



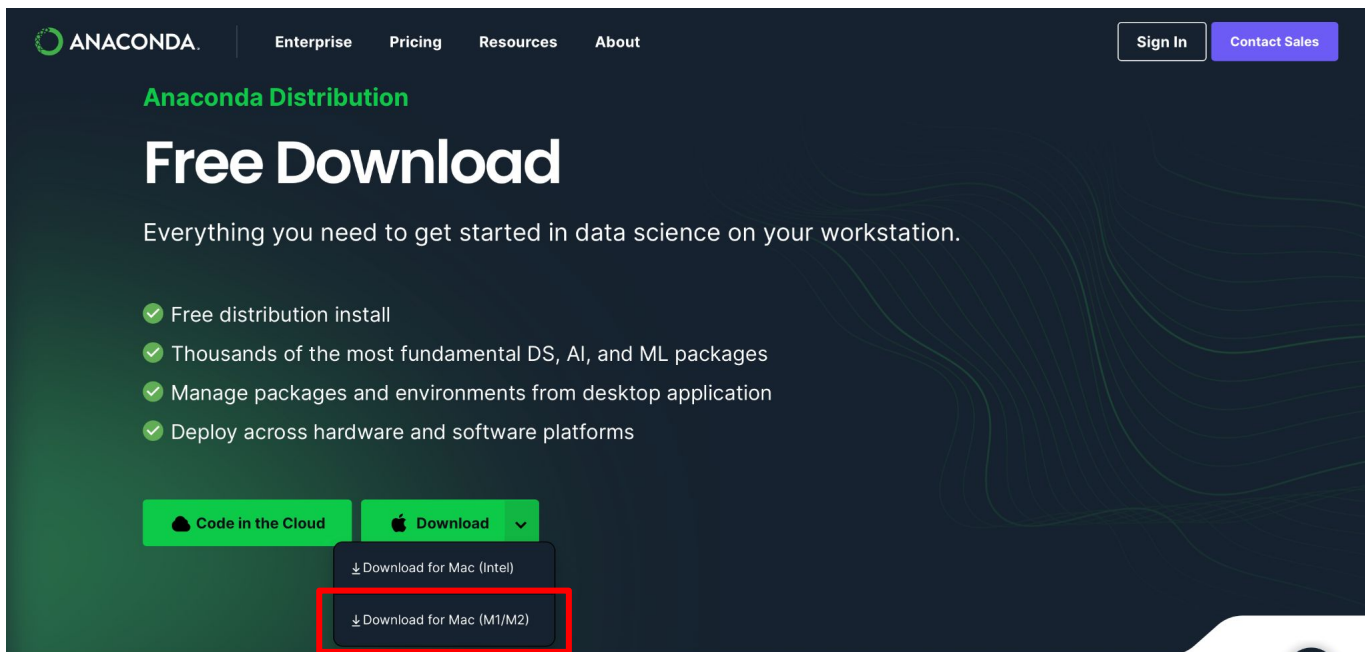
Flask 프로젝트 시작 (Visual Studio Code 사용)



- 환경: MacOS
- Python 설치 (Anaconda 설치 or python package 설치)
- Visual Studio Code • Flask 설치

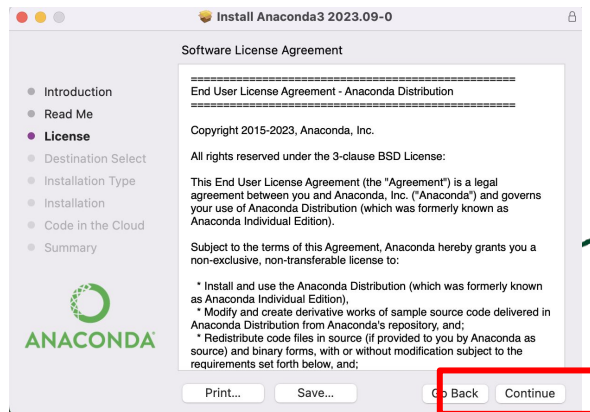
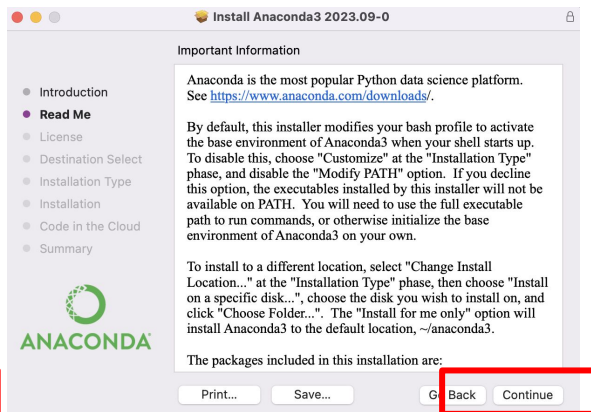
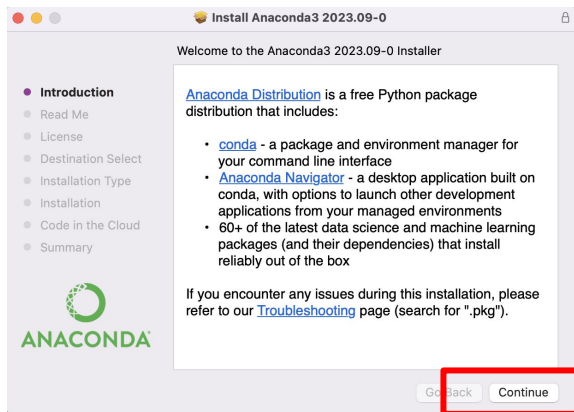
Anaconda 설치

