AI기반데이터 분석 및 AI Agent 개발 과정

Pre-Requisites_실습환경 구성하기

2025.09.22-10.02(9일, 62시간)

Prepared by DaeKyeong

Ph.D.

0



교육 시간표

117L	4 OI +I	201+1	201+1	401+1	F01+I	Total
시간	1일차	2일차 : 3장 데이터 전처리 :	3일차	4일차	5일차 실습28 : 의사 결정 나무	lotai
1교시 08시30분 - 09시20분	1과목 A I기반 데이터 분석 1장 생성형 AI와 데이터 분석	- 분석 주제 탐색 및 문제해결 단계별 접근	. 5장 통계적 가설 검정 및 분석	7장 상관관계 및 연관성 이해 변수 간의 관계 연관성 분석	일급28 : 의사 결정 나무 실습29 : 랜덤 포레스트 실습30 : 타기팅 분석 실습31 : LightGBM으로 모델링하기	
2교시 09시30분 - 10시20분	생성형 Al	실습 6: 분석 주제 탐색 및 문제해결 단계별 접근 실습 7: 주제를 평가하는 새로운 관점 실습 8: 보고서 작성하기	- 통계 이해:기술통계와 추론통계	실습24 : 상관 분석-1 실습25 : 상관 분석-2	10장 머신러닝 기반 데이터 분석-비지도	
3교시 10시30분 - 11시50분	- 실습1 : GPT 빌더로 '홍보기사 작성기 ' 챗봇 만들기 - 실습2 : Data Analyst 사용해보기 - 실습3 : ChatGPT를 활용한 데이터 분 석	- 데이터 확인 및 검증	실습17 : 데이터 특성 파악하기 실습18 : 데이터 샘플링	8장 인과 관계 및 예측 분석 이해	- 실습32 : 군집 분석 - 실습33 : 군집 분석과 LDA 알고리즘	
4교시 13시00분 - 13시50분	- 실습환경 구성하기 - Pre-Requisites_실습환경 구성하기	실습 9: Data Analyst 를 활용한 데이터 확인 실습10: 데이터 시각화 활용 실습11: ChatGPT를이용한 시각화 분석	통계적 가설 검정 A/B테스트	회귀분석	11장 기타 데이터 마이닝 - Association Rule Analysis	
5교시 14시00분 - 14시50분	2장 조사 및 데이터 수집 방법 - 조사란? - 실습1 : ChatGPT를 이용한 시장조사	- 결측값 - 데이터 분포 - 이상치	실습19 : 통계적 가설검정 실습20 : A/B 테스트 분석	실습26 : 회귀 분석-1 실습27 : 회귀 분석-2	12장 텍스트 데이터 분석 텍스트 마이닝 이해 - 텍스트 데이터 분석 이해 - 형태소 분석 - 검색 트렌드 분석 - 텍스트 분류 방법 - 감성 분석 - 연관어 분석	
6교시 15시00분 - 15시50분	3장 데이터 전처리 - 데이터 - 데이터 수집	실습12: ChatGPT를이용한 데이터 전처리 실습13: 결측값 처리하기 실습14: 이상치 처리하기	6장 데이터 준비(Data Preparation) - Data Integration (통합) - Data Reduction (축소) - Data Transformation (변환)	9장 머신러닝 기반 데이터 분석-지도 - 분류 및 회귀 - 트리를 이용한 데이터 분석 - 수치 예측	- 실습34 : 텍스트 데이터 분석 및 텍스 트 마이닝	
7교시 16시00분 - 16시50분		4장 데이터 분석 - 탐색적 데이터 분석 - ChatGPT를 이용한 탐색적 분석	- Feature Engineering & Data Encoding - Cross Validation & Data Splitting - Data Quality Assessment & Model Performance Evaluation			
8교시 17시00분 - 17시15분	- 실습1 : 공공포털 이용 - 실습2-3 : 한글 웹 페이지 크롤링 후 저장 - 실습 4 : Web 스크래핑 - 실습5 : 데이터 인제스트	실습15: ChatGPT를 이용한 탐색적 데이터 분석 실습16 : 범주형 데이터 분석하기	실습21 : 데이터 변환-1 실습22 : 데이터 변환-2 실습23 : 데이터 취합하기			
Total Time	.8			8 :	. 8	3 40 ²

『1-1』실습



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 command 로 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

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/)

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...
- <u>본 강의에서는 윈도 환경에서 아나콘다(Anaconda) 설치 없이</u> <u>파이썬과 주피터 노트북(Jupyter Notebook), VisualStudio</u> Code 을 설치한다.

왜 파이썬인가?

- 복잡한 데이터 가공과 자동화 작업에 인기 있는 언어는 파이썬이다. 파이썬은 많은 회사에서 데이터 과학뿐 아니라 C/C++, Java 등 처럼 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

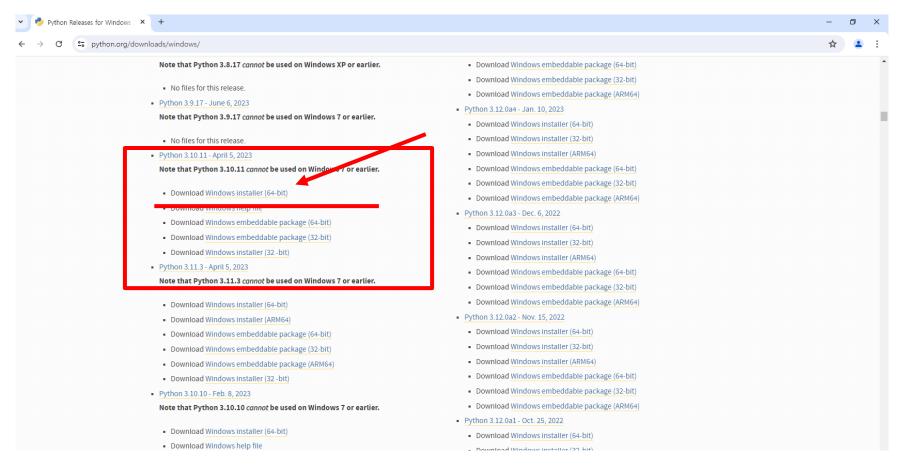
● 아나콘다(Anaconda)가 유료화 조짐을 보이는데 단순히 주피터 노트북만 사용하고자 한다면 가볍게 파이썬과 주피터 노트북을 설치하면 된다. 단, 데이터 분석 등 원하는 라이브러리는 직접 설치해야 한다.

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.

