

The background of the slide is a vibrant, abstract splash of paint in various shades of blue, cyan, and magenta. The paint splatters are of different sizes and densities, creating a dynamic and energetic visual effect. The text is centered over this background.

YOLO 설치

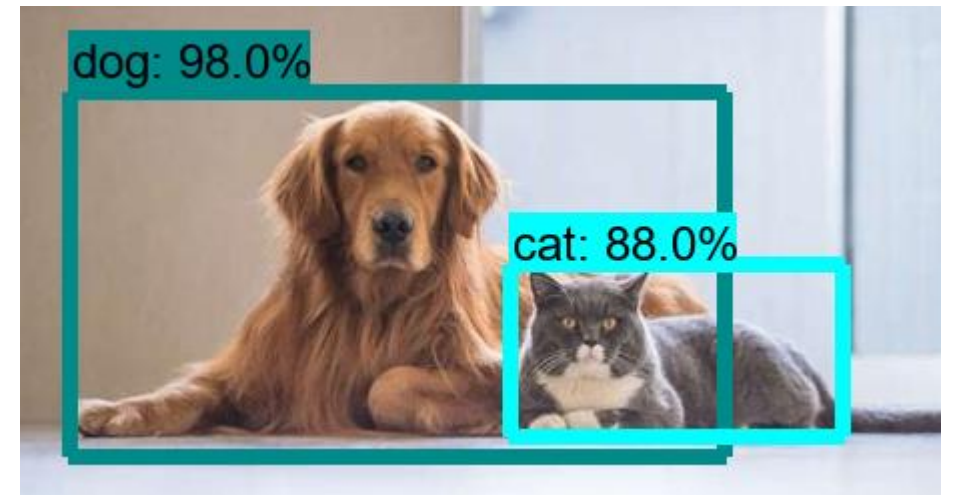
최석재 lingua@naver.com

YOLO v8

You Only Look Once

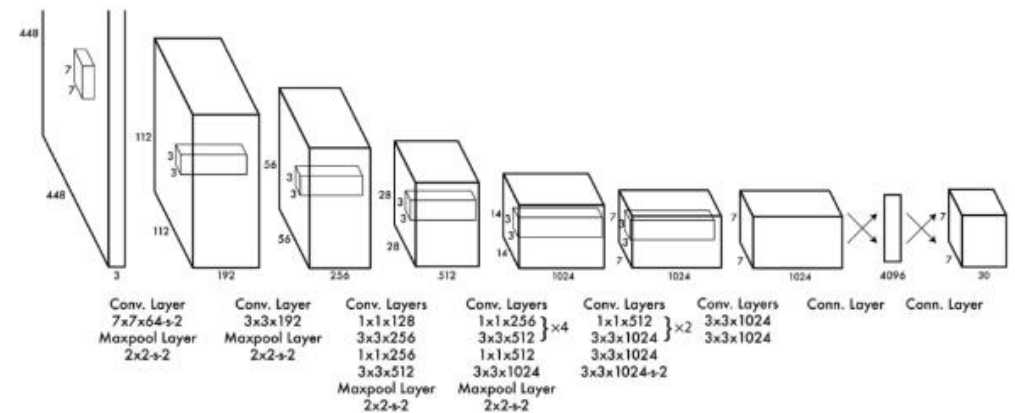
YOLO *You Only Look Once*

- 컴퓨터 비전에서 중요한 주제는 객체를 탐지하는 것이었음
- 기존의 방법은 이미지 내에서 객체가 될 수 있는 경계를 먼저 찾고,
 - 해당 경계가 무엇에 해당하는지 판정하는 방식
- 경계 찾기(Region Proposal)와 이미지 분류(Classification)의 두 단계를 거치는 Two-stage detector
- YOLO는 이미지 내의 경계 탐지와 객체 결정을
 - 한 번의 스캔으로 결정. One-stage detector.
 - 속도가 매우 빠르고, 정확도도 높음



접근 방식

- YOLO는 이미지가 입력되면 정해진 숫자의 grid로 나눔
- 객체의 중심이 그리드 셀에 있는지를 파악하여 경계 box를 예측하여 그려냄
- 동시에 경계 box 내의 객체가 어떠한 이미지인지를 예측



The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

YOLO *history*

- YOLO v1 : 2015년, 워싱턴 대학교의 조셉 레드몬, 알리 파르히가 개발
- YOLO v2 : 2016년
- YOLO v3 : 2018년
- YOLO v4 : 2020년
- YOLO v5 : 2020년
- YOLO v6 : 2022년. 상용화 시작
- YOLO v7 : 2022년. 포즈 추정 추가
- YOLO v8 : 2023년

※ v7(7월)이 v6(9월)보다 일찍 출시됨

<https://github.com/ultralytics/ultralytics/issues/189>

<https://deci.ai/blog/history-yolo-object-detection-models-from-yolov1-yolov8/>

<https://docs.ultralytics.com/ko#yolo-a-brief-history>

YOLO classes

- ultralytics에서 만들어 배포하는 YOLO는
- coco dataset으로 학습되어
- 80가지의 클래스를 분류할 수 있다

<https://paperswithcode.com/dataset/coco>
<https://github.com/ultralytics/ultralytics?tab=readme-ov-file>

0: person	40: wine glass
1: bicycle	41: cup
2: car	42: fork
3: motorcycle	43: knife
4: airplane	44: spoon
5: bus	45: bowl
6: train	46: banana
7: truck	47: apple
8: boat	48: sandwich
9: traffic light	49: orange
10: fire hydrant	50: broccoli
11: stop sign	51: carrot
12: parking meter	52: hot dog
13: bench	53: pizza
14: bird	54: donut
15: cat	55: cake
16: dog	56: chair
17: horse	57: couch
18: sheep	58: potted plant
19: cow	59: bed
20: elephant	60: dining table
21: bear	61: toilet
22: zebra	62: tv
23: giraffe	63: laptop
24: backpack	64: mouse
25: umbrella	65: remote
26: handbag	66: keyboard
27: tie	67: cell phone
28: suitcase	68: microwave
29: frisbee	69: oven
30: skis	70: toaster
31: snowboard	71: sink
32: sports ball	72: refrigerator
33: kite	73: book
34: baseball bat	74: clock
35: baseball glove	75: vase
36: skateboard	76: scissors
37: surfboard	77: teddy bear
38: tennis racket	78: hair drier
39: bottle	79: toothbrush

Windows에서 YOLO 사용

YOLO 설치

- YOLO의 설치가 최근 쉬워졌다
- 아나콘다 프롬프트에서 pip 명령어로 ultralytics와 OpenCV를 설치한다
- 그리고 다음 슬라이드를 참고하여 파이토치를 설치한다
- `pip install ultralytics opencv-python`
- 설치가 끝났으면 YOLO 분석 부분으로 넘어가 진행한다
- 만약 진행이 잘 되지 않으면, [참고] 부분을 확인한다

파이토치 설치

- 여기서 사용할 YOLO는 파이토치에서 동작한다
- 파이토치의 GET STARTED 페이지에서 적절한 버전을 선택한다
- <https://pytorch.org/get-started/locally/>
- GPU 사용시 Tensorflow에서 GPU가 동작할 수 있도록 설정이 이미 끝나 있어야 한다

GPU

PyTorch Build	Stable (2.0.1)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 11.7	CUDA 11.8	ROCm 5.4.2	CPU
Run this Command:	<pre>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118</pre>			

CPU

PyTorch Build	Stable (2.3.1)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.1	CUDA 12.4	CPU
Run this Command:	<pre>pip3 install torch torchvision torchaudio</pre>			

이 부분을 복사한다

[참고] YOLO v5 설치

YOLO v5 설치

- 앞의 방법으로 v8이 잘 설치되지 않았다면 YOLO 설치는 어려울 수 있다
- 이러한 경우 v5를 설치하여 성공하면, 관련 설정으로 v8도 실행될 수 있다
- Windows에서의 설치와 사용 방법을 알아보고,
- 그래도 잘 되지 않는다면 Colab에서 사용하도록 한다

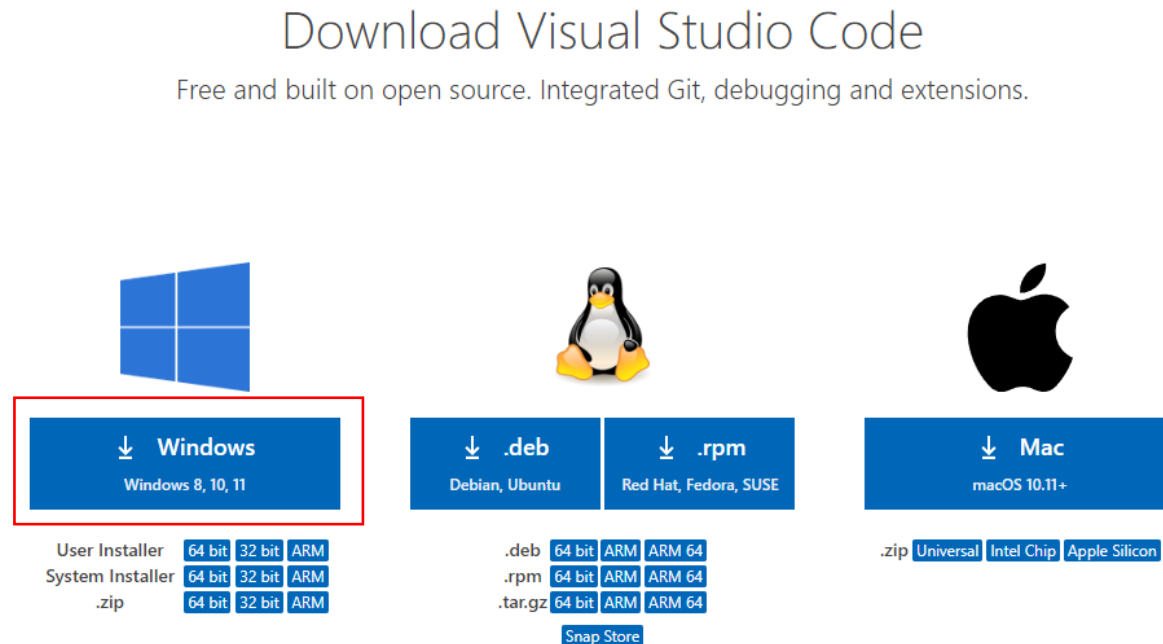
VS Code 설치

- Microsoft에서 무료로 배포하는 개발 환경
- Python뿐만 아니라 JavaScript 등을 사용할 수 있도록 지원하고 있다
- YOLO 실행 시 VS Code에 연결되는 Python이 있으면 잘 진행된다
- <https://code.visualstudio.com/download>

