13기 정규세션

ToBig's 13기 강의자
정민준

환경구축

Contents

Unit 01 | 가상환경

Unit 02 | 환경구축

Unit 03 | 과제 및 실습

Unit 04 | Q & A

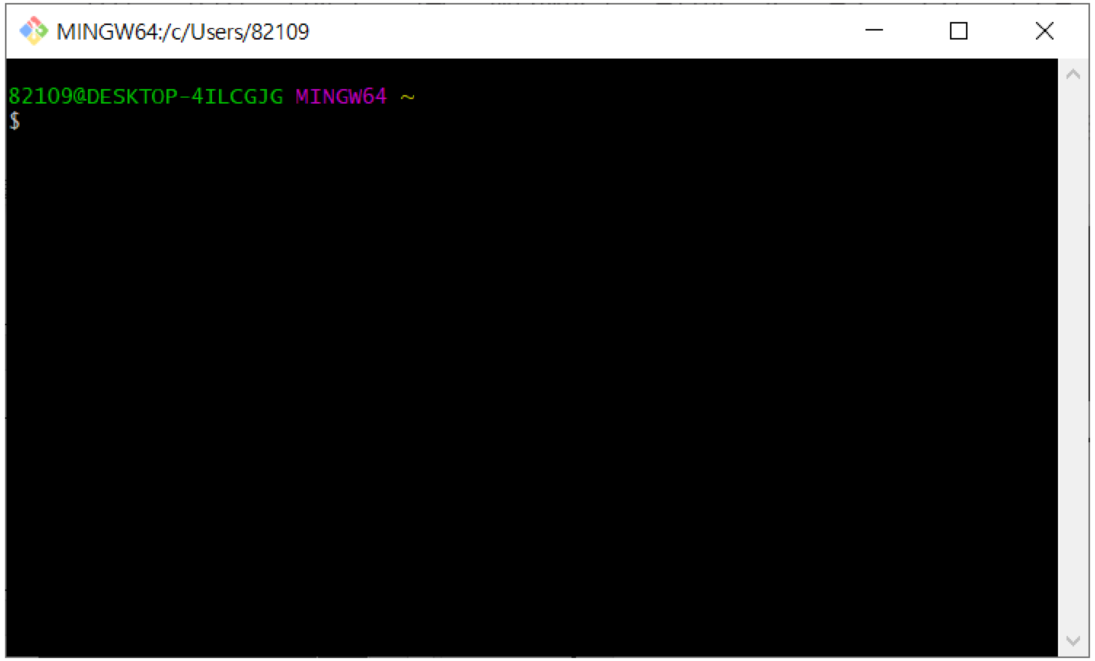
Unit 01 | 가상환경

Unit 01 | 가상환경

가상환경?

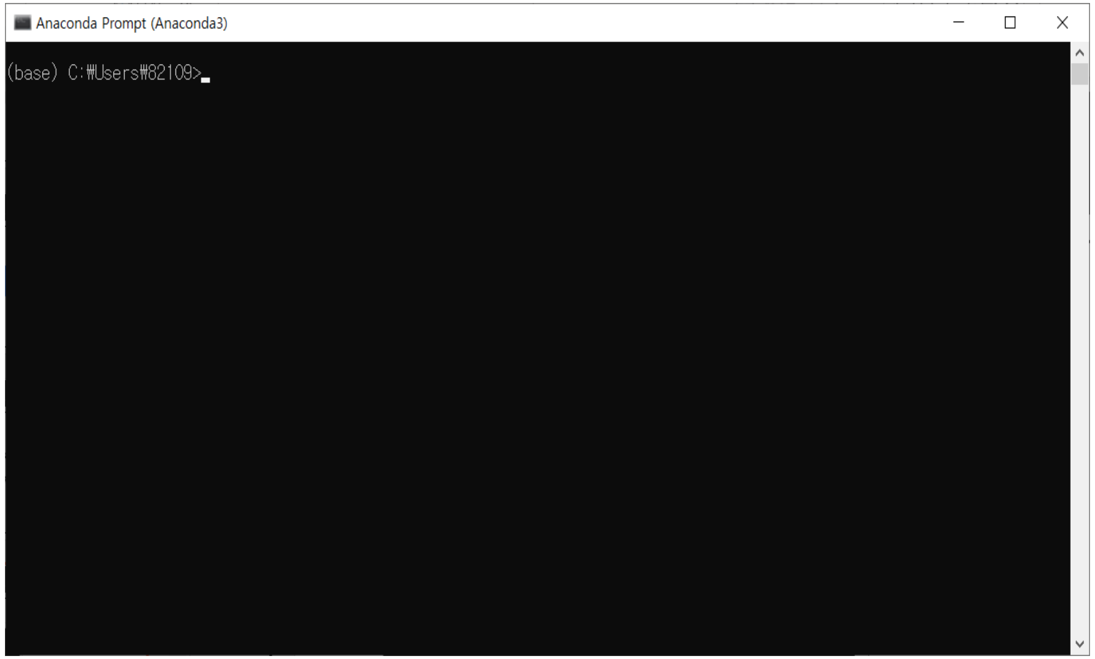
Python 환경구축을 위해 필요한 모듈을 저장!

Unit 01 | 가상환경



MINGW64:/c/Users/82109

```
82109@DESKTOP-4ILCGJG MINGW64 ~
$
```



Anaconda Prompt (Anaconda3)

```
(base) C:₩Users₩82109>_
```

Git bash

Anaconda

Unit 01 | 가상환경

먼저!

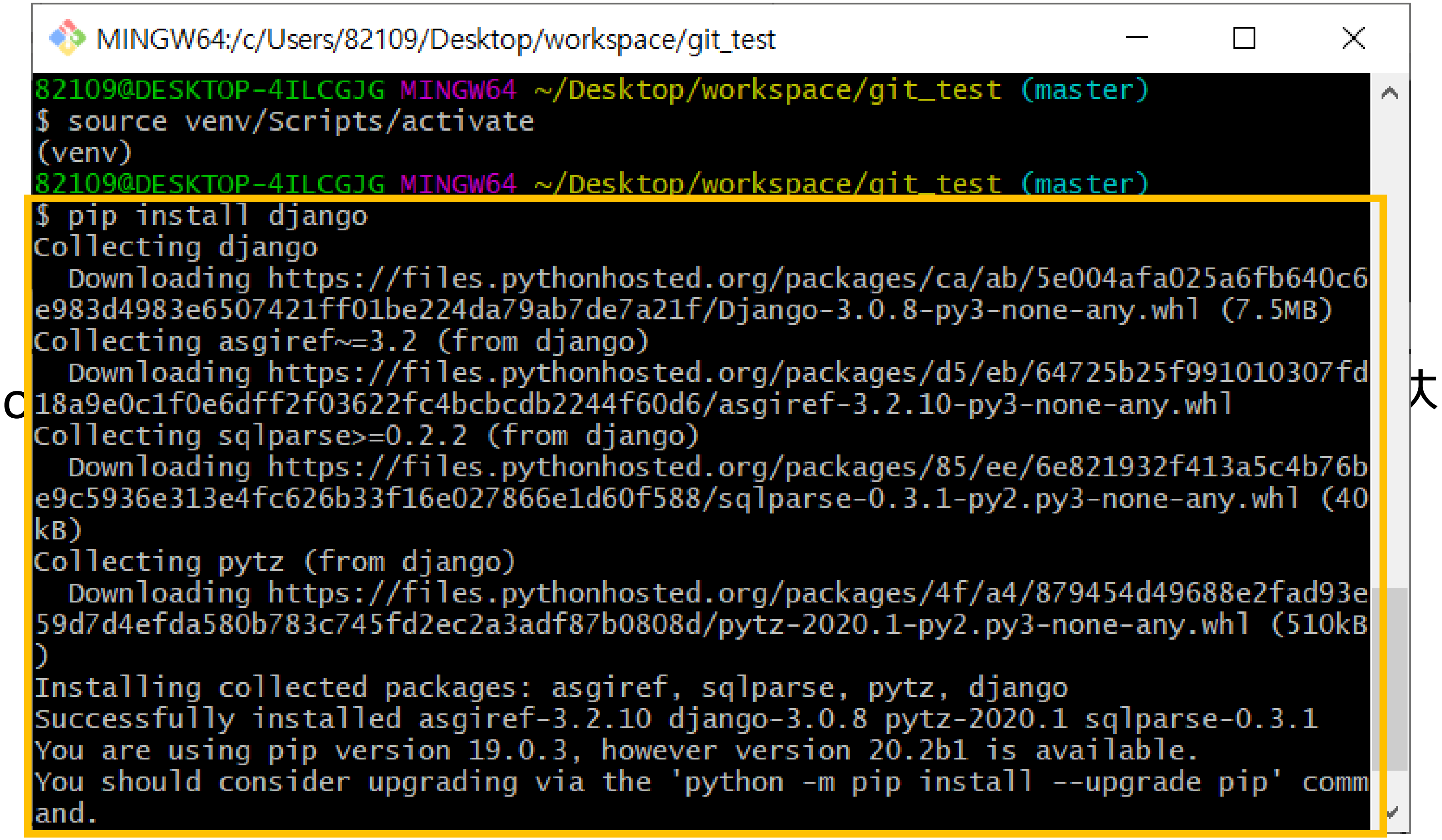
pip은 PyPI(Python Package Index) 저장소로부터 파이썬 패키지를 받아 설치하는 도구!

```
> pip install <가상환경>
```

```
Ex) pip install django
```

