

○ 줘피터 노트북이란?

줘피터 노트북은 IPython과 같이 인터랙티브 형식의 라이브 코드를 제공하는 웹 베이스 어플리케이션입니다. 줘피터 노트북을 사용하면 웹 브라우저 안에서 실행하고 싶은 코드를 입력하고 그 결과를 바로 확인할 수 있습니다. 또한 웹 브라우저에서 일반 프로그램으로는 구현하기 어려운 수학 공식을 표현할 수도 있고, 다양한 그래프를 생성하여 데이터를 시각화 할 수도 있습니다. 나레이션 텍스트나 이미지 등을 추가하여 노트북 다큐먼트를 만들어 공유를 할 수 있기 때문에 파워포인트, 키노트와 같은 프레젠테이션 프로그램으로도 많이 사용되고 있습니다. 줘피터 노트북은 오픈소스이므로 무료로 다운받아 사용할 수 있습니다.

줘피터 노트북은 기존에 IPython Notebook으로 불렸었는데, 몇 년 전에 프로젝트의 이름이 변경됨에 따라 줘피터 노트북이라는 이름으로 변경되었다고 합니다. 줘피터 노트북은 파이썬 커널 뿐만 아니라 Go, R, Julia, C++, Java 등의 많은 종류의 프로그래밍 언어를 지원하고 있습니다. 줘피터 노트북이 지원하는 모든 언어를 확인하려면 아래의 링크를 참조하면 됩니다.

<https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>

○ 줘피터 노트북 설치

만약 본인의 컴퓨터 또는 노트북에 파이썬을 설치할 때 아나콘다 파이썬을 설치하지 않고, 일반 파이썬을 설치하였다면 줘피터 노트북이 설치되지 않았을 것입니다. 줘피터 노트북이 설치되지 않은 경우 아래의 링크를 클릭하여 설치하시면 됩니다.

[2019/08/11 - Windows10 운영체제에서 파이썬\(Python\) 설치하기](#)

○ 줘피터 노트북의 구성

줘피터 노트북은 크게 네 개의 컴포넌트로 구성되어 있습니다.

(1.0) Notebook App : Notebook App은 서버 및 클라이언트 어플리케이션이며 브라우저(프론트엔드)에서 실행되는 클라이언트와 백엔드에서 실행되는 서버로 이루어져 있습니다. 클라이언트와 서버가 같은 로컬 호스트에서 실행된다면 인터넷이 없이도 실행이 가능하며 클라이언트와 서버가 서로 다른 곳에 있다면 인터넷을 통해 실행이 가능합니다.

[2019/08/11 - 프론트엔드\(Frontend\)와 백엔드\(Backend\)](#)

(1.1) Notebook App - Client(Browser) : 노트북의 클라이언트 앱은 브라우저를 통해 노트북 다큐먼트를 실행 또는 수정하게 도와주는 프론트엔드 어플리케이션입니다.

(1.2) Notebook App - Server(Local host or remote host) : 브라우저(클라이언트)와 파이썬 커널 사이에서 HTTP 또는 Websockets을 통하여 들어온 요청을 IPython 커널을 사용하여 처리한 뒤, 결과를 HTML과 JavaScript 형식으로 브라우저에 반영하는 백엔드 어플리케이션입니다.

(2) Notebook Dashboard : 노트북 앱을 실행하면 첫 번째로 보이는 유저 인터페이스이며, 노트북 다큐먼트를 실행하거나 커널을 실행하는데 사용됩니다.

(3) Notebook Document : 노트북 앱에 의해서 생성되는 JSON 형식의 파일이며 .ipynb 확장자를 가집니다. 클라이언트 앱에서 실행되는 코드 또는 데이터의 정보가 저장되는 파일이기도 합니다.

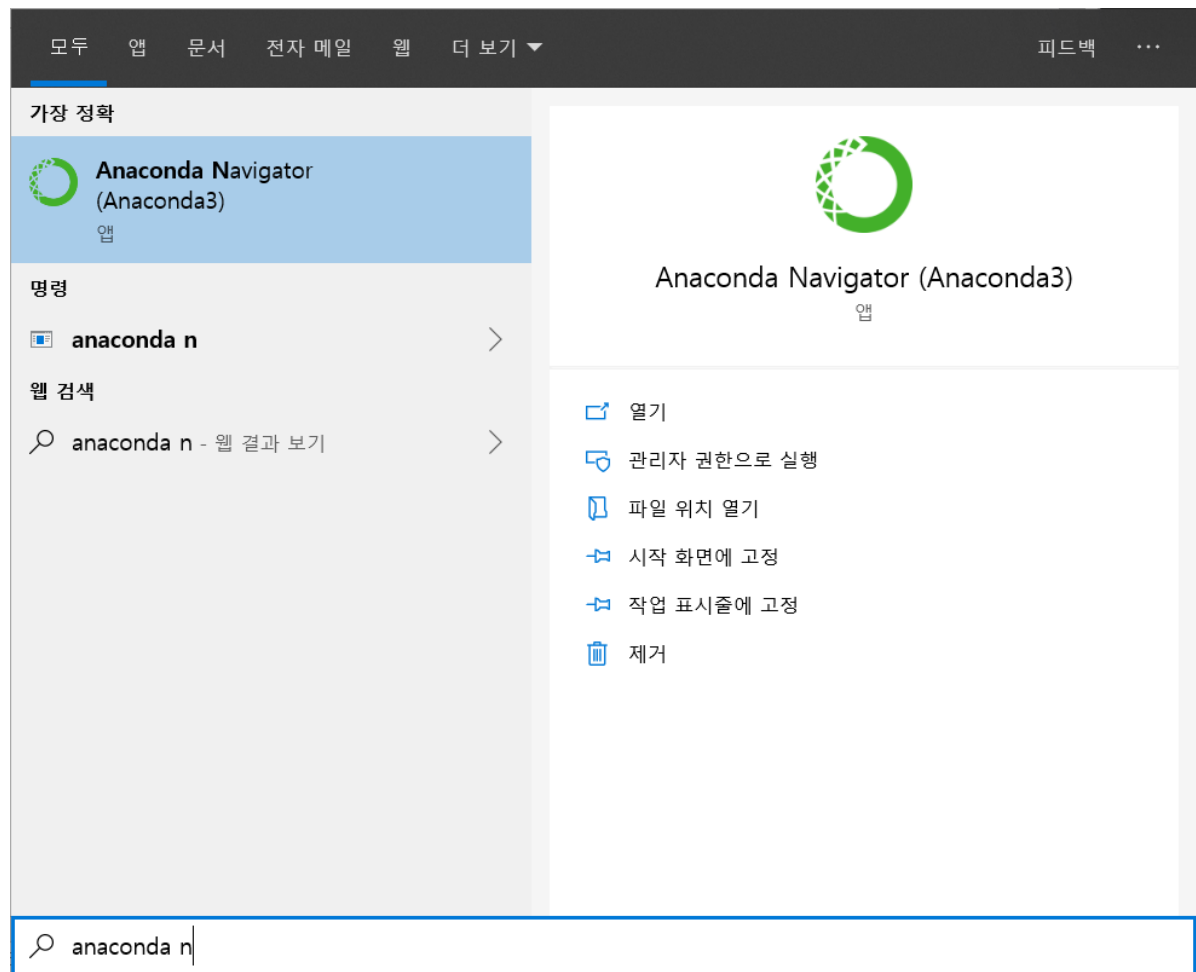
(4) Kerner(IPython) : 서버가 사용하는 “연산 엔진”입니다. 클라이언트 앱이 노트북 다큐먼트 안의 코드를 실행하면, 서버가 HTTP 또는 websocket 프로토콜을 통해 요청을 받고, 이 커널을 사용하여 연산을 한 뒤의 결과 값을 클라이언트에게 넘겨 브라우저에 표시합니다.

○ 서버 기동 방법

쥬피터 노트북의 서버를 기동하는 방법은 여러 가지가 있습니다.

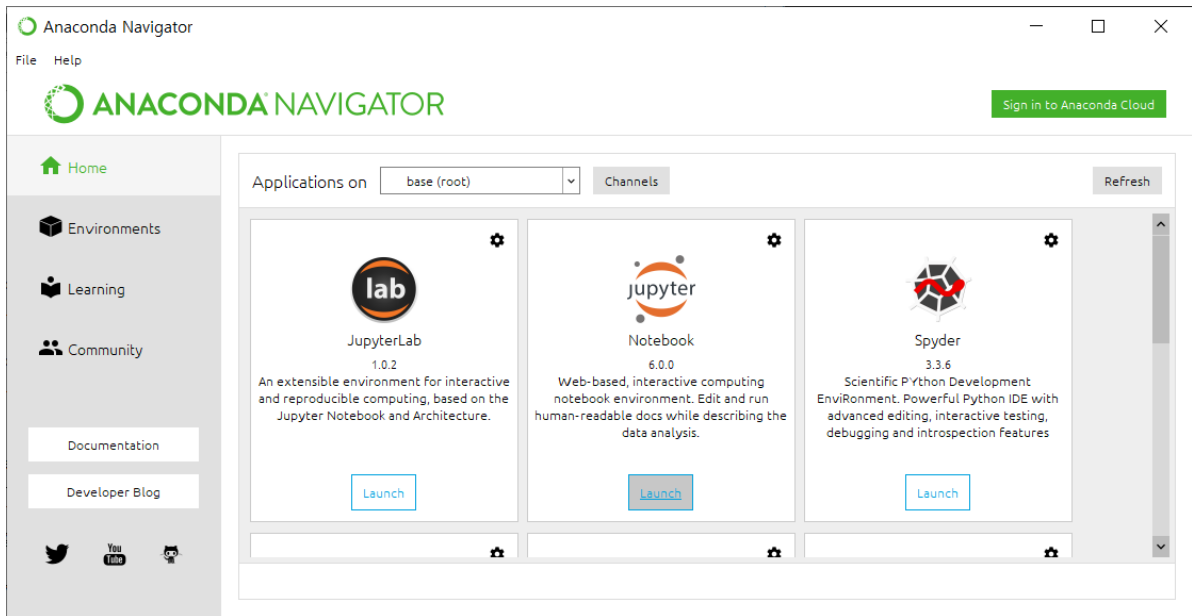
(1) 아나콘다 네비게이션을 통한 서버 기동

윈도우 검색 창에 Anaconda를 검색하고 Anaconda Navigator를 실행합니다.



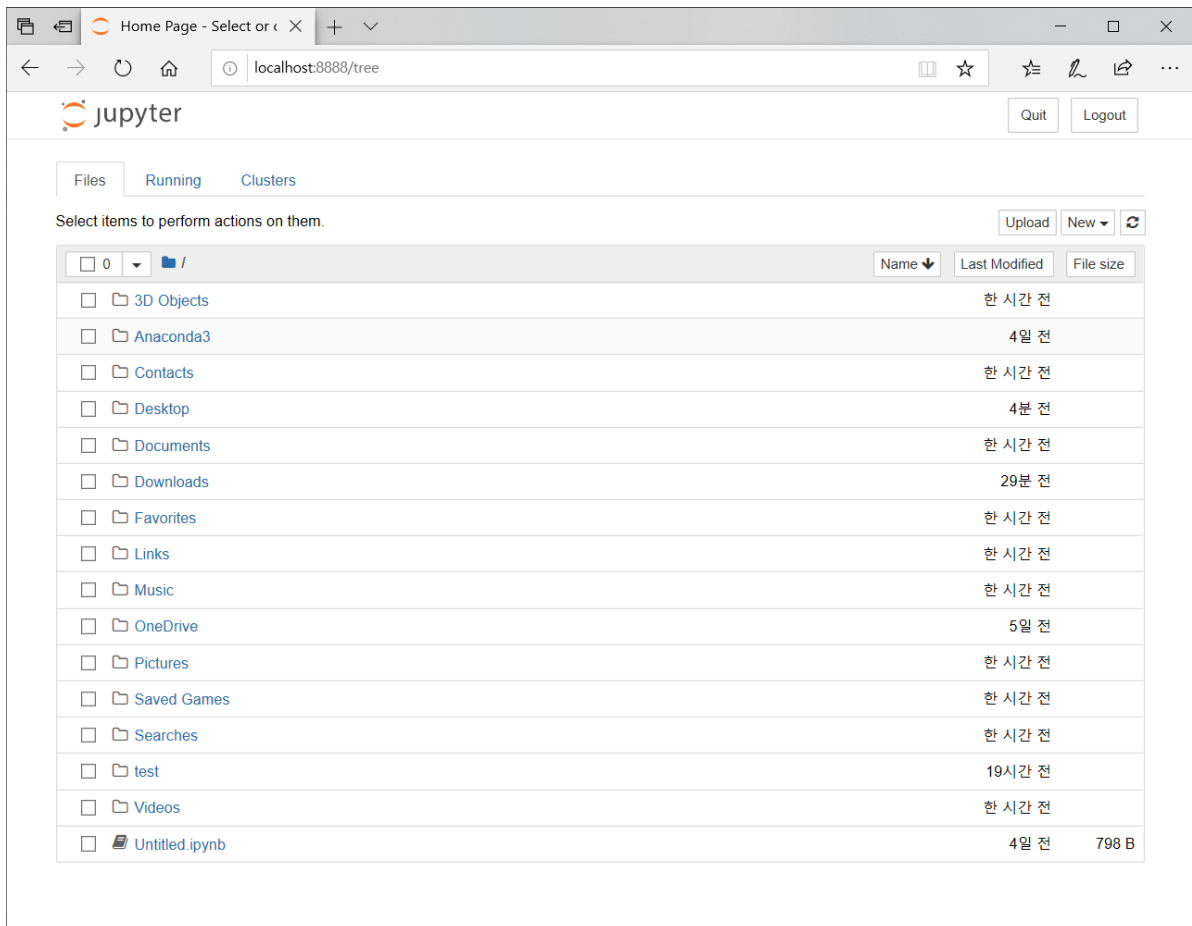
아나콘다 네비게이션을 통한 서버 기동 - 1

Anaconda Navigator가 실행되면, Jupyter Notebook에 있는 Launch 버튼을 클릭합니다.



