

데이터분석 PANDAS



WEB 개발도구

◆ 아나콘다(Anaconda)

- 수 백 개의 파이썬 패키지를 포함하고 있는 파이썬 배포판
- 패키지를 별도로 시간 들여 추가 설치하는 수고 감소
- 기능
 - 패키지 설치 및 관리
 - 가상환경 관리
- 관리 프로그램
 - **conda** 제공



◆ 아나콘다(Anaconda)

■ conda 프로그램

→ 자동 의존성 체크 후 설치

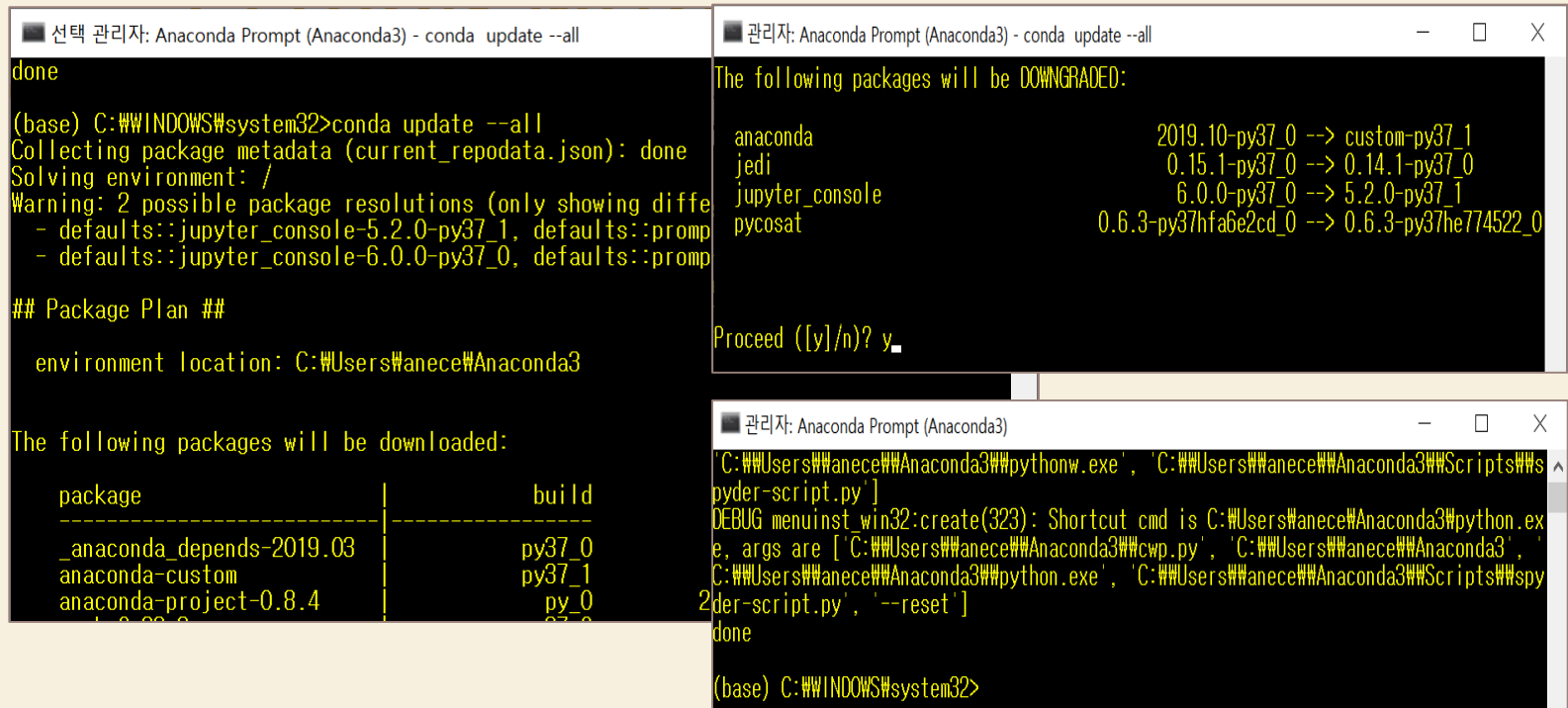
→ 모든 라이브러리 없으므로 pip와 병행 사용

기능	명령어 옵션 커맨드
패키지 설치	\$ conda install numpy # numpy 설치
동시에 여러 패키지 설치	\$ conda install numpy scipy pandas # numpy, scipy, pandas 동시 설치
특정 버전 설치	\$ conda install numpy=1.10 # 특정 버전 설치
패키지 제거	\$ conda remove package_name
패키지 업데이트	\$ conda update package_name
모든 패키지 업데이트	\$ conda update -all
설치된 목록 출력	\$ conda list # 정보 확인
설치하려는 패키지 검색	\$ conda search search_term

◆ 아나콘다(Anaconda)

■ 업데이트

(base) C:\Users\사용자 > conda update --all



The first screenshot shows the command prompt with the command `conda update --all` being executed. It displays the progress of collecting metadata and solving the environment, followed by a package plan showing the packages to be downloaded.

```
done

(base) C:\WINDOWS\system32>conda update --all
Collecting package metadata (current_repodata.json): done
Solving environment: /
Warning: 2 possible package resolutions (only showing differ
- defaults::jupyter_console-5.2.0-py37_1, defaults::promp
- defaults::jupyter_console-6.0.0-py37_0, defaults::promp

## Package Plan ##

environment location: C:\Users\anece\Anaconda3

The following packages will be downloaded:
```

package	build
_anaconda_depends-2019.03	py37_0
anaconda-custom	py37_1
anaconda-project-0.8.4	py_0

The second screenshot shows the list of packages to be downgraded:

```
The following packages will be DOWNGRADED:

anaconda                2019.10-py37_0 --> custom-py37_1
jedi                    0.15.1-py37_0 --> 0.14.1-py37_0
jupyter_console         6.0.0-py37_0 --> 5.2.0-py37_1
pycosat                 0.6.3-py37hfa6e2cd_0 --> 0.6.3-py37he774522_0

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

The third screenshot shows the final output of the update process, including the installation of the new Anaconda version and the creation of a shortcut for the JupyterLab application.