※ 잘 진행이 되지 않을 때만 설치한다

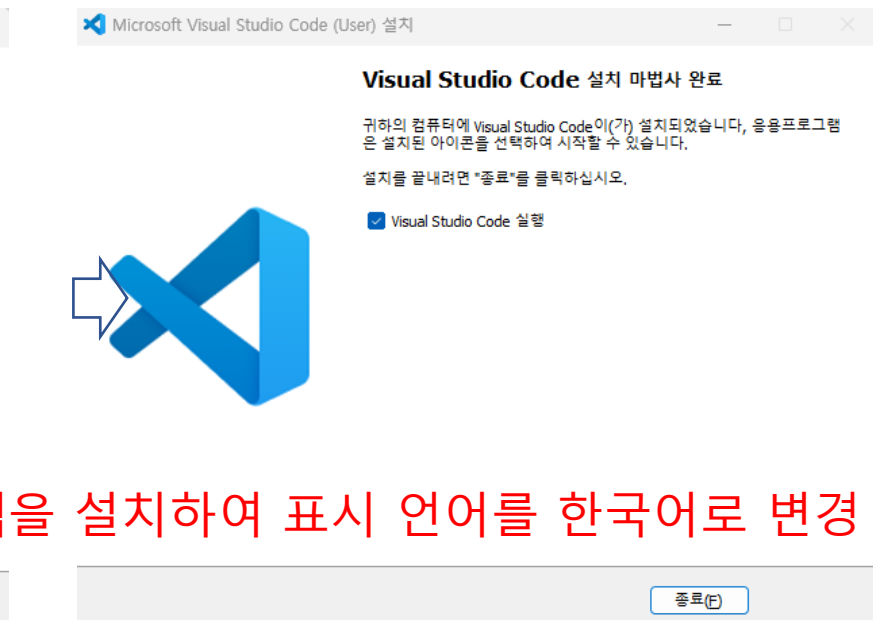
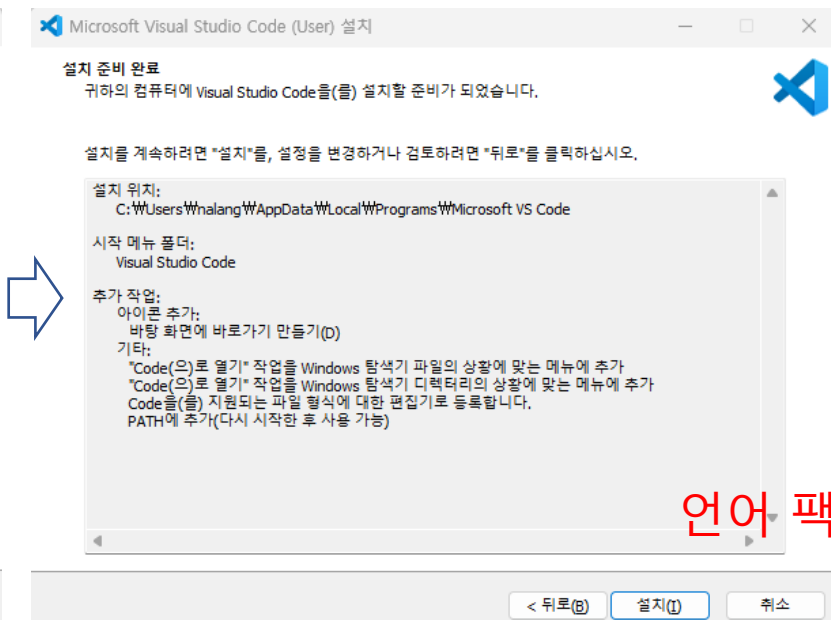
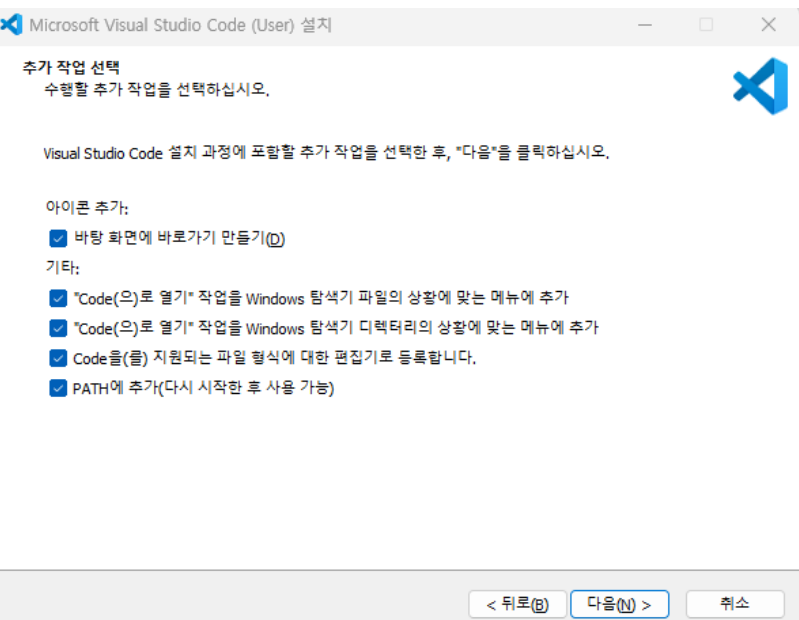
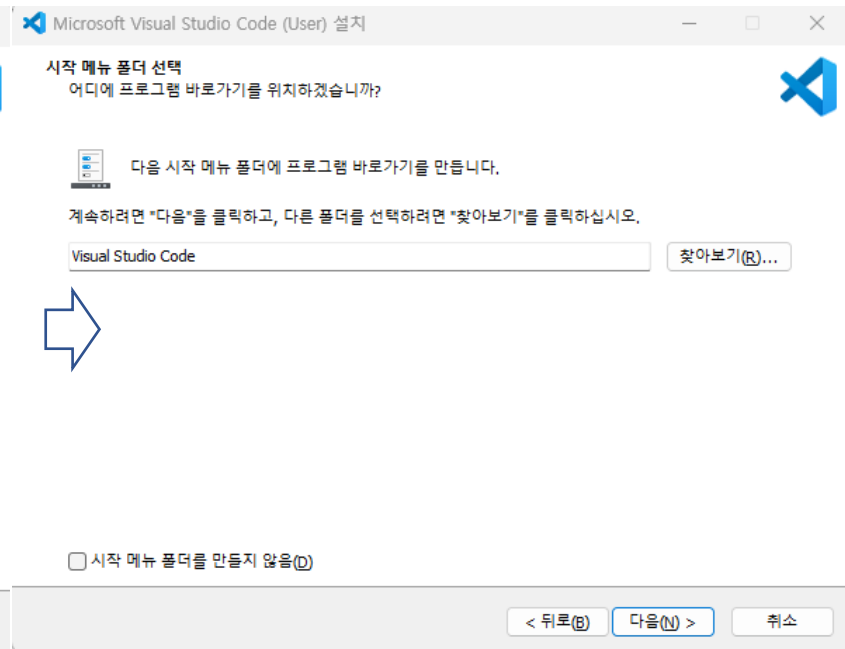
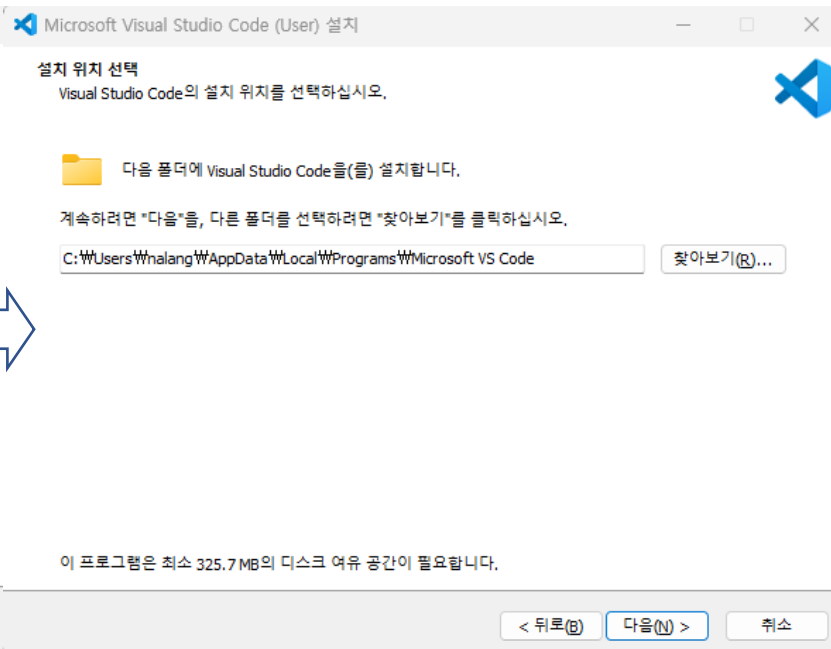
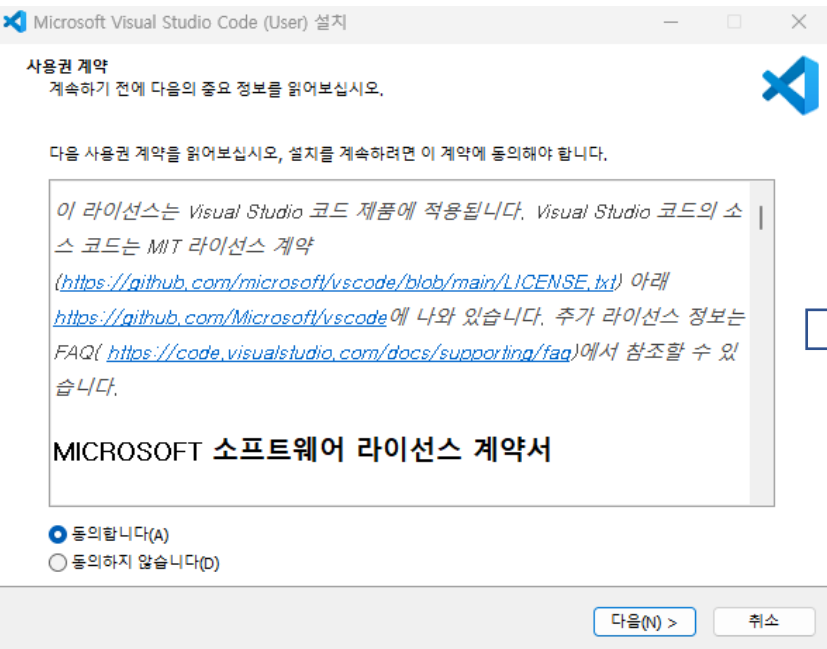
Download Visual Studio Code

