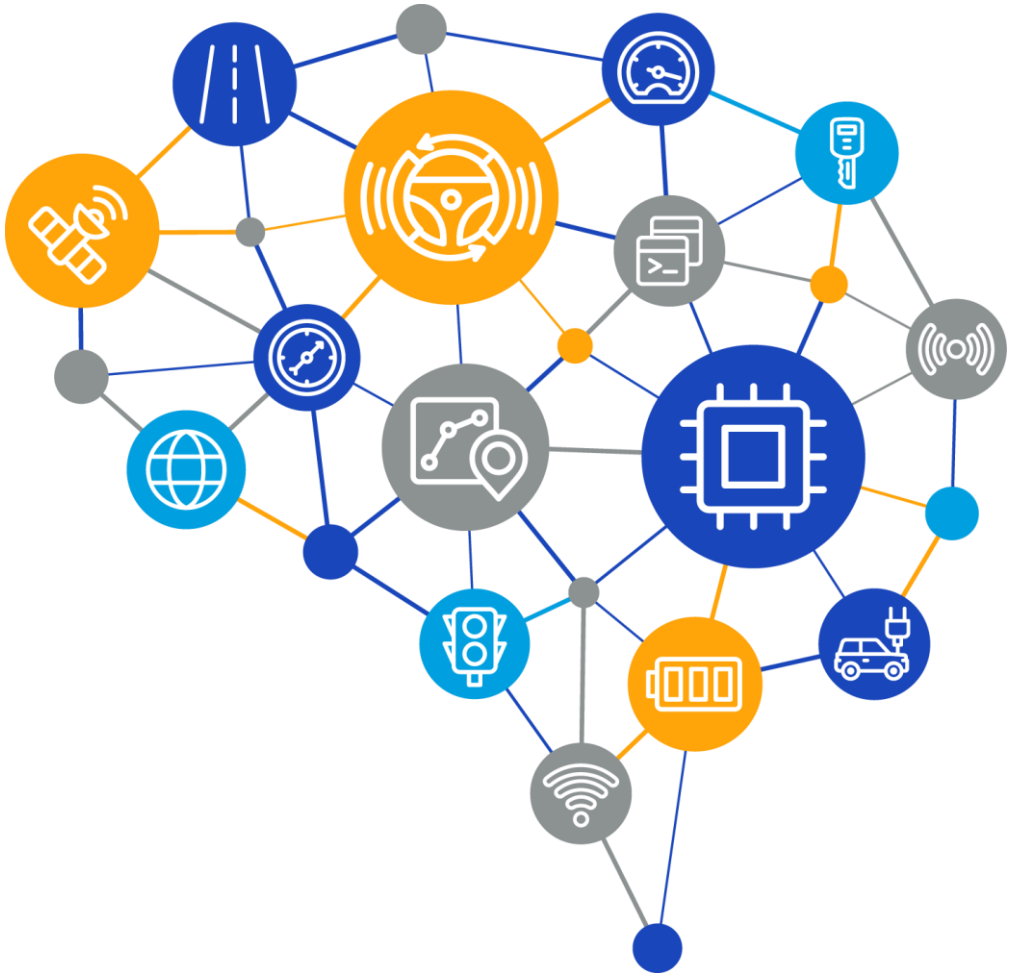
파이참과 주피터노트북

Git & Github
for Open Source Software



2022.07

강환수 교수

DMUAi 동양미래대학교 인공지능소프트웨어학과

AI Experts Who Lead
The Future

AI Experts
Who Lead
The Future

01

주피터 노트북 모듈 설치 방법

모듈로 주피터 노트북 이용 방법

- **모듈 설치 방법 2가지**
  - pip install jupyter
  - pip install notebook

- **실행 방법 2가지**
  - jupyter notebook
  - python –m notebook

- **실습방법**
  - D:₩jupynote 하부에서
  - Venv로 가상환경 jupylab과 nblab 두 개를 만들어
  - 각각 jupyter와 notebook을 설치
    - > pip install jupyter
    - > pip install notebook

  - 각각 다음 2가지 방법으로 실행해 보기
    - > python –m notebook
    - > jupyter notebook

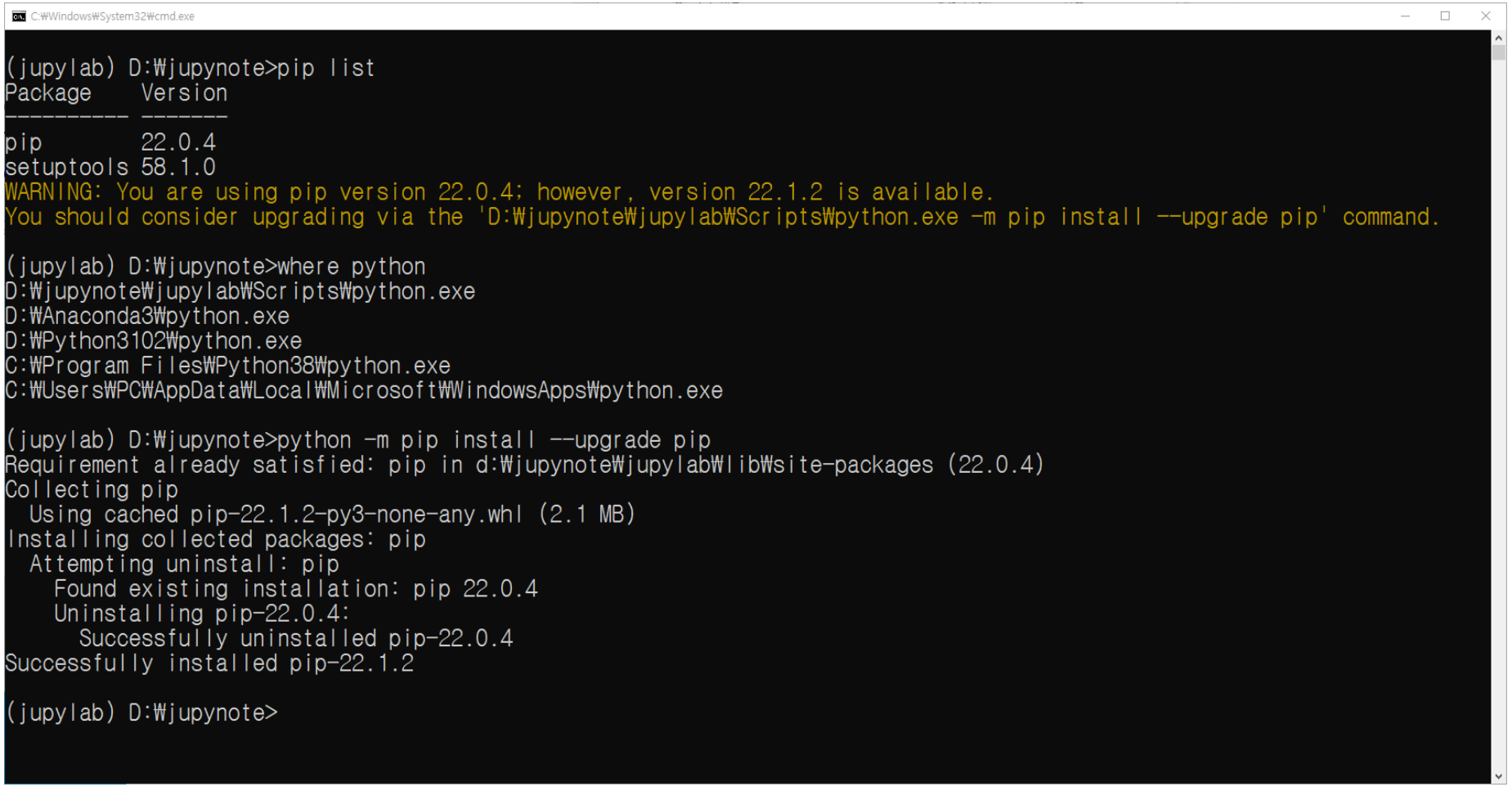
실습1: 가상환경을 만들어 jupyter 설치 후 실행

- **D:₩jupynote에서 cmd 실행**
- **venv로 가상환경 jupylab 생성**

```
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

D:₩jupynote>python –m venv jupylab

D:₩ jupynote >jupylab₩scripts₩activate

(jupylab) D:₩jupynote>
```