```
'C:\Users\anece\Anaconda3\python.exe', 'C:\Users\anece\Anaconda3\Scripts\py
pyder-script.py']
DEBUG menuinst.win32:create(323): Shortcut cmd is C:\Users\anece\Anaconda3\python.ex
e, args are ['C:\Users\anece\Anaconda3\python.exe', 'C:\Users\anece\Anaconda3\
C:\Users\anece\Anaconda3\python.exe', 'C:\Users\anece\Anaconda3\Scripts\spy
der-script.py', '--reset']
done

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

◆ 아나콘다(Anaconda)

■ 가상환경 생성

`conda create -n 가상환경이름 python=파이썬버전`

`(base) C:\Users\사용자>conda create -n EN_PY375 python=3.7`

◆ 아나콘다(Anaconda)

■ 가상환경 확인

```
(base) C:\Users\사용자 > conda env list
```

```
(base) C:\Users\anece>conda env list
# conda environments:
#
base                * C:\Users\anece\Anaconda3
EV_PY375            C:\Users\anece\Anaconda3\envs\EV_PY375

(base) C:\Users\anece>
```

◆ 아나콘다(Anaconda)

■ 채널 설정

defaults(기본 conda 채널)과 conda-forge를 섞어 쓰면 패키지 충돌

[설정]

```
conda config --add channels conda-forge
```

```
conda config --set channel_priority strict
```

[확인]

```
conda config  show
```

```
conda info
```


◆ 아나콘다(Anaconda)

■ 가상환경 실행 & 종료

실행 : `conda activate` 가상환경이름

종료: `conda deactivate`

```
(base) C:\Users\#anece>conda activate EV_PY375
```

```
(EV_PY375) C:\Users\#anece>conda deactivate
```

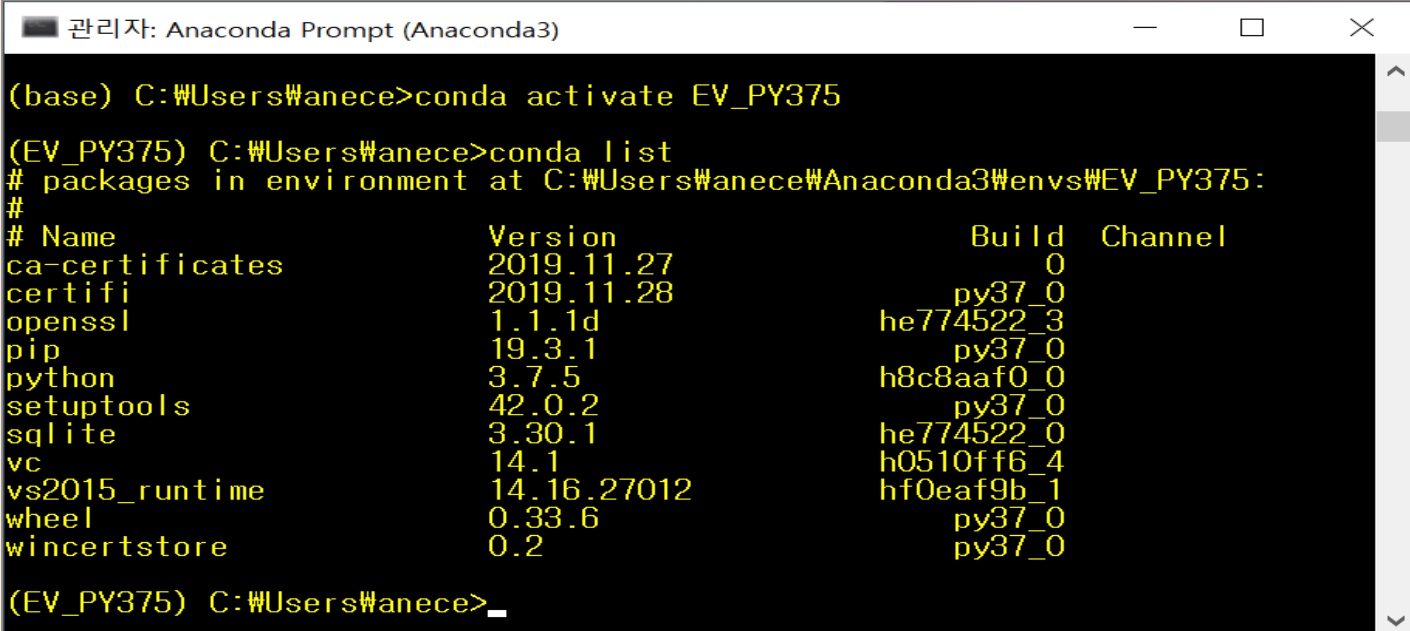
```
(base) C:\Users\#anece>
```

◆ 아나콘다(Anaconda)

■ 가상환경 설치 모듈 체크

설치 모듈 정보 출력 : `conda list`

모듈 설치 명령어 : `conda install` 라이브러리명



```
관리자: Anaconda Prompt (Anaconda3)

(base) C:\Users\Wanece>conda activate EV_PY375

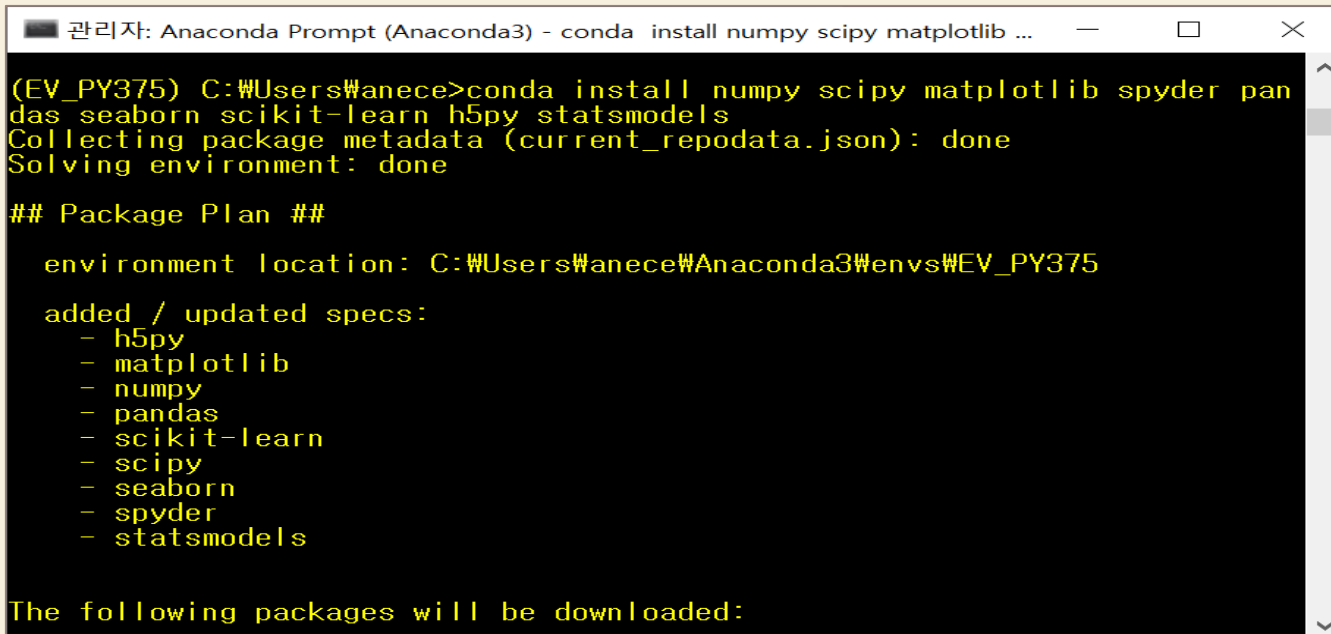
(EV_PY375) C:\Users\Wanece>conda list
# packages in environment at C:\Users\Wanece\Anaconda3\envs\EV_PY375:
#
# Name                                Version                                Build      Channel
ca-certificates                       2019.11.27                             0
certifi                               2019.11.28                             py37_0
openssl                               1.1.1d                                 he774522_3
pip                                    19.3.1                                 py37_0
python                                 3.7.5                                 h8c8aaf0_0
setuptools                             42.0.2                                 py37_0
sqlite                                 3.30.1                                 he774522_0
vc                                     14.1                                  h0510ff6_4
vs2015_runtime                        14.16.27012                           hf0eaf9b_1
wheel                                  0.33.6                                 py37_0
wincertstore                           0.2                                  py37_0

(EV_PY375) C:\Users\Wanece>_
```