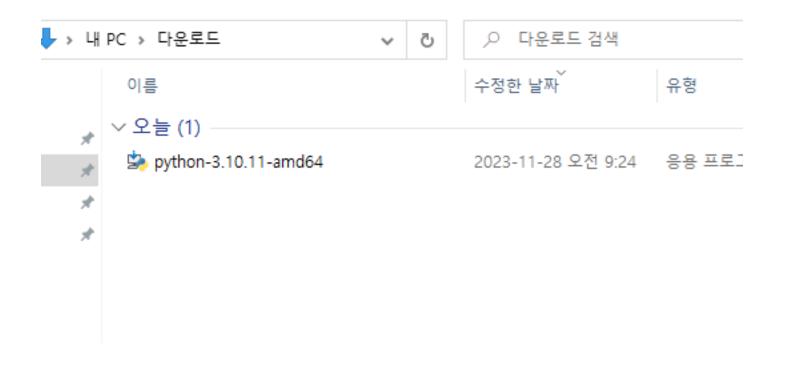
파이썬 다운로드

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



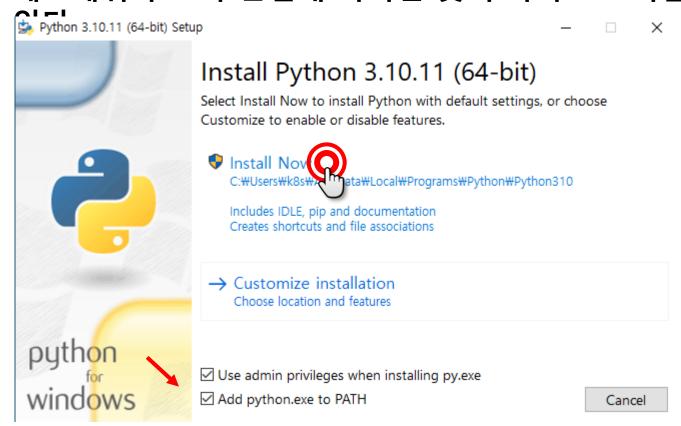
파이썬 다운로드

• python-3.10.11-amd64.exe



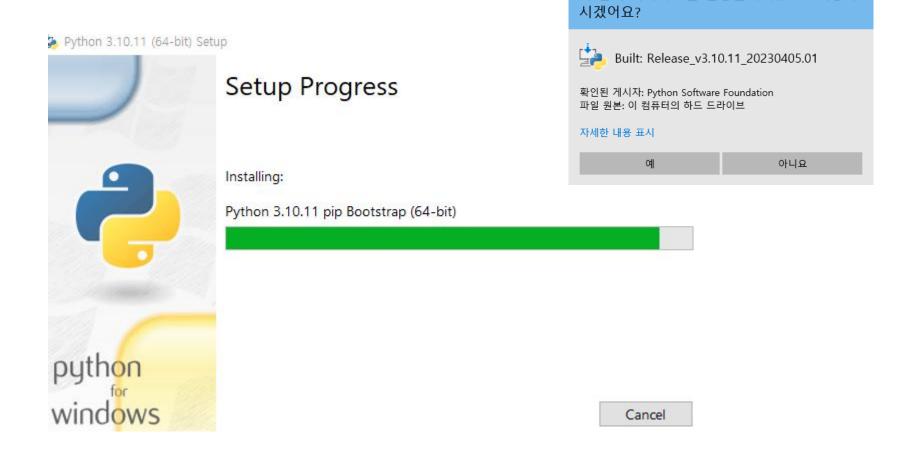
파이썬 설치

● 다운로드 받은 설치파일을 실행하면 다음과 같은 설치 화면을 볼수 있다. 가장 아래쪽에 "Add Python.exe to PATH"를 체크해줘야 보다 손쉽게 파이썬 및 주피터 노트북을 설치할 수



파이썬 설치

● 설치

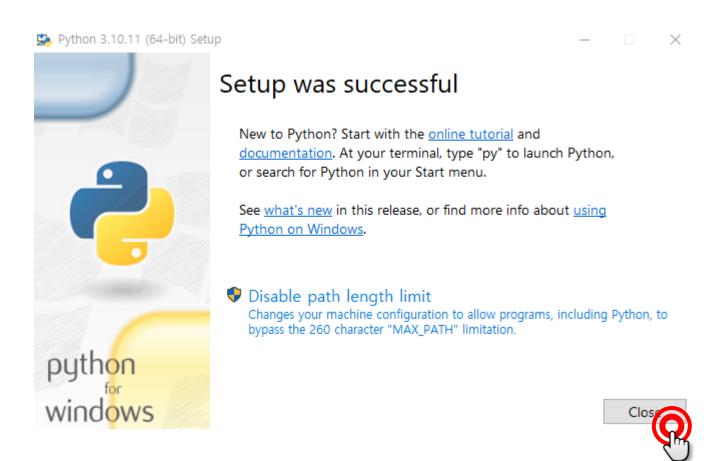


사용자 계정 컨트롤

이 앱이 디바이스를 변경할 수 있도록 허용하

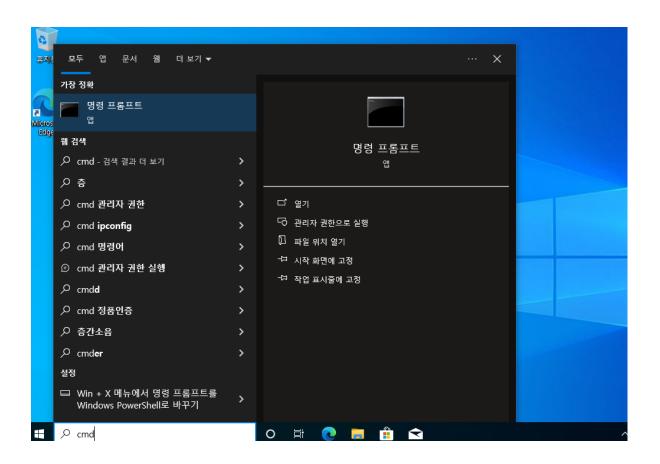
파이썬 설치

● 설치가 끝나면 다음과 같은 화면을 볼 수 있다.



파이썬 설치 확인

● 윈도우 하단바에서 cmd로 검색을 하여 명령 프롬프트를 실행한다.



파이썬 설치 확인

● 윈도우 하단바에서 cmd로 검색을 하여 명령 프롬프트를 실행한다. ...>cd c:\DEV\envs

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

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>

Jupyter Notebook

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

Jupyter Notebook 실행

● 작업 폴더로 이동 후 명령어를 입력한다.