아나콘다 네비게이션을 통한 서버 기동 - 2

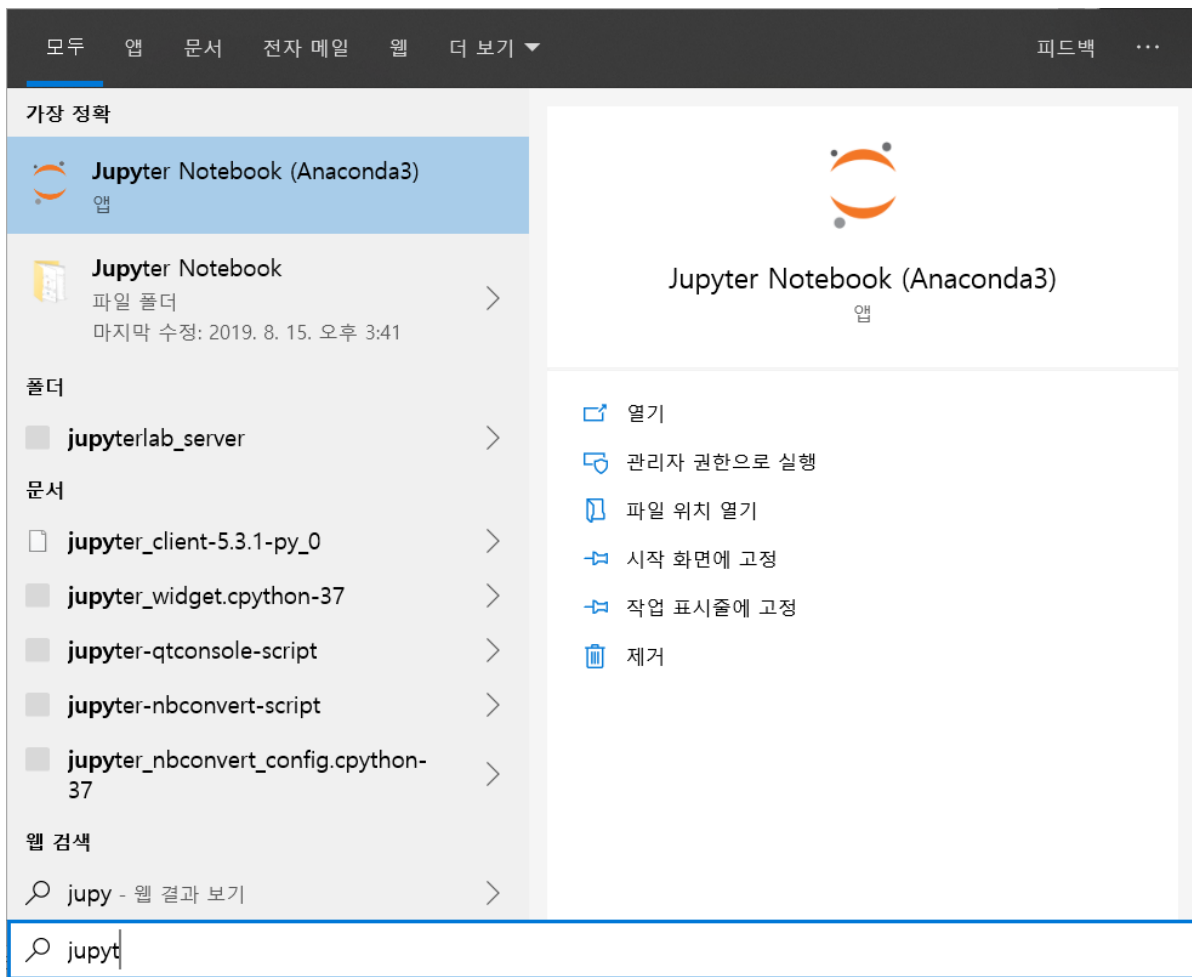
그러면 아래와 같이 웹 브라우저를 통해 주피터 노트북이 실행되는 것을 볼 수 있습니다.



아나콘다 네비게이션을 통한 서버 기동 - 3

(2) 주피터 노트북 데스크탑 어플리케이션을 통한 서버 기동

윈도우 검색 창에 Jupyter를 검색한 후 Jupyter Notebook 어플리케이션을 실행합니다.



쥬피터 노트북 데스크탑 어플리케이션을 통한 서버 기동 - 1

그러면 아래와 같이 명령 프롬프트 창에 자동으로 서버가 실행되고, 웹 브라우저를 통해 쥬피터 노트북이 실행되는 것을 볼 수 있습니다.

```

Jupyter Notebook (Anaconda3)
[15:47:41.699 NotebookApp] The port 8888 is already in use, trying another port.
[15:47:41.863 NotebookApp] JupyterLab extension loaded from C:\Users\lajet\Anaconda3\lib\site-packages\jupyterlab
[15:47:41.863 NotebookApp] JupyterLab application directory is C:\Users\lajet\Anaconda3\share\jupyter\lab
[15:47:41.872 NotebookApp] Serving notebooks from local directory: C:\Users\lajet
[15:47:41.873 NotebookApp] The Jupyter Notebook is running at:
[15:47:41.873 NotebookApp] http://localhost:8889/?token=1cb34b187d5150b509471794170211f7430210897a6dfe58
[15:47:41.874 NotebookApp] or http://127.0.0.1:8889/?token=1cb34b187d5150b509471794170211f7430210897a6dfe58
[15:47:41.874 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 15:47:42.043 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/lajet/AppData/Roaming/jupyter/runtime/nbserver-4316-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=1cb34b187d5150b509471794170211f7430210897a6dfe58
or http://127.0.0.1:8889/?token=1cb34b187d5150b509471794170211f7430210897a6dfe58
[E 15:47:46.485 NotebookApp] Could not open static file ''
[W 15:47:46.672 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (::1) 99.57ms referer=http://localhost:8889/tree?token=1cb34b187d5150b509471794170211f7430210897a6dfe58

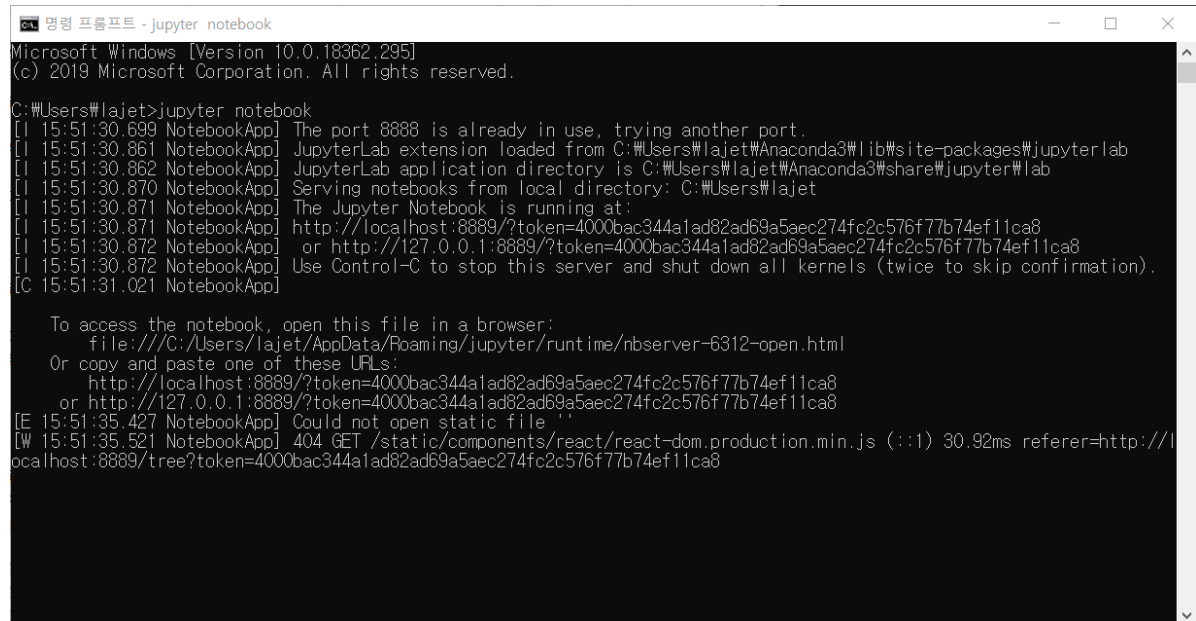
```

쥬피터 노트북 데스크탑 어플리케이션을 통한 서버 기동 - 2

(3) 명령 프롬프트 창에서 커멘드 실행을 통한 서버 기동

윈도우 검색 창에 cmd를 입력하고 명령 프롬프트를 실행한 뒤, 다음과 같은 명령어를 입력하여 Jupyter Notebook Command를 실행하면 다음과 같이 서버가 실행됩니다.

```
jupyter notebook
```



```
Microsoft Windows [Version 10.0.18362.295]
(c) 2019 Microsoft Corporation. All rights reserved.

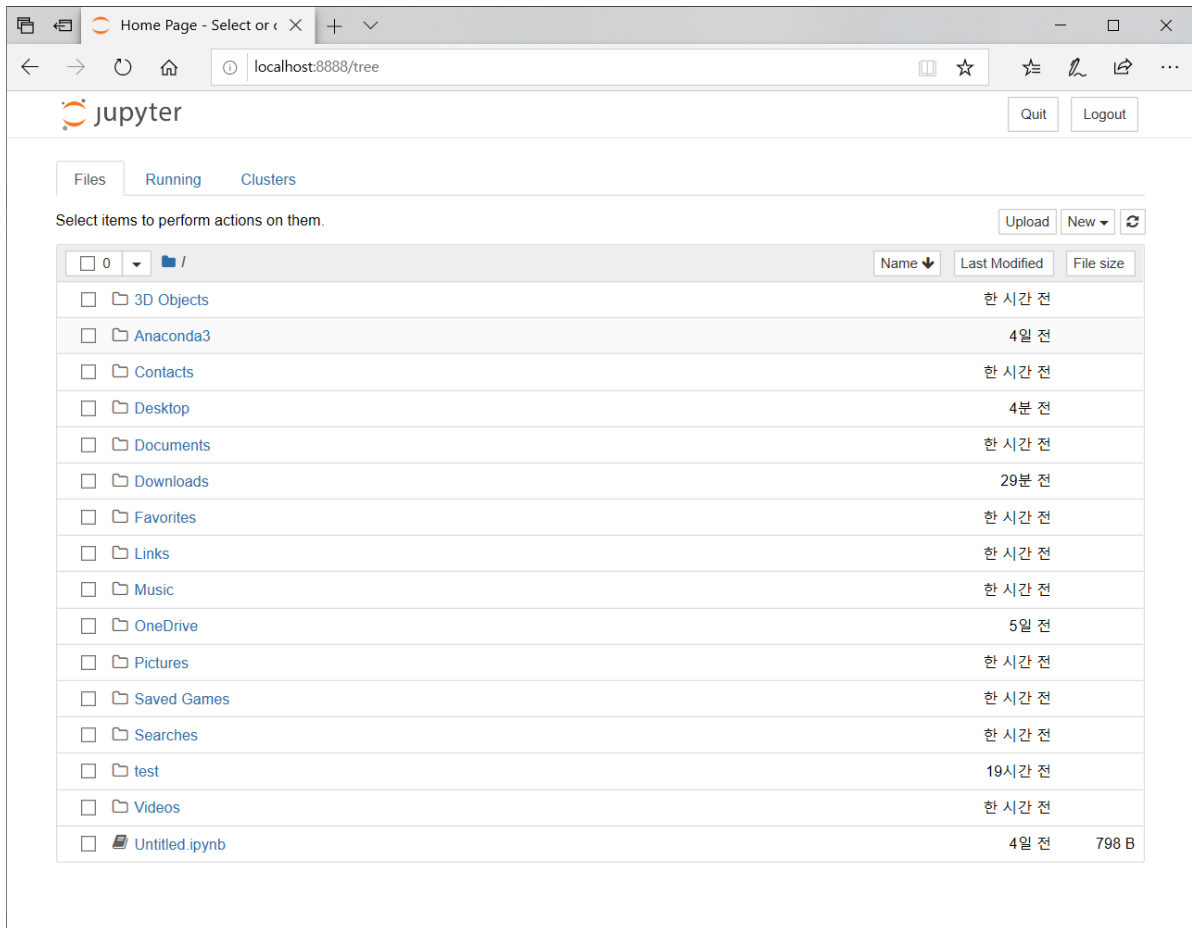
C:\Users\lajet>jupyter notebook
[I 15:51:30.699 NotebookApp] The port 8888 is already in use, trying another port.
[I 15:51:30.861 NotebookApp] JupyterLab extension loaded from C:\Users\lajet\Anaconda3\lib\site-packages\jupyterlab
[I 15:51:30.862 NotebookApp] JupyterLab application directory is C:\Users\lajet\Anaconda3\share\jupyter\lab
[I 15:51:30.870 NotebookApp] Serving notebooks from local directory: C:\Users\lajet
[I 15:51:30.871 NotebookApp] The Jupyter Notebook is running at:
[I 15:51:30.871 NotebookApp] http://localhost:8889/?token=4000bac344a1ad82ad69a5aec274fc2c576f77b74ef11ca8
[I 15:51:30.872 NotebookApp] or http://127.0.0.1:8889/?token=4000bac344a1ad82ad69a5aec274fc2c576f77b74ef11ca8
[I 15:51:30.872 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 15:51:31.021 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/lajet/AppData/Roaming/jupyter/runtime/nbserver-6312-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=4000bac344a1ad82ad69a5aec274fc2c576f77b74ef11ca8
or http://127.0.0.1:8889/?token=4000bac344a1ad82ad69a5aec274fc2c576f77b74ef11ca8
[E 15:51:35.427 NotebookApp] Could not open static file ''
[W 15:51:35.521 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (::1) 30.92ms referer=http://localhost:8889/tree?token=4000bac344a1ad82ad69a5aec274fc2c576f77b74ef11ca8
```

명령 프롬프트 창에서 커멘드 실행을 통한 서버 기동 - 1

서버가 실행되면 마찬가지로 웹 브라우저를 통해 주피터 노트북이 실행되는 것을 볼 수 있습니다.