◆ 아나콘다(Anaconda)

■ 가상환경 설치 모듈 설치

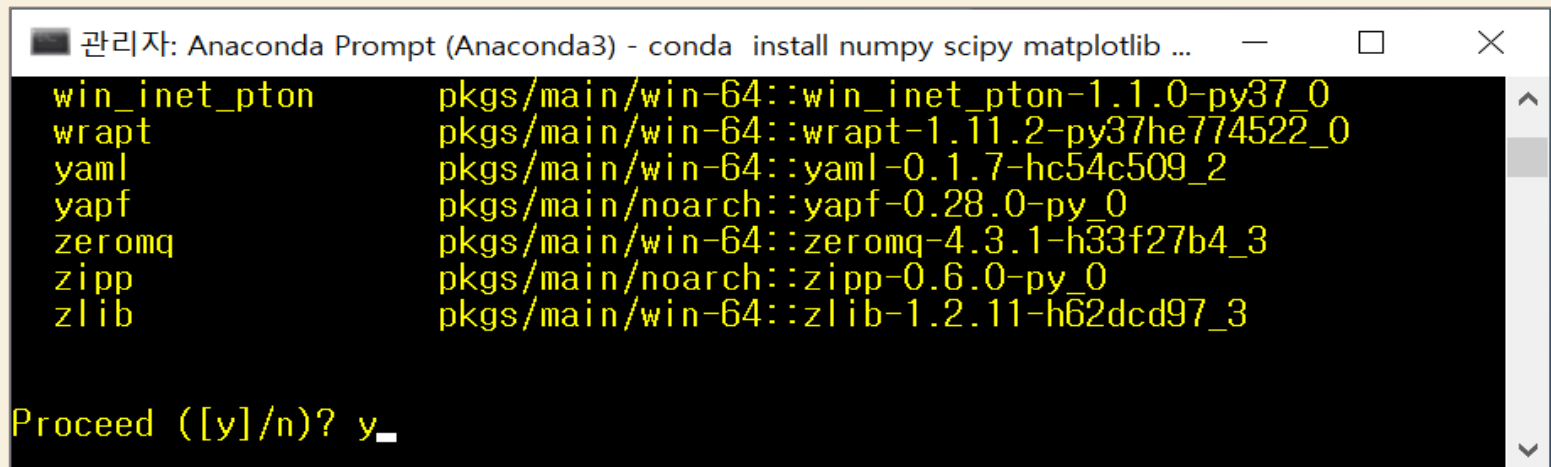
```
conda install numpy scipy matplotlib  
conda install pandas seaborn scikit-learn
```



```
관리자: Anaconda Prompt (Anaconda3) - conda install numpy scipy matplotlib ...  
(EV_PY375) C:\Users\Wanece>conda install numpy scipy matplotlib spyder pandas  
seaborn scikit-learn h5py statsmodels  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
  
## Package Plan ##  
  
environment location: C:\Users\Wanece\Anaconda3\envs\EV_PY375  
  
added / updated specs:  
- h5py  
- matplotlib  
- numpy  
- pandas  
- scikit-learn  
- scipy  
- seaborn  
- spyder  
- statsmodels  
  
The following packages will be downloaded:
```

◆ 아나콘다(Anaconda)

■ 가상환경 설치 모듈 설치



```
관리자: Anaconda Prompt (Anaconda3) - conda install numpy scipy matplotlib ...  
win_inet_pton      pkgs/main/win-64::win_inet_pton-1.1.0-py37_0  
wrapt              pkgs/main/win-64::wrapt-1.11.2-py37he774522_0  
yaml               pkgs/main/win-64::yaml-0.1.7-hc54c509_2  
yapf               pkgs/main/noarch::yapf-0.28.0-py_0  
zeromq             pkgs/main/win-64::zeromq-4.3.1-h33f27b4_3  
zipp               pkgs/main/noarch::zipp-0.6.0-py_0  
zlib               pkgs/main/win-64::zlib-1.2.11-h62dcd97_3  
  
Proceed ([y]/n)? y_
```

◆ Jupyter Notebook

- 프로젝트 주피터의 제품 중 하나
→ "오픈 소스 소프트웨어, 개방형 표준, 그리고 여러 개의 프로그래밍 언어에 걸쳐 **인터랙티브 컴퓨팅**을 위한 서비스 개발"을 위해 설립된 비영리 단체
→ 2014년에 IPython으로부터 파생

- 여러 개의 언어를 통한 실행 환경 지원
- 인터랙티브 컴퓨팅 제품인 주피터 노트북, 주피터허브, 주피터랩 개발 및 지원



◆ Jupyter Notebook

■ 실행

방법1) 시작 > Anaconda3 (64-bit) > Jupyter Notebook

방법2) C:\Users\<사용자>\Anaconda3\python.exe -m notebook

◆ Jupyter Notebook

■ 실행

```
Jupyter Notebook (Anaconda3)
[W 2021-08-11 05:22:49.458 LabApp] 'notebook_dir' has moved from NotebookApp to ServerApp. This config
to ServerApp. Be sure to update your config before our next release.
[I 2021-08-11 05:22:49.471 LabApp] JupyterLab extension loaded from C:\ProgramData\Anaconda3\lib\site-
rlab
[I 2021-08-11 05:22:49.472 LabApp] JupyterLab application directory is C:\ProgramData\Anaconda3\share\
[I 05:22:49.480 NotebookApp] Serving notebooks from local directory: C:\Users\anece
[I 05:22:49.480 NotebookApp] Jupyter Notebook 6.3.0 is running at:
[I 05:22:49.480 NotebookApp] http://localhost:8888/?token=dbc7c9299660ddb75ce072ca190d6b993b0762a45d8e927a
[I 05:22:49.481 NotebookApp] or http://127.0.0.1:8888/?token=dbc7c9299660ddb75ce072ca190d6b993b0762a45d8e927a
[I 05:22:49.481 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to sk
[C 05:22:49.632 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/anece/AppData/Roaming/jupyter/runtime/nbserver-27228-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=dbc7c9299660ddb75ce072ca190d6b993b0762a45d8e927a
or http://127.0.0.1:8888/?token=dbc7c9299660ddb75ce072ca190d6b993b0762a45d8e927a
```

