

# 1장 실습 환경 준비

1. 아나콘다 소개
2. 파이썬 삭제
3. 아나콘다 설치
4. 아나콘다 환경 관리
5. 주피터 노트북 소개
6. 주피터 노트북 활용



# 실습 환경 준비

---



1. 아나콘다 소개
2. 파이썬 삭제
3. 아나콘다 설치
4. 아나콘다 환경 관리
5. 주피터 노트북 소개
6. 주피터 노트북 활용

## 학습 목표

- 데이터 과학에서 아나콘다란 무엇인지 설명할 수 있다.
- 아나콘다와 일반 파이썬의 차이를 설명할 수 있다.
- 아나콘다를 설치할 수 있다.
- 아나콘다를 이용해서 가상환경 관리와 패키지 관리를 수행할 수 있다.
- 데이터 과학에서 주피터 노트북이 무엇인지 설명할 수 있다.
- 주피터 노트북을 활용하여 코드와 텍스트를 통합한 문서를 작성할 수 있다.

# 1. 아나콘다 소개

- 아나콘다(anaconda)를 럭셔리 버전의 파이썬으로 생각해도 좋다.
  - 아나콘다에는 다양한 구성 요소가 기본적으로 포함되어 있다.
- 파이썬 인터프리터
- 데이터 과학을 위한 통합개발 환경:  
Jupyter, Spyder, JupyterLab, R-Studio
- 분석 및 과학 계산 라이브러리:  
Numpy, Pandas, SciPy, Numba, DASK, ...
- 시각화 라이브러리:  
matplotlib, Altair, Bokeh, DataShader, HoloViews, ...
- 기계학습 라이브러리:  
TensorFlow, Scikit-learn, H2O.ai, theano, ...
- conda: 패키지 관리자

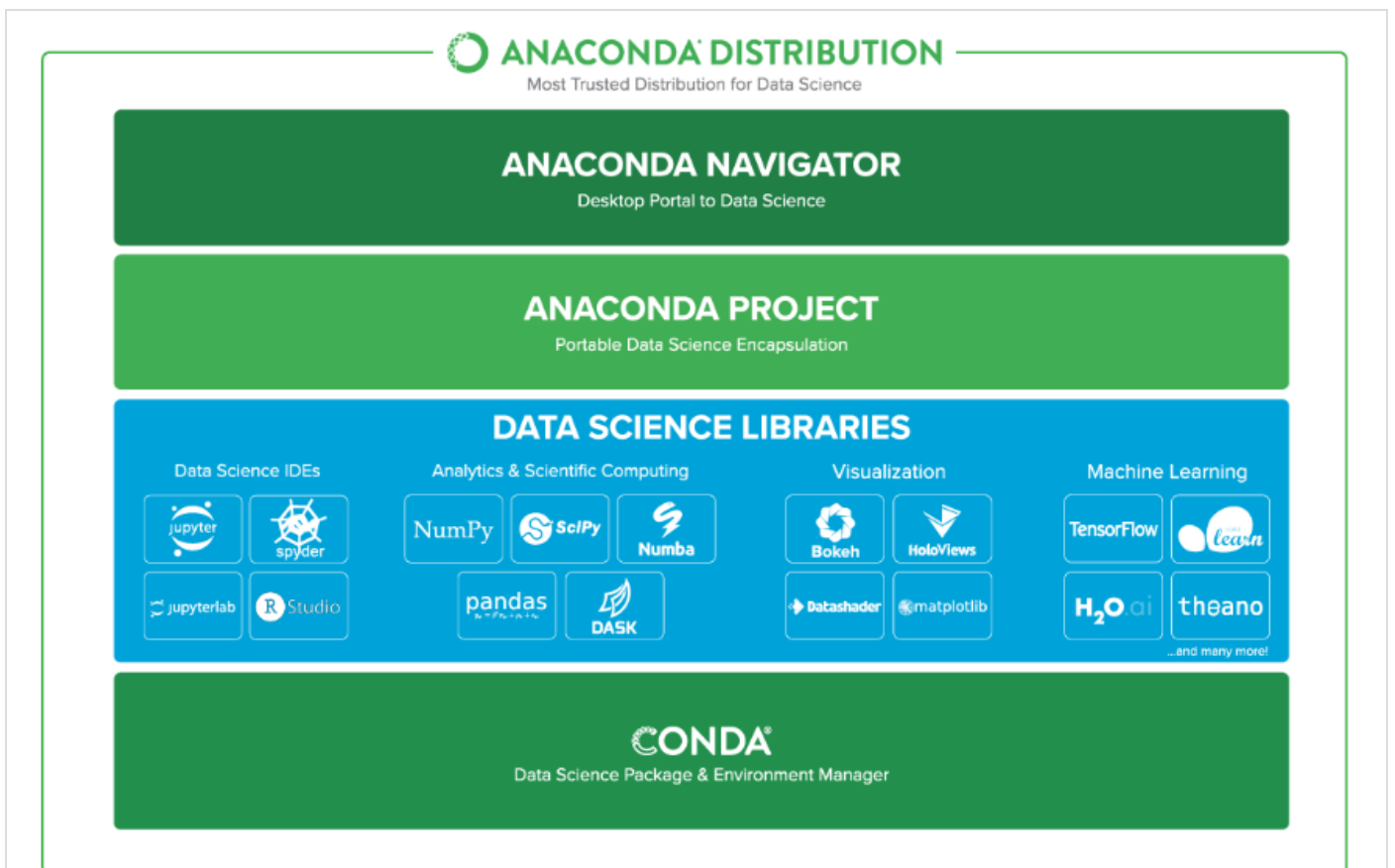


그림 1. 아나콘다의 구성 요소

출처: 예제로 배우는 파이썬 프로그래밍

- 아나콘다의 경량화 배포판인 미니콘다도 존재한다.