명령 프롬프트 창에서 커맨드 실행을 통한 서버 기동 - 2

커맨드를 실행한 현재 폴더가 주피터 노트북에 최상 폴더로 인식되어 실행됩니다. 만약, 자신의 홈 폴더가 아닌 다른 폴더에서 실행하고 싶을 경우, Ctrl+C를 눌러 서버를 중지한 후 cd 명령어를 통해 원하는 폴더로 이동하여 주피터 노트북을 실행하거나 폴더 패스를 명령어 다음에 입력하여 실행하면 됩니다.

cd 이동폴더경로

// 또는

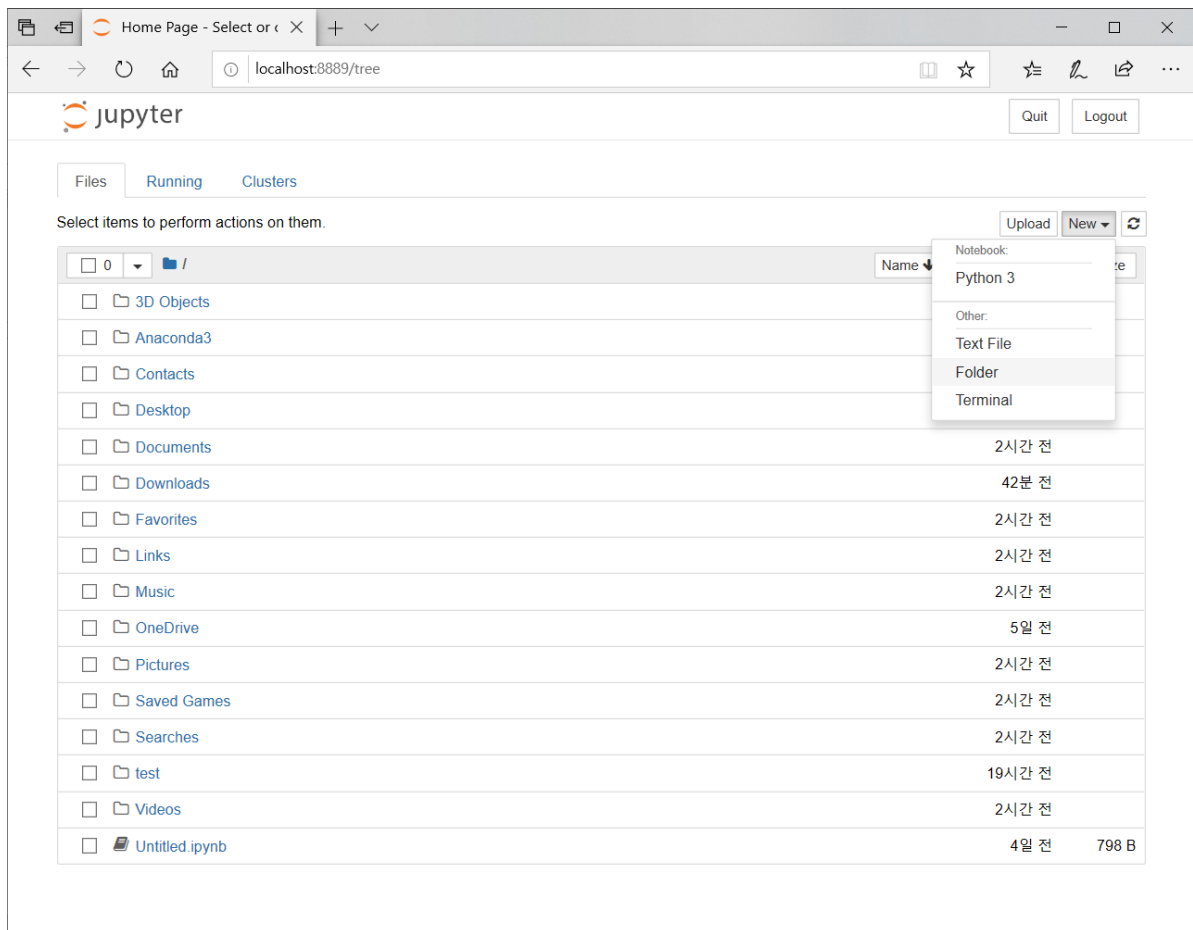
jupyter notebook %이동폴더이름%

○ 폴더 생성 및 다큐먼트 생성

노트북을 사용할 때는 노트북을 저장하는 폴더를 생성하고 노트북의 종류, 프로젝트를 구분하여 하위 폴더를 생성하여 관리하는 것이 편리합니다.

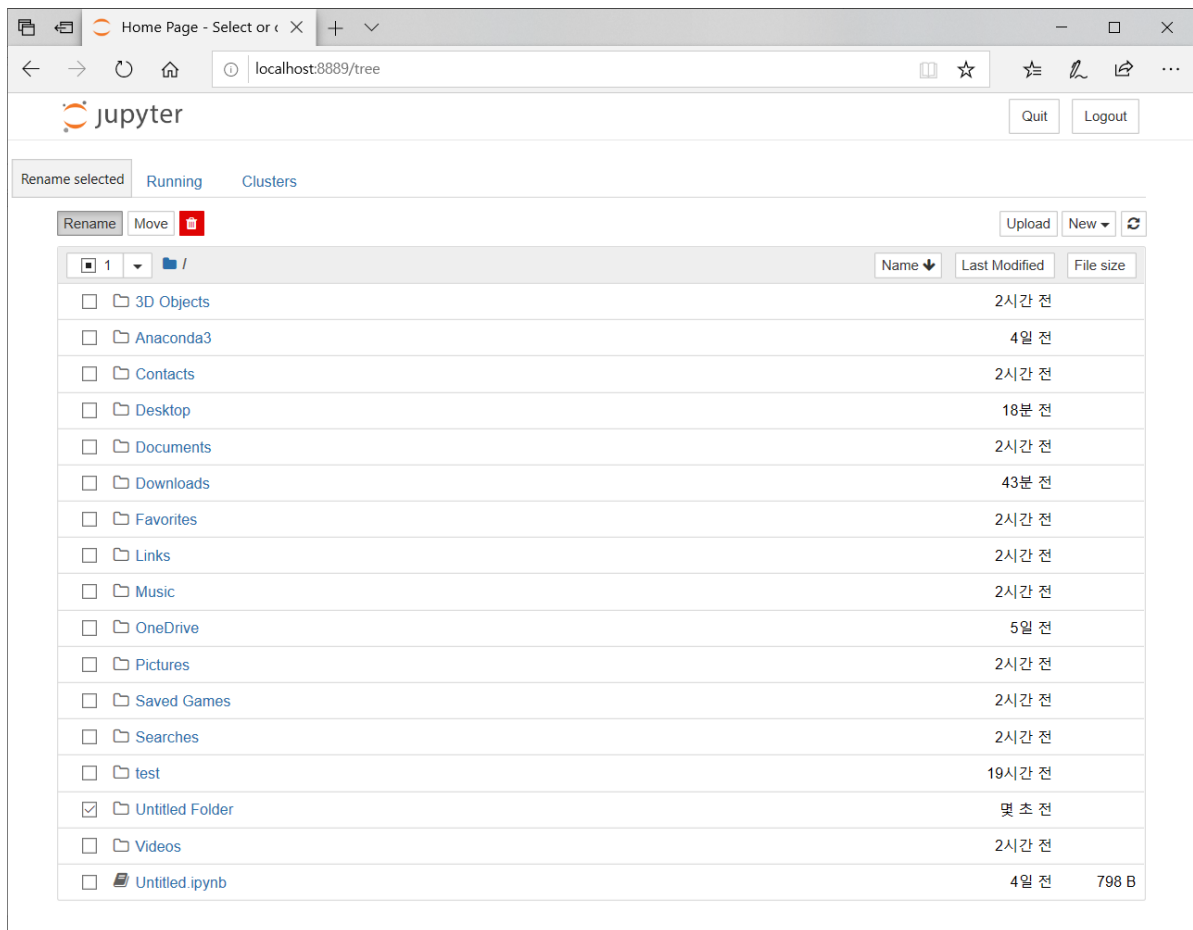
(1) 폴더 생성

주피터 노트북 페이지에서 [New] 버튼을 누르면 나오는 드롭다운 목록 중에서 [Folder] 옵션을 클릭하면 새로운 폴더를 생성할 수 있습니다.



폴더 생성 - 1

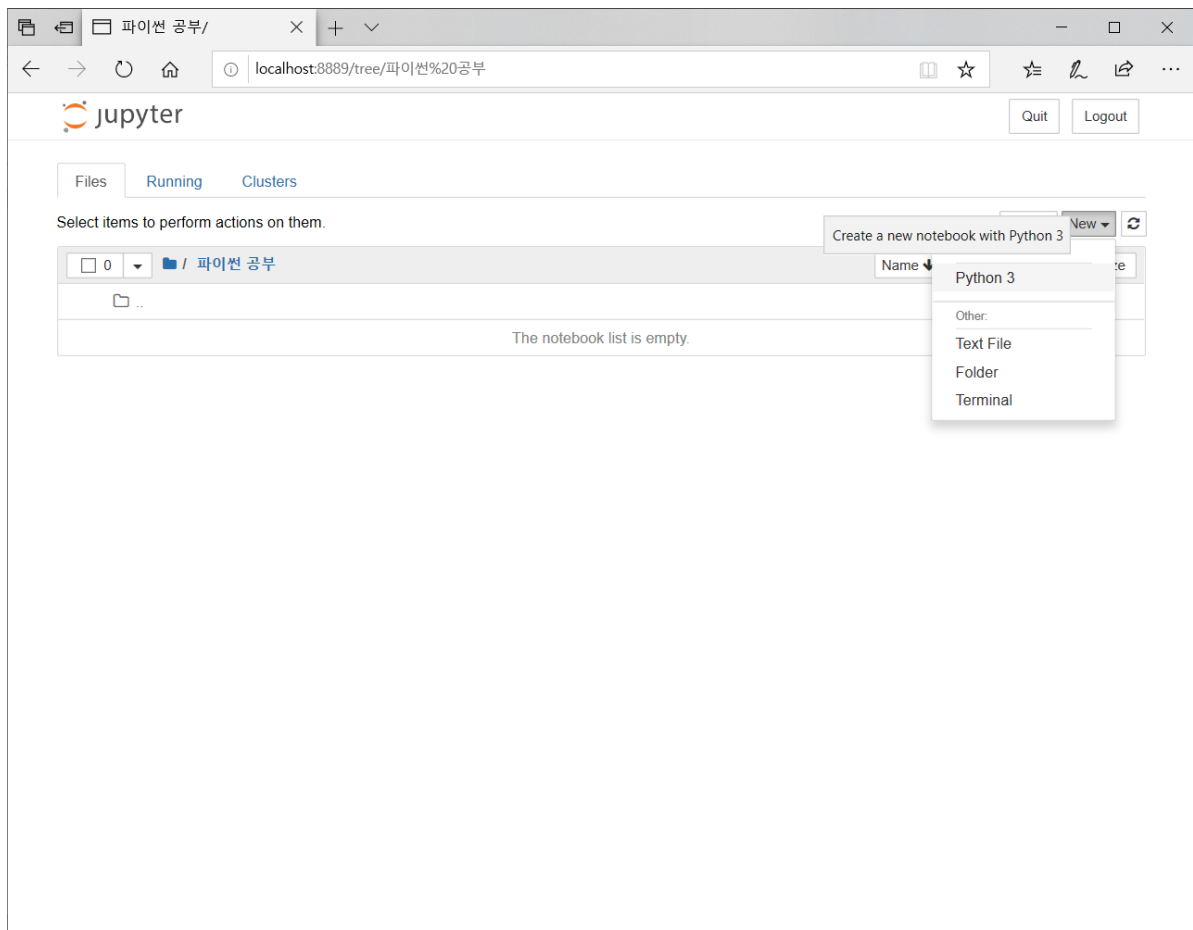
새로 만든 폴더의 이름은 [Untitled Folder]이며, 해당 폴더를 체크한 후 좌측 상단에 있는 [Rename] 버튼을 눌러 폴더의 이름을 변경할 수 있습니다.



