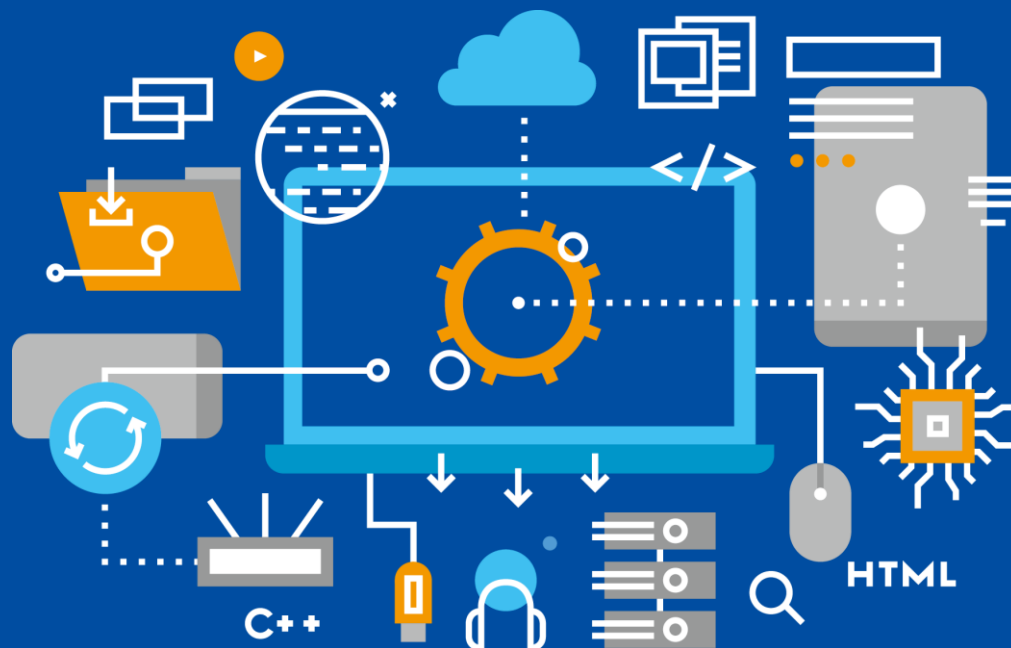


DMU*Ai*

동양미래대학교 인공지능소프트웨어학과

인공지능 서비스
전문 소프트웨어 개발자 인재양성

Dongyang Mirae University
Dept. Of Artificial Intelligence



동양미래대학교 인공지능소프트웨어학과

아나콘다와 주피터 노트북

Dongyang Mirae University
Dept. Of Artificial Intelligence



아나콘다 개요와 설치,
주피터 노트북 개요와 활용

파이썬의 다양한 개발환경

주요 개발환경

- 통합개발환경(IDE)
 - Pycharm
 - Spyder
 - Wing
- 전통 편집기
 - Sublime Text
 - Atom
 - Visual studio code



브라우저 실행환경

- 로컬 실행
 - Jupyter notebook
 - Jupyter lab
- 클라우드 실행
 - Kaggle 노트북
 - Colab
 - <http://colab.research.google.com/>

```
[2] !pip install tensorflow==2.0.0

Collecting tensorflow==2.0.0
  Downloading https://files.pythonhosted.org/packages/45/01/70e53e1188e32739330d15a5e4e996-931bet58a8c30d1d4279892/tensorflow-2.0.0-cv36-cv36-manylinux2010_x86_64.whl (85.9MB)
Requirement already satisfied: keras-applications>=1.0.8 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (1.0.8)
Collecting tensorboard<2.1.0,>=2.0.0
  Downloading https://files.pythonhosted.org/packages/7b/54/99b9e565fca7321098aaa774040b631e80e8940fab0b13475a/tensorboard-2.0.2-cv2-none-any.whl (3.9MB)
Requirement already satisfied: protobuf>=3.6.1 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (3.10.0)
Requirement already satisfied: absl-py>=0.7.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (0.9.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (3.1.0)
Requirement already satisfied: wheel>=0.25 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (0.33.5)
Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (1.15.0)
Requirement already satisfied: google-gast>=0.1.5 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (0.1.8)
Requirement already satisfied: numpy>=2.0,>=1.16.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (1.17.5)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (1.1.0)
Requirement already satisfied: astor>=0.6.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (0.8.1)
Collecting tensorflow-estimator<2.1.0,>=2.0.0
  Downloading https://files.pythonhosted.org/packages/7c/08/8e927332b7019374719145d1d8b21a8b6909d931a616f8b79221ca/tensorflow-estimator-2.0.1-cv2-cv2-none-any.whl (450kB)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (1.1.0)
Requirement already satisfied: gast>=0.2.2 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (0.2.2)
Requirement already satisfied: wrapt>=1.11.1 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (1.11.2)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==2.0.0) (1.12.0)
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (from keras-applications>=1.0.8->tensorflow==2.0.0) (2.8.0)
Requirement already satisfied: reusert>=3,>=2.21.0 in /usr/local/lib/python3.6/dist-packages (from tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0) (2.21.0)
Requirement already satisfied: google-auth>=1.0.5,>=0.4.1 in /usr/local/lib/python3.6/dist-packages (from tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0) (0.4.1)
Collecting google-auth<2.0.1,>=1.6.3
  Downloading https://files.pythonhosted.org/packages/36/19/846771fac3e8a1f0d91c85963648e0385c3d65a20c5034495/google_auth-1.10.0-cv2-cv2-none-any.whl (79kB)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.6/dist-packages (from tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0) (0.16.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.6/dist-packages (from tensorboard<2.1.0,>=2.0.0->tensorflow==2.0.0) (2.6.11)
```

다양한 IDE 비교

• Python IDE 및 코드 편집기 (가이드)

– <https://realpython.com/python-ides-code-editors-guide/>



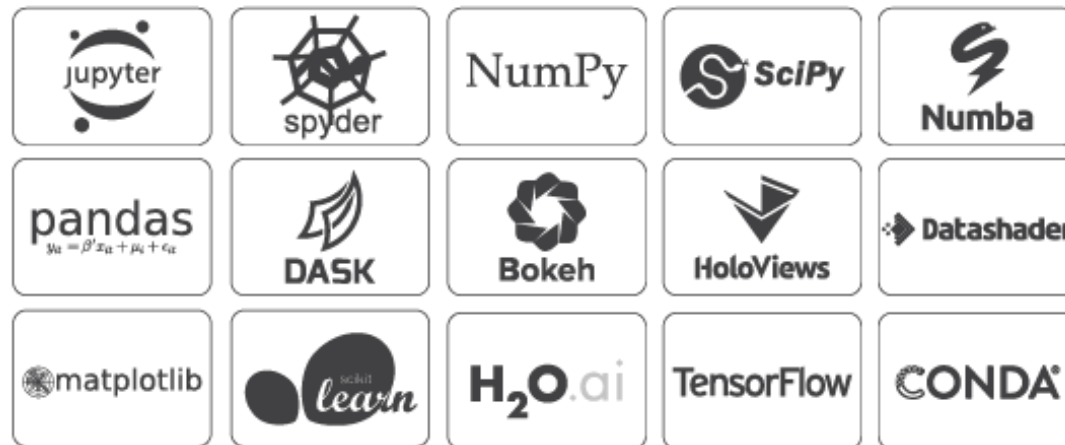
아나콘다 설치

아나콘다 개요 1/2

- **전통 파이썬과 다른 ipython 인터프리터를 사용한 개발 플랫폼**
 - distribution of Python(파이썬 배포판)
 - Python / R 데이터 과학 및 기계 학습을 수행하는 가장 쉬운 방법
 - 데이터 과학자가 수행할 수 있는 단일 머신 1,500 개 이상의 Python / R 데이터 과학 패키지를 빠르게 다운로드
 - 기본적으로 많은 데이터 과학 패키지들이 설치
 - 쉽게 가상환경을 만들고 버전 관리를 할 수 있는 도구 포함
 - conda로 라이브러리, 종속성 및 환경 관리

아나콘다 개요 2/2

- 전통 파이썬과 다른 ipython 인터프리터를 사용한 개발 플랫폼
 - 과학용 모듈이 기본적으로 설치
 - [scikit-learn, TensorFlow 및 Theano]를 사용해 기계학습 및 딥러닝 모델 개발 및 교육
 - dask, numPy, pandas 및 Numba를 사용하여 확장성 및 성능으로 데이터 분석
 - matplotlib, bokeh, datashader 및 holoviews를 사용하여 결과 시각화



아나콘다의 주피터 노트북

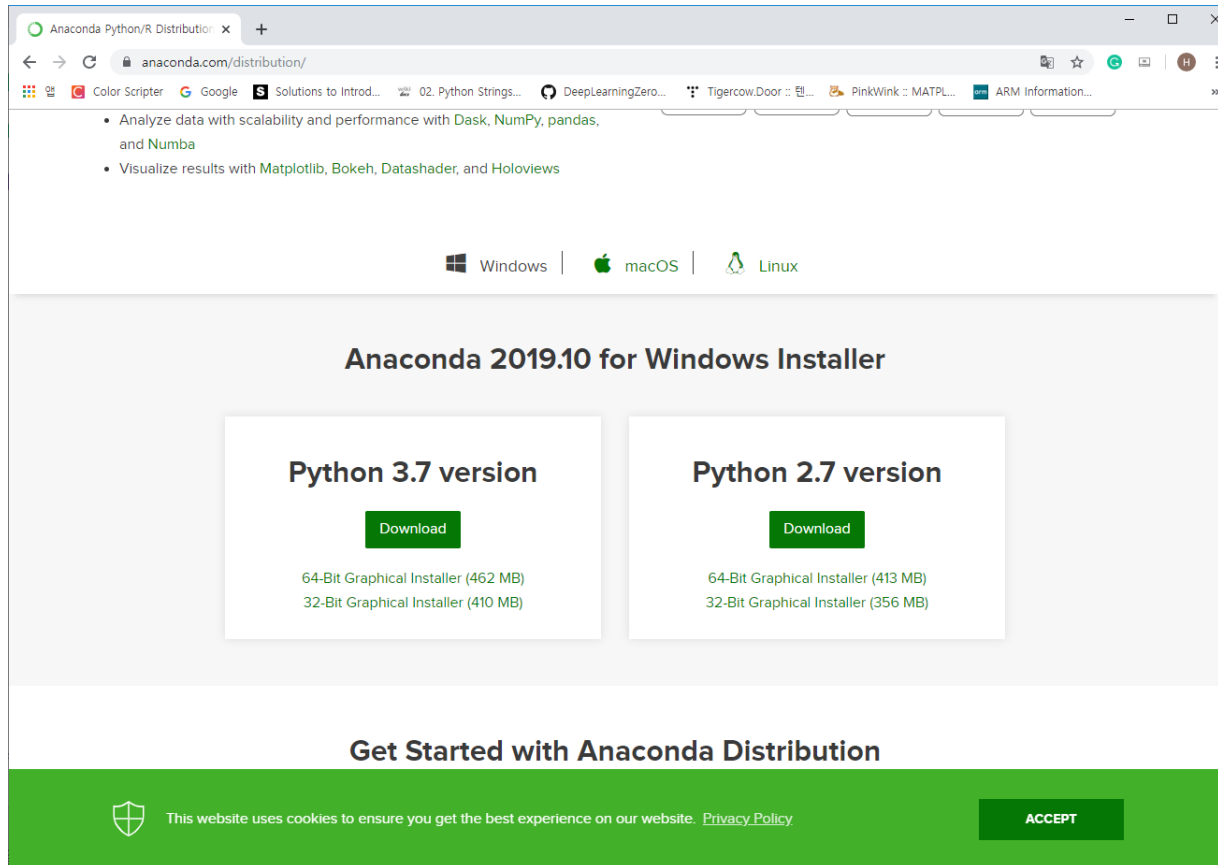
- 초보자에게 아나콘다에서 가장 필요한 것은
 - 주피터 노트북
 - 웹 개발환경
 - Conda
 - 설치 프로그램
- ipython이 jupyter라는 이름으로 변경
- 다양한 개발 환경도 설치
 - jupyter notebook
 - jupyter lab
 - 일반 IDE
 - spyder

