

## Tensorflow-GPU (Windows 환경 기준)

### 1) Anaconda 설치

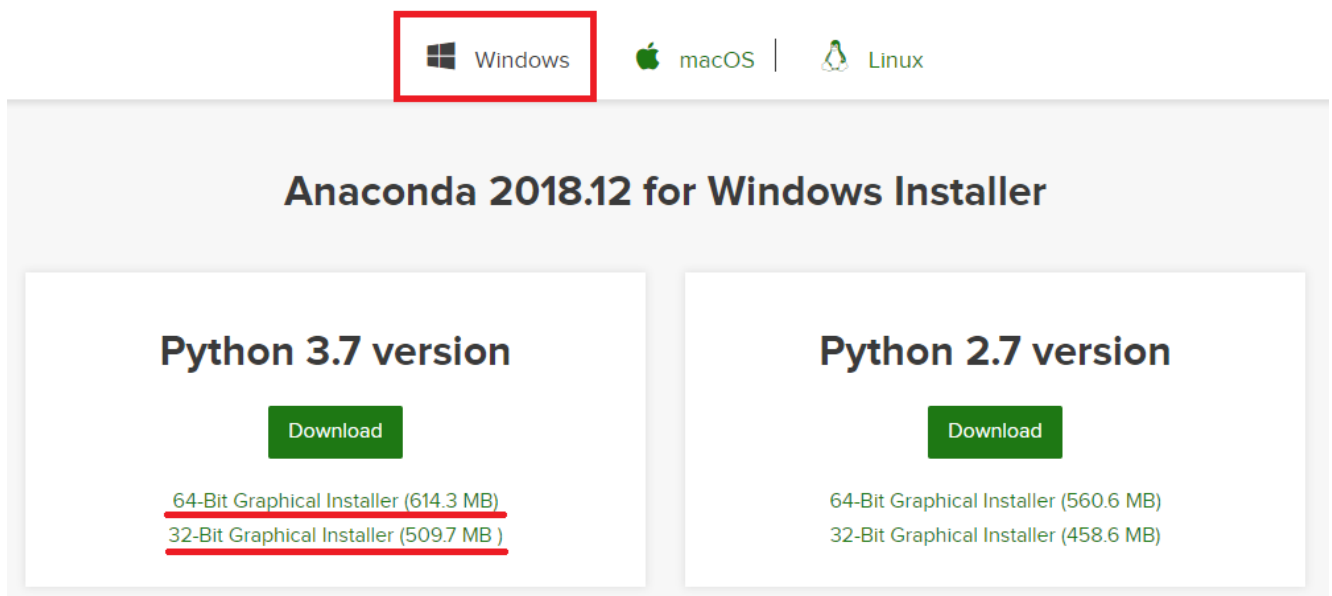
우선 가장 먼저 해야 할 일은 아나콘다를 설치하는 일입니다.

※ **Anaconda**란, 수학, 과학 등 다양한 분야의 파이썬 패키지들을 한 데 모아 둔 플랫폼입니다. 파이썬 사용자들이 하여금 필요한 패키지들을 쉽게 이용하고 관리할 수 있도록 만들었다는 점에서 활용도와 기여도가 높습니다.

#### ① 아나콘다 설치파일 다운로드

<https://www.anaconda.com/distribution/#download-section>

(빨간 네모박스로 표시된 윈도우 아이콘 옆의 OS는 순서대로 맥과 리눅스를 의미)