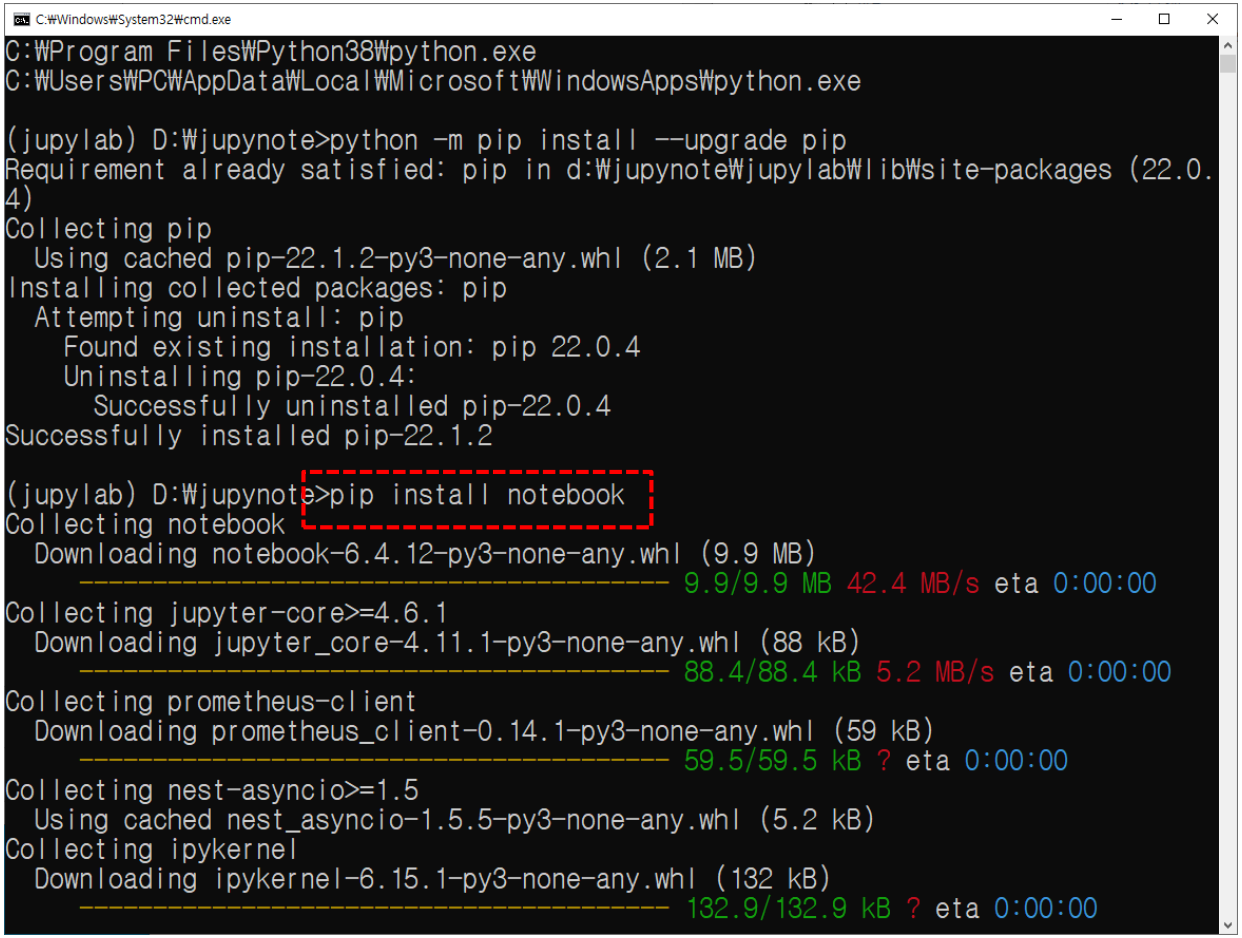
- **Pip upgrade**
  - **> python -m pip install --upgrade pip**