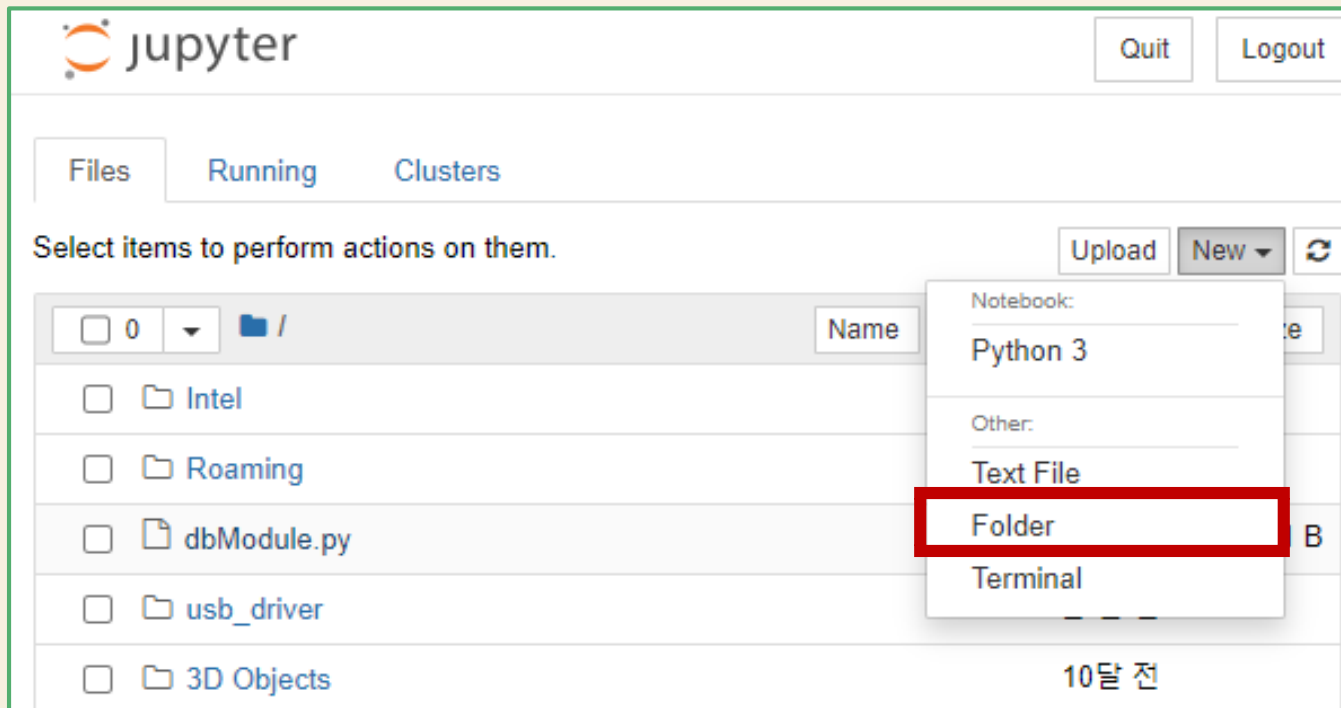
◆ Jupyter Notebook

- 실행 → 브라우저 연동



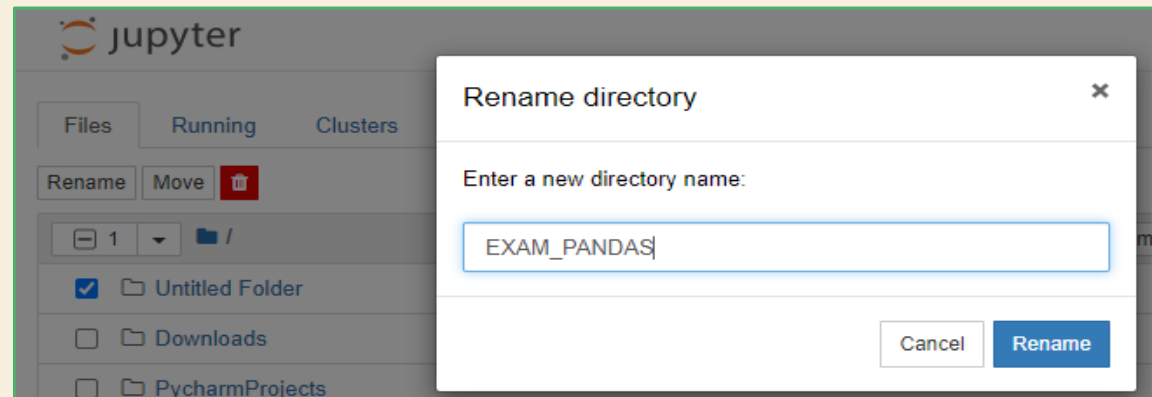
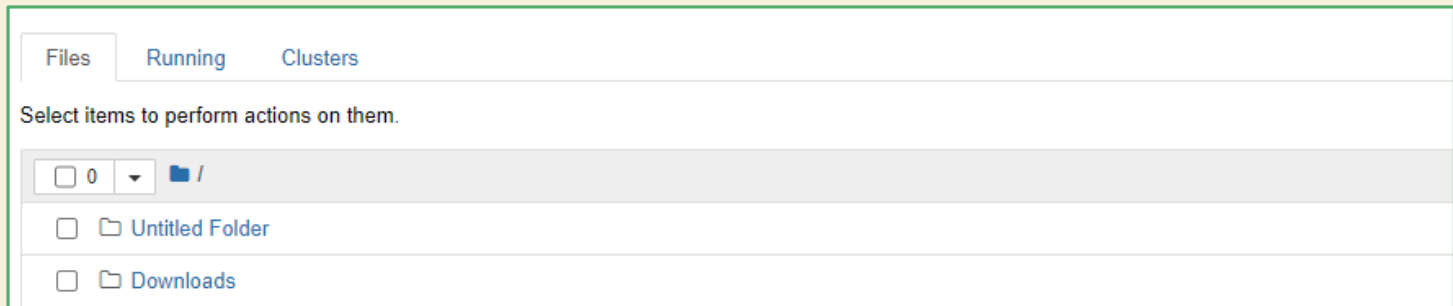
◆ Jupyter Notebook