설치 전 확인

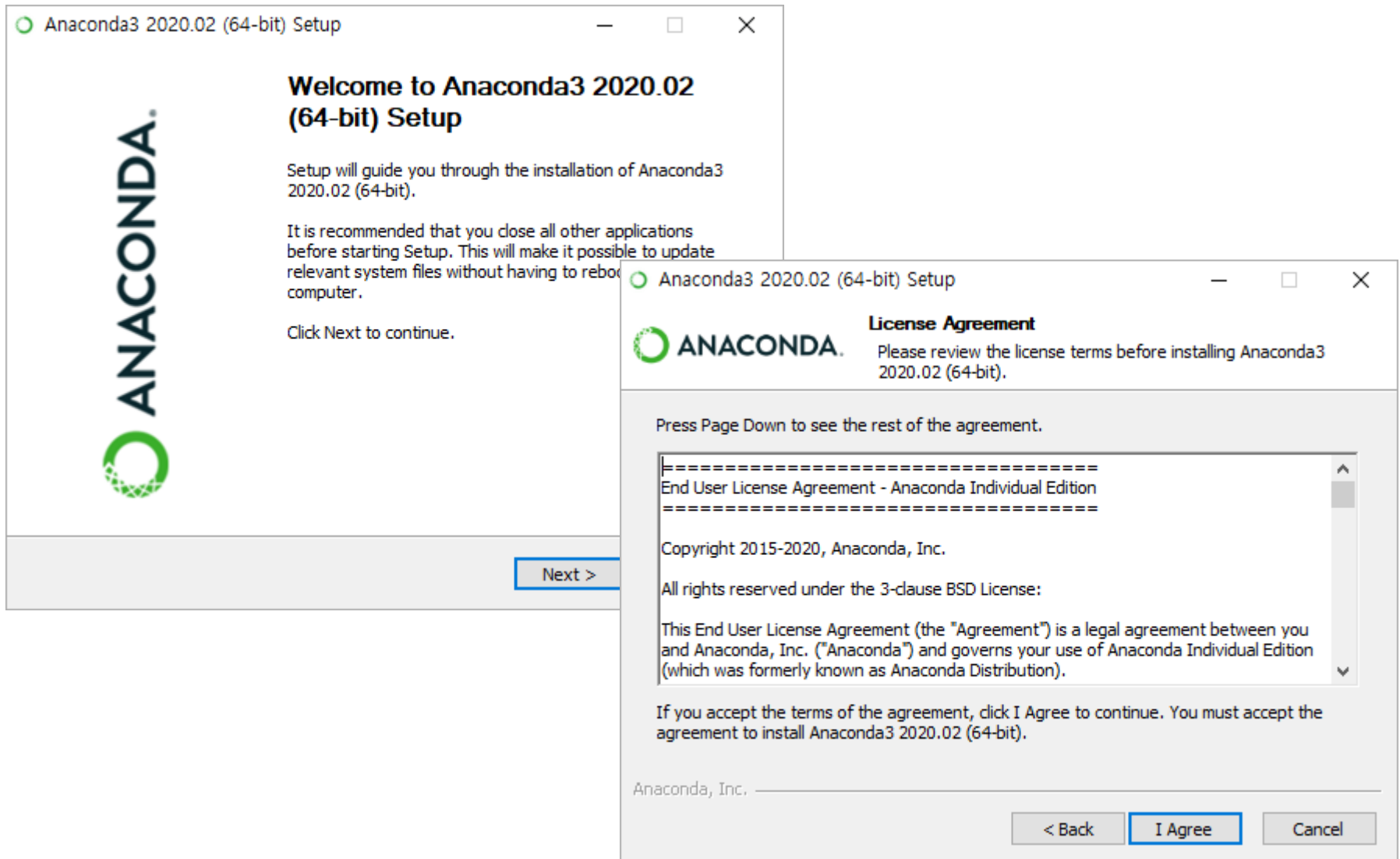
- Path 확인
 - 도스창에서

아나콘다 설치

- 아나콘다 홈페이지에서 아나콘다 설치 파일로 설치



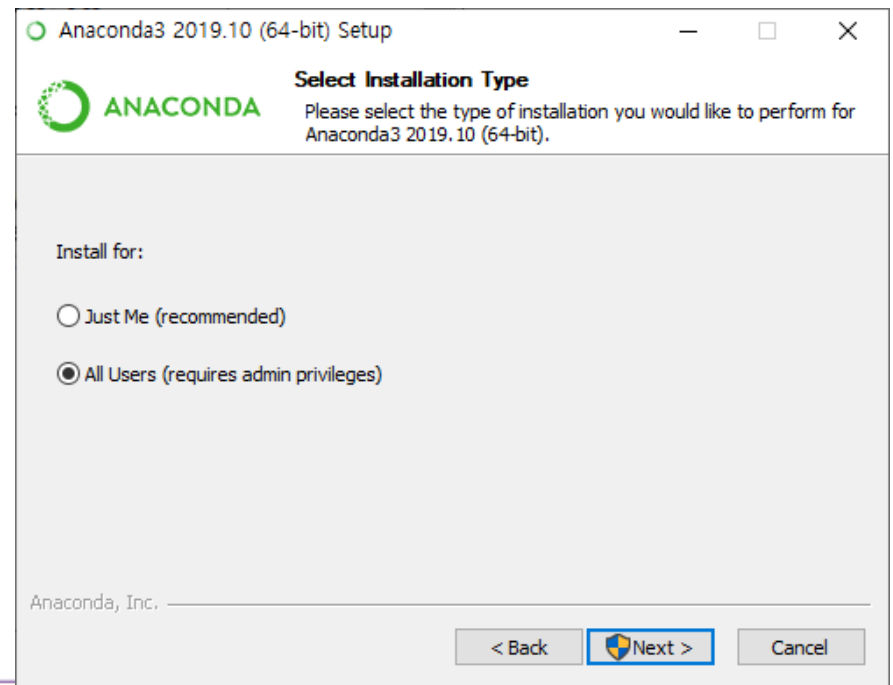
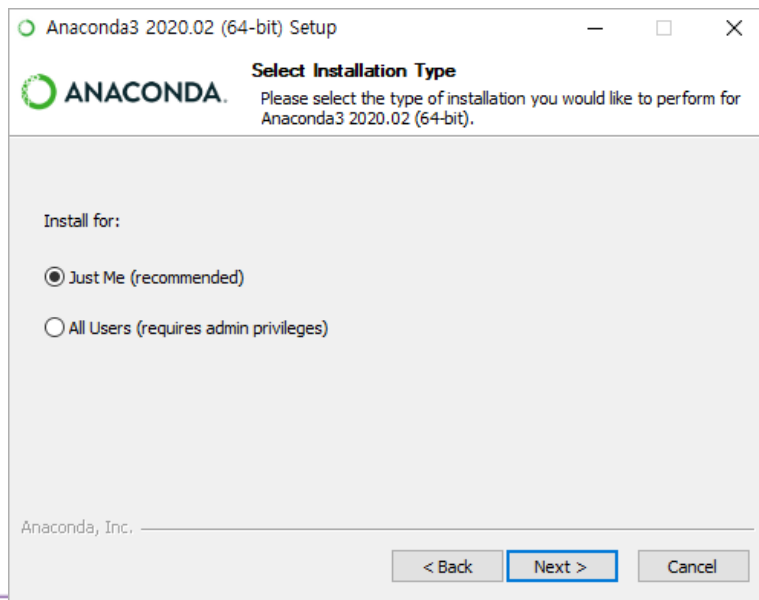
설치



설치

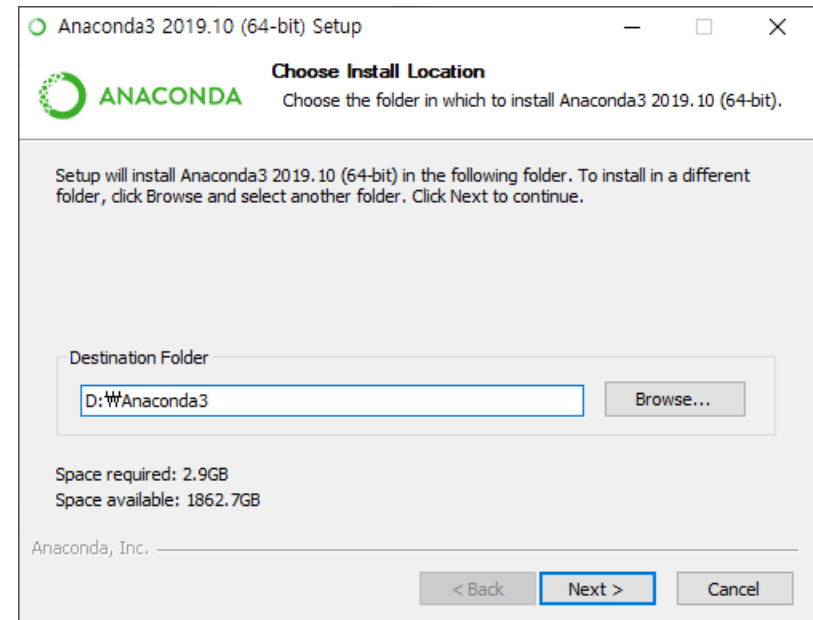
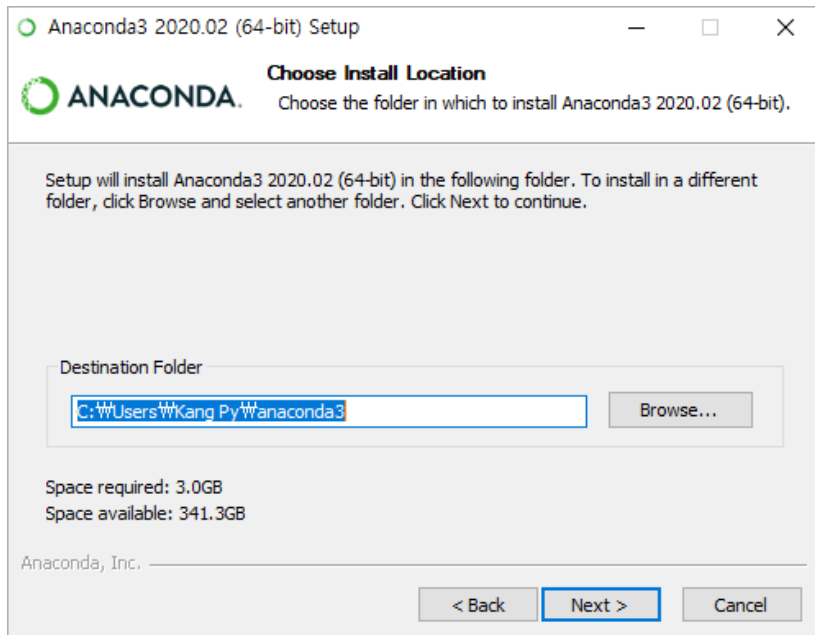
• 편한 것으로 선택

- All Users
 - Just Me와 설치되는 경로가 다름
 - C:\ProgramData\Anaconda3
- Just Me 설치
 - 패키지를 설치 또는 삭제 업그레이드 시 CMD창을 관리자 권한으로 열어 실행해야 하는 경우
 - C:\Users\[사용자]\Anaconda3