<https://www.anaconda.com/download>

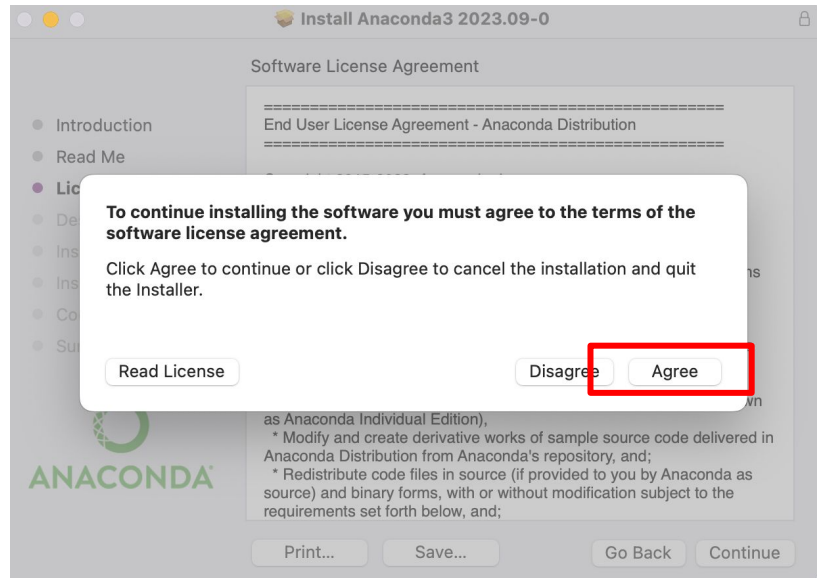
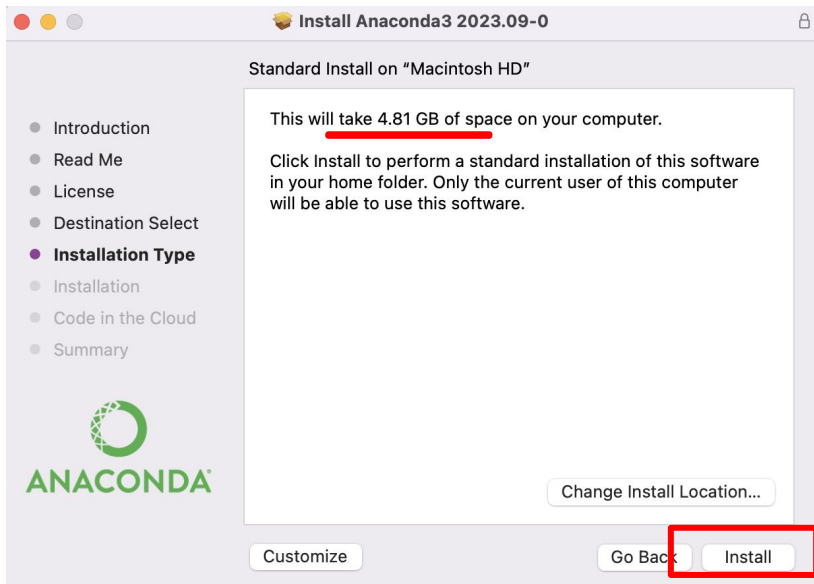


