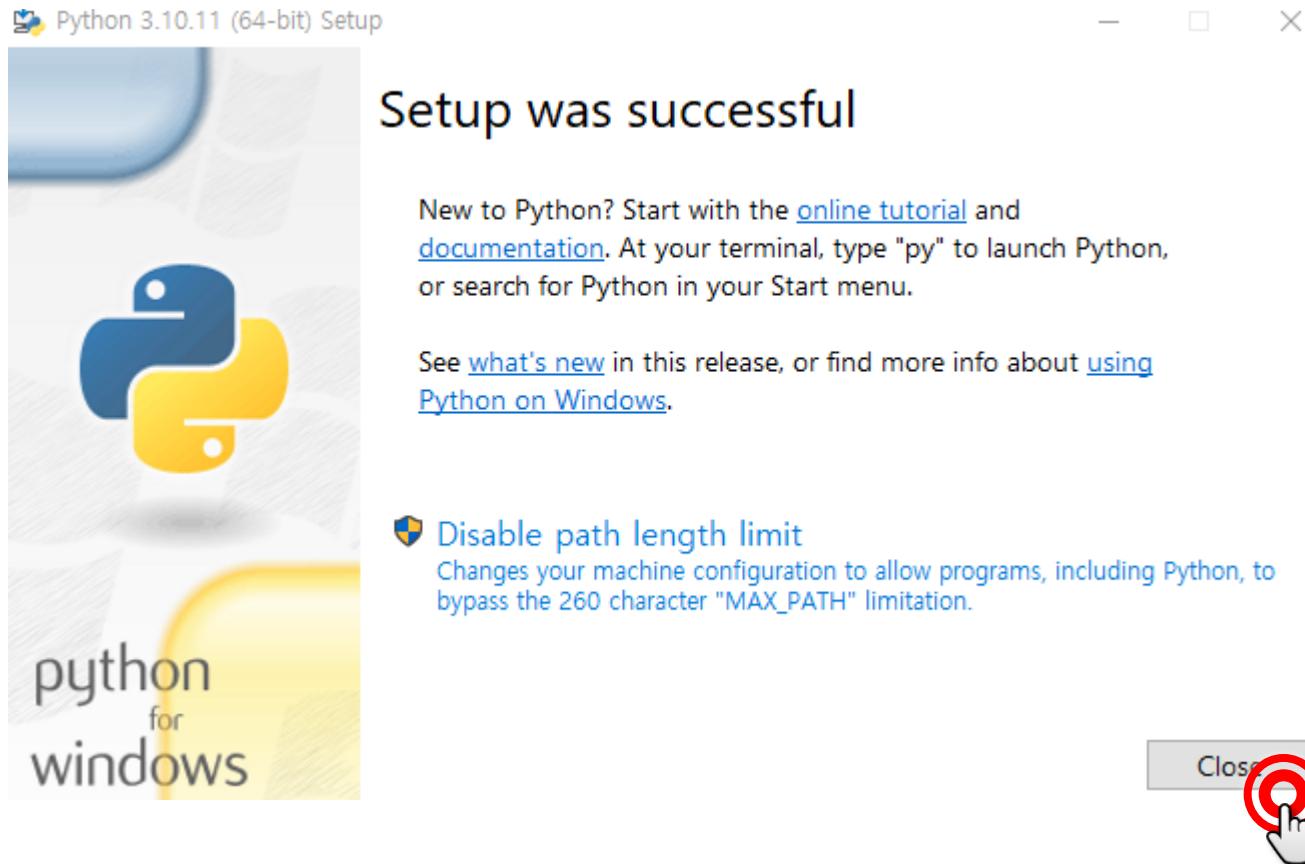
### ● 설치



# 1. 실습환경 구성하기

## 파이썬 설치

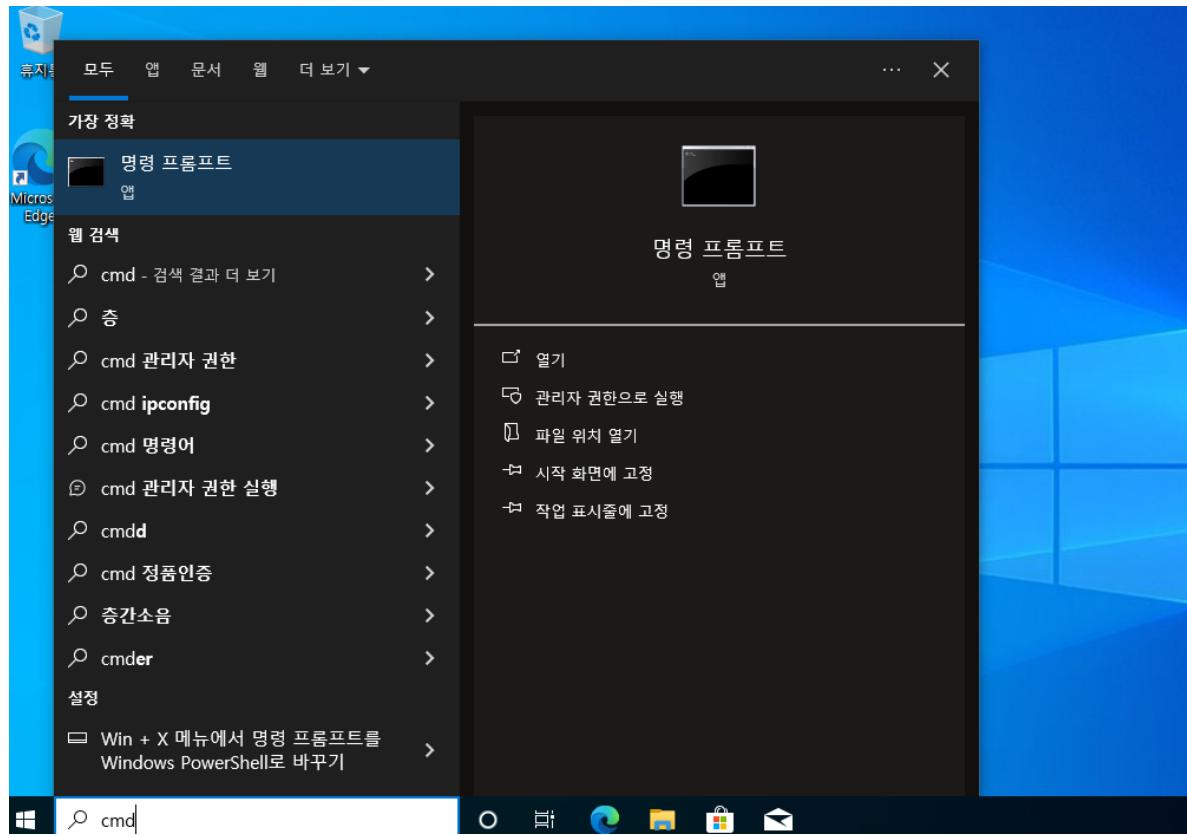
- 설치가 끝나면 다음과 같은 화면을 볼 수 있다.



# 1. 실습환경 구성하기

## 파이썬 설치 확인

- 원도우 하단바에서 cmd로 검색을 하여 명령 프롬프트를 실행한다.



# 1. 실습환경 구성하기

## 파이썬 설치 확인

- 원도우 하단바에서 cmd로 검색을 하여 명령 프롬프트를 실행한다.

...>cd c:\DEV\envs

c:\DEV\envs>python --version

c:\DEV\envs>pip install pip --upgrade

```
명령 프롬프트 - X

::#DEV\envs>python --version
python 3.10.11

::#DEV\envs>pip install pip --upgrade
Requirement already satisfied: pip in c:#users#k8s#appdata#local#programs#python#python310#lib#site-packages (23.0.1)
Collecting pip
  Using cached pip-25.0.1-py3-none-any.whl (1.8 MB)
ERROR: To modify pip, please run the following command:
::#Users#k8s#AppData#Local#Programs#Python#Python310#python.exe -m pip install pip --upgrade

[notice] A new release of pip is available: 23.0.1 => 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
::#DEV\envs>
```



# 1. 실습환경 구성하기

## Jupyter Notebook

- 주피터 노트북을 설치하기 위해 다음의 코드를 cmd 창에 입력한다.