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



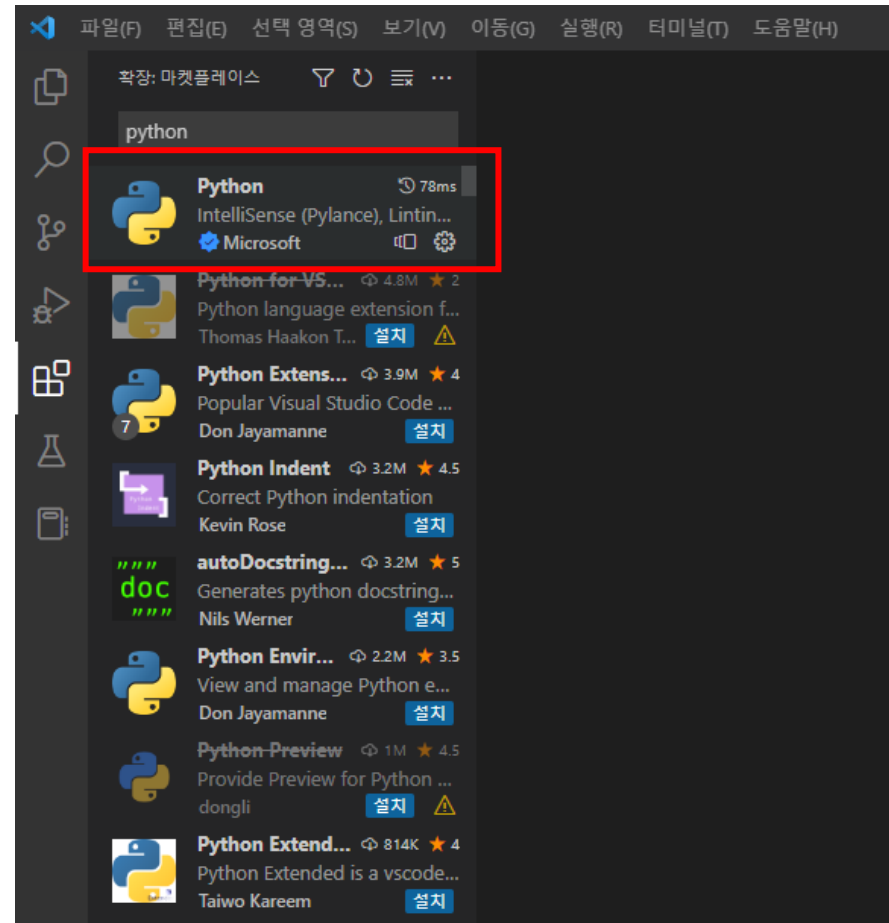
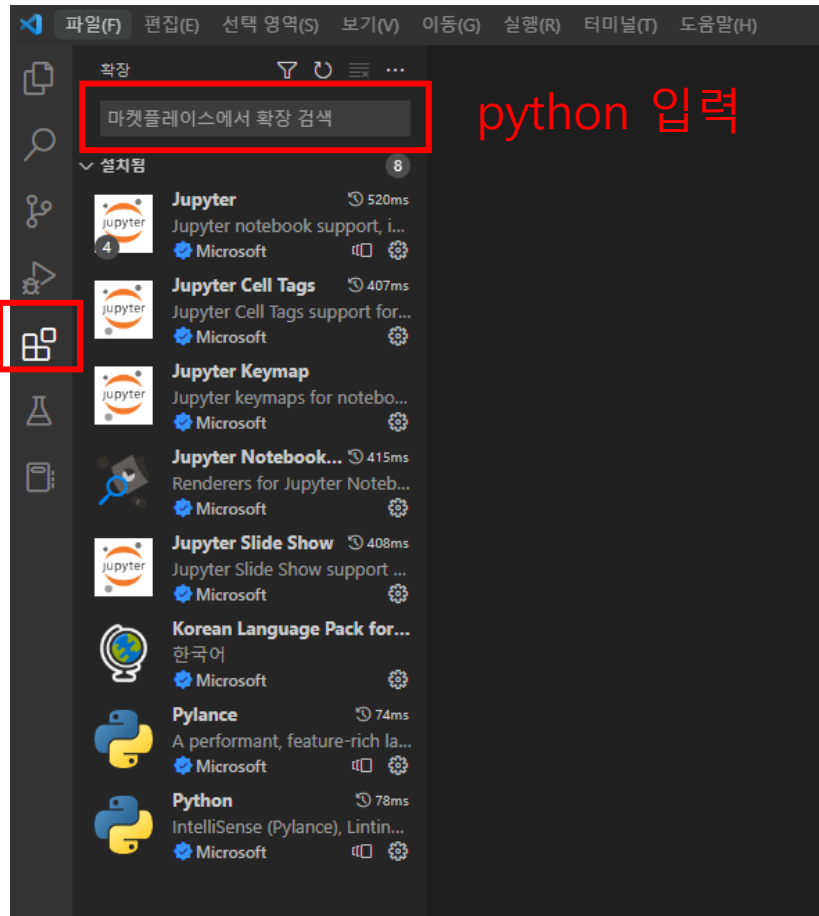
The image shows the Visual Studio Code download page. It features three main sections for different operating systems: Windows, Linux, and Mac. The Windows section is highlighted with a red box. Below each section, there are links to download various installer types (User, System, .deb, .rpm, .tar.gz, .zip) for different architectures (64 bit, 32 bit, ARM). The Linux section also includes a Snap Store link.

Platform	Architecture	Download Link	
Windows	User Installer	64 bit, 32 bit, ARM	
	System Installer	64 bit, 32 bit, ARM	
	.zip	64 bit, 32 bit, ARM	
	Linux	.deb	64 bit, ARM, ARM 64
		.rpm	64 bit, ARM, ARM 64
		.tar.gz	64 bit, ARM, ARM 64
Mac	.zip	Universal, Intel Chip, Apple Silicon	
	Snap Store		



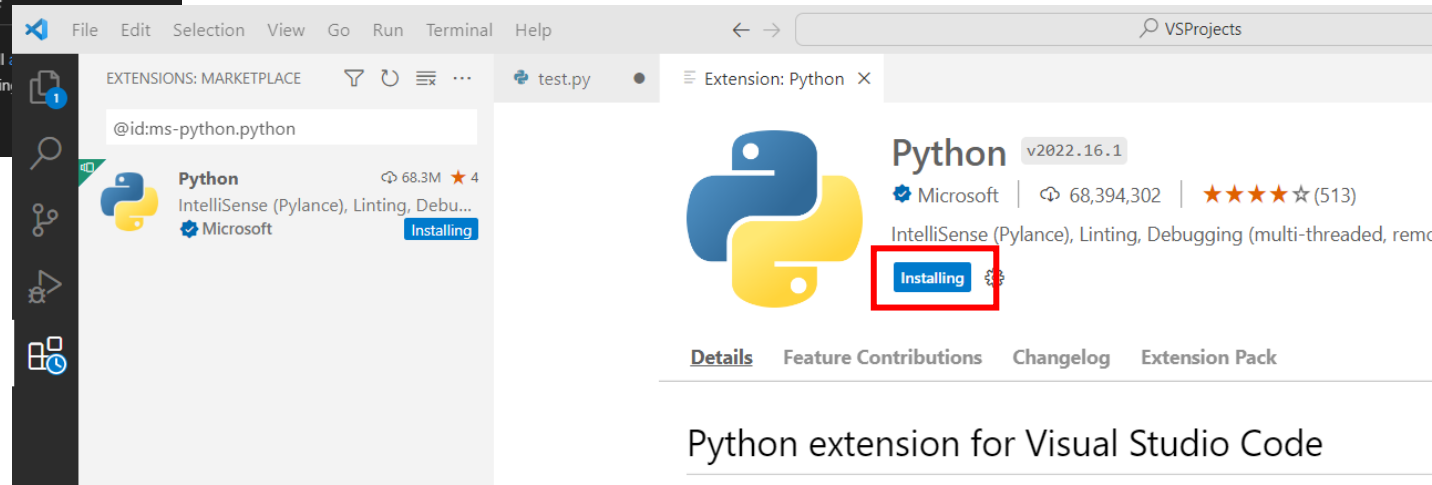
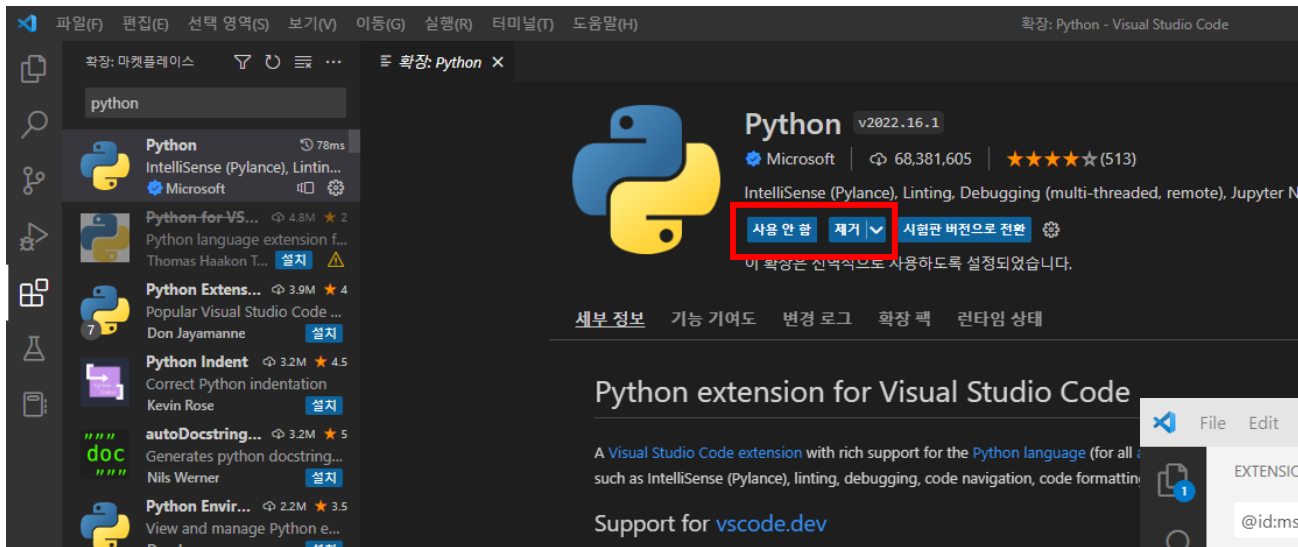
언어 팩을 설치하여 표시 언어를 한국어로 변경

플러그인 설치



파이썬 설치

- 아나콘다의 파이썬과 바로 연결되었다면 아래 첫 번째 그림과 같이 나오고 (완료),
- 그렇지 않다면 두 번째 그림과 같이 나온다. 이 경우 설치를 진행한다