Anaconda 설치

- 다운로드 폴더에 Anaconda3-2023.09-MacOSX-arm64.pkg 파일 실행

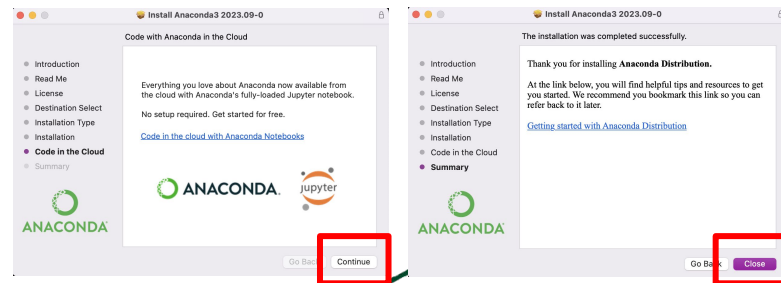
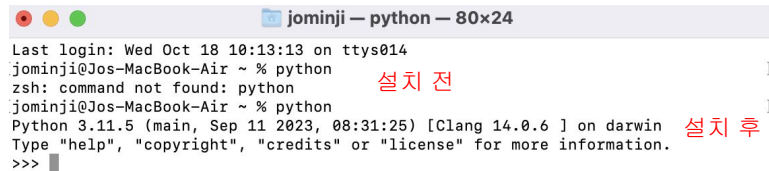
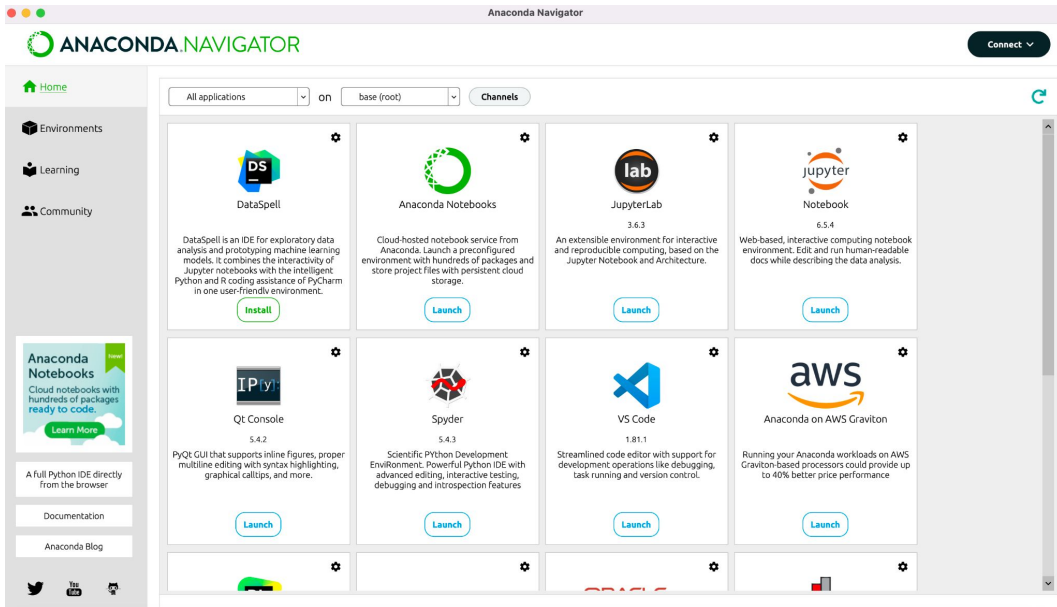


