

CHAPTER 0

Table of contents

1. Install Python and Jupyter Notebook

- 1) Install Python
- 2) Install Jupyter Notebook

2. Introduction to Python and Jupyter Notebook

- 1) Creating Folder
- 2) Creating Notebook

3. Hot key for Jupyter Notebook

- 1) Command Mode
- 2) Edit Mode

1. Install Python

Python은 jupyter notebook 에디터를 이용한다. (구글에서 제공하는 colab도 좋은 에디터이며 별도의 설치과정 이 필요하지는 않다.) 먼저, Python을 설치 후 jupyter notebook을 설치하는 방법과 작업환경을 만드는 방법에 대해 알아보도록 한다.

NOTE)

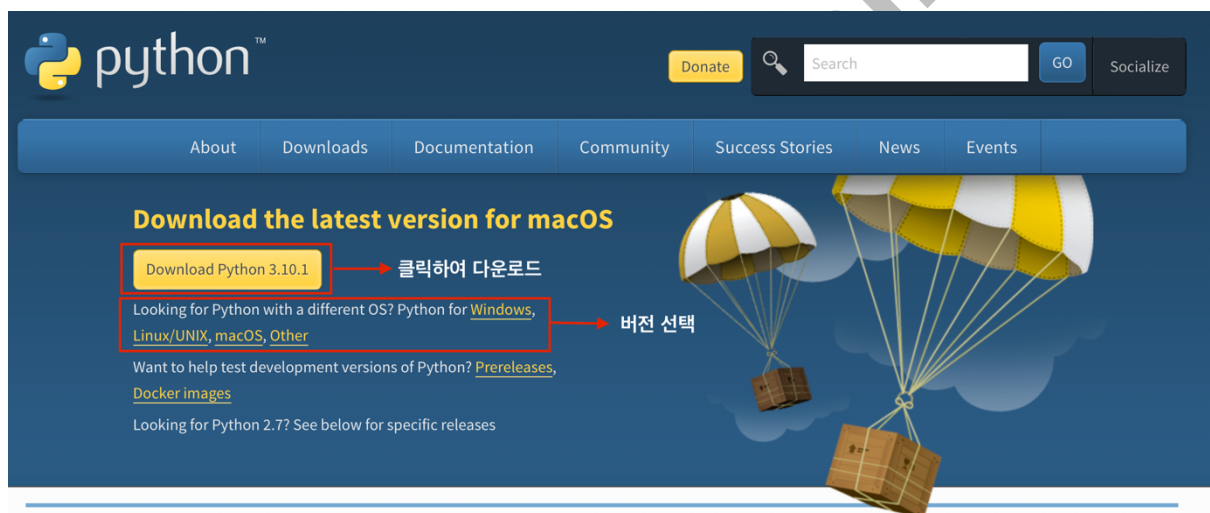
jupyter notebook은 Python을 포함하고 있지 않기 때문에, Python을 설치해야한다.

1) Install Python

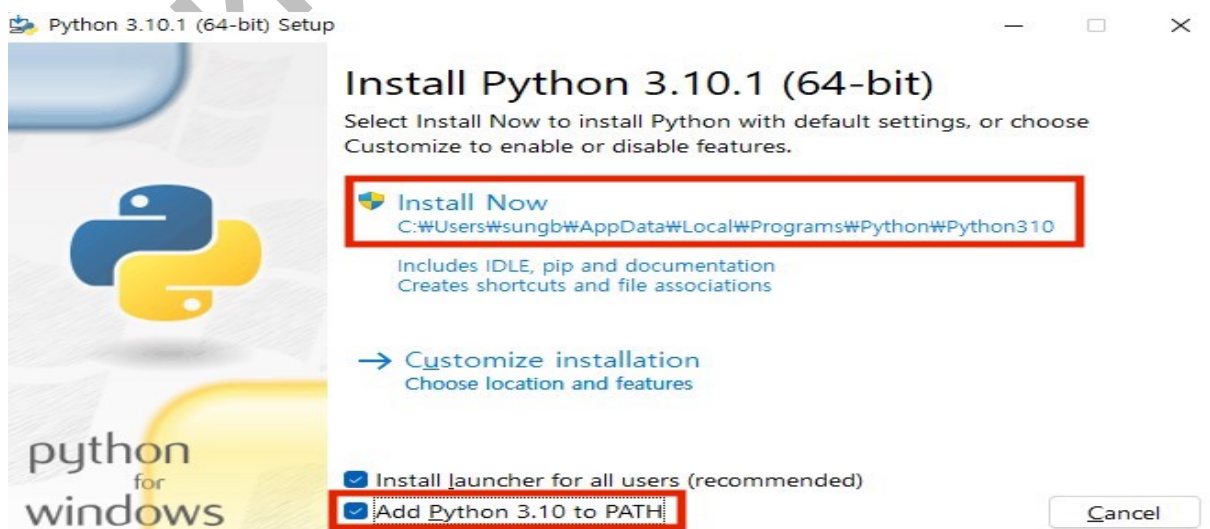
파이썬 공식 홈페이지에서 언어 패키지를 다운로드 받는다.