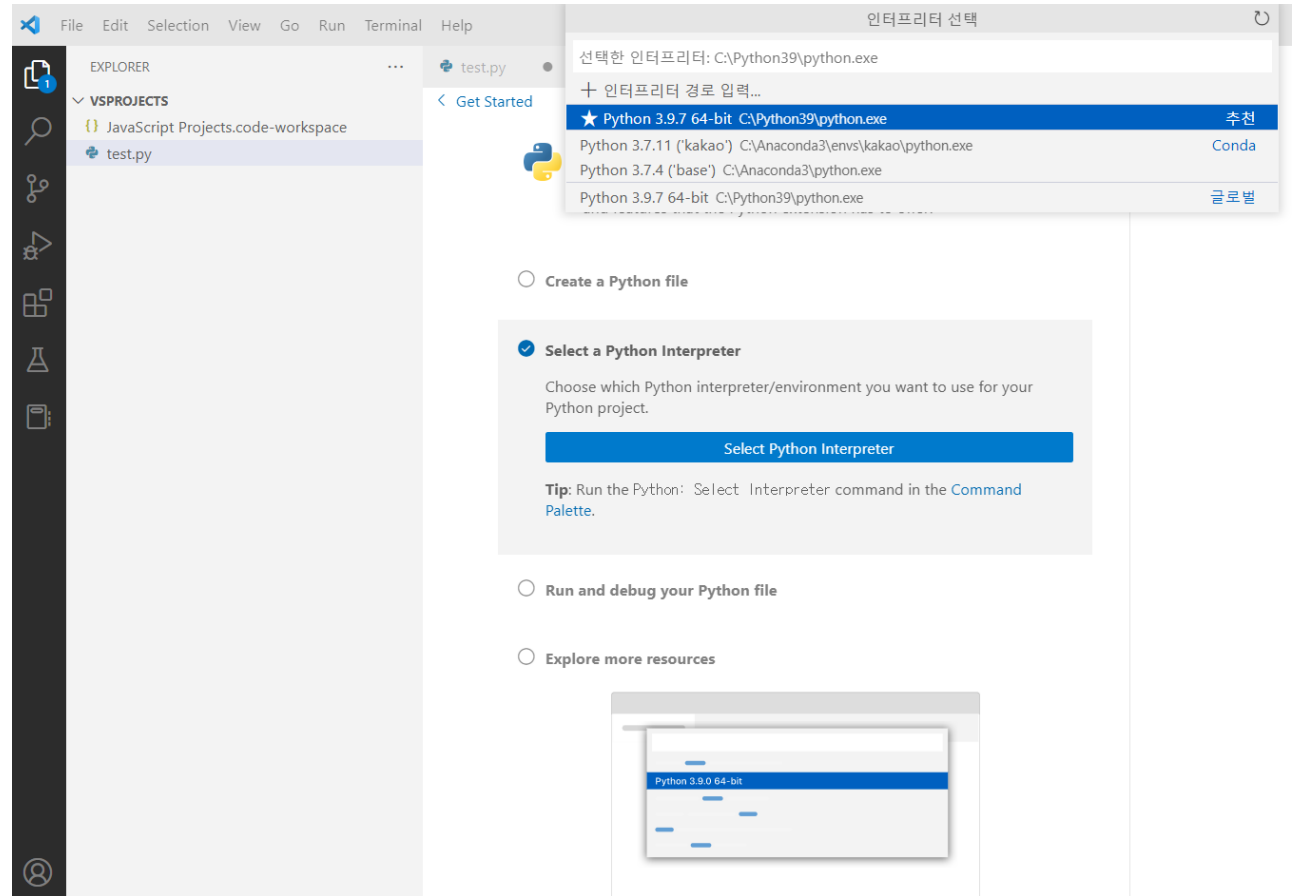


아나콘다와 연결

- 설치가 끝나면 C:\Python39\python.exe와 연결된다

F1 또는 Ctrl+Shift+P 입력 후,
Python: Select Interpreter 입력

```
import sys
print(sys.version)    # 3.9.7
```



파이토치 설치

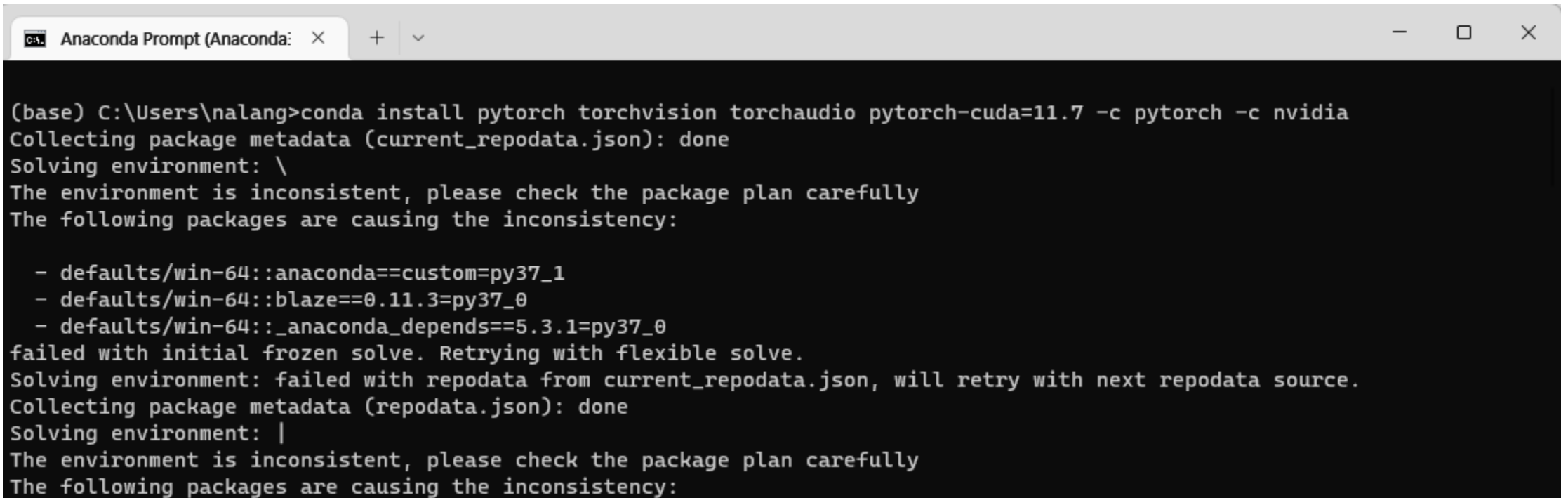
- 여기서 사용할 YOLO는 파이토치에서 동작한다
- 파이토치의 GET STARTED 페이지에서 적절한 버전을 선택한다
- <https://pytorch.org/get-started/locally/>
- Tensorflow에서 GPU가 동작할 수 있도록 설정이 이미 끝나 있어야 한다

PyTorch Build	Stable (2.0.1)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 11.7	CUDA 11.8	ROCm 5.4.2	CPU
Run this Command:	<pre>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118</pre>			

← 이 부분을 복사한다

파이토치 설치

- 복사한 내용을 Anaconda Prompt에 붙여넣은 뒤 실행한다
- 중간에 proceed ([y]/n)? 이 나오면 y를 눌러 계속 진행



```
(base) C:\Users\nalang>conda install pytorch torchvision torchaudio pytorch-cuda=11.7 -c pytorch -c nvidia
Collecting package metadata (current_repodata.json): done
Solving environment: \
The environment is inconsistent, please check the package plan carefully
The following packages are causing the inconsistency:

- defaults/win-64::anaconda==custom=py37_1
- defaults/win-64::blaze==0.11.3=py37_0
- defaults/win-64::_anaconda_depends==5.3.1=py37_0
failed with initial frozen solve. Retrying with flexible solve.
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: |
The environment is inconsistent, please check the package plan carefully
The following packages are causing the inconsistency:
```

파이토치 확인

- 설치가 잘 되었는지 Jupyter Notebook에 다음의 코드를 입력해 확인한다
- `import torch`
- `print(torch.__version__)`
- `a = torch.tensor(1.0)`
- `b = torch.tensor(2.0)`
- `c = a + b`
- `print(c)`

```
2.0.1+cu118  
tensor(3.)
```

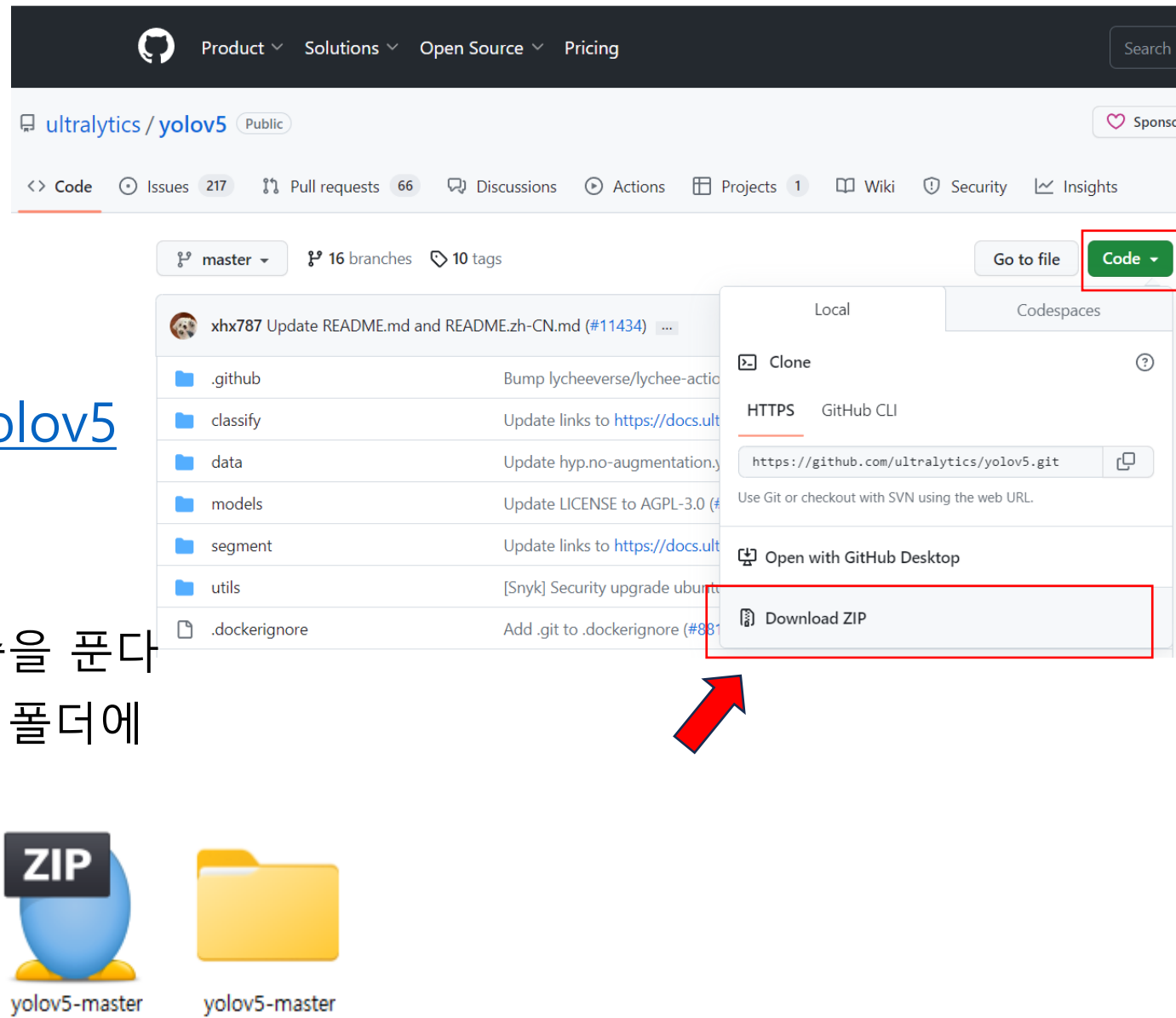
파이토치 버전이 출력되었고
간단한 연산이 이루어졌다

패키지 설치

- 아나콘다 프롬프트에서 torchvision 을 설치한다
- `pip install torchvision`
- 이어서 ultralytics를 설치한다
- `pip install ultralytics`