Unit 01 | 가상환경

먼저!



MINGW64:/c/Users/82109/Desktop/workspace/git_test

```
82109@DESKTOP-4ILCGJG MINGW64 ~/Desktop/workspace/git_test (master)
$ source venv/Scripts/activate
(venv)
82109@DESKTOP-4ILCGJG MINGW64 ~/Desktop/workspace/git_test (master)
$ pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/ca/ab/5e004afa025a6fb640c6
e983d4983e6507421ff01be224da79ab7de7a21f/Django-3.0.8-py3-none-any.whl (7.5MB)
Collecting asgiref~=3.2 (from django)
  Downloading https://files.pythonhosted.org/packages/d5/eb/64725b25f991010307fd
18a9e0c1f0e6dff2f03622fc4bcbcdb2244f60d6/asgiref-3.2.10-py3-none-any.whl
Collecting sqlparse>=0.2.2 (from django)
  Downloading https://files.pythonhosted.org/packages/85/ee/6e821932f413a5c4b76b
e9c5936e313e4fc626b33f16e027866e1d60f588/sqlparse-0.3.1-py2.py3-none-any.whl (40
kB)
Collecting pytz (from django)
  Downloading https://files.pythonhosted.org/packages/4f/a4/879454d49688e2fad93e
59d7d4efda580b783c745fd2ec2a3adf87b0808d/pytz-2020.1-py2.py3-none-any.whl (510kB
)
Installing collected packages: asgiref, sqlparse, pytz, django
Successfully installed asgiref-3.2.10 django-3.0.8 pytz-2020.1 sqlparse-0.3.1
You are using pip version 19.0.3, however version 20.2b1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.
```

pip은 PyPI(Pytho 치하는 도구!

먼저!

pip은 PyPI(Python Package Index) 저장소로부터 파이썬 패키지를 받아 설치하는 도구!

충돌이 생기지 않을까?

```
> pip install <가상환경>
```

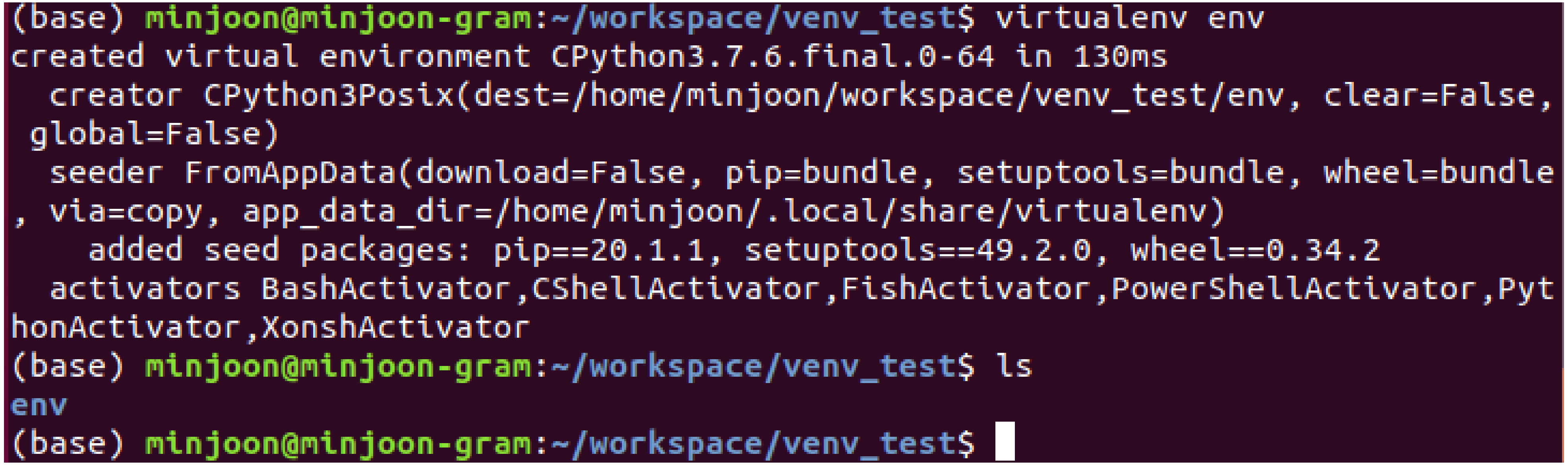
```
Ex) pip install django
```

1. virtualenv를 사용해보자!

Virtualenv를 사용하면 독립적인 가상의 환경을 만들 수 있다

https://virtualenv.pypa.io/en/latest/

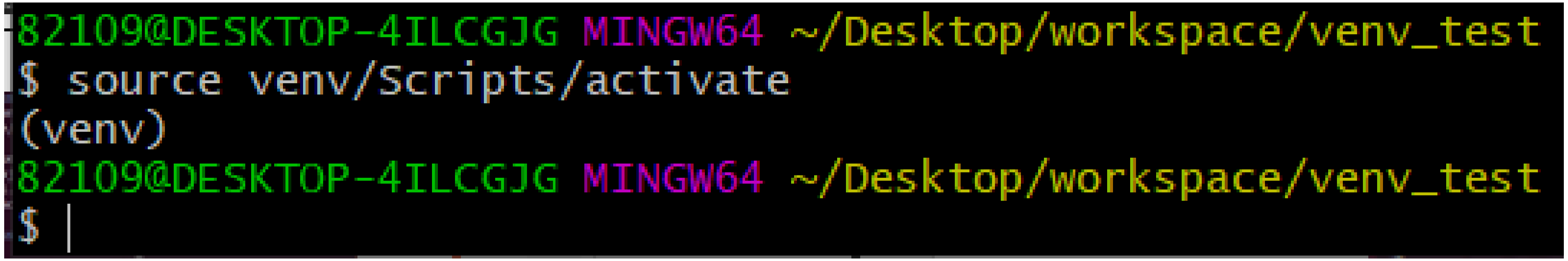
Unit 01 | 가상환경



```
(base) minjoon@minjoon-gram:~/workspace/venv_test$ virtualenv env
created virtual environment CPython3.7.6.final.0-64 in 130ms
  creator CPython3Posix(dest=/home/minjoon/workspace/venv_test/env, clear=False,
 global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle
, via=copy, app_data_dir=/home/minjoon/.local/share/virtualenv)
    added seed packages: pip==20.1.1, setuptools==49.2.0, wheel==0.34.2
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,Pyt
honActivator,XonshActivator
(base) minjoon@minjoon-gram:~/workspace/venv_test$ ls
env
(base) minjoon@minjoon-gram:~/workspace/venv_test$ 
```

virtualenv env

Unit 01 | 가상환경



```
82109@DESKTOP-4ILCGJG MINGW64 ~/Desktop/workspace/venv_test
$ source venv/Scripts/activate
(venv)
82109@DESKTOP-4ILCGJG MINGW64 ~/Desktop/workspace/venv_test
$ |
```