Anaconda 설치



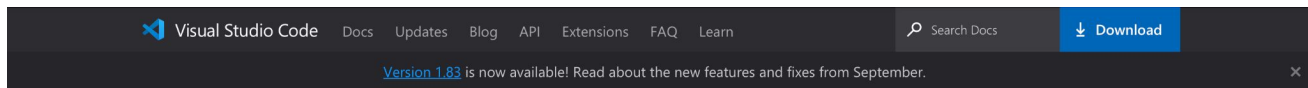
Anaconda 설치

- 아나콘다 설치 확인



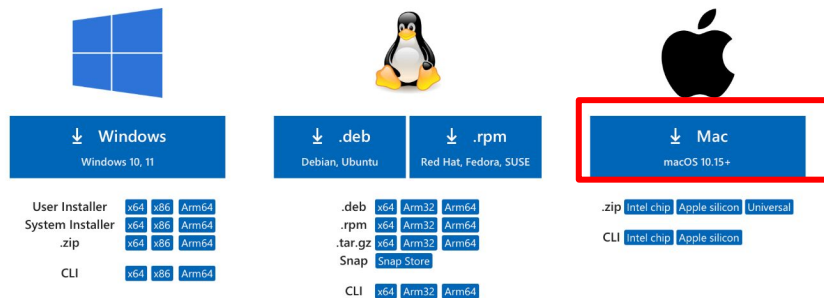
VScode 설치

<https://code.visualstudio.com/download>



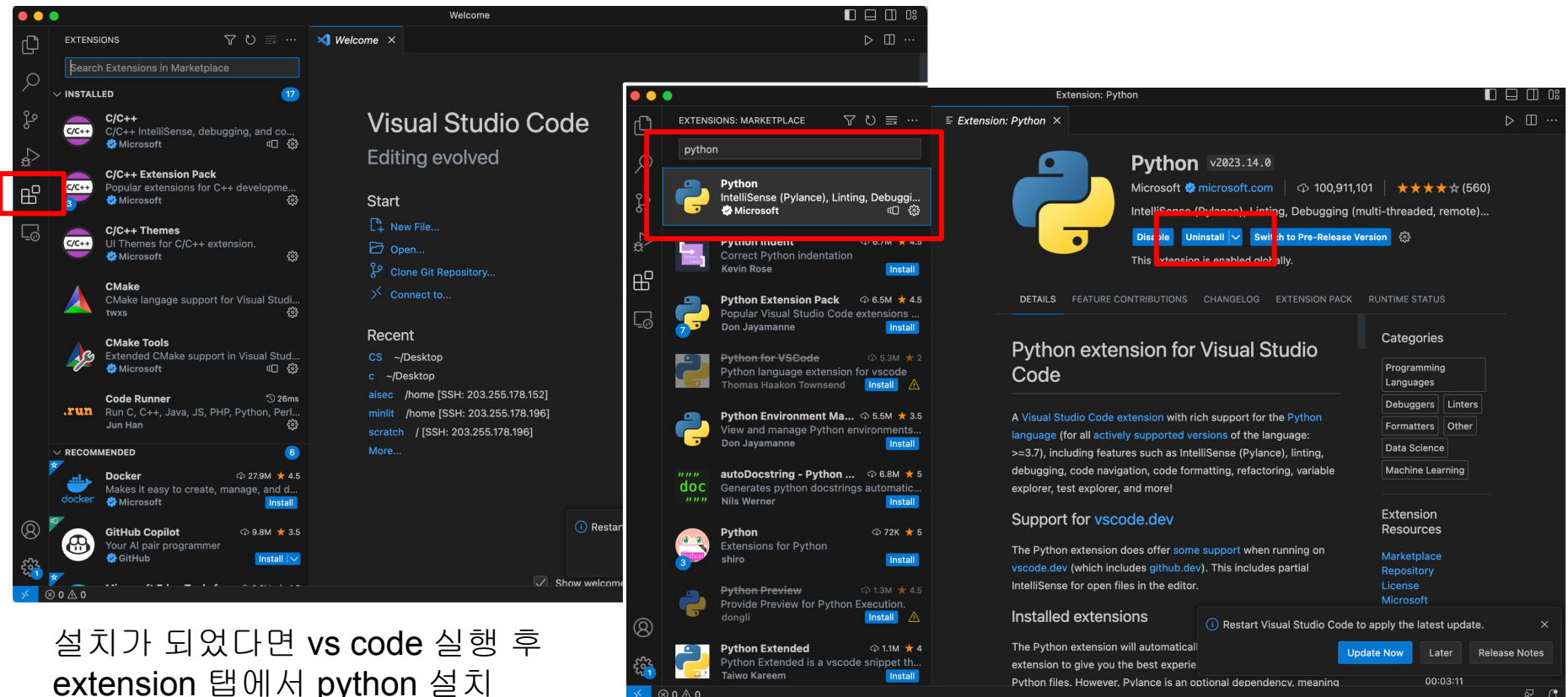
Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