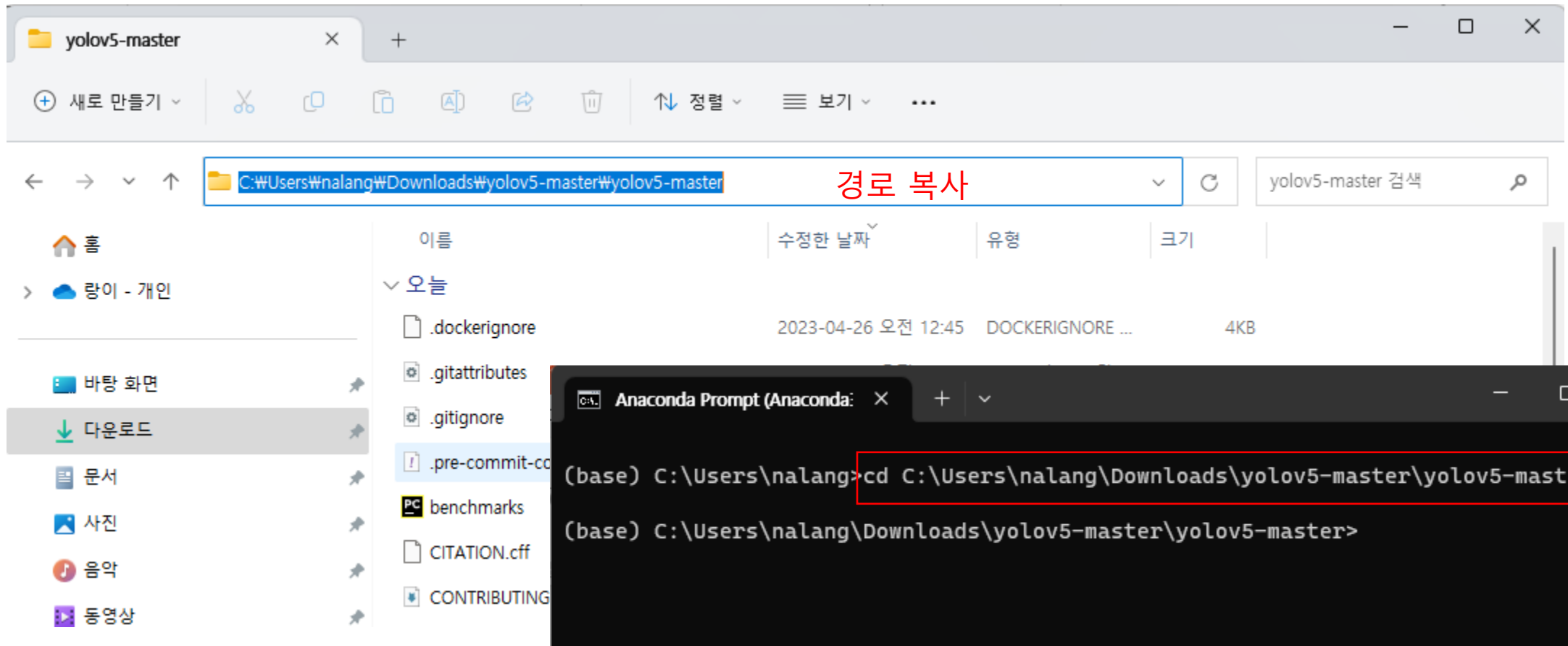
YOLO v5 다운로드

- 다음의 경로에서 모델을 다운로드한다
- <https://github.com/ultralytics/yolov5>
- C:\w의 원하는 경로에 받은 파일의 압축을 푼다
- 여기서는 C:\w 사용자 계정의 다운로드 폴더에 압축을 푼다



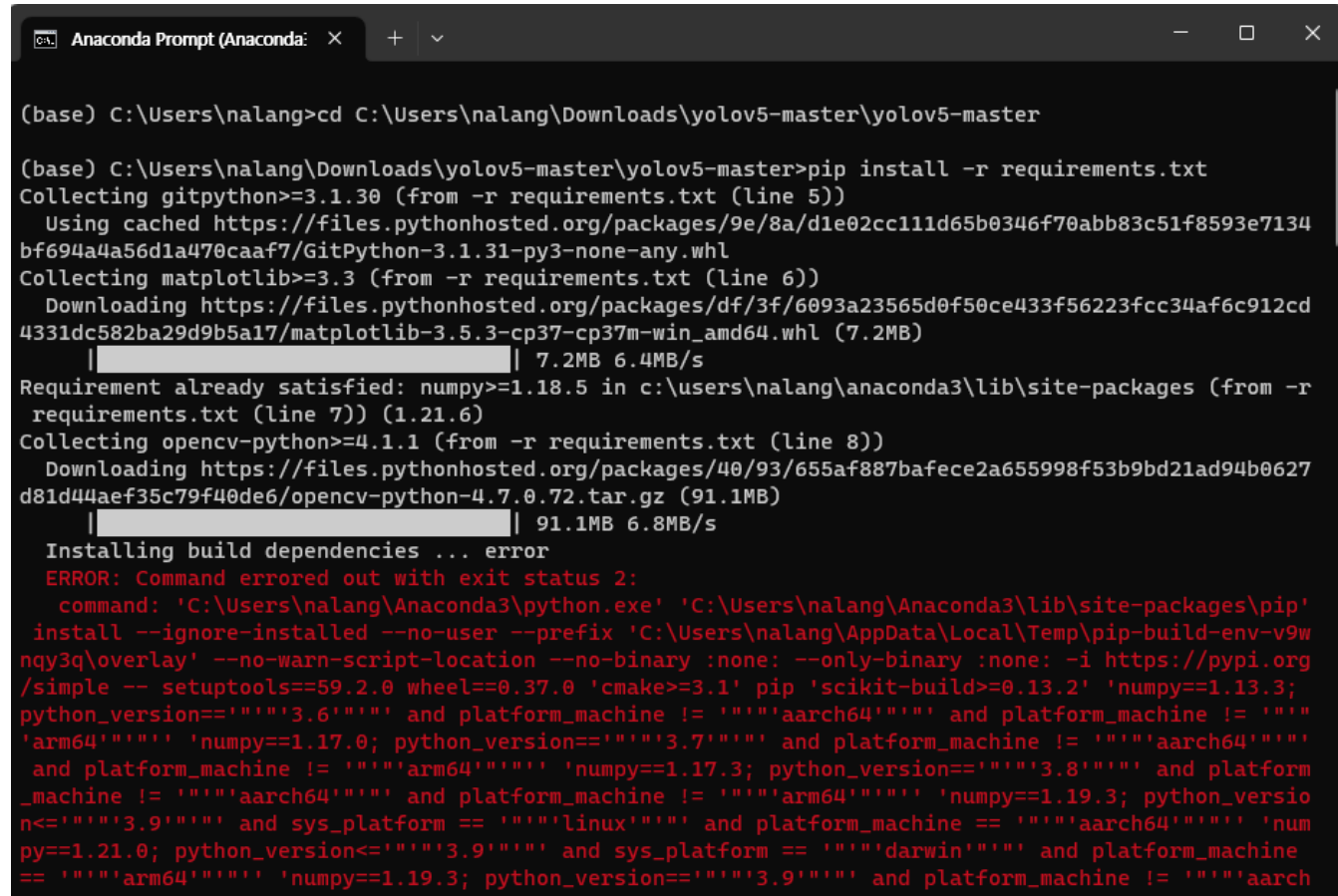
설치 시도

- 아나콘다 프롬프트를 열고, 압축이 풀린 폴더까지 진입한다



설치 시도

- 아나콘다 프롬프트에 다음의 명령어를 입력한다
- `pip install -r requirements.txt`



```
(base) C:\Users\nalang>cd C:\Users\nalang\Downloads\yolov5-master\yolov5-master

(base) C:\Users\nalang\Downloads\yolov5-master\yolov5-master>pip install -r requirements.txt
Collecting gitpython>=3.1.30 (from -r requirements.txt (line 5))
  Using cached https://files.pythonhosted.org/packages/9e/8a/d1e02cc11d65b0346f70abb83c51f8593e7134bf694a4a56d1a470caaf7/GitPython-3.1.31-py3-none-any.whl
Collecting matplotlib>=3.3 (from -r requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/df/3f/6093a23565d0f50ce433f56223fcc34af6c912cd4331dc582ba29d9b5a17/matplotlib-3.5.3-cp37-cp37m-win_amd64.whl (7.2MB)
    |████████████████████| 7.2MB 6.4MB/s
Requirement already satisfied: numpy>=1.18.5 in c:\users\nalang\anaconda3\lib\site-packages (from -r requirements.txt (line 7)) (1.21.6)
Collecting opencv-python>=4.1.1 (from -r requirements.txt (line 8))
  Downloading https://files.pythonhosted.org/packages/40/93/655af887bafec2a655998f53b9bd21ad94b0627d81d44aef35c79f40de6/opencv-python-4.7.0.72.tar.gz (91.1MB)
    |████████████████████| 91.1MB 6.8MB/s
Installing build dependencies ... error
ERROR: Command errored out with exit status 2:
   command: 'C:\Users\nalang\Anaconda3\python.exe' 'C:\Users\nalang\Anaconda3\lib\site-packages\pip'
  install --ignore-installed --no-user --prefix 'C:\Users\nalang\AppData\Local\Temp\pip-build-env-v9wnqy3q\overlay' --no-warn-script-location --no-binary :none: --only-binary :none: -i https://pypi.org/
simple -- setuptools==59.2.0 wheel==0.37.0 'cmake>=3.1' pip 'scikit-build>=0.13.2' 'numpy==1.13.3;
python_version>='3.6'' and platform_machine != 'aarch64' and platform_machine != 'arm64' 'numpy==1.17.0;
python_version>='3.7' and platform_machine != 'aarch64' and platform_machine != 'arm64' 'numpy==1.17.3;
python_version>='3.8' and platform_machine != 'aarch64' and platform_machine != 'arm64' 'numpy==1.19.3;
python_version<='3.9' and sys_platform == 'linux' and platform_machine != 'aarch64' 'numpy==1.21.0;
python_version<='3.9' and sys_platform == 'darwin' and platform_machine != 'arm64' 'numpy==1.19.3;
python_version>='3.9' and platform_machine != 'aarch64'
```

에러 메시지를 보았을 것이다

설치 시도

- 아나콘다 프롬프트에 다음의 명령어를 입력한다
- [pip install git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI](https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI)

```
Anaconda Prompt (Anaconda: x + v)

(base) C:\Users\nalang\Downloads\yolov5-master\yolov5-master>pip install git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI
Collecting git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI
  Cloning https://github.com/philferriere/cocoapi.git to c:\users\nalang\appdata\local\temp\pip-req-build-yey7tdr7
    Running command git clone -q https://github.com/philferriere/cocoapi.git 'C:\Users\nalang\AppData\Local\Temp\pip-req-build-yey7tdr7'
  Building wheels for collected packages: pycocotools
    Building wheel for pycocotools (setup.py) ... done
    Created wheel for pycocotools: filename=pycocotools-2.0-cp37-cp37m-win_amd64.whl size=91147 sha256=6d962a33b65a38f8fb9cacd5f55a87f9ffe10b5e4062addee50ed3fc60f8e11b
    Stored in directory: C:\Users\nalang\AppData\Local\Temp\pip-ephem-wheel-cache-aapxu26o\wheels\69\2b\12\2fa959e49f73d26cff202c2f4e5079096c9c57c8a8509fd75c
  Successfully built pycocotools
  Installing collected packages: pycocotools
ERROR: Exception:
Traceback (most recent call last):
  File "C:\Users\nalang\Anaconda3\lib\site-packages\pip\_internal\cli\base_command.py", line 188, in main
    status = self.run(options, args)
  File "C:\Users\nalang\Anaconda3\lib\site-packages\pip\_internal\commands\install.py", line 407, in run
    use_user_site=options.use_user_site,
  File "C:\Users\nalang\Anaconda3\lib\site-packages\pip\_internal\req\__init__.py", line 58, in install_given_reqs
    **kwargs
  File "C:\Users\nalang\Anaconda3\lib\site-packages\pip\_internal\req\req_install.py", line 928, in install
    use_user_site=use_user_site, pycompile=pycompile,
```