■ 작업 폴더 생성



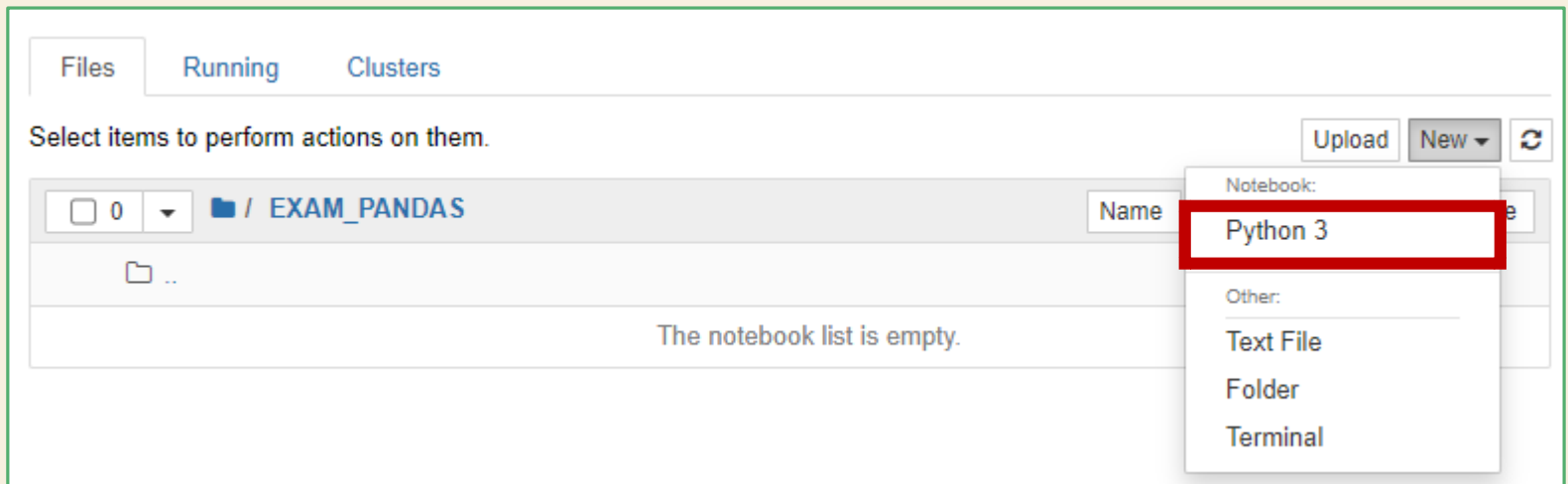
◆ Jupyter Notebook

■ 작업 폴더 생성



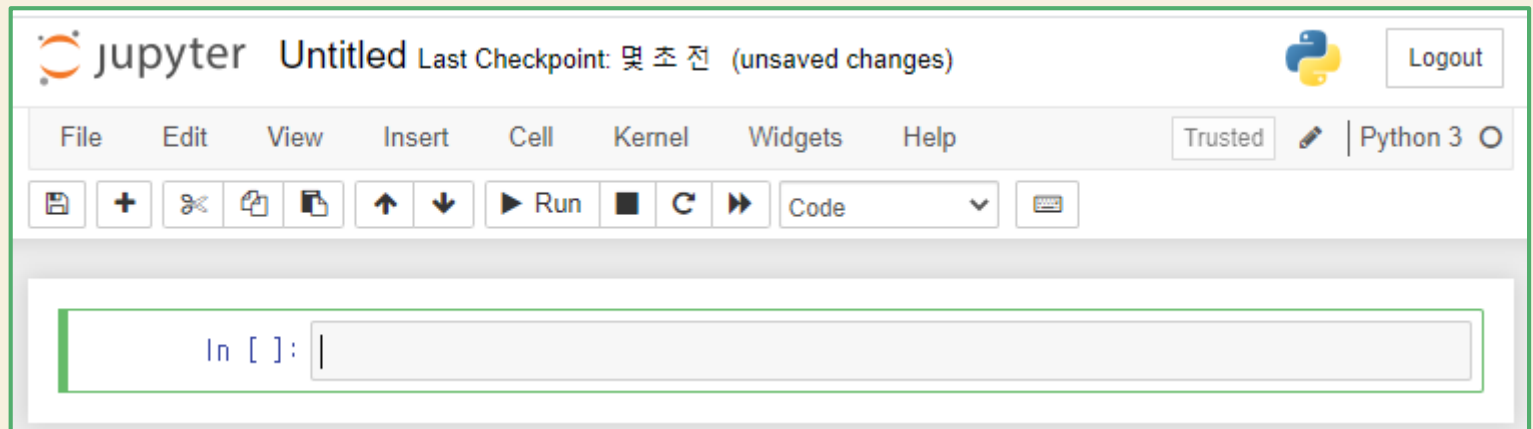
◆ Jupyter Notebook

■ 파일 생성



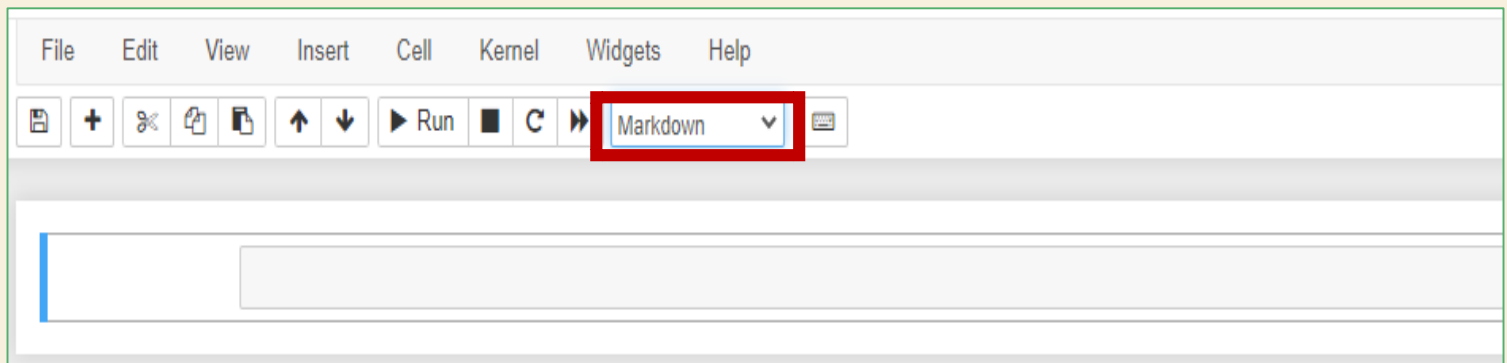
◆ Jupyter Notebook

■ 파일 화면



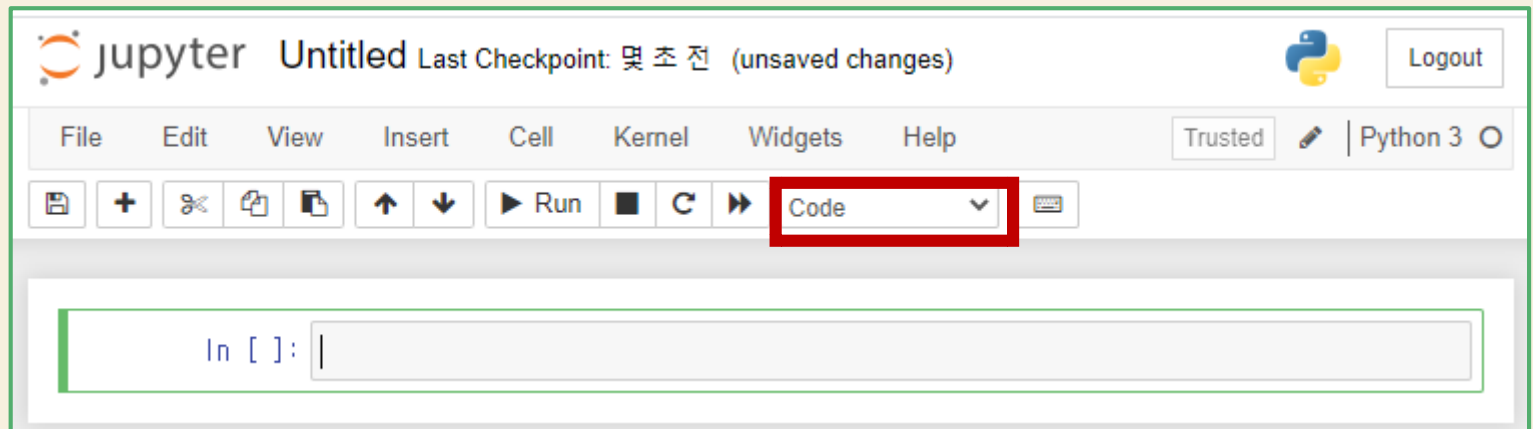
◆ Jupyter Notebook

■ 파일 - 설명 작성



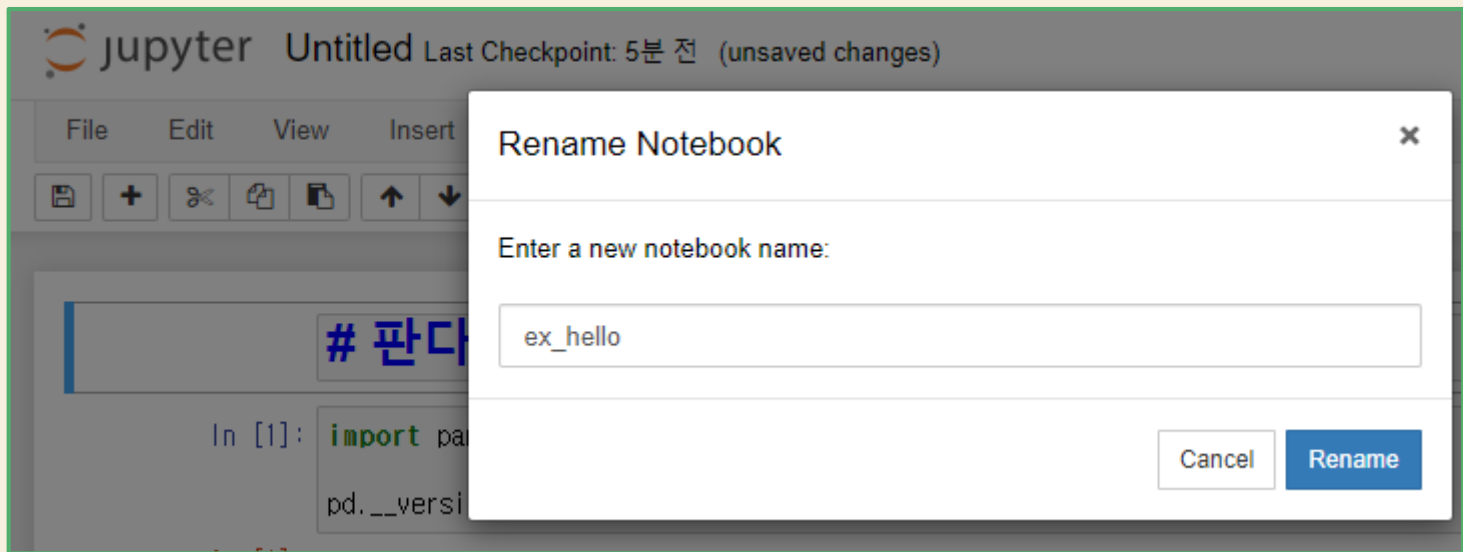
◆ Jupyter Notebook

■ 파일 - 코드 작성



◆ Jupyter Notebook

- 파일 - 파일명 변경



◆ Jupyter Notebook

■ 동작 모드

- Edit 모드

A screenshot of a Jupyter Notebook cell in Edit mode. The cell has a light blue border and a light blue background. The text 'In []:' is visible in the top left corner of the cell, followed by a vertical cursor line.

- 셀을 클릭하거나 키보드에서 **Enter키** 입력
- 셀 테두리 **녹색**으로 전환
- **셀 입력 가능**

- Command 모드