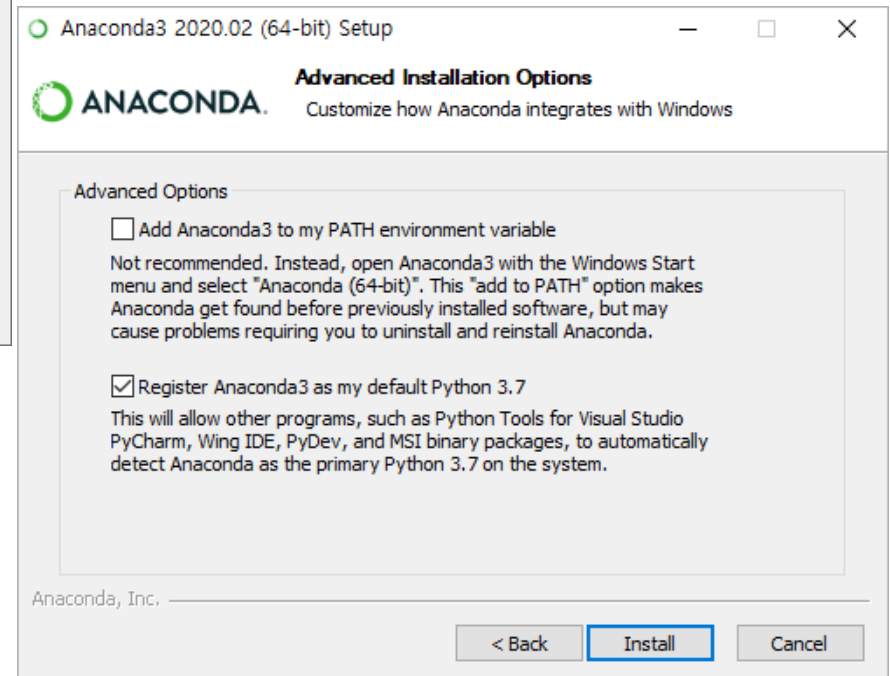
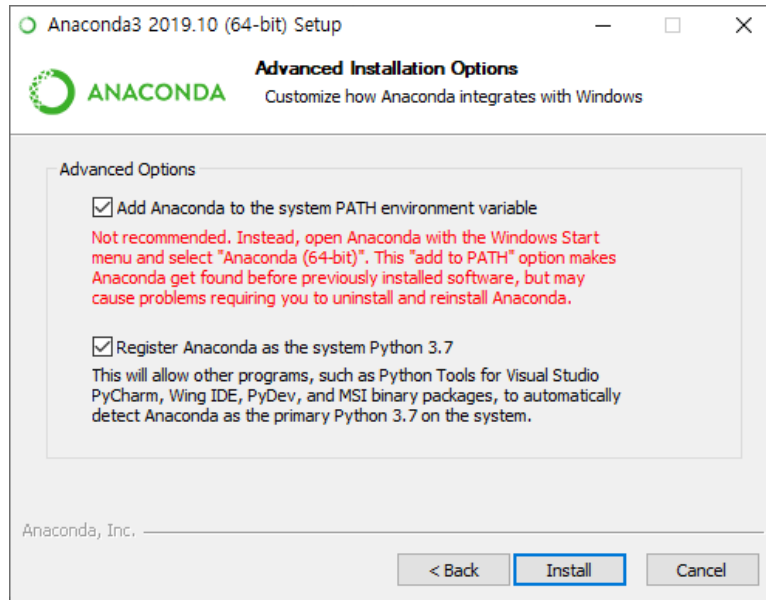
설치

- 설치 폴더
 - 쉽게 이동할 수 있는 폴더로 수정 권장
 - 상위 폴더에서 공백이 없도록
 - 설치가 안될 수 있음

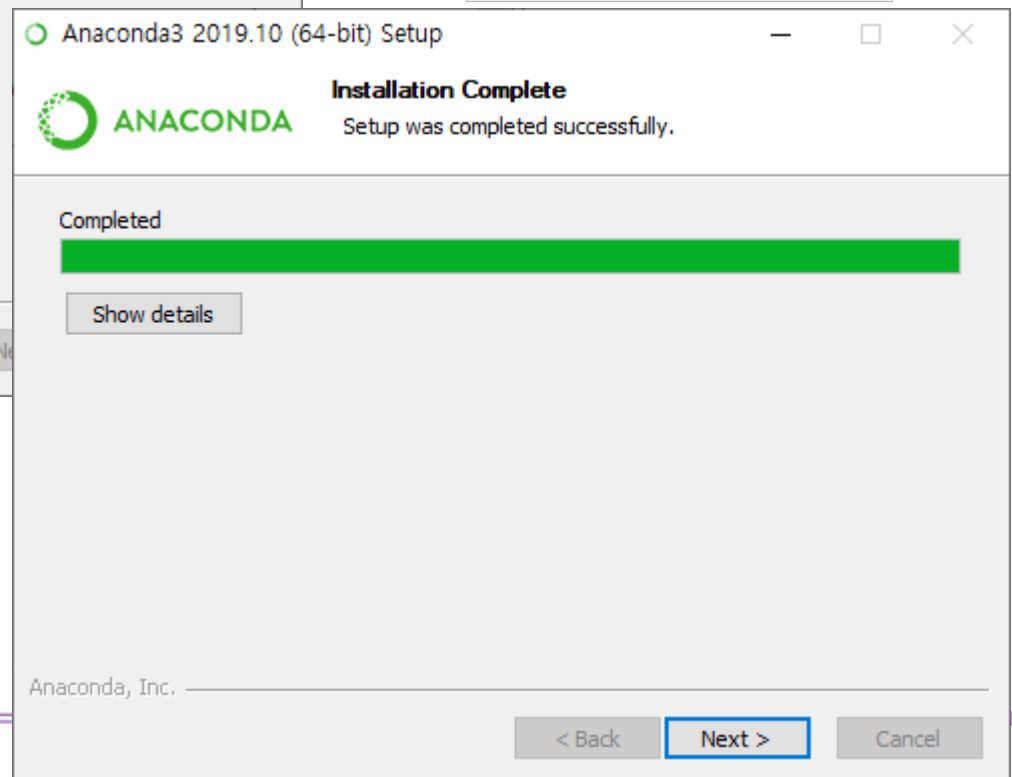
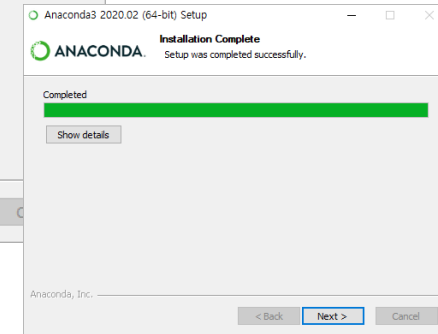
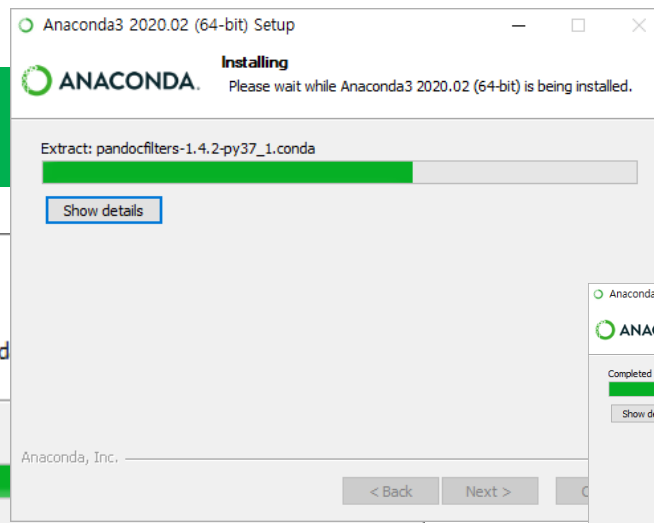
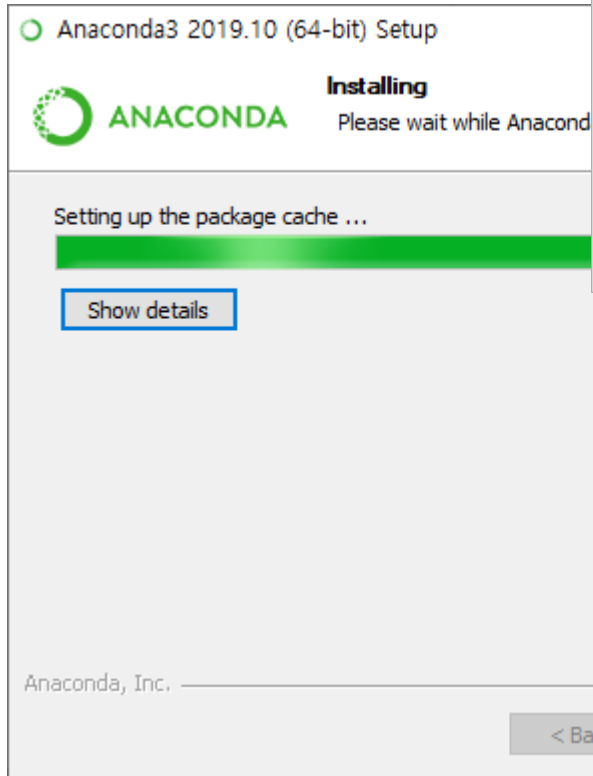