NOTE)

파이썬 공식 사이트 : <https://www.python.org/downloads/>



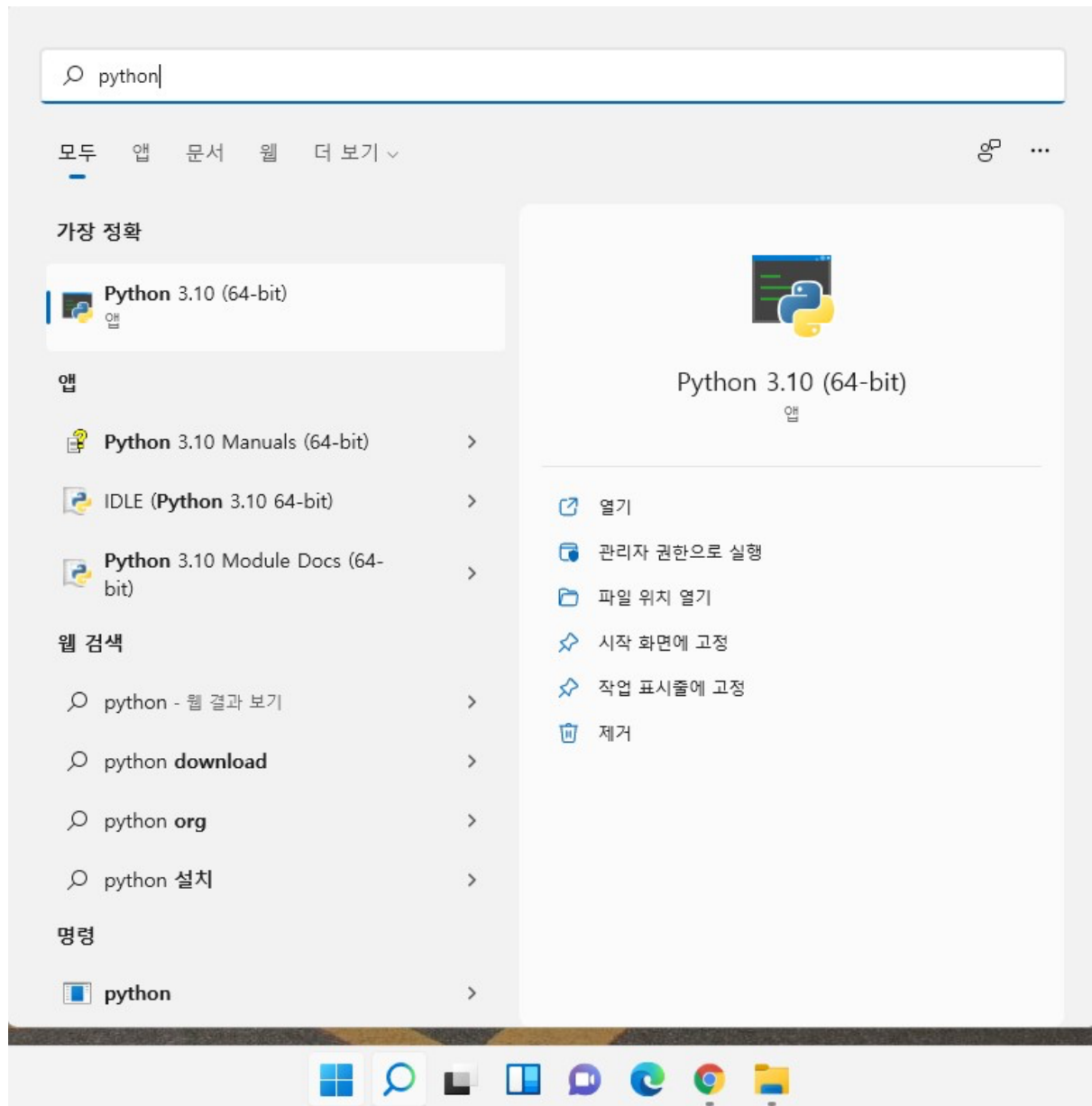
인스톨러를 실행한 후 "Install Now"를 선택하면 설치가 진행된다.



