

Section 1

데이터 분석 작업 환경-FE-PRO

- 1. 데이터 분석도구 개관
- 2. 윈10+아나콘다+텐서플로2-CPU
- 3. 윈10+아나콘다+텐서플로2-GPU
- Jupyter Notebook 원격 4· 접속하기



학습목표

- ❖ 이 워크샵에서는 분석도구 개관에 대해 알 수 있습니다.
- ❖ 윈10+아나콘다+텐서플로2를 설치할 수 있습니다.



Subsection 1

데이터 분석도구 개관



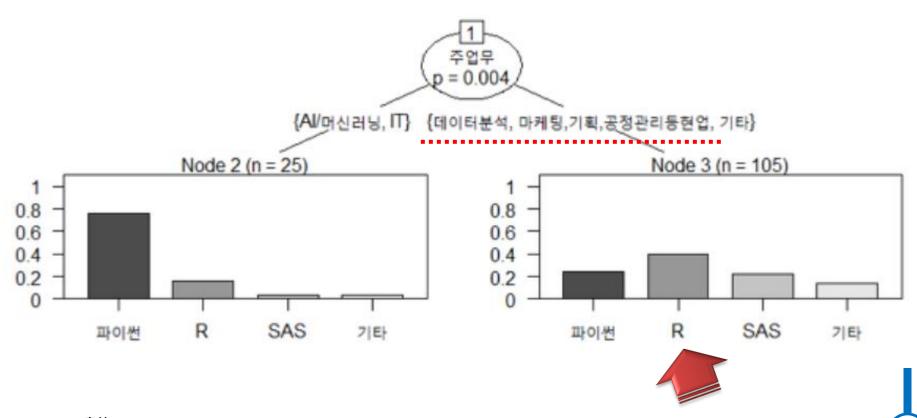
Python vs R

- ❖ 많은 사람들이 R이 통계학자를 위해 특별히 설계되었다고 말합니다(특히 쉽고 강력한 데이터 시각화 기능의 경우). 특히 주로 데이터 분석에 사용할 경우 비교적 배우기 쉽습니다.
- ❖ 반면에 Python은 데이터 분석을 넘어서서 다소 유연합니다. 많은 데이터 과학자와 기계 학습 실무자가 작성한 코드를 동적 웹 응용 프로그램에 통합 할 수 있기 때문에 Python을 선택할 수 있습니다.



Python vs R

- ❖ 2019년 구글 서베이를 사용한 온라인 조사 결과를 보면,
- AI/머신러닝 업무에서 주로 파이썬이 압도적으로 많이 사용되고 있습니다.
- 그러나 일반 데이터 분석에서는 R을 사용되고 있습니다.





Python vs R

- ❖ 주업무와 실무경력 교차 조합별 사용도구 비교
- 데이터 분석 영역에서 1년 이상 실무경력자가 다수 R을 사용하고 있습니다.
- AI/머신러닝 분야에서는 다수가 파이썬을 사용하고 있습니다.

주업무별실무경력별 사용도구 분포



 ${\tt \hat{Z}}{\tt https://docs.google.com/forms/d/e/1FAIpQLSe31ixLe3hGifL9rauxCM687QJ7lQ2S4xlVowrHpe7e8QpnmQ/viewform}$

데이터 분석도구 개관



Python vs R

❖ 주업무가 <u>데이터 분석</u>인 경우 R사용이 높습니다.



데이터 분석도구 개관



Python

❖ 1991년 네덜란드 귀도 반 로섬 (Guido Van Rossum, https://gvanrossum.github.io/)이 발표한 프로그래밍 언 어





Guido van Rossum - Personal Home Page

"Gawky and proud of it."

Who I Am

Read my "King's Day Speech" for some inspiration.

I am the author of the Python programming language. See also my resume and my publications list, a brief bio, assorted writings, presentations and interviews (all about Python), some pictures of me, my new blog, and my old blog on Artima.com. I am @gvanrossum on Twitter.

I am currently a Distinguished Engineer at Microsoft. I have worked for Dropbox, Google, Elemental Security, Zope Corporation, BeOpen.com, CNRI, CWI, and SARA. (See my resume.) I created Python while at CWI.

How to Reach Me

You can send email for me to guido (at) python.org. I read everything sent there, but I receive too much email to respond to everything.

My Name

My name often poses difficulties for Americans.

Pronunciation: in Dutch, the "G" in Guido is a hard G, pronounced roughly like the "ch" in Scottish "loch". (Listen to the sound clip.) However, if you're American, you may also pronounce it as the Italian "Guido". I'm not too worried about the associations with mob assassins that some people have. :-)

☆ ★ □ □ □ :



- ❖ 파이썬이라는 이름은 로섬이 좋아하는 TV 프로그램인 Monty python 코미디언 그룹의 Flying circus 코메디 프로(1969~1974년 영국 BBC에서 방송)에서 따왔습니다.
- ❖ Silly Olympics (바보 올림픽): Flying circus 코메디 프로중하, https://www.youtube.com/watch?v=qgSzGIkFq2A
- ❖ 파이썬의 사전적 의미가 그리스 신화에 나오는 큰 비단뱀을 뜻해서 파이썬 책이나 로고에 뱀 모양이 많이 나옵니다.

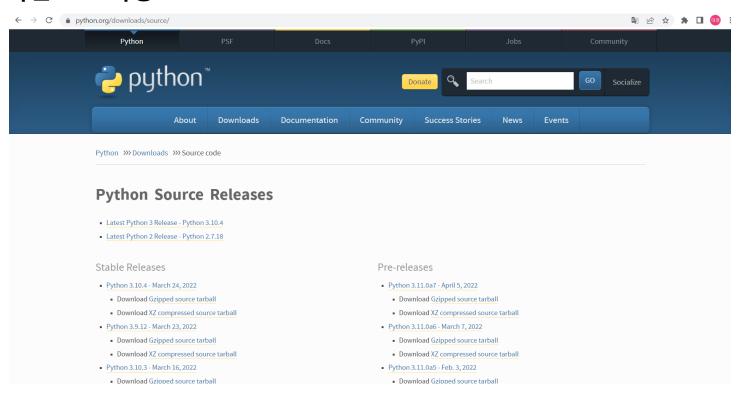


- ❖ 상위 레벨 언어
 - : 사람이 이해하기 쉬운 언어 <-> 하위 레벨 언어 (기계어(명령어가 α 과 α 로 되어 있는 언어))
- ❖ 인터프리터 언어
 - : 상위 레벨 언어의 코드를 기계어(o과 1)로 컴파일 하는 것이 아니라 인터 프리터가 코드를 해석해서 실행하는 언어 <-> 컴파일 언어
- ❖ 객체 지향 언어
 - : 함수 단위 뿐 아니라 객체 단위로도 프로그래밍 가능한 언어 <-> 절차 지향 언어 (함수 단위로만 프로그래밍 가능)
- ❖ 오픈 소스
 - : 소스 코드가 공개 되어 있
 - 음, https://www.python.org/downloads/source/ 에서 소스 코드 다운로드 가능

데이터 분석도구 개관



- ❖ 특징
- ❖ : 소스 코드가 공개 되어 있
 - 음, https://www.python.org/downloads/source/ 에서 소스 코드 다운로드 가능