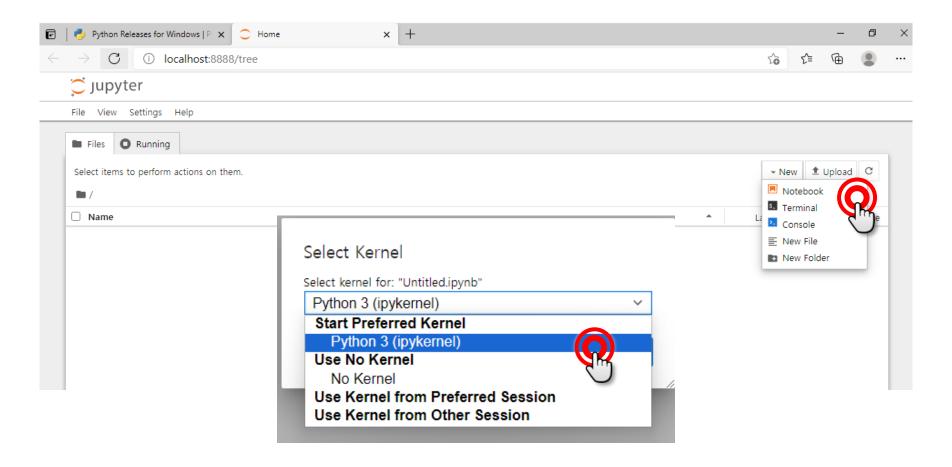
```
C:\DEV>jupyter notebook
[I 2023-11-28 09:43:53.062 ServerApp] Package notebook took 0.0001s to import
[I 2023-11-28 09:43:53.240 ServerApp] Package jupyter_lsp took 0.1706s to import
...

[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, rlanguageserver, sql-language-server, jedi-language-server, julia-language-server, pyright, python-language-server, python-language-server, rvscode-css-languageserver-bin, vscode-html-language-server-bin, vscode-ison-languageserver-bin, vanil-language-server
```

Jupyter Notebook 실행

● 브라우저에서 확인한 주피터 노트북 실행 화면. Notebook을 클릭한다. 커널 선택



Jupyter Notebook 실행

Jupyter Notebook 확인



명령모드와 편집모드, 중요 키보드 단축키, 팁

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

- 공통
- Shift + Enter : run cell, select below
- Ctrl + Enter : run selected cells
- Esc: Enter command mode
- Enter: Enter edit mode
- \Re + 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

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 조작할 수 있는 그래프가 생성된다.

파이썬 패키지 관리와 가상환경

c:\DEV\envs>에 py3_10_basic 가상환경을 생성해 본다.

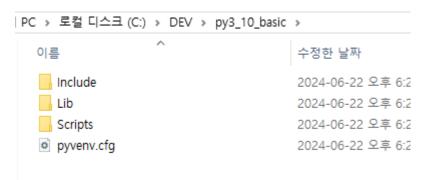
c:\DEV\envs>python -m venv py3_10_basic
c:\DEV\envs>dir py3_10_basic

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>



파이썬 패키지 관리와 가상환경

● 가상환경을 적용시키기 위해 스크립트를 실행 해야 한다. Scripts 폴더안에 activate.bat 라는 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>

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 입력 하면 된다.

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

[...]

2 File(s)

[Scripts]

[Include]

131 bytes

5 Dir(s) 82,227,224,576 bytes free

[Lib]

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.
                                                                              requiremen...
                                              Enter
                                                                           파일(F) 편집(E) 서식(O) 보기(V)
                                             Ctrl+C
                                                                           도움말(H)
                                                                           pandas
(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
```

pyvenv.cfq

requirements.txt

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>
```

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
```

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
```

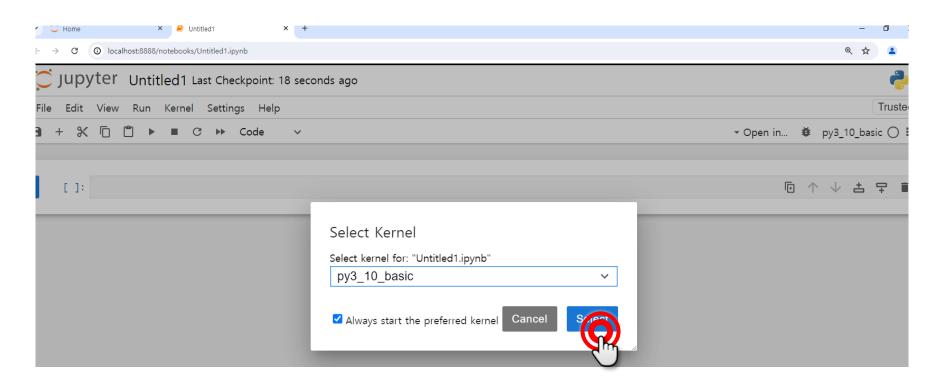
파이썬 패키지 관리와 가상환경

● Jupyter Notebook에 내보내기



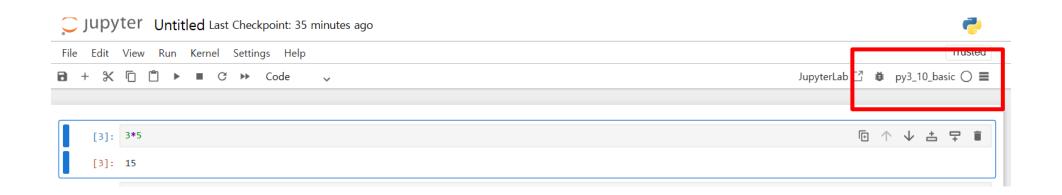
파이썬 패키지 관리와 가상환경

- Jupyter Notebook에 내보내기
- 새로 생긴 가상환경 커널을 선택한다.



파이썬 패키지 관리와 가상환경

• Jupyter Notebook에 py3_10_basic가 선택되어 있다.



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 colorama	3.0.0
comm	0.2.2 1.3.2
cycler debugpy	0.12.1 1.8.14
decorator	5.2.1

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 사용자디렉토리 (Users/\$username/anaconda2/)만을 사용하는 것을 지향하기 때문이다.
- 슈퍼유저로 sudo pip install 실행하기!

팁:

1. `c:\DEV>rmdir /s py3_10_basic py3_10_basic, 계속하시겠습니까(Y/N)? y

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 lctl-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
```

3. Self 점검 문제

Question 1

현재의 작업폴더를 확인해 보자. Windows 운영체제에서 작업폴더를 확인하는 방법은 getcwd() 함수(get current working directory)를 사용하며 관련 코드는 다음과 같다.

Answer 1

import os
os.getcwd()
>>
'c:\\DEV\\'

. , 3. Self 점검 문제

Question 2

상위 폴더로 이동해 보자. 작업폴더 경로를 변경하는 방법은 chdir() 함수(Change the current working directory)를 사용하며 관련 코드는 다음과 같다.

Answer 2