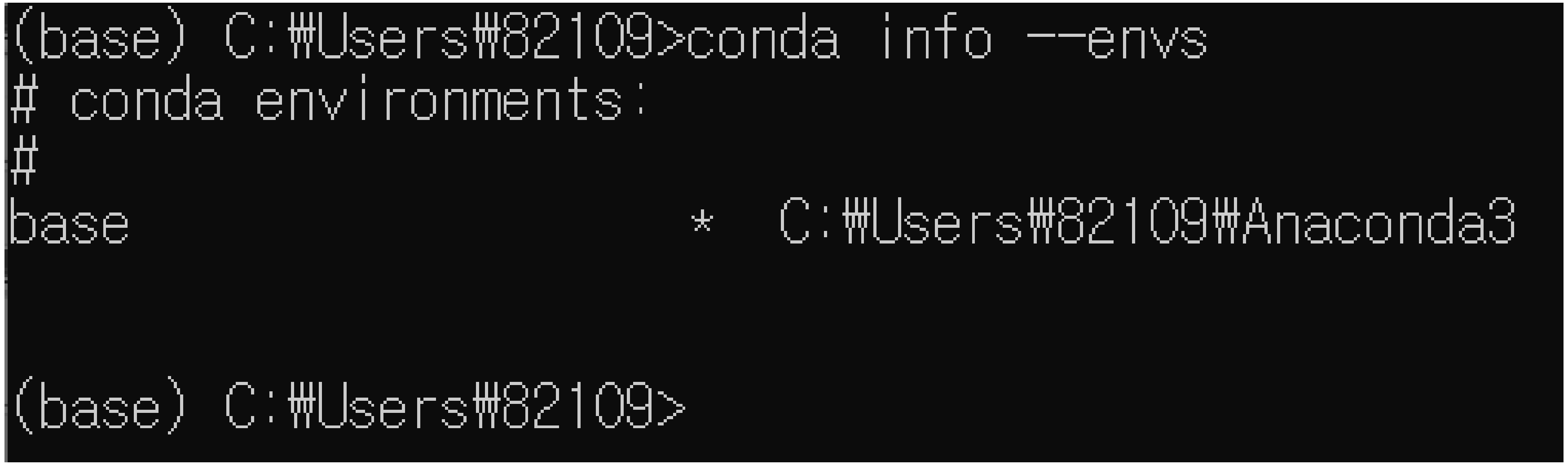
source dir~/activate

2. Anaconda를 사용해보자!

여러가지 수학 및 과학 패키지들을 기본적으로 포함하고 있는 파이썬 배포판

https://www.anaconda.com/

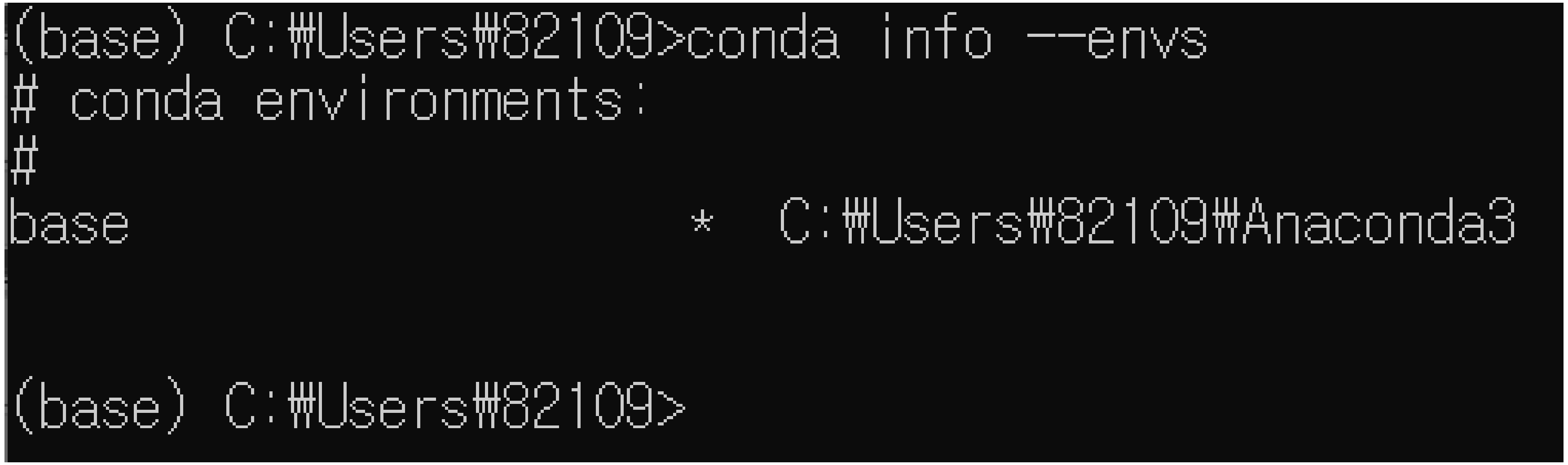
Unit 01 | 가상환경



```
(base) C:₩Users₩82109>conda info --envs
# conda environments:
#
base                  *  C:₩Users₩82109₩Anaconda3


(base) C:₩Users₩82109>
```

conda info --envs

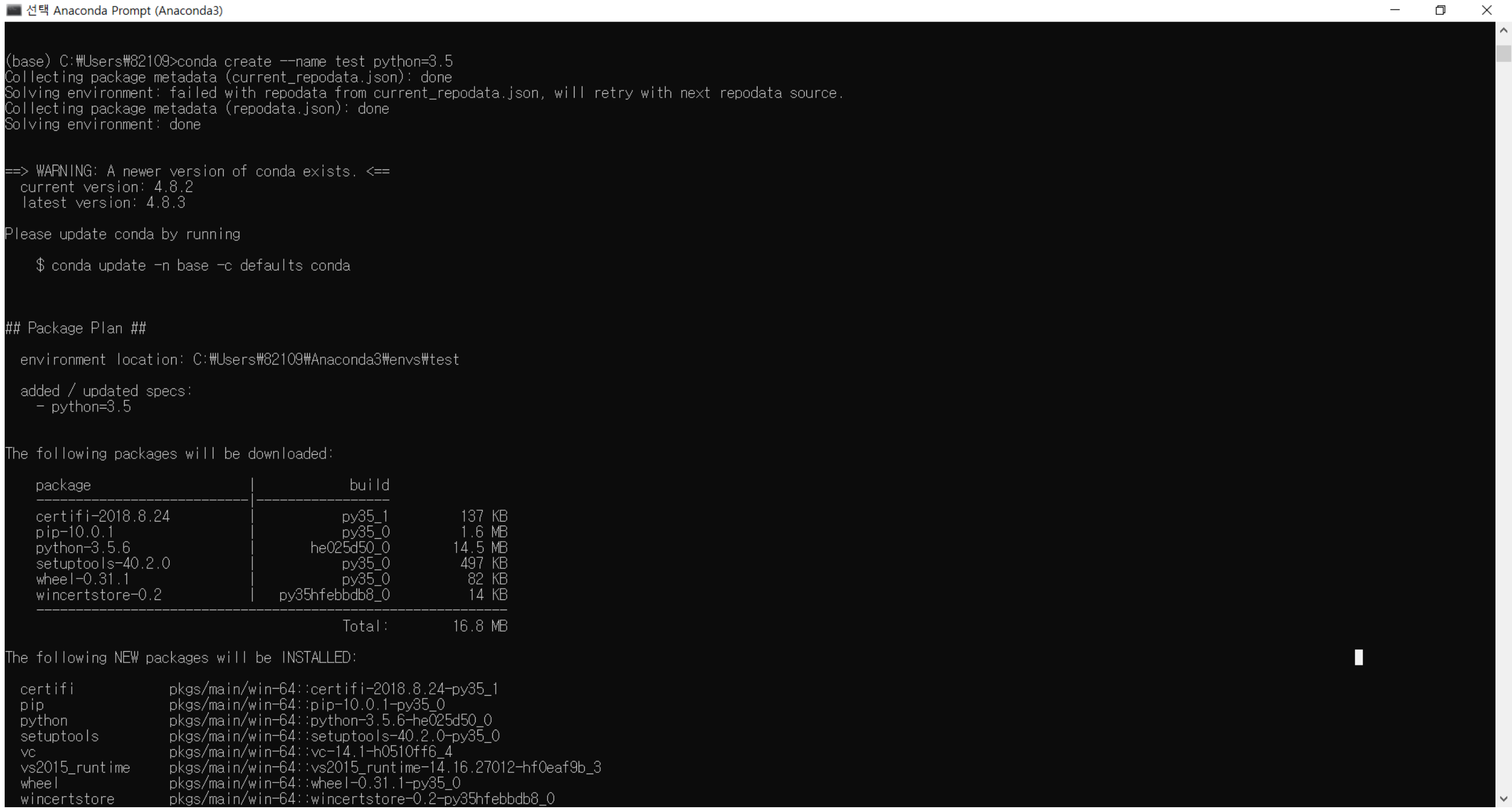
Unit 01 | 가상환경



```
(base) C:₩Users₩82109>conda info --envs
# conda environments:
#
base                  *  C:₩Users₩82109₩Anaconda3


(base) C:₩Users₩82109>
```