설치

- 편의를 위해 Path를 모두 설정하도록

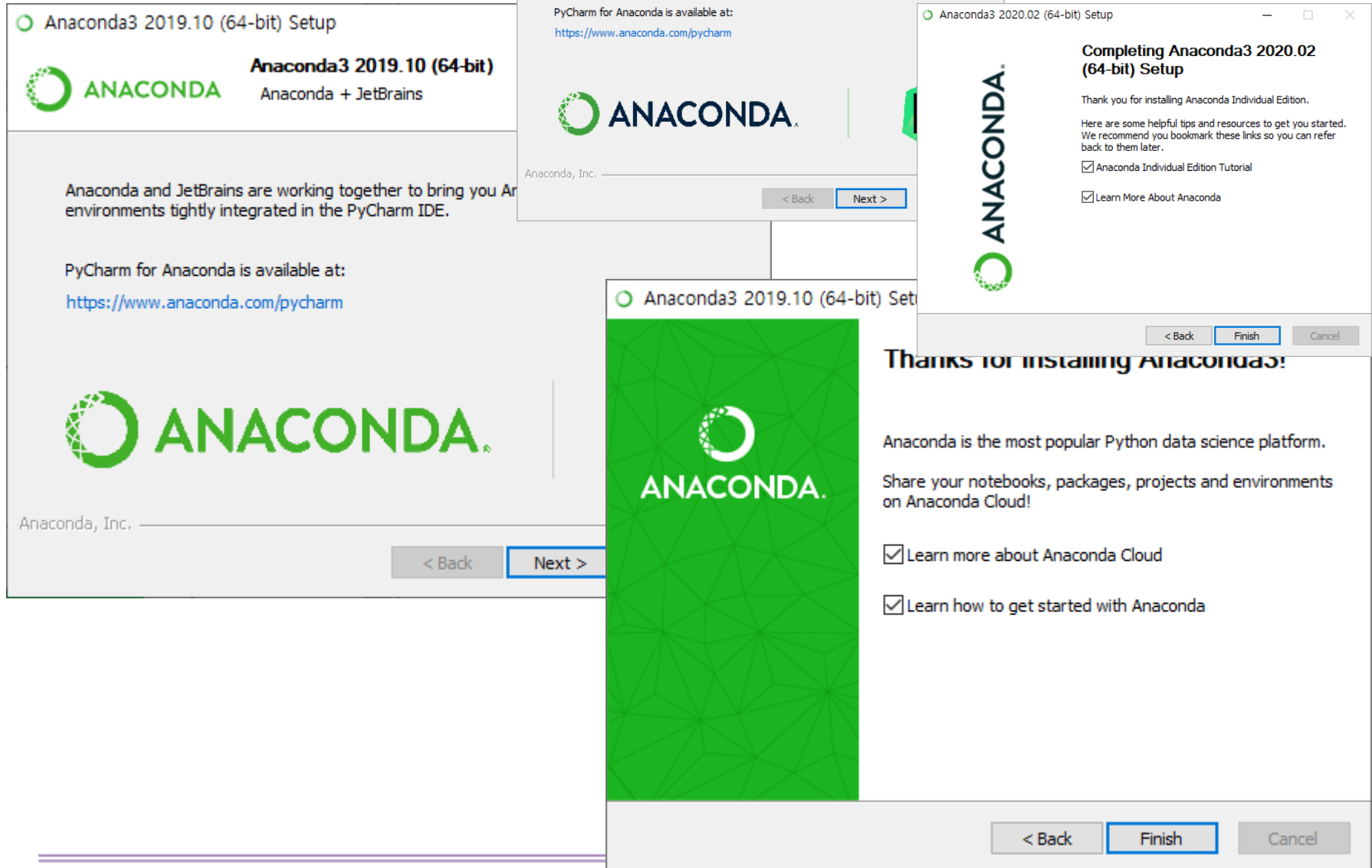


설치



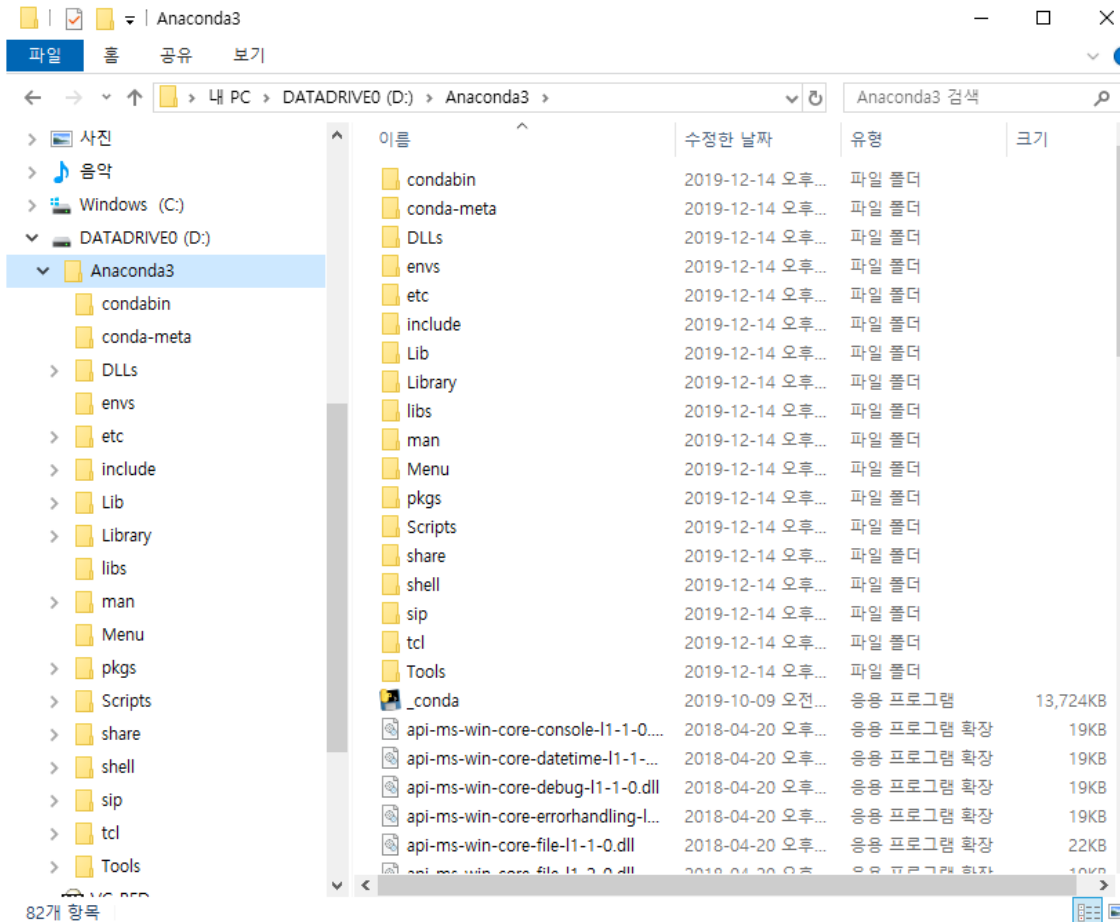
설치

PYTHON PROGRAMMING



설치

• 설치 후 폴더 확인



설치 후 path 확인

- 명령 prompt 창 열어서

설치 폴더가 다음인 경우

D:\ProgramData\20210223

- path 명령어

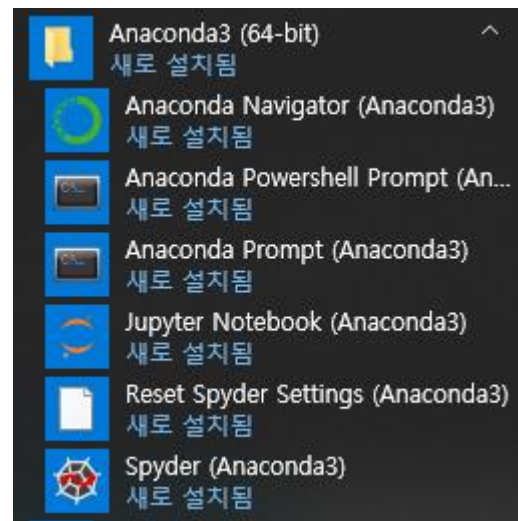
- PATH=D:\ProgramData\20210223\Anaconda3;
 - D:\ProgramData\20210223\Anaconda3\Library\mingw-w64\bin;
 - D:\ProgramData\20210223\Anaconda3\Library\usr\bin;
 - D:\ProgramData\20210223\Anaconda3\Library\bin;
 - D:\ProgramData\20210223\Anaconda3\Scripts;

- 위 path가 다른 파이썬이나 아나콘다보다 먼저 설정되어 있어야 함

다운로드 및 설치

설치 후 메뉴 확인

- 6가지 메뉴
 - Jupyter Notebook
 - 주피터 노트북 사용



기본 브라우저 설정

- '기본 앱' 설정
 - 크롬으로 지정



비디오 플레이어



영화 및 TV

웹 브라우저



Chrome

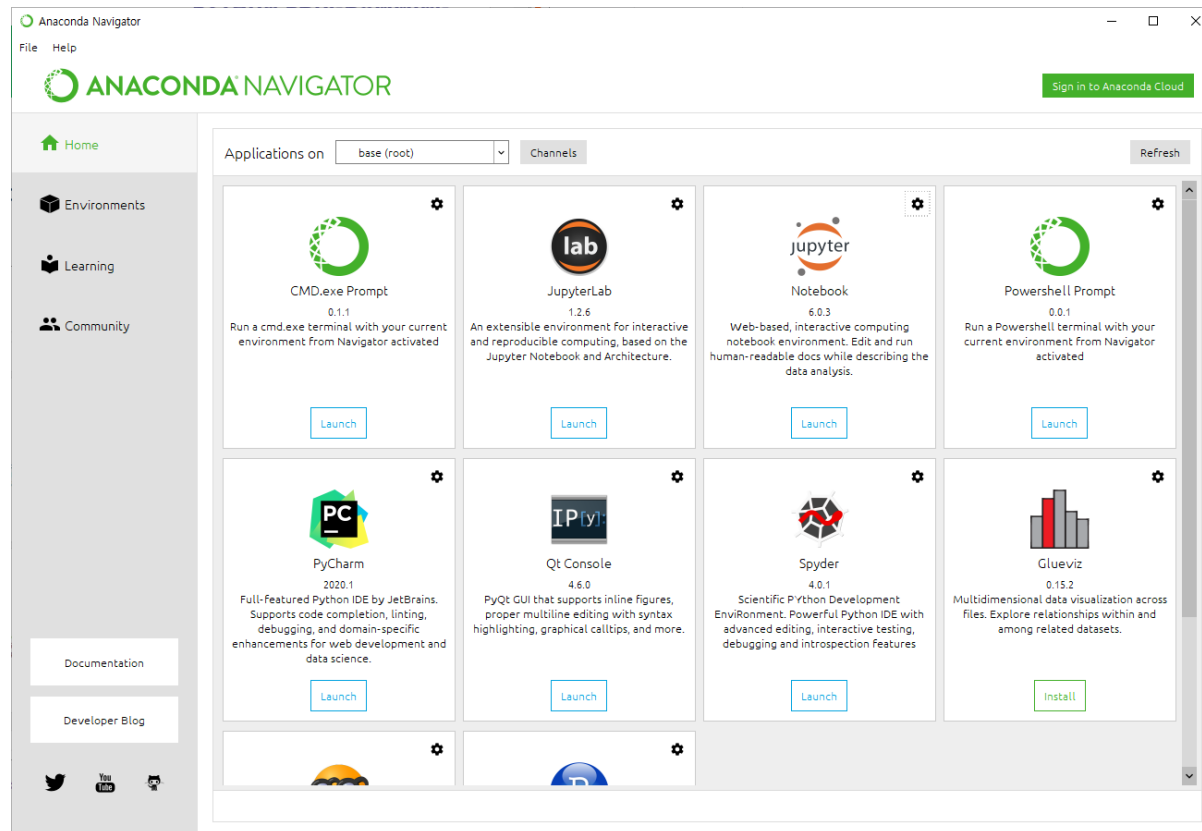
Microsoft 권장 기본값으로 초기화

초기화

설치

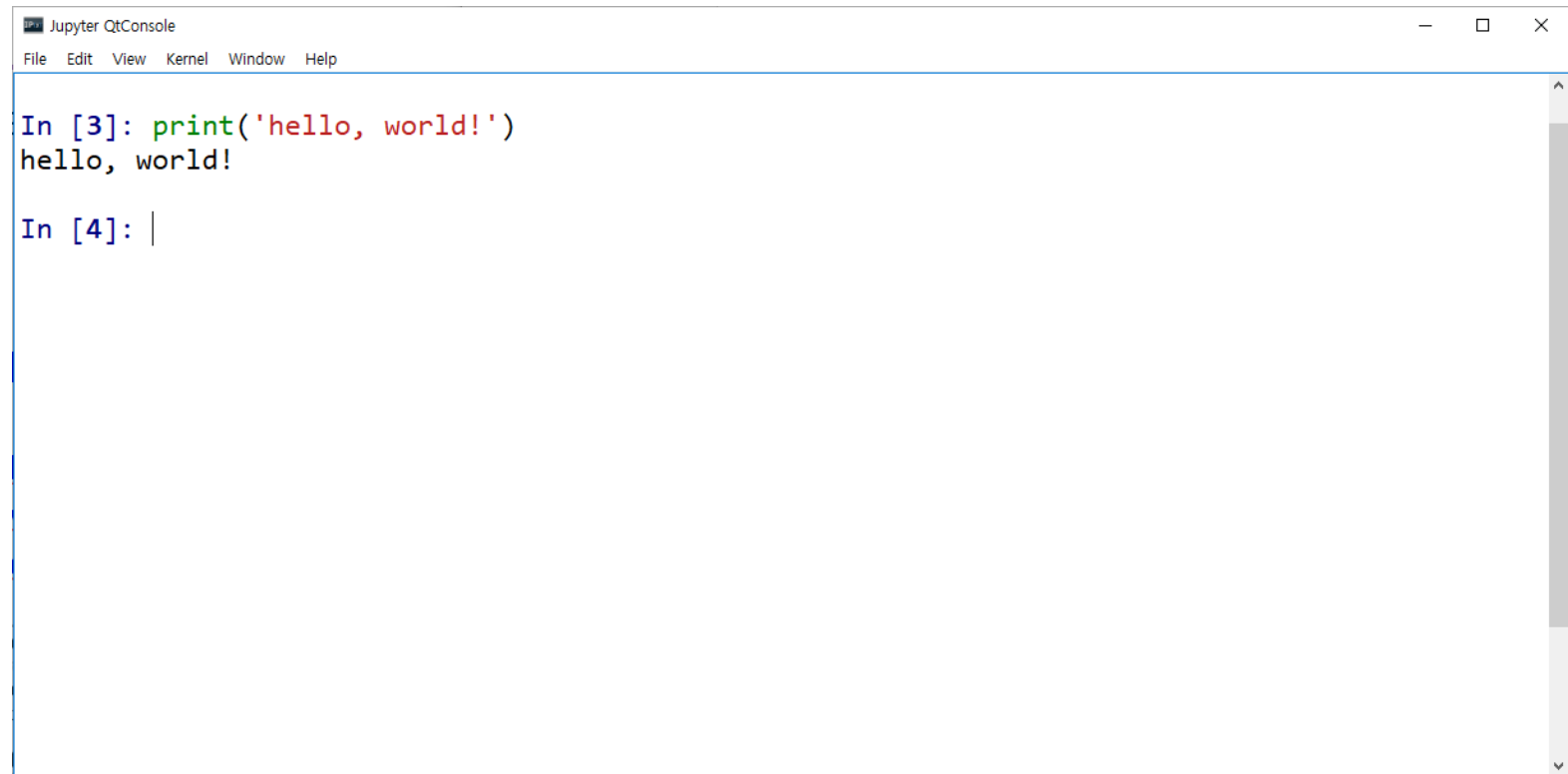
• 아나콘다 네비게이터 실행

- 데이터 과학 분야의 인기 개발 도구: jupyter notebook, jupyter lab
- 인기 IDE: spyder



Jupyter Qtconsole

- 네비게이터의 주피터 QtConsole 메뉴 Launch 버튼
 - D:\ProgramData\Anaconda3\Scripts\jupyter-qtconsole.exe