NOTE)

"Add Python 3.7 to PATH" 옵션을 누락하면 오류가 발생할 수 있다.

설치가 완료되면 "close"를 누르고 종료한다. 파이썬이 정상적으로 설치되었다면 다음과 같이 확인할 수 있다.



2) Install Jupyter Notebook

Jupyter Notebook을 실행하기 위해선 anaconda를 설치하거나 pip를 이용하여 설치하는 방법이 있다. Anaconda는 Jupyter Notebook 이외에도 다른 부수적인 프로그램들이 많이 깔려 메모리를 많이 차지하게 되므로 python만을 이용한다면 pip를 이용하여 Jupyter Notebook을 설치하고 실행하는 것이 좋다.

최근에는 python을 다운로드하는 과정에서 pip가 자동으로 설치된다. 다음 과정을 통해 pip가 설치되었는지 확인한다.

a) pip 설치 확인

관리자 명령 프롬프트를 열고 다음을 입력한다.

```
pip help
```

다음과 같이 출력이 된다면 pip가 설치되었다는 것이므로 바로 Jupyter Notebook을 설치한다.



```

관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>pip help

Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements format.
  list              List installed packages.
  show              Show information about installed packages.
  check             Verify installed packages have compatible dependencies.
  config            Manage local and global configuration.
  search            Search PyPI for packages.
  cache             Inspect and manage pip's wheel cache.
  index             Inspect information available from package indexes.
  wheel             Build wheels from your requirements.
  hash             Compute hashes of package archives.
  completion        A helper command used for command completion.
  debug            Show information useful for debugging.
  help             Show help for commands.

General Options:
  -h, --help            Show help.
  --isolated            Run pip in an isolated mode, ignoring environment variables and user configuration.
  -v, --verbose         Give more output. Option is additive, and can be used up to 3 times.
  -V, --version         Show version and exit.
  -q, --quiet          Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).
  --log <path>        Path to a verbose appending log.
  --no-input           Disable prompting for input.
  --proxy <proxy>      Specify a proxy in the form [user:passwd@]proxy.server:port.
  --retries <retries>  Maximum number of retries each connection should attempt (default 5 times).
  --timeout <sec>      Set the socket timeout (default 15 seconds).
  --exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.
  --trusted-host <hostname> Mark this host or host:port pair as trusted, even though it does not have valid or any HTTPS.
  --cert <path>        Path to PEM-encoded CA certificate bundle. If provided, overrides the default. See 'SSL Certificate Verification' in pip documentation for more information.
  --client-cert <path> Path to SSL client certificate, a single file containing the private key and the certificate in PEM format.
  --cache-dir <dir>   Store the cache data in <dir>.
  --no-cache-dir       Disable the cache.
  --disable-pip-version-check Don't periodically check PyPI to determine whether a new version of pip is available for download. Implied with --no-index.
  --no-color           Suppress colored output.
  --no-python-version-warning Silence deprecation warnings for upcoming unsupported Python versions.
  --use-feature <feature> Enable new functionality, that may be backward incompatible.
  
```

b) pip 설치

pip를 이용하여 Jupyter Notebook을 설치하는 과정이다. 관리자 명령 프롬프트에 다음 명령어를 입력한다.