가상환경 활성화와 pip upgrade



모듈 jupyter를 설치

- 다음
  - pip install jupyter
    - Jupyter로 설치하면 약간 더 많이 설치



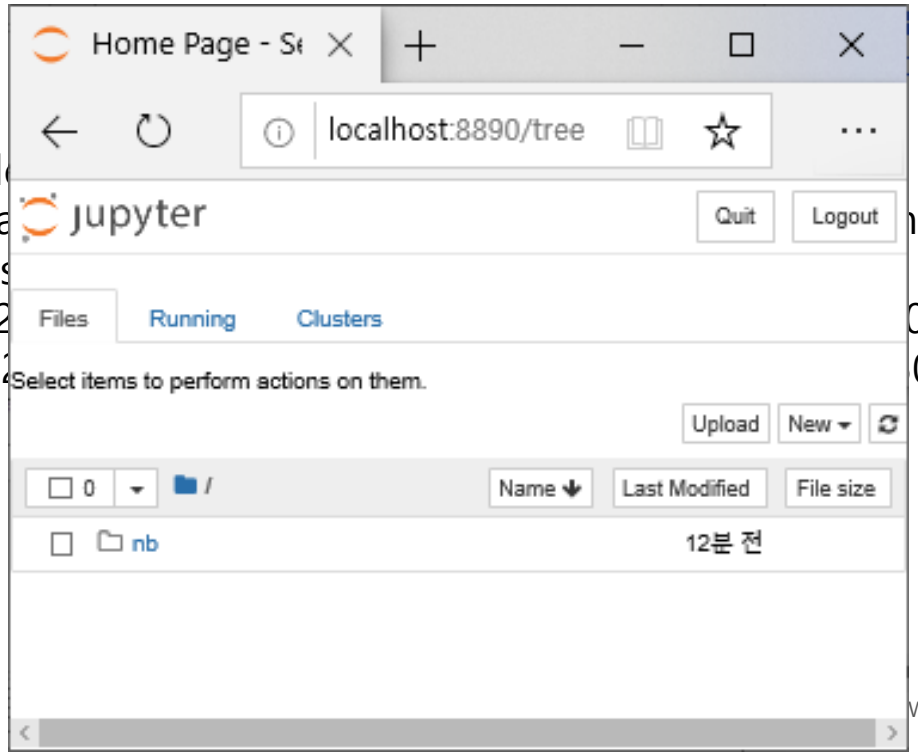
실습1: 노트북 설치

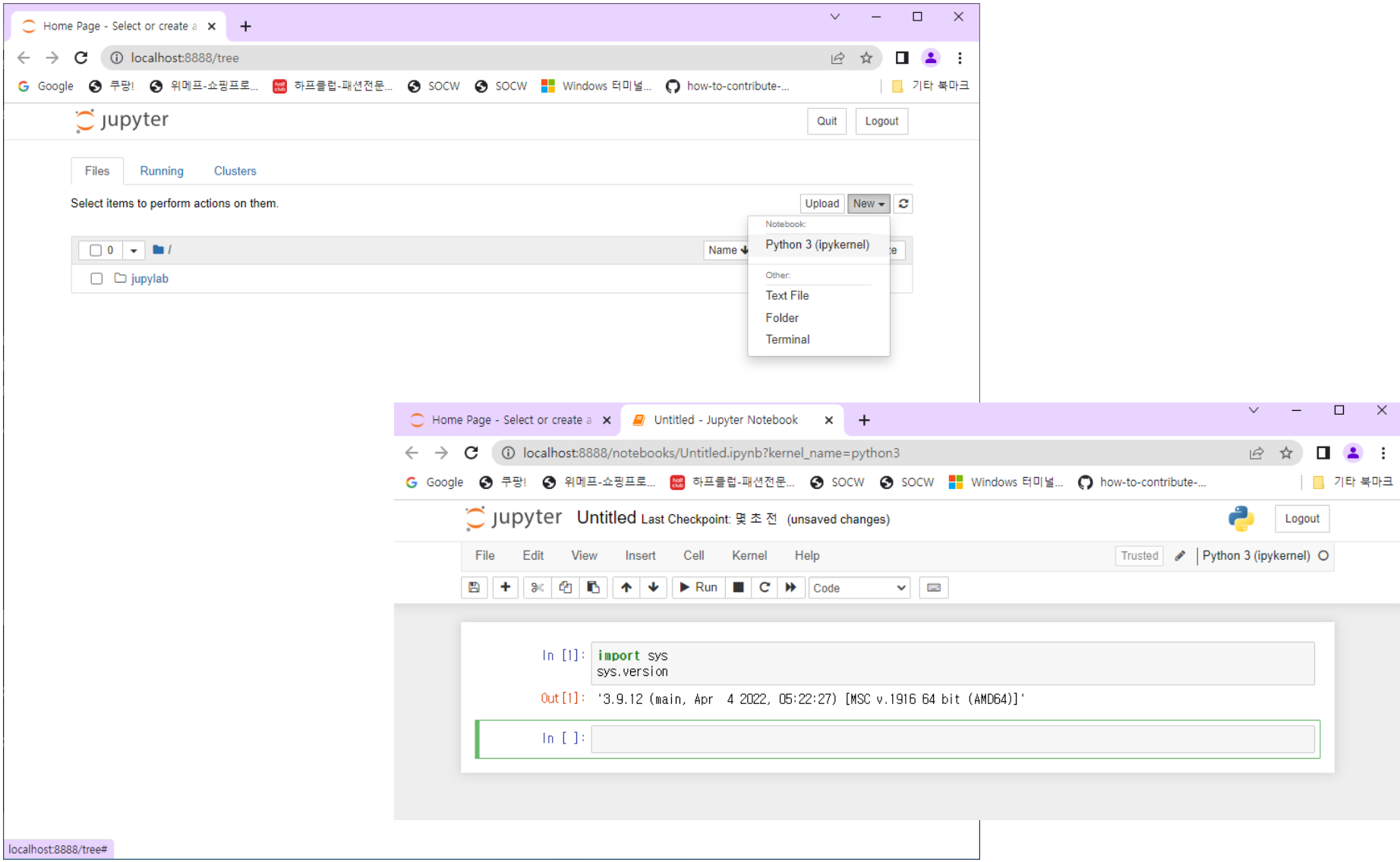

```
(nb) D:₩myVE>pip install notebook
Collecting notebook
  Downloading notebook-6.0.3-py3-none-any.whl (9.7 MB)
     |████████████████████████████████| 9.7 MB 504 kB/s
Collecting terminado>=0.8.1
  Downloading terminado-0.8.3-py2.py3-none-any.whl (33 kB)
Collecting traitlets>=4.2.1
  Downloading traitlets-4.3.3-py2.py3-none-any.whl (75 kB)
     |████████████████████████████████| 75 kB ...

...

Successfully built prometheus-client pandocfilters pyrsistent backcall
Installing collected packages: pywinpty, tornado, terminado, decorator, ipython-genutils, six, traitlets, pyzmq, pywin32,
jupyter-core, python-dateutil, jupyter-client, MarkupSafe, jinja2, more-itertools, zipp, importlib-metadata, pyrsistent,
attrs, jsonschema, nbformat, Send2Trash, pygments, parso, jedi, backcall, colorama, wcwidth, prompt-toolkit,
pickleshare, ipython, ipykernel, pandocfilters, testpath, entrypoints, mistune, defusedxml, webencodings, bleach,
nbconvert, prometheus-client, notebook
Successfully installed MarkupSafe-1.1.1 Send2Trash-1.5.0 attrs-19.3.0 backcall-0.1.0 bleach-3.1.0 colorama-0.4.3
decorator-4.4.1 defusedxml-0.6.0 entrypoints-0.3 importlib-metadata-1.4.0 ipykernel-5.1.3 ipython-7.11.1 ipython-
genutils-0.2.0 jedi-0.15.2 jinja2-2.10.3 jsonschema-3.2.0 jupyter-client-5.3.4 jupyter-core-4.6.1 mistune-0.8.4 more-
itertools-8.1.0 nbconvert-5.6.1 nbformat-5.0.4 notebook-6.0.3 pandocfilters-1.4.2 parso-0.5.2 pickleshare-0.7.5
prometheus-client-0.7.1 prompt-toolkit-3.0.2 pygments-2.5.2 pyrsistent-0.15.7 python-dateutil-2.8.1 pywin32-227
pywinpty-0.5.7 pyzmq-18.1.1 six-1.14.0 terminado-0.8.3 testpath-0.4.4 tornado-6.0.3 traitlets-4.3.3 wcwidth-0.1.8
webencodings-0.5.1 zipp-2.0.1

(nb) D:₩myVE>
```

실습1: 노트북 실행 1

(nb) D:₩myVE>python -m notebook
[I 02:32:12.577 NotebookApp] The port 8888 is already in use, trying another port.
[I 02:32:12.578 NotebookApp] The port 8889 is already in use, trying another port.
[I 02:32:12.592 NotebookApp] Serving notebooks from local directory: D:₩myVE
[I 02:32:12.592 NotebookApp] The Jupyter Notebook is running at:
[I 02:32:12.593 NotebookApp]
http://localhost:8890/?token=93225b8c2a78a4dcfdac04c7066981153f50037ef153501c
[I 02:32:12.594 NotebookApp] or
http://127.0.0.1:8890/?token=93225b8c2a78a4dcfdac04c7066981153f50037ef153501c
[I 02:32:12.594 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 02:32:12.863 NotebookApp]

To access the notebook, open this fil
file:///C:/Users/USER/AppData/Roa
Or copy and paste one of these URLs
http://localhost:8890/?token=9322
or http://127.0.0.1:8890/?token=932