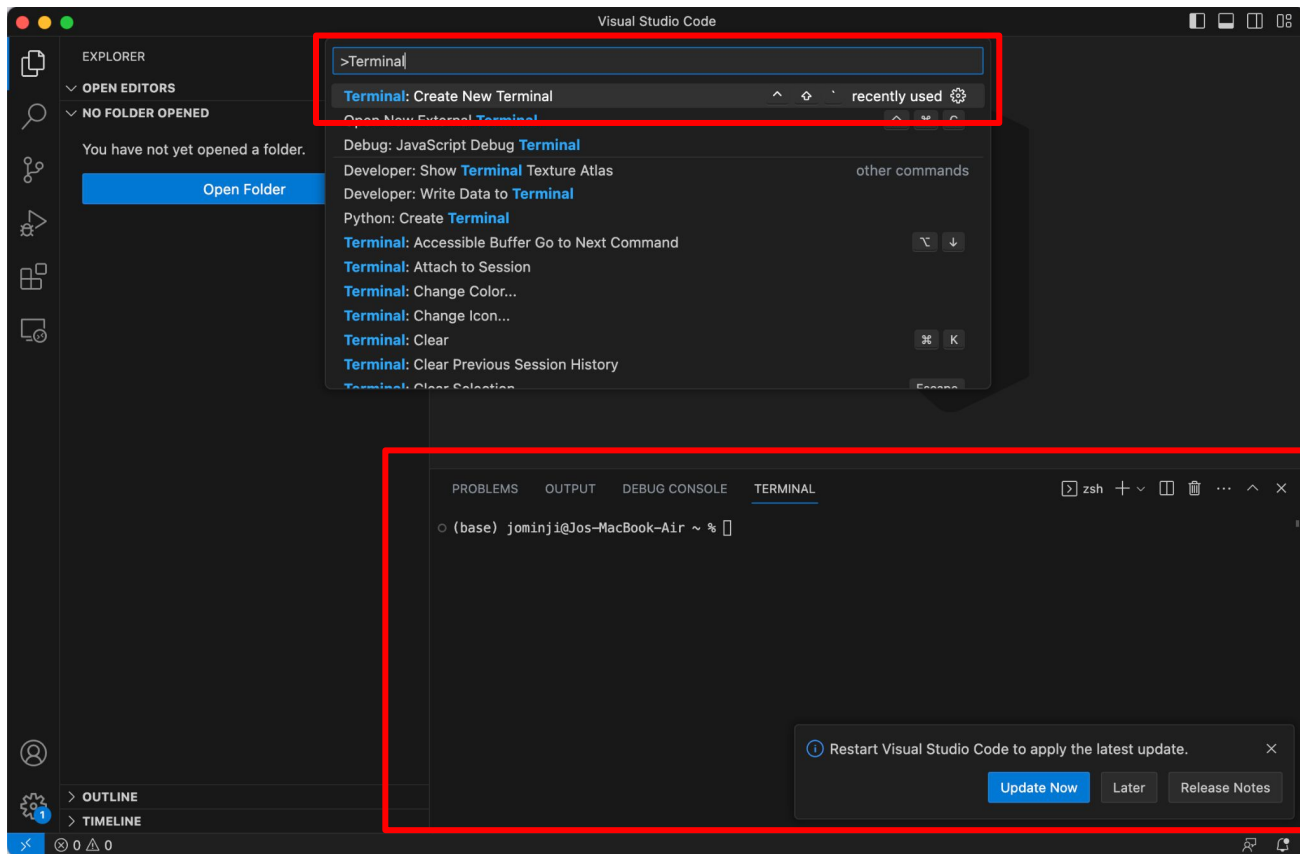
The image shows the main download section of the Visual Studio Code website. It features three columns for different operating systems: Windows, Linux, and Mac. Each column has a platform icon (Windows logo, Tux penguin, and Apple logo respectively) and a blue button with a download icon and the platform name. Below each button are links to various installation methods and their supported architectures.

Platform	Installation Method	Supported Architectures
Windows	User Installer	x64, x86, Arm64
	System Installer	x64, x86, Arm64
	.zip	x64, x86, Arm64
	CLI	x64, x86, Arm64
Linux	.deb	x64, Arm32, Arm64
	.rpm	x64, Arm32, Arm64
	.tar.gz	x64, Arm32, Arm64
	Snap	Snap Store
	CLI	x64, Arm32, Arm64
Mac	.zip	Intel chip, Apple silicon, Universal
	CLI	Intel chip, Apple silicon

VScode 설치



아나콘다 가상환경 생성



F1을 누른 뒤 terminal 검색,
Terminal:Create New
Terminal 클릭

아래 terminal 창 확인

아나콘다 가상환경 생성 및 활성화

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
○ (base) jominji@Jos-MacBook-Air ~ % conda create -n OSWF python==3.10
```

```
● (base) jominji@Jos-MacBook-Air ~ % conda activate OSWF
○ (OSWF) jominji@Jos-MacBook-Air ~ %
```

가상환경 이름으로 바뀐 것을 확인