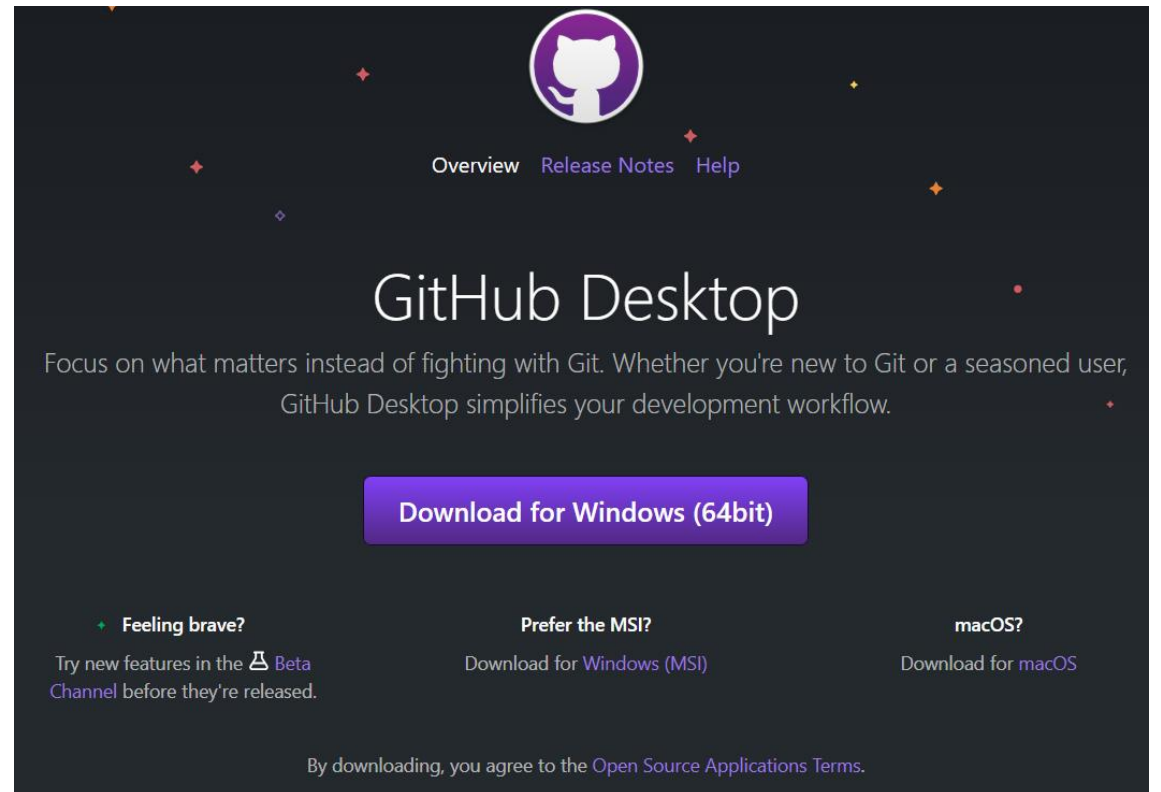
또 에러 메시지를 보았을 것이다

깃헙 데스크탑 설치

- 또는 깃헙 관련 에러 메시지를 보았다면 GitHub Desktop을 설치한다

- <https://desktop.github.com/>

에서 다운로드 받아 실행한다



깃헙 데스크탑 설치



Welcome to GitHub Desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Sign in below to get started with your existing projects.

[Sign in to GitHub.com](#)

[Sign in to GitHub Enterprise](#)

New to GitHub? [Create your free account.](#)

Skip this step

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#).

GitHub Desktop sends usage metrics to improve the product and inform feature decisions. [Learn more about user metrics.](#)

Sign in to GitHub
to continue to GitHub Desktop

Username or email address
lingua72@gmail.com

Password
..... [Forgot password?](#)

[Sign in](#)

New to GitHub? [Create an account.](#)

Device verification

Email

We just sent your authentication code via email to l*****@gmail.com. The code will expire at 8:36AM KST.

Device Verification Code
XXXXXX

[Verify](#)

Configure Git

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

☒ Use my GitHub account name and email address

☐ Configure manually

Name

SukjaeChoi

Email

lingua72@gmail.com

[Finish](#)

[Cancel](#)

Example commit

Fix all the things

SukjaeChoi • 30 minutes ago



File Edit View Repository Branch Help

Let's get started!

Add a repository to GitHub Desktop to start collaborating

[Create a tutorial repository...](#)

Filter your repositories

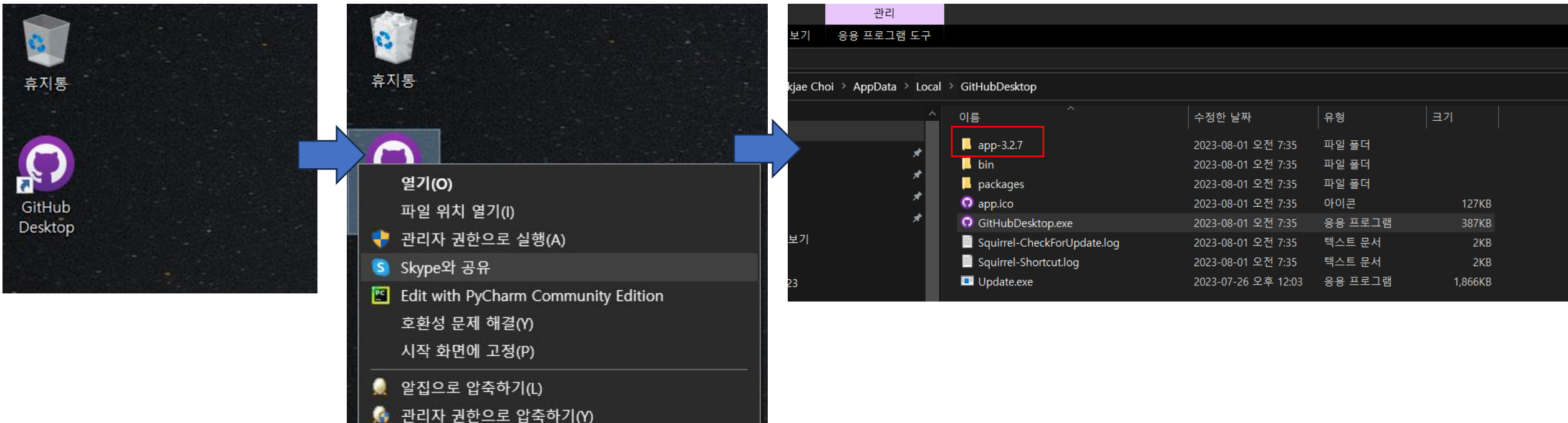
[Refresh](#)

Your repositories

- SukjaeChoi/HelloWorld
- SukjaeChoi/RHINO



파일 위치 열기

- 방금 설치한 GitHub Desktop 을 닫는다
- 바탕화면에 있는 바로가기 버튼을 우클릭하여 파일위치 열기를 누른다
- app-3.2.7 (버전명은 다를 수 있음) 폴더로 진입한다



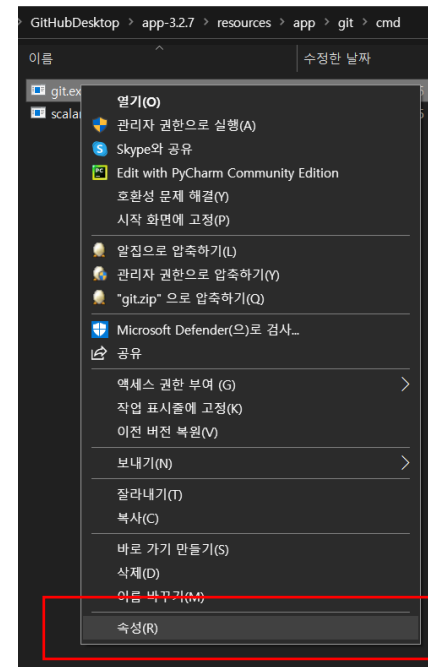
git.exe

- 다음 위치까지 들어간다
- C:\Users\lingu\AppData\Local\GitHubDesktop\app-3.2.7\resources\app\git\cmd

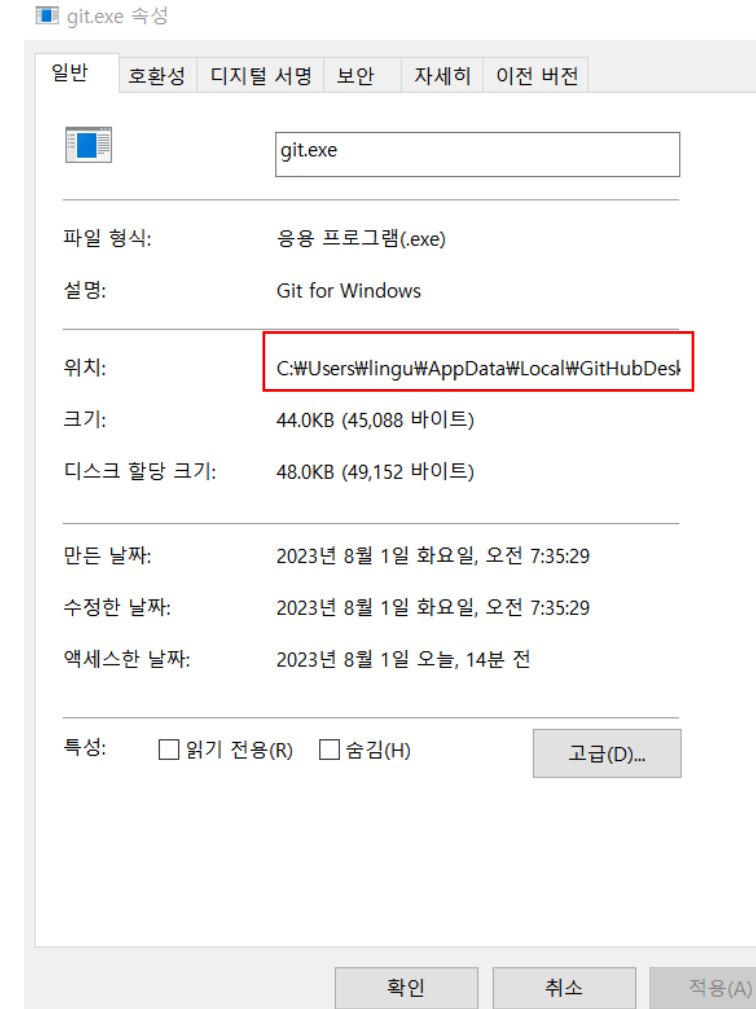
> GitHubDesktop > app-3.2.7 > resources > app > git > cmd				
이름	수정한 날짜	유형	크기	
 git.exe	2023-08-01 오전 7:35	응용 프로그램	45KB	
 scalar.exe	2023-08-01 오전 7:35	응용 프로그램	45KB	

