

세션 1

실습을 위한 환경설정

안내

파이썬 프로그래밍 환경 만들기

최근 프로그래밍 개발 소프트웨어는 코딩 편집과 디버깅을 돕기 위한 기능은 물론 소스 버전 관리, 원격 개발, 협업 개발까지 지원하고 있다. 하나의 개발 툴로 여러가지 프로그래밍 언어를 이용한 개발도 가능하고, 다른 플랫폼의 소프트웨어를 개발할 수도 있다. 이러한 소프트웨어를 IDE(Integrated Development Environment)라고 하는데, 처음 프로그래밍을 공부하는 개발자가 어떤 IDE를 선택할 지 결정하는 것은 쉽지 않은 일이다. IDE 소프트웨어는 종류도 여러가지가 있고 프로그래밍 환경을 만드는 방법도 너무 다양하기 때문에 모든 방법을 설명할 수도 없을 뿐 아니라 한 가지 방법 만이 최상의 환경이라고 이야기할 수도 없다.

본 교육 프로그램에서는 윈도우 운영체제에서 파이썬 개발하기 위한 가장 기본적인 개발환경을 추천하려고 한다. 파이썬 소프트웨어를 직접 설치하기도 하지만 파이썬 언어의 특성상 소프트웨어 개발 범위가 확장될 때마다 여러가지 패키지를 추가적으로 설치해야 한다. 그런데 문제는 파이썬 버전과 패키지들 간의 버전 차로 인해 여러가지 문제가 자주 발생하게 된다. 이러한 문제를 최소화하기 위해 파이썬과 관련 패키지들을 묶어서 통합적으로 버전 관리를 하는 소프트웨어가 아나콘다(Anaconda)이다. 본 교육 프로그램에서는 아나콘다를 설치하여 파이썬과 관련 패키지를 통합적으로 설치하도록 하겠다. 아나콘다를 설치하게 되면 파이썬 프로그래밍을 위한 IDE 기능을 갖는 소프트웨어 툴도 함께 설치된다. 그러나 본 교육 프로그램에서는 윈도우 운영체제 초창기부터 소프트웨어 개발 툴로 개발자들이 오래 전부터 많이 활용하고 있는 마이크로소프트사의 비주얼스튜디오(Visual Studio)를 사용하려고 한다. 마지막으로 인공지능 패키지로서 텐서플로우를 아나콘다 환경으로 설치하려고 한다.

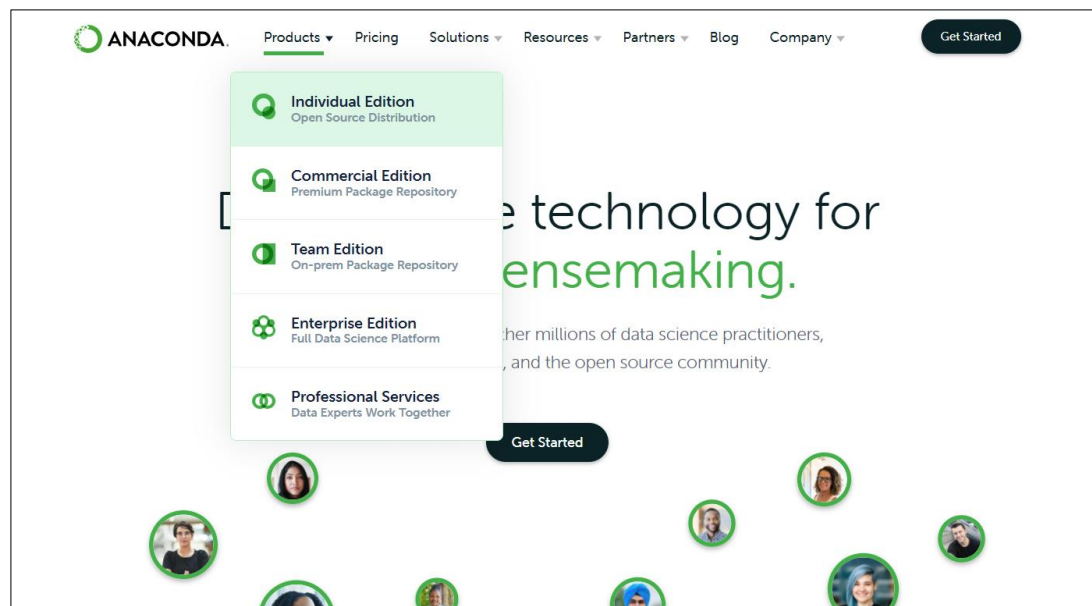
- 아나콘다 (Anaconda 4.10.3)
- 비주얼 스튜디오 2022 커뮤니티 (Visual Studio 2022 Community)
- 텐서플로우 (TensorFlow 2.8.0)

1. **아나콘다(Anaconda)**
2. 비주얼 스튜디오(Visual Studio)
3. 텐서플로우(TensorFlow)
4. 텐서플로우(TensorFlow)
Object Detection API 설치

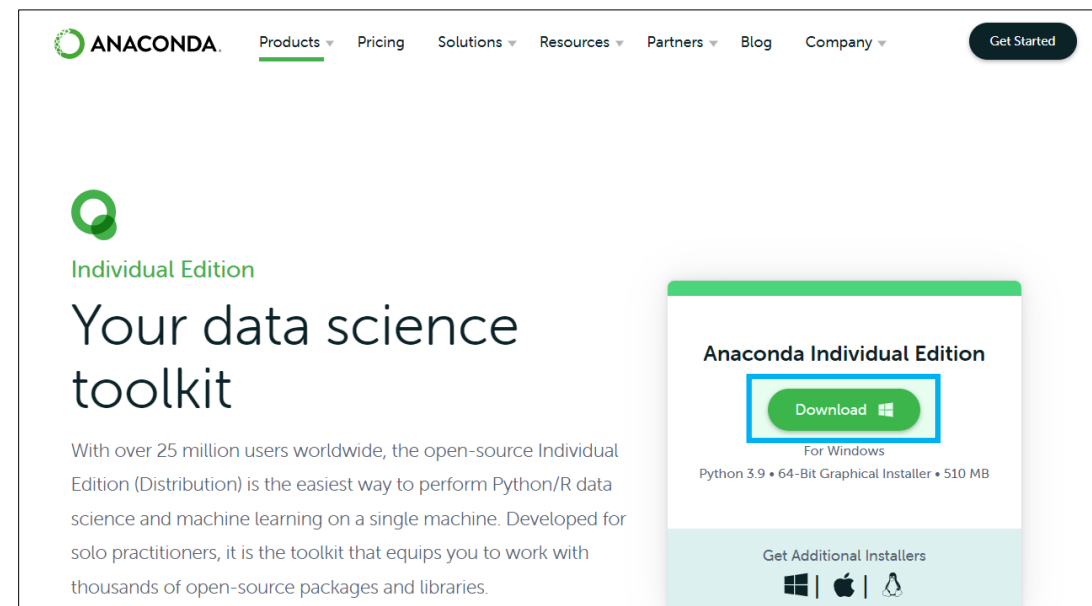
1.1 아나콘다 설치 프로그램 다운받기

1.1.1 웹브라우저에서 아나콘다 홈페이지 접속 (anaconda.com)

1.1.2 'Products > Individual Edition' 메뉴 선택



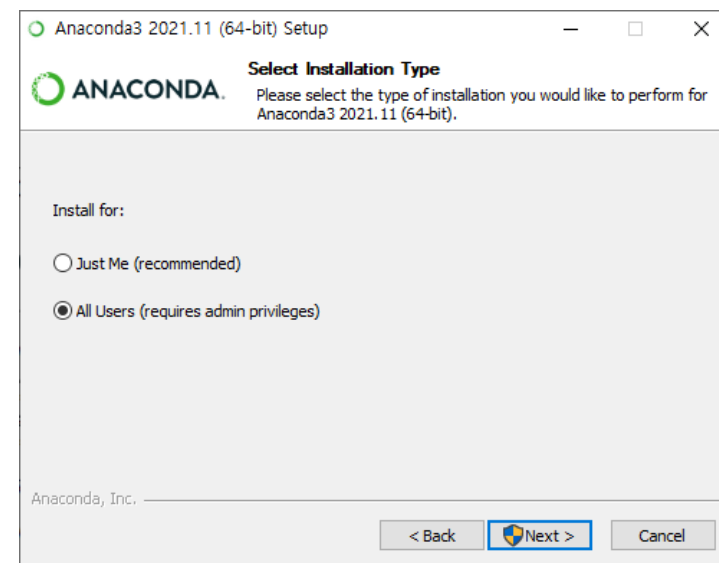
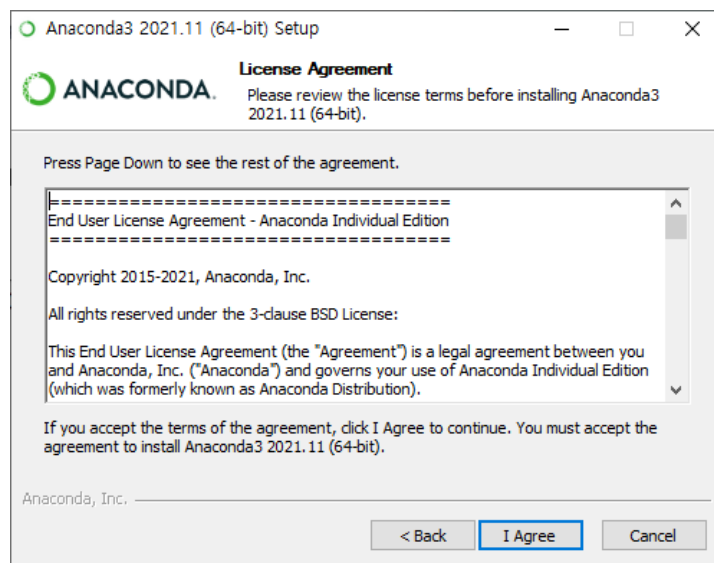
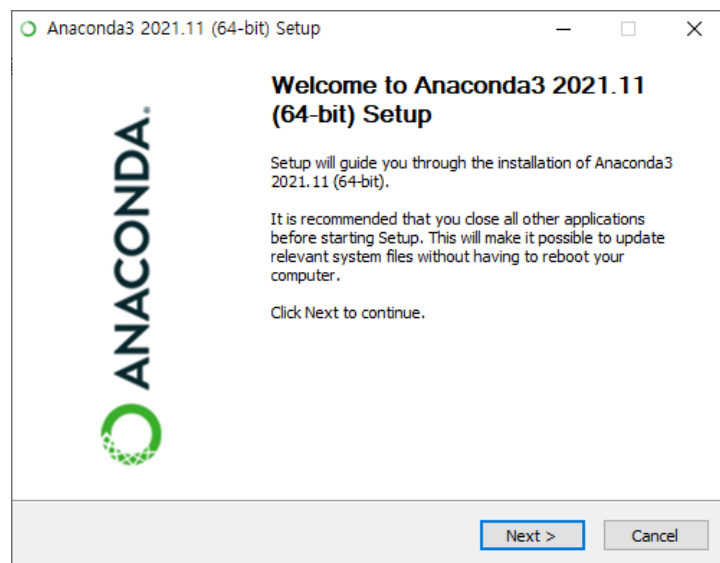
1.1.3 'Download' 버튼 클릭