- 아나콘다: conda, 파이썬 및 150 종 패키지 등을 포함한 완전 버전 (~500MB)
  - 미니콘다: conda, 파이썬 및 표준 내장 패키지만을 포함한 경량 버전
- 아나콘다에 대한 추가적 자료는 다음을 참고하라.
    - 예제로 배우는 파이썬 프로그래밍
    - Tutorials from docs.anaconda.com

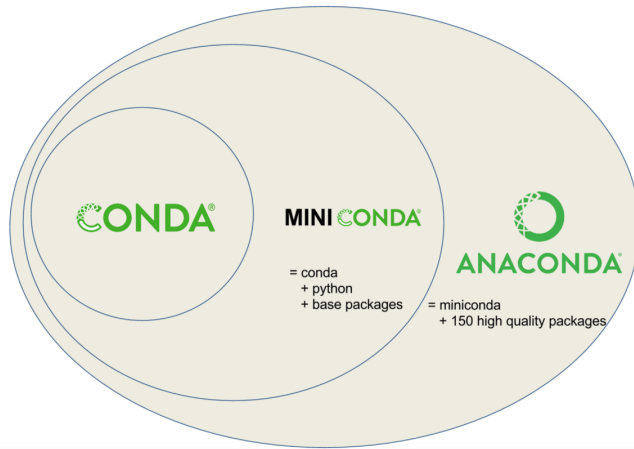


그림 2. miniconda와 anaconda의 차이점

출처: <https://linuxnetmag.com>

- 아나콘다에 대한 소개를 마치고,  
아나콘다 설치를 위한 파이썬 삭제를 진행하자.

## 2. 파이썬 삭제

- 아나콘다는 (일반) 파이썬이 완벽하게 삭제된 상태에서 설치해야 한다.
- 윈도우 **설정 > 앱 > 앱 및 기능** 리스트에서 파이썬 관련 항목을 모두 **제거** 한다.
  - **Python 3.x.x** 항목 제거
  - **Python Launcher** 항목 제거



그림 3. (일반) 파이썬 로고

출처: [python.org](https://python.org)

- (일반) 파이썬은 여러 버전을 함께 설치 가능하므로, 위와 같은 방식으로 모든 버전을 제거한다.
- (일반) 파이썬 삭제에 대하여 더 상세하게 알고 싶다면 다음 자료를 참고하라.
  - 윈도우 및 맥에서 파이썬 삭제 방법을 소개하는 **위키하우의 파이썬 삭제하는 방법**
  - 파이썬 실행 및 파이썬 버전 확인 명령을 실행하여 삭제 여부를 확인하는 방법을 소개하는 **파이썬 삭제 및 제거하기**
  - 파이썬에 관련한 환경변수를 정리하는 법을 소개하는 **환경변수 경로 Path 추가 및 삭제 방법**
- 아나콘다 설치를 위한 파이썬 삭제를 마치고, 본격적으로 아나콘다 설치를 진행하자.

### 3. 아나콘다 설치

- 설치 파일을 다운로드한다.
  - **아나콘다 공식 배포** 사이트에서 적합한 배포판을 다운로드하라.
  - 편역 중인 현 시점(2021년 늦은 여름) 기준으로 Python 3.8 버전을 제공하고 있으며, 운영체제 종류와 비트 버전에 따라서 적절한 설치 파일을 선택하여 다운로드하면 된다.
- 아나콘다를 설치한다.
  - 다운로드된 폴더에서  
**Anaconda3-2021.05-Windows-x86\_64.exe** 파일을 관리자 권한으로 실행
    - 다운로드 선택 사항 및 시점에 따라 파일 이름이 다르다.
    - 설치에 필요한 하드 디스크 최소 공간은 6GB이며, 권장 공간은 20GB이다.
  - 대부분 'I Agree' 또는 'Next' 단추를 클릭하면 되는데, 아래 항목에 대해서는 주의하라.
    - Install for 'Just Me' (recommended)
    - Destination Folder 'C:\Anaconda3' (기본값을 변경하여 여기에 설치해야 편하다.)
    - 'Add Anaconda to my PATH environment variable' (기본값대로 체크 해제한다.)
    - 'Register Anaconda as my default Python 3.8' (기본값대로 체크한다.)
    - Finish
- 아나콘다 설치 버전을 확인한다.
  - 윈도우 시작 메뉴에서 'Anaconda3 (64-bit)' 폴더 및 메뉴 항목을 확인하라.
  - **윈도 시작 메뉴 > Anaconda3 (64-bit)**

**Anaconda Powershell Prompt (anaconda3)** 를 실행하고,  
다음 작업을 수행하라.

```
# 콘다 버전 확인
(base) PS C:\> conda --version
conda 4.10.3
# 파이썬 버전 확인
(base) PS C:\> python --version
Python 3.8.11
```

- 위 버전은 교재 집필 시점 기준이며, 시간이 지나면 버전이 더 올라갈 것이다.

- 아나콘다 설치를 마쳤으니,  
아나콘다 환경 관리를 공부하자.