git.exe

- git.exe 파일의 '속성(R)'을 클릭하고,

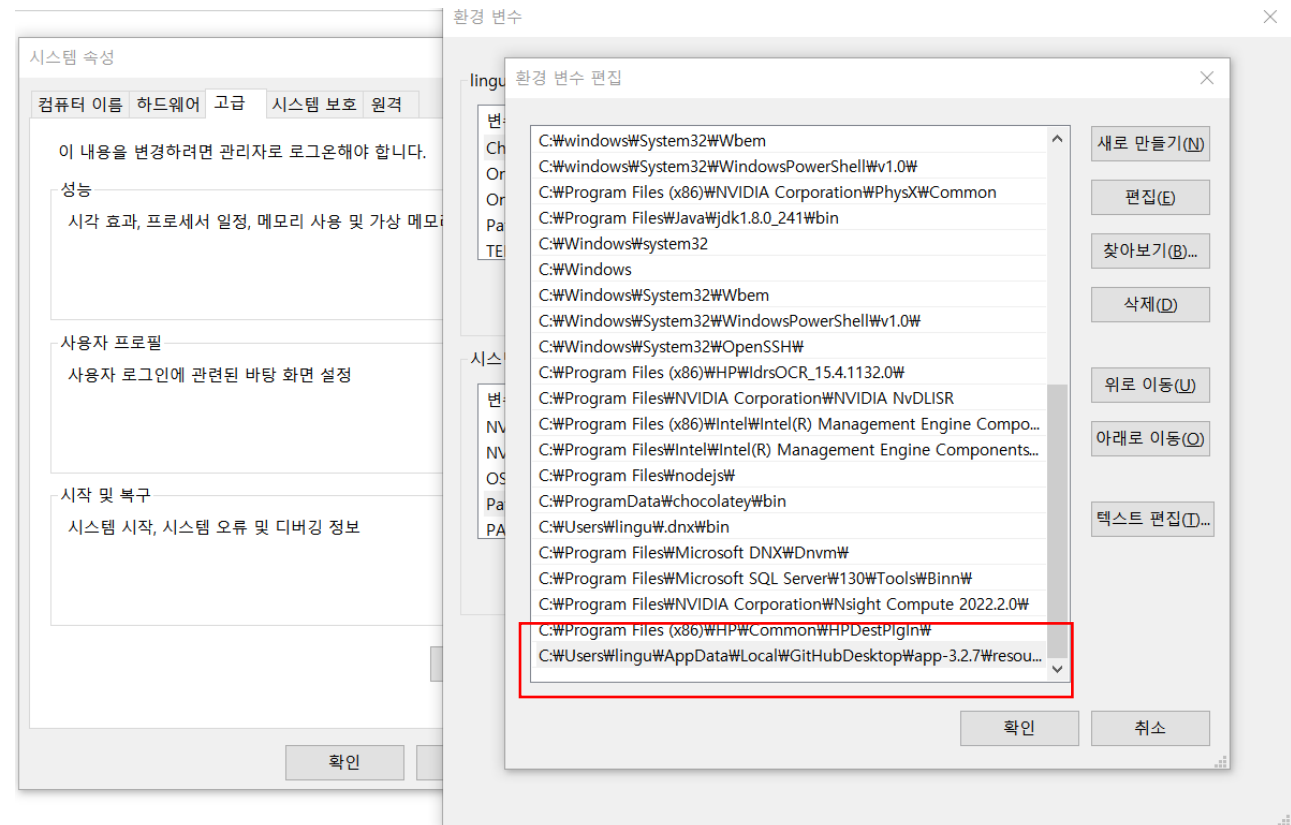


- '위치' 내용을 복사한다
- C:\Users\lingu\AppData\Local\GitHubDesktop\app-3.2.7\resources\app\git\cmd



환경변수 등록

- 내 pc(내컴퓨터) 우클릭 > 속성 > 고급 시스템 설정 > 고급 탭 > 환경변수 > 시스템 변수의 path 항목에
- 복사한 경로를 추가한다
- 모두 '확인'을 눌러 닫는다



깃헙 버전 확인

- cmd 창에서 `git --version` 을 입력하여 버전이 나오는지 확인한다

```
C:\> 명령 프롬프트
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

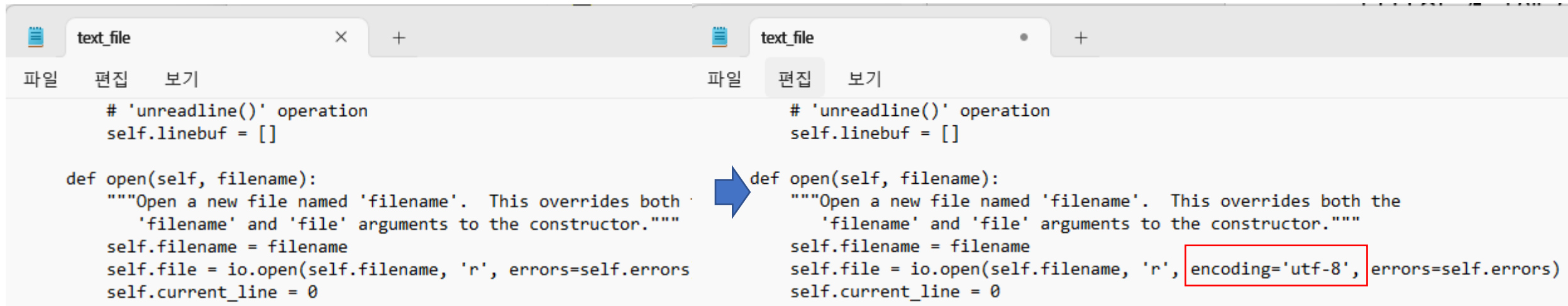
C:\Users\lingu>git --version
git version 2.39.3.windows.1

C:\Users\lingu>
```



인코딩 설정 1

- 다음의 경로에 있는 text_file.py 파일을 메모장 등으로 연다
- C:\Users\<사용자계정명>\Anaconda3\Lib\distutils 또는 C:\Anaconda3\Lib\distutils
- 115 줄에 있는 open() 함수 부분에 encoding='utf-8', 을 입력한다



The image shows a side-by-side comparison of a Python file named 'text_file.py'. On the left, the original code is shown. On the right, the code is modified to include an encoding parameter. A blue arrow points from the original code to the modified code, indicating the change.

```
# 'unreadline()' operation
self.linebuf = []

def open(self, filename):
    """Open a new file named 'filename'. This overrides both
    'filename' and 'file' arguments to the constructor."""
    self.filename = filename
    self.file = io.open(self.filename, 'r', errors=self.errors)
    self.current_line = 0
```

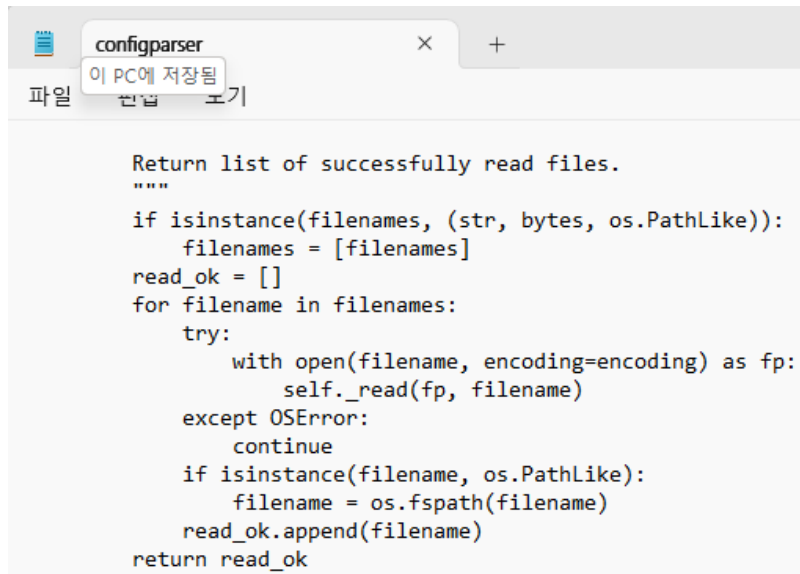
→

```
# 'unreadline()' operation
self.linebuf = []

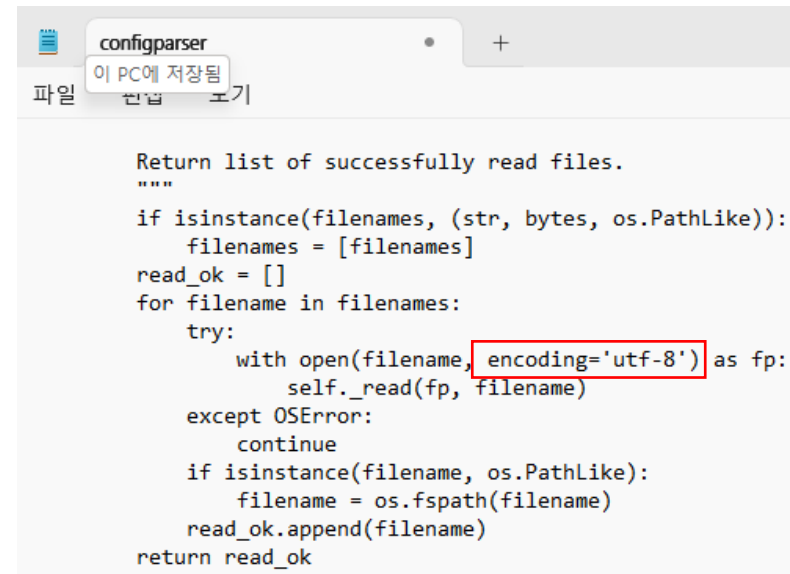
def open(self, filename):
    """Open a new file named 'filename'. This overrides both the
    'filename' and 'file' arguments to the constructor."""
    self.filename = filename
    self.file = io.open(self.filename, 'r', encoding='utf-8', errors=self.errors)
    self.current_line = 0
```


인코딩 설정 2

- 다음의 경로에 있는 configparser.py 파일을 메모장 등으로 연다
- C:\Users\<사용자계정명>\Anaconda3\Lib
- 695 줄에 있는 open() 함수 부분에 encoding=encoding을 encoding='utf-8'으로 수정한다



```
Return list of successfully read files.
"""
if isinstance(filename, (str, bytes, os.PathLike)):
    filenames = [filename]
read_ok = []
for filename in filenames:
    try:
        with open(filename, encoding=encoding) as fp:
            self._read(fp, filename)
    except OSError:
        continue
    if isinstance(filename, os.PathLike):
        filename = os.fspath(filename)
    read_ok.append(filename)
return read_ok
```



```
Return list of successfully read files.
"""
if isinstance(filename, (str, bytes, os.PathLike)):
    filenames = [filename]
read_ok = []
for filename in filenames:
    try:
        with open(filename, encoding='utf-8') as fp:
            self._read(fp, filename)
    except OSError:
        continue
    if isinstance(filename, os.PathLike):
        filename = os.fspath(filename)
    read_ok.append(filename)
return read_ok
```