1.2 Anaconda 설치

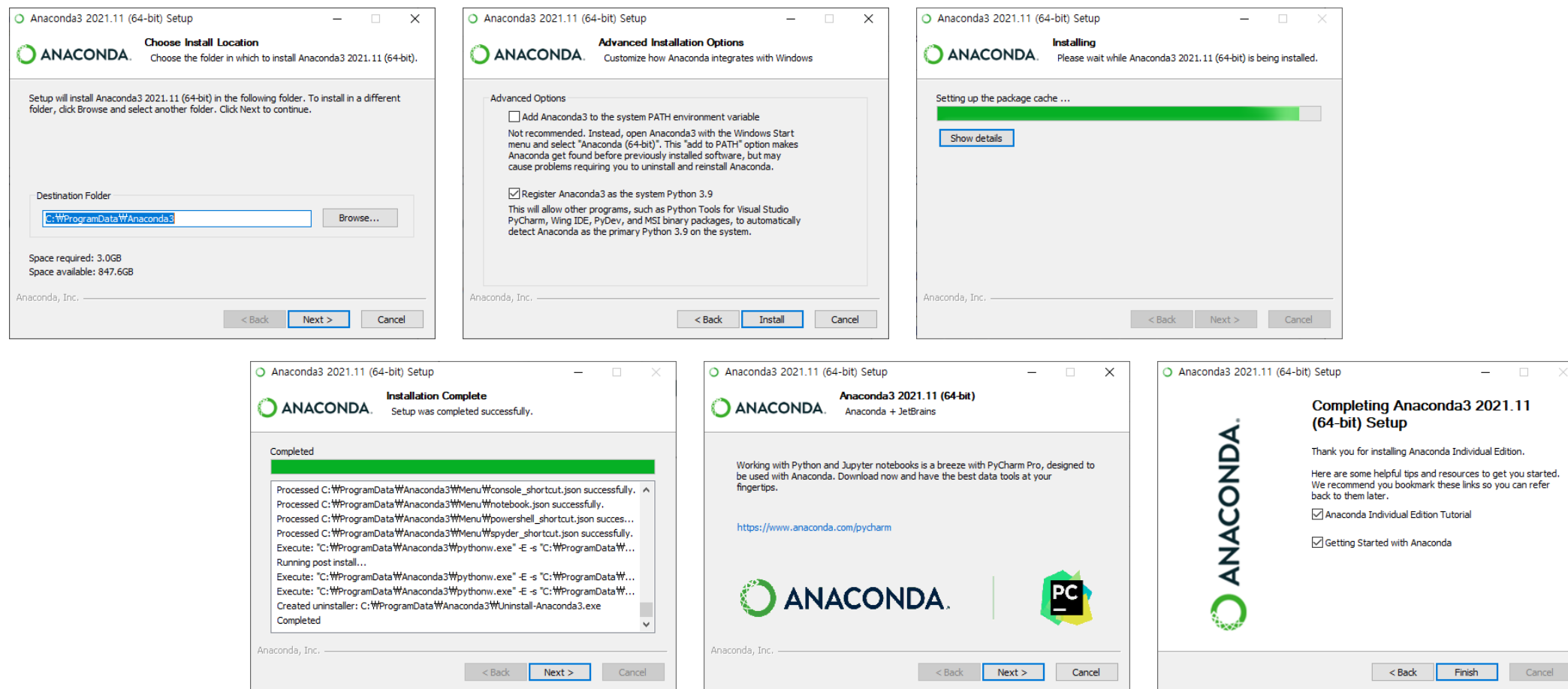
1.2.1 'Anaconda3-2021.11-Windows-x86_64.exe' 클릭

1.2.2 Anaconda 설치 프로그램의 설치 안내에 따라 설치 (1)



1.2 Anaconda 설치

1.2.2 Anaconda 설치 프로그램의 설치 안내에 따라 설치 (2)

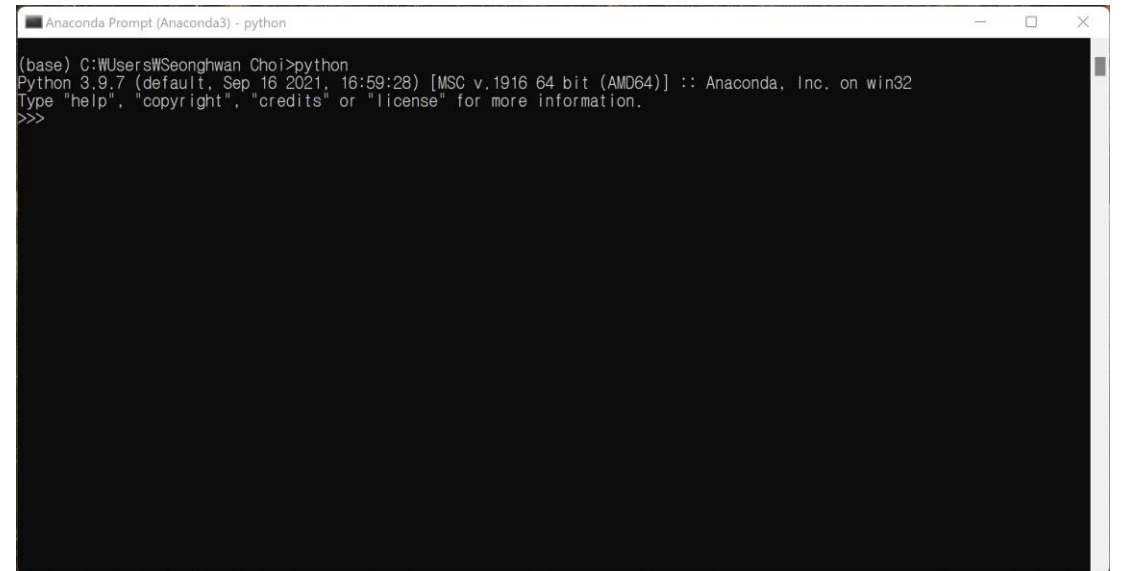
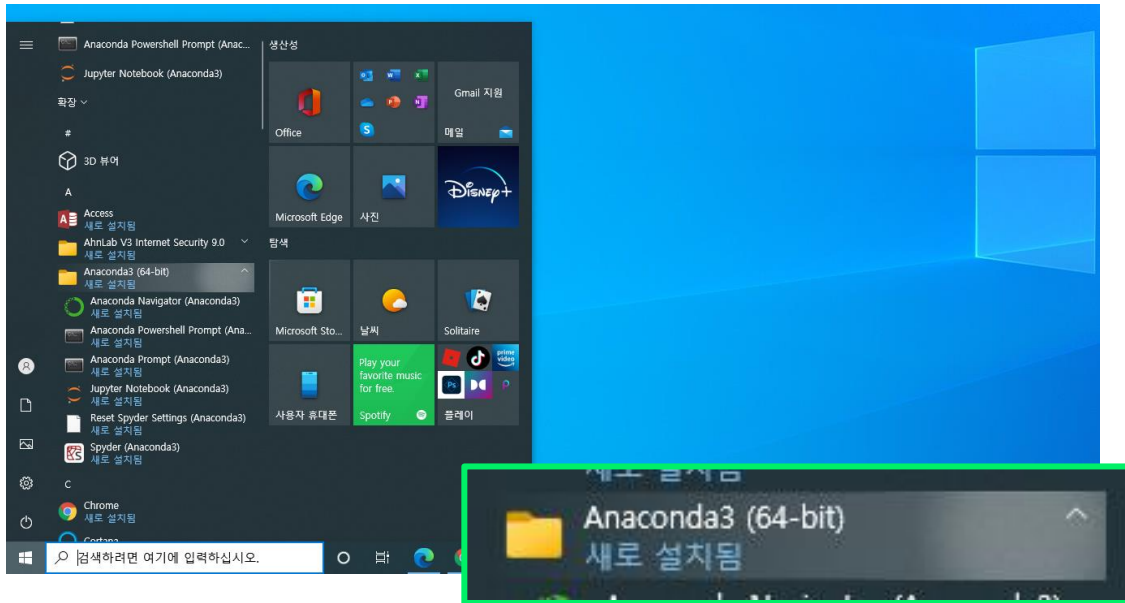