## 4. 아나콘다 환경 관리

- 아나콘다 사용자들은 가상환경을 애용한다.
- 가상환경은 특정 작업에 필요한 라이브러리 설치/제거를 독립적으로 유지하기 위해 필요하다.
  - 프로젝트마다 사용하는 파이썬 및 라이브러리의 종류와 버전이 다를 수 있다.
  - 이런 상황에서는 프로젝트마다 가상환경을 독립적으로 관리하면 편리하다.
  - 특정 프로젝트에 적합한 라이브러리를 다른 가상환경과는 독립적으로 유지할 수 있다.
- 가상환경을 관리하는 다양한 시나리오에 대처할 수 있어야 한다.
  - 타인에게 자신의 가상환경을 전달하여 복제해야 하는 경우
  - 새로운 컴퓨터에 자신이 사용하던 가상환경을 복제해야 하는 경우
  - 필요없는 가상환경을 삭제해야 하는 경우
  - 새로운 가상환경을 생성해야 되는 경우
  - 특정 가상환경을 최신화 해야 하는 경우
- 아나콘다 자체를 삭제하고 새로 설치해야 하는 경우도 있다.
- **conda** 는 아나콘다의 패키지 관리자인데, 이를 통해 가상환경을 관리(생성, 제거, 활성화, 비활성화)하고, 특정 가상환경의 패키지를 관리(설치, 갱신, 삭제)한다.
- 일반 파이썬에서는 **pyvenv** 나 **virtualenv** 로 가상환경을 관리하고, **pip** 로 패키지를 관리한다.
- 아나콘다에서는 **conda** 로 가상환경 관리와 패키지 관리를 모두 수행한다.
- 아래에 예제로 제시하는 작업들은 **Anaconda Powershell Prompt** 창에서 실행한다.  
**원도 시작 메뉴 > Anaconda3 (64-bit) > Anaconda Powershell Prompt (anaconda3)** 를 실행하고, 다음 예제를 입력하여 실행하라.
- 아래 예제에서 **(base) PS C:\>** 부분은 커맨드 창이 프롬프트이므로 우리가 입력하는 부분이 아니다.

- 여기서 **(base)** 부분은 현재 **base** 라는 이름의 가상환경이 활성화된 상태라는 표시이다.
  - **C:\** 부분은 현행 디렉토리를 표시한다.
- 가상환경의 목록을 확인하고, 새롭게 생성하는 작업은 다음과 같이 수행한다.

```
# 현존하는 가상환경의 목록을 확인한다.
(base) PS C:\> conda env list

# 특정 버전이 지정된 파이썬을 포함하는 my_vnv_1 가상환경을 생성한다.
(base) PS C:\> conda create --name my_vnv_1 python=3.6

# 특정 범위의 버전을 지정한 패키지를 포함하는 my_vnv_2 가상환경을 생성한다.
(base) PS C:\> conda create --name my_vnv_2 "python>3.5,<3.8" "django<3.1"

# 버전 지정을 생략하면 우선 탐색되는 채널에서 해당 버전으로 처리된다.
(base) PS C:\> conda create --name my_vnv_3 python django

# 추가된 가상환경 목록을 다시 확인한다.
(base) PS C:\> conda env list
```

- 가상환경을 활성화 하고, 패키지를 설치/현행화/제거하는 작업은 다음과 같이 수행한다.

```
# 특정 이름을 지정하여 가상환경을 활성화 한다.
# 이후에 수행하는 작업은 모두 활성화된 가상환경에 반영된다.
(base) PS C:\> conda activate my_vnv_3
(my_vnv_3) PS C:\> # 활성화된 가상환경이 프롬프트에 표시된다.

# 예를 들어서, pandas 패키지를 설치한다.
(my_vnv_3) PS C:\> conda install pandas

# 예를 들어서, pandas 패키지를 현행화 한다.
(my_vnv_3) PS C:\> conda update pandas

# 예를 들어서, pandas 패키지를 제거한다.
(my_vnv_3) PS C:\> conda remove pandas

# 현행 가상환경에서 현행화 가능한 패키지를 열거하라.
(my_vnv_3) PS C:\> conda search --outdated

# 현행 가상환경에 설치된 모든 패키지를 현행화 하라.
(my_vnv_3) PS C:\> conda update --all
```

- 가상환경을 비활성화 할 수 있다.

```
# 가상환경을 비활성화 한다.
# 지금부터 수행되는 작업은 모두 기본 가상환경에 반영된다.
(my_vnv_3) PS C:\> conda deactivate
(base) PS C:\> # 자동적으로 기본 가상환경이 활성화된다.
```

- 아래 예제는 **base** 가상환경에서 이루어지므로, 아나콘다 자체를 현행화 한다.

```
(base) PS C:\> conda update conda          # 콘다 자체를 현행화
(base) PS C:\> conda update --all          # 아나콘다 패키지 전체를 현행화
```

- 기존 가상환경을 동결 저장하였다가 다시 복원할 수도 있다.
  - 기존 가상환경에 관한 정보를 **yaml** 파일로 동결 저장할 수 있다.
  - 동결 저장한 **yaml** 파일로부터 가상환경을 복원할 수 있다.

```
(base) PS C:\> conda env list              # 가상환경 목록 확인
(base) PS C:\> conda env export --name my_vnv_2 > my_vnv_2.yaml  # 가상환경 동결(정보 저장)
(base) PS C:\> conda env create --file my_vnv_2.yaml             # 가상환경 복원
```

- (비활성화 상태의) 가상환경을 삭제하는 작업은 다음과 같이 수행한다.

```
# 지정한 이름의 가상환경을 삭제하라.
(base) PS C:\> conda env remove --name my_vnv3
(base) PS C:\> conda env list          # 삭제 후 가상환경 목록 확인
```