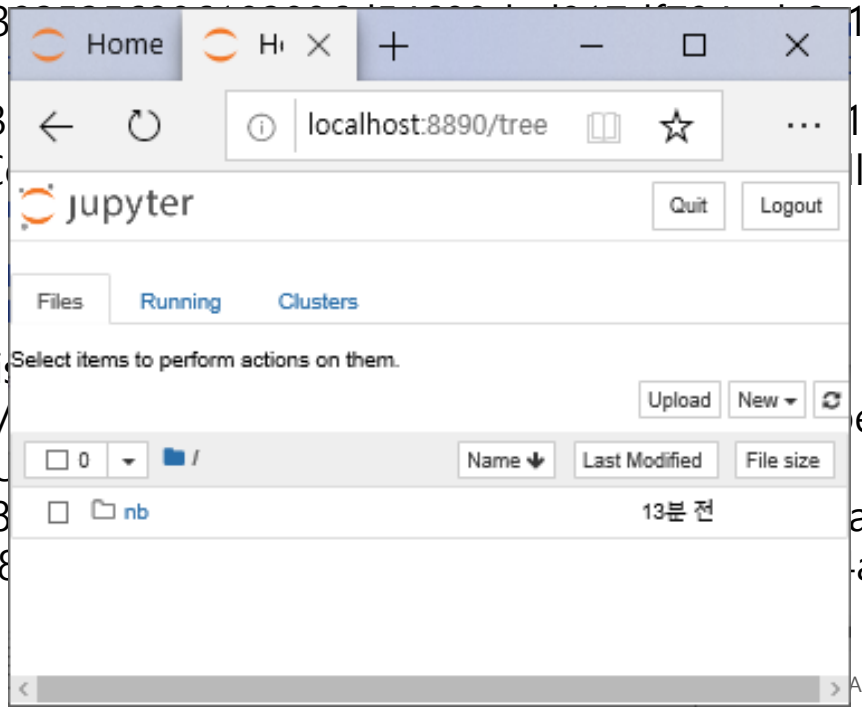

Home Page - Select or create a
localhost:8888/tree
jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
Notebook:
Python 3 (ipykernel)
Other:
Text File
Folder
Terminal
jupylab
Untitled - Jupyter Notebook
localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3
Untitled Last Checkpoint: 몇 초 전 (unsaved changes)
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code
In [1]: import sys
sys.version
Out[1]: '3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]'
In [ ]:
localhost:8888/tree#

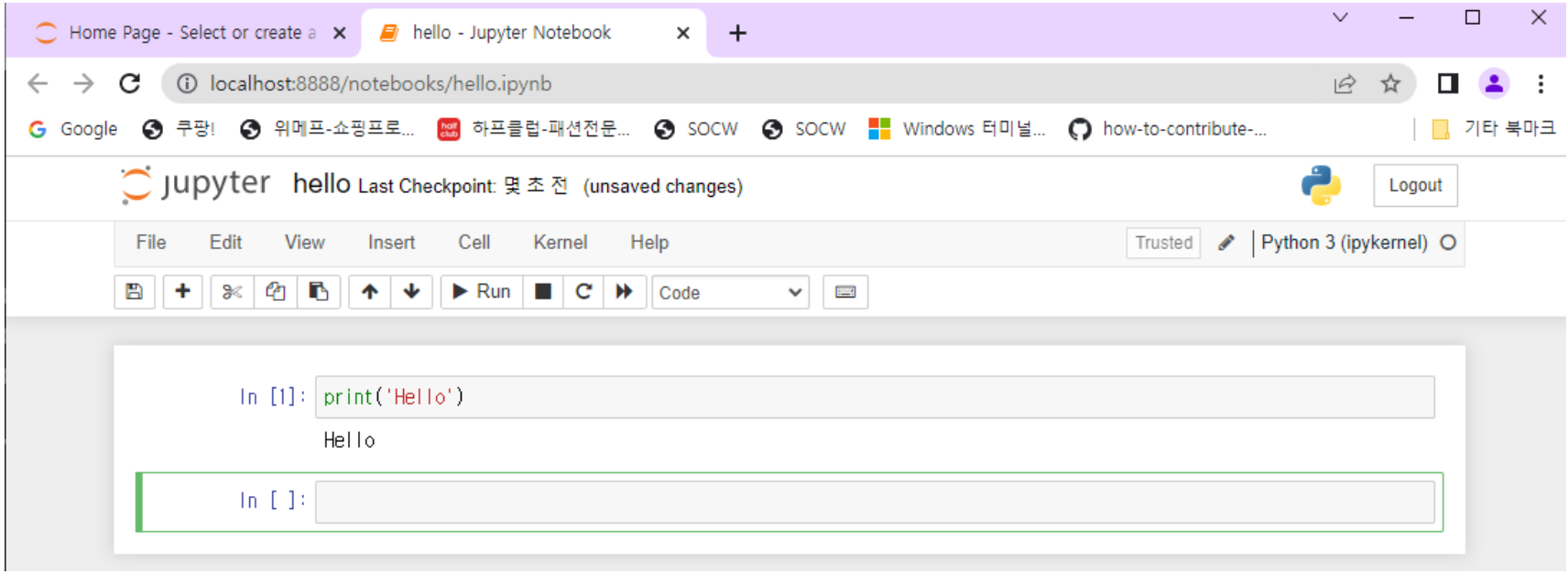
실습1: 노트북 실행 2

(nb) D:₩myVE>where jupyter
D:₩myVE₩nb₩Scripts₩jupyter.exe
C:₩Python₩Anaconda3₩Scripts₩jupyter.exe

(nb) D:₩myVE>jupyter notebook
[I 02:35:53.067 NotebookApp] The port 8888 is already in use, trying another port.
[I 02:35:53.068 NotebookApp] The port 8889 is already in use, trying another port.
[I 02:35:53.082 NotebookApp] Serving notebooks from local directory: D:₩myVE
[I 02:35:53.082 NotebookApp] The Jupyter Notebook is running at:
[I 02:35:53.083 NotebookApp]
http://localhost:8890/?token=895033 1d045
[I 02:35:53.084 NotebookApp] or
http://127.0.0.1:8890/?token=895033 1d045
[I 02:35:53.084 NotebookApp] Use C l kernels (twice
to skip confirmation).
[C 02:35:53.331 NotebookApp]

To access the notebook, open thi
file:///C:/Users/USER/AppData/ en.html
Or copy and paste one of these U
http://localhost:8890/?token=8 aeb6a1d045
or http://127.0.0.1:8890/?token=8 aeb6a1d045




Home Page - Select or create a
hello - Jupyter Notebook
localhost:8888/notebooks/hello.ipynb
Google
쿠팡!
위메프-쇼핑프로...
하프클럽-패션전문...
SOCW
SOCW
Windows 터미널...
how-to-contribute-...
기타 북마크
jupyter
hello Last Checkpoint: 몇 초 전 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3 (ipykernel)
Run
Code
In [1]:
print('Hello')
Hello
In [ ]:

실습2: 가상환경을 만들어 노트북 설치 후 실행

- **venv로 가상환경 nblab 생성**

```
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

D:₩myVE>python –m venv nblab

D:₩myVE>nblab₩scripts₩activate

(nblab) D:₩myVE>
```