1.3 Anaconda 설치 확인하기

1.3.1 윈도우 시작(Start) 메뉴에서 'Anaconda 3' 설치 확인

1.3.2 Anaconda Prompt (Anaconda 3) 실행

1.3.3 'python' 실행



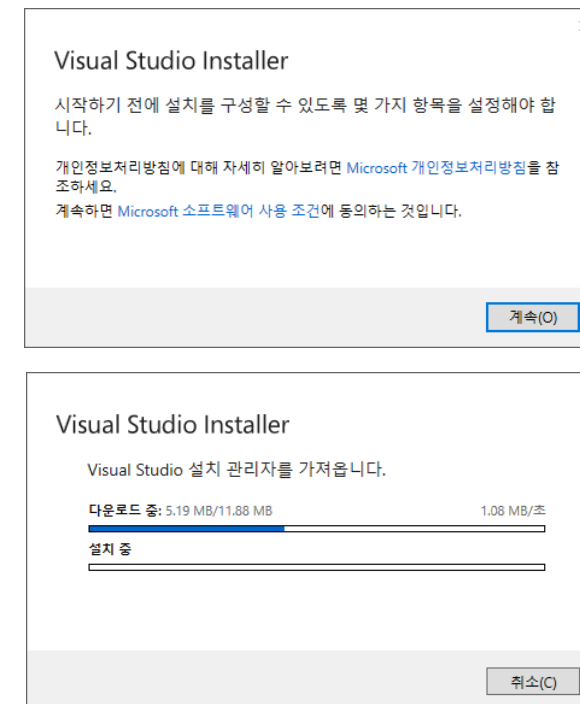
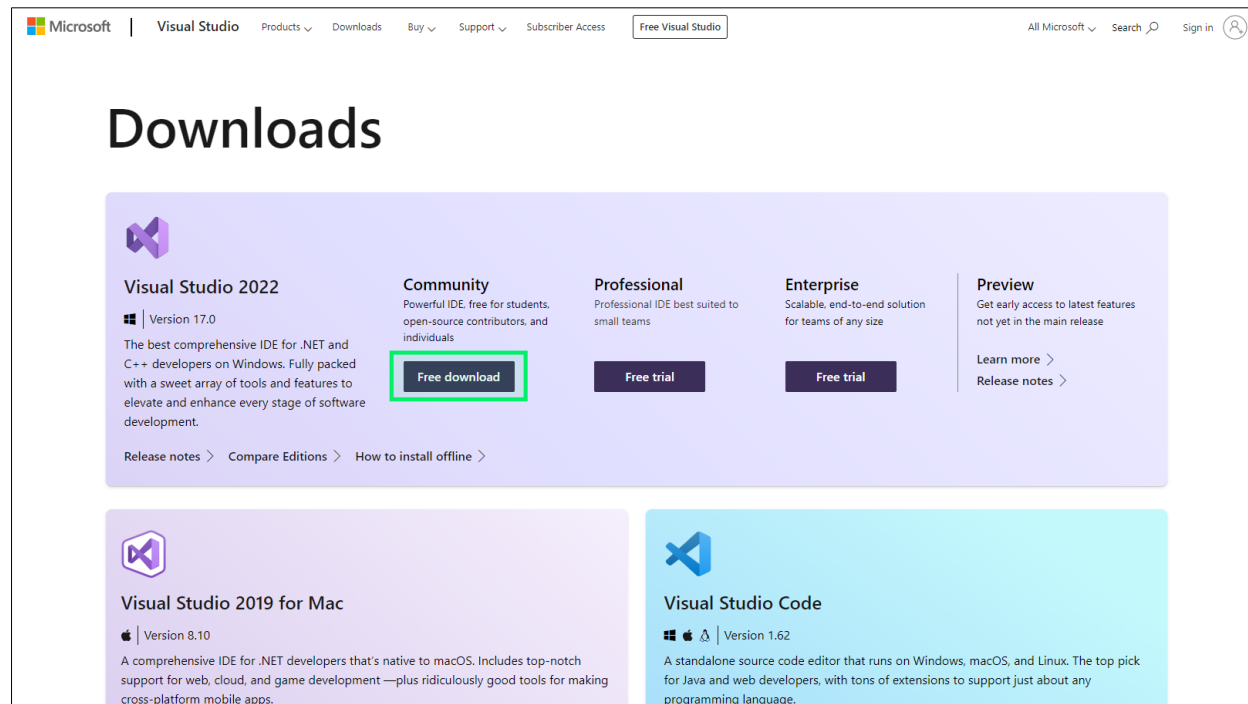
1. 아나콘다(Anaconda)
2. **비주얼 스튜디오(Visual Studio)**
3. 텐서플로우(TensorFlow)
4. 텐서플로우(TensorFlow)
Object Detection API 설치

2.1 비주얼 스튜디오 다운받기

2.1.1 웹브라우저에서 비주얼 스튜디오 웹사이트 접속하기 (visualstudio.microsoft.com)

2.1.2 Community 버전을 다운받기 위해 ‘Free download’ 버튼을 클릭

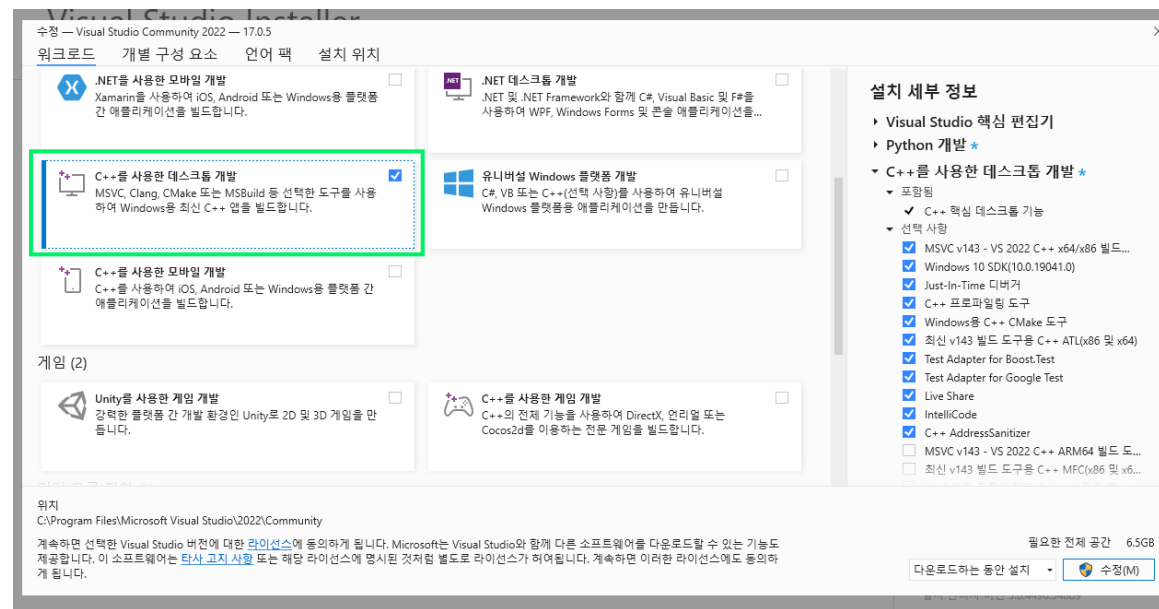
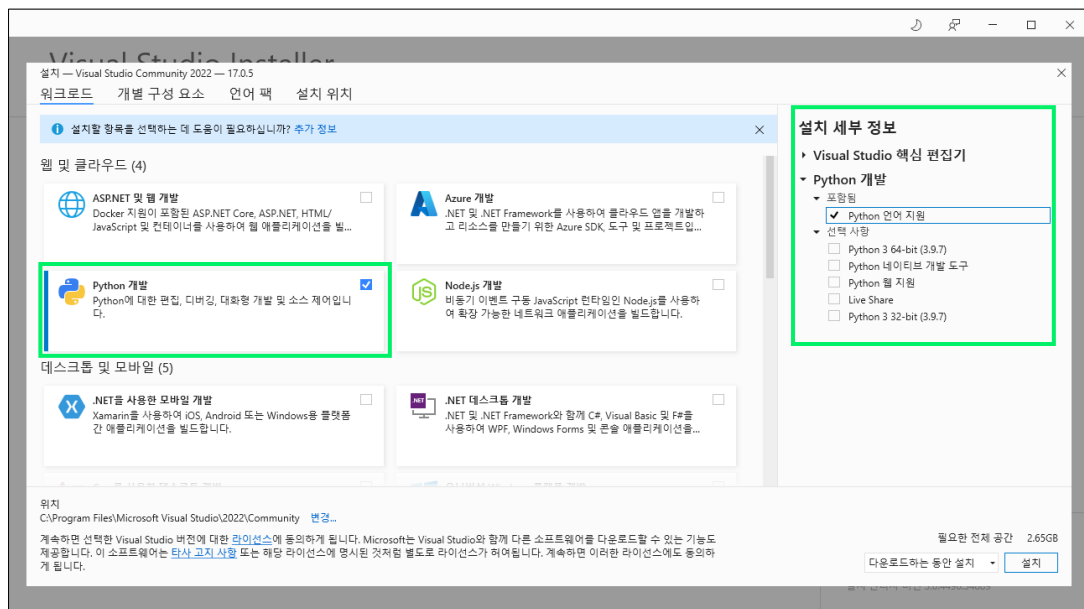
2.1.3 다운받은 VisualStudioSetup.exe 프로그램을 실행하여 비주얼 스튜디오 설치 프로그램을 시작



2.2 비주얼 스튜디오 설치

2.2.4 ‘설치 관리자’ 다운로드가 완료되면 설치 옵션을 설정하기 위한 윈도우가 열림(1)