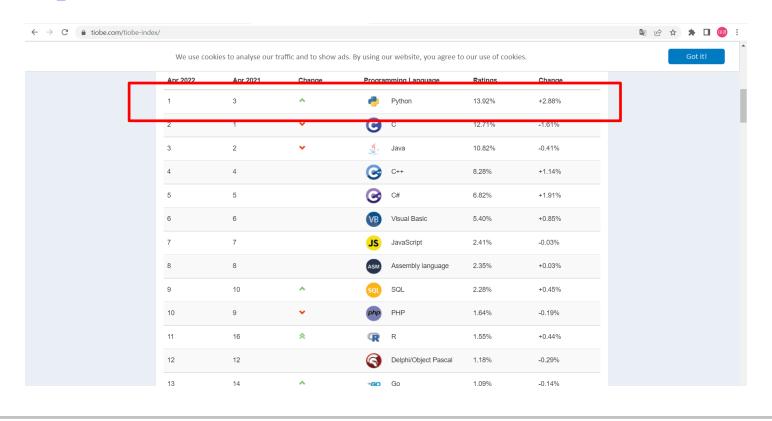


- ❖ 역사
- ❖ 파스칼 (교육용 언어, 1969년)
- ❖ ABC (교육용 언어, 1980년대, CWI(네덜란드 국립 수학 및 컴퓨터 과학 연구기관, 귀도 반 로섬))
- ❖ 파이썬 (교육용 언어, 1989년 연말 연휴, 귀도 반 로섬)

데이터 분석도구 개관



- ❖ 파이썬 점유율 순위
- ❖ TIOBE에서 선정한 프로그래밍 언어 월간 점유율 순위 (Apr 2022 기준) 1위 차지
- https://www.tiobe.com/tiobe-index/





- ❖ 대형 웹 사이트 회사들이 파이썬 (장고 패키지) 사용 웹 사이트 구축
- ❖ 구글, 유튜브, 인스타그램, 피인터리스트, 레딧, 드롭박스

Google Colab 이란?

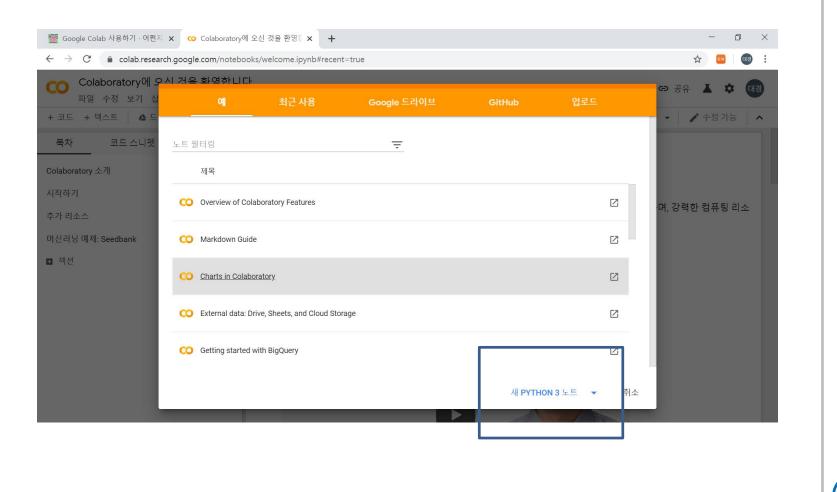
- ❖ Google Colab은 AI개발자들을 위해 구글에서 제공하는 무료 클라우드 서비스
- ❖ 풀 네임은 Google Colaboratory
- **❖** Google Drive + Jupyter Notebook
 - · Google Drive처럼 협업 가능(동시에 수정 가능)
- ❖ https://colab.research.google.com/로 접속시 사용 가능
- ❖ 컴퓨터 사양(19년 12월 기준)
 - Ubuntu 17.10
 - · CPU 제논 2.3GHz
 - 메모리 13G
 - GPU: K8o 또는 T4:
 - TPU도 사용 가능
- ❖ GPU 사용시 최대 12시간
- ❖ Github의 소스 코드를 Colab에서 사용 가능





Google Colab 이란?

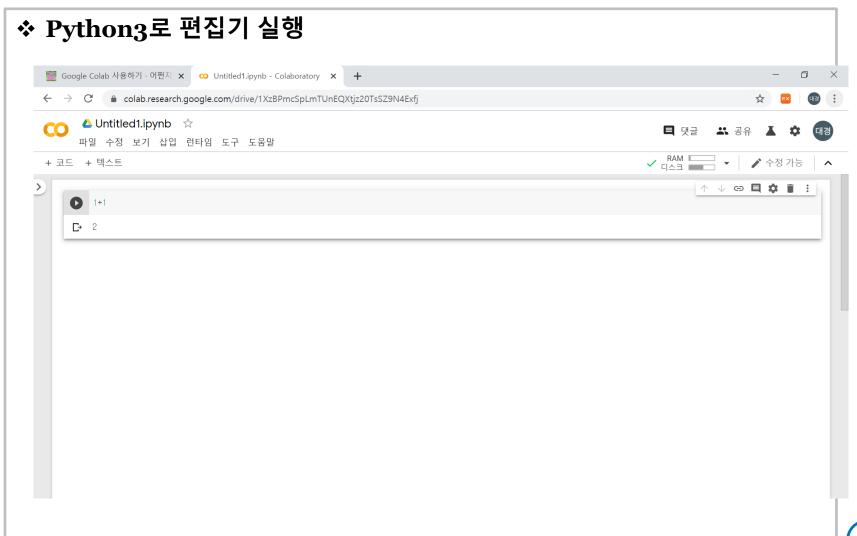
❖ https://colab.research.google.com/로 접속시 사용 가능





Google Colab

Google Colab 이란?



Google Colab



UI

- ❖ 붉은 부분
 - ❖ 해당 노트북의 목차
 - ❖ 코드 스니펫
 - ❖ 재사용 가능한 소스 코드로 다양한 예제 코드가 있음
 - Uploading files from your local file system, Using BigQuery, Listing files in Google Drive, Install library, etc
 - ❖ 연결된 파일
- ❖ 노란 부분
 - ❖ 헤더 보이기 유무
- ❖ 파란 부분
 - ❖ 코드 부분



Unit 2

Google Colab

Colab 리눅스 명령어 사용

❖ 명령문 앞에 !을 붙인다.

OS

!cat /etc/issue.net

Ubuntu 18.04.2 LTS

CPU 사양

!head /proc/cpuinfo

processor : 0

vendor_id : GenuineIntel

cpu family : 6

model :85

model name : Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz

stepping : 4

microcode : 0x1000157

cpu MHz : 2999.998

cache size : 25344 KB

physical id : o

메모리 사양

!head -n 3 /proc/meminfo

MemTotal: 7814692 kB MemFree: 533060 kB MemAvailable: 6261620 kB

Unit 2

Google Colab

Colab 리눅스 명령어 사용

```
!df-h
          Size Used Avail Use% Mounted on
Filesystem
        4.8G 60M 4.5G 2%/
none
         64M o 64M o%/dev
tmpfs
tmpfs
         3.8G o 3.8G o%/sys/fs/cgroup
shm
         64M o 64M o%/dev/shm
/dev/nvmeon1p1 117G 69G 49G 59% /goorm
/dev/nbd1p1 4.8G 60M 4.5G 2% /workspace
        3.8G o 3.8G o%/proc/acpi
tmpfs
         3.8G o 3.8G o%/proc/scsi
tmpfs
tmpfs
         3.8G o 3.8G o%/sys/firmware
!python -version
Python 3.7.4
import tensorflow as tf
print(tf. version )
2.8.2
!git clone https://github.com/looker2zip/bigStudy
Cloning into 'bigStudy'...
remote: Enumerating objects: 118, done.
```

Receiving objects: 100% (118/118), 25.23 MiB | 30.32 MiB/s, done.