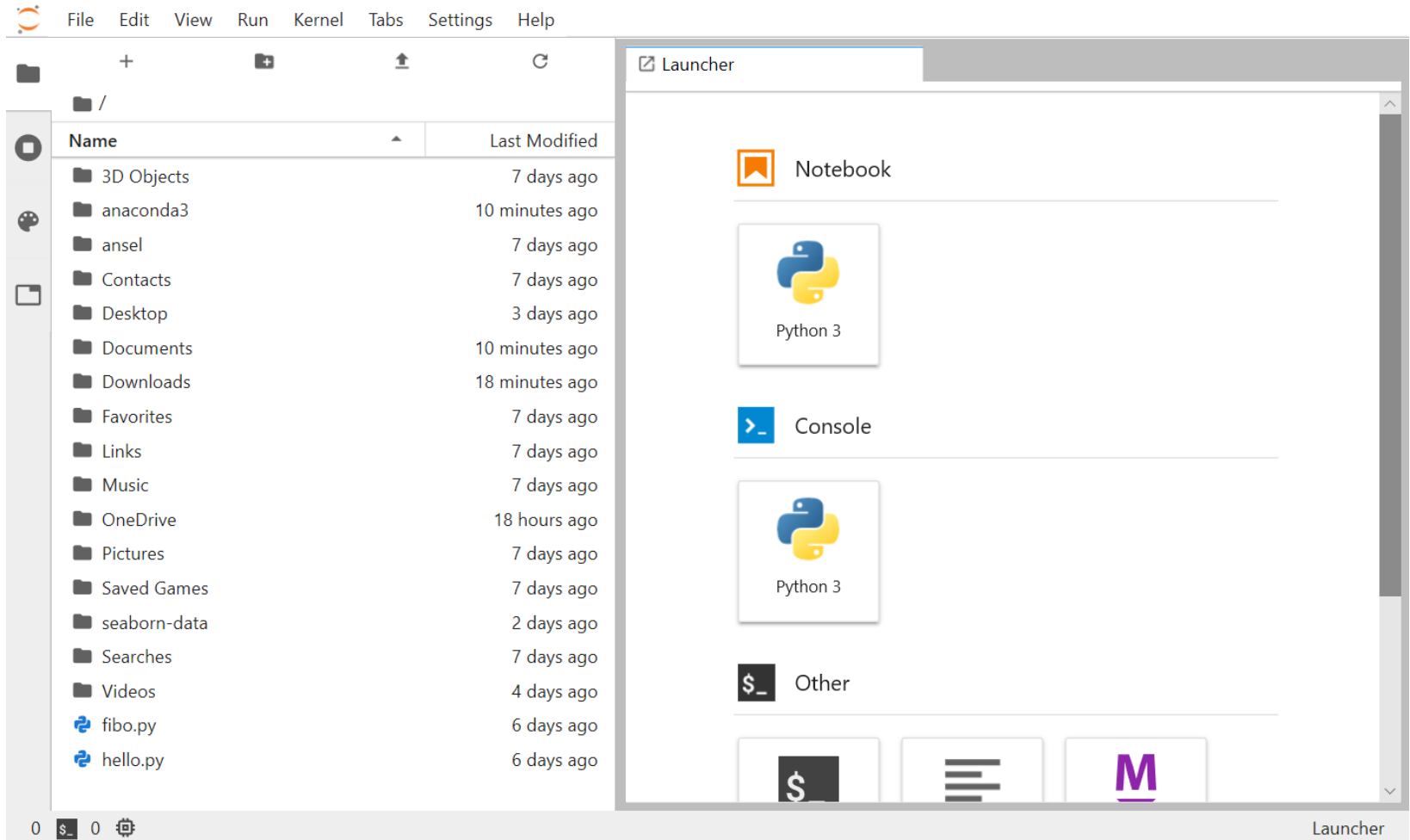


```
Jupyter QtConsole
File Edit View Kernel Window Help

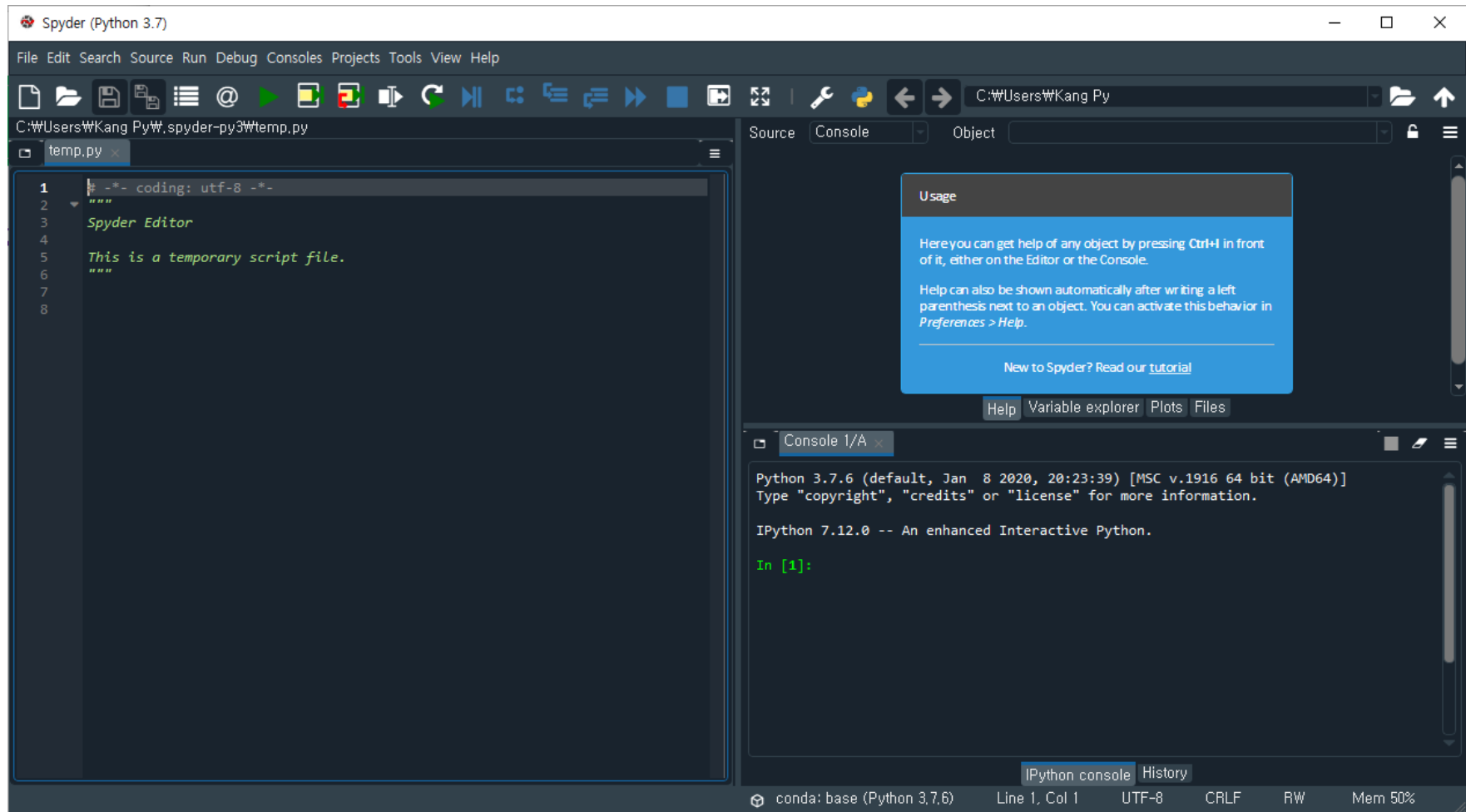
In [3]: print('hello, world!')
hello, world!

In [4]: |
```

Jupyter Lab

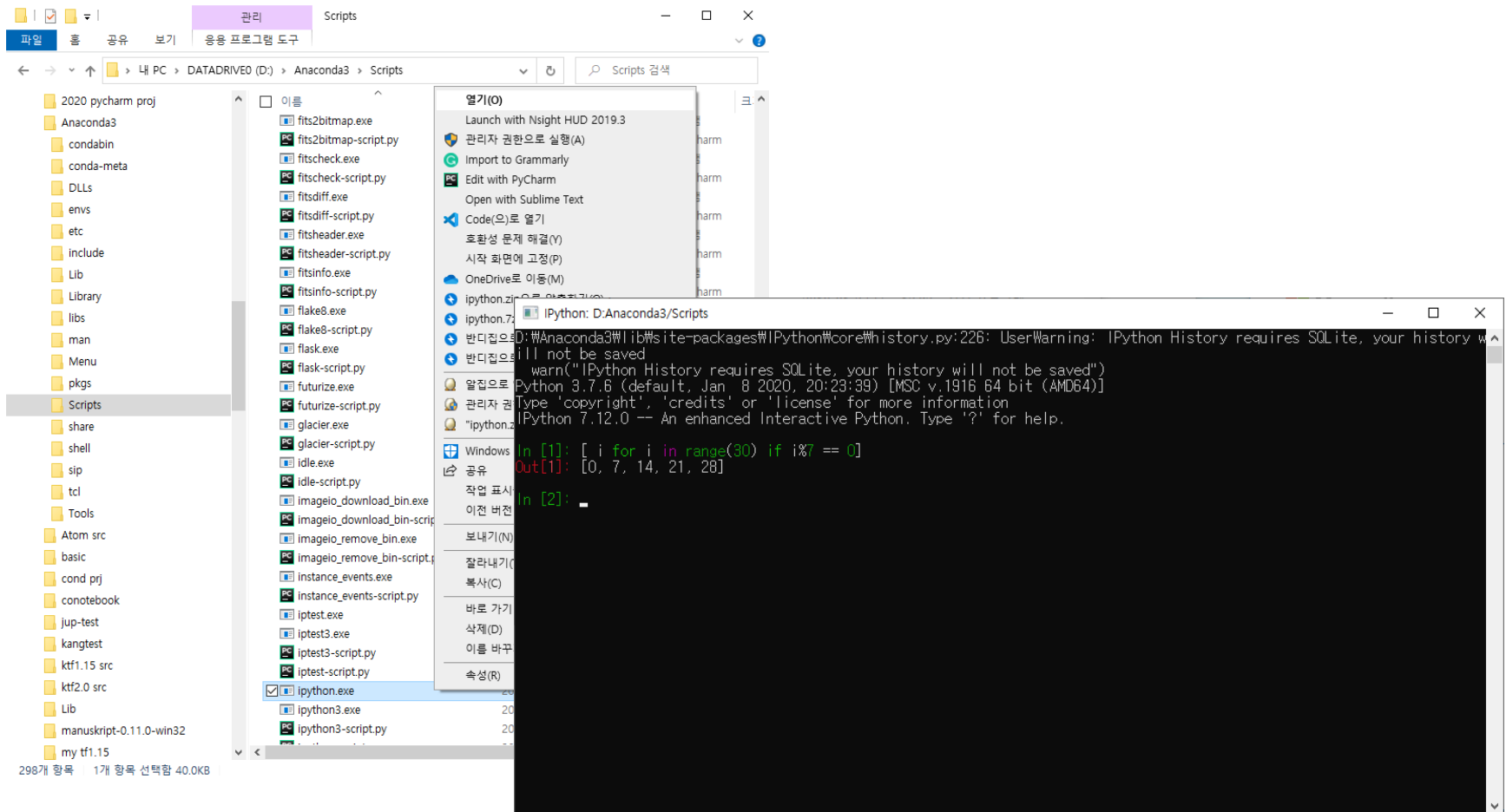


spyder



명령어 ipython

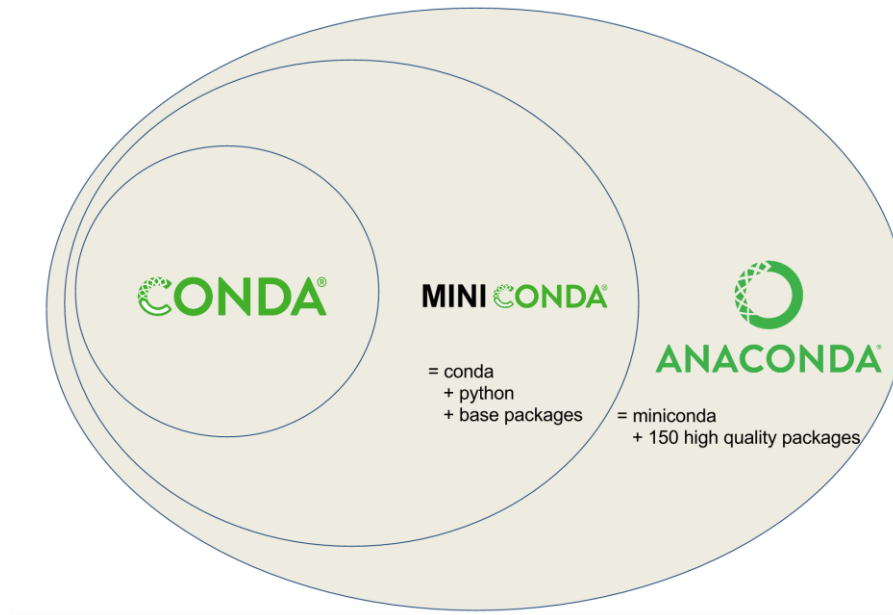
• [Anaconda3]/Scripts/ipython 셸



아나콘다와 미니콘다

- 아나콘다의 단점

- 너무 크다는 것
 - 150개의 패키지와 데이터 과학 패키지들이 포함
 - 쉽게 가상환경을 만들고 버전 관리를 할 수 있는 도구 포함
- 해결 방법
 - 미니콘다의 사용
 - <https://docs.conda.io/en/latest/miniconda.html>