```
os.chdir("c:\DEV\envs\py3_10_basic") os.getcwd()
```

>>

'C:\\DEV\\envs\\py3_10_basic'

. , 3. Self 점검 문제

Question 3

하위 폴더로 이동해 보자. ".." 은 상위 폴더로 이동함을 뜻하며 현재 경로의 하위 폴더로 이동하고자 할 경우 그에 맞는 폴더명 또는 경로를 chdir() 함수에 기입하면 된다.

Answer 3

os.chdir("..")
os.getcwd()

>> 'c:\\DEV\\envs'

3. Self 점검 문제

Question 4 %pwd !dir or %ls

명령어 확인해 보기

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

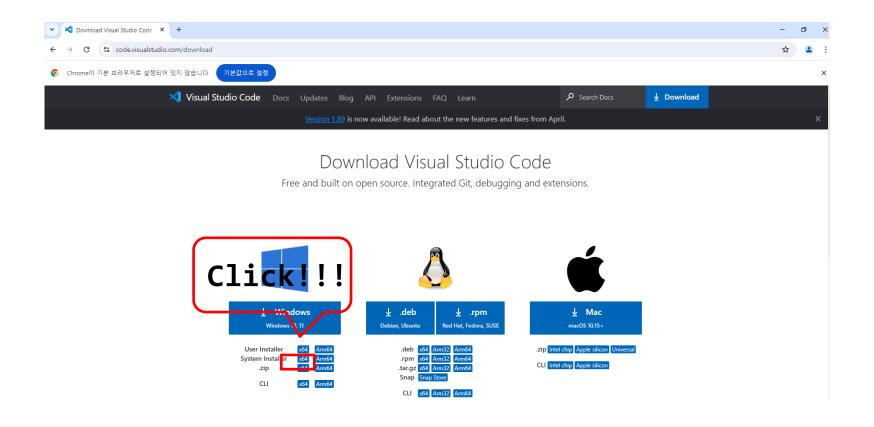
>>
    C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: BEDO-C858

c:\DEV\envs 디렉터리

2025-04-23 오후 04:51 < DIR> ...
2025-04-23 오후 04:51 < DIR> ...
2025-04-23 오후 05:02 < DIR> py3_10_basic 0개 파일 0 바이트 3개 디렉터리 51,199,102,976 바이트 남음
```

Visual Studio Code 다운로드

https://code.visualstudio.com/download



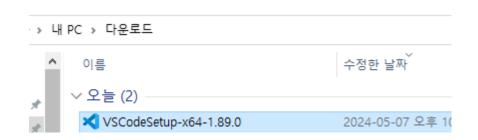
Visual Studio Code 다운로드

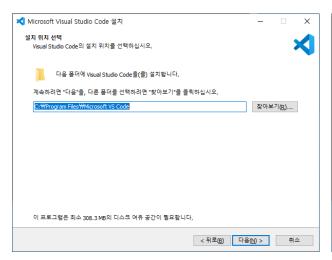
- User Installer, System Installer, zip 압축파일이 배포되고 있습니다.
- User Installer의 경우 다음 위치에 설치되며 유저 인터페이스의 디폴트 언어가 영어가 됩니다.
- C:\Users\사용자이름\AppData\Local\Programs\Microsoft VS Code
- System Installer의 경우에는 다음 위치에 설치되며 유저 인터페이스의 디폴트 언어가 영어가 됩니다.
- C:\Program Files\Microsoft VS Code