#### ※ 유의사항 ※

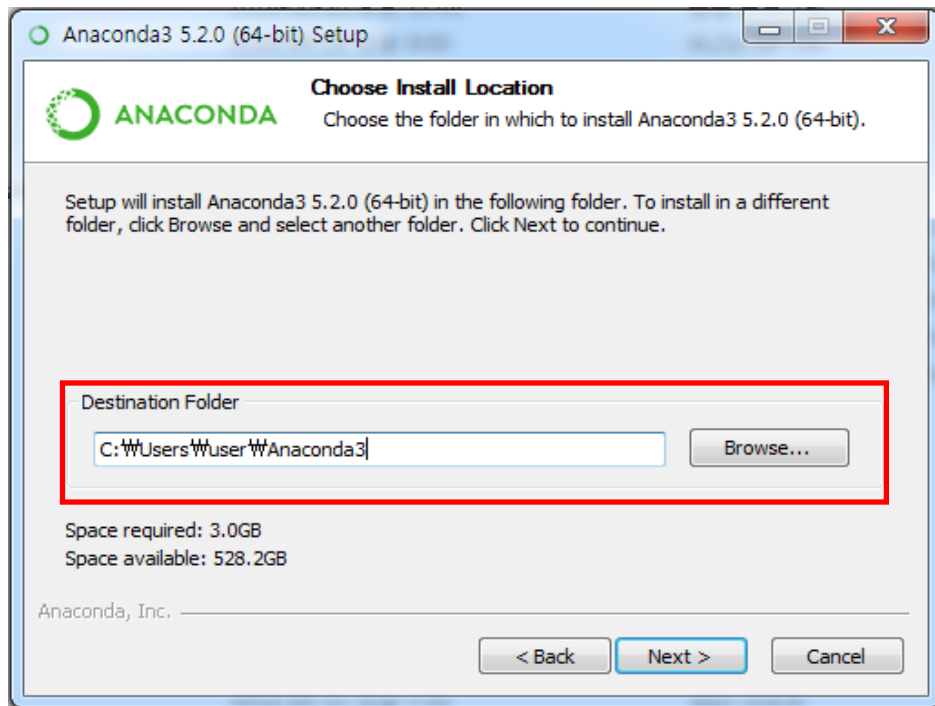
- ① **Version check!** 아나콘다의 버전마다 그에 맞게 연동되는 파이썬의 버전이 다릅니다. 이 버전을 꼭 체크해 주셔야 합니다. (아나콘다 5.2는 파이썬 3.6과 연동됩니다.)
- ② **Run as administrator.** 웹사이트에서 다운로드한 설치파일과 설치 이후에 실행할 명령 프롬프트 모두 **관리자 권한으로 실행**해 줍니다.

위 그림의 윈도우 아이콘을 선택하면 아래 화면으로 이동합니다.

32비트/64비트 운영체제 각각 설치파일이 따로 있으니,

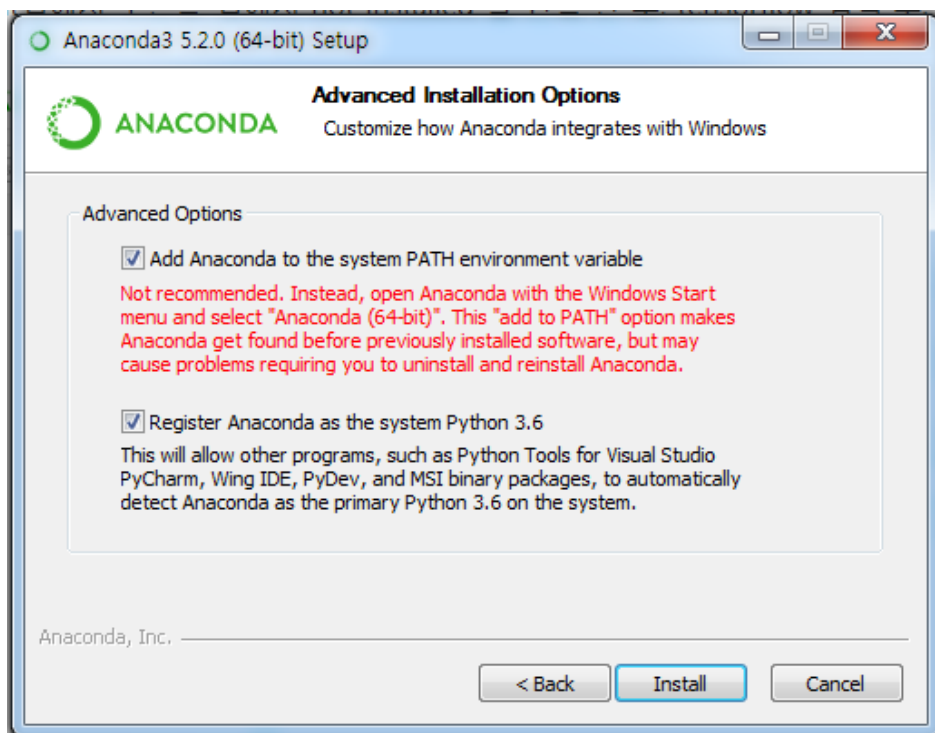
**본인의 컴퓨터 사양에 맞는 것을 선택합니다.**

## ② 아나콘다 설치



경로명에 한글이 포함되어 있지는 않은지 확인합니다.

몇몇 파이썬 라이브러리의 호환성 문제 때문에,  
설치대상 폴더에는 아스키 코드가 아닌 문자가 포함될 수 없습니다.



두 옵션 모두 체크해 줍니다.