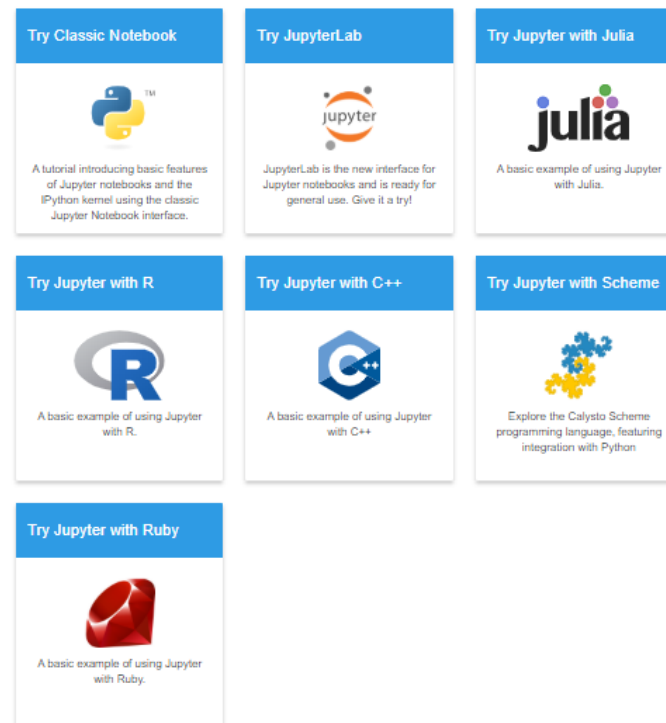


파이썬 개발환경
주피터 노트북 개요

주피터 노트북

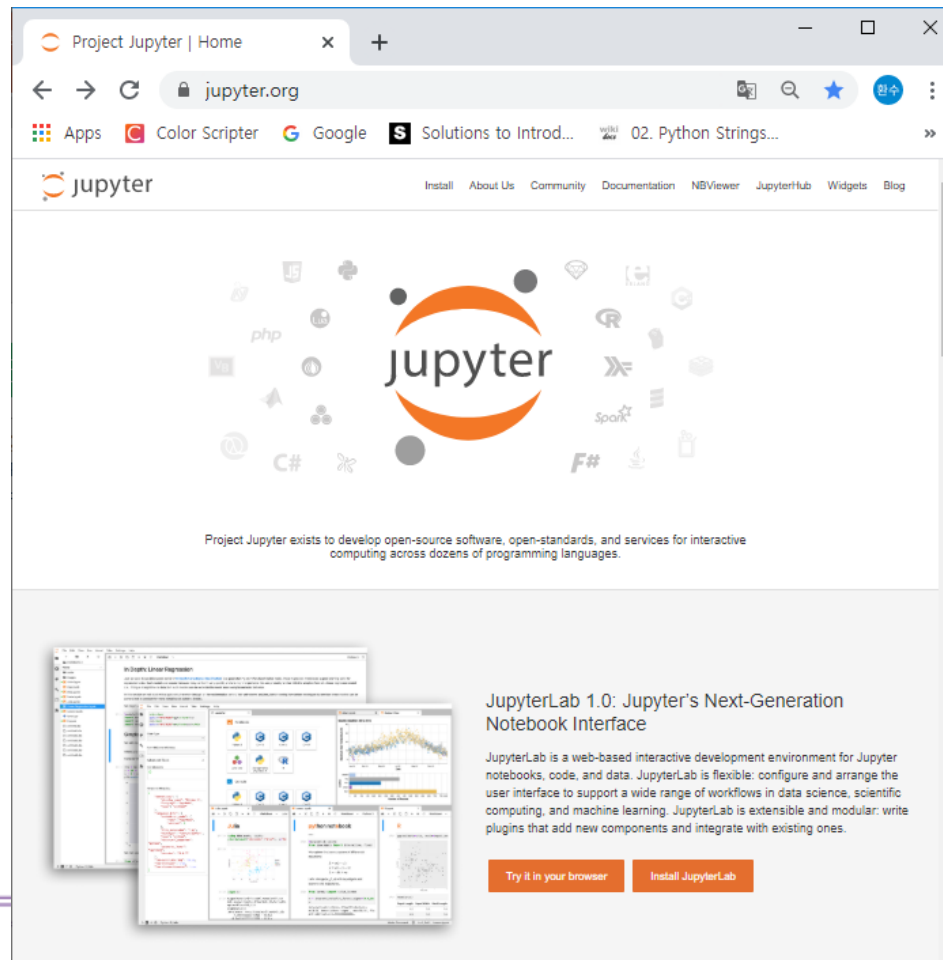
• Jupyter Notebook

- 소스와 설명인 문서를 여러 셀로 구분하여 저장, 실행하는 웹 브라우저 기반 개발환경
 - 라이브 코드, 방정식, 시각화 및 설명 텍스트가 포함 된 문서를 작성하고 공유할 수 있는 오픈 소스 웹 응용 프로그램
 - 데이터 정리 및 변환, 수치 시뮬레이션, 통계 모델링, 데이터 시각화, 기계 학습 등의 용도로 사용
 - 다양한 언어 지원
- 주피터 랩
 - 주피터 앱의 업그레이드 버전
 - 다른 IDE와 유사



주피터 홈페이지

- <https://jupyter.org>
 - Project Jupyter는 수십 개의 프로그래밍 언어에서 대화형 컴퓨팅을 위한 오픈 소스 소프트웨어, 오픈 표준 및 서비스를 개발하기 위해 존재



주피터 설치 방법

- 아나콘다 설치
- 그 외 다양한 주피터 설치 방법
 - Miniconda 설치
 - 모듈 notebook 설치
 - 모듈 jupyter 설치

가장 쉬운 실행 방법

- **아나콘다 설치 후 메뉴 실행**
 - Jupyter Notebook
- **기본 사용 방법**
 - 다음 사이트 참고
 - <https://copycoding.tistory.com/72>

주피터 노트북

- 자신의 PC에서 실행되는 서버
 - 검은 프롬프트 창에서 시작

```

Jupyter Notebook
[00:18:37.890 NotebookApp] JupyterLab extension loaded from D:\DNN\Anaconda3\lib\site-packages\jupyterlab
[00:18:37.891 NotebookApp] JupyterLab application directory is D:\DNN\Anaconda3\share\jupyter\lab
[00:18:37.893 NotebookApp] Serving notebooks from local directory: C:\Users\USER
[00:18:37.894 NotebookApp] The Jupyter Notebook is running at:
[00:18:37.894 NotebookApp] http://localhost:8888/?token=c73da5a2fc3c98297cd86d06b387ee4302a8cc38dd86cfbd
[00:18:37.894 NotebookApp] or http://127.0.0.1:8888/?token=c73da5a2fc3c98297cd86d06b387ee4302a8cc38dd86cfbd
[00:18:37.894 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 00:18:38.149 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/USER/AppData/Roaming/jupyter/runtime/nbserver-38708-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=c73da5a2fc3c98297cd86d06b387ee4302a8cc38dd86cfbd
or http://127.0.0.1:8888/?token=c73da5a2fc3c98297cd86d06b387ee4302a8cc38dd86cfbd
[E 00:18:40.654 NotebookApp] Could not open static file ''
[W 00:18:40.710 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (::1) 12.96ms referer=http://localhost:8888/tree?token=c73da5a2fc3c98297cd86d06b387ee4302a8cc38dd86cfbd

```

- 위 창을 종료하면 더 이상 사용 불가

프롬프트 창의 내용

• 다음 주요 정보가 표시

- 실행이 된 폴더 정보
- 종료 방법
- 실행 방법

```
[I 00:18:37.890 NotebookApp] JupyterLab extension loaded from D:\DNN\Anaconda3\lib\site-packages\jupyterlab
[I 00:18:37.891 NotebookApp] JupyterLab application directory is D:\DNN\Anaconda3\share\jupyter\lab
[I 00:18:37.893 NotebookApp] Serving notebooks from local directory: C:\Users\USER
[I 00:18:37.894 NotebookApp] The Jupyter Notebook is running at:
[I 00:18:37.894 NotebookApp] http://localhost:8888/?token=c73da5a2fc3c98297cd86d06b387ee4302a8cc38dd86cfbd
[I 00:18:37.894 NotebookApp] or http://127.0.0.1:8888/?token=c73da5a2fc3c98297cd86d06b387ee4302a8cc38dd86cfbd
[I 00:18:37.894 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 00:18:38.149 NotebookApp]
```

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

<file:///C:/Users/USER/AppData/Roaming/jupyter/runtime/nbserver-38708-open.html>

Or copy and paste one of these URLs:

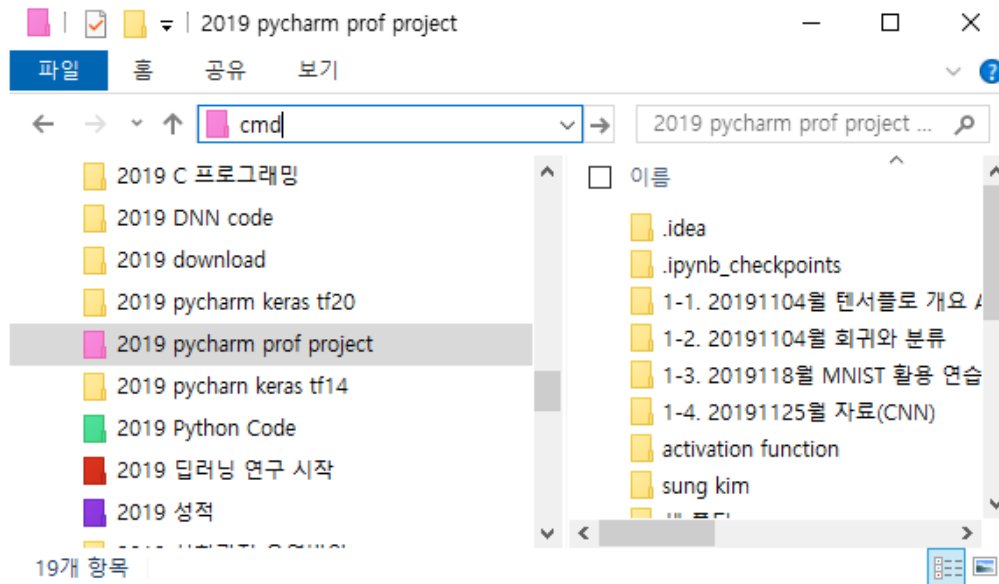
<http://localhost:8888/?token=c73da5a2fc3c98297cd86d06b387ee4302a8cc38dd86cfbd>

or <http://127.0.0.1:8888/?token=c73da5a2fc3c98297cd86d06b387ee4302a8cc38dd86cfbd>

원하는 폴더에서 주피터 노트북 실행 방법

방법 1: 프롬프트 창에서 명령어로 실행

- **프롬프트 창을 원하는 폴더에서 열기**
 - 탐색기에서 cmd 입력
 - 폴더에서 shift + 오른쪽 마우스 클릭
 - 여기에 프롬프트(PowerShell) 창 열기



프롬프트 창, 실행 명령어 1

- 다음 명령으로 현재 폴더에서 소스 편집 가능
 - `python -m notebook`

library 모듈을 스크립트의 형태로
실행시키는 것을 의미

```

C:\Users\USER>where python
C:\Python\Anaconda3\python.exe
C:\Python\Python37-32\python.exe
C:\Users\USER\AppData\Local\Microsoft\WindowsApps\python.exe

C:\Users\USER>python -m notebook
[I 00:36:18.555 NotebookApp] The port 8888 is already in use, trying another port.
[I 00:36:18.637 NotebookApp] JupyterLab extension loaded from C:\Python\Anaconda3\lib\site-packages\jupyterlab
[I 00:36:18.638 NotebookApp] JupyterLab application directory is C:\Python\Anaconda3\share\jupyter\lab
[I 00:36:18.639 NotebookApp] Serving notebooks from local directory: C:\Users\USER
[I 00:36:18.639 NotebookApp] The Jupyter Notebook is running at:
[I 00:36:18.639 NotebookApp] http://localhost:8889/?token=499c63a54099030ff65b6cad2c74407654f2c86e718c0fad
[I 00:36:18.639 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 00:36:18.882 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/USER/AppData/Roaming/jupyter/runtime/nbserver-48032-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=499c63a54099030ff65b6cad2c74407654f2c86e718c0fad
  
```


프롬프트 창, 실행 명령어 2

- 다음 명령으로 현재 폴더에서 소스 편집 가능
 - jupyter notebook

The image shows a Windows command prompt window and a web browser window side-by-side.