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

아래와 같이 다운로드가 출력된다.

```
C:\WINDOWS\system32>curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total       Spent    Left     Speed
100 1734k  100 1734k    0     0  867k      0  0:00:02  0:00:02 --:--:-- 834k
```

다운로드 후 pip 설치를 위해 다음 명령어를 입력한다.

```
python get-pip.py
```

pip가 정상적으로 설치가 되었으면 다음과 같이 출력된다.

```
관리자: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>python get-pip.py
Collecting pip
  Using cached https://files.pythonhosted.org/packages/00/b6/9cfa56b4081ad13874b0c6f96af8ce1/pip-19.3.1-py2.py3-none-any.whl
Installing collected packages: pip
  Found existing installation: pip 19.3.1
  Uninstalling pip-19.3.1:
    Successfully uninstalled pip-19.3.1
Successfully installed pip-19.3.1
```

a)의 과정을 반복하여 pip가 제대로 설치되었는지 확인한다.

c) jupyter notebook 설치

프롬프트를 열고 다음을 입력한다.(관리자 권한 모드로 들어가지 않는다.)

```
pip3 install jupyter
```

```
관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>pip3 install jupyter
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting qtconsole
  Downloading qtconsole-5.2.2-py3-none-any.whl (120 kB)
Collecting nbconvert
  Downloading nbconvert-6.4.0-py3-none-any.whl (557 kB)
Collecting jupyter-console
  Downloading jupyter_console-6.4.0-py3-none-any.whl (22 kB)
Collecting notebook
  Downloading notebook-6.4.7-py3-none-any.whl (9.9 MB)
Collecting ipynbutils
  Downloading ipynbutils-7.6.5-py2.py3-none-any.whl (121 kB)
Collecting ipykernel
  Downloading ipykernel-6.6.1-py3-none-any.whl (126 kB)
Collecting ipython>=7.23.1
  Downloading ipython-8.0.0-py3-none-any.whl (747 kB)
Collecting traitlets<6.0,>=5.1.0
  Downloading traitlets-5.1.1-py3-none-any.whl (102 kB)
Collecting debugpy<2.0,>=1.0.0
  Downloading debugpy-1.5.1-cp310-cp310-win_amd64.whl (4.4 MB)
Collecting tornado<7.0,>=4.2
  Downloading tornado-6.1.1.tar.gz (497 kB)
Collecting jupyter-client<8.0
  Downloading jupyter_client-7.1.0-py3-none-any.whl (129 kB)
Collecting matplotlib-inline<0.2.0,>=0.1.0
  Downloading matplotlib_inline-0.1.3-py3-none-any.whl (8.2 kB)
Collecting nest-asyncio
```

NOTE)

이때, 설치 후 Warning pip 버전 관련 메시지가 출력될 경우 업그레이드를 한다.

```
pip3 install --user --upgrade pip
```

d) Jupyter notebook 실행

관리자 프롬프트를 열고 다음을 입력한다.

```
jupyter notebook
```

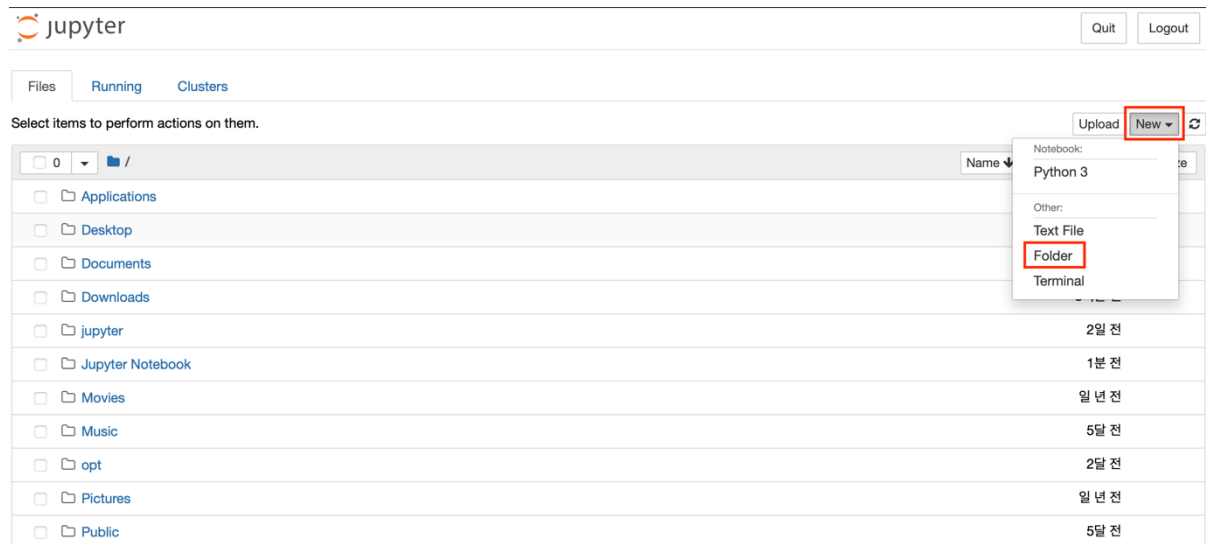
다음과 같은 창이 뜨면 jupyter notebook이 실행된 것이다.