<u>conda create -- name <가상환경명> python=3.x</u>

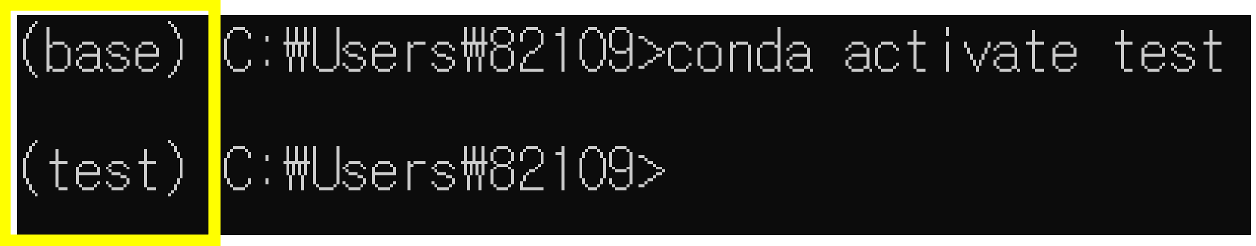
Unit 01 | 가상환경



```
(base) C:₩Users₩82109>conda create --name test python=3.5
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.8.2
  latest version: 4.8.3

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: C:₩Users₩82109₩Anaconda3₩envs₩test

  added / updated specs:
    - python=3.5


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    certifi-2018.8.24          |           py35_1         137 KB
    pip-10.0.1                 |           py35_0         1.6 MB
    python-3.5.6               |       he025d50_0        14.5 MB
    setuptools-40.2.0          |           py35_0         497 KB
    wheel-0.31.1               |           py35_0          82 KB
    wincertstore-0.2           |   py35hfebbdb8_0          14 KB
    ------------------------------------------------------------
                                           Total:        16.8 MB

The following NEW packages will be INSTALLED:

  certifi            pkgs/main/win-64::certifi-2018.8.24-py35_1
  pip                pkgs/main/win-64::pip-10.0.1-py35_0
  python             pkgs/main/win-64::python-3.5.6-he025d50_0
  setuptools         pkgs/main/win-64::setuptools-40.2.0-py35_0
  vc                 pkgs/main/win-64::vc-14.1-h0510ff6_4
  vs2015_runtime     pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_3
  wheel              pkgs/main/win-64::wheel-0.31.1-py35_0
  wincertstore       pkgs/main/win-64::wincertstore-0.2-py35hfebbdb8_0
```

conda create -- name <가상환경명> python=3.x

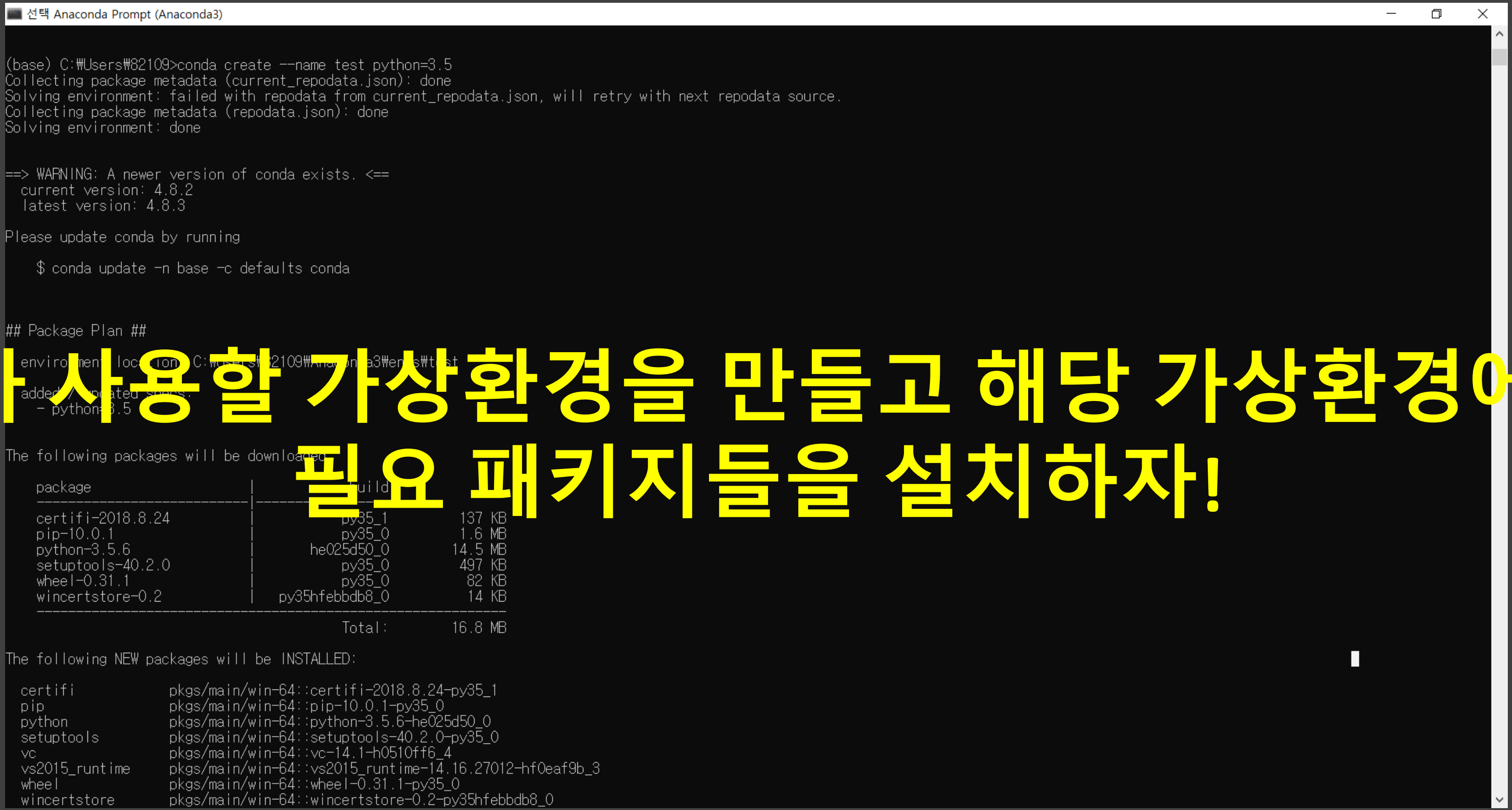
Unit 01 | 가상환경



```
(base) C:₩Users₩82109>conda activate test

(test) C:₩Users₩82109>
```

conda activate <가상환경 이름>

Unit 01 | 가상환경



```
(base) C:₩Users₩82109>conda create --name test python=3.5
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.8.2
  latest version: 4.8.3

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: C:₩Users₩82109₩Anaconda3₩envs₩test

  added / updated specs:
    - python=3.5


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    certifi-2018.8.24          |           py35_1         137 KB
    pip-10.0.1                 |           py35_0         1.6 MB
    python-3.5.6               |       he025d50_0        14.5 MB
    setuptools-40.2.0          |           py35_0         497 KB
    wheel-0.31.1               |           py35_0          82 KB
    wincertstore-0.2           |   py35hfebbdb8_0          14 KB
    ------------------------------------------------------------
                                           Total:        16.8 MB

The following NEW packages will be INSTALLED:

  certifi            pkgs/main/win-64::certifi-2018.8.24-py35_1
  pip                pkgs/main/win-64::pip-10.0.1-py35_0
  python             pkgs/main/win-64::python-3.5.6-he025d50_0
  setuptools         pkgs/main/win-64::setuptools-40.2.0-py35_0
  vc                 pkgs/main/win-64::vc-14.1-h0510ff6_4
  vs2015_runtime     pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_3
  wheel              pkgs/main/win-64::wheel-0.31.1-py35_0
  wincertstore       pkgs/main/win-64::wincertstore-0.2-py35hfebbdb8_0
```