```
c:\DEV\envs>pip install jupyter
```

```
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting ipywidgets
  Downloading ipywidgets-8.1.1-py3-none-any.whl (139 kB)
----- 139.4/139.4 kB 4.2 MB/s eta 0:00:00
...
```

```
ngs-0.5.1 websocket-client-1.6.4 widgetsnbextension-4.0.9
```

```
[notice] A new release of pip is available: 23.0.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
c:\DEV\envs>python.exe -m pip install --upgrade pip
```

```
Requirement already satisfied: pip in c:\users\k8s\appdata\local\programs\python\python310\lib\site-packages (23.0.1)
Collecting pip
  Using cached pip-23.3.1-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.0.1
    Uninstalling pip-23.0.1:
      Successfully uninstalled pip-23.0.1
Successfully installed pip-23.3.1
```

```
c:\DEV\envs>
```



# 1. 실습환경 구성하기

## Jupyter Notebook

- python.exe -m pip를 사용하는 것은 특정 Python 버전 및 환경에 대한 명시적인 관리를 의미. 이 방식을 사용하면 특정 가상환경의 pip를 업그레이드하거나 패키지를 설치할 수 있다.
- 반면 pip install pip는 시스템 전체에서 pip를 업그레이드하므로, 시스템 전역에 영향을 미침.