항목 1: 자동으로 시스템 환경변수에 아나콘다를 추가해 주는 옵션  
항목 2: 아나콘다를 시스템의 메인 파이썬으로 인식하기 위한 옵션

2) CUDA와 cuDNN 설치 (GPU를 사용하지 않는 실습생은 3단계로 가시면 됩니다.)

**CUDA:** NVIDIA에서 제공하는 병렬처리 연산을 위한 플랫폼. GPU 사용 환경 제공.

**cuDNN:** NVIDIA의 CUDA 환경에서 Deep neural network를 지원하는 라이브러리.

아나콘다와 파이썬처럼, CUDA와 cuDNN도 버전에 따라 페어가 존재합니다.

Table 1. CUDA Toolkit and Compatible Driver Versions

CUDA Toolkit	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 7.0 (7.0.28)	>= 346.46	>= 347.62
CUDA 7.5 (7.5.16)	>= 352.31	>= 353.66
CUDA 8.0 (8.0.44)	>= 367.48	>= 369.30
CUDA 8.0 (8.0.61 GA2)	>= 375.26	>= 376.51
CUDA 9.0 (9.0.76)	>= 384.81	>= 385.54
CUDA 9.1 (9.1.85)	>= 390.46	>= 391.29
CUDA 9.2 (9.2.88)	>= 396.26	>= 397.44
CUDA 9.2 (9.2.148 Update 1)	>= 396.37	>= 398.26

NVIDIA 그래픽카드 드라이버와 CUDA 플랫폼 사이의 호환성  
그래픽카드의 드라이버 버전과 CUDA의 버전 사이에 호환이 가능해야 합니다.

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

[Download cuDNN v7.1.2 \(Mar 21, 2018\), for CUDA 9.1 & 9.2](#)

[Download cuDNN v7.1.2 \(Mar 21, 2018\), for CUDA 9.0](#)

[Download cuDNN v7.0.5 \(Dec 11, 2017\), for CUDA 9.1](#)

[Download cuDNN v7.0.5 \(Dec 5, 2017\), for CUDA 9.0](#)

[Download cuDNN v7.0.5 \(Dec 5, 2017\), for CUDA 8.0](#)

[Download cuDNN v7.0.4 \(Nov 13, 2017\), for CUDA 9.0](#)

[Download cuDNN v6.0 \(April 27, 2017\), for CUDA 8.0](#)

[Download cuDNN v6.0 \(April 27, 2017\), for CUDA 7.5](#)

[Download cuDNN v5.1 \(Jan 20, 2017\), for CUDA 8.0](#)

[Download cuDNN v5.1 \(Jan 20, 2017\), for CUDA 7.5](#)

어느 버전의 CUDA를 설치했느냐에 따라 cuDNN의 버전도 같이 결정됩니다.

CUDA 설치 링크: <https://developer.nvidia.com/cuda-90-download-archive>

cuDNN 설치 링크: <https://developer.nvidia.com/cudnn>

## CUDA Toolkit 9.0 Downloads

### Select Target Platform ⓘ

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System

Windows

Linux

Mac OSX

[2019년 2월] CUDA 설치 링크의 첫 화면입니다.

현재 기준 최신 버전인 CUDA 10.0가 pip와 아직 호환되지 않는 문제가 있기 때문에,  
CUDA 9.0을 설치하여 실습을 진행하겠습니다.

## CUDA Toolkit 9.0 Downloads

### Select Target Platform ⓘ

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System

Windows

Linux

Mac OSX

Architecture ⓘ

x86\_64

Version

10

8.1

7

Server 2016

Server 2012 R2

Installer Type ⓘ

exe [network]

exe [local]