●System Installer를 설치

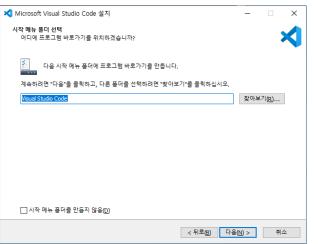
Visual Studio Code 설치

● Visual Studio Code 설치



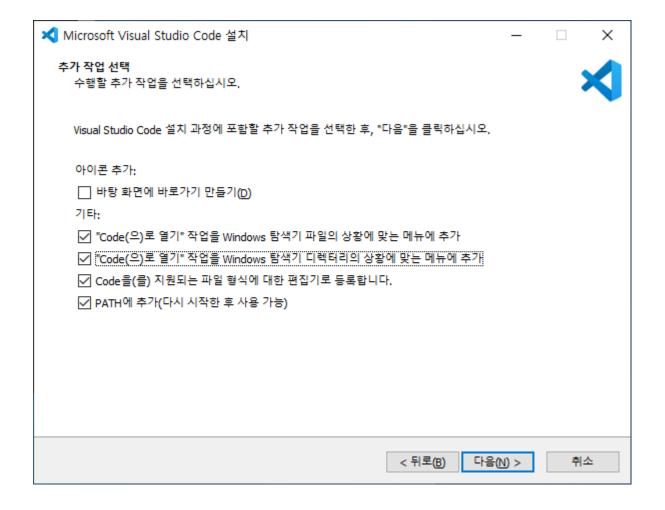






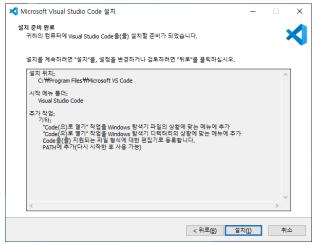
Visual Studio Code 설치

● Visual Studio Code 설치

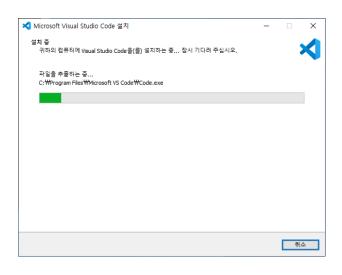


Visual Studio Code 설치

● Visual Studio Code 설치

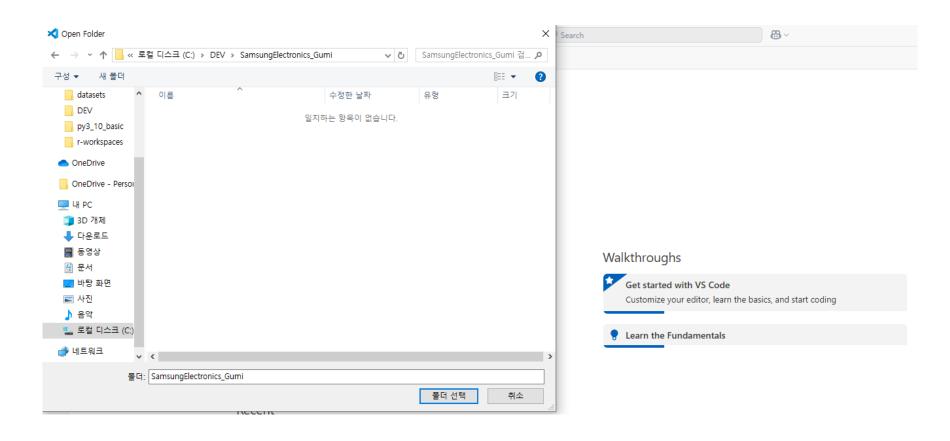






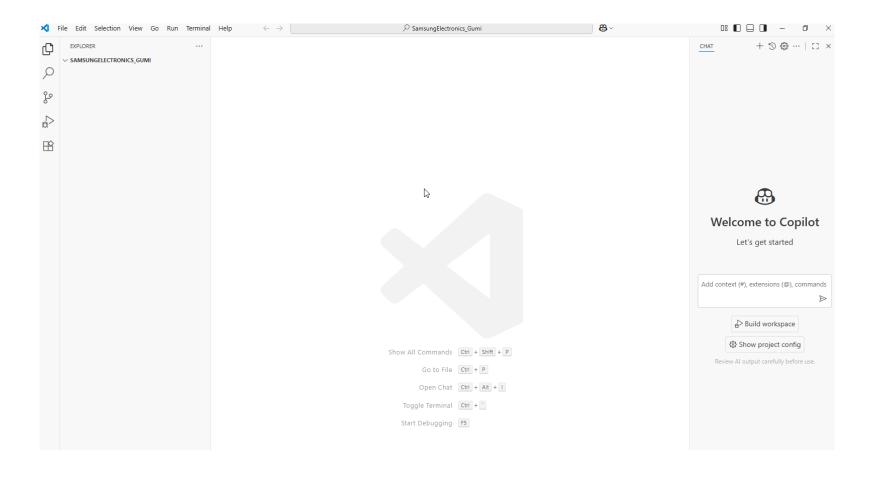
Visual Studio Code 사용

- Visual Studio Code를 실행하여 메뉴에서 File > Open Folder를 선택
- C:\DEV\SamsungElectronics Gumi선택



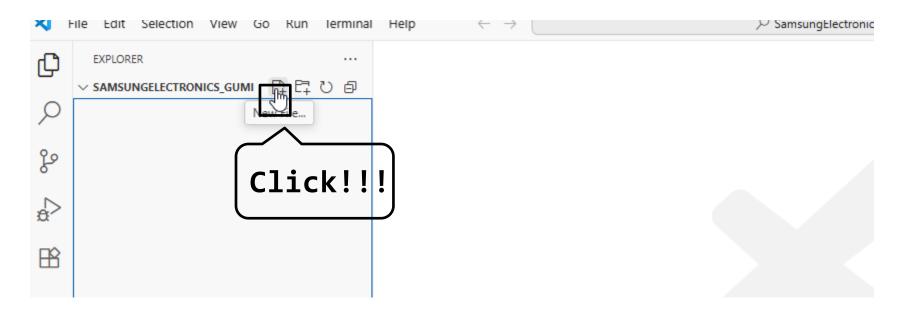
Visual Studio Code 사용

● Visual Studio Code 첫 화면



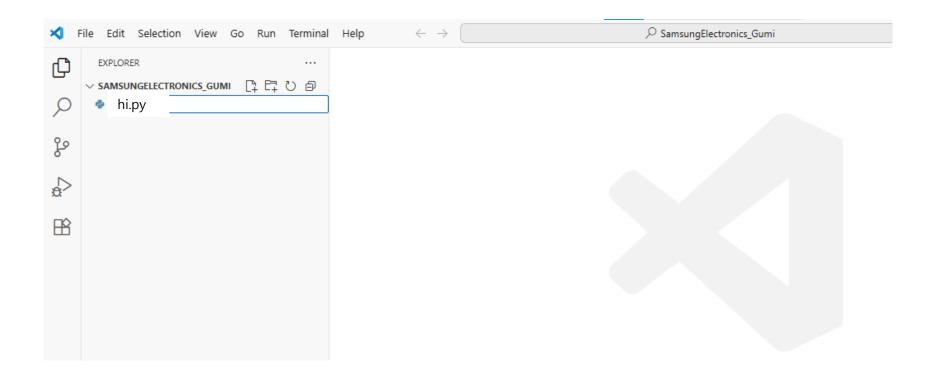
Visual Studio Code 사용

● 해당 폴더가 Visual Studio Code 왼쪽에 보이게 됩니다. New File 아이콘을 클릭



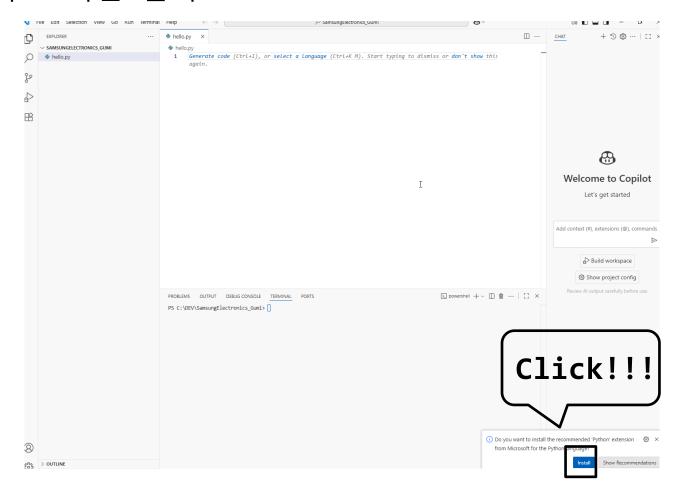
Visual Studio Code 사용

● Hi.py를 입력 후, 엔터를 누르면 해당 파일이 추가되면서 오른쪽에 hi.py 파일이 열리게 됩니다.