# 1. 실습환경 구성하기

## Jupyter Notebook 실행

- 작업 폴더로 이동 후 명령어를 입력한다.

```
c:\DEV\envs>cd ..  
C:\DEV>jupyter notebook  
[I 2023-11-28 09:43:53.062 ServerApp] Package notebook took 0.0001s to import  
[I 2023-11-28 09:43:53.240 ServerApp] Package jupyter_lsp took 0.1706s to import
```

...

```
[C 2023-11-28 09:43:55.285 ServerApp]
```

To access the server, open this file in a browser:  
<file:///C:/Users/k8s/AppData/Roaming/jupyter/runtime/jpserver-3796-open.html>

Or copy and paste one of these URLs:

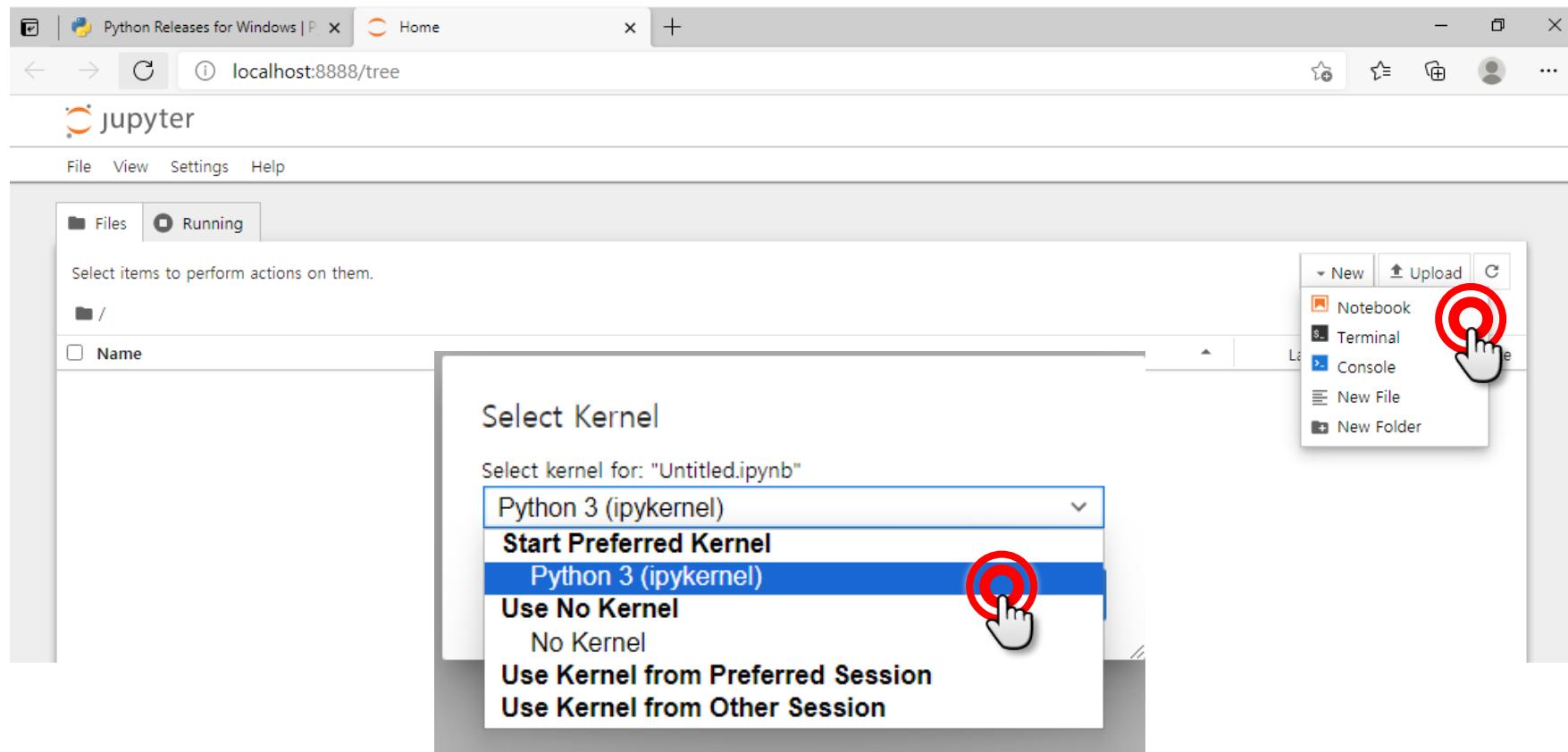
<http://localhost:8888/tree?token=aa659f87c01c17fed8e2a95ff41a0e035bd3bbac199a09be>  
<http://127.0.0.1:8888/tree?token=aa659f87c01c17fed8e2a95ff41a0e035bd3bbac199a09be>

```
[I 2023-11-28 09:44:07.113 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs,  
javascript-typescript-langserver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-  
languageserver, sql-language-server, texlab, typescript-language-server, unified-language-server, vscode-css-languageserver-bin,  
vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
```