폴더 생성 - 2

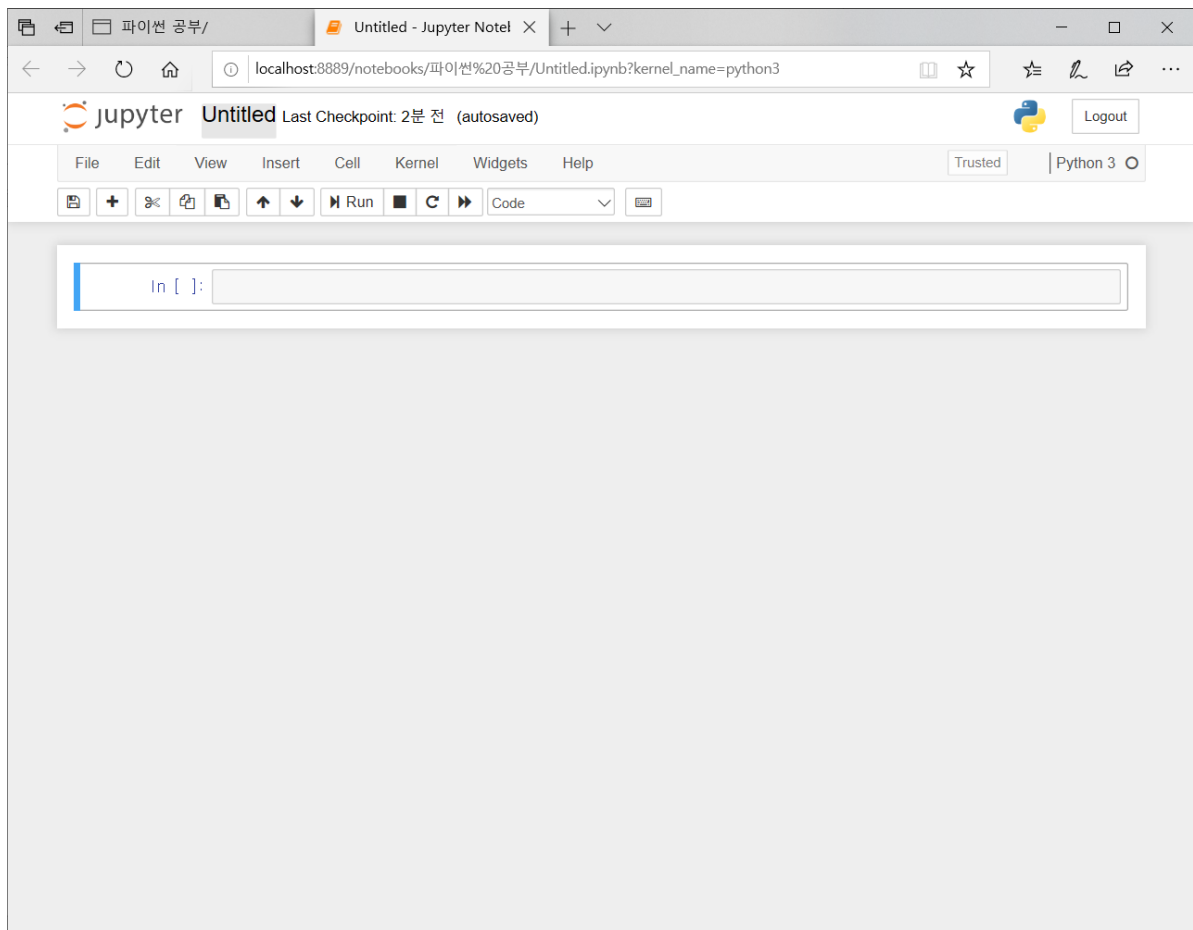
(2) 다큐먼트 생성

새로 만든 폴더의 이름을 파이썬으로 변경했고, 이 폴더에 다큐먼트를 생성하겠습니다. 새로운 폴더를 클릭하여 해당 폴더로 이동한 뒤 마찬가지로 우측상단에 있는 [New] 버튼을 누르면 나오는 드롭다운 목록 중에서 [Python 3]를 클릭하면 새로운 다큐먼트를 생성할 수 있습니다.



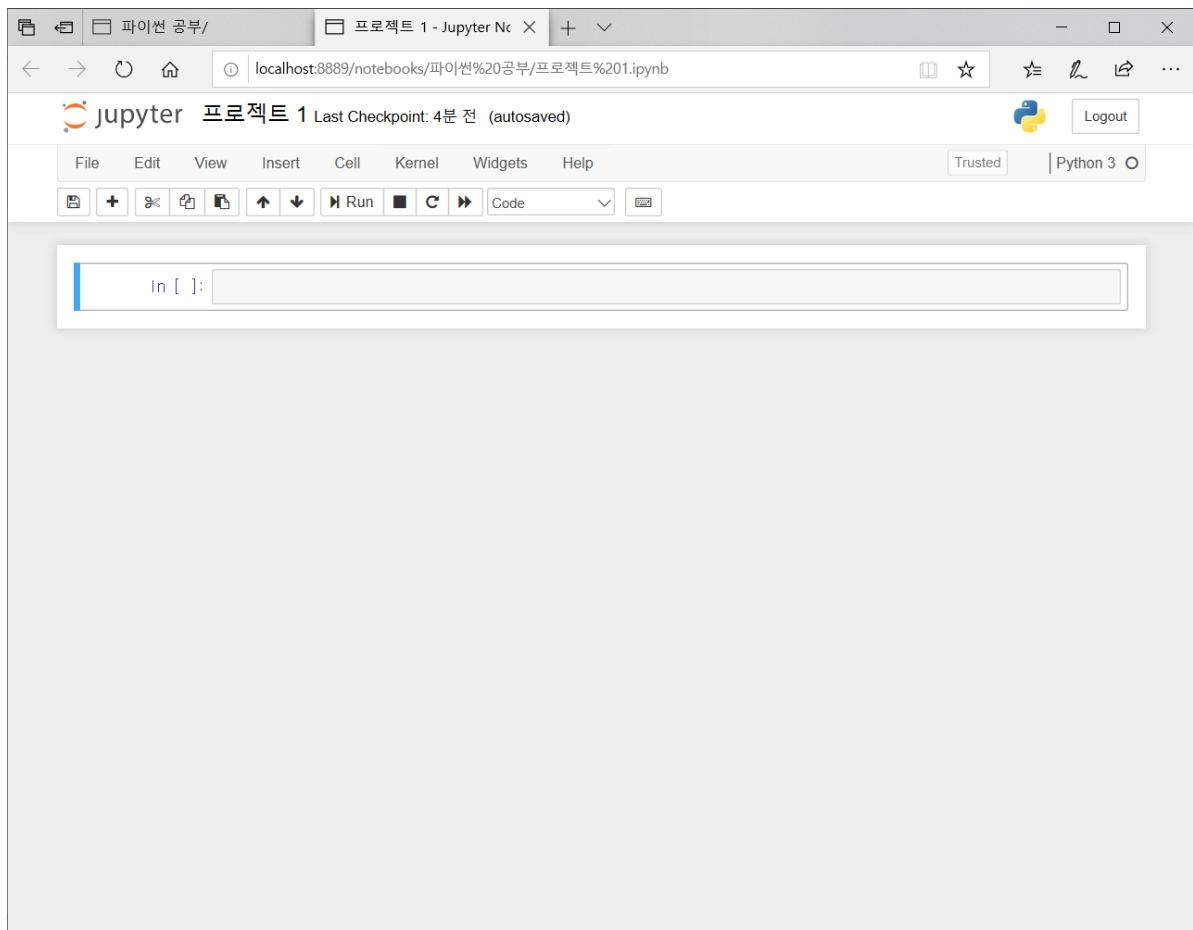
다큐먼트 생성 - 1

새롭게 생성된 다큐먼트의 이름도 역시 [Untitled]이며, 해당 부분을 클릭하면 다큐먼트의 이름을 변경할 수 있습니다.



다큐먼트 생성 - 2

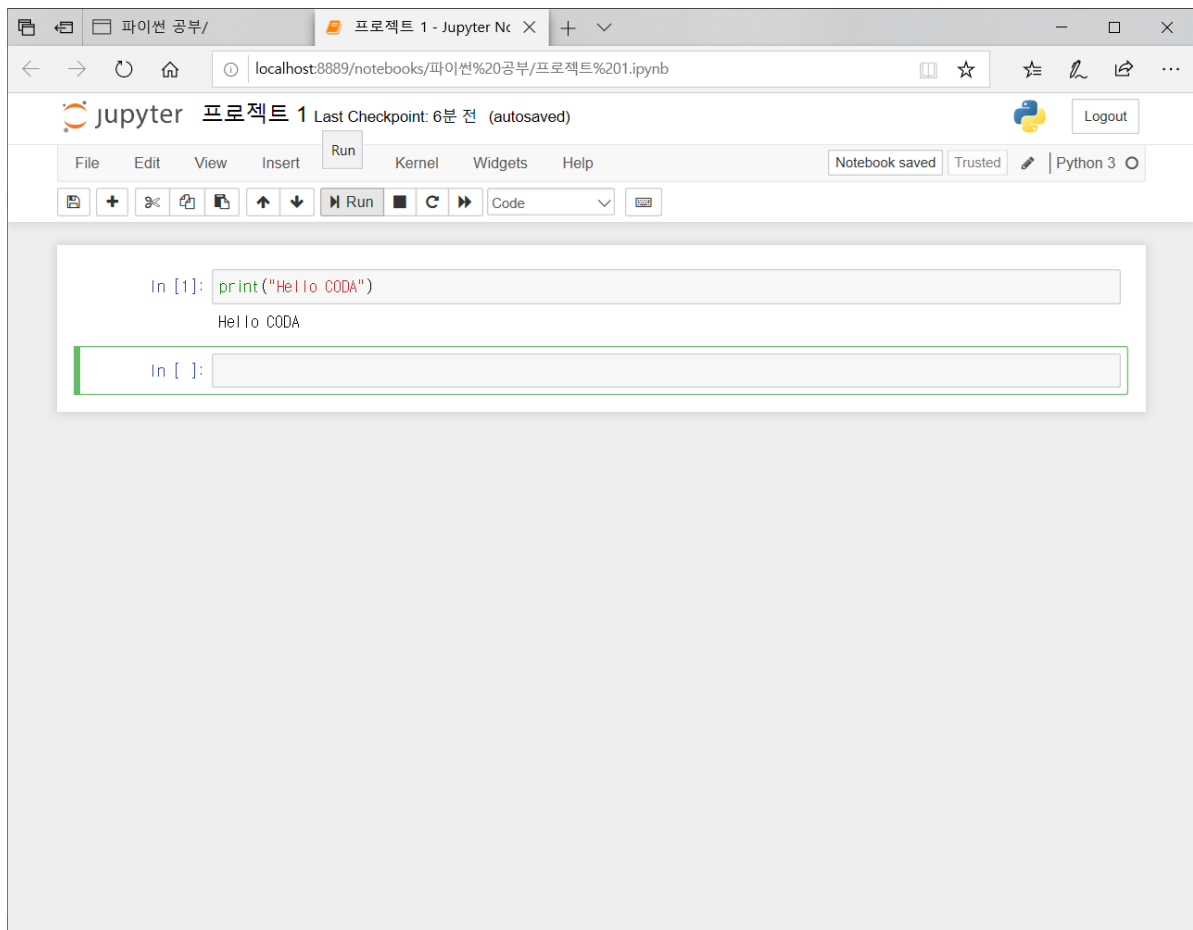
저는 [프로젝트 1]이라는 이름을 붙여주었으며, 이렇게 새로 만든 다큐먼트에 본격적으로 코드를 실행해 보도록 하겠습니다.