2. Introduction to Python and Jupyter Notebook

Python을 이용한 작업환경이 완성되었다. 코드를 작성하기 위해 notebook을 생성하는 방법을 알아보도록 한다.

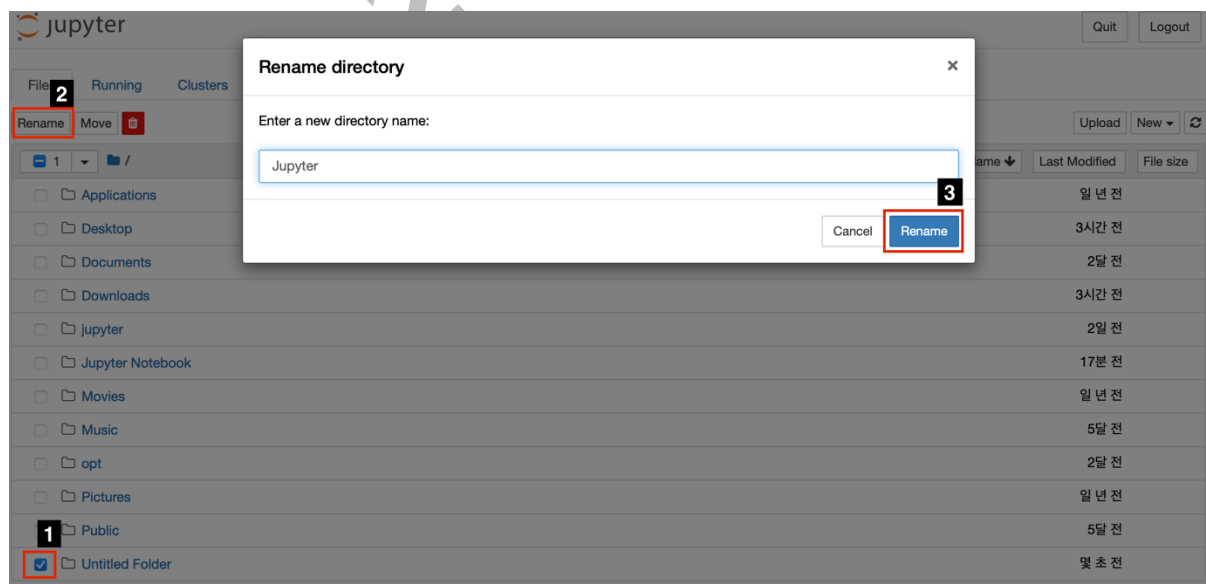
1) Creating Folder



위의 그림을 참고하여 새로운 폴더를 만든다. notebook의 저장 경로가 복잡할 경우 매번 파일을 찾는데 시간이 걸리므로 jupyter notebook이 실행되고 가장 첫화면에 폴더를 만드는 것을 추천한다.