# 1. 실습환경 구성하기

## Jupyter Notebook 실행

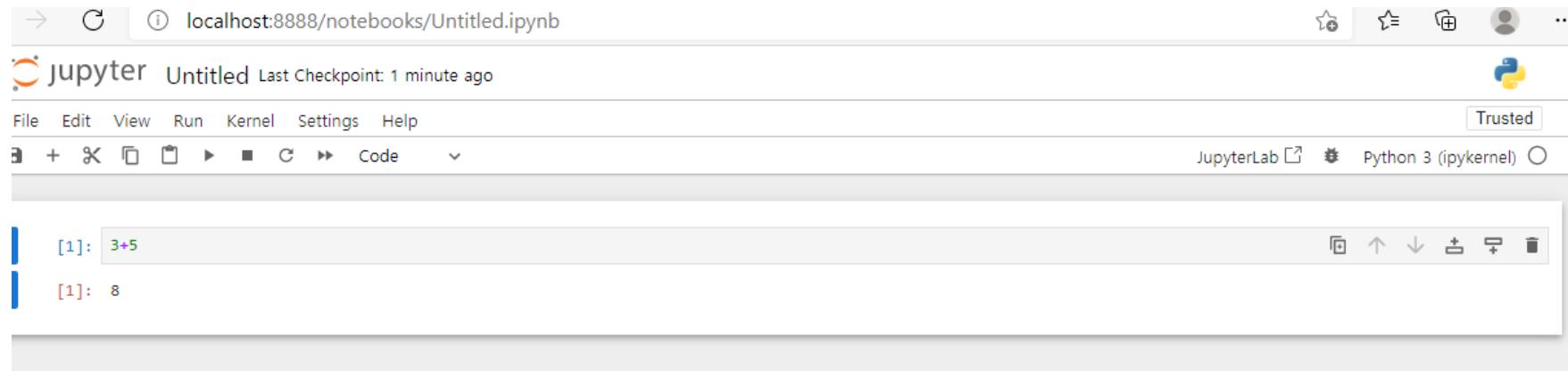
- 브라우저에서 확인한 주피터 노트북 실행 화면. Notebook을 클릭 한다. 커널 선택



# 1. 실습환경 구성하기

## Jupyter Notebook 실행

- Jupyter Notebook 확인



The screenshot shows a web browser window displaying a Jupyter Notebook at the URL `localhost:8888/notebooks/Untitled.ipynb`. The browser's address bar and various control icons are visible at the top. The main content area of the browser displays the Jupyter interface. At the top of the notebook, it says "jupyter Untitled Last Checkpoint: 1 minute ago". Below this is a menu bar with File, Edit, View, Run, Kernel, Settings, and Help. To the right of the menu is a "Trusted" badge with a Python logo. The notebook itself has one cell, labeled [1], containing the code `3+5`. The output of this cell is the number `8`, displayed in red text. There are also standard notebook controls like cell selection, run, and refresh buttons.



# 1. 실습환경 구성하기

## 명령모드와 편집모드, 중요 키보드 단축키, 팁

주피터 노트북은 아무 때나 Esc 키를 누르면 명령모드(command mode)로 들어가고 Enter 키나 셀을 클릭하면 편집모드(Edit mode)로 들어가게 된다.

### ● 공통

- Shift + Enter : run cell, select below
- Ctrl + Enter : run selected cells
- Esc : Enter command mode
- Enter : Enter edit mode
- ⌘ + Shift + P : Show command palette

### ● 명령모드 (grey cell border with a blue left margin)

- Y / M : Change cell type to code / markdown
- Shift + ↑/↓ : Extend cell selections
- Shift + M : Merge selected cells, or merge the current cell with the one below
- D,D : delete selected cells
- C / X / V : copy / cut / paste below
- A / B : add cell above / below

### ● 편집모드 (green cell border)

- Tab : Code completion
- Shift + Tab : Python tooltip
- Shift + Tab (twice) : Detailed Python tooltip



# 1. 실습환경 구성하기

## 명령어 이용

### shell(명령 프롬프트)의 이용

- 명령창에서 쓰는 명령을 그대로 쓰되, 맨 앞에 !를 입력하여 사용 가능하다.
- !dir

### 매직 명령어 이용

- 맨 앞에 %를 붙이고 특정 명령을 수행할 수 있다. 이는 파이썬 문법에는 포함되지 않은, Jupyter notebook만의 기능이다.

매직 명령어	설명
%pwd	현재 디렉토리 경로 출력
%ls	윈도우의 dir, Linux의 ls 명령과 같음
...	...
%matplotlib	그래프를 그리는 코드 위에 따로 설정한다. %matplotlib inline으로 설정하면 코드 셀의 바로 아래에, %matplotlib tk로 설정하면 별도 창에 그래프가 출력된다. %matplotlib notebook으로 하면 코드 셀 바로 아래에 동적으로 그래프를 조작 할 수 있는 그래프가 생성된다.



# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

c:\DEV\envs>에 py3\_10\_basic 가상환경을 생성해 본다.

```
c:\DEV\envs>python -m venv py3_10_basic  
c:\DEV\envs>dir py3_10_basic
```

C 드라이브의 볼륨에는 이름이 없습니다.  
볼륨 일련 번호: BED0-C858

c:\DEV\envs\py3\_10\_basic 디렉터리

```
2024-06-22 오후 06:20 <DIR> .  
2024-06-22 오후 06:20 <DIR> ..  
2024-06-22 오후 06:20 <DIR> Include  
2024-06-22 오후 06:20 <DIR> Lib  
2024-06-22 오후 06:20 118 pyvenv.cfg  
2024-06-22 오후 06:20 <DIR> Scripts  
    1개 파일 118 바이트  
    5개 디렉터리 68,937,621,504 바이트 남음
```

c:\DEV\envs>

PC > 로컬 디스크 (C:) > DEV > py3_10_basic >	
이름	수정한 날짜
Include	2024-06-22 오후 6:2
Lib	2024-06-22 오후 6:2
Scripts	2024-06-22 오후 6:2
pyvenv.cfg	2024-06-22 오후 6:2



# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

- 가상환경을 적용시키기 위해 스크립트를 실행 해야 한다. Scripts 폴더안에 activate.bat라는 파일을 실행 시켜야 한다.

```
c:\DEV\envs>cd py3_10_basic
```

```
c:\DEV\envs\py3_10_basic>dir/w/p
```

C 드라이브의 볼륨에는 이름이 없습니다.  
볼륨 일련 번호: BED0-C858

```
c:\DEV\envs\py3_10_basic 디렉터리
```

```
[.]      [..]      [Include]    [Lib]      pyvenv.cfg  [Scripts]  
      1개 파일          118 바이트  
      5개 디렉터리  68,936,171,520 바이트 남음
```

```
c:\DEV\envs\py3_10_basic>Scripts\activate.bat
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>
```



# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>python
```

```
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> 3+5
```

```
8
```

```
>>> quit()
```

가상환경에서 빠져 나오고 싶다면 deactivate 를 입력하면 된다.

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>deactivate
```

```
c:\DEV\envs\py3_10_basic>
```



# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

### ● 라이브러리 확인 및 설치

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>python
```

```
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import pandas as pd
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
ModuleNotFoundError: No module named 'pandas'
```

```
>>> quit()
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>copy con requirements.txt
```

```
pandas
```

```
1 file(s) copied.
```

Enter  
Ctrl+C



```
(py3_10_basic) c:\DEV\envs\py3_10_basic>dir/w/p
```

```
Volume in drive C is Windows
```

```
Volume Serial Number is 24A6-A8B5
```

```
Directory of c:\DEV\envs\py3_10_basic
```

[.]	[..]	[Include]	[Lib]	pyvenv.cfg	requirements.txt
[Scripts]					
2 File(s)		131 bytes			
5 Dir(s)		82,227,224,576 bytes free			



# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

- 라이브러리 확인 및 설치

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>type requirements.txt  
pandas
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>pip install -r requirements.txt
```

```
Collecting pandas
```

```
  Downloading pandas-2.1.3-cp310-cp310-win_amd64.whl (10.7 MB)  
----- 10.7/10.7 MB 20.5 MB/s eta 0:00:00
```

```
Collecting pytz>=2020.1
```

```
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)  
----- 502.5/502.5 kB ? eta 0:00:00
```

```
...
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>python
```

```
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import pandas as pd
```

```
>>> quit()
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>
```



# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

- Jupyter Notebook에 내보내기

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>pip install ipykernel
```

```
Collecting ipykernel
  Using cached ipykernel-6.27.1-py3-none-any.whl (114 kB)
Collecting nest-asyncio
  Using cached nest_asyncio-1.5.8-py3-none-any.whl (5.3 kB)
Collecting jupyter-core!=5.0.*,>=4.12
  Using cached jupyter_core-5.5.0-py3-none-any.whl (28 kB)
```

...

```
width-0.2.12
```

```
[notice] A new release of pip is available: 23.0.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>python.exe -m pip install --upgrade pip
```

```
Requirement already satisfied: pip in c:\DEV\envs\py3_10_basic\lib\site-packages (23.0.1)
Collecting pip
  Using cached pip-23.3.1-py3-none-any.whl (2.1 MB)
```

...

```
(py3_10_basic) c:\DEV\envs\py3_10_basic>python -m ipykernel install --user --name
py3_10_basic --display-name "py3_10_basic"
```

```
Installed kernelspec py3_10_basic in C:\Users\k8s\AppData\Roaming\jupyter\kernels\py3_10_basic
```



# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

- Jupyter Notebook에 내보내기

```
(py3_10_basic) c:\DEV\envs\py3_10_basic> deactivate
```

```
c:\DEV\envs\py3_10_basic>cd ..
```

```
c:\DEV\envs>dir/w/p
```

```
C 드라이브의 볼륨에는 이름이 없습니다.  
볼륨 일련 번호: BED0-C858
```

```
c:\DEV\envs 디렉터리
```

```
[.]           [...]          [.ipynb_checkpoints] [py3_10_basic]      Untitled.ipynb  
    1개 파일          794 바이트  
4개 디렉터리  30,615,150,592 바이트 남음
```