다큐먼트 생성 - 3

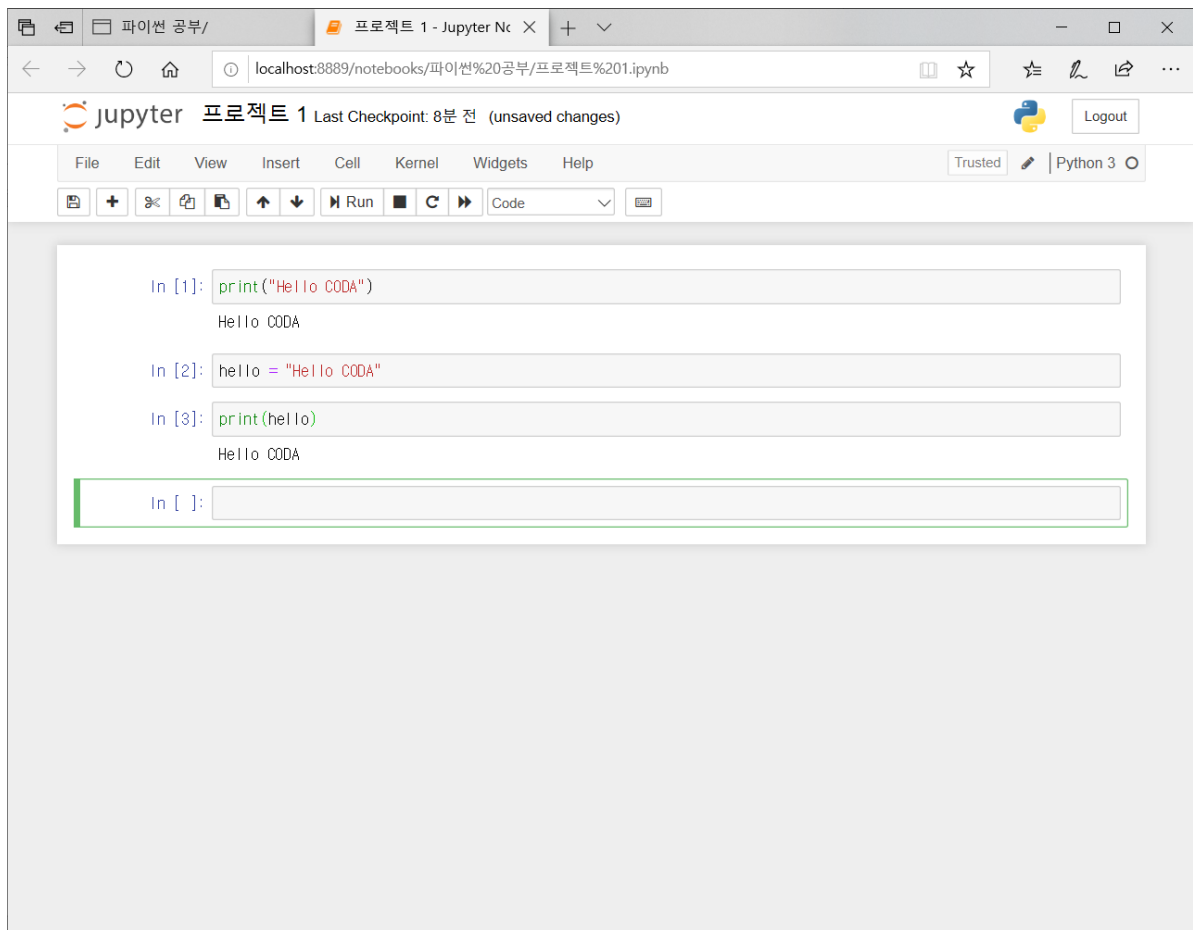
○ 코드 실행

다음과 같은 코드를 입력하여 [Run]버튼을 눌러 “Hello CODA”를 출력해보도록 하겠습니다. 그러면 다음과 같이 Hello CODA가 출력된 것을 볼 수 있습니다.



코드 실행 - 1

이번에는 [2]번 셀에서 hello라는 변수에 Hello CODA라는 문자를 할당하고 [3]번 셀에서 hello 변수를 참조하여 출력해보도록 하겠습니다. 그러면 다음과 같이 Hello CODA가 출력된 것을 볼 수 있습니다.



The screenshot shows a Jupyter Notebook window titled '프로젝트 1 - Jupyter Notebook'. The browser address bar shows 'localhost:8889/notebooks/파이썬%20공부/프로젝트%201.ipynb'. The notebook interface includes a top bar with the Jupyter logo, the title '프로젝트 1', and a 'Last Checkpoint: 8분 전 (unsaved changes)' message. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A toolbar contains icons for saving, adding cells, undo, redo, and running code. The main area displays three code cells:

```
In [1]: print("Hello CODA")
Hello CODA

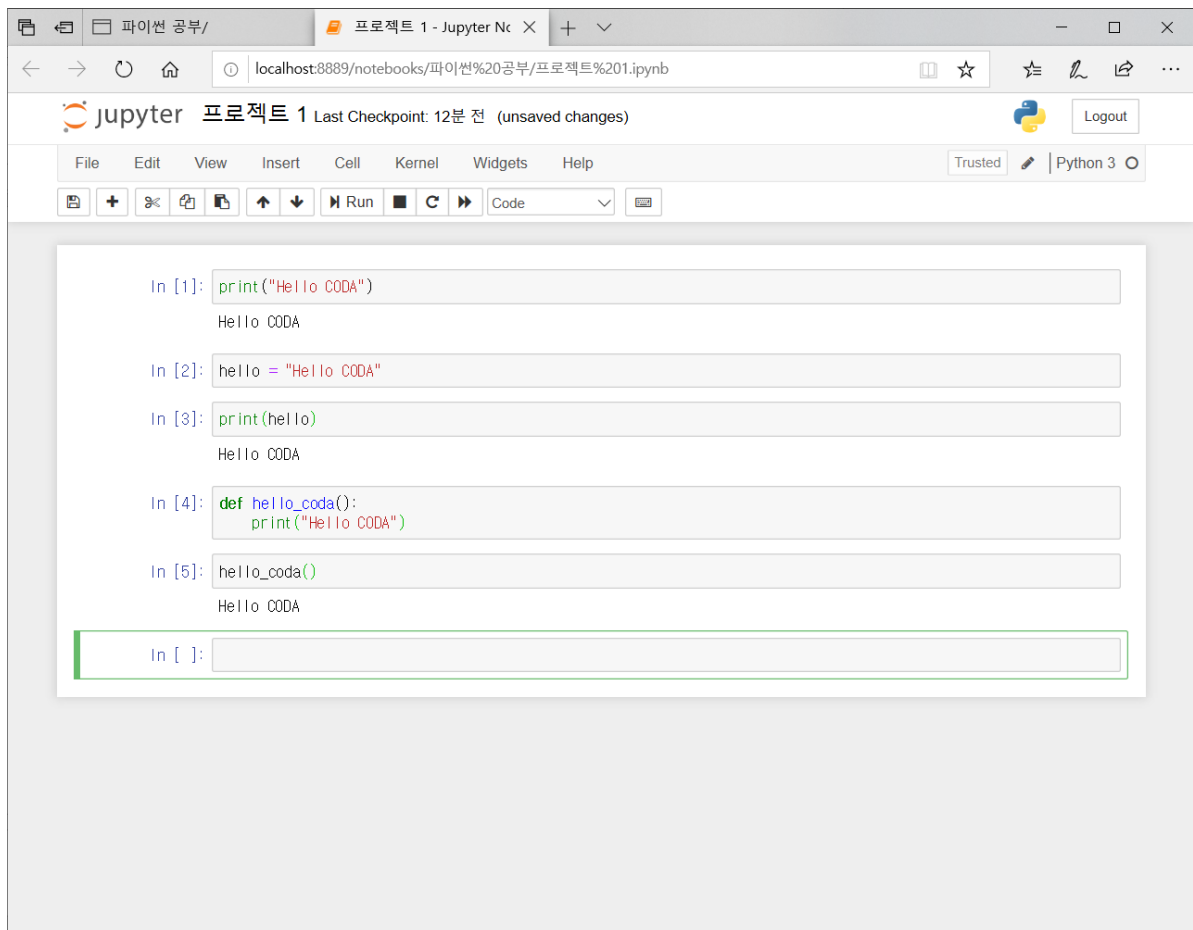
In [2]: hello = "Hello CODA"

In [3]: print(hello)
Hello CODA

In [ ]:
```

코드 실행 - 2

다음은 [3]번 셀에서 `hello_coda`라는 함수를 정의하고 [4]번 셀에서 `hello_coda`라는 함수를 출력해보도록 하겠습니다. 그러면 다음과 같이 `hello_coda` 함수 안에 할당된 `Hello CODA`가 출력된 것을 볼 수 있습니다.



```
In [1]: print("Hello CODA")
Hello CODA

In [2]: hello = "Hello CODA"

In [3]: print(hello)
Hello CODA

In [4]: def hello_coda():
        print("Hello CODA")

In [5]: hello_coda()
Hello CODA

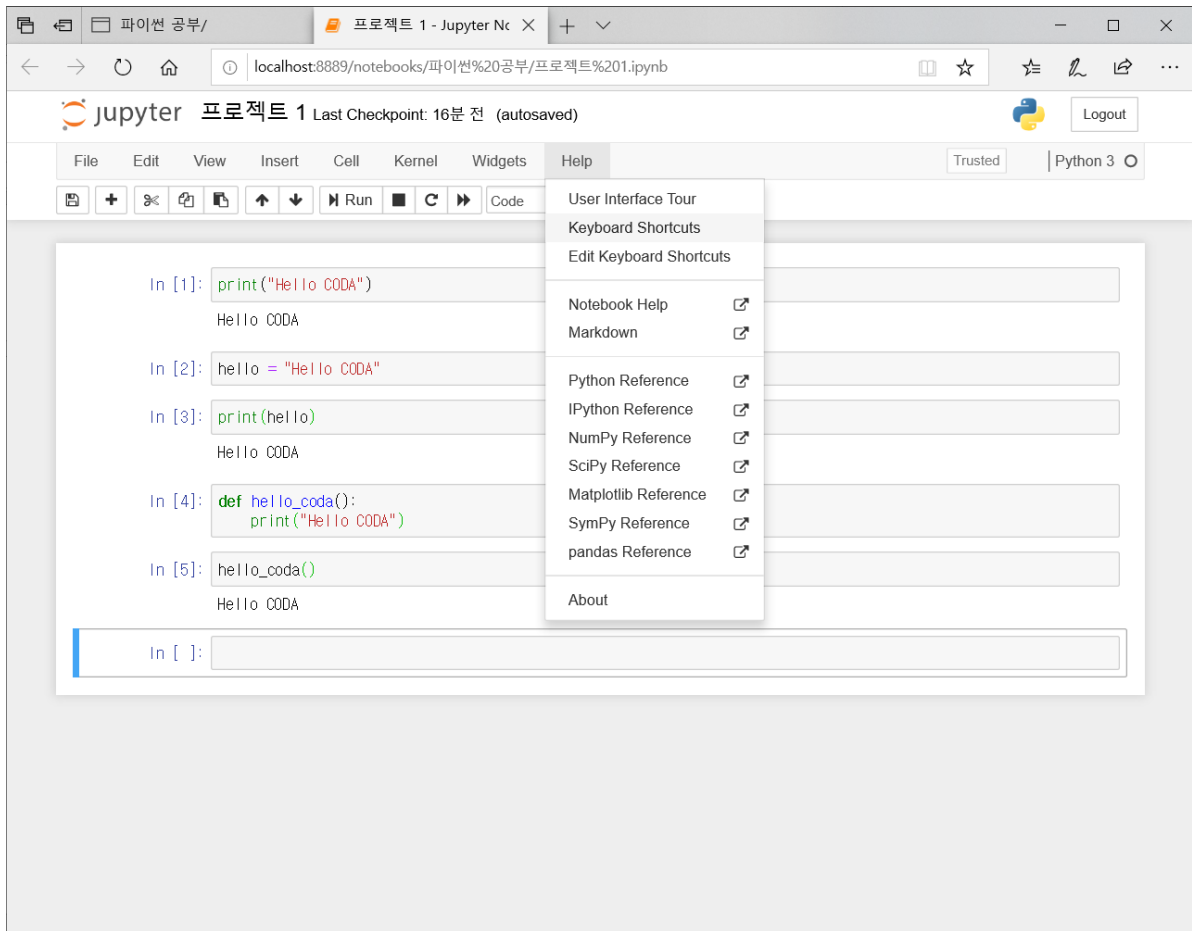
In [ ]:
```

코드 실행 - 3

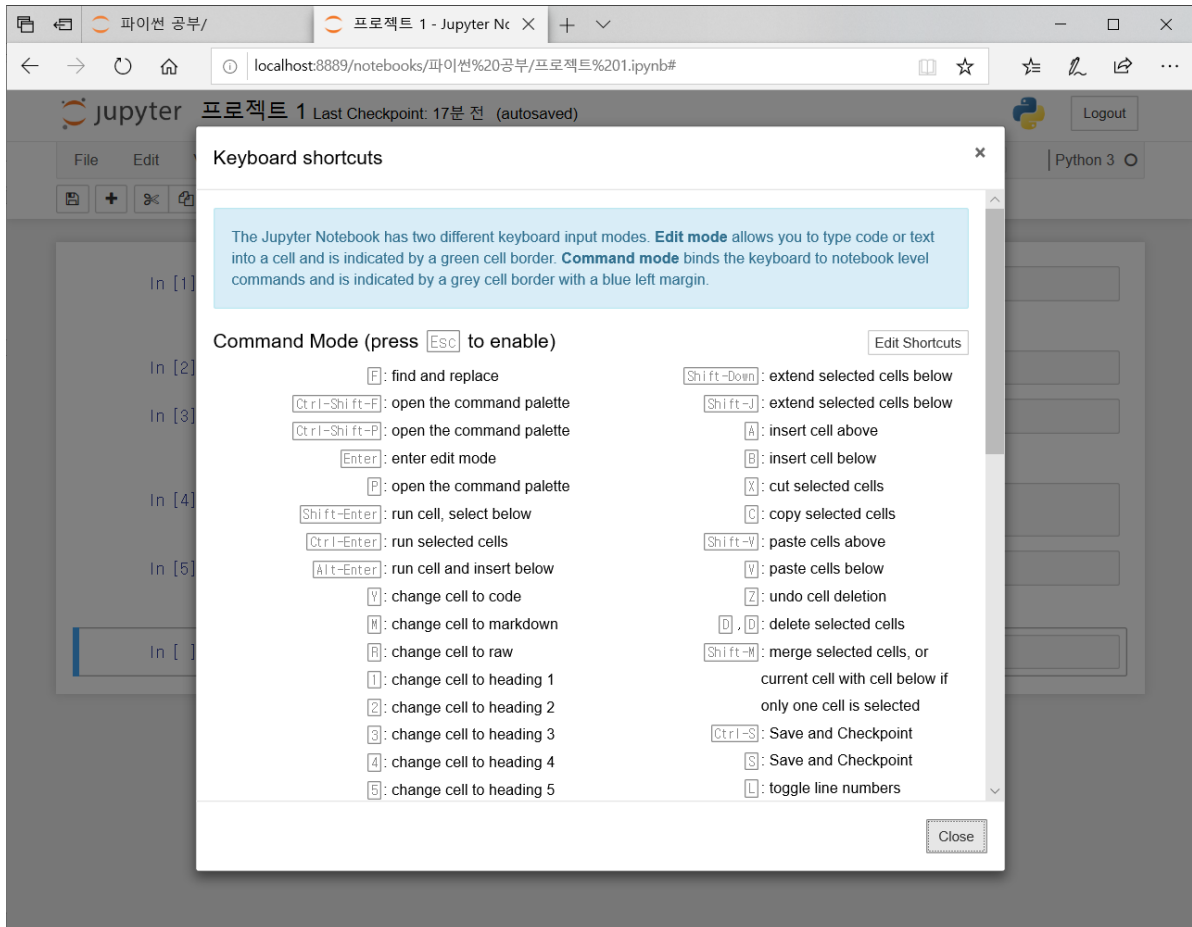
이렇게 주피터 노트북 내부에 정의된 변수나 함수는 같은 다큐먼트에 저장되어 셀의 순서와 상관없이 공유됩니다. 따라서 셀이 바뀌었다고 해서 변수를 다시 정의할 필요가 없습니다.