설치

- 아나콘다 프롬프트를 새로 열고,
- C:\Users\<사용자계정>\Downloads\yolov5-master 경로로 다시 진입한 후,
- 다음의 명령어를 다시 입력한다
- [pip install git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI](https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI)

이번에는 성공!

```
Anaconda Prompt (Anaconda3)

(base) C:\Users\lingu>cd C:\Users\lingu\Downloads\yolov5-master

(base) C:\Users\lingu\Downloads\yolov5-master>pip install git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI
Collecting git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI
  Cloning https://github.com/philferriere/cocoapi.git to c:\users\lingu\appdata\local\temp\pip-req-build-c9nmfiol
  Running command git clone -q https://github.com/philferriere/cocoapi.git 'C:\Users\lingu\AppData\Local\Temp\pip-req-build-c9nmfiol'
Building wheels for collected packages: pycocotools
  Building wheel for pycocotools (setup.py) ... done
  Created wheel for pycocotools: filename=pycocotools-2.0-cp37-cp37m-win_amd64.whl size=71737 sha256=35d24940b9ecd5f3811192f5cfb2b0598f081d4a032db6848a6894dbea6e57d
  Stored in directory: C:\Users\lingu\AppData\Local\Temp\pip-ephem-wheel-cache-lp5hww73\wheels\69\2b\12\2fa959e49f73d26c
ff202c2f4e5079096c9c57c8a8509fd75c
Successfully built pycocotools
Installing collected packages: pycocotools
Successfully installed pycocotools-2.0

(base) C:\Users\lingu\Downloads\yolov5-master>
```

설치

- 아나콘다 프롬프트에 다음의 명령어를 다시 입력한다
- `pip install -r requirements.txt`

한참 진행이 이루어진다

```
Anaconda Prompt (Anaconda: x + v

Stored in directory: C:\Users\nalang\AppData\Local\Temp\pip-ephem-wheel-cache-cuvkd21t\wheels\69\2
b\12\2fa959e49f73d26cff202c2f4e5079096c9c57c8a8509fd75c
Successfully built pycocotools
Installing collected packages: pycocotools
Successfully installed pycocotools-2.0

(base) C:\Users\nalang\Downloads\yolov5-master\yolov5-master>pip install -r requirements.txt
Collecting gitpython>=3.1.30 (from -r requirements.txt (line 5))
  Using cached https://files.pythonhosted.org/packages/9e/8a/d1e02cc111d65b0346f70abb83c51f8593e7134
bf694a4a56d1a470caaf7/GitPython-3.1.31-py3-none-any.whl
Collecting matplotlib>=3.3 (from -r requirements.txt (line 6))
  Using cached https://files.pythonhosted.org/packages/df/3f/6093a23565d0f50ce433f56223fcc34af6c912c
d4331dc582ba29d9b5a17/matplotlib-3.5.3-cp37-cp37m-win_amd64.whl
Requirement already satisfied: numpy>=1.18.5 in c:\users\nalang\anaconda3\lib\site-packages (from -r
requirements.txt (line 7)) (1.21.6)
Collecting opencv-python>=4.1.1 (from -r requirements.txt (line 8))
  Using cached https://files.pythonhosted.org/packages/40/93/655af887bafec2a655998f53b9bd21ad94b062
7d81d44aef35c79f40de6/opencv-python-4.7.0.72.tar.gz
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing wheel metadata ... done
Collecting Pillow>=7.1.2 (from -r requirements.txt (line 9))
  Downloading https://files.pythonhosted.org/packages/cb/3c/4f3ef1a14e903d7b2bc43672c20f732b874e1e50
a9a58ac9a1726ef3773d/Pillow-9.5.0-cp37-cp37m-win_amd64.whl (2.5MB)
  | 2.5MB 6.8MB/s
Requirement already satisfied: psutil in c:\users\nalang\anaconda3\lib\site-packages (from -r requir
ements.txt (line 10)) (5.6.3)
Collecting PyYAML>=5.3.1 (from -r requirements.txt (line 11))
  Downloading https://files.pythonhosted.org/packages/d1/c0/4fe04181b0210ee2647cfbb89ecd10a36eef89f1
0d8aca6a192c201bbe58/PyYAML-6.0-cp37-cp37m-win_amd64.whl (153kB)
```

PyYAML 에러

- PyYAML 관련 에러 메시지를 보았다면 아래를 입력한다
- `pip install --ignore-installed PyYAML`

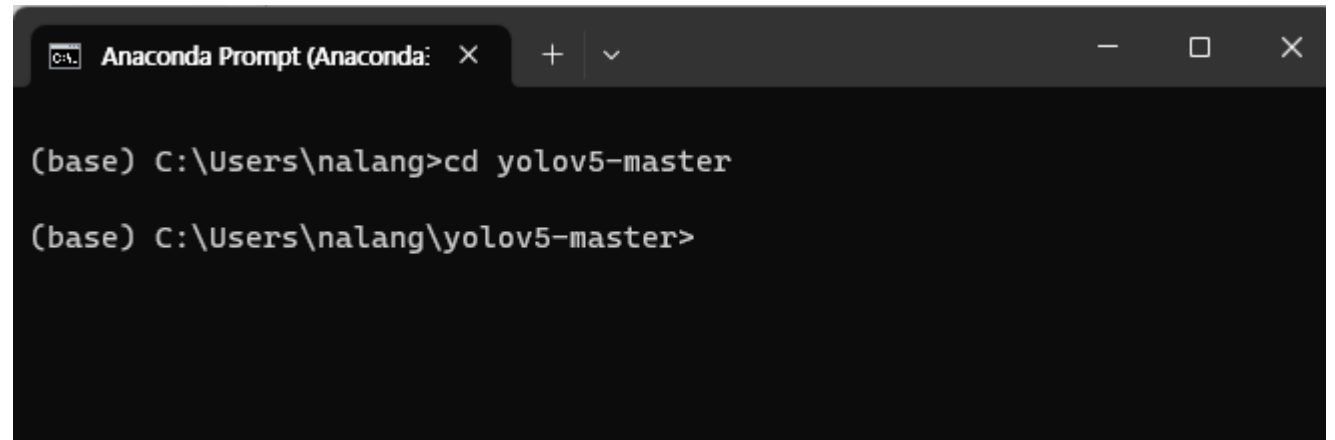
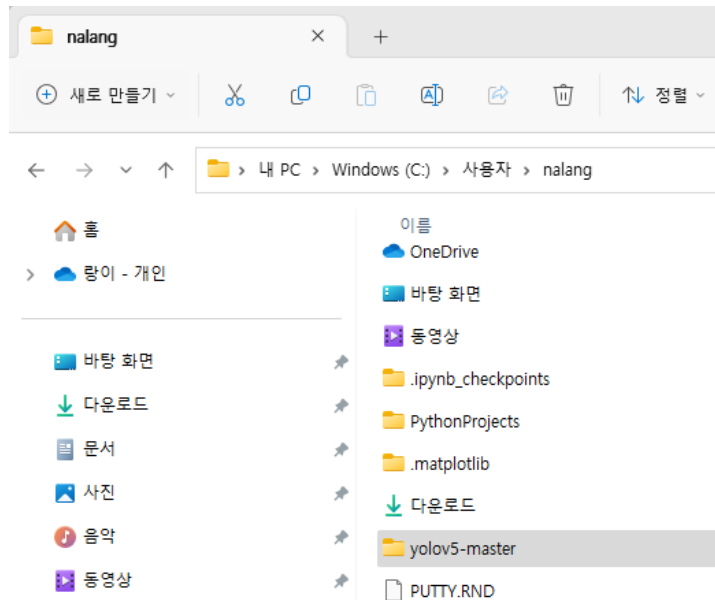
이번에는 성공!

```
Anaconda Prompt (Anaconda: x) + v
Successfully built opencv-python
ERROR: conda 4.12.0 requires ruamel_yaml_conda>=0.11.14, which is not installed.
ERROR: tensorflow 2.1.0 has requirement scipy==1.4.1; python_version >= "3", but you'll have scipy 1
.7.3 which is incompatible.
ERROR: tensorflow 2.1.0 has requirement tensorboard<2.2.0,>=2.1.0, but you'll have tensorboard 2.9.1
which is incompatible.
ERROR: tensorflow 2.1.0 has requirement tensorflow-estimator<2.2.0,>=2.1.0rc0, but you'll have tenso
rflow-estimator 2.9.0 which is incompatible.
ERROR: tensorboard 2.9.1 has requirement google-auth-oauthlib<0.5,>=0.4.1, but you'll have google-au
th-oauthlib 0.5.2 which is incompatible.
Installing collected packages: smmap, gitdb, gitpython, fonttools, Pillow, matplotlib, opencv-python
, PyYAML, requests, scipy, thop, seaborn, setuptools
Found existing installation: Pillow 6.2.0
Uninstalling Pillow-6.2.0:
Successfully uninstalled Pillow-6.2.0
Found existing installation: matplotlib 3.1.1
Uninstalling matplotlib-3.1.1:
Successfully uninstalled matplotlib-3.1.1
Found existing installation: PyYAML 5.1.2
ERROR: Cannot uninstall 'PyYAML'. It is a distutils installed project and thus we cannot accurately
determine which files belong to it which would lead to only a partial uninstall.

(base) C:\Users\nalang\Downloads\yolov5-master\yolov5-master>pip install --ignore-installed PyYAML
Collecting PyYAML
Using cached https://files.pythonhosted.org/packages/d1/c0/4fe04181b0210ee2647cfbb89ecd10a36eef89f
10d8aca6a192c201bbe58/PyYAML-6.0-cp37m-win_amd64.whl
Installing collected packages: PyYAML
Successfully installed PyYAML-6.0
```

폴더 복사

- 안쪽의 yolov5-master 폴더를 원하는 위치에 옮길 수 있다
- 예를 들어, C:\Users\W <사용자계정> 폴더로 옮긴다
- 먼저 아나콘다 프롬프트를 닫아야 한다
- 옮긴 뒤에, 아나콘다 프롬프트를 열고, 해당 위치로 이동한다 yolo5가 설치된 위치에서 실행해야 함



WebCam에서 사용

- 웹캠에서 YOLO를 사용해본다
- 아나콘다 프롬프트에 다음의 명령어를 입력한다
- `python detect.py --source 0` # 하이픈 2번
- 웹캠이 하나라면 `source 0` 이고,
- 웹캠이 둘이라면 `source 1` 을 입력할 수 있다

WebCam에서 사용

※ YOLOv8에서도

<https://github.com/ultralytics/yolov5> 에서 파일을 다운로드 받아 같은 방식으로 실행하면 WebCam에서 동작하게 할 수 있다

- 종료는 'q' 버튼 입력