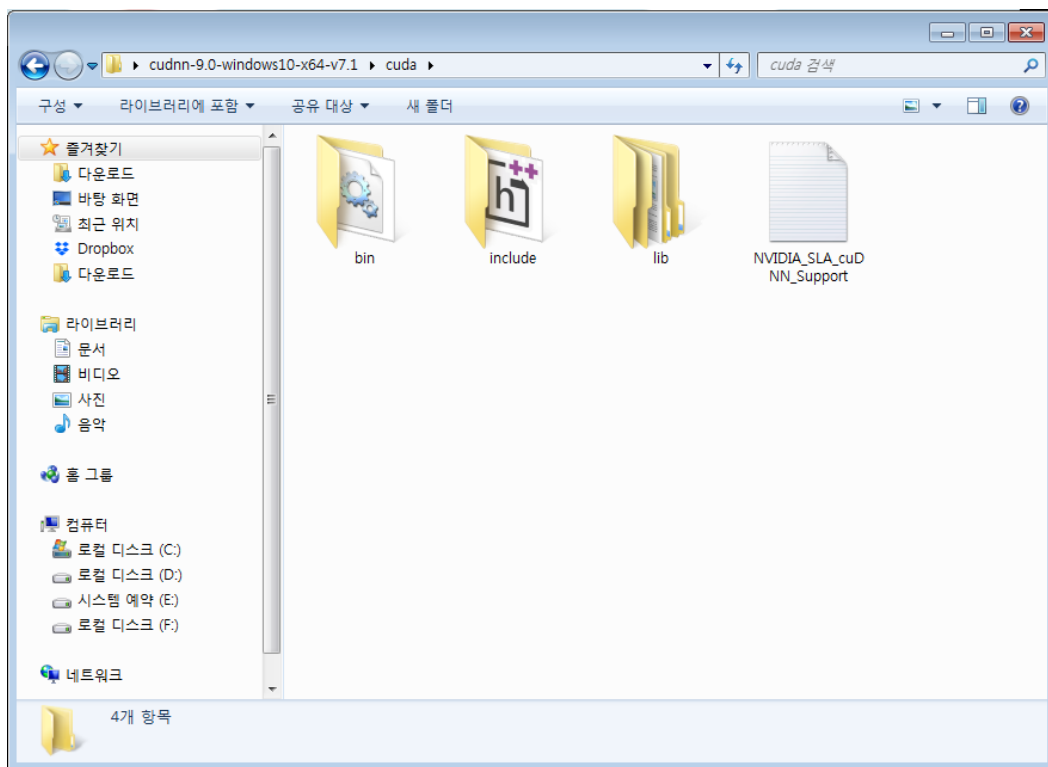
본인의 OS에 맞는 CUDA 설치파일을 다운로드받고, CUDA를 설치합니다.

# Membership Required

The downloadable file or page you have requested, requires membership of the NVIDIA Developer Program. Please login to gain access or use the button below and complete the short application for this free to join program. Thank you.

Join now

CUDA를 설치한 후에는, cuDNN을 설치합니다.  
cuDNN을 설치하기 위해서는, 별도의 회원가입과 로그인 과정이 필요합니다.



다운받은 cuDNN 압축파일을 압축 해제한 모습

본인이 설치한 CUDA의 버전과 연동되는 cuDNN을 설치한 후,  
다운로드받은 압축파일을 해제하고 3개의 파일을 CUDA 설치 경로에 넣어  
줍니다.

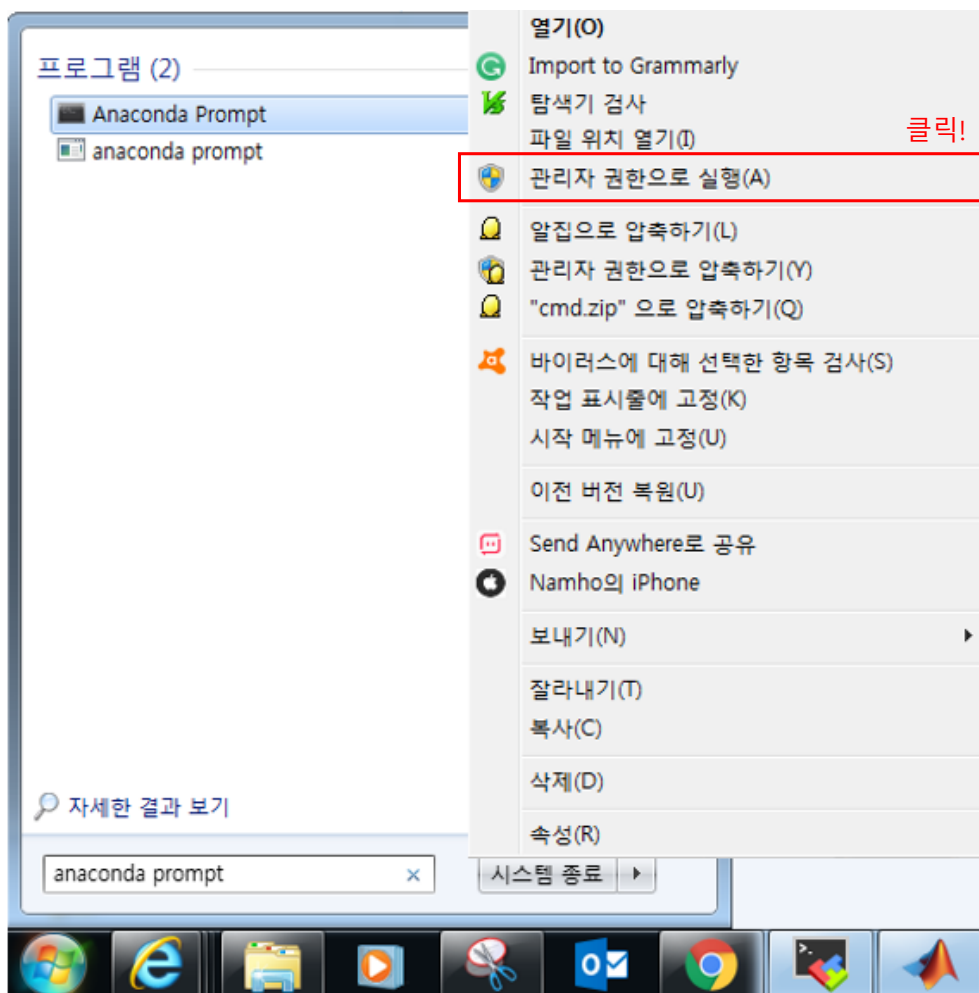
CUDA-9.0 기본 경로: <C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0>