Resolving deltas: 100% (25/25), done.



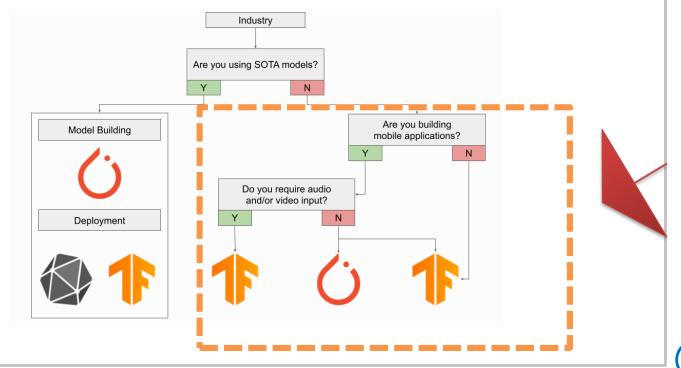
Subsection 2

Unit 1

Machine Learning/AI Issue

Should I Use PyTorch or TensorFlow?

- ❖ 내가 산업체에 있다면?
- 산업 환경에서 딥 러닝 엔지니어링을 수행하는 경우 TensorFlow를 사용하고 있을 가능성이 높으며 계속 사용해야 합니다. TensorFlow의 강력한 배포 프레임워크와 종단 간 TensorFlow Extended 플랫폼은 모델을 생산해야 하는 사람들에게 매우 중요합니다. 모델 모니터링 및 아티팩트 추적과 함께 gRPC 서버에 쉽게 배포하는 것은 업계에서 사용하기 위한 중요한 도구입니다.

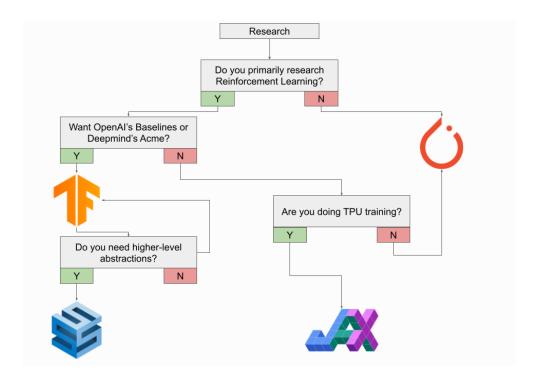




Machine Learning/AI Issue

Should I Use PyTorch or TensorFlow?

- ❖ 내가 연구원이라면?
- 연구원이라면 거의 확실하게 PyTorch를 사용하고 있으며 현재로서는 계속 사용하고 있을 것입니다.



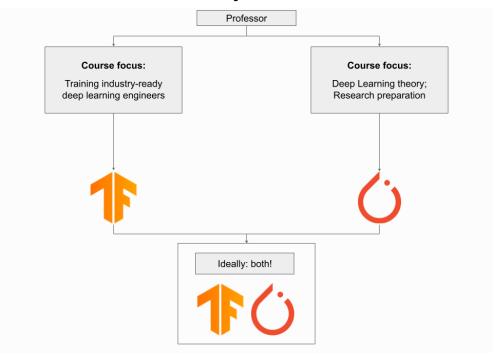


Machine Learning/AI Issue

Should I Use PyTorch or TensorFlow?

❖ 내가 교수라면?

- 교육 과정의 초점이 딥 러닝 이론뿐만 아니라 전체 딥 러닝 프로세스에서 역량을 발 휘하여 기초를 다질 수 있는 업계 준비 딥 러닝 엔지니어를 양성하는 것이라면 TensorFlow를 사용해야 합니다.
- 딥 러닝 연구를 수행할 수 있도록 학생들을 준비시키기 위한 고급 학부 과정이나 초기 대학원 수준 과정을 가르치는 경우 PyTorch를 사용해야 합니다.

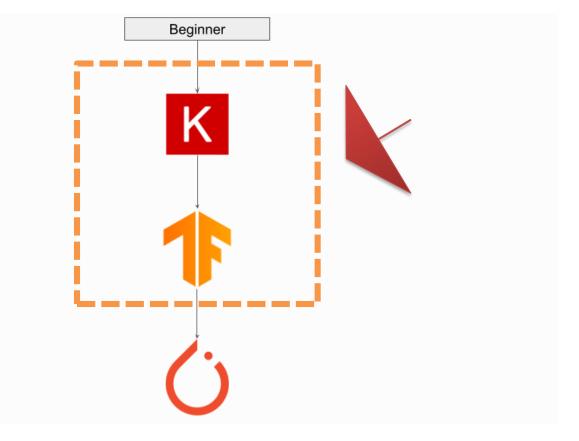




Machine Learning/AI Issue

Should I Use PyTorch or TensorFlow?

- ❖ 내가 완전 초보자라면?
- 딥 러닝에 관심이 있고 막 시작하려는 완전 초보자라면 Keras 를 사용하는 것이 좋습니다 .





윈10에서 Anaconda Prompt를 이용해 가상환경 만들기

❖ Tensorflow2를 기준으로 설치

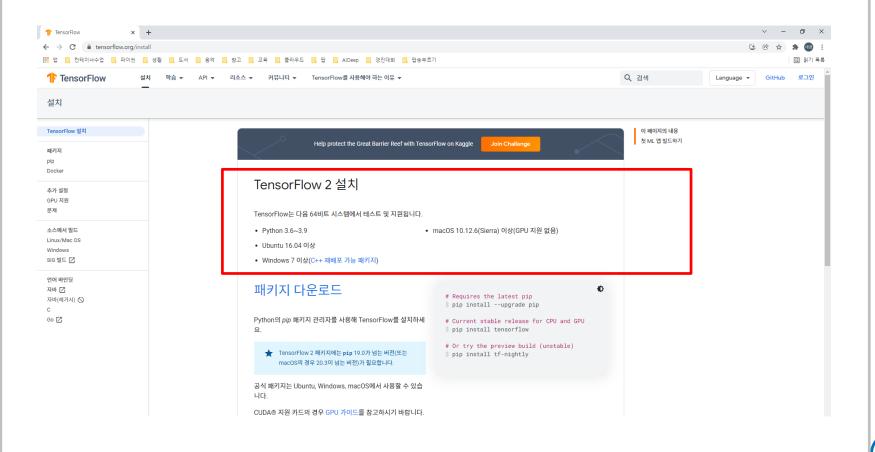
- 1. 아나콘다 설치
- 2. 텐서플로2
- 3. JupyterNotebook 설치



- ❖ 가상환경(Virtual Environment)을 이용하면 Python 버전 간의 의존성을 고려해서 가상의 격리된 환경을 만들어줌으로써 버전이 다름으로 인해 발생할 수 있는 호환이나 충돌 문제를 미연에 방지할 수 있음.
- ❖ 파이썬 배포판 종류
 - Intel사의 인텔 최적화 배포판
 - https://www.intel.com/content/www/us/en/developer/tools/oneapi/distribution-for-python.html
 - Anaconda Python 배포판에서는 conda command 로 가상환경, Python 패키지 설치를 관리할 수 있음.
- ❖ Anaconda 설치