Command Prompt Window:

```

C:\Windows\System32\cmd.exe - jupyter notebook
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\2019 pycharm keras tf20>where jupyter
C:\Python\Anaconda3\Scripts\jupyter.exe

D:\2019 pycharm keras tf20>jupyter notebook
[I 00:42:22.750 NotebookApp] The port 8888 is already in use, trying another port.
[I 00:42:22.799 NotebookApp] JupyterLab extension loaded
[I 00:42:22.799 NotebookApp] JupyterLab application started
[I 00:42:22.800 NotebookApp] Serving notebooks from http://localhost:8889/
[I 00:42:22.800 NotebookApp] The Jupyter Notebook is available at http://localhost:8889/?token=38e274c22bec59
[I 00:42:22.800 NotebookApp] Use Control-C to stop this process and kill the server
[C 00:42:23.052 NotebookApp]
  
```

To access the notebook, open this file in a browser:
 file:///C:/Users/USER/AppData/Roaming/jupyterlab/
 Or copy and paste one of these URLs:
 http://localhost:8889/?token=38e274c22bec59

Web Browser Window:

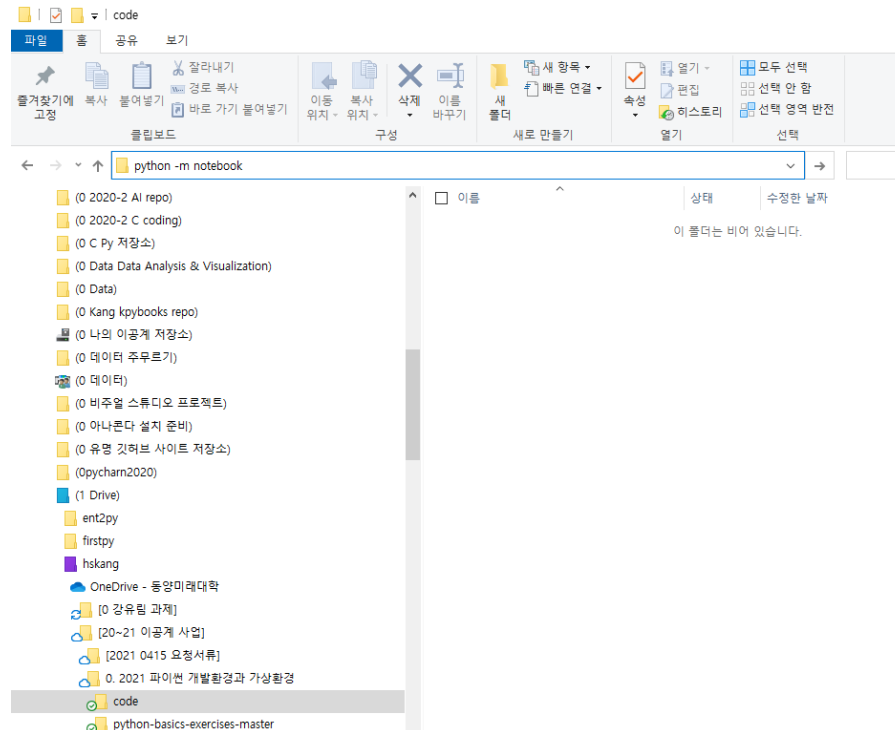
The browser shows the JupyterLab interface at localhost:8889/tree. The interface includes a sidebar with a file explorer showing the following files and folders:

Name	Last Modified	File size
2019 1227금 lecture	한 달 전	
2019 1230월 lecture	25일 전	
2020 fashion MNIST	19일 전	
20191021월 남파이 맷플롯리브 수업자료.ipynb	2달 전	282 kB
Untitled.ipynb	2달 전	72 B
맷플롯리브 한글처리.ipynb	2달 전	14 kB
텐서플로2.0 방식의 선형회귀.ipynb	2달 전	61.3 kB
TF 2.0 기본.py	2달 전	578 B
텐서플로2 회귀.py	2달 전	106 B

주피터 노트북 실행 방법과 코딩 실행

실행 명령어

- jupyter notebook
- python -m notebook
- 원하는 폴더 하부에서 바로 명령어 가능



주피터 노트북 기본 사용

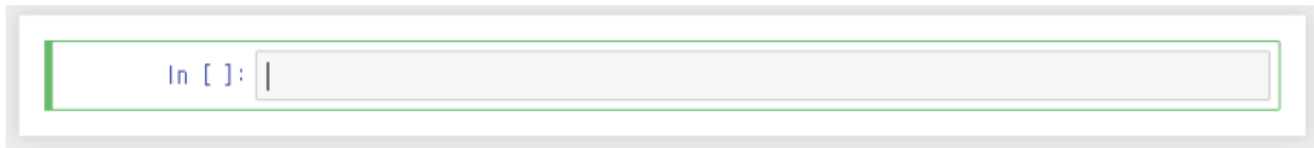
- <https://hogni.tistory.com/29>

3. 모드 소개

주피터 노트북에는 두가지 모드가 있습니다. Edit과 Command입니다.

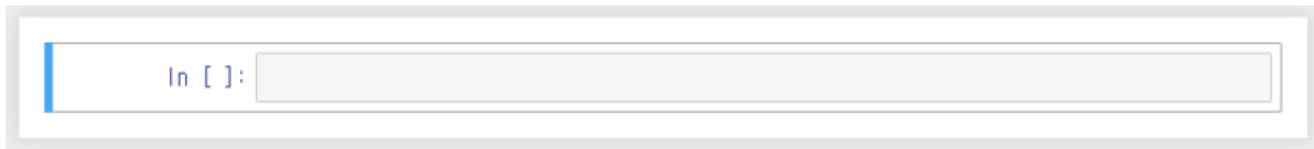
Edit 모드

Edit 모드를 시작하려면 셀을 클릭하거나 키보드에서 Enter를 누르면 됩니다. 셀의 테두리가 녹색이면 Edit 모드입니다. Edit 모드일 때만 셀에 내용을 입력할 수 있습니다.



Command 모드

Command 모드를 시작하려면 셀 외부의 아무 곳이나 클릭하거나 키보드에서 Esc를 누르면 됩니다. 셀의 테두리가 파란색이면 Command 모드입니다. Command 모드일 때는 노트북을 편집할 수 있습니다. 다만 셀에 내용은 입력할 수 없습니다.



✓ command 모드에서 키보드 위아래 버튼 눌러서 셀을 옮겨 다닐 수 있습니다.

✓ command 모드에서 키보드 m를 눌러서 code 입력창을 markdown 입력창으로 변경할 수 있습니다. 다시 code 입력창으로 돌아오려면 y를 누르면 됩니다.

셀 다루기

4. 셀 다루기

이번에는 셀을 추가, 삭제, 복사, 붙여 넣고 셀을 위아래로 이동시키는 방법을 알아보겠습니다.

셀 추가하기

- 현재 셀 위에 추가: (command 모드) a
- 현재 셀 아래 추가: (command 모드) b

셀 삭제하기

- 현재 셀 삭제: (command 모드) dd
- 셀 제거 취소하기: (command 모드) z

셀 복사하기

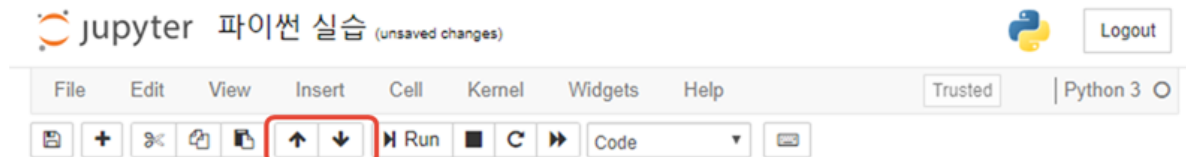
- 현재 셀 복사: (command 모드) c

셀 붙여넣기

- 현재 셀 위에 붙여 넣기: (command 모드) Shift + v
- 현재 셀 아래 붙여 넣기: (command 모드) v

셀 이동시키기

- 위아래 버튼을 클릭해서 현재 셀을 이동시킬 수 있습니다.



셀 실행

- **Shift + Enter**
 - 현재 셀 실행 후 다음 셀로 이동(다음 셀이 없으면 생성)
- **Ctrl + Enter**
 - 현재 셀 실행
- **Alt + Enter**
 - 현재 셀 실행 후 다음 셀 삽입해 이동

5. 코드 실행하기

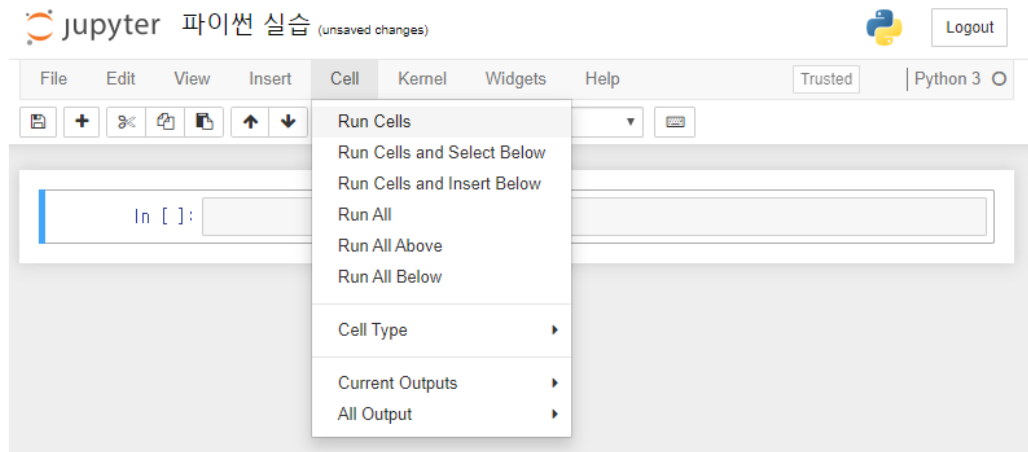
Edit 모드에서 코드를 실행시킬 수 있습니다.

셀 단위 실행하기

- 현재 셀을 실행하고 다음 셀로 이동하기: (command 모드) Shift + Enter
- 현재 셀을 실행하기: (command 모드) Ctrl + Enter

전체 코드 실행하기

- Run Cells을 클릭하면 모든 셀의 코드가 순서대로 실행됩니다.