○ 단축키(Keyboard Shortcuts)

주피터 노트북에서 새로운 셀을 생성하거나, 셀에 입력한 코드를 실행할 때 마우스로 클릭하는 것 보다는 단축키를 사용하는 것이 더욱 효율적이므로 중요한 단축키를 익힌 후 사용하는 것을 권장합니다. 다큐먼트 상단에 [Help]버튼을 클릭하면 생성되는 드롭다운 목록 중에 [Keyboard shortcuts]을 클릭하면 사용 가능한 모든 단축키에 대한 정보를 볼 수 있습니다. 또한 Esc 와 H키를 누르면 단축키에 대한 정보를 바로 볼 수 있습니다.



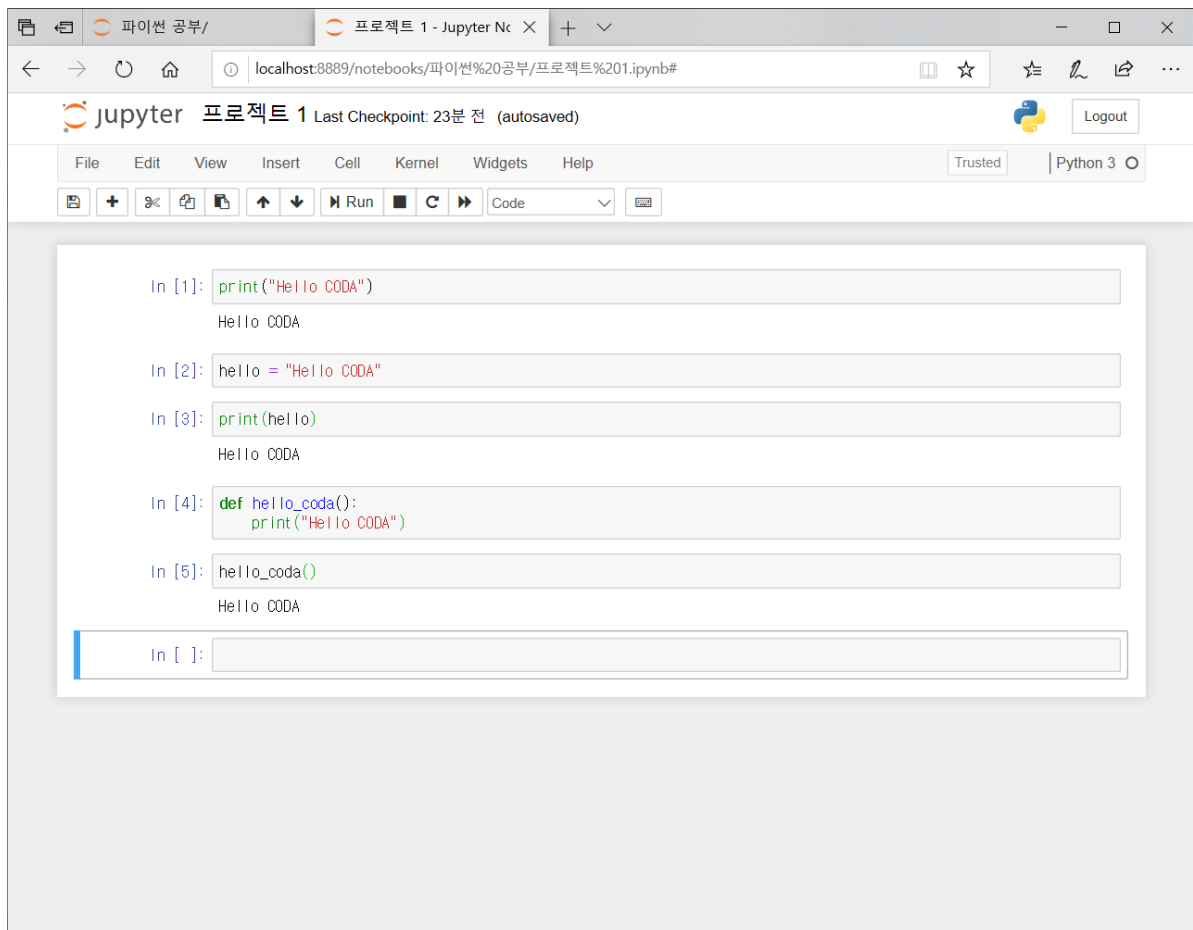
단축키(Keyboard Shortcuts) - 1



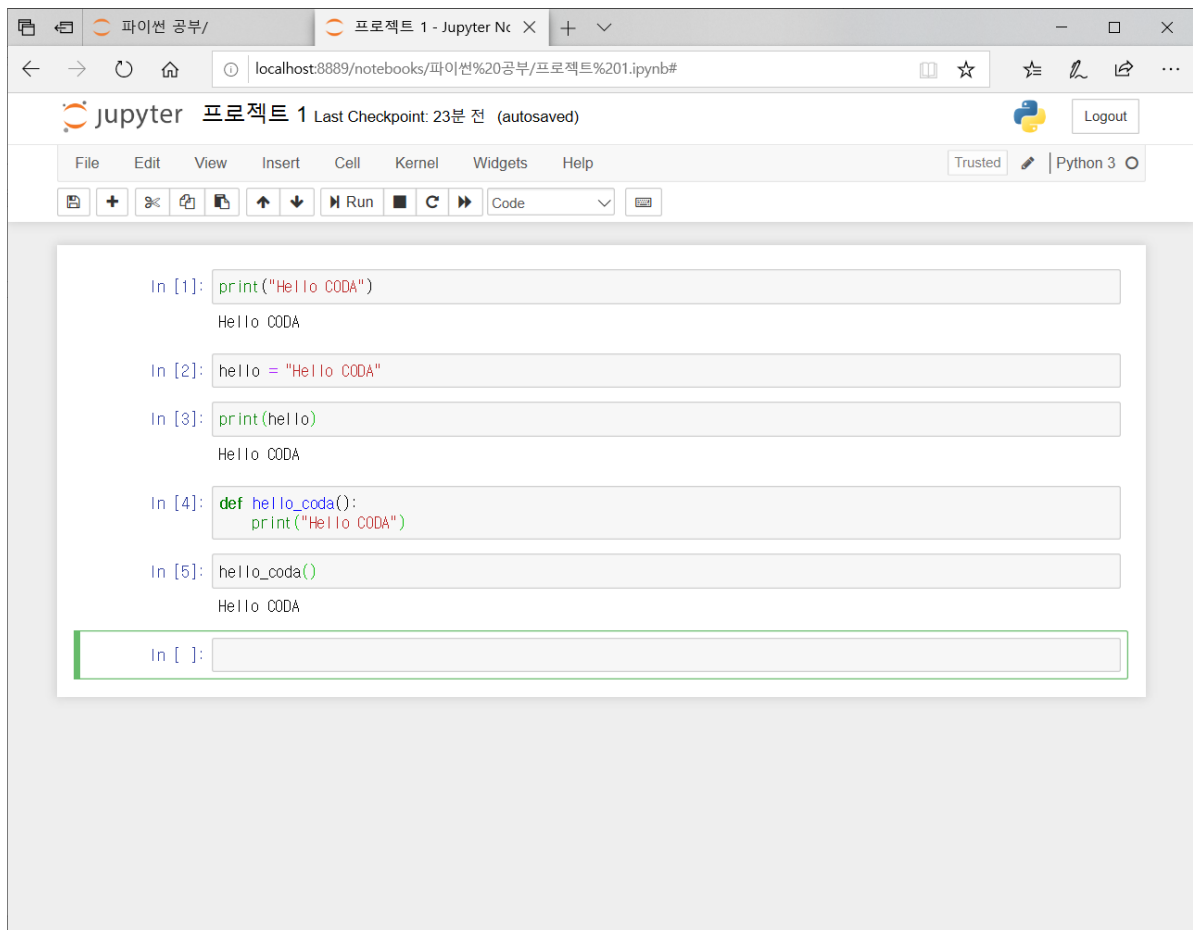
단축키(Keyboard Shortcuts) - 2

○ Edit 모드와 Command 모드

이번에는 셀을 생성하는 방법과 삭제하는 방법에 대해 조금 더 자세하게 알아보도록 하겠습니다. 셀의 모드에는 Edit 모드와 Command 모드가 있습니다. 셀을 선택하면 좌측에 녹색 또는 파란색 띠가 보이는데, 녹색 띠가 Edit 모드, 파란색 띠가 Command 모드입니다.



Command 모드

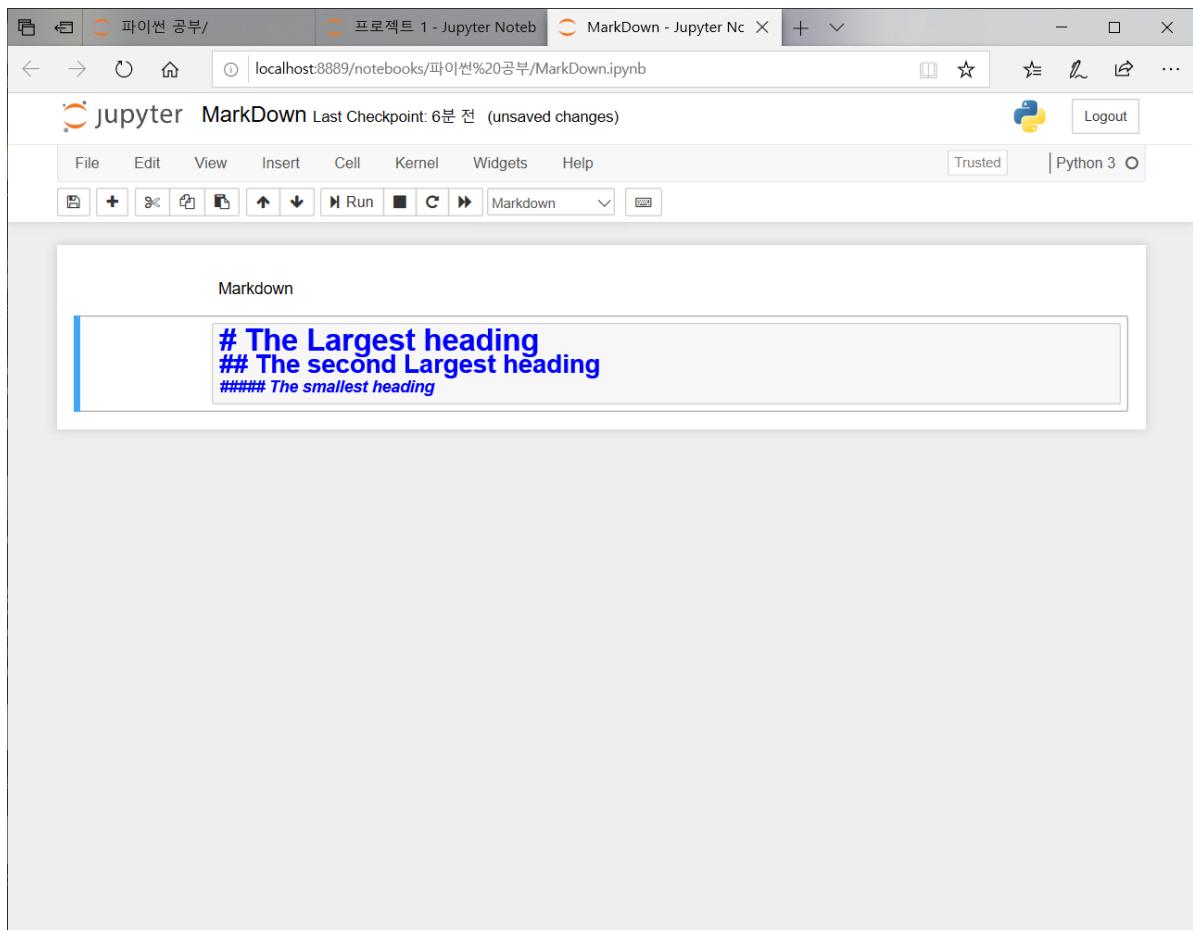


Edit 모드

셀 내부를 클릭하면 Edit 모드가 되며, 말 그대로 셀 안의 문자를 입력/수정할 수 있습니다. 하지만 Edit 모드에서 셀을 생성하거나 삭제하는 단축키를 사용할 수 없으며 단축키를 사용하기 위해서는 Command 단축키로 전환할 필요가 있습니다. 셀 외부 좌측 띠 부분을 클릭하거나 Esc 버튼을 누르면 Command 모드로 전환됩니다.