bin 폴더의 cudnn64\_7.dll, include의 cudnn.h, lib의 cudnn.lib을  
각각 CUDA 경로의 bin, include, lib으로 옮겨 주시면 됩니다.

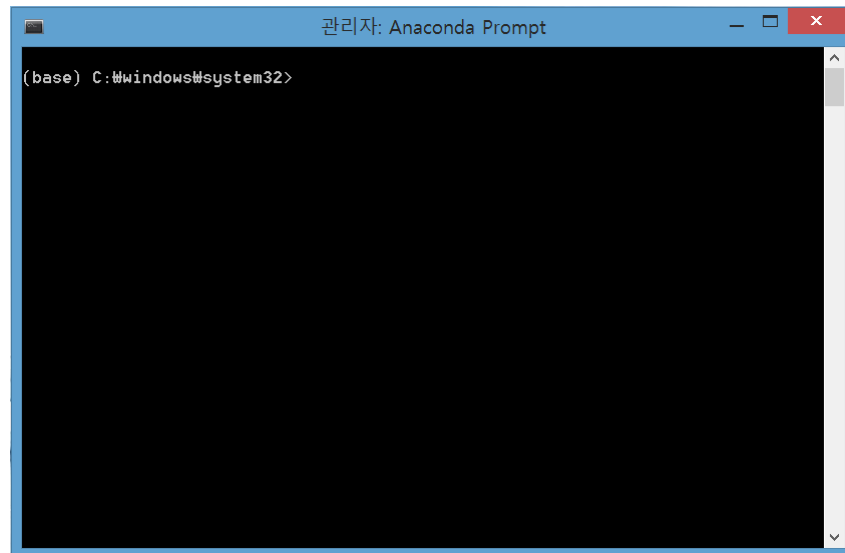
### 3) 가상환경 만들기

아나콘다를 설치한 후에는, 아나콘다 명령 프롬프트를 실행하고 가상환경을 만들어 주어야 합니다. 가상환경이 필요한 이유는, 파이썬 라이브러리들이 서로 충돌하는 것을 사전에 예방하기 위해서입니다. 라이브러리 간 충돌이란, 파이썬의 다양한 버전의 다양한 모듈들이 서로 의존성을 갖고 있는 경우가 많기 때문에 연동이 불가한 부적절한 버전과 사용하고자 했을 때 오작동을 일으키는 것을 의미합니다. **이것을 막기 위한 가상환경은 개개인이 원하는 파이썬 환경을 구축하기 위해, 필요한 모듈들만 담아 놓는 바구니라고 생각하면 쉽습니다.** 독립적인 가상환경들을 구축함으로써, 여러 프로젝트를 한 머신에서 충돌 없이 병렬적으로 수행하는 것이 가능합니다.

Virtualenv와 conda 중 필자는 conda 가상환경을 생성해 보았습니다.



아나콘다를 설치했으니, 아나콘다 명령 프롬프트를 실행합니다.