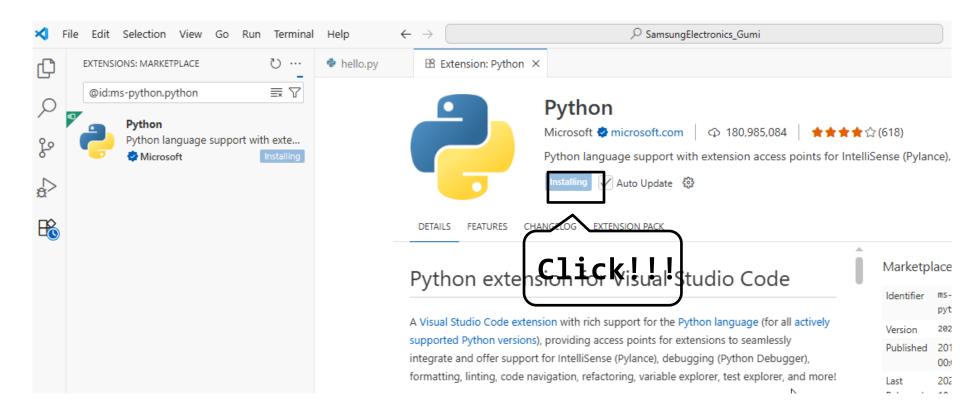
Visual Studio Code 사용

● 오른쪽 아래에 Python 확장을 설치하는지 물어보는 메시지 박스가 보이면 설치



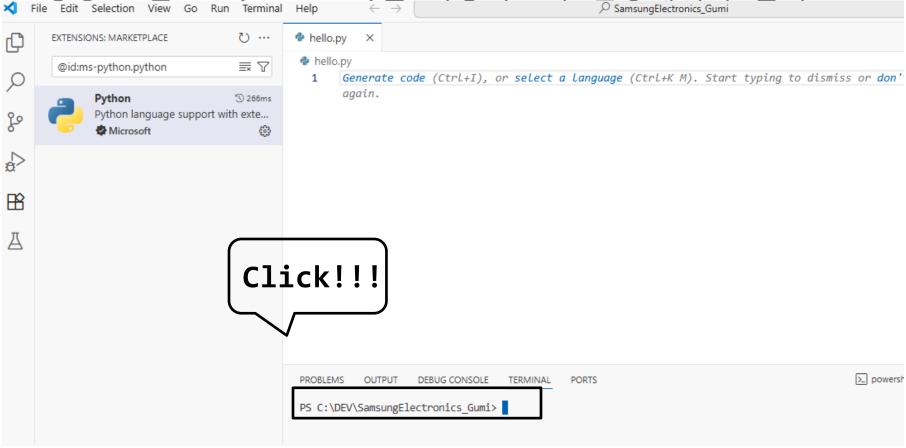
Visual Studio Code 사용

● 오른쪽 아래에 Python 확장을 설치하는지 물어보는 메시지 박스가 보이면 설치



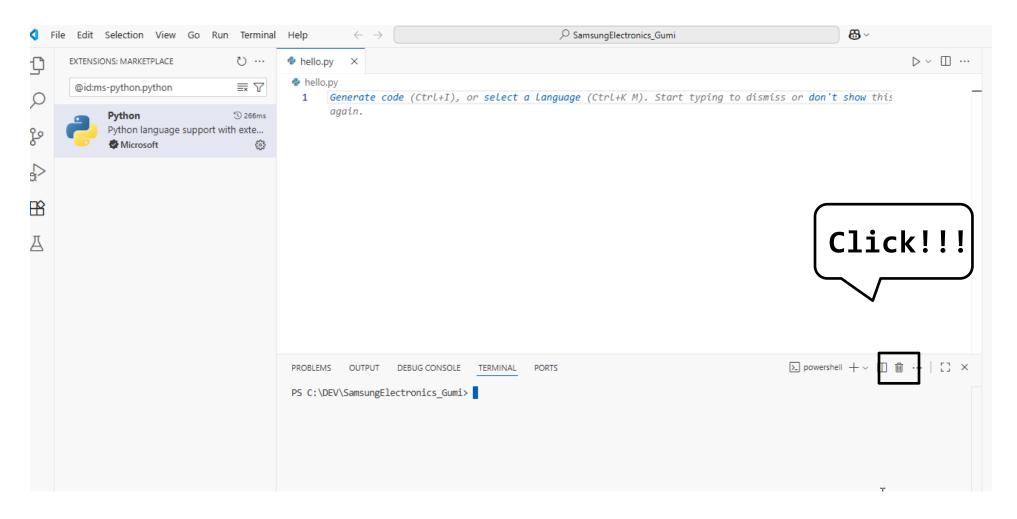
Visual Studio Code 사용

● 메뉴에서 View > Terminal을 선택한다. 다음처럼 PowerShell이 실행되면 가상환경이 적용되지 않기 때문에 명령 프롬프트(cmd.exe)를 사용하도록 변경해줘야 한다.



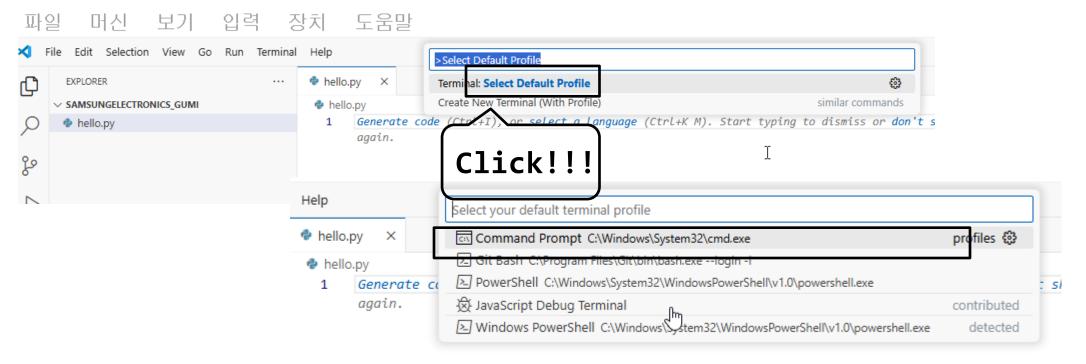
Visual Studio Code 사용

● 오른쪽에 보이는 쓰레기통 아이콘을 클릭



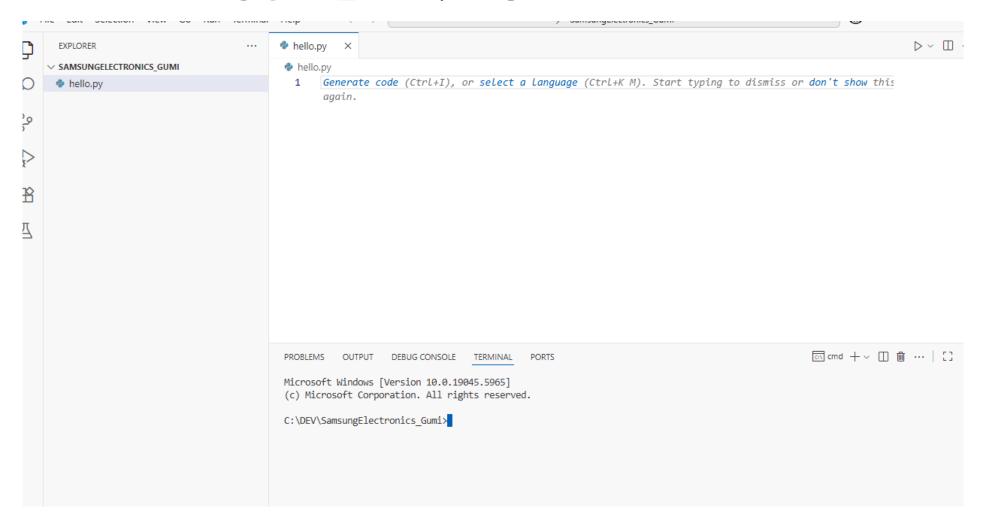
Visual Studio Code 사용

● 가상 환경 설정 시 Visual Studio Code에서 사용하는 터미널(Terminal)의 경우 PowerShell이 기본(Default)로 설정되어있어서, 가상환경 실행이 잘 안 되는 경우가 있으니 [Ctrl + Shift + P] 키를 눌러 나타나는 창에 "Select Default Profile"이라고 기본 터미널 설정을 할 수 있는 메뉴가 나타난다. 선택한다