○ 마크다운(Markdown)

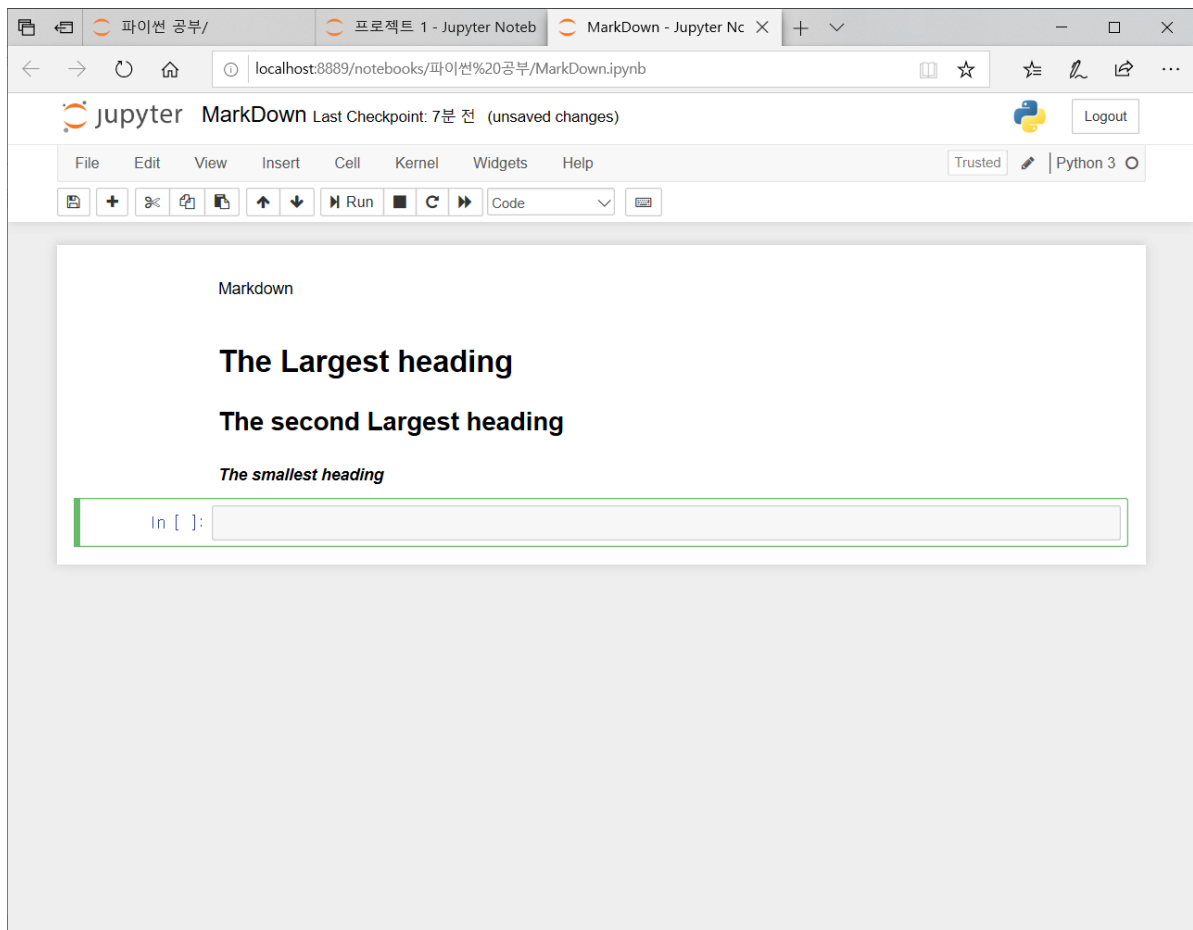
마크다운(Markdown)은 Light-weight Markup Language라고도 불리며, 마크다운을 사용하면 플레인텍스트 형식으로도 간단히 HTML과 같은 위치 텍스트를 만들어 낼 수도 있습니다. 마크다운은 주로 ReadMe 파일을 만드는 데 사용되며 마크다운을 사용하여 근사한 Notebook 프레젠테이션을 만들 수도 있습니다. 먼저 Markdown이라는 이름의 새로운 다큐먼트를 만들겠습니다. 마크다운을 적용할 셀을 선택하고 우측 상단에 보이는 [Code]메뉴를 클릭하면 나타나는 드롭다운 메뉴 중에서 [Markdown]을 클릭하면 Markdown모드로 전환합니다.



마크다운(Markdown) - 1

마크다운 모드로 전환되면 In[]이라는 글자가 사라집니다. 그리고 셀을 실행하면 셀 안에 있는 마크다운 코드가 HTML 형식으로 나타나는 것을 볼 수 있습니다.

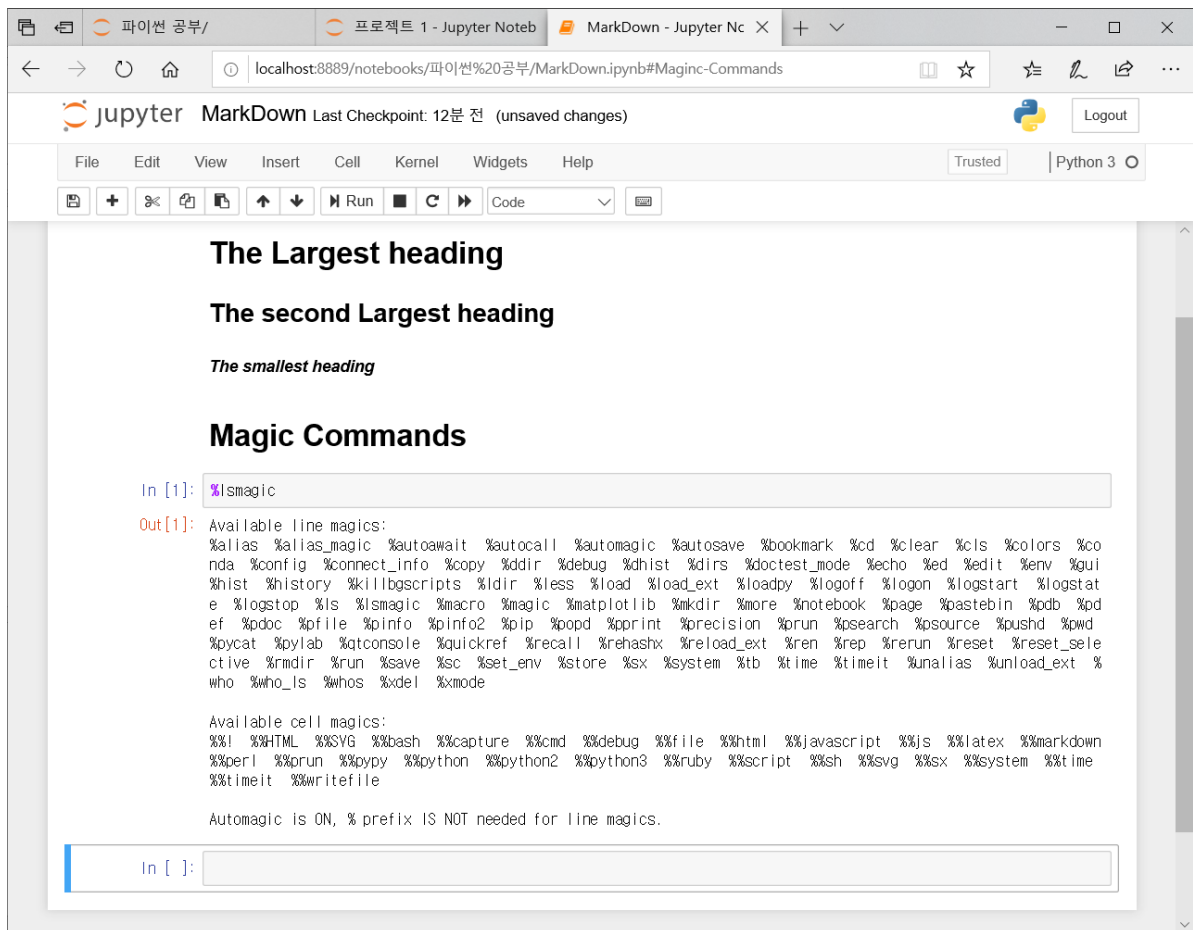
(사진 - 22)



마크다운(Markdown) - 2

○ 매직 커맨드(Magic Commands)

이번에는 IPython이 제공하는 편리한 Magic Commands를 사용해보겠습니다. 매직 커맨드를 사용하면 간단히 현재 폴더의 파일을 리스트하거나 새로운 폴더를 생성할 수 있습니다. 그 밖에도 환경변수를 설정하거나 프로그램의 시간을 간단히 측정하는 기능 등의 많은 편리한 기능들을 사용할 수 있습니다. 매직 커맨드는 [Code]모드에서 실행해야 하며, 사용 가능한 매직 커맨드 기능을 출력해보면 다음과 같습니다.



매직 커맨드(Magic Commands) - 1

- 사용 가능한 OS 커맨드를 출력해보면 다음과 같습니다.

who %who_ls %whos %xdel %xmode

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html %%javascript %%js %%latex %%markdown
%%perl %%prun %%pypy %%python %%python2 %%python3 %%ruby %%script %%sh %%svg %%sx %%system %%time
%%timeit %%writefile

Automagic is ON, % prefix IS NOT needed for line magics.

OS Commands

```
In [2]: %alias
```

Total number of aliases: 8

```
Out[2]: [('copy', 'copy'),  
         ('ddir', 'dir /ad /on'),  
         ('echo', 'echo'),  
         ('ldir', 'dir /ad /on'),  
         ('ls', 'dir /on'),  
         ('mkdir', 'mkdir'),  
         ('ren', 'ren'),  
         ('rmdir', 'rmdir')]
```

```
In [ ]:
```

매직 커맨드(Magic Commands) - 2

- 현재 폴더의 파일 목록 확인은 다음과 같습니다.

Localhost:8889/notebooks/파이썬%20공부/MarkDown.ipynb#Maginc-Commands

Jupyter MarkDown Last Checkpoint: 15분 전 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```

('echo', 'echo'),
('mkdir', 'mkdir'),
('rmkdir', 'rmkdir'),
('ls', 'ls'),
('rmdir', 'rmdir'),
('ren', 'ren')

```

현재 폴더의 파일 리스트

In [3]: %ls

C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: A6B1-8ADD

C:\Users\lajet\파이썬 공부 디렉터리

날짜	시간	타입	이름	크기
2019-08-15	오후 04:50	<DIR>	.	
2019-08-15	오후 04:50	<DIR>	..	
2019-08-15	오후 04:36	<DIR>	.ipynb_checkpoints	
2019-08-15	오후 04:50		MarkDown.ipynb	7,190
2019-08-15	오후 04:35		프로젝트 1.ipynb	1,620
		2개 파일		8,810 바이트
		3개 디렉터리		92,711,444,480 바이트 남음

In []:

매직 커맨드(Magic Commands) - 3

- 새로운 폴더를 생성은 다음과 같습니다**. **

The screenshot shows a Jupyter Notebook interface with a file explorer view. The notebook is titled "MarkDown - Jupyter Notebook" and is running on "localhost:8889". The file explorer shows a directory structure with files like "MarkDown.ipynb", "프로젝트 1.ipynb", and a newly created folder "New_Folder". The terminal output shows the execution of "mkdir New_Folder" and "ls" commands, confirming the folder's creation.

매직 커맨드(Magic Commands) - 4

***•* 환경 변수 설정과 참조는 다음과 같습니다.**

The screenshot shows a Jupyter Notebook interface with the following content:

File Explorer Output:

```
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: A6B1-8ADD

C:\Users\lajet\파이썬 공부 디렉터리

2019-08-15 오후 04:52 <DIR> .
2019-08-15 오후 04:52 <DIR> ..
2019-08-15 오후 04:36 <DIR> .ipynb_checkpoints
2019-08-15 오후 04:52 8,193 Markdown.ipynb
2019-08-15 오후 04:52 <DIR> New_Forder
2019-08-15 오후 04:35 1,620 프로젝트 1.ipynb
                2개 파일 9,813 바이트
                4개 디렉터리 92,713,406,464 바이트 남음
```

Environment Variable Setting and Reference:

```
In [5]: # with magic command
%env IS_DOCKER_ENV=true
%env IS_DOCKER_ENV

env: IS_DOCKER_ENV=true

Out[5]: 'true'
```

The interface also shows a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The status bar indicates the kernel is Python 3 and the notebook is trusted.

매직 커맨드(Magic Commands) - 5

● 프로그램 시간 측정은 다음과 같습니다.

2019-08-15 오후 04:52 <DIR> New_Forder
 2019-08-15 오후 04:35 1,620 프로젝트 1.ipynb
 2개 파일 9,813 바이트
 4개 디렉터리 92,713,406,464 바이트 남음

환경변수 설정 및 참조