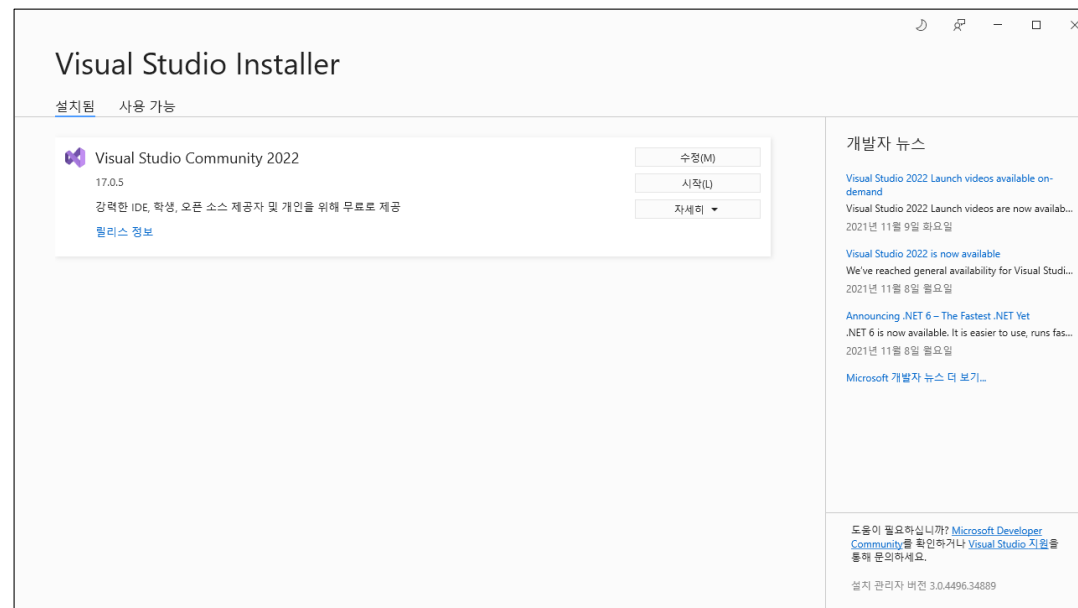
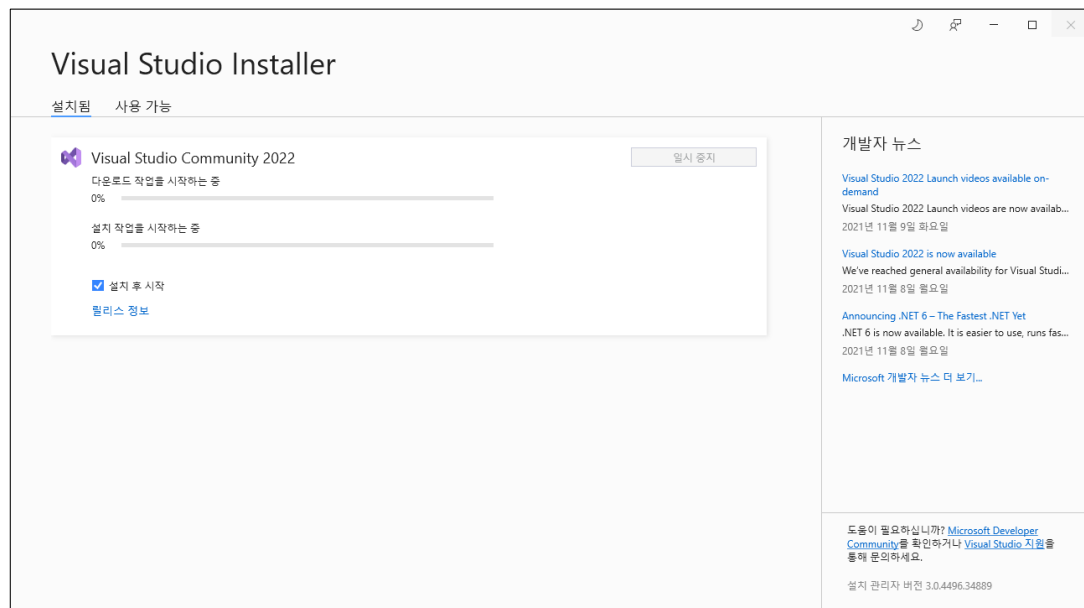
- ‘워크로드’에서 ‘웹 및 클라우드’의 ‘Python 개발’ 옵션을 체크
- 우측 ‘설치 세부 정보’에서 ‘Python 언어 지원’이 기본으로 체크되어 있고, ‘Python 3 xx-bit’ 등 다른 옵션은 언체크
- 이미 아나콘다를 설치했기 때문에 별도의 파이썬 소프트웨어를 설치할 필요가 없음
- 마지막으로 우측 하단의 ‘설치’ 버튼을 클릭 C++ 선택 필수



2.2 비주얼 스튜디오 설치

2.2.4 ‘설치 관리자’ 다운로드가 완료되면 설치 옵션을 설정하기 위한 윈도우가 열림(2)

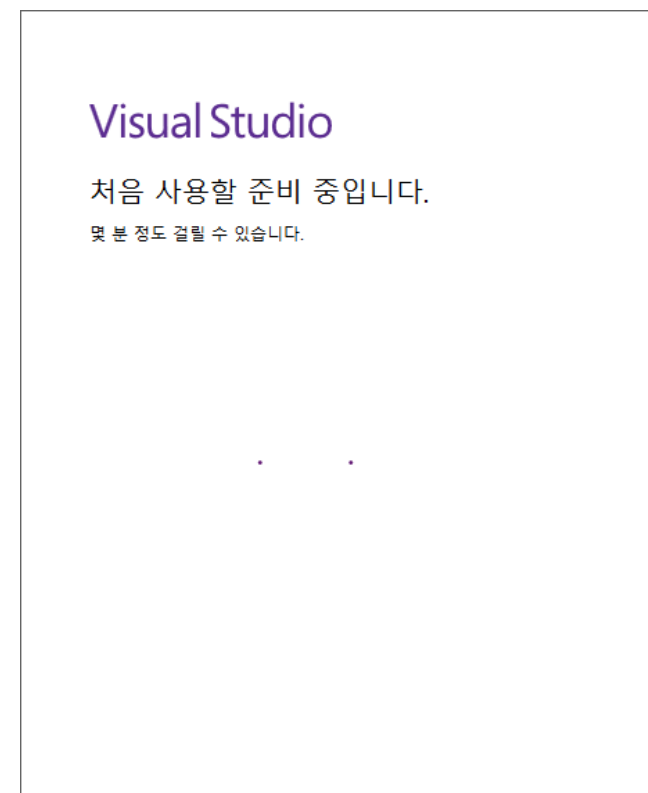
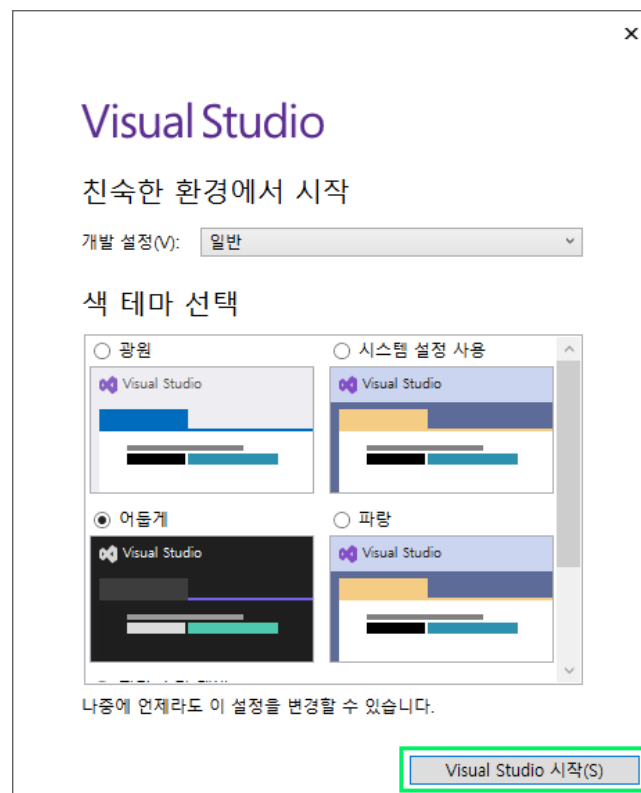
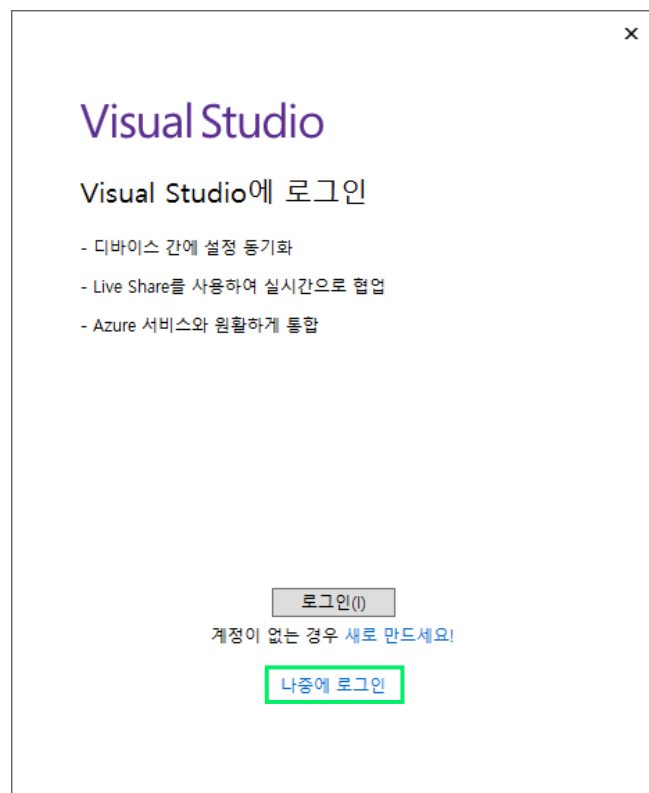
- ‘워크로드’에서 ‘웹 및 클라우드’의 ‘Python 개발’ 옵션을 체크
- 우측 ‘설치 세부 정보’에서 ‘Python 언어 지원’이 기본으로 체크되어 있고, ‘Python 3 xx-bit’ 등 다른 옵션은 언체크
- 이미 아나콘다를 설치했기 때문에 별도의 파이썬 소프트웨어를 설치할 필요가 없음
- 마지막으로 우측 하단의 ‘설치’ 버튼을 클릭 C++ 선택 필수