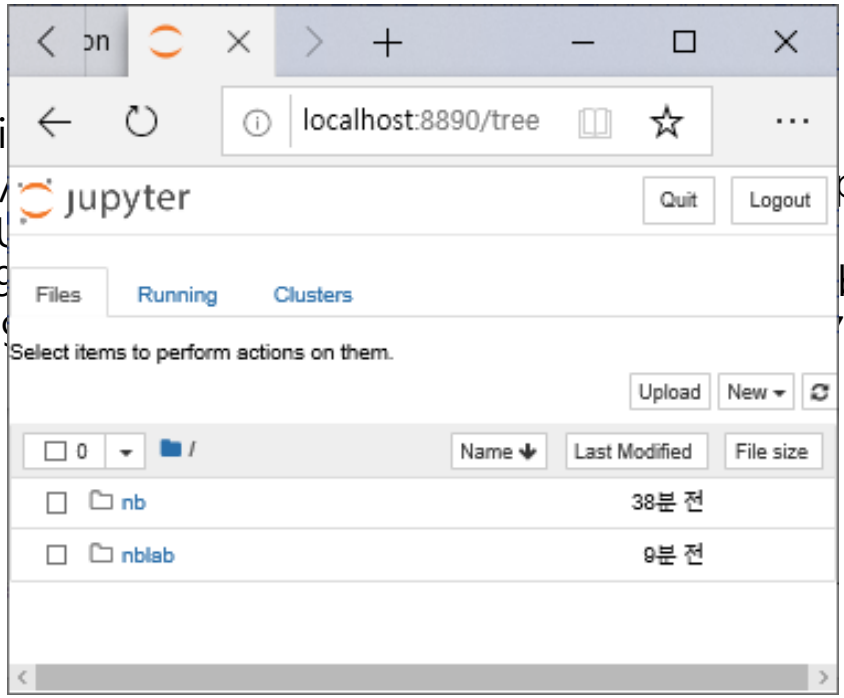
실습2: 노트북 설치 jupyter로

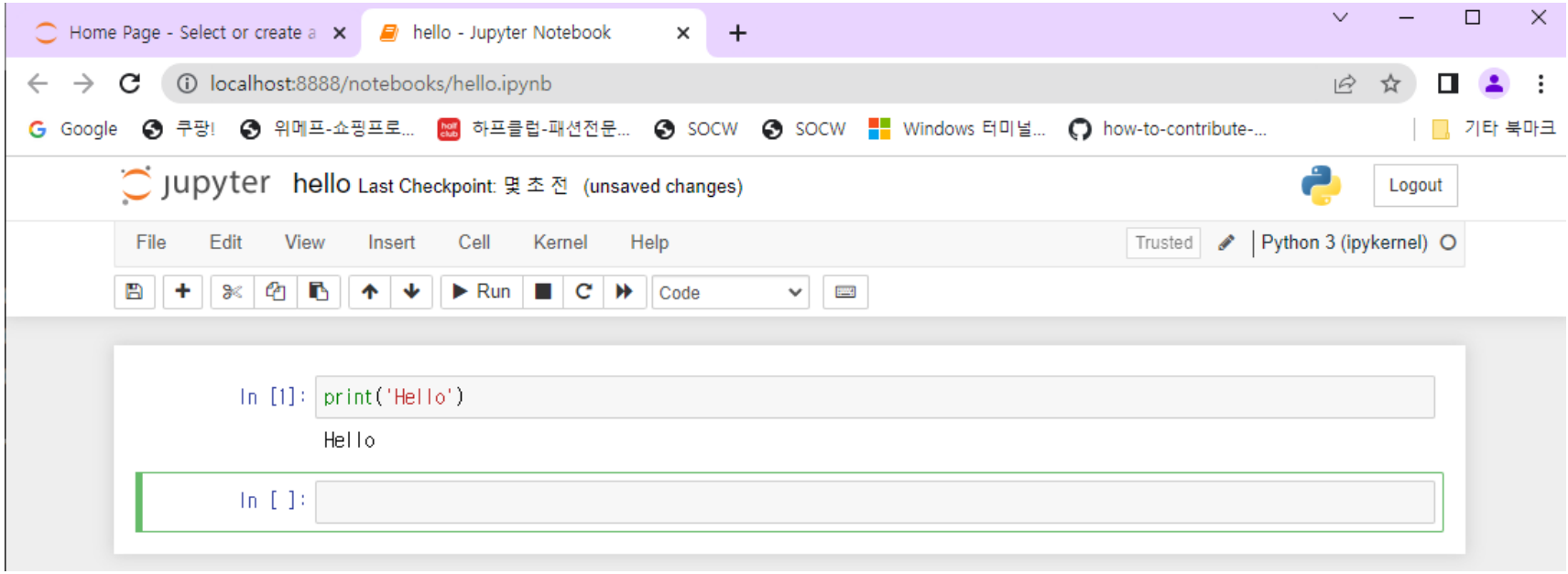

```
(nblab) D:₩myVE>pip list
Package    Version
---------- -------
pip        10.0.1
setuptools 39.0.1
You are using pip version 10.0.1, however version 20.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

(nblab) D:₩myVE>python -m pip install --upgrade pip
Collecting pip
  Using cached https://files.pythonhosted.org/packages/57/36/67f809c135c17ec9b8276466cc57f35b98c240f55c780689ea29fa32f512/pip-20.0.1-py2.py3-none-any.whl
Installing collected packages: pip
  Found existing installation: pip 10.0.1
    Uninstalling pip-10.0.1:
      Successfully uninstalled pip-10.0.1
Successfully installed pip-20.0.1

(nblab) D:₩myVE>pip list
Package    Version
---------- -------
pip        20.0.1
setuptools 39.0.1

(nblab) D:₩myVE>pip install jupyter
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting qtconsole
  Downloading qtconsole-4.6.0-py2.py3-none-any.whl (121 kB)
     |████████████████████████████████| 121 kB 226 kB/s
Collecting ipywidgets
  Downloading ipywidgets-7.5.1-py2.py3-none-any.whl (121 kB)
     |████████████████████████████████| 121 kB ...

...

Installing collected packages: ipython-genutils, decorator, six, traitlets, pywin32, jupyter-core, python-dateutil, pyzmq, tornado, jupyter-client, parso, jedi, wcwidth, prompt-toolkit, colorama, pygments, pickleshare, backcall, ipython, ipykernel, qtconsole, pywinpty, terminado, MarkupSafe, jinja2, Send2Trash, more-itertools, zipp, importlib-metadata, pyrsistent, attrs, jsonschema, nbformat, webencodings, bleach, defusedxml, entrypoints, pandocfilters, testpath, mistune, nbconvert, prometheus-client, notebook, widgetsnbextension, ipywidgets, jupyter-console, jupyter
    Running setup.py install for pyrsistent ... done
Successfully installed MarkupSafe-1.1.1 Send2Trash-1.5.0 attrs-19.3.0 backcall-0.1.0 bleach-3.1.0 colorama-0.4.3 decorator-4.4.1 defusedxml-0.6.0 entrypoints-0.3 importlib-metadata-1.4.0 ipykernel-5.1.3 ipython-7.11.1 ipython-genutils-0.2.0 ipywidgets-7.5.1 jedi-0.15.2 jinja2-2.10.3 jsonschema-3.2.0 jupyter-1.0.0 jupyter-client-5.3.4 jupyter-console-6.1.0 jupyter-core-4.6.1 mistune-0.8.4 more-itertools-8.1.0 nbconvert-5.6.1 nbformat-5.0.4 notebook-6.0.3 pandocfilters-1.4.2 parso-0.5.2 pickleshare-0.7.5 prometheus-client-0.7.1 prompt-toolkit-3.0.2 pygments-2.5.2 pyrsistent-0.15.7 python-dateutil-2.8.1 pywin32-227 pywinpty-0.5.7 pyzmq-18.1.1 qtconsole-4.6.0 six-1.14.0 terminado-0.8.3 testpath-0.4.4 tornado-6.0.3 traitlets-4.3.3 wcwidth-0.1.8 webencodings-0.5.1 widgetsnbextension-3.5.1 zipp-2.0.1
```

실습2: 노트북 실행 1

(nblab) D:₩myVE>python -m notebook
[I 03:00:44.517 NotebookApp] The port 8888 is already in use, trying another port.
[I 03:00:44.518 NotebookApp] The port 8889 is already in use, trying another port.
[I 03:00:44.529 NotebookApp] Serving notebooks from local directory: D:₩myVE
[I 03:00:44.529 NotebookApp] The Jupyter Notebook is running at:
[I 03:00:44.529 NotebookApp]
http://localhost:8890/?token=9e39780498cecdccbdc61fad416d6ab471d6a7bf797ab2ca
[I 03:00:44.529 NotebookApp] or
http://127.0.0.1:8890/?token=9e39780498cecdccbdc61fad416d6ab471d6a7bf797ab2ca
[I 03:00:44.530 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 03:00:44.786 NotebookApp]