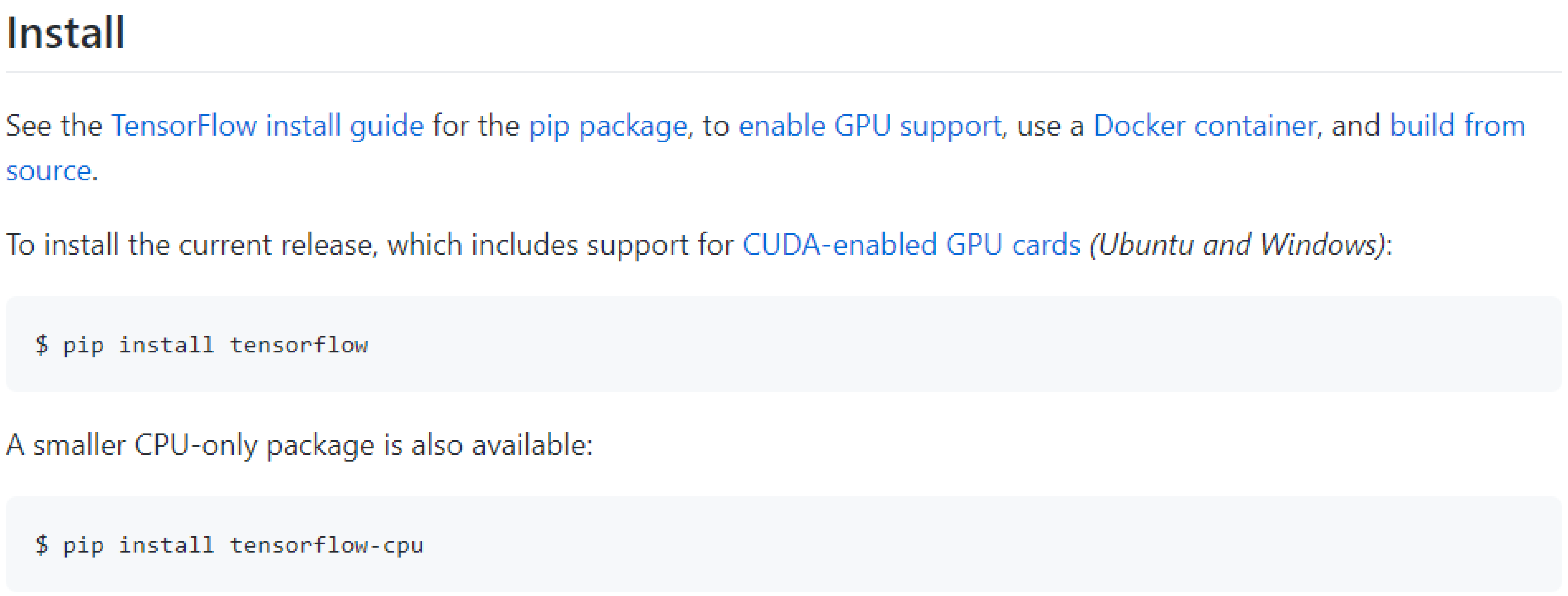
내가 사용할 가상환경을 만들고 해당 가상환경에서 필요 패키지들을 설치하자!

conda create – name <가상환경명> python=3.x

Unit 02 | 환경구축

프로그램을 실행하고 싶은데
어떻게 환경을 만들지?

Unit 02 | 환경구축



Install

See the TensorFlow install guide for the pip package, to enable GPU support, use a Docker container, and build from source.

To install the current release, which includes support for CUDA-enabled GPU cards *(Ubuntu and Windows)*:

```
$ pip install tensorflow
```

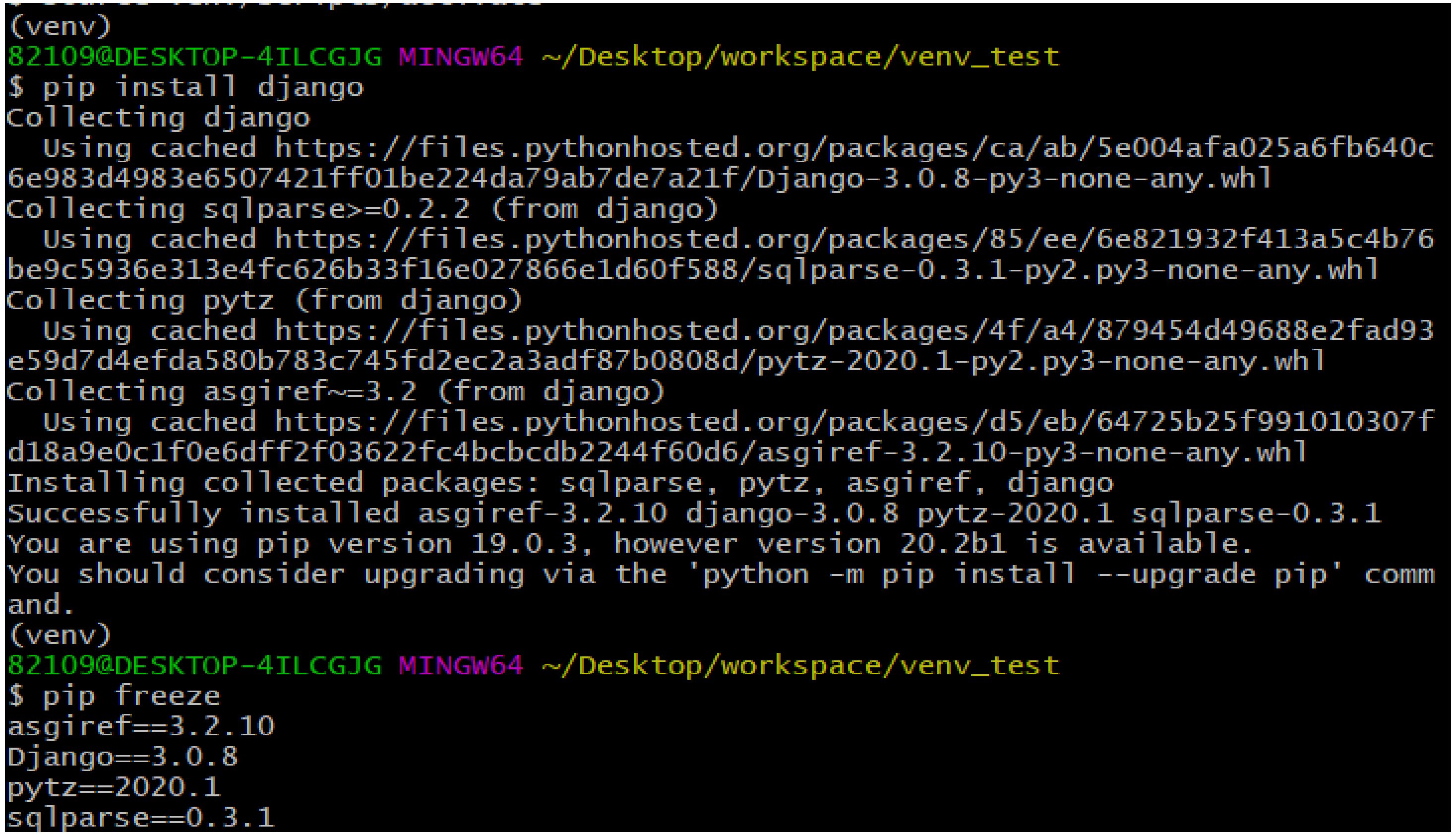
A smaller CPU-only package is also available:

```
$ pip install tensorflow-cpu
```