A screenshot of a Jupyter Notebook cell in Command mode. The cell has a light blue border and a light blue background. The text 'In []:' is visible in the top left corner of the cell, followed by a vertical cursor line.

- 셀 외부의 아무 곳이나 클릭하거나 키보드에서 **Esc키** 입력
- 셀 테두리 **파랑색**으로 전환
- 셀에 내용 입력 불가 → **노트북 편집** 가능

◆ Jupyter Notebook

■ 셀(Cell) 다루기

- 셀 추가

- 현재 셀 위에 추가 : command 모드 + a
- 현재 셀 아래에 추가 : command 모드 + b

- 셀 삭제

- 현재 셀 삭제 : command 모드 + dd
- 셀 제거 취소 : command 모드 + z

◆ Jupyter Notebook

■ 셀(Cell) 다루기

- 셀 복사

- 현재 셀 복사 : command 모드 + c

- 셀 붙여넣기

- 현재 셀 위에 붙여 넣기: command 모드 + Shift+v

- 현재 셀 아래에 붙여 넣기 : command 모드 + v

◆ Jupyter Notebook

■ 셀(Cell) 다루기

- 셀 실행

- 현재 셀 실행 : command 모드 + Ctrl + Enter
- 현재 셀 실행 + 다음 셀 이동 : command 모드 + Shift + Enter