2.3 비주얼 스튜디오 실행

2.3.1 설치 완료 후 Visual Studio 로그인을 하거나 새로운 계정을 만들 수 있음. ‘나중에 로그인’ 클릭

2.3.2 개발 환경을 설정 후 ‘Visual Studio 시작’ 버튼 클릭

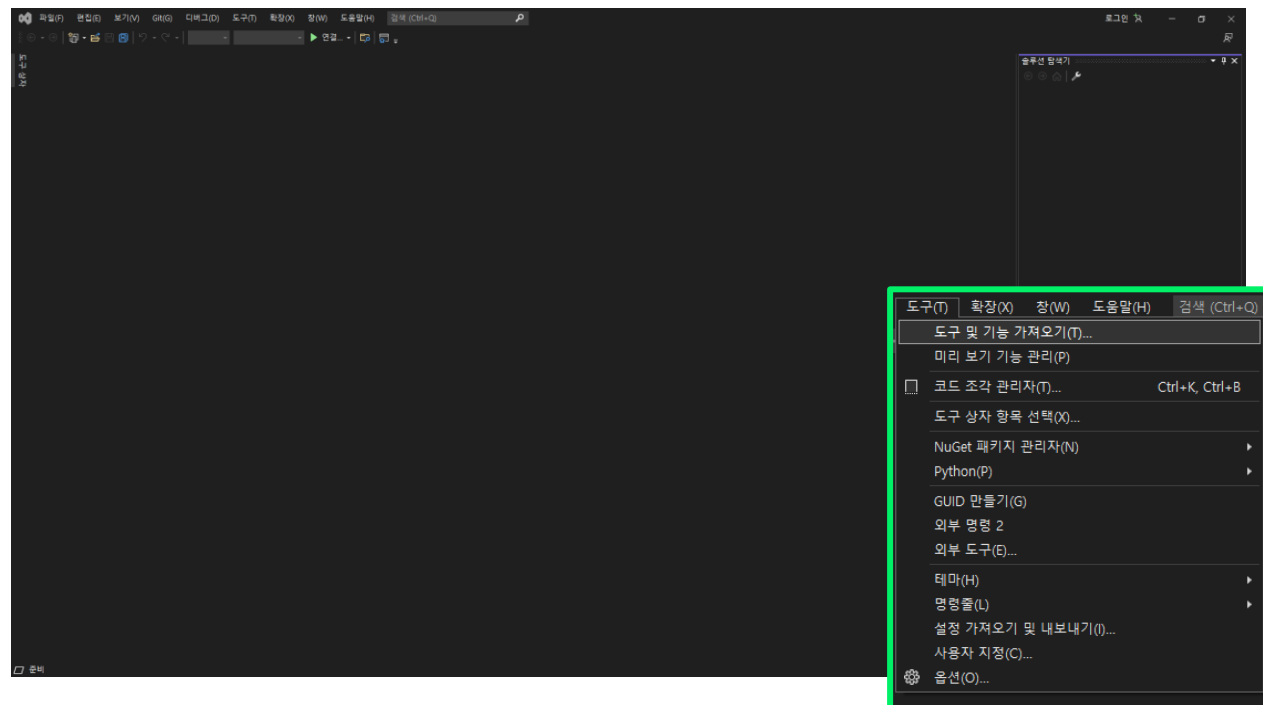
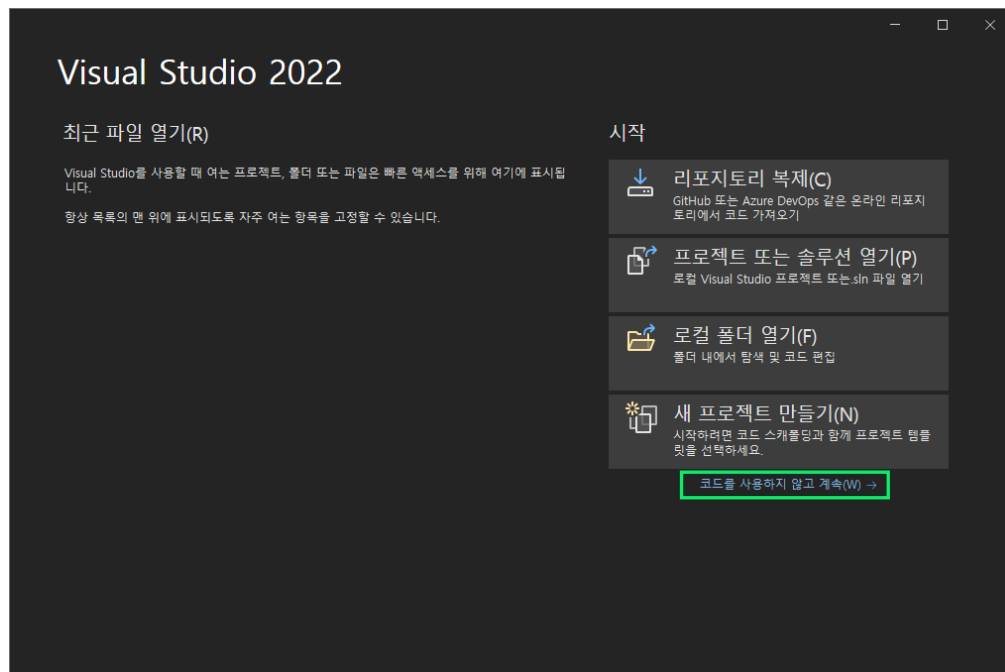


2.3 비주얼 스튜디오 실행

2.3.3 Visual Studio 2022 실행된 것을 확인, ‘코드를 사용하지 않고 계속(W) →’ 옵션을 클릭

2.3.4 Visual Studio 202 프로그램 실행 확인

2.3.5 ‘코드를 사용하지 않고 계속’을 선택한 후에 ‘도구’ 선택 → ‘도구 및 기능 가져오기’ 선택



1. 아나콘다(Anaconda)
2. 비주얼 스튜디오(Visual Studio)
3. **텐서플로우(TensorFlow)**
4. 텐서플로우(TensorFlow)
Object Detection API 설치

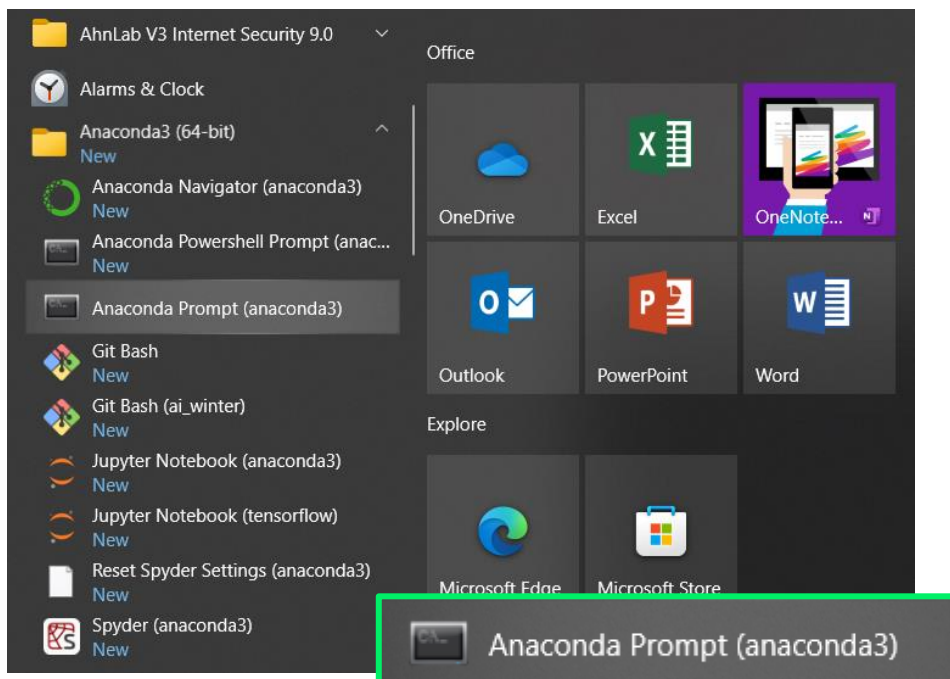
3.1 텐서플로우 설치

3.1.1 Anaconda Prompt 열기

3.1.2 conda env 생성하기

> conda create -n <환경이름> pip python=<python 버전>

우측 예제에서는 환경이름은 2022_ai, python 버전은 3.8 사용



```
Anaconda Prompt (anaconda3) - conda deactivate - deactivate - conda create -n 2022_ai pip python=3.8

(base) C:\Users\JHbaek>conda create -n 2022_ai pip python=3.8
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\JHbaek\anaconda3\envs\2022_ai

added / updated specs:
- pip
- python=3.8

The following packages will be downloaded:

package | build | size
-----|-----|-----
sqlite-3.37.2 | h2bbff1b_0 | 799 KB
-----|-----|-----
Total: 799 KB

The following NEW packages will be INSTALLED:

ca-certificates pkgs/main/win-64::ca-certificates-2021.10.26-haa95532_4
certifi pkgs/main/win-64::certifi-2021.10.8-py38haa95532_2
openssl pkgs/main/win-64::openssl-1.1.1m-h2bbff1b_0
pip pkgs/main/win-64::pip-21.2.2-py38haa95532_0
python pkgs/main/win-64::python-3.8.12-h6244533_0
setuptools pkgs/main/win-64::setuptools-58.0.4-py38haa95532_0
sqlite pkgs/main/win-64::sqlite-3.37.2-h2bbff1b_0
vc pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
wincertstore pkgs/main/win-64::wincertstore-0.2-py38haa95532_2

Proceed ([y]/n)?
```

3.1 텐서플로우 설치

3.1.3 생성한 가상환경 활성화: 항상 가상환경에서 작업해야 함

> conda activate <환경이름>

(base)에서 (환경이름)으로 바뀐 것 확인하기

Anaconda Prompt (anaconda3) - conda deactivate - deactivate - conda deactivate

```
(base) C:\Users\JHBaek>conda activate 2022_ai
(2022_ai) C:\Users\JHBaek>
(2022_ai) C:\Users\JHBaek>
(2022_ai) C:\Users\JHBaek>conda deactivate

(base) C:\Users\JHBaek>
(base) C:\Users\JHBaek>
(base) C:\Users\JHBaek>
(base) C:\Users\JHBaek>
(base) C:\Users\JHBaek>conda activate 2022_ai
(2022_ai) C:\Users\JHBaek>
```