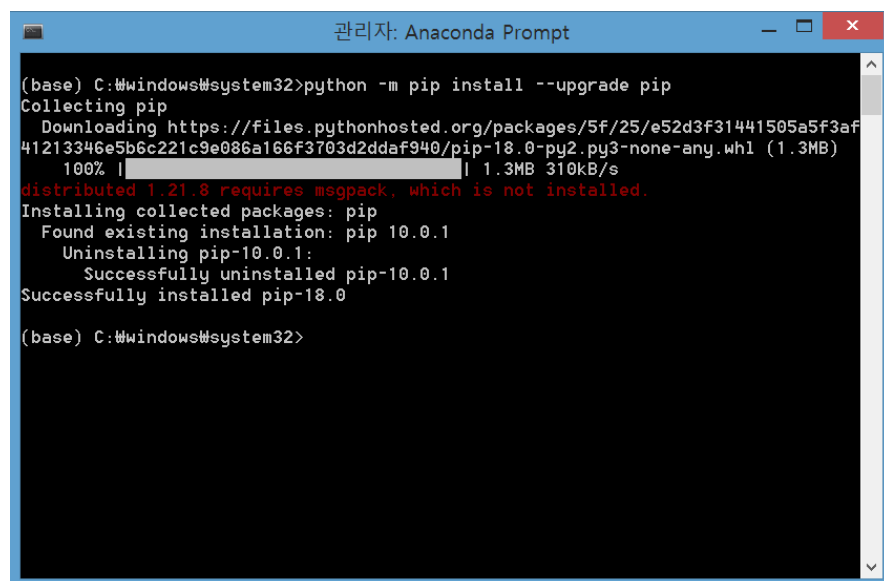


실행 첫 화면

커맨드 창을 실행한 이후에는, 다음 명령어들을 차례대로 수행합니다.

- ① `python -m pip install --upgrade pip`
- ② `conda create -n <가상환경 이름> python=3.7`
- ③ `conda activate <가상환경 이름>`
- ④ `pip install tensorflow-gpu`
- ⑤ `conda install matplotlib`
- ⑥ `pip install keras`
- ⑦ `conda install jupyter notebook`

① **pip**를 설치합니다. 파이썬으로 작성된 패키지 라이브러리를 총괄하는 시스템입니다.



pip 설치 결과

- ② 본격적으로 가상환경을 생성하는 문장입니다. 이 때, 우리가 설치한 아나콘다와 연동되는 파이썬의 버전을 명시해 주어야 합니다. 이것은, 앞서 아나콘다를 설치할 때 파이썬의 버전을 체크해야 했던 이유 중 하나입니다.
- 필자는 가상환경 이름을 tensorflow로 지정해 주었습니다.

```
관리자: Anaconda Prompt

(base) C:\windows\system32>conda create -n tensorflow python=3.6
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.5.4
  latest version: 4.5.8

Please update conda by running

  $ conda update -n base conda

## Package Plan ##

environment location: C:\ProgramData\Anaconda3\envs\tensorflow

added / updated specs:
- python=3.6

The following packages will be downloaded:

package | build | size
-----|-----|-----
uc-14.1 | h0510ff6_3 | 5 KB
python-3.6.6 | hea74fb7_0 | 21.6 MB
vs2015_runtime-15.5.2 | 3 | 2.2 MB
setuptools-39.2.0 | py36_0 | 571 KB
-----|-----|-----
Total: 24.3 MB

The following NEW packages will be INSTALLED:

certifi: 2018.4.16-py36_0
pip: 10.0.1-py36_0
python: 3.6.6-hea74fb7_0
setuptools: 39.2.0-py36_0
uc: 14.1-h0510ff6_3
vs2015_runtime: 15.5.2-3
wheel: 0.31.1-py36_0
wincertstore: 0.2-py36h7fe50ca_0

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

가상환경 생성

파란 네모박스: 가상환경 이름으로 폴더가 생성되는 것을 확인할 수 있습니다.



- ③ **가상환경을 활성화**하는 명령어입니다. 이 명령어는 해당 가상환경을 메인 시스템이나 다른 가상환경과 구분짓는 역할을 합니다. 버전에 따라 activate <가상환경 이름> 만으로 이 명령을 수행하는 경우도 있는데, 가상환경을 설치하고 나서 아래 붉은 네모박스와 같이 사용자에게 명령어를 인지시켜 주는 부분을 확인하면 됩니다.