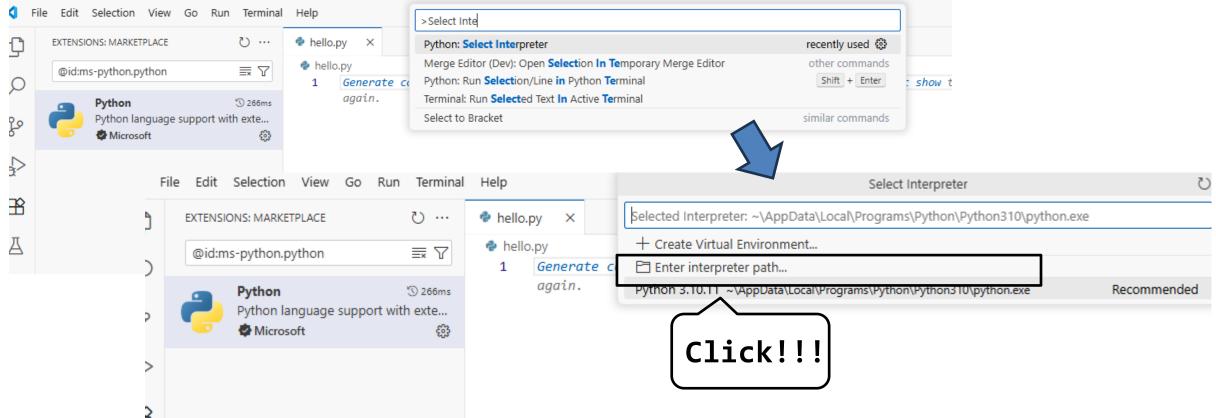
Visual Studio Code 사용

● Ctrl + ` 명령 프롬프트가 실행



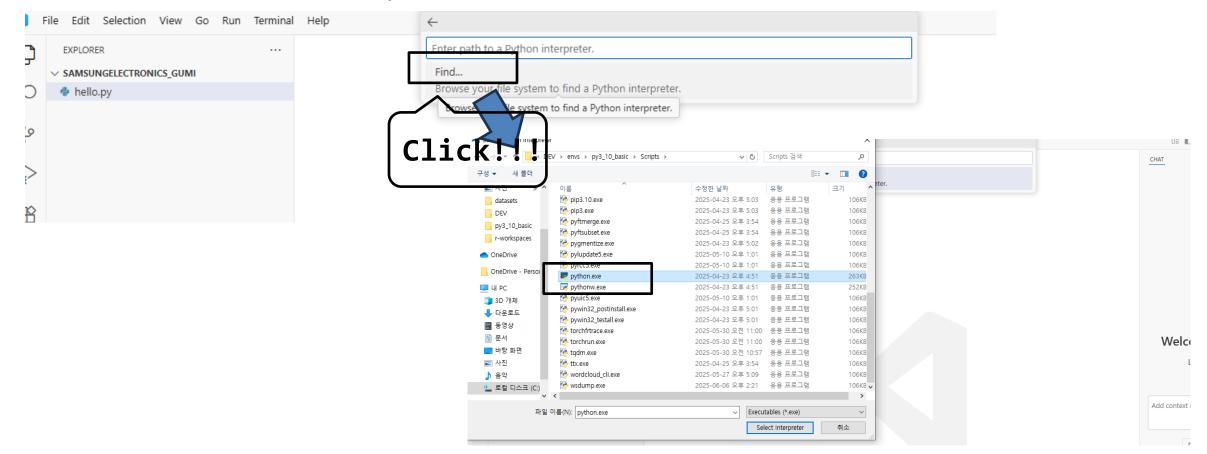
Visual Studio Code 사용

● Python 버전 변경 시, 앞에서 생성한 가상환경의 인터프리터로 변경하기 위해서는 Ctrl + Shift +P를 누른 후, Select interpreter를 입력하여 검색되는 다음 항목을 선택



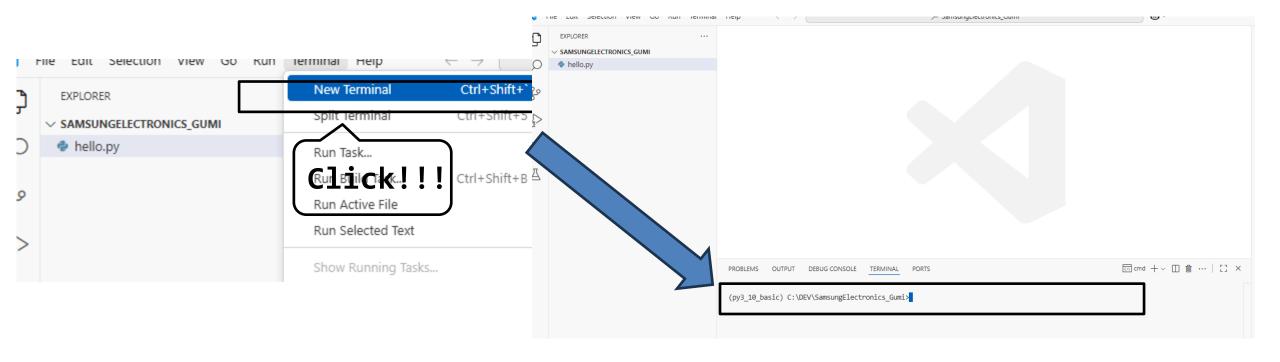
Visual Studio Code 사용

● Python 버전 변경 시, Conda에서 생성한 가상환경의 인터프리터로 변경하기 위해서는 Ctrl + Shift +P를 누른 후, Select interpreter를 입력하여 검색되는 다음 항목을 선택



Visual Studio Code 사용

● (py3_10_basic) C:\DEV\SamsungElectronics_Gumi> 성공

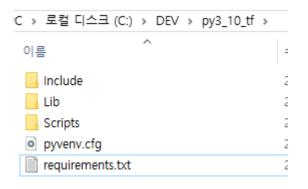


TensorFlow 2 가상환경

```
c:\DEV>cd envs
c:\DEV\envs>python -m venv py3_10_tf
c:\DEV\envs>cd py3_10_tf
c:\DEV\envs\py3_10_tf>Scripts\activate
(py3_10_tf) c:\DEV\envs\py3_10_tf>dir/w/p
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: BEDo-C858
c:\DEV\envs\py3_10_tf 디렉터리
[.]
            [Include] [Lib]
                           pyvenv.cfg [Scripts]
      [..]
      1개 파일 118 바이트
      5개 디렉터리 56,743,809,024 바이트 남음
(py3_10_tf) c:\DEV\envs\py3_10_tf>
```

requirements.txt 복사

-11 1	PC > 로컬 디스크 (C:) > DEV >	py5_10_basic
	이름	수정한 날짜
	Include	2024-06-22
	Lib	2024-06-22
	Scripts	2024-06-22
١	share	2024-06-22
١	pyvenv.cfg	2024-06-22
	requirements.txt	2024-06-22