3.1 텐서플로우 설치

3.1.4 Tensorflow 버전 설치하기

> pip install --ignore-installed --upgrade tensorflow

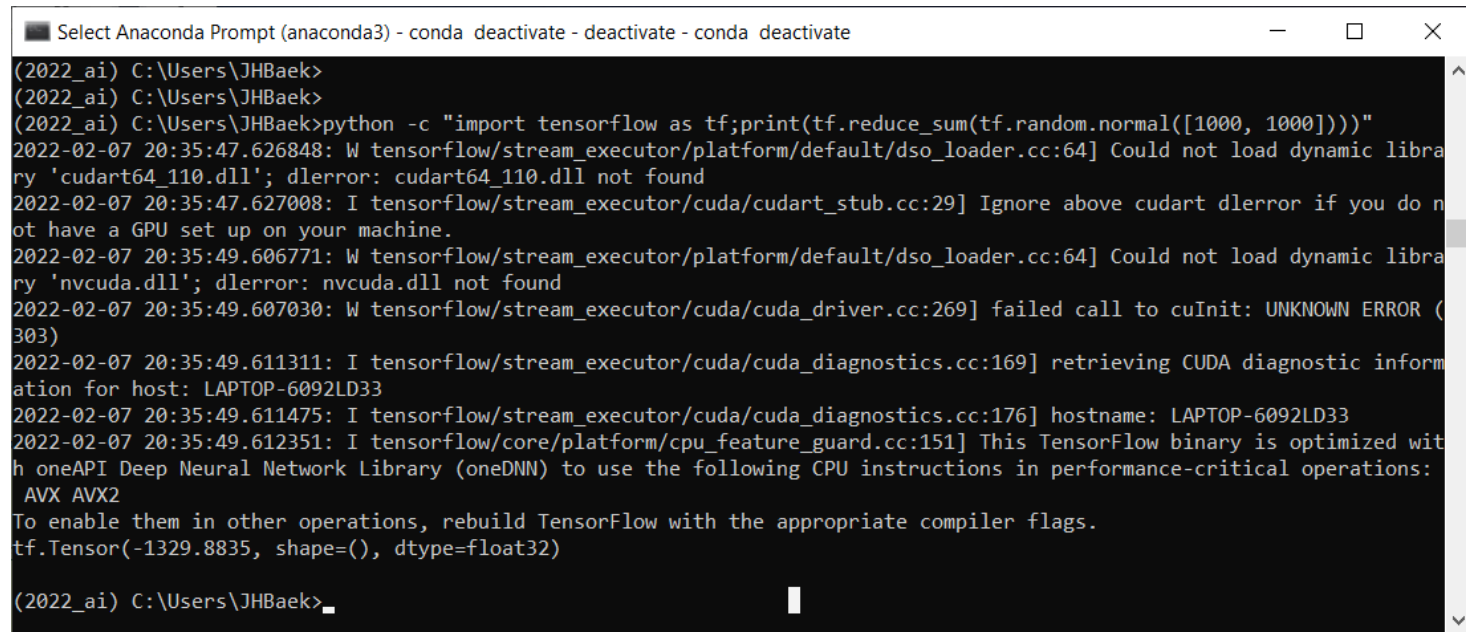
```
Anaconda Prompt (anaconda3) - conda deactivate - deactivate - conda deactivate - pip install --ignore-installed --upgrade tensorflow... -
(base) C:\Users\JHbaek>conda activate 2022_ai
(2022_ai) C:\Users\JHbaek>
(2022_ai) C:\Users\JHbaek>
(2022_ai) C:\Users\JHbaek>conda deactivate
(base) C:\Users\JHbaek>
(base) C:\Users\JHbaek>
(base) C:\Users\JHbaek>
(base) C:\Users\JHbaek>
(base) C:\Users\JHbaek>conda activate 2022_ai
(2022_ai) C:\Users\JHbaek>pip install --ignore-installed --upgrade tensorflow
```

```
Anaconda Prompt (anaconda3)
Downloading certifi-2021.10.8-py2.py3-none-any.whl (149 kB)
|-----| 149 kB 3.3 MB/s
Collecting charset-normalizer~=2.0.0
Downloading charset-normalizer-2.0.11-py3-none-any.whl (39 kB)
Collecting idna<4,>=2.5
Downloading idna-3.3-py3-none-any.whl (61 kB)
|-----| 61 kB 4.1 MB/s
Collecting urllib3<1.27,>=1.21.1
Downloading urllib3-1.26.8-py2.py3-none-any.whl (138 kB)
|-----| 138 kB 6.4 MB/s
Collecting oauthlib>=3.0.0
Downloading oauthlib-3.2.0-py3-none-any.whl (151 kB)
|-----| 151 kB 3.3 MB/s
Building wheels for collected packages: termcolor
Building wheel for termcolor (setup.py) ... done
Created wheel for termcolor: filename=termcolor-1.1.0-py3-none-any.whl size=4848 sha256=4353be350f20369a8693b8294cc33519e7a426956d53a1689c3f4f46bea46496
Stored in directory: c:\users\jhbaek\appdata\local\pip\cache\wheels\af\16\9c\5473df82468f958445479c59e784896fa24f4a5fc024b0f501
Successfully built termcolor
Installing collected packages: urllib3, pyasn1, idna, charset-normalizer, certifi, zipp, six, requests, pyasn1-modules, o
authlib, cachetools, requests-oauthlib, importlib-metadata, google-auth, wheel, werkzeug, tensorboard-plugin-wit, tensorboard-
data-server, setuptools, protobuf, numpy, markdown, grpcio, google-auth-oauthlib, absl-py, wrapt, typing-extensions, tf-estima
tor-nightly, termcolor, tensorflow-io-gcs-filesystem, tensorboard, opt-einsum, libclang, keras-preprocessing, keras, h5py, goo
gle-pasta, gast, flatbuffers, astunparse, tensorflow
Successfully installed absl-py-1.0.0 astunparse-1.6.3 cachetools-5.0.0 certifi-2021.10.8 charset-normalizer-2.0.11 flatbuffers
-2.0 gast-0.5.3 google-auth-2.6.0 google-auth-oauthlib-0.4.6 google-pasta-0.2.0 grpcio-1.43.0 h5py-3.6.0 idna-3.3 importlib-me
tadata-4.10.1 keras-2.8.0 keras-preprocessing-1.1.2 libclang-13.0.0 markdown-3.3.6 numpy-1.22.2 oauthlib-3.2.0 opt-einsum-3.3.
0 protobuf-3.19.4 pyasn1-0.4.8 pyasn1-modules-0.2.8 requests-2.27.1 requests-oauthlib-1.3.1 rsa-4.8 setuptools-60.7.1 six-1.16
.0 tensorboard-2.8.0 tensorboard-data-server-0.6.1 tensorboard-plugin-wit-1.8.1 tensorflow-2.8.0 tensorflow-io-gcs-filesystem-
0.24.0 termcolor-1.1.0 tf-estimator-nightly-2.8.0.dev2021122109 typing-extensions-4.0.1 urllib3-1.26.8 werkzeug-2.0.2 wheel-0.
37.1 wrapt-1.13.3 zipp-3.7.0
```

3.1 텐서플로우 설치

3.1.5 Tensorflow가 제대로 설치되었는지 확인하기

> python -c "import tensorflow as tf; print(tf.reduce_sum(tf.random.normal([1000, 1000])))"



```
Select Anaconda Prompt (anaconda3) - conda deactivate - deactivate - conda deactivate
(2022_ai) C:\Users\JHBaek>
(2022_ai) C:\Users\JHBaek>
(2022_ai) C:\Users\JHBaek>python -c "import tensorflow as tf;print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
2022-02-07 20:35:47.626848: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlderror: cudart64_110.dll not found
2022-02-07 20:35:47.627008: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-02-07 20:35:49.606771: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlderror: nvcuda.dll not found
2022-02-07 20:35:49.607030: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-02-07 20:35:49.611311: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: LAPTOP-6092LD33
2022-02-07 20:35:49.611475: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-6092LD33
2022-02-07 20:35:49.612351: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:
 AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
tf.Tensor(-1329.8835, shape=(), dtype=float32)

(2022_ai) C:\Users\JHBaek>_
```

1. 아나콘다(Anaconda)
2. 비주얼 스튜디오(Visual Studio)
3. 텐서플로우(TensorFlow)
4. **텐서플로우(TensorFlow)**
Object Detection API 설치

4.1 텐서플로우(TensorFlow) 디렉터리

4.1.1 원하는 위치에 tensorflow 디렉터리 생성하기

(예: C:\Users\JHBaek\tensorflow)

4.1.2 위에서 만든 tensorflow 디렉터리 하위에 tensorflow Models 다운로드

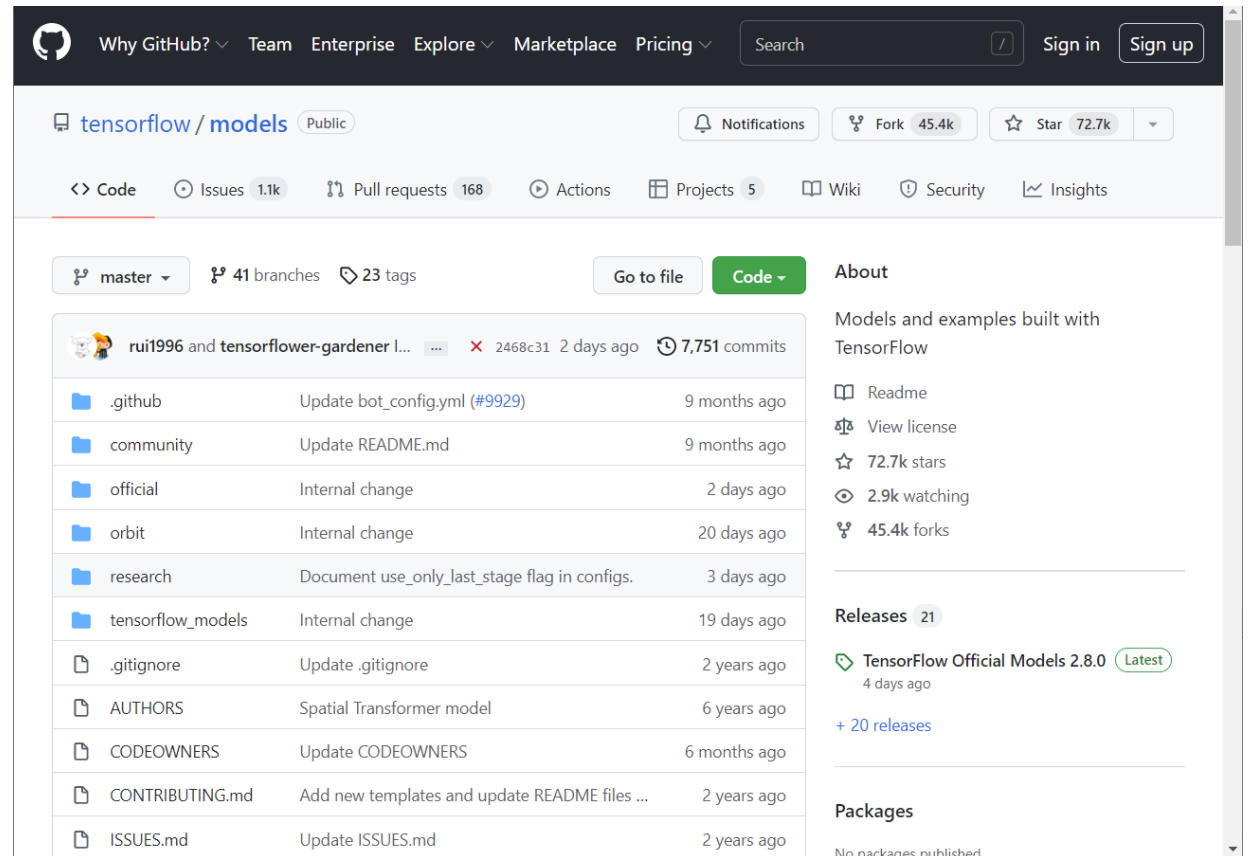
(git을 사용하거나 zip 파일을 다운로드 받고 unzip 후 디렉터리명을 models-master에서 models로 바꾸기)

<https://github.com/tensorflow/models>

tensorflow 하위에 workspace라는 디렉터를 만들어 자신이 만드는 코드나 데이터, 모델 등 저장

C:\Users\JHBaek\tensorflow\models

C:\Users\JHBaek\tensorflow\workspace



4.2 Protobuf 설치하기

4.2.1 자신의 시스템에 맞는 버전 확인후 다운로드 받기 (<https://github.com/protocolbuffers/protobuf/releases/tag/v3.19.4>)