```
Downloading and Extracting Packages
uc-14.1 | 5 KB | ##### | 100%
python-3.6.6 | 21.6 MB | ##### | 100%
vs2015_runtime-15.5. | 2.2 MB | ##### | 100%
setuptools-39.2.0 | 571 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate tensorflow
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:\windows\system32>
```

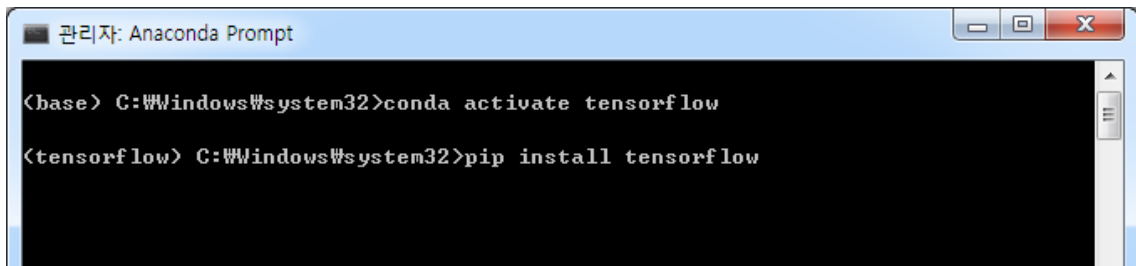
가상환경 생성 마지막 부분에 사용자에게 활성화/비활성화 명령어를 알려 줍니다.

```
(base) C:\windows\system32>conda activate tensorflow
(tensorflow) C:\windows\system32>
```

가상환경을 활성화시키고 나니, 문장 앞에 (tensorflow) 가 추가되었습니다.

④ Tensorflow 라이브러리를 설치합니다.

CPU 버전을 사용하고 싶다면, tensorflow-gpu 대신 **tensorflow**로 작성해야 합니다.

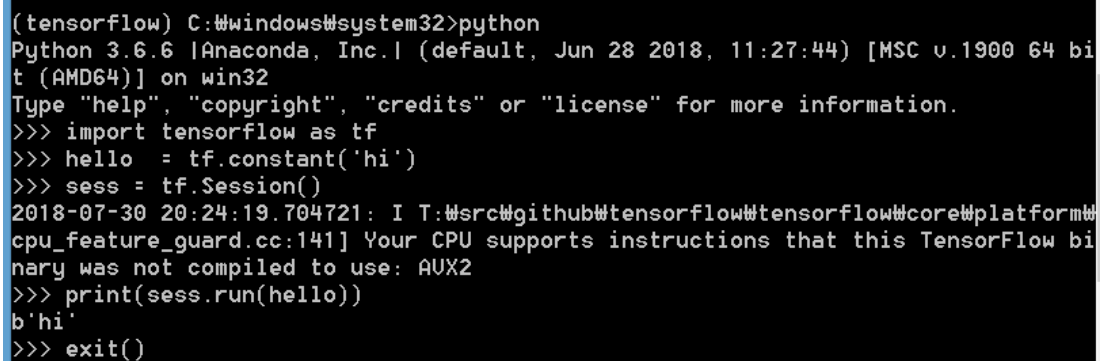


```
관리자: Anaconda Prompt

(base) C:\Windows\system32>conda activate tensorflow

(tensorflow) C:\Windows\system32>pip install tensorflow
```

Tensorflow 라이브러리 설치



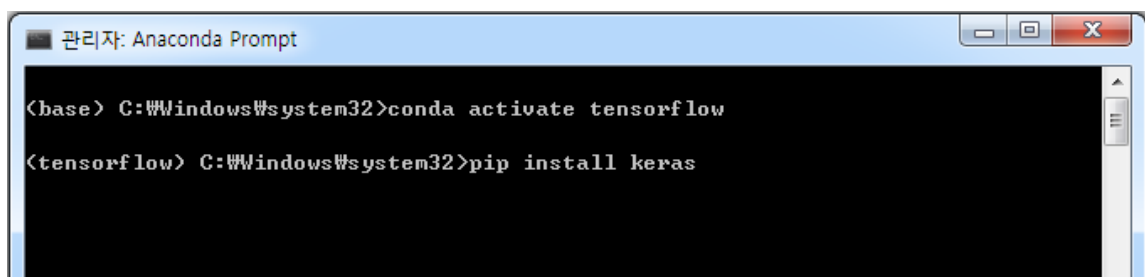
```
(tensorflow) C:\Windows\system32>python
Python 3.6.6 |Anaconda, Inc.| (default, Jun 28 2018, 11:27:44) [MSC v.1900 64 bi
t (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> hello = tf.constant('hi')
>>> sess = tf.Session()
2018-07-30 20:24:19.704721: I T:\src\github\tensorflow\tensorflow\core\platform\w
cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow bi
nary was not compiled to use: AVX2
>>> print(sess.run(hello))
b'hi'
>>> exit()
```

Tensorflow가 잘 설치되었는지 확인하는 방법입니다.

⑤ conda 명령어로 matplotlib 라이브러리를 설치합니다.

⑥ Keras 라이브러리를 설치합니다.

위의 Tensorflow 때처럼 python을 열고 Keras가 잘 설치되었는지 확인해 줍니다.

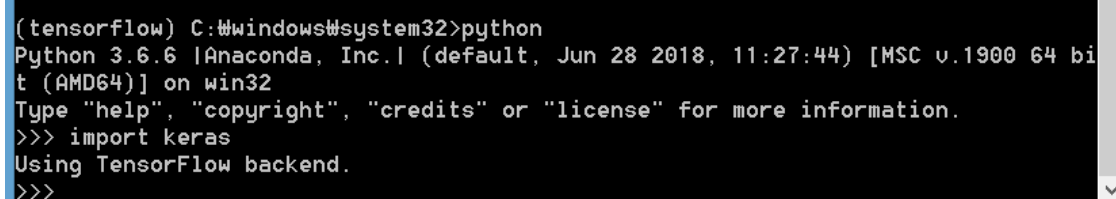