```
● (base) jominji@Jos-MacBook-Air ~ % conda activate OSWF
○ (OSWF) jominji@Jos-MacBook-Air ~ % conda install flask
```

terminal 창에서

`conda create -n <이름>`
`python==<버전>` 원하는 가상환경
이름과 파이썬 버전을 정하여
가상환경 생성

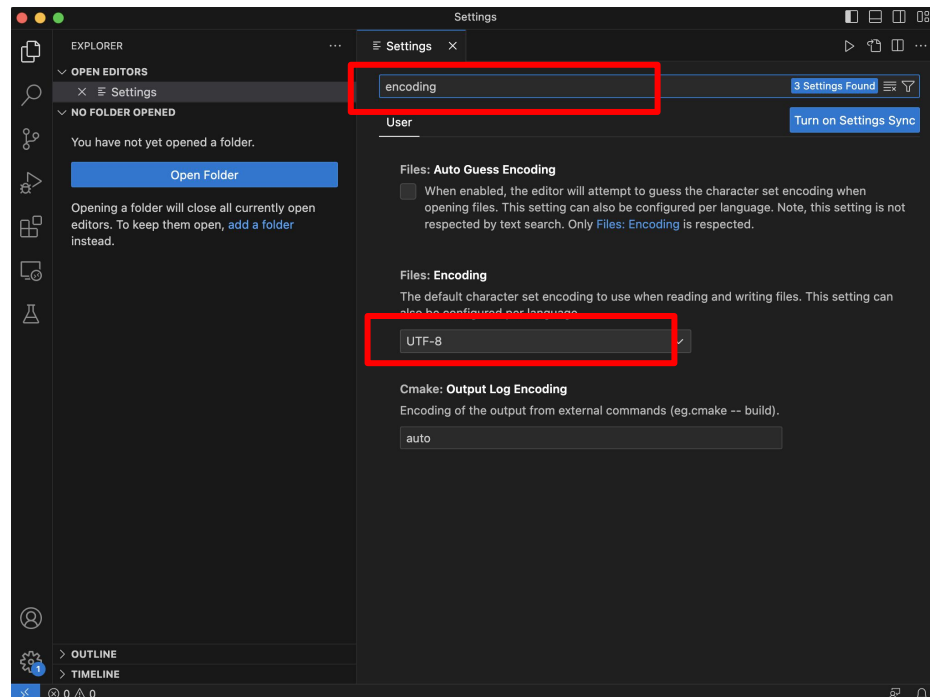
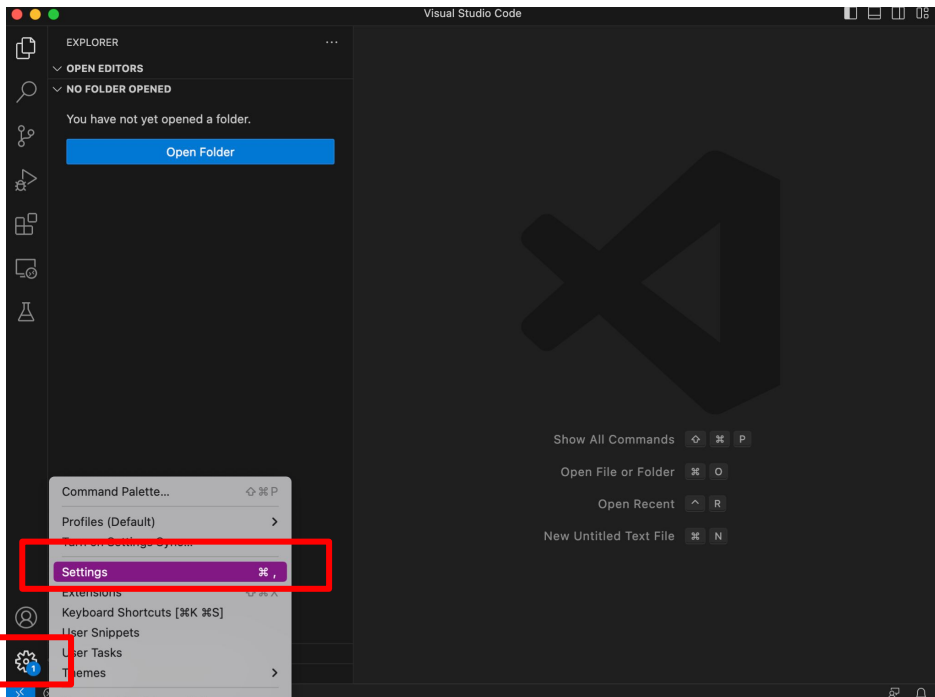
가상환경 생성이 끝나면

`conda activate <가상환경이름>`
을 통해 가상환경 활성화

가상환경 활성화 후

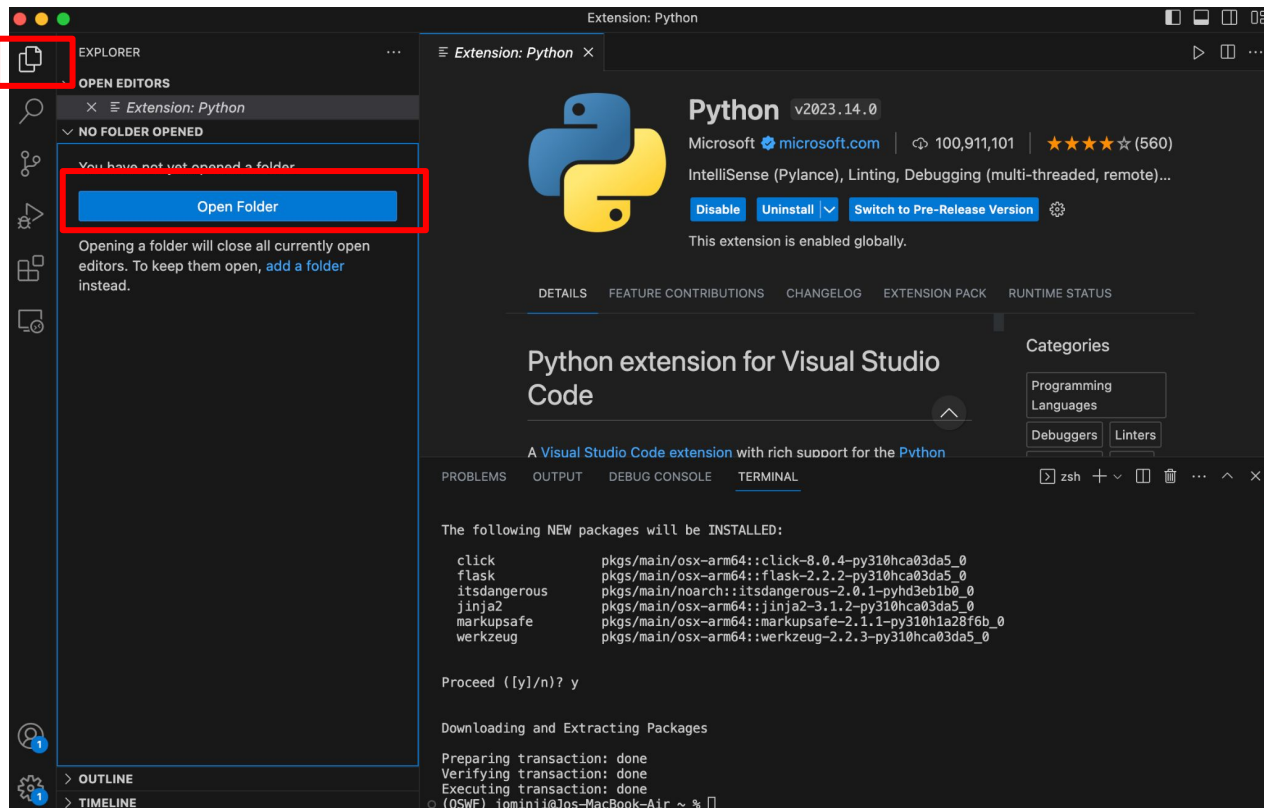
`conda install flask`을 통해