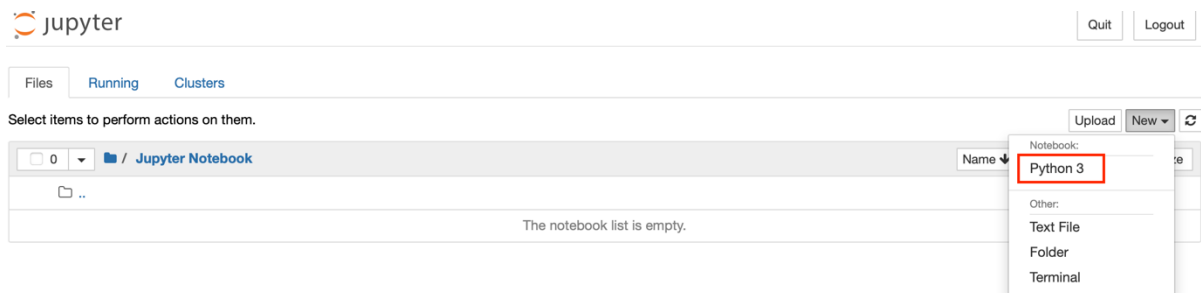
NOTE)

폴더명 바꾸는 방법



2) Creating Notebook

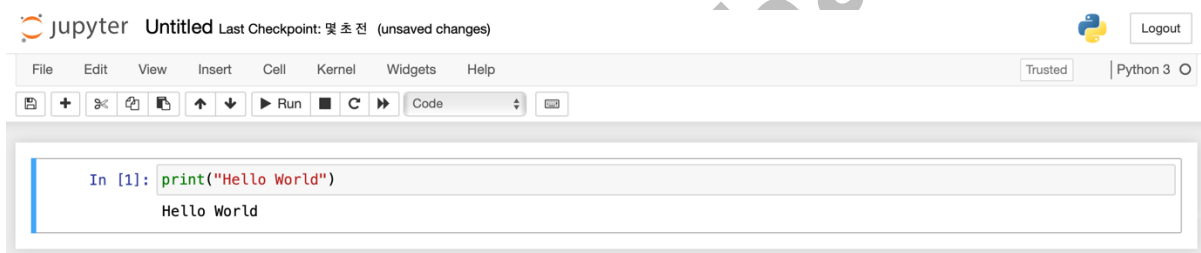
코드를 작성할 notebook을 생성해보도록 한다.



셀에 다음을 입력한 후 실행한다.

```
print("Hello World!")
```

아래와 같은 결과가 나온다.

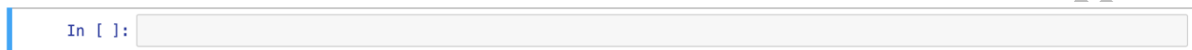


3. Hot key for Jupyter Notebook

jupyter notebook을 이용하는데 있어 몇가지 단축키를 알아두면 유용하다.

1) Command Mode

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



- 위로 셀 추가
[a]
- 아래로 셀 추가
[b]
- 선택 셀 삭제
[d,d] (d를 두번 누름) 혹은 [x]
- 선택 셀 복사하기
[c]
- 선택 셀 아래에 붙여넣기
[p]
- 선택 셀과 아래 셀과 합치기
[shift] + [m]
- 실행결과 열기/닫기
[o]
- Markdown으로 변경
[m]
- code로 변경
[y]
- 파일 저장
[ctrl]+[s] 혹은 [s]
- 선택 셀의 코드 입력 모드로 돌아가기
[enter]

2) Edit Mode

[enter]를 눌러 셀이 아래와 같이 초록색이 된 상태(코드 입력 모드)에서 해당 단축키를 누른다.

In []: |

- 선택 셀의 코드 전체 선택
[ctrl]+[a]
- 선택 셀 내 실행 취소
[ctrl]+[z]
- 선택 셀 내 다시 실행
[ctrl]+[y]
- 커서 위치 라인 주석처리
[ctrl]+[/]
- 선택 셀 코드 실행
[ctrl]+[enter]
- 선택 셀 코드 실행 후 다음 셀로 이동(다음 셀이 없는 경우 새로 추가)
[shift]+[enter]
- 커서 위치에서 셀 둘로 나누기
[shift]+[ctrl]+[-]
- 파일 저장
[ctrl]+[s]
- 셀 선택 모드로 돌아가기
[esc] 또는 [ctrl]+[m]

NOTE)

mac의 경우 [ctrl] 대신 [command]를 누른다.