Unit 02 | 환경구축

https://github.com/MVIG-SJTU/AlphaPose

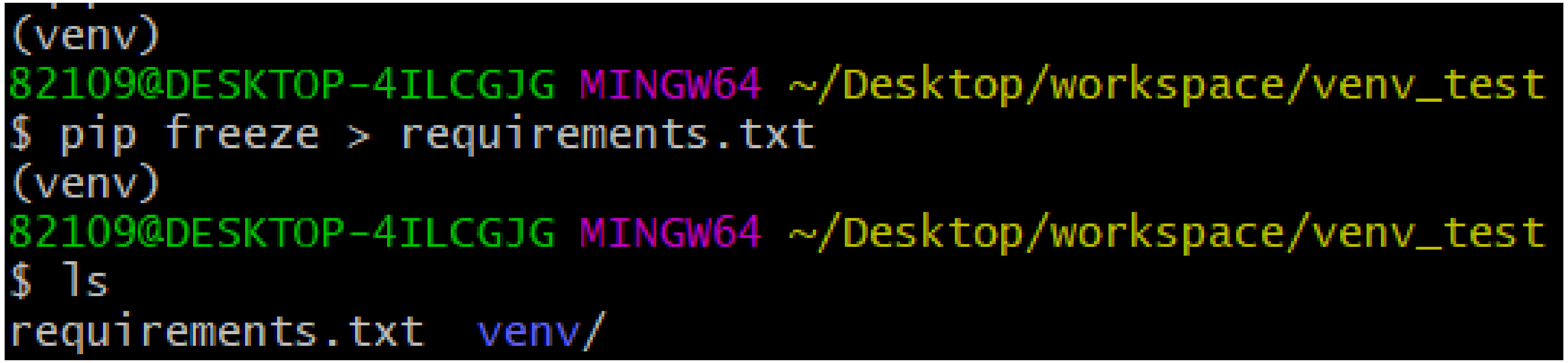
Unit 02 | 환경구축



```
(venv)
82109@DESKTOP-4ILCGJG MINGW64 ~/Desktop/workspace/venv_test
$ pip install django
Collecting django
  Using cached https://files.pythonhosted.org/packages/ca/ab/5e004afa025a6fb640c
6e983d4983e6507421ff01be224da79ab7de7a21f/Django-3.0.8-py3-none-any.whl
Collecting sqlparse>=0.2.2 (from django)
  Using cached https://files.pythonhosted.org/packages/85/ee/6e821932f413a5c4b76
be9c5936e313e4fc626b33f16e027866e1d60f588/sqlparse-0.3.1-py2.py3-none-any.whl
Collecting pytz (from django)
  Using cached https://files.pythonhosted.org/packages/4f/a4/879454d49688e2fad93
e59d7d4efda580b783c745fd2ec2a3adf87b0808d/pytz-2020.1-py2.py3-none-any.whl
Collecting asgiref~=3.2 (from django)
  Using cached https://files.pythonhosted.org/packages/d5/eb/64725b25f991010307f
d18a9e0c1f0e6dff2f03622fc4bcbcdb2244f60d6/asgiref-3.2.10-py3-none-any.whl
Installing collected packages: sqlparse, pytz, asgiref, django
Successfully installed asgiref-3.2.10 django-3.0.8 pytz-2020.1 sqlparse-0.3.1
You are using pip version 19.0.3, however version 20.2b1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.
(venv)
82109@DESKTOP-4ILCGJG MINGW64 ~/Desktop/workspace/venv_test
$ pip freeze
asgiref==3.2.10
Django==3.0.8
pytz==2020.1
sqlparse==0.3.1
```

pip freeze

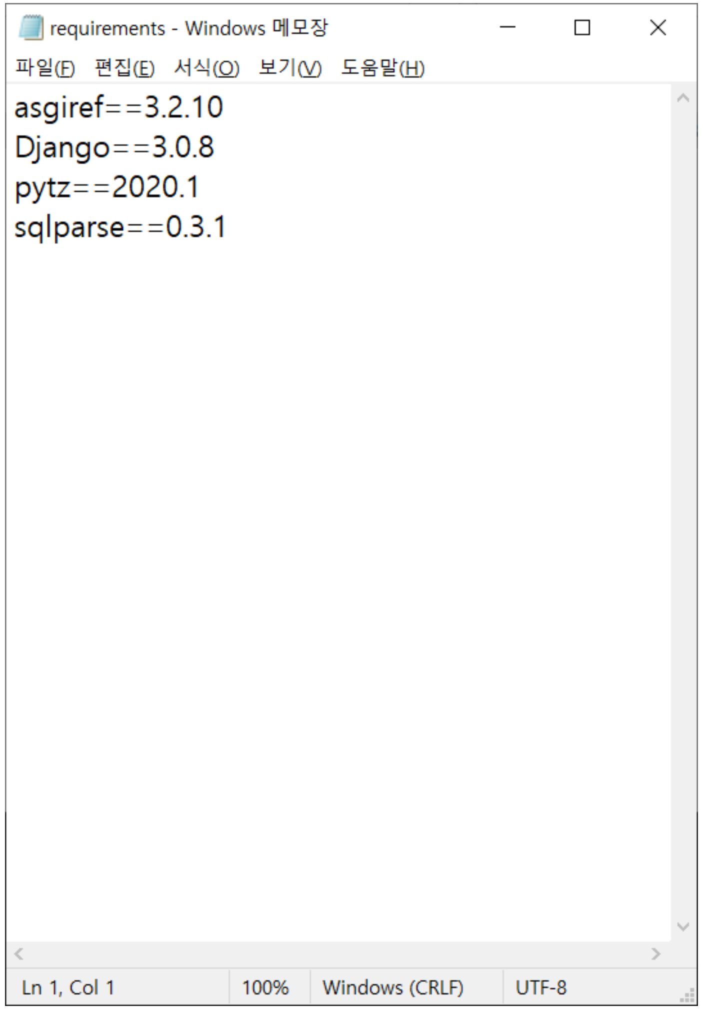
Unit 02 | 환경구축



```
(venv)
82109@DESKTOP-4ILCGJG MINGW64 ~/Desktop/workspace/venv_test
$ pip freeze > requirements.txt
(venv)
82109@DESKTOP-4ILCGJG MINGW64 ~/Desktop/workspace/venv_test
$ ls
requirements.txt  venv/
```

pip freeze > requirements.txt

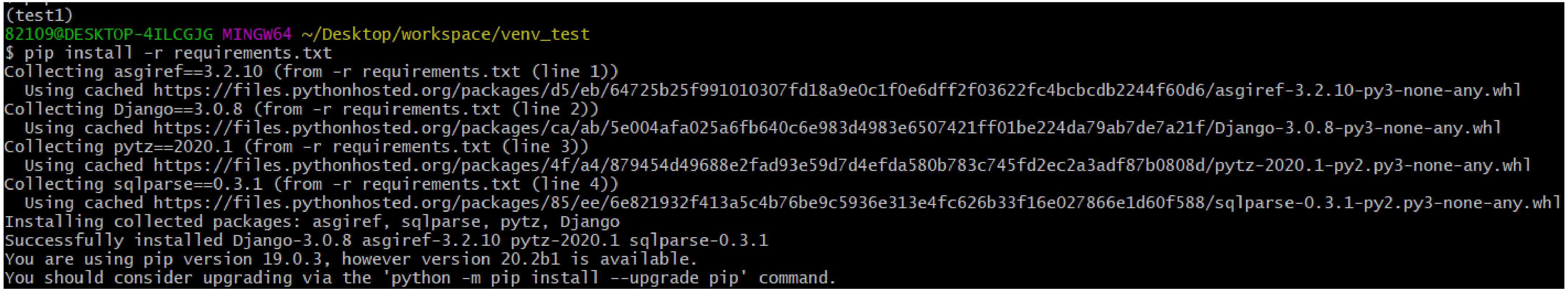
Unit 02 | 환경구축


requirements - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
asgiref==3.2.10
Django==3.0.8
pytz==2020.1
sqlparse==0.3.1
```

Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

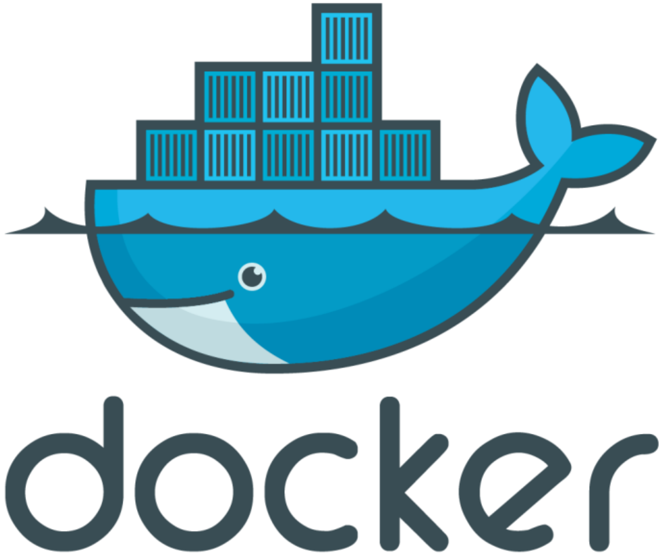
Unit 02 | 환경구축