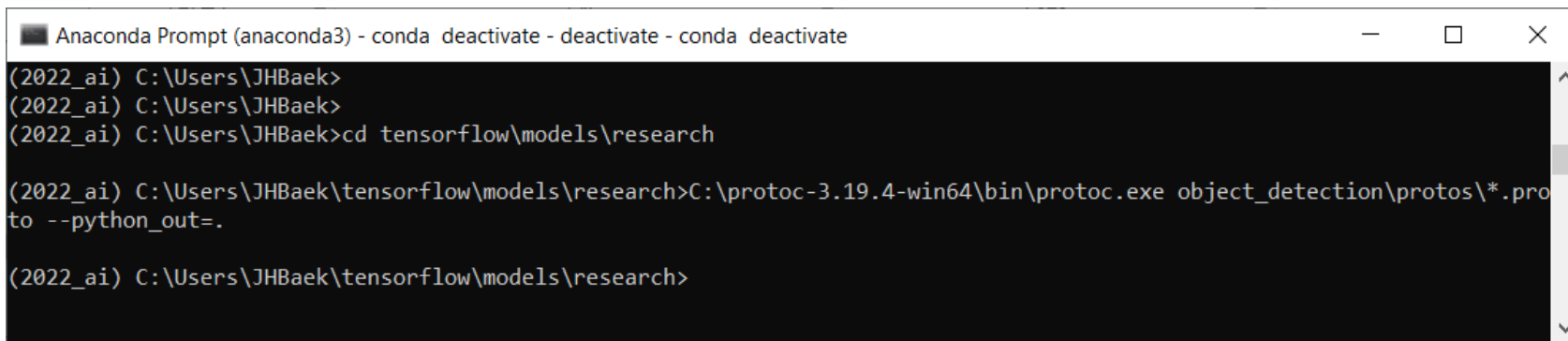
4.2.2 unzip 후 원하는 위치(예: C:\Program Files\)\로 소스코드 옮기기

4.2.3 현재 디렉터리 위치 변경 후 protoc 실행하기

```
> cd tensorflow\models\research
```

```
> C:\protoc-3.19.4-win64\bin\protoc.exe object_detection\protos\*.proto --python_out=.
```

(예: C:\Program Files\protoc-3.19.4-win64\bin\protoc.exe object_detection\protos*.proto --python_out=.)



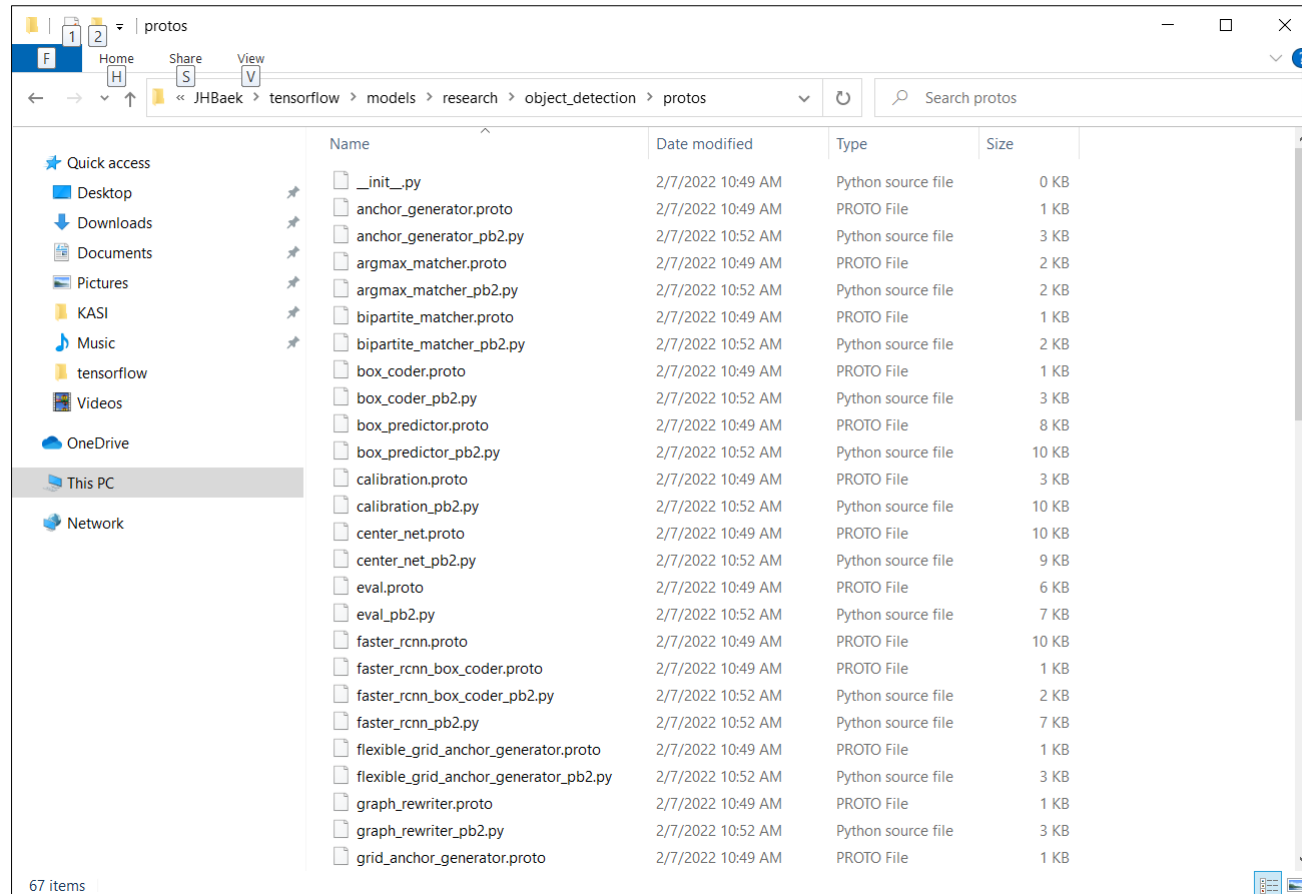
```
Anaconda Prompt (anaconda3) - conda deactivate - deactivate - conda deactivate
(2022_ai) C:\Users\JHBaek>
(2022_ai) C:\Users\JHBaek>
(2022_ai) C:\Users\JHBaek>cd tensorflow\models\research

(2022_ai) C:\Users\JHBaek\tensorflow\models\research>C:\protoc-3.19.4-win64\bin\protoc.exe object_detection\protos\*.proto --python_out=.

(2022_ai) C:\Users\JHBaek\tensorflow\models\research>
```

4.2 Protobuf 설치하기

..\tensorflow\models\research\object_detection\protos\에 *_pb2.py 파일들이 생성된 것을 확인



4.3 COCO API (pycocotools) 설치

> conda install cython

```
(2022_ai) C:\Users\JHBaek\tensorflow>pip install cython
Collecting cython
  Using cached Cython-0.29.27-py2.py3-none-any.whl (983 kB)
Installing collected packages: cython
Successfully installed cython-0.29.27

(2022_ai) C:\Users\JHBaek\tensorflow>
```

> pip install git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI

```
(2022_ai) C:\Users\JHBaek\tensorflow>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\jhbaek\appdata\local\temp\pip-req-build-uq6u8s2v
  Running command git clone -q https://github.com/philferriere/cocoapi.git 'C:\Users\JHBaek\AppData\Local\Temp\pip-req-build-uq6u8s2v'
  Resolved https://github.com/philferriere/cocoapi.git to commit 2929bd2ef6b451054755dfd7ceb09278f935f7ad
Building wheels for collected packages: pycocotools
  Building wheel for pycocotools (setup.py) ... done
  Created wheel for pycocotools: filename=pycocotools-2.0-cp38-cp38-win_amd64.whl size=74983 sha256=3cc1c4d85ca873f2dd480d314d47800c610366fb56ca798dc47a1cf6d819e6d3
  Stored in directory: C:\Users\JHBaek\AppData\Local\Temp\pip-ephem-wheel-cache-6rk12f7y\wheels\bd\1c\0d\8c82e1b9bc855b82e1eb53eadea4459efe171d2daf5a222701
Successfully built pycocotools
Installing collected packages: pycocotools
Successfully installed pycocotools-2.0

(2022_ai) C:\Users\JHBaek\tensorflow>
```

4.3 COCO API (pycocotools) 설치

cf) git이 존재하지 않는 경우

git를 확인해보고 없을 경우에는 conda install git Cython을 통해 설치할 수 있음.

```
(ai_2022) C:\Users\kasi\tensorflow>conda install git Cython
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

```
==> WARNING: A newer version of conda exists. <==
  current version: 4.10.3
  latest version: 4.11.0
```