(py3 $_10_t$) c:\DEV\py3 $_10_t$ >dir/w/p

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

c:\DEV\py3_10_tf 디렉터리

[.] [..] [Include] [Lib] pyvenv.cfg requirements.txt [Scripts] 2개 파일 953 바이트 5개 디렉터리 68,334,325,760 바이트 남음

(py3_10_tf) c:\DEV\py3_10_tf>

requirements.txt 복사

(py3_10_tf) c:\DEV\py3_10_tf>pip install -r requirements.txt

Collecting asttokens==2.4.1

Using cached asttokens-2.4.1-py2.py3-none-any.whl (27 kB)

Collecting colorama==0.4.6

Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)

Collecting comm==0.2.2

Using cached comm-0.2.2-py3-none-any.whl (7.2 kB)

Collecting contourpy==1.2.1

Downloading contourpy-1.2.1-cp310-cp310-win_amd64.whl (187 kB)

------ 187.5/187.5 kB 5.7 MB/s eta 0:00:00

Collecting cycler==0.12.1

Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)

쥬피터 노트북 내보내기

```
pip install ipykernel
python.exe -m pip install --upgrade pip

(py3_10_tf) c:\DEV\py3_10_tf>python -m ipykernel install --user --name py3_10_tf --display-
name "py3_10_tf"
Installed kernelspec py3_10_tf in C:\Users\k8s\AppData\Roaming\jupyter\kernels\py3_10_tf

(py3_10_tf) c:\DEV\envs\py3_10_tf>deactivate
c:\DEV\envs\py3_10_tf>cd ..

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

라이브러리

```
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 : 2.0.0 pandas 버전 : 2.2.2 matplotlib 버전 : 3.9.0 scipy 버전 : 1.13.1 IPython 버전 : 8.25.0 sklearn : 1.5.0
```

TensorFlow 2 설치

(py3_10_tf) c:\DEV\py3_10_tf>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 tensorflow as tf

Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'tensorflow'

>>> quit()

(py3_10_tf) c:\DEV\py3_10_tf>pip install --upgrade tensorflow

Collecting tensorflow

Downloading tensorflow-2.16.1-cp310-cp310-win_amd64.whl.metad

•••

Attempting uninstall: numpy

Found existing installation: numpy 2.0.0

Uninstalling numpy-2.0.0:

Successfully uninstalled numpy-2.0.0

Successfully installed MarkupSafe-2.1.5 absl-py-2.1.0 astunparse-1.6.3 certifi-2024.6.2 charset-normalizer-3.3.2 flatbuffers-24.3.25 gast-0.5.4 google-pasta-0.2.0 grpcio-1.64.1 h5py-3.11.0 idna-3.7 keras-3.3.3 libclang-18.1.1 markdown-3.6 markdown-it-py-3.0.0 mdurl-0.1.2 ml-dtypes-0.3.2 namex-0.0.8 numpy-1.26.4 opt-einsum-3.3.0 optree-0.11.0 protobuf-4.25.3 requests-2.32.3 rich-13.7.1 tensorboard-2.16.2 tensorboard-data-server-0.7.2 tensorflow-2.16.1 tensorflow-intel-2.16.1 tensorflow-io-gcs-filesystem-0.31.0 termcolor-2.4.0 urllib3-2.2.2 werkzeug-3.0.3 wheel-0.43.0 wrapt-1.16.0

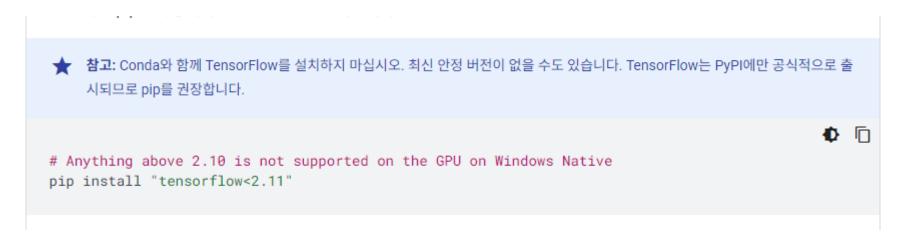
(py3_10_tf) c:\DEV\py3_10_tf>



Setting up a machine learning / deep learning environment

GPU에서 TensorFlow 2 설치

https://www.tensorflow.org/install/pip?hl=ko#windows-native



pip install "numpy<2"

(py3_10_tf_gpu) c:\DEV\envs\py3_10_tf_gpu>pip install "tensorflow<2.11"







C:\DEV\envs\py3_10_tf_gpu>nvcc --version nvcc: NVIDIA (R) Cuda compiler driver Copyright (c) 2005-2022 NVIDIA Corporation Built on Wed_Sep_21_10:41:10_Pacific_Daylight_Time_2022 Cuda compilation tools, release 11.8, V11.8.89 Build cuda_11.8.r11.8/compiler.31833905_0

Setting up a machine learning / deep learning environment

GPU에서 TensorFlow 2 설치 TensorFlow가 GPU

```
import tensorflow as tf
print("TensorFlow version:", tf.__version___)
print("GPU available:", tf.test.is_gpu_available())
import tensorflow as tf
print(tf.config.list_physical_devices('GPU'))
. is_build_with_cuda()
import tensorflow as tf print(tf.test.is_built_with_cuda())
output: True
. gpu_device_name()
tf.test.gpu_device_name()
output: '/device:GPU:0'
```

TensorFlow 2 설치 확인

```
[1]: import tensorflow as tf

[2]: with tf.compat.v1.Session() as sess:
    helloworld = tf.constant("Hello World!")
    print(sess.run(helloworld))

WARNING:tensorflow:From C:\Users\k8s\AppData\Local\Temp\ipykernel_9808\1734762880.py:1: The name tf.Session is deprecated. Please use tf.compa t.v1.Session instead.
    b'Hello World!'

[3]: helloworld = tf.constant("Hello World!")
    tf.print(helloworld)

Hello World!
```

THANK YOU.

앞으로의 엔지니어는 단순한 '코더'나 '기계 조작자'가 아니라 뇌-기계 인터 페이스를 통해 지식과 능력을 즉각 확장하는 존재(뉴로-인터페이스: Neuro Interface)가 될 수 있습니다.

○ 목표 달성을 위한 여정이 시작됩니다. → 궁금한 점이 있으시면 언제든 문의해주세요! → 함께 코더와 프롬프트 전문가로 성장해 나갑시다!