```
(test1)
82109@DESKTOP-4ILCGJG MINGW64 ~/Desktop/workspace/venv_test
$ pip install -r requirements.txt
Collecting asgiref==3.2.10 (from -r requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/d5/eb/64725b25f991010307fd18a9e0c1f0e6dff2f03622fc4bcbcdb2244f60d6/asgiref-3.2.10-py3-none-any.whl
Collecting Django==3.0.8 (from -r requirements.txt (line 2))
  Using cached https://files.pythonhosted.org/packages/ca/ab/5e004afa025a6fb640c6e983d4983e6507421ff01be224da79ab7de7a21f/Django-3.0.8-py3-none-any.whl
Collecting pytz==2020.1 (from -r requirements.txt (line 3))
  Using cached https://files.pythonhosted.org/packages/4f/a4/879454d49688e2fad93e59d7d4efda580b783c745fd2ec2a3adf87b0808d/pytz-2020.1-py2.py3-none-any.whl
Collecting sqlparse==0.3.1 (from -r requirements.txt (line 4))
  Using cached https://files.pythonhosted.org/packages/85/ee/6e821932f413a5c4b76be9c5936e313e4fc626b33f16e027866e1d60f588/sqlparse-0.3.1-py2.py3-none-any.whl
Installing collected packages: asgiref, sqlparse, pytz, Django
Successfully installed Django-3.0.8 asgiref-3.2.10 pytz-2020.1 sqlparse-0.3.1
You are using pip version 19.0.3, however version 20.2b1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