To access the notebook, open thi
file:///C:/Users/USER/AppData/ pen.html
Or copy and paste one of these U
http://localhost:8890/?token=9 bf797ab2ca
or http://127.0.0.1:8890/?token=9 7bf797ab2ca



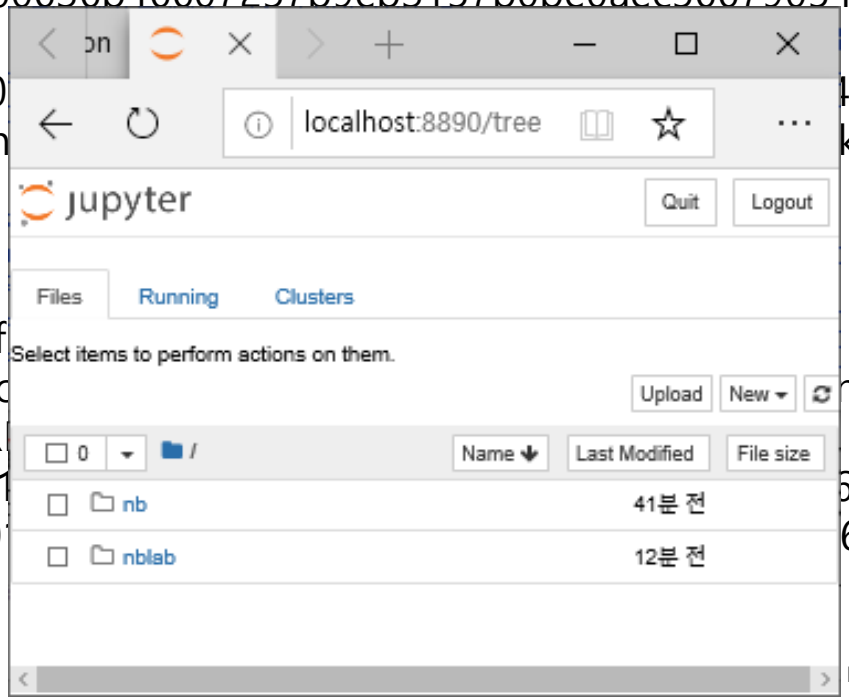

Home Page - Select or create a
hello - Jupyter Notebook
localhost:8888/notebooks/hello.ipynb
Google
쿠팡!
위메프-쇼핑프로...
하프클럽-패션전문...
SOCW
SOCW
Windows 터미널...
how-to-contribute-...
기타 북마크
jupyter
hello Last Checkpoint: 몇 초 전 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3 (ipykernel)
Run
Code
In [1]:
print('Hello')
Hello
In [ ]:

실습2: 노트북 실행 2

(nblab) D:₩myVE>where jupyter
D:₩myVE₩nblab₩Scripts₩jupyter.exe
C:₩Python₩Anaconda3₩Scripts₩jupyter.exe

(nblab) D:₩myVE>jupyter notebook
[I 03:03:37.866 NotebookApp] The port 8888 is already in use, trying another port.
[I 03:03:37.867 NotebookApp] The port 8889 is already in use, trying another port.
[I 03:03:37.879 NotebookApp] Serving notebooks from local directory: D:₩myVE
[I 03:03:37.879 NotebookApp] The Jupyter Notebook is running at:
[I 03:03:37.880 NotebookApp]
http://localhost:8890/?token=b010caa00636b46607257b9eb3157b0be0aec56679054009
[I 03:03:37.881 NotebookApp] or
http://127.0.0.1:8890/?token=b010caa0 4009
[I 03:03:37.881 NotebookApp] Use Con kernels (twice to skip confirmation).
[C 03:03:38.147 NotebookApp]

To access the notebook, open this f
file:///C:/Users/USER/AppData/Ro n.html
Or copy and paste one of these URL
http://localhost:8890/?token=b01 679054009
or http://127.0.0.1:8890/?token=b0 679054009



모듈 notebook, jupyter 설치 비교

(nb) D:₩myVE>pip freeze
attrs==19.3.0
backcall==0.1.0
bleach==3.1.0
colorama==0.4.3
decorator==4.4.1
defusedxml==0.6.0
entrypoints==0.3
importlib-metadata==1.4.0
ipykernel==5.1.3
ipython==7.11.1
ipython-genutils==0.2.0
jedi==0.15.2
Jinja2==2.10.3
jsonschema==3.2.0
jupyter-client==5.3.4
jupyter-core==4.6.1
MarkupSafe==1.1.1
mistune==0.8.4
more-itertools==8.1.0
nbconvert==5.6.1
nbformat==5.0.4
notebook==6.0.3
pandocfilters==1.4.2
parso==0.5.2
pickleshare==0.7.5
prometheus-client==0.7.1
prompt-toolkit==3.0.2
Pygments==2.5.2
pyrsistent==0.15.7
python-dateutil==2.8.1
pywin32==227
pywinpty==0.5.7
pyzmq==18.1.1
Send2Trash==1.5.0
six==1.14.0
terminado==0.8.3
testpath==0.4.4
tornado==6.0.3
traitlets==4.3.3
wcwidth==0.1.8
webencodings==0.5.1
zipp==2.0.1

(nblab) D:₩myVE>pip freeze
attrs==19.3.0
backcall==0.1.0
bleach==3.1.0
colorama==0.4.3
decorator==4.4.1
defusedxml==0.6.0
entrypoints==0.3
importlib-metadata==1.4.0
ipykernel==5.1.3
ipython==7.11.1
ipython-genutils==0.2.0
ipywidgets==7.5.1
jedi==0.15.2
Jinja2==2.10.3
jsonschema==3.2.0
jupyter==1.0.0
jupyter-client==5.3.4
jupyter-console==6.1.0
jupyter-core==4.6.1
MarkupSafe==1.1.1
mistune==0.8.4
more-itertools==8.1.0
nbconvert==5.6.1
nbformat==5.0.4
notebook==6.0.3
pandocfilters==1.4.2
parso==0.5.2
pickleshare==0.7.5
prometheus-client==0.7.1
prompt-toolkit==3.0.2
Pygments==2.5.2
pyrsistent==0.15.7
python-dateutil==2.8.1
pywin32==227
pywinpty==0.5.7
pyzmq==18.1.1
qtconsole==4.6.0
Send2Trash==1.5.0
six==1.14.0
terminado==0.8.3
testpath==0.4.4
tornado==6.0.3
traitlets==4.3.3
wcwidth==0.1.8
webencodings==0.5.1
widgetsnbextension==3.5.1
zipp==2.0.1