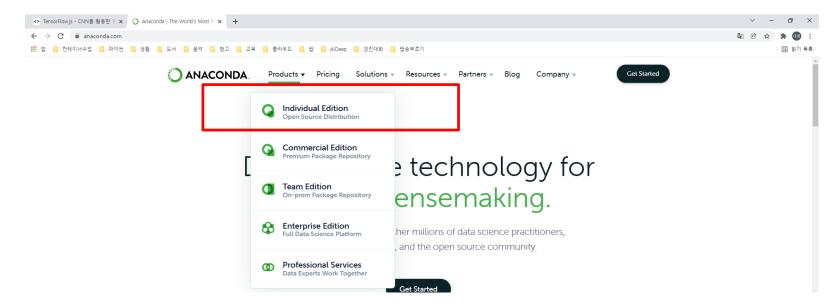


- https://www.tensorflow.org/install
- ❖ Python 3.5 부터 3.8을 지원



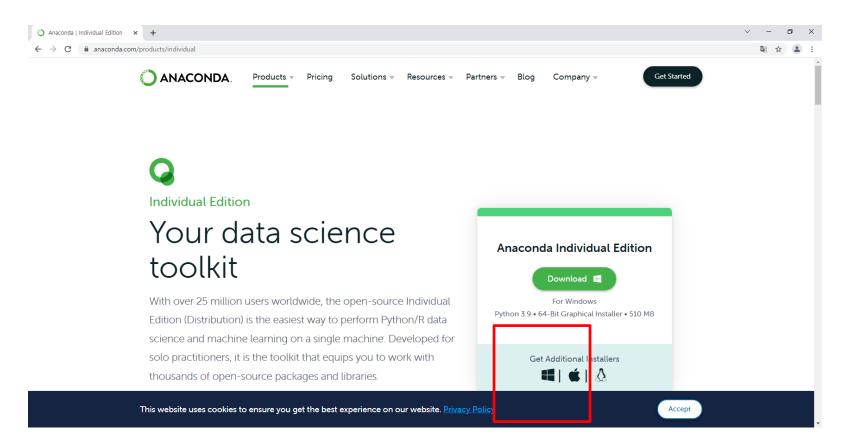


- https://www.anaconda.com/
- ❖ Individual Edition 선택



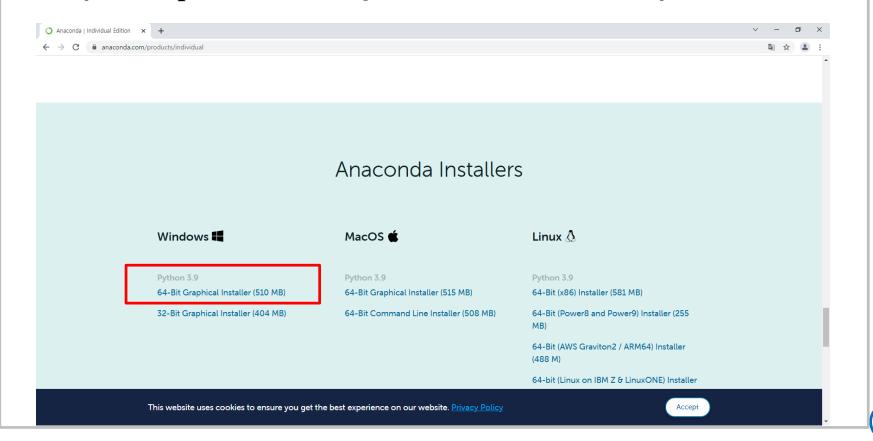


- https://www.anaconda.com/
- ❖ Get Additional Installers > 윈도 선택





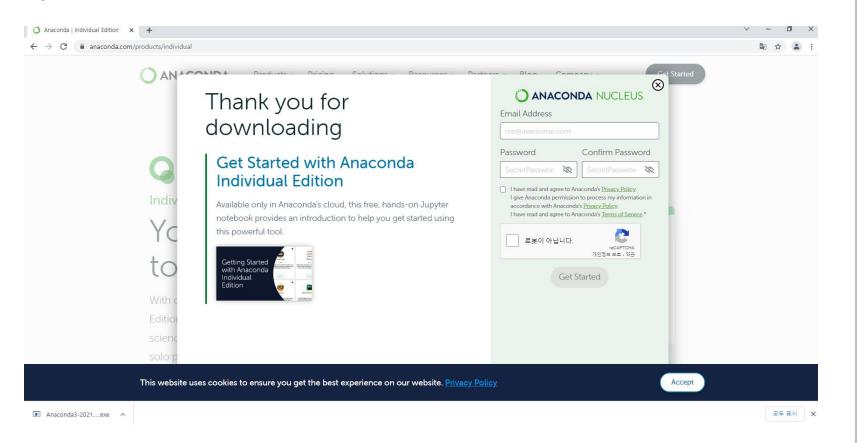
- https://www.anaconda.com/
- * Python 3.9
 - 64-Bit Graphical Installer (510 MB) 클릭-2021년 12월 04일 기준