- 입력모드 변경

- 현재 셀 Markdown으로 변경 : command 모드 + m
- 현재 셀 Code로 변경 : command 모드 + y

◆ Jupyter Notebook

■ 패키지 관리

- 확인

```
In [ ]: !pip list
```

- 설치

```
In [ ]: !pip install package_name
```

- 업그레이드

```
In [ ]: !pip install package_name --upgrade
```

- 정보 보기

```
In [ ]: !pip show pandas
```

◆ Jupyter Notebook

■ Markdown Language

- 텍스트 기반의 **마크업언어**
- 2004년 존그루버가 만들었고 쉽게 쓰고 읽을 수 있으며 HTML 변환 가능
- **특수기호와 문자**를 이용한 매우 **간단한 구조의 문법** 사용
- 웹에서도 보다 빠르게 콘텐츠를 작성, 직관적으로 인식

◆ Jupyter Notebook

■ Markdown Language

헤더 Headers

- # H1
- ## H2
- ### H3
- #### H4
- ##### H5
- ##### H6

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

◆ Jupyter Notebook

■ Markdown Language

블록쿼터 BlockQuote

- 인용 블록 의미 >
- 안에 다른 Markdown 요소 포함 가능

```
> This is a first blockquote.  
>     > This is a second blockquote.  
>     >     > This is a third blockquote.
```

This is a first blockquote.

This is a second blockquote.

This is a third blockquote.

◆ Jupyter Notebook

■ Markdown Language

목록 List

- 순서있는 목록(번호)
 - 숫자. 항목명
- 순서없는 목록
 - *, +, - 항목명

1. 첫 번째
2. 두 번째
3. 세 번째

1. 첫 번째
2. 두 번째
3. 세 번째

- * 빨강
- * 녹색
- * 파랑

- + 빨강
- + 녹색
- + 파랑

- 빨강
- 녹색
- 파랑

- 빨강
 - 녹색
 - 파랑

- 빨강
 - 녹색
 - 파랑

- 빨강
 - 녹색
 - 파랑

◆ Jupyter Notebook

■ Markdown Language

목록 List

- 순서있는 & 순서없는 혼합

- * 1단계
 - 2단계
 - + 3단계
 - + 4단계

- 1단계
 - 2단계
 - 3단계
 - 4단계

◆ Jupyter Notebook

■ Markdown Language

들여쓰기 Intention

- 4개 공백 또는 하나의 탭 + 한 줄 띄어쓰기

The diagram illustrates the indentation rules for Markdown in Jupyter Notebook. It shows two examples of how to format text and code blocks using 4 spaces or a tab.

Left Example (Using 4 spaces for indentation):

```
This is a normal paragraph:
    공백
    This is a code block.
    공백
    end code block.
```

Right Example (Using a tab for indentation):

```
This is a normal paragraph:
    This is a code block.
    end code block.
```

◆ Jupyter Notebook

■ Markdown Language

코드블럭

- 방법 1) `<pre><code>{code}</code></pre>`

```
<pre>
<code>
public class BootSpringBootApplication {
    public static void main(String[] args) {
        System.out.println("Hello, Honeymon");
    }
}
</code>
</pre>
```

```
public class BootSpringBootApplication {
    public static void main(String[] args) {
        System.out.println("Hello, Honeymon");
    }
}
```

◆ Jupyter Notebook

■ Markdown Language

코드블럭

- 방법 2) ""코 드 ""

```
...  
public class BootSpringBootApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello, Honeymon");  
    }  
}  
...
```

```
public class BootSpringBootApplication {  
    public static void main(String[] args) {  
        System.out.println("Hello, Honeymon");  
    }  
}
```

◆ Jupyter Notebook

■ Markdown Language

코드블럭

- 방법 2) ""코 드 ""

문법강조(Syntax highlighting)

```java

```
public class BootSpringBootApplication {
 public static void main(String[] args) {
 System.out.println("Hello, Honeymon");
 }
}
...
```

```
public class BootSpringBootApplication {
 public static void main(String[] args) {
 System.out.println("Hello, Honeymon");
 }
}
```

## ◆ Jupyter Notebook

### ■ Markdown Language

줄바꿈 / 개행

- Space 2번 입력

```
good Luck Happy
2022 2022 2022
Merry Christmas
HaHaHaHa
```

```
good Luck Happy 2022 2022 2022
Merry Christmas
HaHaHaHa
```

## ◆ Jupyter Notebook

### ■ Markdown Language

문장 내 강조

- 볼드(진하게) → `_진하게_`
- 이탤릭 → `_이탤릭 기울여서_`
- 취소선 → `~~취소~~`
- 밑줄 → `<u>밑줄</u>`

**볼드(진하게)**  
*이탤릭체(기울여서)*  
~~취소선~~  
밑줄

## ◆ Jupyter Notebook

### ■ Markdown Language

수식

- \$ 수식 \$ : 왼쪽 정렬
- \$\$ 수식 \$\$ : 가운데 정렬