flask 설치

Encoding 확인



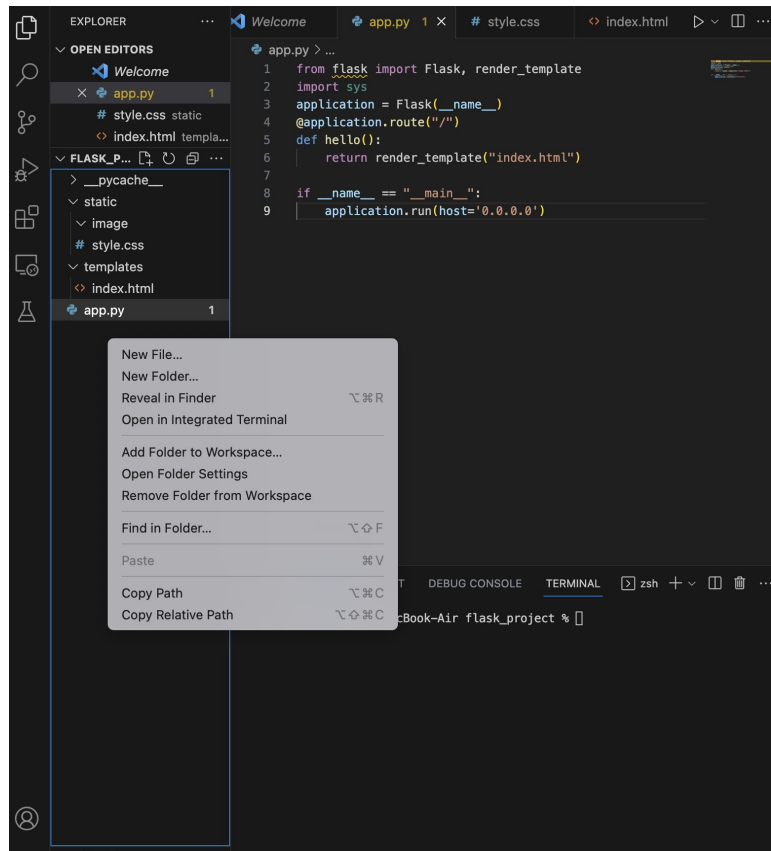
설정에 들어간 후 encoding에서 UTF-8인지 확인

폴더 생성



explorer 탭에서 open folder
선택하여 작업할 폴더 선택

폴더 생성



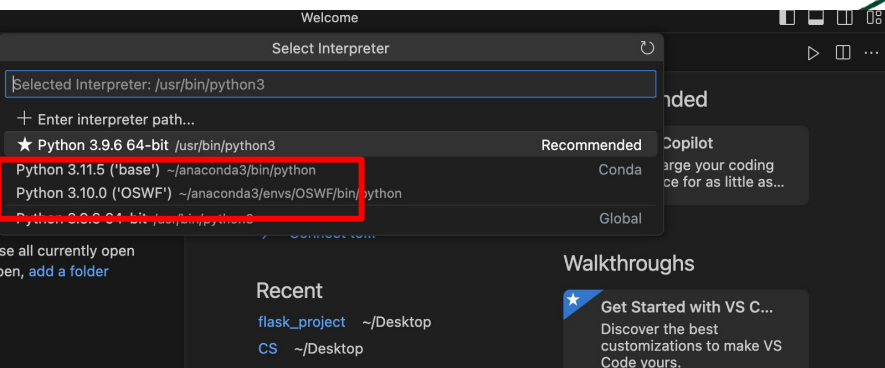
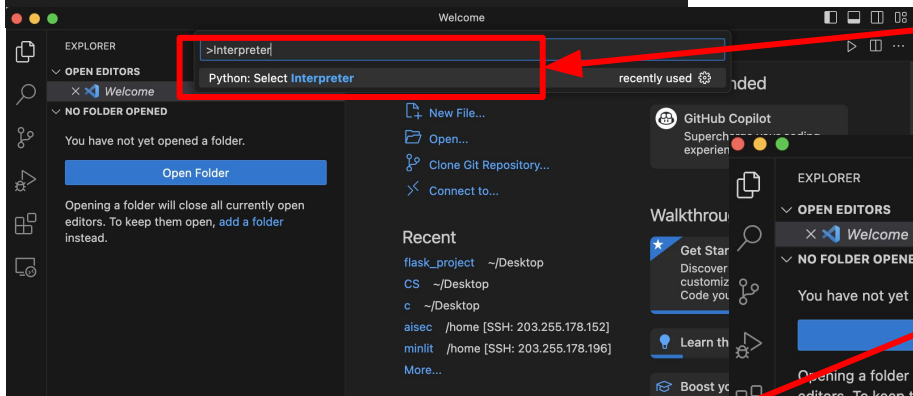
폴더/파일은 강의를 따라 생성

Interpreter 선택

- 가상환경을 활성화하였다면 문제되지 않지만 파이썬 파일에서도 가상환경에 설치된 패키지가 인식되도록 하려면 interpreter를 선택

f1 누른 후 interpreter 검색, Python: Select interpreter 선택