Anaconda 설치

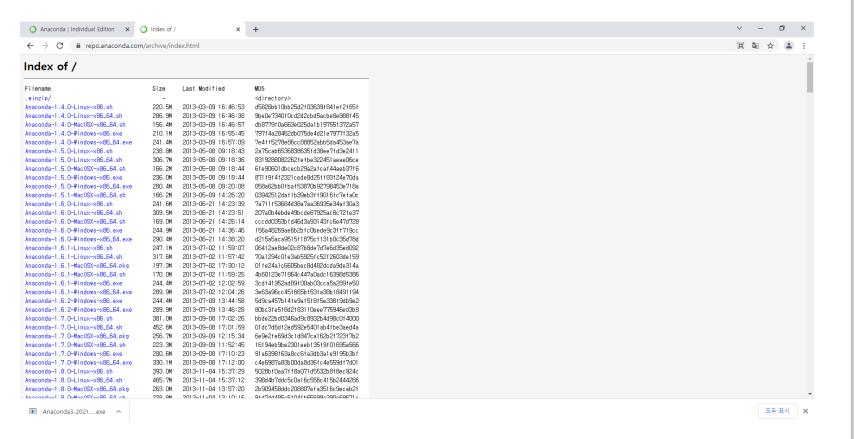
❖ 다운로드 중





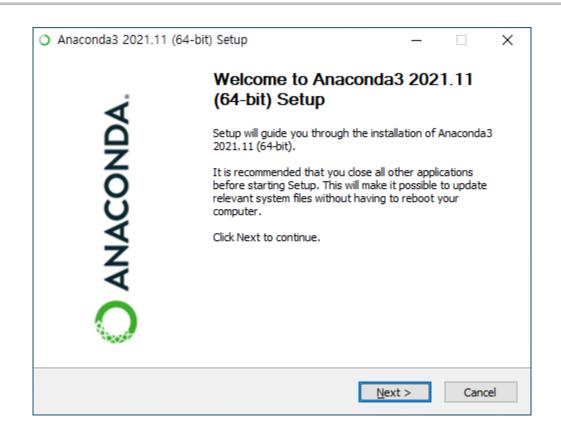
Anaconda 설치

❖ Old 버전 다운받을 시 https://repo.anaconda.com/archive/index.html 이용





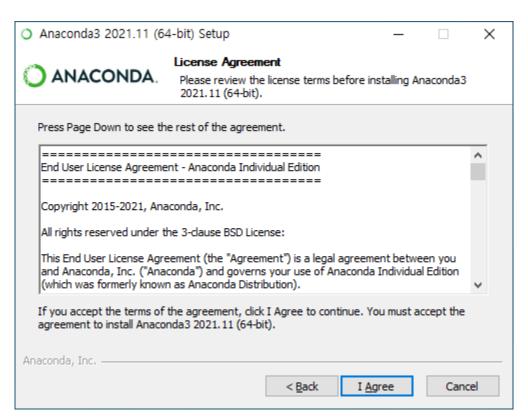






Anaconda 설치

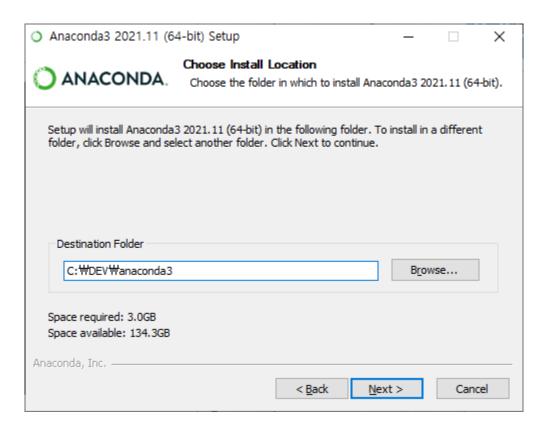
* I Agree 클릭





Anaconda 설치

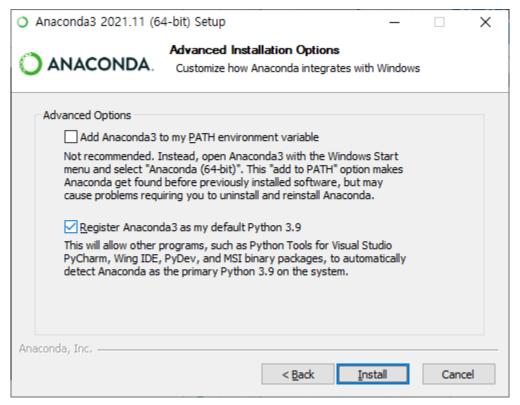
* C:\DEV\anaconda3로 경로 설정하고 Next





Anaconda 설치

* Register Anaconda3 as my default Python 3.8 ' 만 체크하고 Install 클 릭





Anaconda 설치

* Add Anaconda3 to my PATH environment variable:

- PATH 환경 변수에 아나콘다를 추가할지 선택
- <u>아나콘다 외에 다른 파이썬 인터프리터를 환경변수에 등록해서 사용한다면 체크 해</u> 제
- 아나콘다만을 사용하는 경우, 아나콘다가 주력일 경우로 윈도우 cmd 창에서 파이썬을 실행할 경우, 현재 PC에 파이썬을 설치한 적이 없는 경우에는 체크
- (체크할 경우 윈도우 cmd 창 경로와 상관없이 아나콘다를 파이썬으로 인식)

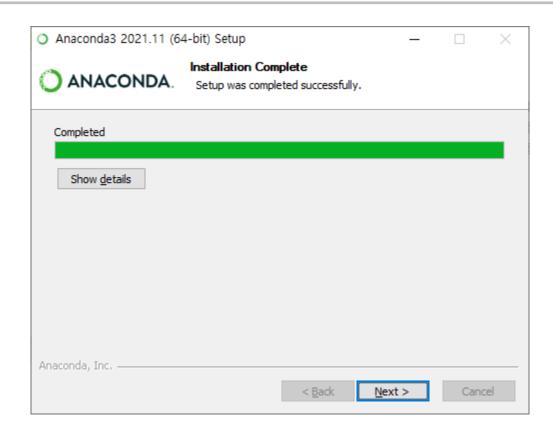
* Register Anaconda3 as my default Python 3.8:

- 아나콘다를 기본 파이썬으로 등록할지 여부를 선택
- (체크할 경우 개발 도구나 에디터에서 아나콘다를 파이썬으로 인식)



Anaconda 설치

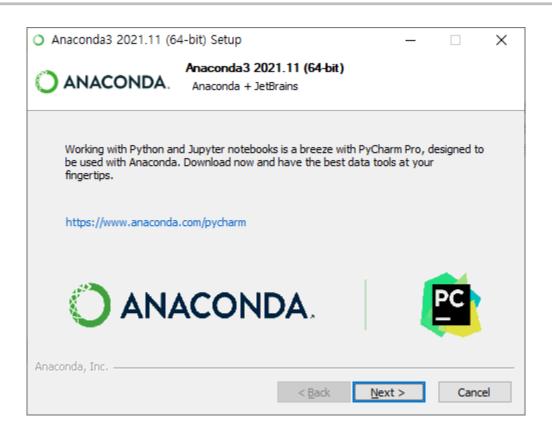
* Next 클릭





Anaconda 설치

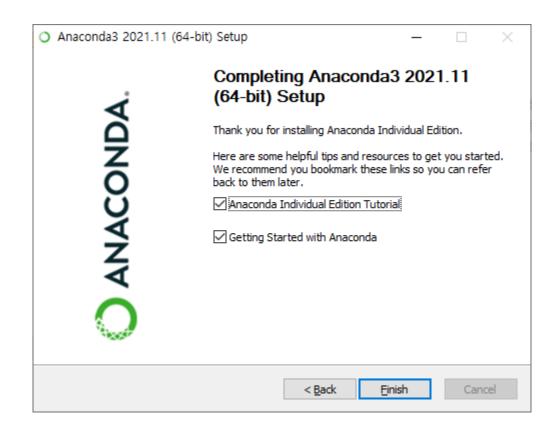
* Next 클릭





Anaconda 설치

* Finish 클릭(설치 완료)

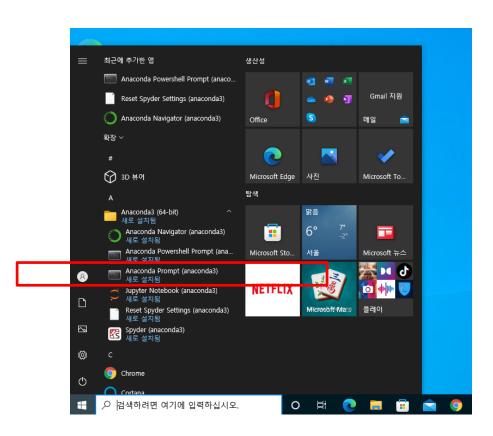






Anaconda 설치

❖ 시작 -> Anaconda3 (64-bit) -> Anaconda Prompt (anaconda3) 실행 해서 간단하게 python 테스트해보기





Anaconda 설치 완료

❖ 시작 -> Anaconda3 (64-bit) -> Anaconda Prompt (anaconda3) 실행 해서 간단하게 python 테스트해보기

```
■ Anaconda Prompt (anaconda3)
(base) C:\Users\Uk8s>python -V
Python 3.9.7
(base) C:\Users\Uk8s>_
```



TensorFlow 2 설치

• pip을 업그레이드

(base) C:\Windows\system32>pip install --upgrade pip

신경망과 딥러닝 환경 구축

Windows10에서 Anaconda Prompt를 이용해 가상환경 만들기

❖ Environment py36tf112 설치

신경망과 딥러닝 환경 구축

Windows10에서 Anaconda Prompt를 이용해 가상환경 만들기

❖ Environment py36tf112 설치

(py36tf112) C:₩Users₩DaeKyeong>python

Python 3.6.10 |Anaconda, Inc.| (default, Jan 7 2020, 15:18:16) [MSC v.1916 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> import tensorflow as tf