- 기존 아나콘다를 제거할 수도 있다.
  - 윈도우 **설정 > 앱 > 앱 및 기능** 리스트에서 기존 아나콘다를 제거한다.
  - 기존 아나콘다가 설치되었던 폴더가 남아있는지 확인 후 삭제한다.
  - 폴더가 남아있을 경우, 같은 이름으로는 가상환경을 다시 설치할 수 없다.
- **conda** 가 패키지를 설치하려면, 온라인 저장소에서 이를 다운로드해야 한다.
  - 패키지를 공급하는 온라인 저장소가 콘다 채널이다.
  - **conda** 를 설치하면 기본 채널 여러 개가 자동적으로 설정되며, 미리 정해진 우선 순위에 따라 채널을 순차적으로 탐색한다.
  - 버전을 특정하지 않은 패키지 설치 과정에서는 채널 탐색 순위가 패키지 버전보다 중요하다. 먼저 탐색되는 채널의 패키지가 버전에 관계없이 설치된다.
  - 콘다 채널의 탐색 순위를 수정하거나 추가할 수 있다.

```
# conda-forge라는 채널을 탐색 순위에 추가
(base) PS C:\> conda config --add channels conda-forge
# 나중에 추가된 채널이 최상위 탐색 순위로 설정됨
(base) PS C:\> conda config --add channels bioconda
```

- 중요한 콘다 명령을 정리하면 다음과 같다.

Conda 명령	작업
<b>conda info</b>	가상환경에 관한 정보를 확인한다.

Conda 명령	작업
<code>conda env list</code>	가용한 가상환경 목록과 폴더 위치를 확인한다.
<code>conda create --name vnv</code>	특정 가상환경 vnv를 생성한다.
<code>conda env remove --name vnv</code>	특정 가상환경 vnv를 삭제한다.
<code>conda activate vnv</code>	특정 가상환경 vnv를 활성화 한다.
<code>conda deactivate</code>	활성화된 현행 가상환경을 비활성화하고, 기본 가상환경을 활성화 한다.
<code>conda list</code>	활성화된 현행 가상환경에 설치되어 있는 패키지를 나열한다.
<code>conda search pkg</code>	특정 패키지 pkg를 패키지 저장소에서 탐색한다.
<code>conda install pkg</code>	활성화된 현행 가상환경에 특정 패키지 pkg를 설치한다.
<code>conda update pkg</code>	현행 가상환경에 설치되어 있는 특정 패키지 pkg를 최신화한다.
<code>conda remove pkg</code>	현행 가상환경에서 특정 패키지 pkg를 삭제한다.
<code>conda config --get channels</code>	설정되어 있는 가용 채널과 탐색 우선순위를 확인한다.
<code>conda config --add channels chnl</code>	특정 채널 chnl을 가용 채널에 추가하고, 최상위 탐색 우선순위를 부여한다.
<code>conda config --remove channels chnl</code>	특정 채널 chnl을 가용 채널에서 삭제한다.

- 아나콘다에서 가상 환경을 관리하는 방법을 마치고,  
주피터 노트북을 공부하자.

## 5. 주피터 노트북 소개



그림 4. 주피터 로고

출처: Benjamin Pryke, [How to Use Jupyter Notebook in 2020: A Beginner's Tutorial](#)

- 주피터 노트북은 데이터 과학에서 사용하는 개발 및 문서화 도구이다.
- 소스 코드와 텍스트를 단일 문서로 작성할 수 있다.
  - 소스 코드와 관련해서는, 작성한 소스 코드와 시각화를 포함하는 실행 결과를 단일 문서로 통합할 수 있다.
  - 텍스트와 관련해서는, 마크다운 문법으로 웹 문서 작성 요소를 간편하게 단일 문서로 통합할 수 있다.
- 오픈소스 **Project Jupyter**의 결과로 무료 사용이 가능하다.
- 아나콘다 **base** 가상환경에는 주피터 노트북이 이미 설치되어 있다.  
다음 명령으로 주피터 노트북을 실행할 수 있다.

```
(base) PS C:\> jupyter notebook      # 기본 가상환경에서 주피터 노트북 실행
```

- 그러나 **base** 가상환경이 아닌, 자신만의 가상환경을 만들어서 작업하기를 권한다.
- 신규 가상환경에는 주피터 노트북을 별도로 설치해야 한다.
- 여기서 생성하는 **py4ds** 가상환경은 앞으로 우리 수업에서 계속 활용한다.

```
# 새 가상환경 생성 및 활성화  
(base) PS C:\> conda create --name py4ds python      # 파이썬이 설치된 가상환경 py4ds 생성
```

(base) PS C:\> conda activate py4ds	# 가상환경 py4ds 활성화
(py4ds) PS C:\> jupyter notebook	# 패키지 미설치 상태이므로 오류 발생
# 새 가상환경에 주피터 설치 및 노트북 실행	
(py4ds) PS C:\> conda install jupyter	# 가상환경 py4ds 내부에 주피터 패키지 설치
(py4ds) PS C:\> jupyter notebook	# 패키지 설치 상태이므로 정상 실행됨
...	# 실행된 주피터 노트북에서 작업 수행

- 원하는 작업 폴더로 디렉토리를 변경한 후에 주피터 노트북을 실행하면 편하다.

```

# 보통 사용자 폴더에서 셸이 시작됨
(base) PS C:\Users\logis>
# 우선 작업 폴더로 디렉토리 변경
# 작업 폴더 `C:\py4ds`가 이미 생성되어 있다고 가정함
(base) PS C:\Users\logis> cd C:\py4ds
# 작업 폴더에서 가상환경 활성화
(base) PS C:\py4ds> conda activate py4ds
# 가상환경이 활성화된 작업 폴더에서 노트북 실행
(py4ds) PS C:\py4ds> jupyter notebook
# 노트북 환경에서 필요한 작업을 수행
...