```
c:\DEV\envs>cd ..
```

```
C:\DEV>jupyter notebook
```

# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

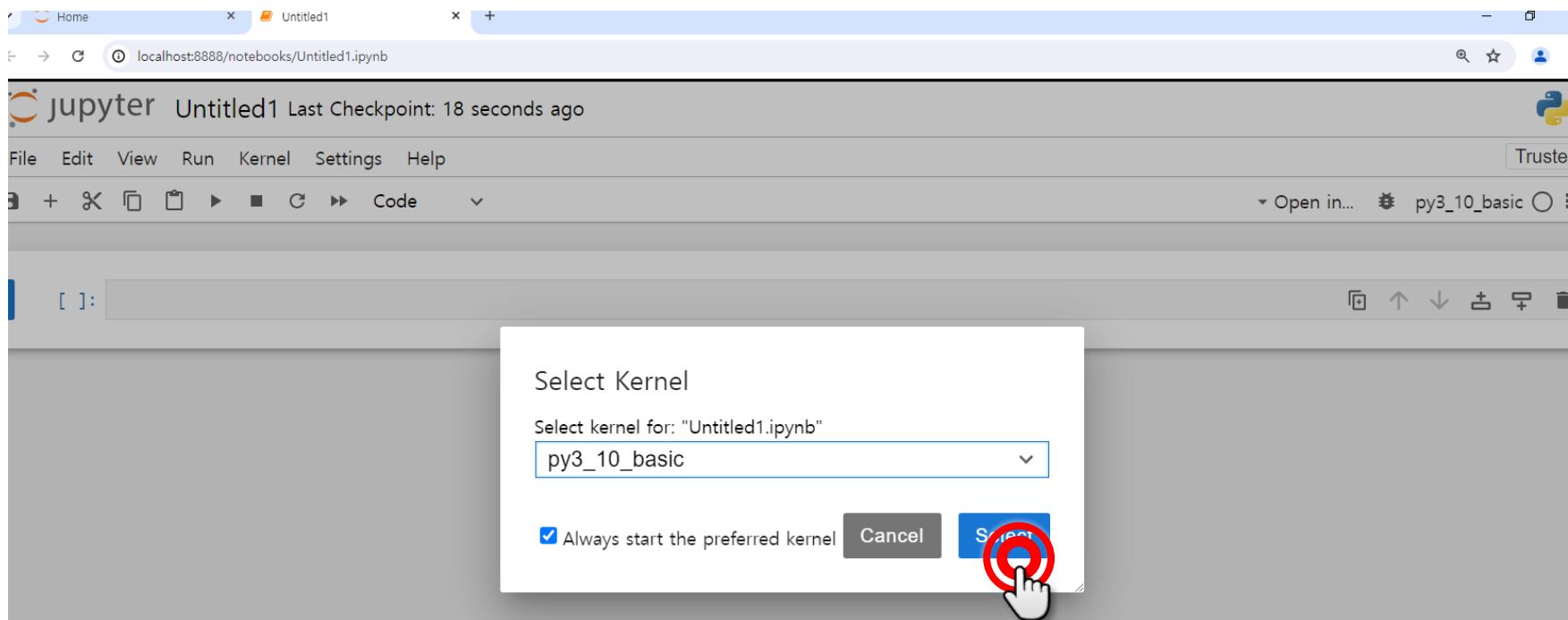
- Jupyter Notebook에 내보내기



# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

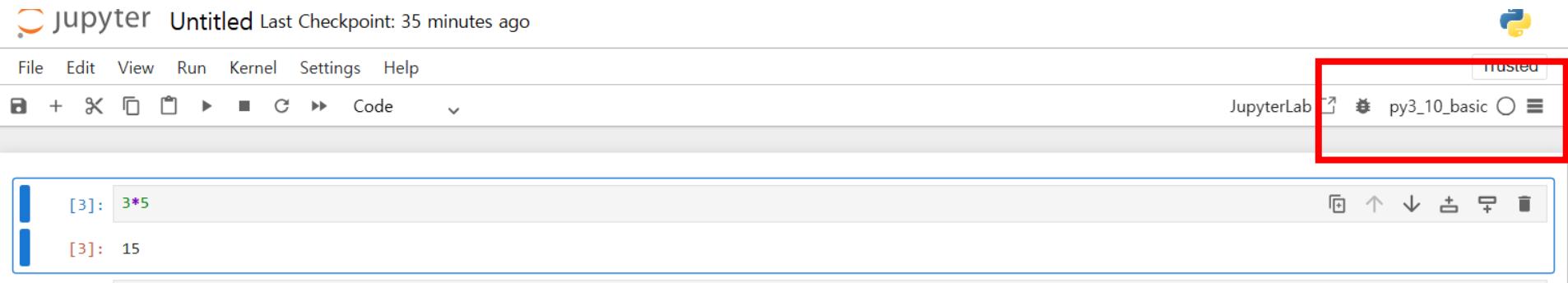
- Jupyter Notebook에 내보내기
- 새로 생긴 가상환경 커널을 선택한다.



# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 가상환경

- Jupyter Notebook에 py3\_10\_basic가 선택되어 있다.





# 1. 실습환경 구성하기

## 파이썬 패키지 관리와 설치

- 파이썬 패키지 관리 툴인 pip를 사용하면 된다. 즉 pip list 를 사용한다.

### pip list