C:₩Users₩DaeKyeong₩Anaconda3₩envs₩py36tf112₩lib₩site-

packages\tensorflow\python\framework\dtypes.py:523: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

(py36tf112) C:₩Users₩DaeKyeong>pip uninstall numpy

Found existing installation: numpy 1.18.1

Uninstalling numpy-1.18.1:

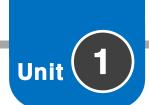
Would remove:

c:\u00a8users\u00a8daekyeong\u00a8anaconda3\u00a8envs\u00a8py36tf112\u00a8lib\u00a8site-packages\u00a8numpy-1.18.1.dist-info\u00a8* c:\u00a8users\u00a8daekyeong\u00a8anaconda3\u00a8envs\u00a8py36tf112\u00a8lib\u00a8site-packages\u00a8numpy\u00a8*

c:\u00e4users\u00f8daekyeong\u00a8anaconda3\u00a8envs\u00a8py36tf112\u00a8scripts\u00a8f2py.exe

Proceed (y/n)? y

Successfully uninstalled numpy-1.18.1



신경망과 딥러닝 환경 구축

Windows10에서 Anaconda Prompt를 이용해 가상환경 만들기

❖ Environment py36tf112 설치

```
(py36tf112) C:₩Users₩DaeKyeong>pip install numpy==1.14.0
Collecting numpy==1.14.0
Downloading numpy-1.14.0-cp36-none-win_amd64.whl (13.4 MB)
```

13.4 MB 1.7 MB/s

Installing collected packages: numpy Successfully installed numpy-1.14.0

(py36tf112) C:₩Users₩DaeKyeong>python

Python 3.6.10 |Anaconda, Inc.| (default, Jan 7 2020, 15:18:16) [MSC v.1916 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> import tensorflow as tf

>>>



신경망과 딥러닝 환경 구축

Windows10에서 Anaconda Prompt를 이용해 가상환경 만들기

❖ Environment tf37_cpu 설치

(base) C:₩Users₩DaeKyeong>conda create --name tf37_cpu python=3.7

(base) C:₩Users₩DaeKyeong>conda activate tf37_cpu

(tf37_cpu) C:₩Users₩DaeKyeong>pip install --upgrade tensorflow

Collecting tensorflow

Downloading tensorflow-2.9.1-cp37-cp37m-win_amd64.whl (444.0 MB)

444.0 MB 131 kB/s

(tf37_cpu) C:₩Users₩DaeKyeong>python

Python 3.7.13 (default, Mar 28 2022, 08:03:21) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32 Type "help", "copyright", "credits" or "license" for more information.

>>> import tensorflow as tf

>>> quit()

(tf37 cpu) C:\Users\DaeKyeong>

신경망과 딥러닝 환경 구축

Jupyter Notebook

(tf37_cpu) C:₩Users₩DaeKyeong>pip install ipykernel

Collecting ipykernel

Using cached ipykernel-6.13.1-py3-none-any.whl (133 kB)

Collecting nest-asyncio

Using cached nest_asyncio-1.5.5-py3-none-any.whl (5.2 kB)

•••

(tf37_cpu) C:\Users\DaeKyeong>python -m ipykernel install --user --name tf37_cpu --display-name "tf37_cpu"

Installed kernelspec tf37_cpu in C:\Users\apro621\AppData\Roaming\jupyter\kernels\tf37_cpu

(tf37_cpu) C:₩Users₩DaeKyeong>

신경망과 딥러닝 환경 구축

Jupyter Notebook

(tf37_cpu) C:₩Users₩DaeKyeong>pip install ipykernel

Collecting ipykernel

Using cached ipykernel-6.13.1-py3-none-any.whl (133 kB)

Collecting nest-asyncio

Using cached nest_asyncio-1.5.5-py3-none-any.whl (5.2 kB)

•••

(tf37_cpu) C:\Users\DaeKyeong>python -m ipykernel install --user --name tf37_cpu --display-name "tf37_cpu"

Installed kernelspec tf37_cpu in C:\Users\apro621\AppData\Roaming\jupyter\kernels\tf37_cpu

(tf37_cpu) C:₩Users₩DaeKyeong>



TensorFlow 2 설치

• 가상환경 생성

(base) C:\Windows\system32>conda create -n tf38 cpu pip python=3.8

Collecting package metadata (current_repodata.json): done
Solving environment: done
Package Plan

 $environment\ location: C: \ DEV \ anaconda 3 \ envs \ tensor flow$

 $added \ / \ updated \ specs:$

- pip

- python=3.8

The following packages will be downloaded:

package	build	
certifi-2021.10.8 pip-21.2.2	100 3000 = 3	152 KB MB
python-3.8.12	1	о МВ
setuptools-58.0.4	py38haa95532_0	779 KB
wincertstore-0.2	py38haa95532_2	15 KB

Total: 18.8 MB

(base) C:\Windows\system32>

윈10에 DataAnalysis 환경 구성

TensorFlow 2 설치

```
The following NEW packages will be INSTALLED:
 ca-certificates pkgs/main/win-64::ca-certificates-2021.10.26-haa95532_2
            pkgs/main/win-64::certifi-2021.10.8-py38haa95532_0
 certifi
 openssl
              pkgs/main/win-64::openssl-1.1.1l-h2bbff1b_o
 pip
           pkgs/main/win-64::pip-21.2.2-py38haa95532_0
             pkgs/main/win-64::python-3.8.12-h6244533_0
 python
 setuptools
               pkgs/main/win-64::setuptools-58.0.4-py38haa95532 0
 sqlite
            pkgs/main/win-64::sqlite-3.36.0-h2bbff1b o
           pkgs/main/win-64::vc-14.2-h21ff451 1
 \mathbf{vc}
 vs2015 runtime pkgs/main/win-64::vs2015 runtime-14.27.29016-h5e58377 2
 wheel
             pkgs/main/noarch::wheel-0.37.0-pyhd3eb1b0 1
                pkgs/main/win-64::wincertstore-0.2-py38haa95532 2
 wincertstore
Proceed ([y]/n)?
done
# To activate this environment, use
#
   $ conda activate tensorflow
#
# To deactivate an active environment, use
   $ conda deactivate
```



TensorFlow 2 설치

• 가상환경으로 이동

(base) C:\Windows\system32>conda activate tf38_cpu

(tf38_cpu) C:\Windows\system32>

윈10에 DataAnalysis 환경 구성

TensorFlow 2 설치

- 텐서플로우 설치
- CPU 버전인 경우

(tf38_cpu) C:\Windows\system32>pip install --ignore-installed --upgrade tensorflow-cpu

Collecting tensorflow-cpu

Downloading tensorflow_cpu-2.7.0-cp38-cp38-win_amd64.whl (237.1 MB)

| 237.1 MB 10 kB/s

Collecting tensorboard~=2.6

•••

Created wheel for termcolor: filename=termcolor-1.1.o-py3-none-any.whl size=4847 sha256=994991cc97dab572143862539b91478c492ce703f1778ccee57d7bbfe8ac37b7 Stored in directory: c:\users\k8s\appdata\local\pip\cache\wheels\ao\16\9c\5473df82468f958445479c59e784896fa24f4a5fc024bof501 Successfully built termcolor