```

- 주피터 노트북에 대한 소개를 마치고,  
본격적으로 주피터 노트북 활용 방법을 공부하자.

## 6. 주피터 노트북 활용

- 주피터 노트북을 실행하면, 웹 브라우저에 새로운 탭이 열리고 노트북 대시보드가 나타난다.
  - 주피터 노트북은 웹 브라우저에서 동작하는 웹 앱이다.
  - **http://localhost:8888/tree** URL을 통하여 노트북 대시보드가 제공된다.



- 노트북 대시보드 화면에서는 다음과 같은 메뉴 항목과 버튼을 제공한다.
  - **Files** 탭에서는 폴더 및 파일 목록을 확인할 수 있다.
  - **Running** 탭에서는 실행 중인 터미널 및 노트북을 확인할 수 있다.
  - **Clusters** 탭에서는 'IPython parallel' 기능과 관련한 내용을 확인할 수 있다.
  - **Nbextensions** 탭에서는 노트북 확장 기능과 관련한 내용을 확인/설정 할 수 있다.
  - **Clusters** 및 **Nbextensions** 은 별도로 추가 기능을 설치해야 이용할 수 있다.
  - **Quit** 버튼으로 모든 노트북 작업을 종료한다.
- 새 노트북을 작성하려면,  
Files 탭에서 원하는 폴더로 이동한 상태에서,  
화면 우측 상단에 있는 **New** 드롭다운 버튼 > **Python 3** 메뉴 항목을 클릭한다.

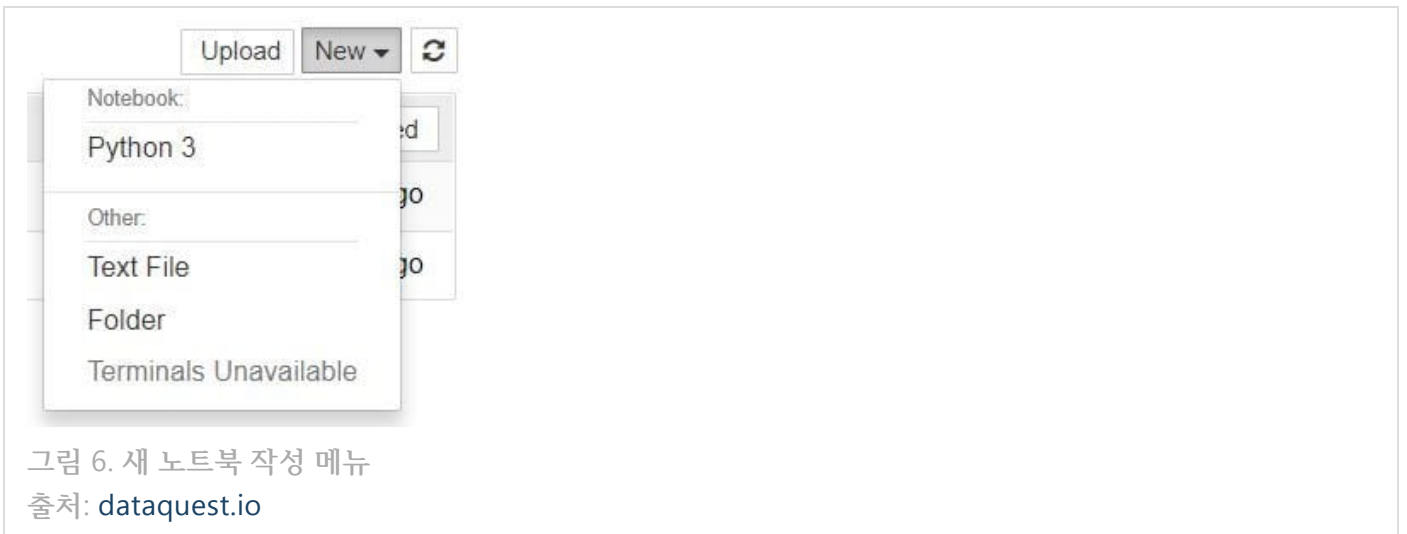


그림 6. 새 노트북 작성 메뉴

출처: dataquest.io

- 이렇게 하면, 웹 브라우저에 새로운 탭이 열리면서 **Untitled** 노트북이 생성되고, 빈 코드 셀이 나타난다.