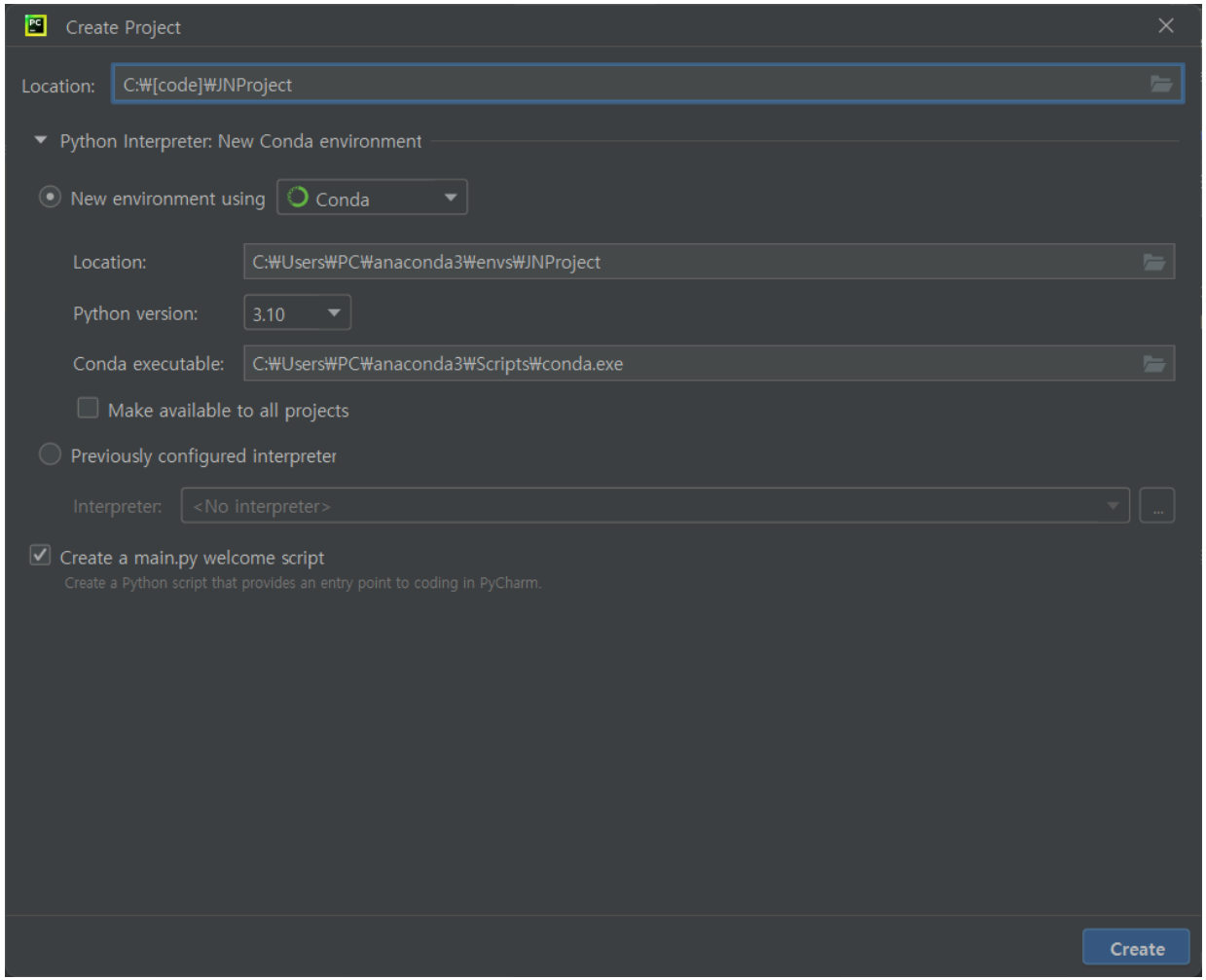
AI Experts
Who Lead
The Future

02

파이참에서 conda 가상환경에 주피터 노트북 설치 및 실행

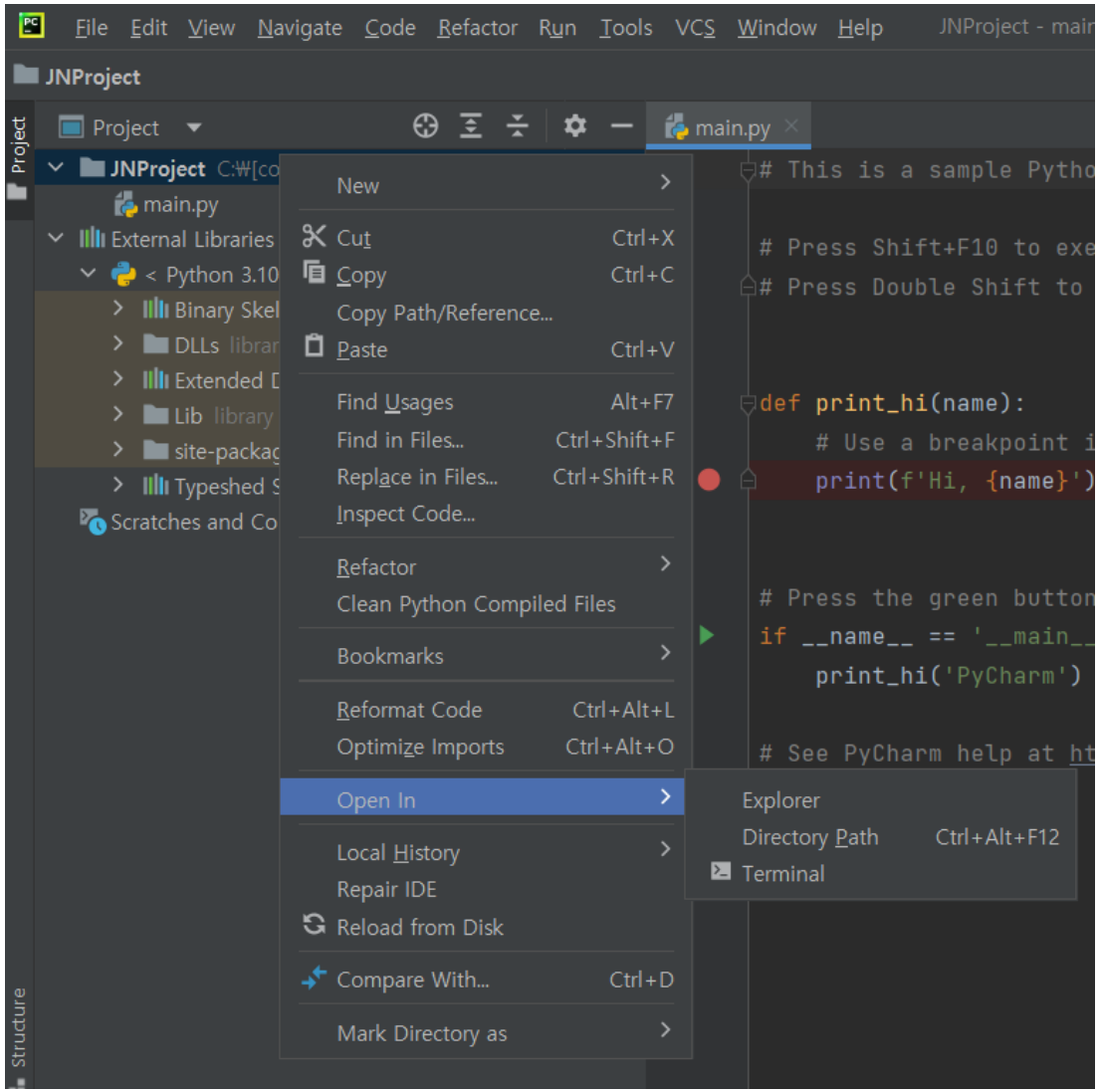
프로젝트 생성

- **프로젝트 이름 JNProject**