```
Please update conda by running
```

```
$ conda update -n base -c defaults conda
```

```
## Package Plan ##
```

```
environment location: C:\Users\kasi\conda\envs\ai_2022
```

```
added / updated specs:
```

```
- cython
- git
```

```
The following packages will be downloaded:
```


4.4 Object Detection API 설치하기

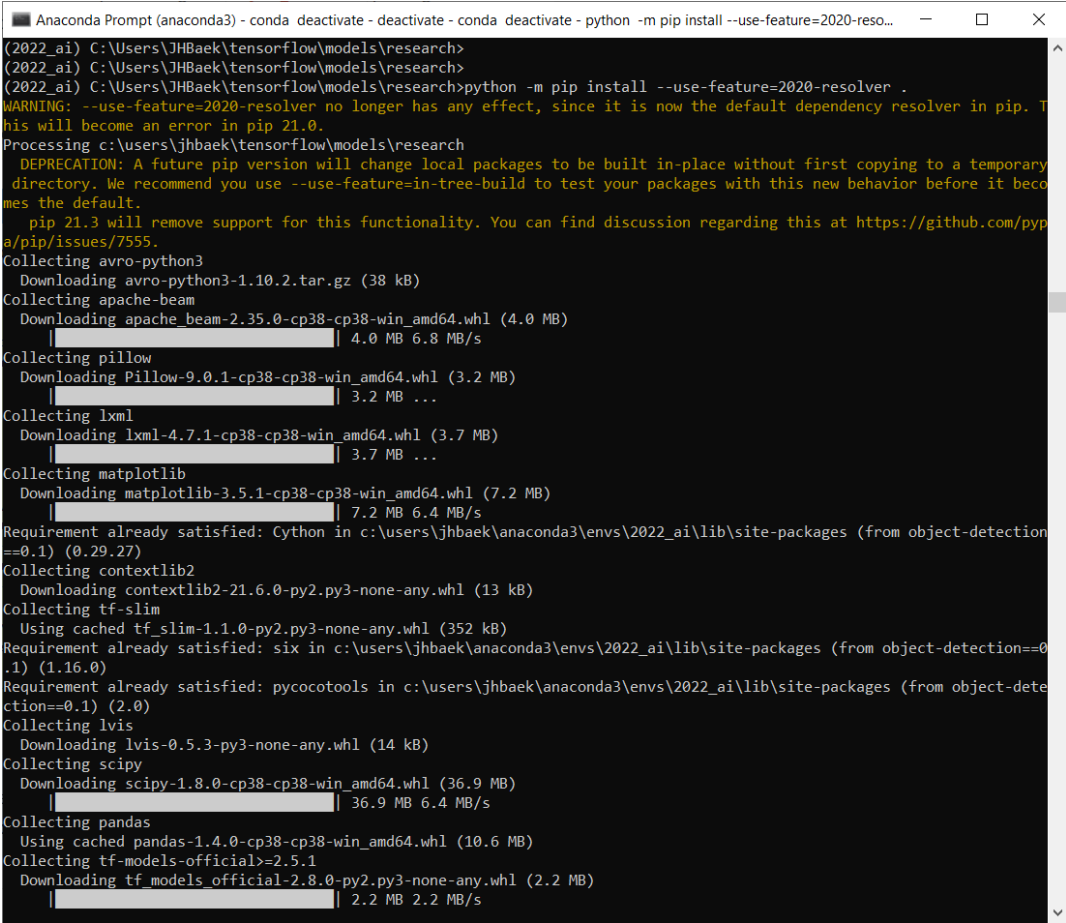
4.4.1 탐색기에서

tensorflow/models/research/object_detection/packages/tf2/
에 있는 setup.py를 tensorflow/models/research로
복사/이동하기

4.4.2 Anaconda prompt에서

tensorflow/models/research로 이동하기

> python -m pip install --use-feature=2020-resolver .

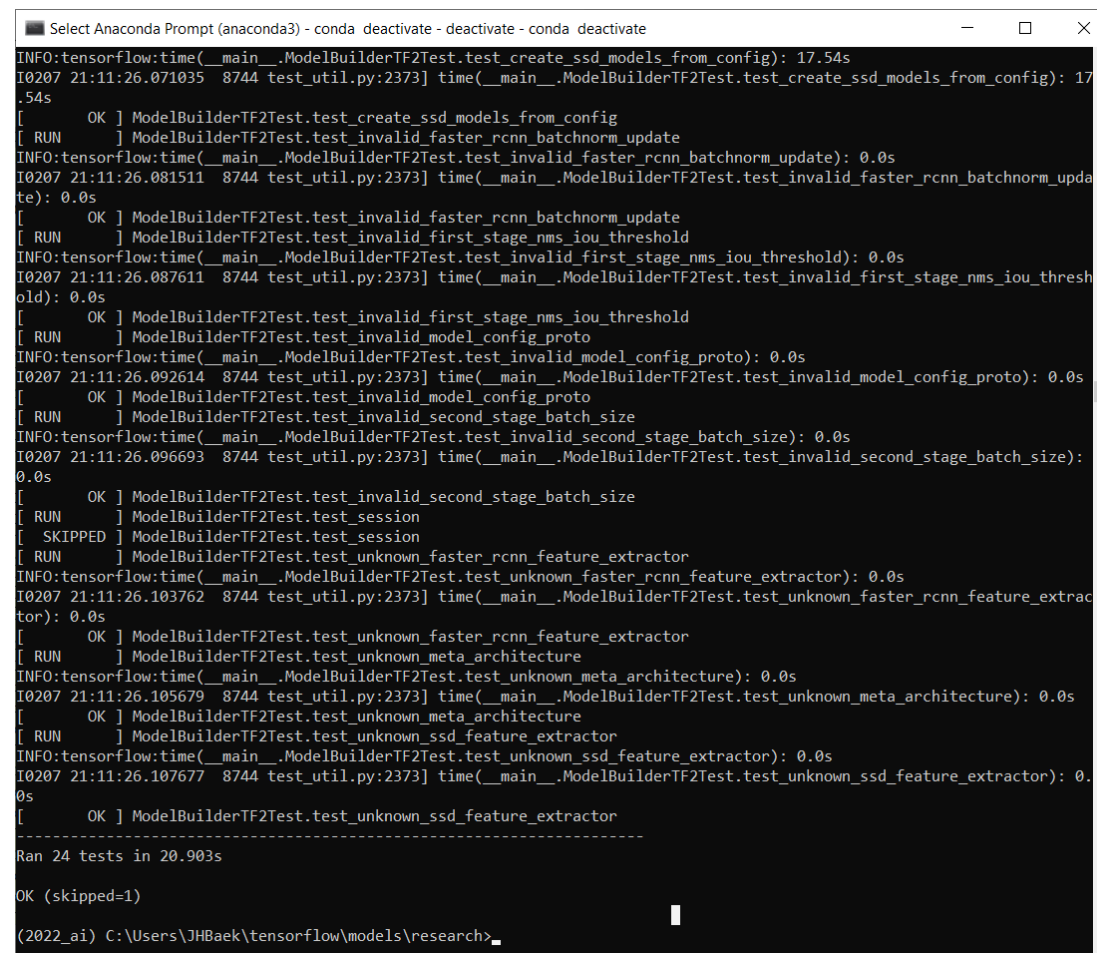


```
Anaconda Prompt (anaconda3) - conda deactivate - deactivate - conda deactivate - python -m pip install --use-feature=2020-reso...
(2022_ai) C:\Users\JHbaek\tensorflow\models\research>
(2022_ai) C:\Users\JHbaek\tensorflow\models\research>
(2022_ai) C:\Users\JHbaek\tensorflow\models\research>python -m pip install --use-feature=2020-resolver .
WARNING: --use-feature=2020-resolver no longer has any effect, since it is now the default dependency resolver in pip. This
this will become an error in pip 21.0.
Processing c:\users\jhbaek\tensorflow\models\research
  DEPRECATION: A future pip version will change local packages to be built in-place without first copying to a temporary
  directory. We recommend you use --use-feature=in-tree-build to test your packages with this new behavior before it beco
  mes the default.
  pip 21.3 will remove support for this functionality. You can find discussion regarding this at https://github.com/pypp
  a/pip/issues/7555.
Collecting avro-python3
  Downloading avro-python3-1.10.2.tar.gz (38 kB)
Collecting apache-beam
  Downloading apache_beam-2.35.0-cp38-cp38-win_amd64.whl (4.0 MB)
  | 4.0 MB 6.8 MB/s
Collecting pillow
  Downloading Pillow-9.0.1-cp38-cp38-win_amd64.whl (3.2 MB)
  | 3.2 MB ...
Collecting lxml
  Downloading lxml-4.7.1-cp38-cp38-win_amd64.whl (3.7 MB)
  | 3.7 MB ...
Collecting matplotlib
  Downloading matplotlib-3.5.1-cp38-cp38-win_amd64.whl (7.2 MB)
  | 7.2 MB 6.4 MB/s
Requirement already satisfied: Cython in c:\users\jhbaek\anaconda3\envs\2022_ai\lib\site-packages (from object-detection==0
==0.1) (0.29.27)
Collecting contextlib2
  Downloading contextlib2-21.6.0-py2.py3-none-any.whl (13 kB)
Collecting tf-slim
  Using cached tf-slim-1.1.0-py2.py3-none-any.whl (352 kB)
Requirement already satisfied: six in c:\users\jhbaek\anaconda3\envs\2022_ai\lib\site-packages (from object-detection==0
.1) (1.16.0)
Requirement already satisfied: pycocotools in c:\users\jhbaek\anaconda3\envs\2022_ai\lib\site-packages (from object-dete
ction==0.1) (2.0)
Collecting lvis
  Downloading lvis-0.5.3-py3-none-any.whl (14 kB)
Collecting scipy
  Downloading scipy-1.8.0-cp38-cp38-win_amd64.whl (36.9 MB)
  | 36.9 MB 6.4 MB/s
Collecting pandas
  Using cached pandas-1.4.0-cp38-cp38-win_amd64.whl (10.6 MB)
Collecting tf-models-official>=2.5.1
  Downloading tf_models_official-2.8.0-py2.py3-none-any.whl (2.2 MB)
  | 2.2 MB 2.2 MB/s
```

4.4 Object Detection API 설치하기

4.4.3 설치 확인하기

> python object_detection/builders/model_builder_tf2_test.py



```
Select Anaconda Prompt (anaconda3) - conda deactivate - deactivate - conda deactivate
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_ssd_models_from_config): 17.54s
I0207 21:11:26.071035 8744 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_create_ssd_models_from_config): 17.54s
[ OK ] ModelBuilderTF2Test.test_create_ssd_models_from_config
[ RUN ] ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update): 0.0s
I0207 21:11:26.081511 8744 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update): 0.0s
[ OK ] ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update
[ RUN ] ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold): 0.0s
I0207 21:11:26.087611 8744 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold): 0.0s
[ OK ] ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold
[ RUN ] ModelBuilderTF2Test.test_invalid_model_config_proto
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_model_config_proto): 0.0s
I0207 21:11:26.092614 8744 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_invalid_model_config_proto): 0.0s
[ OK ] ModelBuilderTF2Test.test_invalid_model_config_proto
[ RUN ] ModelBuilderTF2Test.test_invalid_second_stage_batch_size
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_second_stage_batch_size): 0.0s
I0207 21:11:26.096693 8744 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_invalid_second_stage_batch_size): 0.0s
[ OK ] ModelBuilderTF2Test.test_invalid_second_stage_batch_size
[ RUN ] ModelBuilderTF2Test.test_session
[ SKIPPED ] ModelBuilderTF2Test.test_session
[ RUN ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor): 0.0s
I0207 21:11:26.103762 8744 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor): 0.0s
[ OK ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
[ RUN ] ModelBuilderTF2Test.test_unknown_meta_architecture
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_meta_architecture): 0.0s
I0207 21:11:26.105679 8744 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_unknown_meta_architecture): 0.0s
[ OK ] ModelBuilderTF2Test.test_unknown_meta_architecture
[ RUN ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_ssd_feature_extractor): 0.0s
I0207 21:11:26.107677 8744 test_util.py:2373] time(__main__.ModelBuilderTF2Test.test_unknown_ssd_feature_extractor): 0.0s
[ OK ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
-----
Ran 24 tests in 20.903s

OK (skipped=1)

(2022_ai) C:\Users\JHbaek\tensorflow\models\research>
```