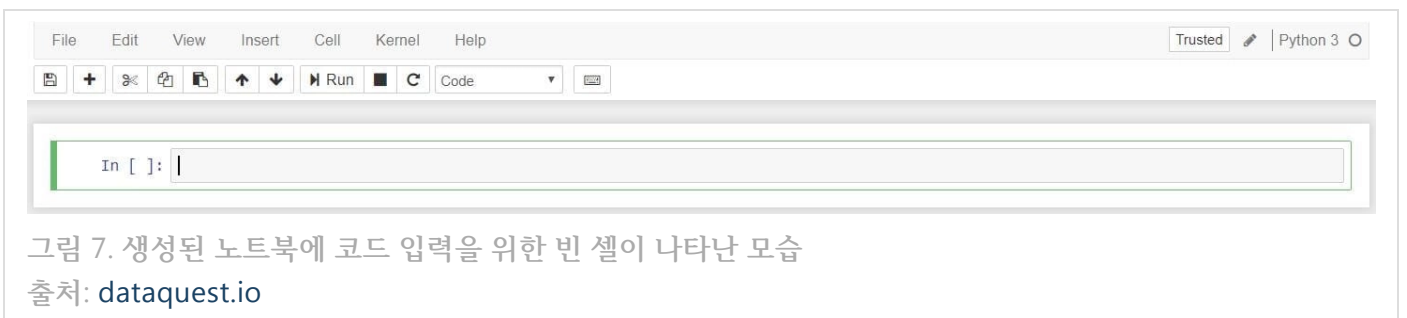


그림 7. 생성된 노트북에 코드 입력을 위한 빈 셀이 나타난 모습

출처: dataquest.io

- 다시 대시보드 탭으로 전환하여, **Untitled.ipynb** 파일이 생성되었으며, 초록색 아이콘으로 표시됨을 확인하라.
- 노트북 파일은 확장자가 **.ipynb** 로 지정된다.
- **.ipynb** 파일은 노트북의 내용을 JSON 형식으로 저장한 텍스트 파일이다.
- 진보적인 문서 편집기라고도 볼 수 있는 주피터 노트북에서는 다음 사항을 단일 문서에 통합적으로 작성할 수 있다.
  - 파이썬 소스 코드를 작성하고 문서 내에서 실행할 수 있다.
  - 파이썬 코드 실행 결과를 문서에 함께 보여줄 수 있다.
  - 마크다운 문법으로 웹 문서의 거의 모든 요소를 (HTML보다 훨씬 간편하게) 작성할 수 있다.
  - 마크업(markup)이 아니라 마크다운(markdown)이라는 표현을 사용하고 있다.
- 주피터 노트북을 활용하려면 커널과 셀의 개념을 이해해야 한다.
  - **\*\*커널(kernel)\*\***은 노트북 문서에 작성된 코드를 실행하기 위한 "연산 엔진"이다.
  - **코드 셀**은 커널이 실행할 파이썬 코드를 입력/저장하는 공간이다.
  - **마크다운 셀**은 노트북에 보여줄 마크다운 텍스트를 입력/저장하는 공간이다.



- 코드 셀은 파이썬 코드를 작성하는 셀이다.
  - 여기에 작성된 코드는 커널이 실행해 준다.
  - 코드 셀의 코드가 실행된 결과는 코드 셀 바로 밑에 출력된다.
  - 코드 셀에는 셀 왼쪽에 **In [1]** 과 같은 셀 번호가 표시되는데,
    - "In"은 "Input"이란 의미이다.
    - "□" 내부의 번호는 코드 셀이 실행된 순서를 표시하며, 해당 코드 셀이 실행될 때마다 자동적으로 증가한다.
- 마크다운 셀은 마크다운 문법으로 형식화된 텍스트를 작성하는 셀이다.
  - 마크다운 셀이 실행되면 해당 셀에 마크다운 텍스트의 실행 결과가 출력된다.
  - 마크다운 문법을 (심하게) 간략화된 **HTML** 문법이라고 생각해도 좋다.
  - **HTML**로 작성한 웹 문서가 브라우저에 표시되듯이, 마크다운 문법으로 작성한 텍스트도 마크다운 셀을 실행하면 브라우저에서 (렌더링되듯이) 표시된다.
- 새로 만들어진 **Untitled.ipynb** 파일이 열려 있는 탭에서 노트북의 사용법을 연습하자.
- 제일 먼저 코드 셀과 마크다운 셀을 구분할 줄 알아야 한다.
  - <그림 7>과 같이 노트북이 열려있는 탭에는, 상단 메뉴 바로 밑에 아이콘들이 보이는데, 이 중에서 **Code** 라고 표시된 드롭다운 리스트를 열면 **Code** 외에도 **Markdown** 등의 항목이 준비되어 있다.
  - 이 드롭다운 리스트에서 어느 항목을 선택하느냐에 따라서 현재 선택되어 있는 셀의 종류를 코드 셀이나 마크다운 셀로 변경하여 지정할 수 있다.
  - <그림 7>에 보이는 첫 셀을 코드 셀에서 마크다운 셀로, 다시 반대로 변경하면서 무엇이 바뀌는지 살펴보자.
    - 코드 셀로 지정되면, 셀 왼쪽에 **In [ ]: ▶** 표시가 나오는데, 코드 실행과 관련한 표시이다.
    - 마크다운 셀로 지정되면, **In [ ]: ▶** 표시가 나오지 않는다.
- 첫 셀을 코드 셀로 지정한 상태에서 코드를 작성하고 실행하여 보자.
  - 코드 셀 내부에 **print('Hello')** 라고 코드를 입력하라.
  - 코드 셀을 실행하는 방법이 다양하게 준비되어 있는데, 다음 세 방법 중의 하나를 사용하라.
    1. ▶ 기호를 마우스로 클릭한다

2. **Ctrl+Enter** 키를 키보드에서 입력한다.