```
app.py > ...  
1 from flask import Flask, render_template  
2 import sys  
3 application = Flask(__name__)  
4 @application.route("/")  
5 def hello():  
6     return render_template("index.html")  
7  
8 if __name__ == "__main__":  
9     application.run(host='0.0.0.0')
```



만들어 두었던 가상환경 선택

Interpreter 선택

```
app.py > ...  
1  from flask import Flask, render_template  
2  import sys  
3  application = Flask(__name__)  
4  @application.route("/")  
5  def hello():  
6      return render_template("index.html")  
7  
8  if __name__ == "__main__":  
9      application.run(host='0.0.0.0')
```

flask가 인식됨을 확인

Flask 실행

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL zsh + - [ ] [X] ... ^
(base) jominji@Jos-MacBook-Air flask_project % conda activate OSWF
(OSWF) (base) jominji@Jos-MacBook-Air flask_project % flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

가상환경 활성화 후 flask run을 통해 실행

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL zsh + - [ ] [X] ... ^
(base) jominji@Jos-MacBook-Air flask_project % conda activate OSWF
(OSWF) (base) jominji@Jos-MacBook-Air flask_project % flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

링크로 들어가서 결과물 확인 가능

[EwhaYelp](#)

- [List](#)
- [Review](#)
- [Register_restaurant](#)
- [Contact](#)

가상환경 비활성화 : conda deactivate