주피터 노트북 연습

- <https://github.com/dongupak/Prime-Python>

대화창 실습 : person 딕셔너리의 생성과 조회

```
In [0]: person = {'이름' : '홍길동', '나이' : 26, '몸무게' : 82 }
```

```
In [0]: person['이름']
```

```
In [0]: person['나이']
```

```
In [0]: person['몸무게']
```

대화창 실습 : students 딕셔너리의 생성과 조회

```
In [0]: students = { 2019001:'이순신', 2019002:'김유신', 2019003:'강감찬' }
students[2019001]
```

```
In [0]: students[2019002]
```

```
In [0]: students[2019003]
```

대화창 실습 : 딕셔너리의 항목 삽입

```
In [0]: person = {'이름' : '홍길동', '나이' : 26, '몸무게' : 82 }
```

```
In [0]: person['직업'] = '율도국의 왕' # 새로운 키:값의 항목 삽입
```

```
In [0]: person
```

```
Out[0]: {'나이': 26, '몸무게': 82, '이름': '홍길동', '직업': '율도국의 왕'}
```

주피터 노트북 주요 편집 단축키

주요 단축키 모음

- <https://kkokkilkon.tistory.com/151>

2 모드

- 명령모드
- 편집모드

(1) 셀 선택 모드 (Command Mode)

[esc] 또는 [ctrl] + [m]를 눌러 셀이 아래와 같이 파란색이 된 상태(셀 선택 모드)에서 해당 단축키 누른다.



위로 셀 추가

[a]

아래로 셀 추가

[b]

선택 셀 삭제

[d][d] (d를 두번 누름)

선택 셀 잘라내기 (삭제로 써도 무방)

[x]

선택 셀 복사하기

[c]

선택 셀 아래에 붙여넣기

[p] **V**

선택 셀과 아래 셀과 합치기

[shift] + [m]

실행결과 열기/닫기

[o]

Markdown으로 변경

[m]

Code로 변경

[y]

파일 저장

[ctrl] + [s] 또는 [s]

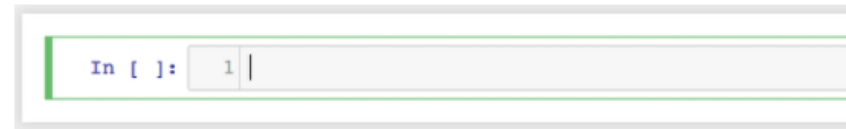
선택 셀의 코드 입력 모드로 돌아가기

[enter]

편집 모드

(2) 코드 입력 모드 (Edit Mode)

[enter]를 눌러 셀이 아래와 같이 초록색이 된 상태(코드 입력 모드)에서 해당 단축키 누름



선택 셀의 코드 전체 선택

[ctrl] + [a]

선택 셀 내 코딩 취소

[ctrl] + [z]

선택 셀 내 취소된 코딩 다시

[ctrl] + [y]

커서 위치 라인 주석처리

[ctrl] + [/]

선택 셀 코드 실행

[ctrl] + [enter]

선택 셀 코드 실행 후 다음 Cell로 이동 (없으면 새로 추가)

[shift] + [enter]

커서 위치에서 셀 둘로 나누기

[shift] + [ctrl] + [-]

파일 저장

[ctrl] + [s]

셀 선택 모드로 돌아가기

[esc]

주피터 노트북 단축키 정리

셀의 모드 구분

• 명령(command) 모드와 편집(edit) 모드

- 편집 모드: 녹색, 셀 클릭

In []:	1	
Esc	Command Mode로 전환	
Ctrl-Z	되돌리기	
Ctrl-Y	앞으로 되돌리기	
Tab	코드 완성 or 들여쓰기	

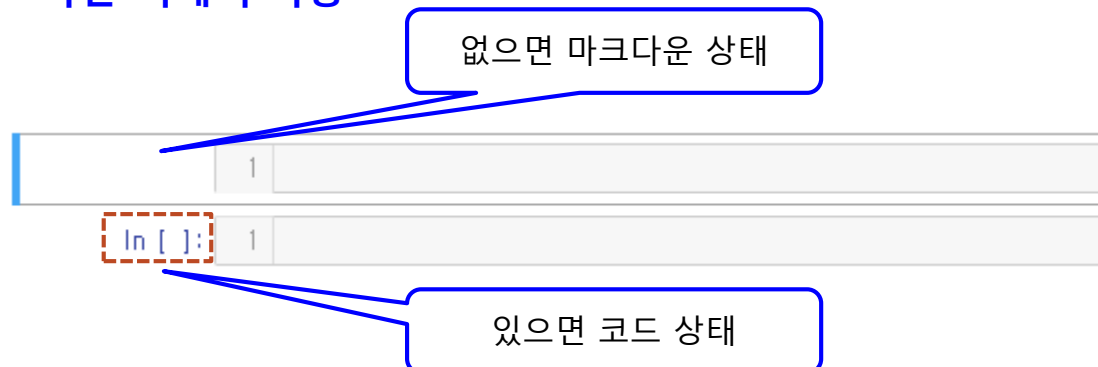
- 명령 모드: 파란색, Esc 키(ctrl+M) 또는 외부 클릭

In []:	1	
Enter	Edit Mode로 전환	
Y	Code로 전환	
M	Markdown로 전환	
D,D	셀 삭제	
X	셀 잘라내기	
V	셀 붙여넣기	
A	위에 셀 추가	
B	아래에 셀 추가	
H	단축키 도움말	

• 코드와 마크다운

셀의 생성과 코드와 마크다운 간의 변환

- 셀 코딩 이후 실행
 - Shift + Enter
 - 셀 실행 후 다음 셀로 이동, 다음 셀이 없으면 셀 생성
 - Ctrl + Enter
 - 단순 셀 실행 후 명령 모드로 전환
 - Alt + Enter
 - 셀 실행 후 다음 셀 삽입
- 명령 모드에서 위와 아래에 셀 삽입 후 이동
 - A(bove)과 B(elow)
- 셀의 코드와 마크다운 간의 변환
 - 명령 모드에서 Y(코드)와 M(마크다운)으로 변환
 - 파란 색에서 가능



자동 완성 tab 키

- 변수나 객체 입력 후 . 이후 tab
 - 사용할 수 있는 변수나 객체, 함수 등을 보여줌
- 변수나 객체, 함수 입력 이후 Shift + tab
 - 내부 특성이 조회

```
In [3]: 1 import random as rd
        2
        3 print(rd.random())
        4 print(rd.randint(1, 3))
        5 print(rd.)
```

0.69422281671
1

Type Markdown

```
In [ ]: 1 sum
```

```
In [ ]: 1
```

betavariate
BPF
choice
choices
expovariate
gammavariate
gauss
getrandbits
getstate
LOG4

```
In [4]: 1 import random as rd
        2
        3 print(rd.random())
        4 print(rd.randint(1, 3))
        5 print(rd.choice(range(10)))
        6 print(rd.choices())
        7 rd
```

Type: module
String form: <module 'random' from 'd:\myve\nb\lib\random.py'>
File: d:\myve\nb\lib\random.py
Docstring:

정보와 도움말 조회 ?, ?? 붙이고 실행

- ?를 붙이고 실행

- 내부 정보 조회

```
In [5]: 1 list?
```

Init signature: list(iterable=(), /)
Docstring:
Built-in mutable sequence.

If no argument is given, the constructor creates a new empty list.
The argument must be an iterable if specified.

Type: type
Subclasses: _HashedSeq, StackSummary, SList, _ImmutableLineList, FormattedText, Node
List, _ExplodedList, Stack, _Accumulator, _ymd, ...

- ??를 붙이고 실행

- 도움말 정보, 대부분 위와 동일

주요 명령 단축키

- **D,D**
 - delete selected cell
- **Up, Down(메뉴)**
 - 셀 이동
- **Shift-M**
 - merge cell below
- **Ctrl-Shift-Subtract(-)**
 - split cell

주피터 노트북 단축키 정리

- 'Jupyter notebook cheat sheet'으로 검색
 - 다양한 치트 쉬트
 - <https://www.dataday.life/notes/jupyter-notebooks/cheat-sheet-jupyter-notebook-keyboard-shortcuts/>
 - <https://medium.com/edureka/jupyter-notebook-cheat-sheet-88f60d1aca7>

명령 모드(파란색) 입력

In []:

항목	설명	
모드 변경	Enter(편집 모드로 변경)	
저장	Ctrl + s	S
셀 타입	y(code)	m(markdown)
셀 생성	A(above)	B(below)
셀 편집	X(cut)	C(copy)
	V(아래 붙이기)	Shift+v(위에 붙이기)
	dd(삭제)	z(undo)
바로 마크다운의 헤드라인 편집	1(#head1)	6(#head6)
셀 실행	Shift + enter	현 셀 실행 다음 셀 이동 (없을 시 생성 후 이동)
	Ctrl + enter	현 셀 실행
	Alt + enter	현 셀 실행 다음 셀 삽입
여러 셀 선택	Shift + click cell	연속된 여러 셀 선택
	Shift + 화살표 위/아래	위/아래 연속된 여러 셀 선택
셀 병합	Shift + m	현재 셀과 다음 셀 병합
		여러 셀이 선택되었다면 선택된 셀 병합
도움말	H(도움말)	Esc(도움말 없애기)
	ctrl+shift+p(명령 팔레트)	Esc(명령 팔레트 없애기)

편집 모드(녹색) 입력

In []:

항목	설명	
모드 변경	ESC(명령모드)	
편집	Ctrl + A	전체 선택
	Ctrl + C	복사
	Ctrl + V	붙여 넣기
	Ctrl + Z	최소(undo)
	Ctrl + Y	취소 다시 실행(redo)
	Ctrl + [, Ctrl +]	현재 커서에서 들여쓰기, 내여쓰기
저장	Ctrl + S	파일 저장
셀 추가	ESC + A	위에 추가
	ESC + B	아래에 추가
다중 커서	Ctrl + click	여러 줄을 동시에 코딩 가능
주석 달기	Ctrl + /	여러 줄이면 모두 주석. 토글 기능
셀 분리	Ctrl + shift + -	현재 커서 이후를 다음 셀로 분리
코드 편집 도움	tab	코드 중에 필요한 코드를 보여줌
	Shift + tab	툴팁(tooltip) 보여줌(함수 인자 등 설명)
	? ??	함수의 도움말 전체를 보여줌
셀 실행	Shift + enter	현 셀 실행 다음 셀 이동(없을 시 생성 후 이동)
	Ctrl + enter	현 셀 실행
	Alt + enter	현 셀 실행 다음 셀 삽입
코드 선택	shift + click	이전 커서에서 현재 커서까지 선택
명령 팔레트	ctrl+shift+p(명령 팔레트)	Esc(명령 팔레트 없애기)

키보드 단축키 보기

- 명령 모드에서 h

Command Mode (press `Esc` to enable)

`F`: find and replace
`Ctrl-Shift-F`: open the command palette
`Ctrl-Shift-P`: open the command palette
`Enter`: enter edit mode
`P`: open the command palette
`Shift-Enter`: run cell, select below
`Ctrl-Enter`: run selected cells
`Alt-Enter`: run cell and insert below
`Y`: change cell to code
`M`: change cell to markdown
`R`: change cell to raw
`1`: change cell to heading 1
`2`: change cell to heading 2
`3`: change cell to heading 3
`4`: change cell to heading 4
`5`: change cell to heading 5
`6`: change cell to heading 6
`K`: select cell above
`Up`: select cell above
`Down`: select cell below
`J`: select cell below
`Shift-K`: extend selected cells above
`Shift-Up`: extend selected cells above
`Shift-Down`: extend selected cells below

PYTHON PROGRAMMING

Edit Shortcuts

`Shift-J`: extend selected cells below
`Ctrl-A`: select all cells
`A`: insert cell above
`B`: insert cell below
`X`: cut selected cells
`C`: copy selected cells
`Shift-V`: paste cells above
`V`: paste cells below
`Z`: undo cell deletion
`D`, `D`: delete selected cells
`Shift-M`: merge selected cells, or
current cell with cell below if
only one cell is selected
`Ctrl-S`: Save and Checkpoint
`S`: Save and Checkpoint
`L`: toggle line numbers
`O`: toggle output of selected cells
`Shift-O`: toggle output scrolling of
selected cells
`H`: show keyboard shortcuts
`I`, `I`: interrupt the kernel
`Q`, `Q`: restart the kernel (with dialog)
`Eso`: close the pager
`Q`: close the pager
`Shift-L`: toggles line numbers in all
cells, and persist the setting
`Shift-Space`: scroll notebook up
`Space`: scroll notebook down