```
%matplotlib inline
```

```
py3_10_basic 가상환경으로 change kernel
```

```
pip list
```

Package	Version
asttokens	3.0.0
colorama	0.4.6
comm	0.2.2
contourpy	1.3.2
cycler	0.12.1
debugpy	1.8.14
decorator	5.2.1



# 1. 실습환경 구성하기

팁:

- 슈퍼유저로 sudo pip install 실행하지 말기!

(유닉스/맥 전문가에게만 적용되는 이야기입니다) 아나콘다 환경에서 패키지를 설치할 때 조심할 것은 절대 슈퍼유저 권한으로 sudo pip install 을 실행하지 말라는 것이다. 아나콘다는 운영체제의 파일을 최대한 건들지 않고 깔끔하게 사용자 디렉토리 (Users/\$username/anaconda2/)만을 사용하는 것을 지향하기 때문이다.

- 슈퍼유저로 sudo pip install 실행하기!



# 1. 실습환경 구성하기

| **팁:**

1. 'c:\DEV>rmdir /s py3\_10\_basic  
py3\_10\_basic, 계속하시겠습니까(Y/N)? y



# 1. 실습환경 구성하기

| **팁:**

1. 'ipykernel' 라이브러리 다운로드

`pip install ipykernel`

2. 커널 추가

`python -m ipykernel install --user --name = 가상환경이름`

3. 제거하기

`jupyter kernelspec uninstall 가상환경이름`

| **예 :**

`c:\DEV>jupyter kernelspec uninstall py3_10_basic`

Kernel specs to remove:

  py3\_10\_basic      C:\Users\k8s\AppData\Roaming\jupyter\kernels\py3\_10\_basic

Remove 1 kernel specs [y/N]: y

Removed C:\Users\k8s\AppData\Roaming\jupyter\kernels\py3\_10\_basic

## 2. 몇가지 라이브러리 설치

### 파이썬 라이브러리 설치와 확인

```
pip install matplotlib seaborn scikit-learn
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic> pip install matplotlib seaborn scikit-learn
```

```
...  
Collecting matplotlib
```

```
...  
  
Ictl-3.5.0 tzdata-2024.1
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic> python.exe -m pip install --upgrade pip
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic> python -m pip install scipy
```

```
Collecting scipy
```

```
  Using cached scipy-1.14.0-cp310-cp310-win_amd64.whl.metadata (60 kB)
```

```
Collecting numpy<2.3,>=1.23.5 (from scipy)
```

```
  Using cached numpy-2.0.0-cp310-cp310-win_amd64.whl.metadata (60 kB)
```

```
(py3_10_basic) c:\DEV\envs\py3_10_basic> deactivate
```

```
c:\DEV\envs\py3_10_basic> cd ..
```

```
c:\DEV\envs>cd..
```

```
c:\DEV>jupyter notebook
```

## 2. 몇가지 라이브러리 설치

### 파이썬 라이브러리 설치와 확인

```
import sys
print("python 버전 : {}".format(sys.version))
import numpy as np
print("numpy 버전 : {}".format(np.__version__))
import pandas as pd
print("pandas 버전 : {}".format(pd.__version__))
import matplotlib
print("matplotlib 버전 : {}".format(matplotlib.__version__))
import scipy as sp
print("scipy 버전 : {}".format(sp.__version__))
import IPython
print("IPython 버전 : {}".format(IPython.__version__))
import sklearn
print("sklearn : {}".format(sklearn.__version__))
```