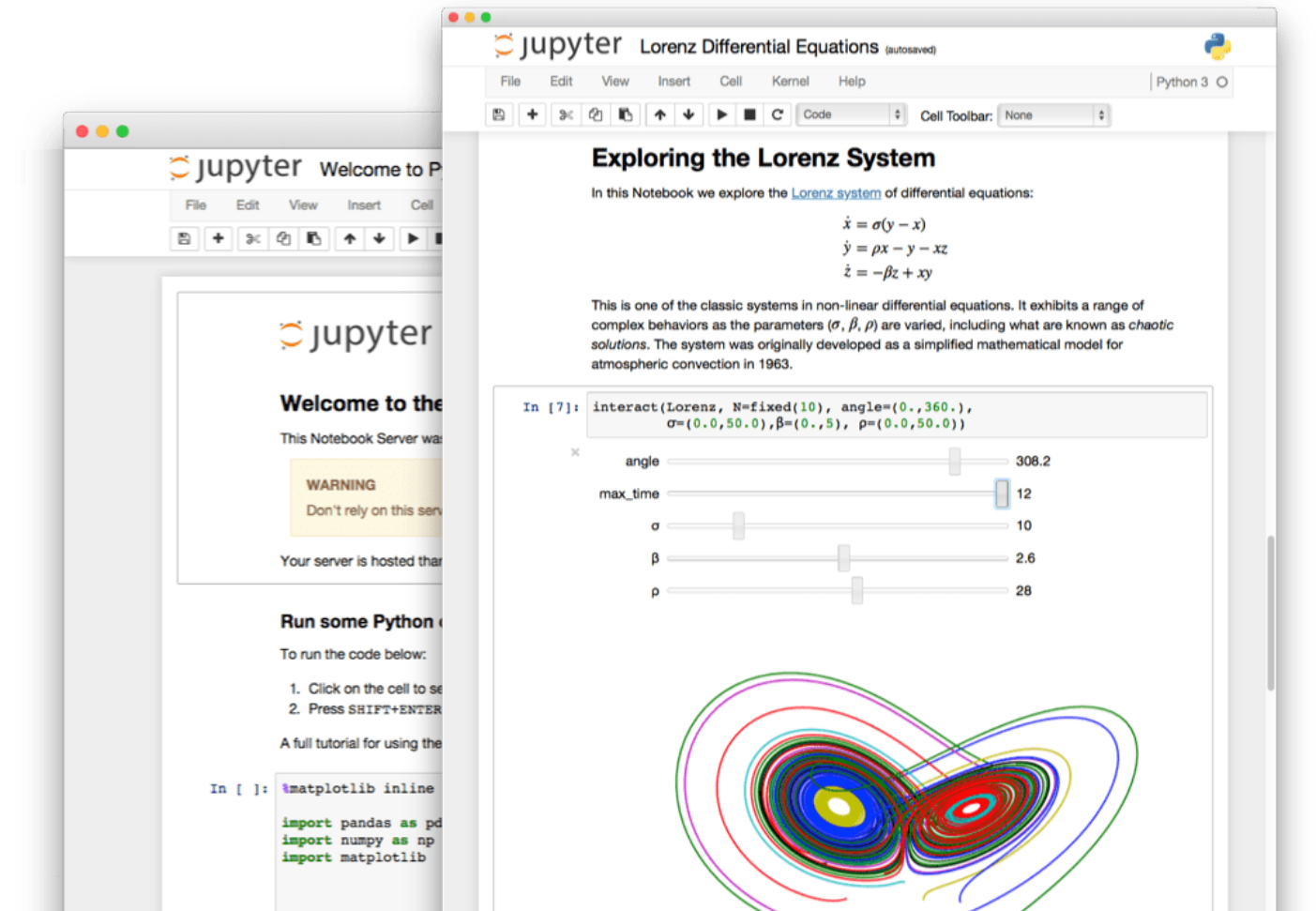
3.  아이콘을 마우스로 클릭한다.

- 코드가 실행되면 실행된 결과가 코드 셀 바로 밑에 출력된다.
  - **In [ ]: ▶** 표시가 **In [1]: ▶** 표시로 변경되면서 해당 셀의 실행 순서가 표시된다.
  - 방금 코드를 실행하는 방법은 (포커스가 주어진) 현행 코드 셀만 실행하라는 명령이다.
  - 같은 코드 셀을 여러번 반복 실행하여, 이 숫자가 증가하는 것을 확인하라.
  - 축하! 주피터 노트북에서 첫 코드를 멋지게 작성하였다.
- 이번에는 마크다운 셀을 작성해보자.
    - 현재 노트북에는 코드 셀 하나만 있는 상태이다.
    - 노트북 메뉴 중에서 **Insert > Insert Cell Below** 항목을 클릭하여 새로운 셀을 생성하라.
    - 새로 생성된 셀은 기본적으로 코드 셀로 지정된다.
    - 이를 마크다운 셀로 변경하라.
    - 변경된 마크다운 셀에 다음 두 줄을 복사하여 붙여넣고, 실행하라.

[주피터 공식 웹 사이트](<https://jupyter.org/>)에서 퍼온 그림입니다.  
![노트북 화면 예시](<https://jupyter.org/assets/jupyterpreview.png>)

- 마크다운 셀을 실행하는 방법도 코드 셀과 유사하지만, 방법 1은 쓸 수 없다.
  - 마크다운 셀이 실행되면 마법과 같은 일이 벌어진다.
- 여러분이 보는 결과는 아래와 같다
    - 실행 결과 첫 줄에 보이는 링크를 클릭해 보라.
    - 실행 결과 그림에 보이는 노트북 예시 화면을 보라.

주피터 공식 웹 사이트에서 퍼온 그림입니다.