Installing collected packages: urllib3, pyasn1, idna, charset-normalizer, certifi, zipp, six, setuptools, rsa, requests, pyasn1-modules, oauthlib, cachetools, requests-oauthlib, importlib-metadata, google-auth, wheel, werkzeug, tensorboard-plugin-wit, tensorboard-data-server, protobuf, numpy, markdown, grpcio, google-auth-oauthlib, absl-py, wrapt, typing-extensions, termcolor, tensorflow-io-gcs-filesystem, tensorflow-estimator, tensorboard, opt-einsum, libclang, keras-preprocessing, keras, h5py, google-pasta, gast, flatbuffers, astunparse, tensorflow-cpu

Successfully installed absl-py-1.0.0 astunparse-1.6.3 cachetools-4.2.4 certifi-2021.10.8 charset-normalizer-2.0.9 flatbuffers-2.0 gast-0.4.0 google-auth-2.3.3 google-auth-oauthlib-0.4.6 google-pasta-0.2.0 grpcio-1.42.0 h5py-3.6.0 idna-3.3 importlib-metadata-4.8.2 keras-2.7.0 keras-preprocessing-1.1.2 libclang-12.0.0 markdown-3.3.6 numpy-1.21.4 oauthlib-3.1.1 opt-einsum-3.3.0 protobuf-3.19.1 pyasn1-0.4.8 pyasn1-modules-0.2.8 requests-2.26.0 requests-oauthlib-1.3.0 rsa-4.8 setuptools-59.4.0 six-1.16.0 tensorboard-2.7.0 tensorboard-data-server-0.6.1 tensorboard-plugin-wit-1.8.0 tensorflow-cpu-2.7.0 tensorflow-estimator-2.7.0 tensorflow-io-gcs-filesystem-0.22.0 termcolor-1.1.0 typing-extensions-4.0.1 urllib3-1.26.7 werkzeug-2.0.2 wheel-0.37.0 wrapt-1.13.3 zipp-3.6.0

(tf38_cpu) C:\Windows\system32>

윈10에 DataAnalysis 환경 구성

TensorFlow 2 설치

• 텐서플로우 설치 테스트

(tf38_cpu) C:\Windows\system32>python

Python 3.8.12 (default, Oct 12 2021, 03:01:40) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32 Type "help", "copyright", "credits" or "license" for more information.

- >>> import tensorflow as tf
- >>> with tf.compat.v1.Session() as sess:
- ... helloworld = tf.constant("Hello World!")
- ... print(sess.run(helloworld))

•••

2021-12-04 18:35:34.557406: 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.

b'Hello World!'

- >>> helloworld = tf.constant("Hello World!")
- >>> tf.print(helloworld)

Hello World!

>>> quit()

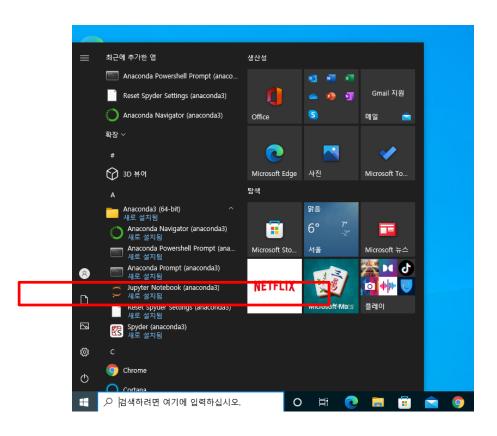
(tf38_cpu) C:\Windows\system32>



윈10에 DataAnalysis 환경 구성

Jupyter Notebook

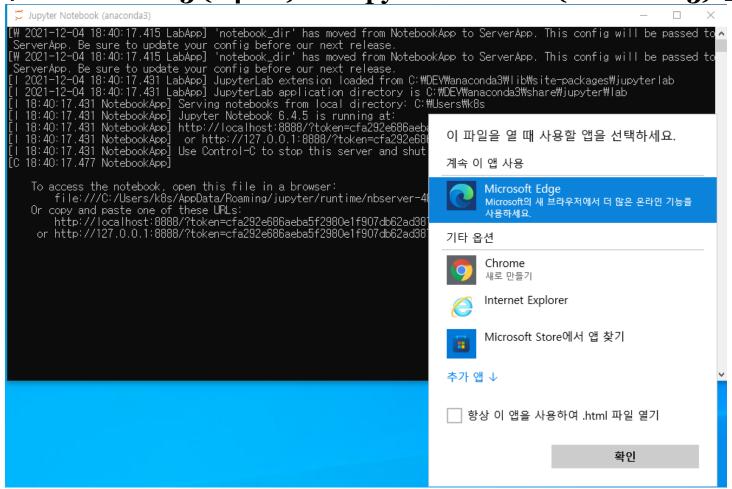
❖ * 시작 -> Anaconda3 (64-bit) -> Jupyter Notebook(anaconda3) 실행





Jupyter Notebook

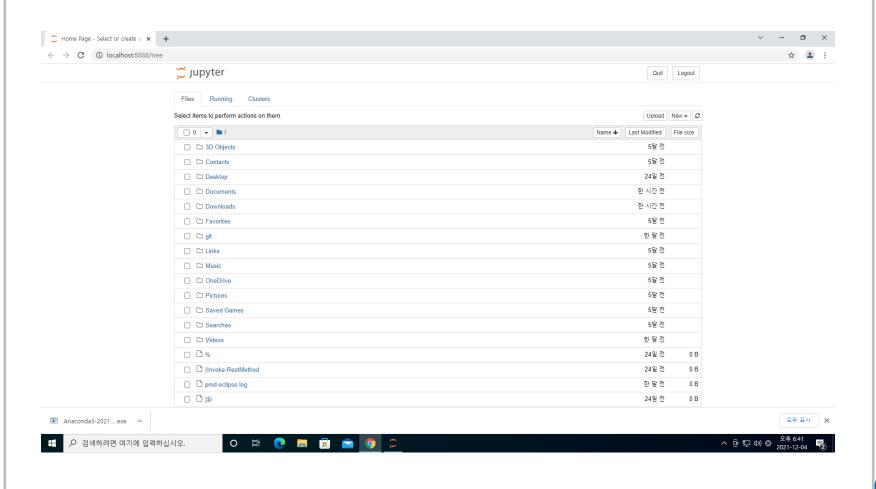
❖ 시작 -> Anaconda3 (64-bit) -> Jupyter Notebook(anaconda3) 실행





Jupyter Notebook

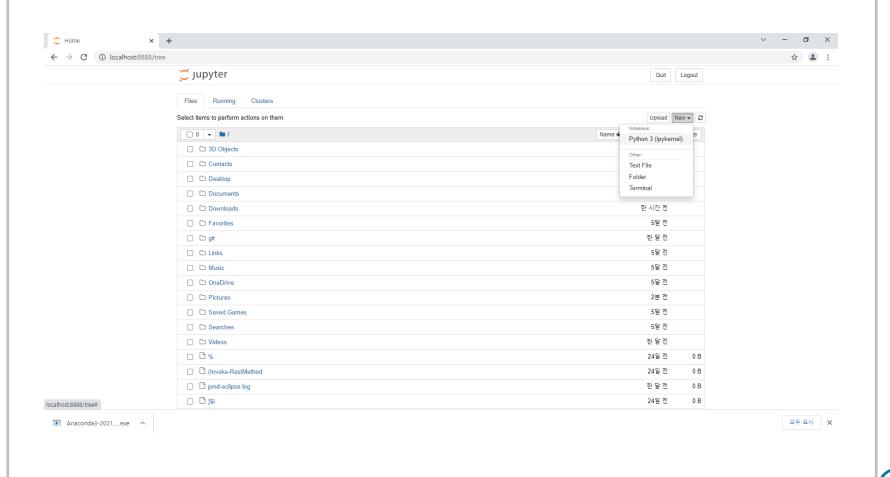
❖ 시작 -> Anaconda3 (64-bit) -> Jupyter Notebook(anaconda3) 실행





Jupyter Notebook

❖ 메인화면에서 오른쪽 상단부분에 New 를 눌러 Python3 로 새파일 생성





Jupyter Notebook

그런데!!!