```
$y=ax+b$
$$y=ax+b$$
```

$$y = ax + b$$

$$y = ax + b$$

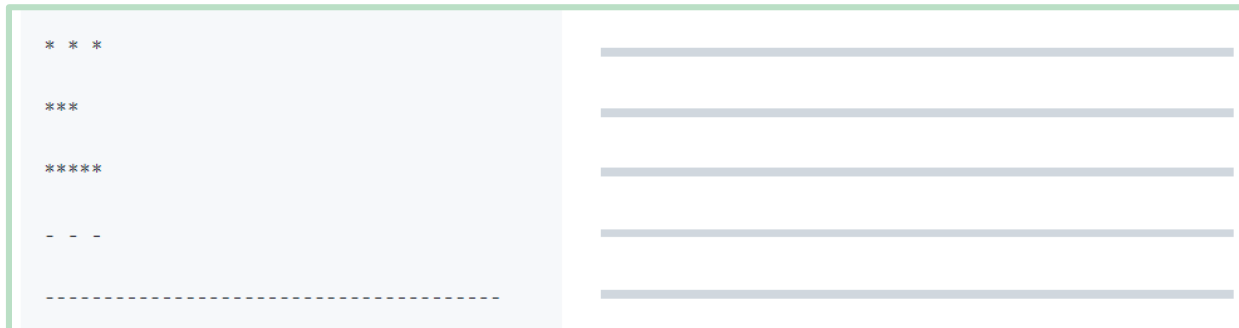


## ◆ Jupyter Notebook

### ■ Markdown Language

수평선

- 미리보기로 출력할 때 페이지 나누기 용도로 많이 사용



## ◆ Jupyter Notebook

### ■ Markdown Language

#### 매직명령어

- 터미널 명령어 사용
- 형식 : %명령어
- %pwd
- %ls
- %mkdir %rmdir
- %load 파일명 => 파일 내용 코드셀로 출력
- %%writefile 파일명 => 파일 저장

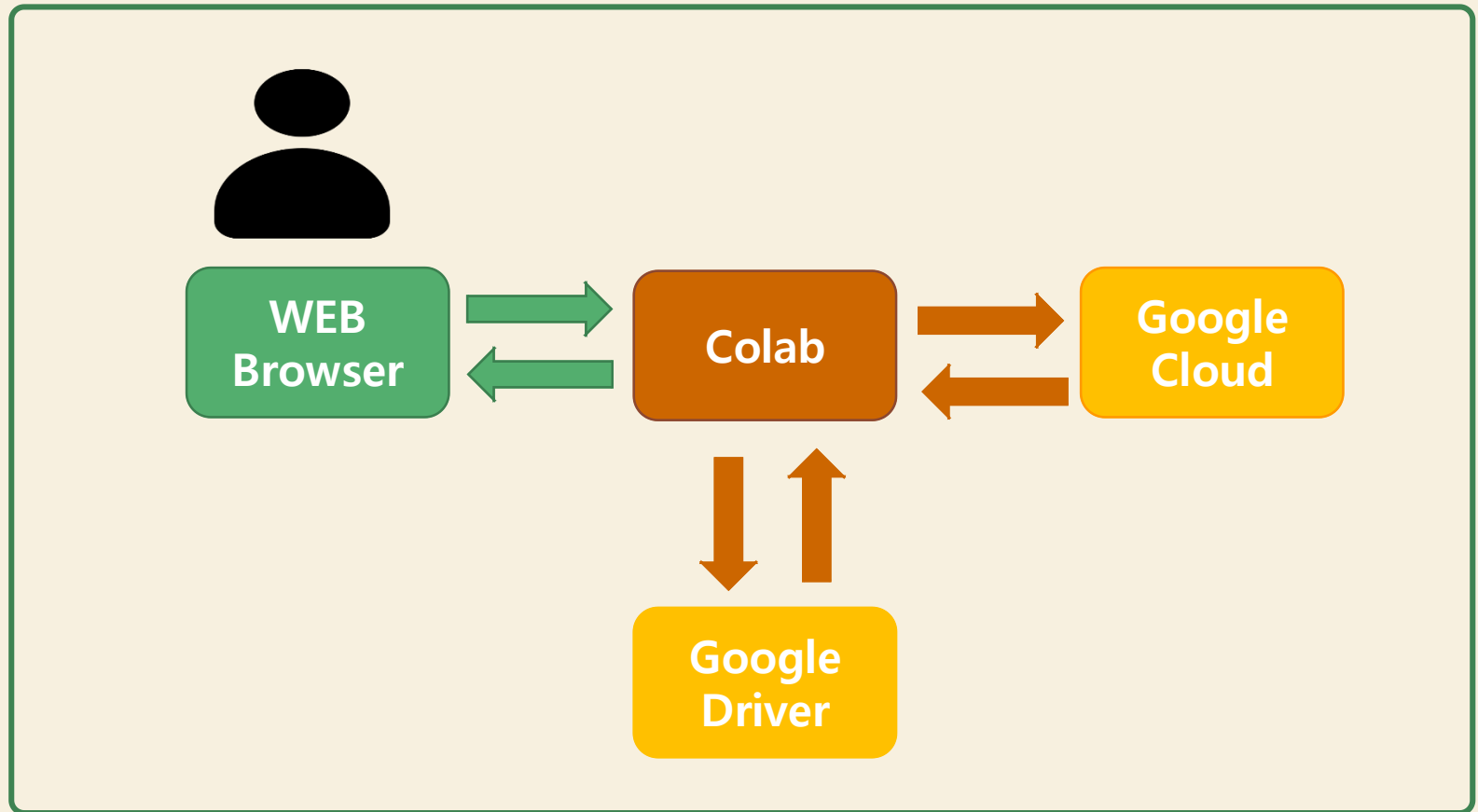
```
In [1]: %lsmagic
Out[1]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave
%copy %ddir %debug %dhist %dirs %doctest_mode %echo %ed %edit
%load_ext %loadpy %logoff %logon %logstart %logstate %logstop
%page %pastebin %pdb %odef %doc %file %info %info2 %pip
%pycat %pylab %otconsole %quickref %recall %rehashx %reload_ext
%sc %set_env %store %sx %system %tb %time %timeit %unalias

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html
python %%python2 %%python3 %%ruby %%script %%sh %%svg %%sx %
Autonagic is ON, % prefix IS NOT needed for line magics.
```

## ◆ 구글 코랩(CoLab)

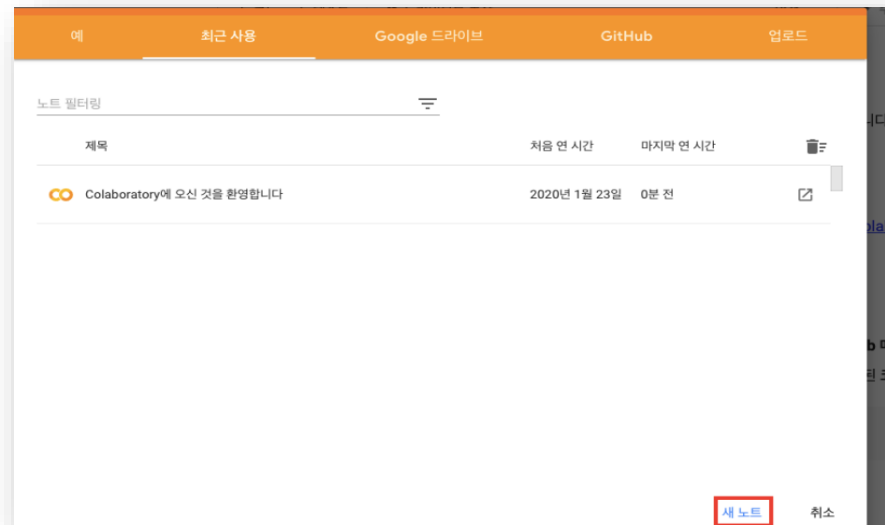
- Google Laboratory 서비스 줄임말
- 클라우드 기반으로 주피터 노트북 개발환경
- 교육과 과학 연구를 목적으로 개발된 도구
- 2017년 무료 공개
- 구글 드라이브, 도커, 리눅스, 구글 클라우드 기술로 이루어짐
- 저 사양 컴퓨터이거나 GPU가 없는 컴퓨터에 추천
- 데이터 분석 및 AI를 위한 패키지 기본 설치되어 있음
- 단, 원하는 버전의 환경 사용은 불가

## ◆ 구글 코랩(CoLab)



## ◆ 구글 코랩(CoLab)

- Google 회원가입 & 로그인
- 새노트 클릭



## ◆ 구글 코랩(CoLab)

- 입력 후 실행 → **Shift+Enter**