```
In [5]: # with magic command
%env IS_DOCKER_ENV=true
%env IS_DOCKER_ENV

env: IS_DOCKER_ENV=true
Out[5]: 'true'
```

프로그램 시간 측정

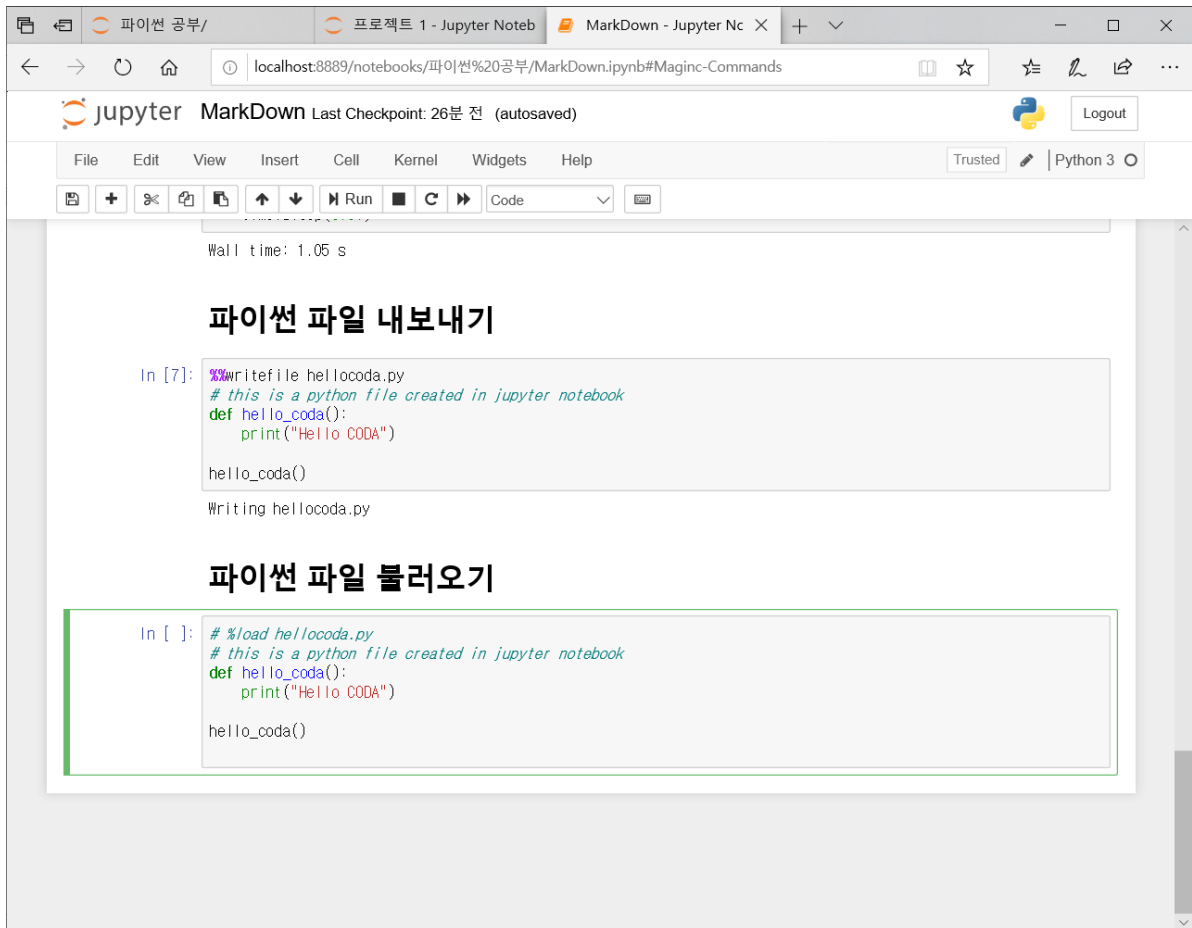
```
In [6]: %%time
import time
for _ in range(100):
    time.sleep(0.01)

Wall time: 1.05 s
```

```
In [ ]:
```

매직 커맨드(Magic Commands) - 6

****** 주피터 노트북에서 작성한 코드를 외부 파일로 내보내기 하거나 외부 파이썬 파일을 불러올 경우는 다음과 같습니다.



매직 커맨드(Magic Commands) - 7

○ Pandas

이번에는 과학분야와 데이터 사이언스 분야에서 많이 사용되고 있는 Pandas 라이브러리에 대해 간단하게 설명하도록 하겠습니다. Pandas는 Excel로는 다룰 수 없는 빅 데이터와 같은 크고 복잡한 데이터를 분석할 수 있으며, Excel, CSV, JSON 형식의 데이터를 불러와서 DataFrame이라는 형식으로 만들어 분석할 수 있는 편리한 라이브러리입니다. Pandas에 대한 자세한 설명은 추후에 따로 포스팅하도록 하겠습니다.

○ 차트 라이브러리의 사용 방법

쥬피터 노트북을 사용하는 이유 중에 빼놓을 수 없는 중요한 기능 중 하나가 차트입니다. 쥬피터 노트북에서 사용할 수 있는 차트 라이브러리는 다음과 같습니다.

matplotlib, Seaborn, ggplot, Bokeh, pygal, Plotly

이 중에서 Plotly 라이브러리를 예시로 사용해보도록 하겠습니다. Plotly 라이브러리는 Anaconda 파이썬에 기본 라이브러리가 아니므로 Package Manager를 사용하여 라이브러리를 설치하도록 하겠습니다.

(1) Plotly 라이브러리 설치

윈도우 검색 창에 cmd를 입력하여 명령 프롬프트를 실행하고, pip install plotly를 입력하여 라이브러리를 설치하도록 합니다.

```
선택 명령 프롬프트
Microsoft Windows [Version 10.0.18362.295]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\lajet>pip install plotly
Collecting plotly
  Downloading https://files.pythonhosted.org/packages/63/2b/4ca10995bfbdefd65c4238f9a2d3fde33705d18dd50914dd13302ec1daf1
/plotly-4.1.0-py2.py3-none-any.whl (7.1MB)
    | 7.1MB 1.3MB/s
Requirement already satisfied: six in c:\users\lajet\anaconda3\lib\site-packages (from plotly) (1.12.0)
Collecting retrying>=1.3.3 (from plotly)
  Downloading https://files.pythonhosted.org/packages/44/ef/beae4b4ef80902f22e3af073397f079c96969c69b2c7d52a57ea9ae61c9d
/retrying-1.3.3.tar.gz
Building wheels for collected packages: retrying
  Building wheel for retrying (setup.py) ... done
  Stored in directory: C:\Users\lajet\AppData\Local\pip\Cache\wheels\d7#a9#33#acc7b709e2a35caa7d4cae442f6fe6fbf2c43f8082
3d46460c
Successfully built retrying
Installing collected packages: retrying, plotly
Successfully installed plotly-4.1.0 retrying-1.3.3

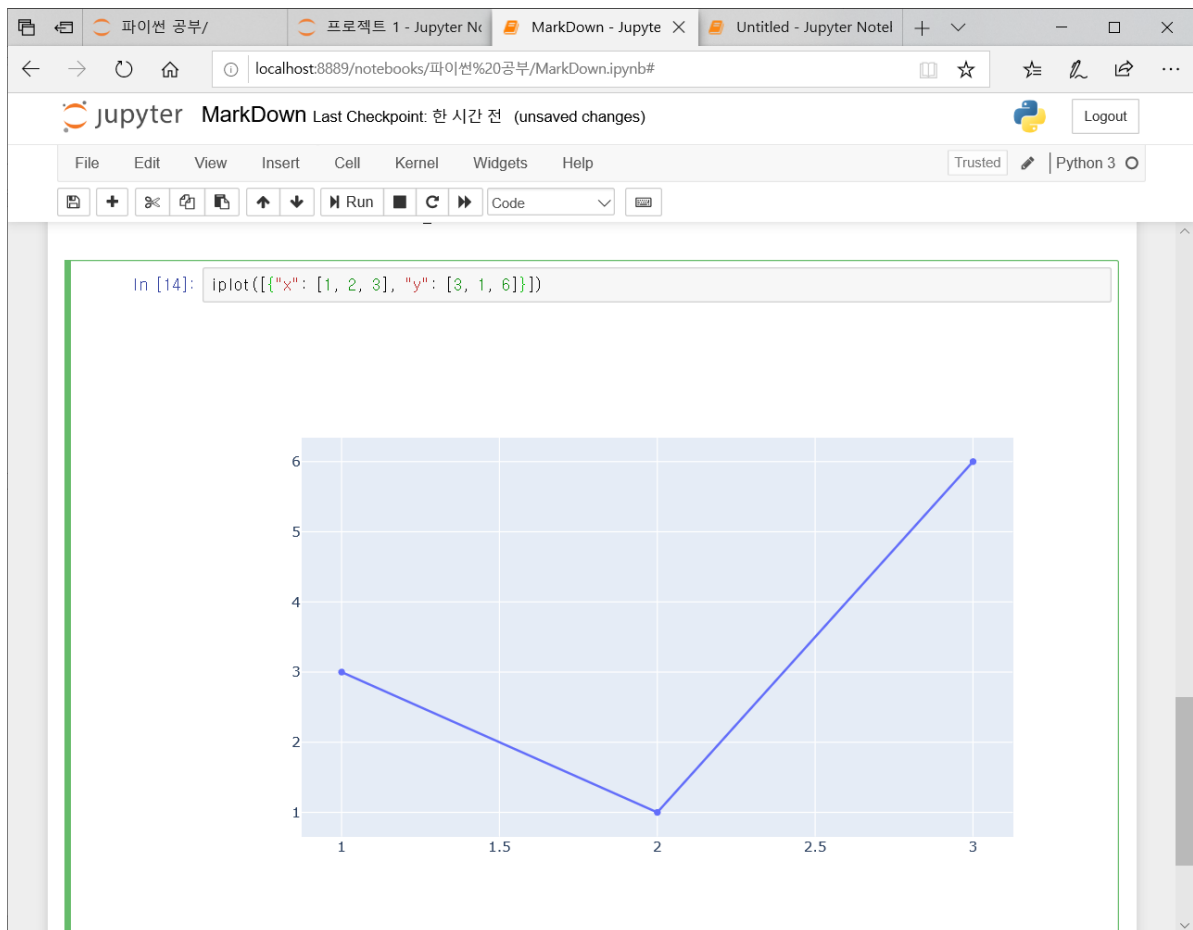
C:\Users\lajet>
C:\Users\lajet>
C:\Users\lajet>
C:\Users\lajet>
```

Plotly 라이브러리 설치 - 1

(2) Plotly 라이브러리 실행

다시 주피터 노트북으로 돌아와서 Plotly 라이브러리를 참조한 코드를 실행하면 다음과 같습니다.

```
from plotly import __version__
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
iplot([{"x": [1, 2, 3], "y": [3, 1, 6]}])
```



Plotly 라이브러리 설치 - 2

○ 설정 변경 방법

쥬피터 노트북의 설정을 위해서는 쥬피터 폴더에 `jupyter_note_config.py` 파일을 생성해야 합니다. 쥬피터 폴더는 유저 홈 폴더 아래의 `.jupyter`입니다. 설정 파일 생성 명령어는 명령 프롬프트 창에서 다음과 같이 입력하면 됩니다.

```
jupyter notebook --generate-config
```

아래와 같이 설정 파일이 생성된 것을 볼 수 있습니다.

```
명령 프롬프트

The directory to use for notebooks and kernels.
--browser=<Unicode> (NotebookApp.browser)
Default: ''
Specify what command to use to invoke a web browser when opening the
notebook. If not specified, the default browser will be determined by the
`webbrowser` standard library module, which allows setting of the BROWSER
environment variable to override it.
--pylab=<Unicode> (NotebookApp.pylab)
Default: 'disabled'
DISABLED: use %pylab or %matplotlib in the notebook to enable matplotlib.
--gateway-url=<Unicode> (GatewayClient.url)
Default: None
The url of the Kernel or Enterprise Gateway server where kernel
specifications are defined and kernel management takes place. If defined,
this Notebook server acts as a proxy for all kernel management and kernel
specification retrieval. (JUPYTER_GATEWAY_URL env var)

To see all available configurables, use `--help-all`

Examples
-----

jupyter notebook                # start the notebook
jupyter notebook --certfile=mycert.pem # use SSL/TLS certificate
jupyter notebook password      # enter a password to protect the server

[C 17:34:32.453 NotebookApp] Bad config encountered during initialization:
[C 17:34:32.454 NotebookApp] Unrecognized flag: '--ganerate-config'

C:\Users\lajet>
```

설정 변경 방법 - 1

생성된 설정 파일은 기본 설정 파일로써 모든 항목들이 주석처리되어 있으나 각각의 항목에 대한 기본 설정 값을 표시하고 있습니다. 주피터 노트북 서비스를 서버 상에서 서비스를 제공할 목적이 아니고, 학습을 목적으로 사용한다면 주피터 노트북이 사용할 기본 설정 포트 번호, 주피터 노트북이 실행되는 폴더, 주피터 노트북이 버퍼로써 사용할 수 있는 메모리 제한 정도만 변경한 후 사용하면 됩니다.