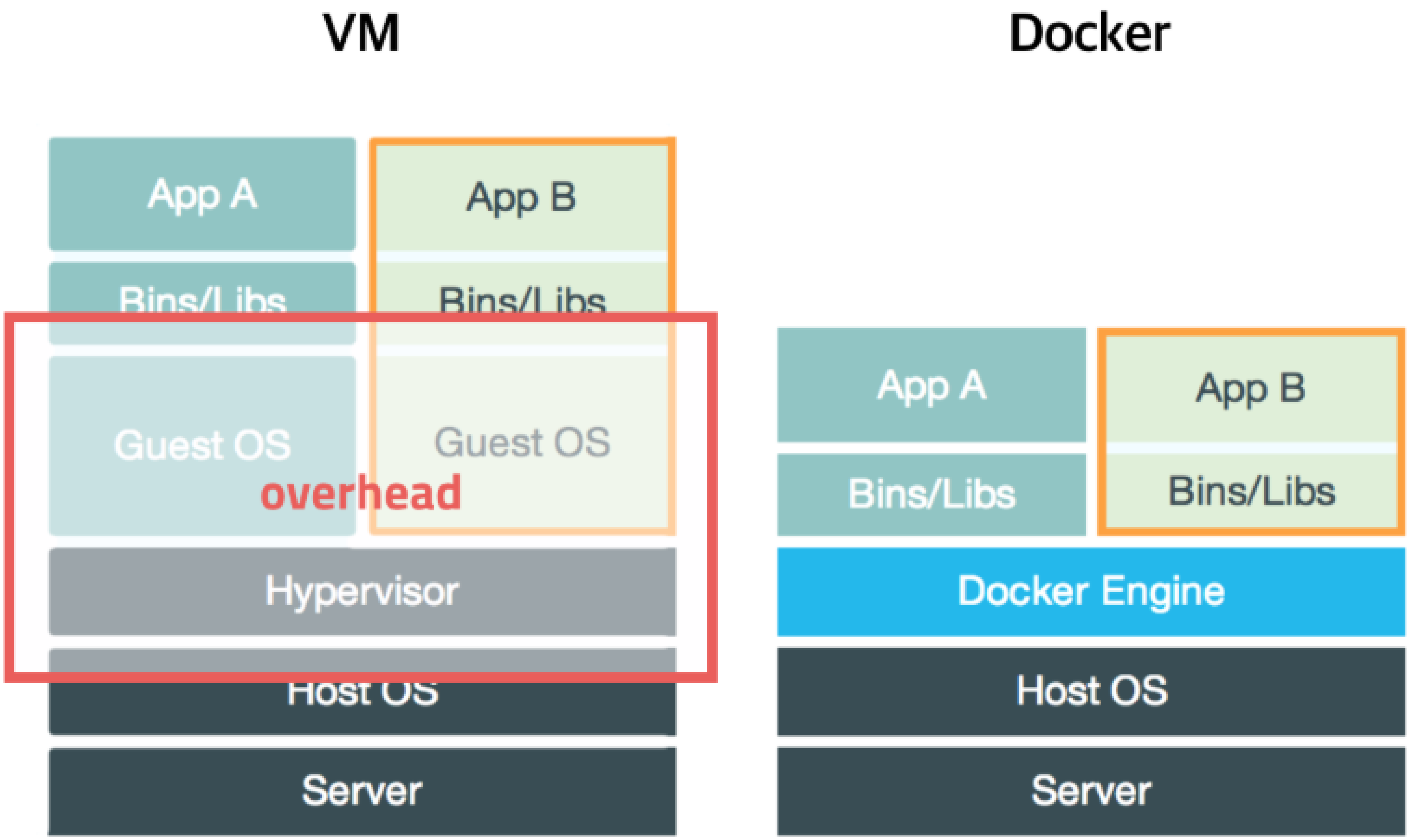
pip install –r requirements.txt

Unit 02 | 환경구축

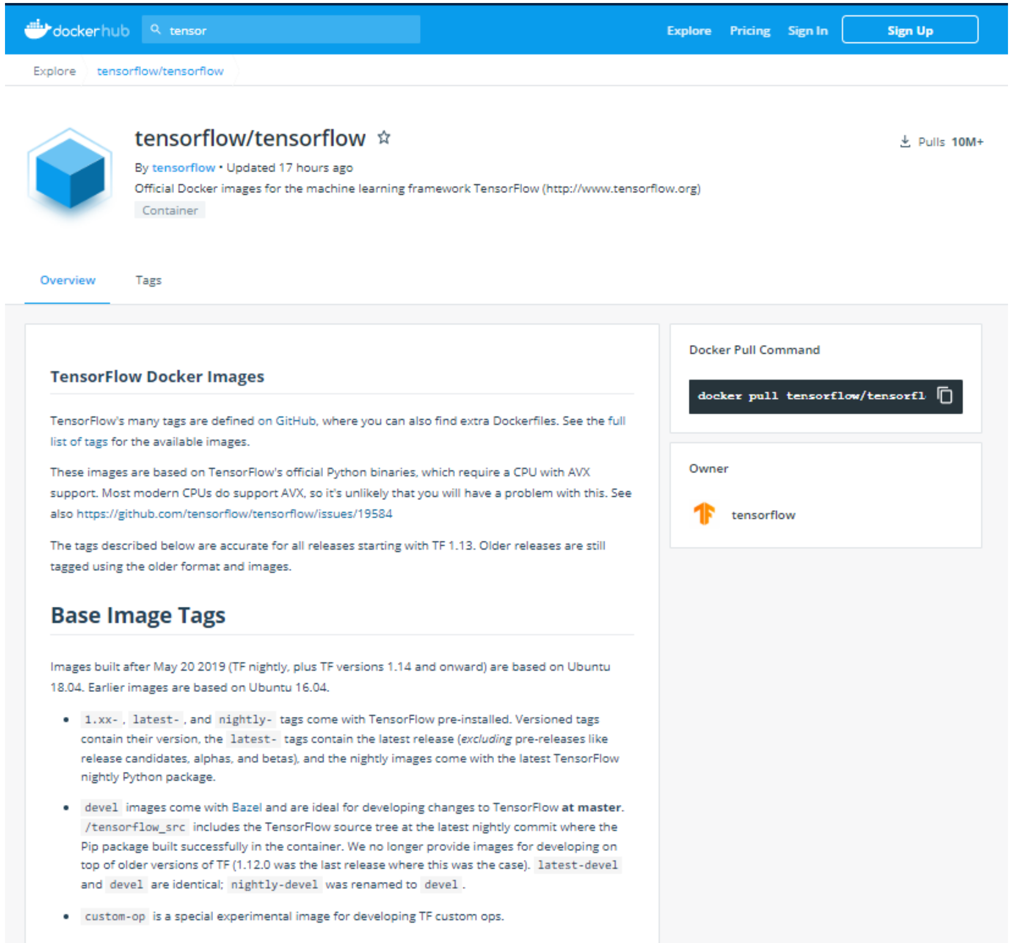


컨테이너 기반의 오픈소스 가상화 플랫폼

Unit 02 | 환경구축


VM
Docker
App A
App B
Bins/Libs
Bins/Libs
Guest OS
Guest OS
overhead
Hypervisor
Host OS
Server
App A
App B
Bins/Libs
Bins/Libs
Docker Engine
Host OS
Server

Unit 02 | 환경구축



tensorflow/tensorflow ☆

By tensorflow • Updated 17 hours ago

Official Docker images for the machine learning framework TensorFlow (http://www.tensorflow.org)

TensorFlow Docker Images

TensorFlow's many tags are defined on GitHub, where you can also find extra Dockerfiles. See the full list of tags for the available images.

These images are based on TensorFlow's official Python binaries, which require a CPU with AVX support. Most modern CPUs do support AVX, so it's unlikely that you will have a problem with this. See also https://github.com/tensorflow/tensorflow/issues/19584

The tags described below are accurate for all releases starting with TF 1.13. Older releases are still tagged using the older format and images.

Base Image Tags

Images built after May 20 2019 (TF nightly, plus TF versions 1.14 and onward) are based on Ubuntu 18.04. Earlier images are based on Ubuntu 16.04.

- `1.xx-` , `latest-` , and `nightly-` tags come with TensorFlow pre-installed. Versioned tags contain their version, the `latest-` tags contain the latest release (*excluding* pre-releases like release candidates, alphas, and betas), and the nightly images come with the latest TensorFlow nightly Python package.
- `devel` images come with Bazel and are ideal for developing changes to TensorFlow **at master**. `/tensorflow_src` includes the TensorFlow source tree at the latest nightly commit where the Pip package built successfully in the container. We no longer provide images for developing on top of older versions of TF (1.12.0 was the last release where this was the case). `latest-devel` and `devel` are identical; `nightly-devel` was renamed to `devel` .
- `custom-op` is a special experimental image for developing TF custom ops.

Docker Pull Command

```
docker pull tensorflow/tensorfl
```

Owner

tensorflow

Docker hub

Unit 03 | 과제 및 실습

AWS EC2

아마존 웹 서비스, Elastic Compute Cloud

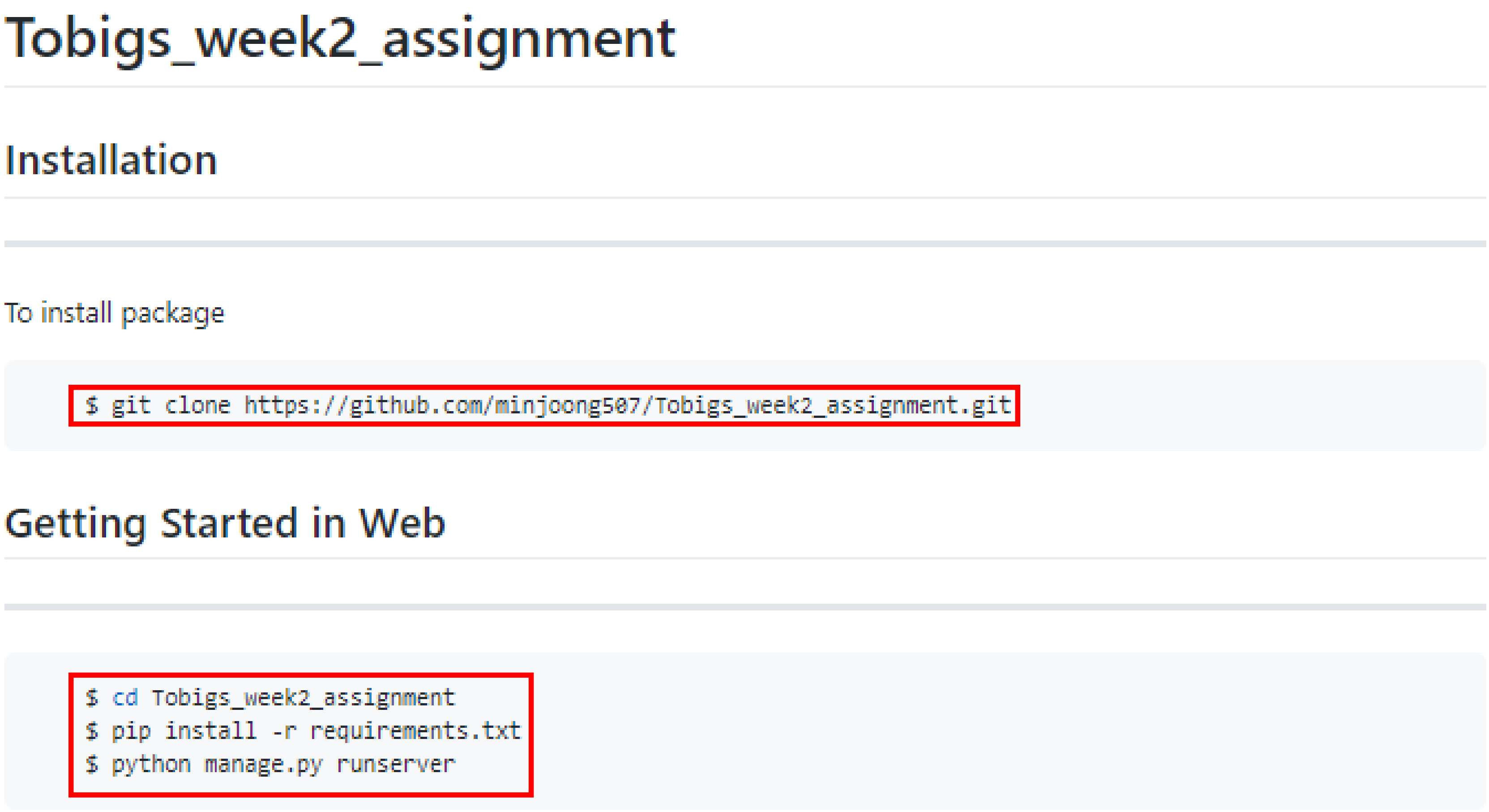
Unit 03 | 과제 및 실습

Assignment

EC2 환경에서 코드를 돌려보자! AWS를 처음 사용하시는 분도 계시고 파이썬 환경 자체를 다뤄보는게 분명 어려우신 분들도 계시기 때문에 이번 과제는 의무 제출이 아닙니다! 제가 안내한 github의 소스코드를 작동시키는 과정을 따라하는게 이번 과제의 목표입니다.

Github address : https://github.com/minjoong507/Tobigs_week2_assignment

Unit 03 | 과제 및 실습



Tobigs_week2_assignment

Installation

To install package

```
$ git clone https://github.com/minjoong507/Tobigs_week2_assignment.git
```

Getting Started in Web

```
$ cd Tobigs_week2_assignment
$ pip install -r requirements.txt
$ python manage.py runserver
```

Q&A

들어주셔서 감사합니다.