```
관리자: Anaconda Prompt

(base) C:\Windows\system32>conda activate tensorflow

(tensorflow) C:\Windows\system32>pip install keras
```

Keras 라이브러리 설치



```
(tensorflow) C:\Windows\system32>python
Python 3.6.6 |Anaconda, Inc.| (default, Jun 28 2018, 11:27:44) [MSC v.1900 64 bi
t (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import keras
Using TensorFlow backend.
>>>
```

Keras가 잘 설치되었는지 확인하는 방법입니다.

## ⑦ Jupyter notebook을 설치합니다.

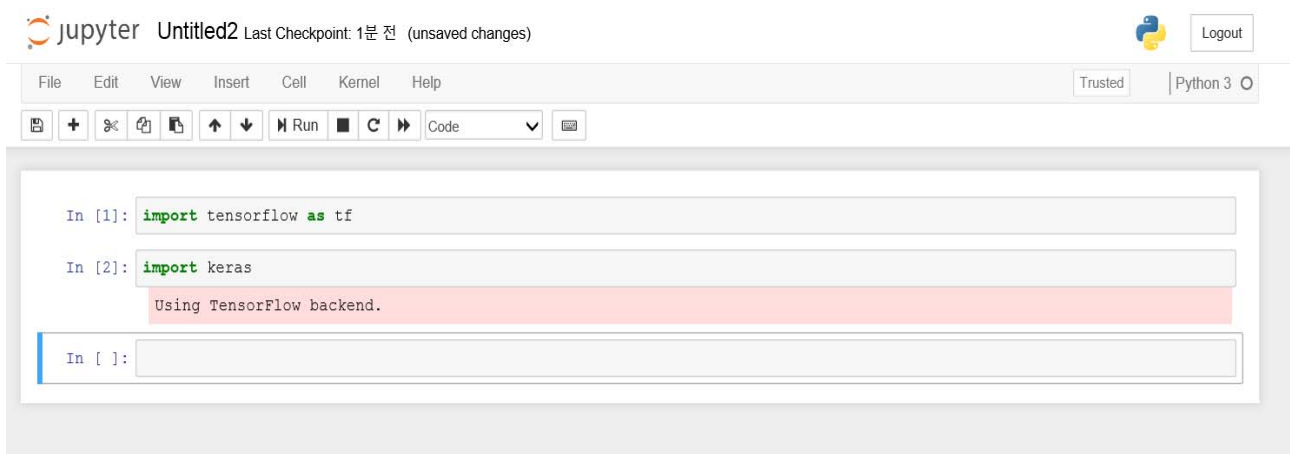
이것은 일종의 오픈 소스 웹 어플리케이션으로  
최근 머신 러닝, 통계 모델링, 데이터 클리닝  
등에 다양한 분야에 활용되고 있습니다.



가장 큰 장점은 실시간으로 마치 대화하듯 인터랙티브하게 데이터를 처리하고 시각화할 수 있도록 설계되어 있는 부분입니다. 40종의 다양한 프로그래밍 언어를 지원하고 깃허브, 드롭박스 등을 통해 간편하게 공유할 수 있다는 특징을 가진 개발 환경입니다.

```
관리자: Anaconda Prompt
(base) C:\Windows\system32>conda activate tensorflow
(tensorflow) C:\Windows\system32>conda install jupyter notebook
```

Jupyter notebook 설치



최종적으로, Jupyter notebook에서도 라이브러리가 잘 동작하는 것을 확인했습니다.

### \* Tensorflow의 특정 버전을 설치하고 싶다면? [Example: Tensorflow-gpu 1.0.0 RC0]

```
pip install -upgrade https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-1.0.0rc0-cp35-cp35m-win_amd64.whl
```

위와 같은 형태의 명령어가 버전에 따라 존재합니다.