```
python 버전 : 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)]
numpy 버전 : 2.2.5
pandas 버전 : 2.2.3
matplotlib 버전 : 3.10.1
scipy 버전 : 1.15.2
IPython 버전 : 8.35.0
sklearn : 1.6.1
```



# 참고

## 문헌

1. <http://www.ncs.go.kr>
2. NELLDAL/E/JOHN LEWIS지음, 조영석/김대경/박찬영/송창근 역, 단계별로배우는컴퓨터과학, 홍릉과학출판사,2018
3. David A. Watt, Programming Language Syntax and Semantics, Prentice-Hall, 1991.
4. 혼자 공부하는 머신러닝+딥러닝 박해선 지음 | 한빛미디어 | 2020년 12월
5. 머신러닝 실무 프로젝트 ,아리가 미치아키, 나카야마 신타, 니시바야시 다카시 지음 | 심효섭 옮김 | 한빛미디어 | 2018년 06월
6. 파이썬을 활용한 머신러닝 쿡북 크리스 알본 지음 | 박해선 옮김 | 한빛미디어 | 2019년 09월
7. 처음 배우는 머신러닝 김의중 지음 | 위키북스 | 2016년 07월
8. 파이썬으로 배우는 머신러닝의 교과서 : 이토 마코토 지음 | 박광수(아크몬드) 옮김 | 한빛미디어 | 2018년 11월
9. <https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/>
10. 기타 서적 및 웹 사이트 자료 다수 참조



# 참고

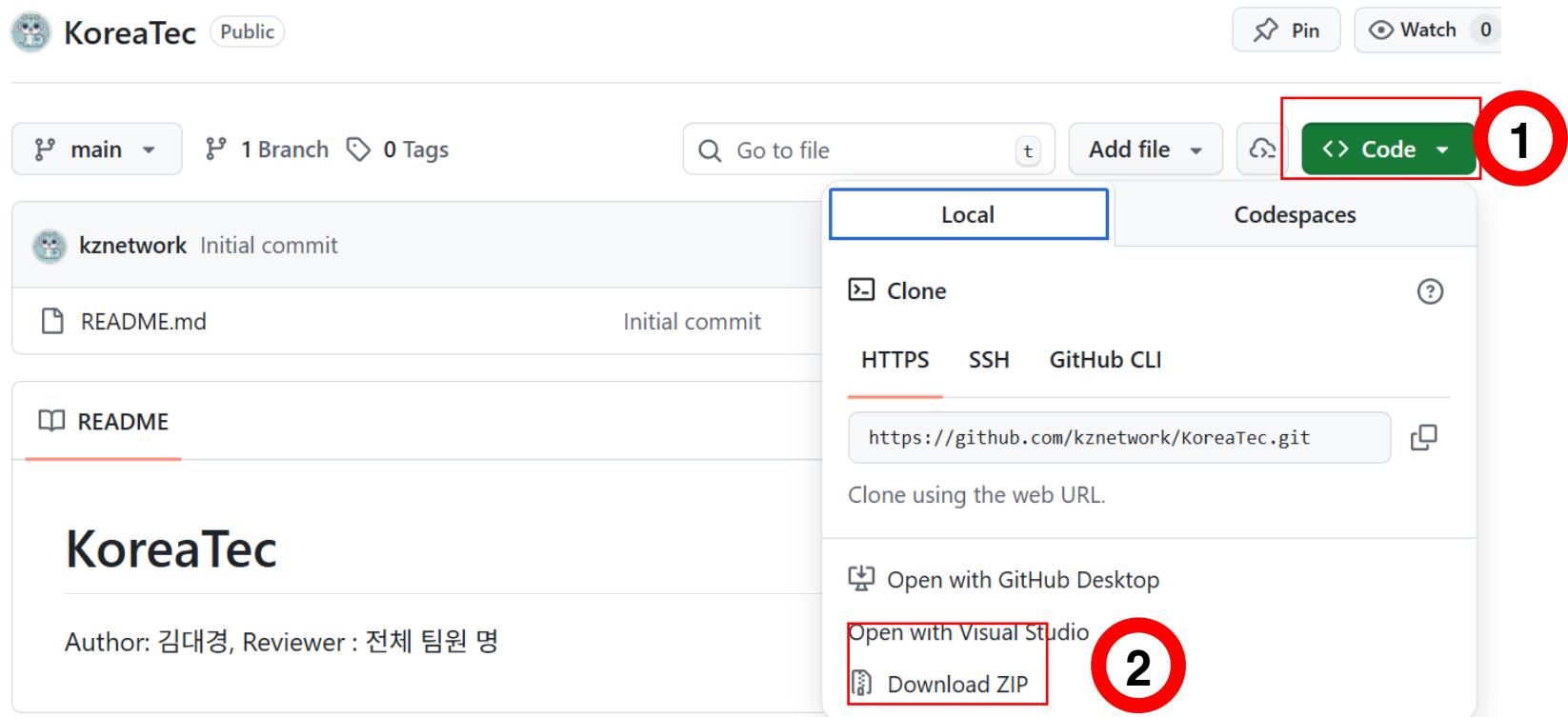
## 문헌

1. <https://a16z.com/emerging-architectures-for-lm-applications/>
2. [https://mattboegner.com/knowledge-retrieval-architecture-for-llms/?fbclid=IwAR36f7aHhYhKFcuSnNMVhJEmGmIEJ8b8r2wbXU2cK\\_9kKAFuV7qPI8dxmTA](https://mattboegner.com/knowledge-retrieval-architecture-for-llms/?fbclid=IwAR36f7aHhYhKFcuSnNMVhJEmGmIEJ8b8r2wbXU2cK_9kKAFuV7qPI8dxmTA)
3. <https://huyenchip.com/2023/04/11/lm-engineering.html?fbclid=IwAR3CwobE7Ia9IgGDR9SqWmDtmhRRkO6jv1yCNtZEPFstX1rqx9nkrEYIRIQ>

# 참고

## 강의 소스 다운로드

1. <https://github.com/kznetwork/KoreaTec>에서 강의 소스를 다운받을 수 있으나, 사전 예고없이 변경될 수 있음.





감사합니다.

❖ Mobile: 010-9591-1401  
❖ E-mail: onlooker2zip@naver.com