<u>주피터 노트북에는 파이썬</u> 커널이 [default]만 보임

윈10에 DataAnalysis 환경 구성

Jupyter Notebook

❖ Jupyter notebook에 가상환경 커널 추가하기

(base) C:\Users\k8s>conda activate tf38_cpu

(tf38_cpu) C:\Users\k8s>pip install ipykernel

Collecting ipykernel

Downloading ipykernel-6.6.o-py3-none-any.whl (126 kB)

126 kB 6.4 MB/s

Collecting tornado<7.0,>=4.2

Downloading tornado-6.1-cp38-cp38-win_amd64.whl (422 kB)

•••

Successfully installed backcall-0.2.0 colorama-0.4.4 debugpy-1.5.1 decorator-5.1.0 entrypoints-0.3 ipykernel-6.6.0 ipython-7.30.1 jedi-0.18.1 jupyter-client-7.1.0 jupyter-core-4.9.1 matplotlib-inline-0.1.3 nest-asyncio-1.5.4 parso-0.8.3 pickleshare-0.7.5 prompt-toolkit-3.0.23 pygments-2.10.0 python-dateutil-2.8.2 pywin32-302 pyzmq-22.3.0 tornado-6.1 traitlets-5.1.1 wcwidth-0.2.5

(tf38_cpu) C:\Users $\k8s$ >



Jupyter Notebook

(tf38_cpu) C:\Users\k8s>python -m ipykernel install --user --name tf38_cpu --display-name "tf38_cpu"

Installed kernelspec tf38_cpu in C:\Users\apro621\AppData\Roaming\jupyter\kernels\tf38_cpu

(tf38_cpu) C:\Users\k8s>jupyter notebook

I 2021-12-04 18:56:05.621 LabApp] JupyterLab extension loaded from C:\DEV\anaconda3\lib\site-packages\jupyterlab

 $[I\ 2021-12-04\ 18:56:05.621\ LabApp]\ Jupyter Lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ jupyter\ lab\ application\ directory\ is\ C:\ DEV\ anaconda3\ share\ sha$

[I 18:56:05.621 NotebookApp] Serving notebooks from local directory: C:\Users\k8s

[I 18:56:05.621 NotebookApp] Jupyter Notebook 6.4.5 is running at:

 $[I\ 18:56:05.621\ Notebook App]\ or\ http://127.0.o.1:8888/? token=f3e22a560ac5177c02a9fdoea19df9fde5d28b3daed8bf05$

[I 18:56:05.621 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 18:56:05.684 NotebookApp]

To access the notebook, open this file in a browser:

file:///C:/Users/k8s/App Data/Roaming/jupyter/runtime/nbserver-7100-open.html

Or copy and paste one of these URLs:

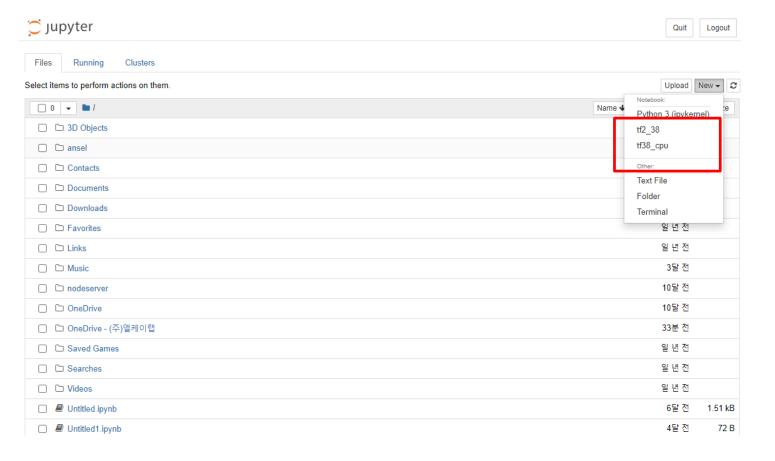
http://localhost: 8888/? token = f3e22a560ac5177c02a9fdoea19df9fde5d28b3daed8bfo5

 $or\ http://127.o.o.1:8888/?token=f3e22a560ac5177c02a9fd0ea19df9fde5d28b3daed8bf05$



Jupyter Notebook

❖ 가상환경 커널 tf38_cpu이 잘 보임





Jupyter Notebook

❖ 가상환경 커널 tf38_cpu에서 텐서플로 코딩

```
In [1]: import tensorflow as tf

In [2]: with tf.compat.v1.Session() as sess:
    helloworld = tf.constant("Hello World!")
    print(sess.run(helloworld))

b'Hello World!'

In [3]: helloworld = tf.constant("Hello World!")
    tf.print(helloworld)

Hello World!
```

❖ 소스 코드

윈10에 DataAnalysis 환경 구성 ipython



❖ 아이파이썬 동작 확인

```
■ 선택 Anaconda Prompt (anaconda3)
                                                                                                                                                                                                               \times
(base) C:\Users\k8s>ipython
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
| Python 7.29.0 -- An enhanced Interactive Python. Type '?' for help.
  n [2]: quit
(base) C:\Users\k8s>_
```

신경망과 딥러닝 환경 구축

Windows10에서 Anaconda Prompt를 이용해 가상환경 만들기

```
import sys
print("python 버전 : {}".format(sys.version))
import pandas as pd
print("pandas 버전 : {}".format(pd. version ))
import matplotlib
print("matplotlib 버전: {}".format(matplotlib. version ))
import numpy as np
print("numpy 버전 : {}".format(np. version ))
import scipy as sp
print("scipy 버전: {}".format(sp. version ))
import IPython
print("IPython 버전: {}".format(IPython. version ))
import sklearn
print("sklearn: {}".format(sklearn. version ))
```



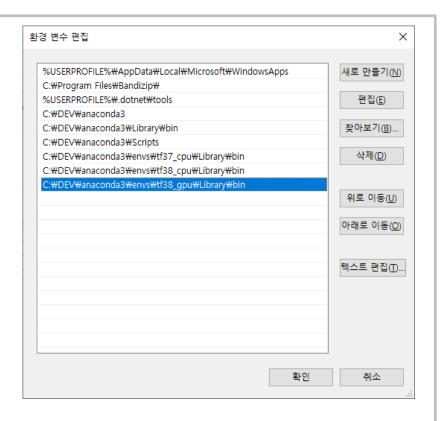
가상환경 삭제

conda remove --name 가상환경이름 --all



'사용자 변수'에 있는 Path에 다음과 같이 설정

anaconda3 anaconda3\Library\bin anaconda3\Scripts anaconda3\envs\tf37_cpu\Library\bin anaconda3\envs\tf38_cpu\Library\bin anaconda3\envs\tf38_gpu\Library\bin





Mobile: 010-9591-1401E-mail: onlooker2zip@naver.com