- 위에서 우리는 단 두 줄의 마크다운 명령으로 놀라운 결과를 만들어 냈다.
  - 첫째 줄에서  
[주피터 공식 웹 사이트](https://jupyter.org/) 부분은 HTML 하이퍼 링크에 해당한다.  
[링크에 보여줄 텍스트](링크\_웹\_주소) 형식으로 입력한다.
  - 둘째 줄에서  
![노트북 화면 예시](https://jupyter.org/assets/jupyterpreview.png) 부분은 HTML 이미지 요소에 해당한다.  
![대체 속성](이미지\_경로) 형식으로 입력한다.
- 마크다운 문법을 익히면 다양한 웹 문서 요소를 간편하게 작성할 수 있다.
  1. 다양한 제목을 만들 수 있다.

## 문서 제목

## 문서 부제목

# This is a H1

---

## This is a H2

### This is a H3

#### This is a H4

##### This is a H5

###### This is a H6

##### H7은 없다.

- **Untiled.ipynb** 노트북의 새 마크다운 셀에 다음과 같이 입력해보라.  
마크다운 셀을 실행하여 바로 위에 보이는 모습과 같이 렌더링되는지 확인하라.

```
문서 제목
=====
```

```
문서 부제목
-----
```

```
# This is a H1
## This is a H2
### This is a H3
#### This is a H4
##### This is a H5
##### This is a H6

##### H7은 없다.
```

2. 다양한 목록을 만들 수 있다.

### 순서 있는 목록

1. 데이터 분석
2. 데이터 시각화
3. 빅데이터 처리

## 4. 기계학습

### 순서 없는 목록

- 프로그래밍 언어
  - 파이썬
    - 아나콘다
    - 미니콘다
  - 자바
  - C
- **Untiled.ipynb** 노트북의 새 마크다운 셀에 다음과 같이 입력해보라.  
마크다운 셀을 실행하여 바로 위에 보이는 모습과 같이 렌더링되는지 확인하라.

<u>순서 있는 목록</u>

1. 데이터 분석
1. 데이터 시각화
1. 빅데이터 처리
1. 기계학습

<u>순서 없는 목록</u>

- 프로그래밍 언어
  - 파이썬
    - 아나콘다
    - 미니콘다
  - 자바
  - C

- 첫째 목록은 번호 있는 목록이고, 둘째 목록은 번호 없는 목록이다.
  - 번호 있는 목록을 작성할 때, 번호를 모두 **1.** 으로 지정했지만, 제대로 번호가 증가하면서 표시되었다는 점도 주목하라.
  - 밑줄이 그어진 텍스트를 표시하기 위해서 HTML 태그를 이용했다.
3. 다양한 텍스트 효과를 연출할 수 있다.

single asterisks

single underscores

double asterisks

double underscores

cancelline

- 아래와 같이 단어 앞뒤에 특수 문자를 붙여서 기울임체, 굵은체 등의 효과를 지정할 수 있다.

- 새 마크다운 셀에 아래 내용을 복사하여 붙여넣기 하면, 한 줄로 길게 입력된다.
- 위 결과와 같게 만들려면, 매 행마다 행의 끝에 공백 문자를 두개 이상 입력해야 한다.

```
*single asterisks*
_single underscores_
**double asterisks**
__double underscores__
~~cancelline~~
```

#### 4. 줄바꿈과 문단 나누기를 할 수 있다.

- 텍스트에서 강제 줄바꿈을 할 수 있다.  
앞서 텍스트 효과에 대한 예제에서는, 행마다 행의 끝에 공백 문자를 2개씩 입력해 두었다.  
이렇게 하면 강제 줄바꿈이 적용된다.
- 강제 줄바꿈이 아니라 정상적인 문단 나누기를 할 수 있다.  
문단을 나누려면, 문단과 문단 사이에 빈 행을 입력한다.

#### 5. 링크와 이미지를 삽입하는 방법은 앞에서 예시한 바 있다.

#### 6. 이들 요소 외에도, 표(table), 블록인용(blockquote), 코드 블록 및 수평선 등을 위한 마크다운 문법을 공부해야 한다.

- 한글로 작성된 **마크다운 사용법**을 참고하라.
- 영문으로 작성된 **Markdown By John Gruber**는 마크다운을 처음 만든 사람이 직접 작성한 안내 자료이다.

#### 7. 지금까지 작업한 **Untitled.ipynb** 노트북 파일을 **노트북 연습.ipynb** 파일로 저장한다.

- 노트북 메뉴에서 **File > Save as...** 항목을 클릭하고, 확장자를 제외하고 "노트북 연습"이라고 입력하고 **Save** 단추를 누른다.
- 웹 브라우저에서 주피터 노트북 대시보드 탭을 확인하면, 방금 지정한 이름으로 새로운 노트북이 저장되어 있을 것이다.

#### 8. 주피터 노트북 대시보드 탭에서 우측 상단에 보이는 **Quit** 버튼을 클릭하면, **Anaconda Powershell Prompt** 창에서 실행 중이던 **jupyter notebook** 이 종료되어 있을 것이다.

- 본 장에서는 파이썬 프로그래밍을 위한 실습 환경을 구축하였다.
  - 아나콘다와 주피터 노트북에 대하여 공부하였다.
  - 아나콘다와 주피터 노트북을 설치하였다.
  - 주피터 노트북에서 코드와 텍스트가 통합된 문서 작성 방법을 공부하였다.
- 다음 장에서는 본격적인 데이터 분석 공부의 첫 순서로 넘파이(NumPy) 공부를 시작해 보자.