  - 새로운 Conda로 생성



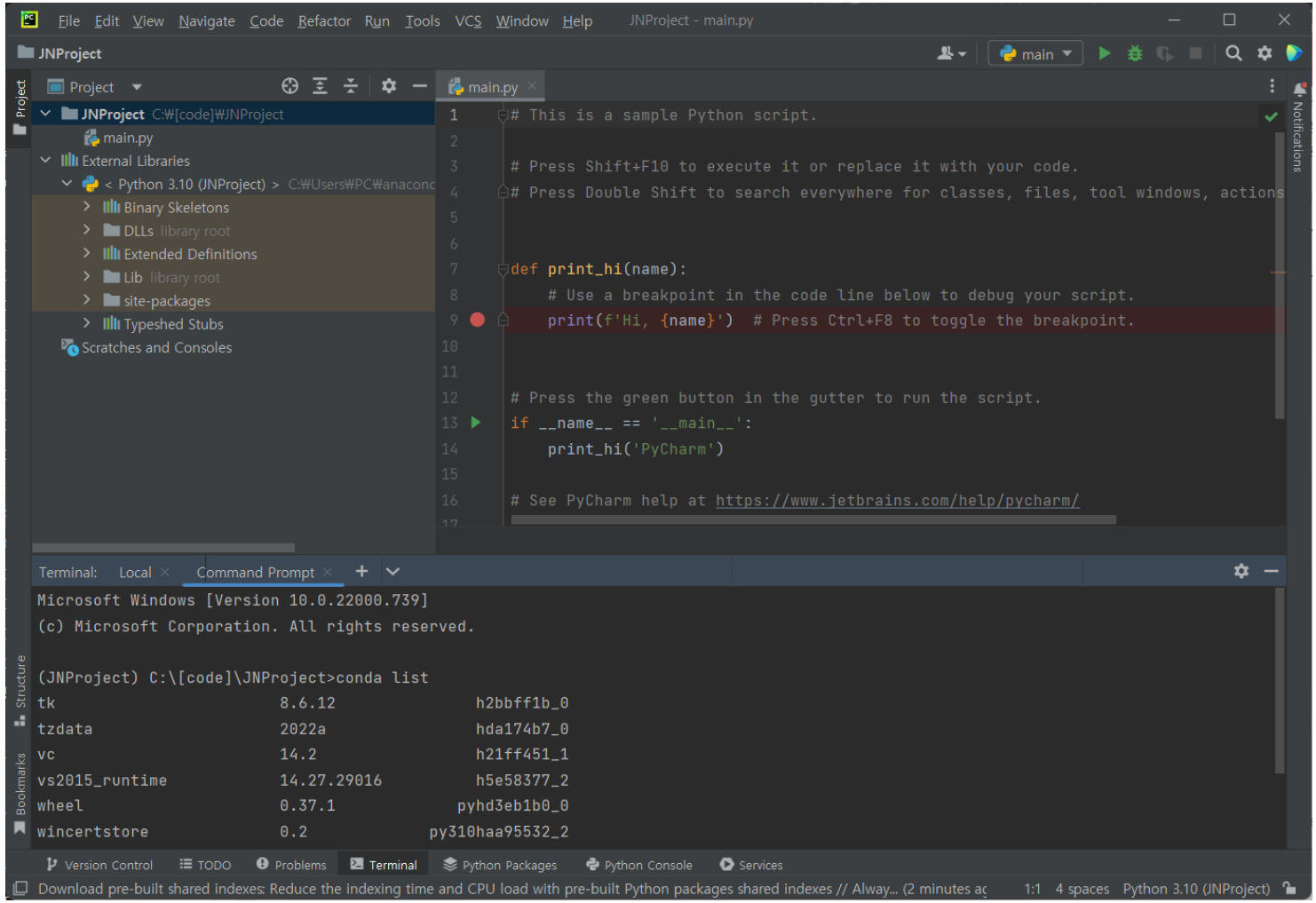
프로젝트 완성

- 터미널 실행



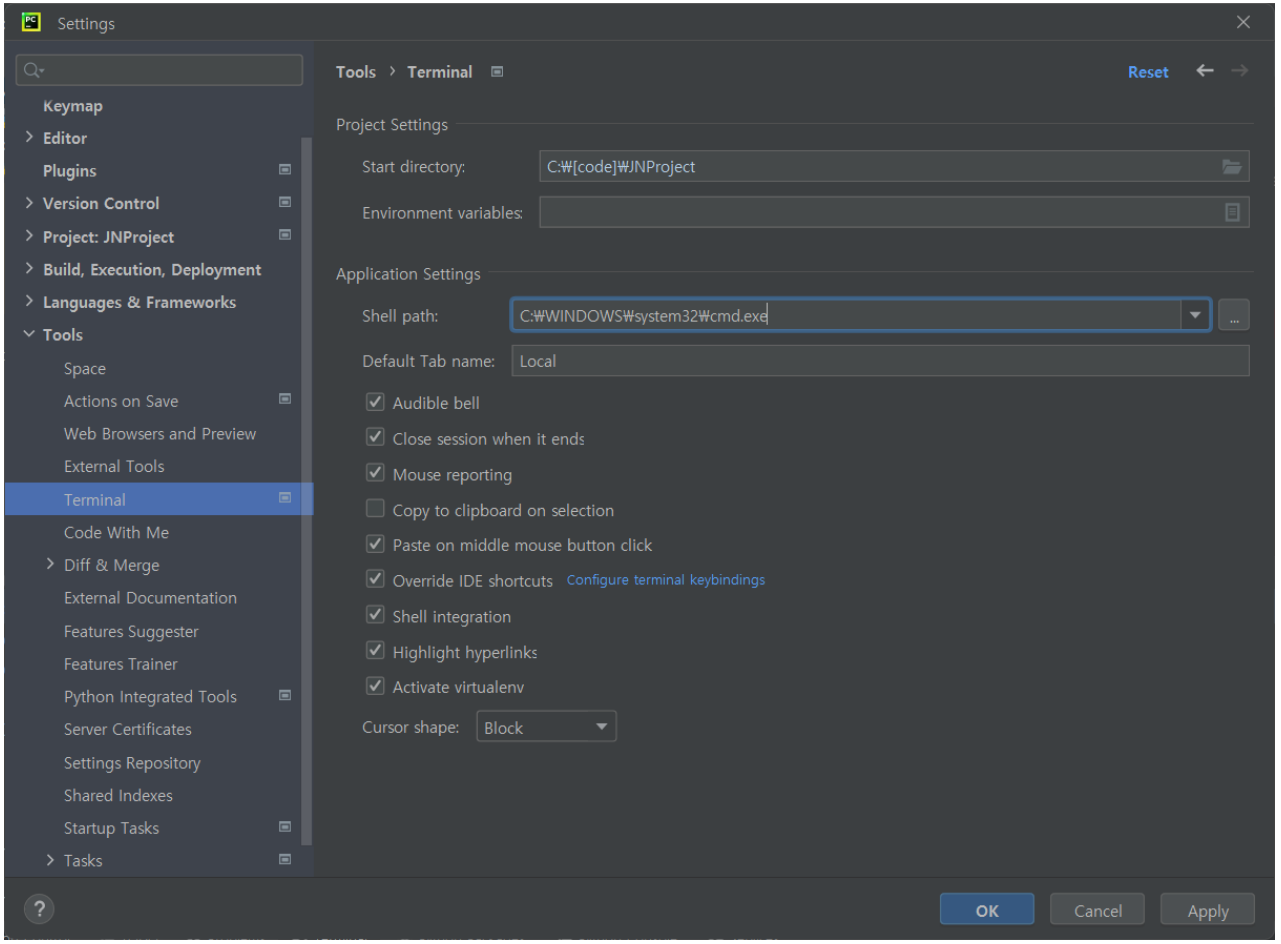
현재 설치 모듈 확인

- **파워쉘에서 명령 프롬프트로 수정**
  - 명령어 conda list



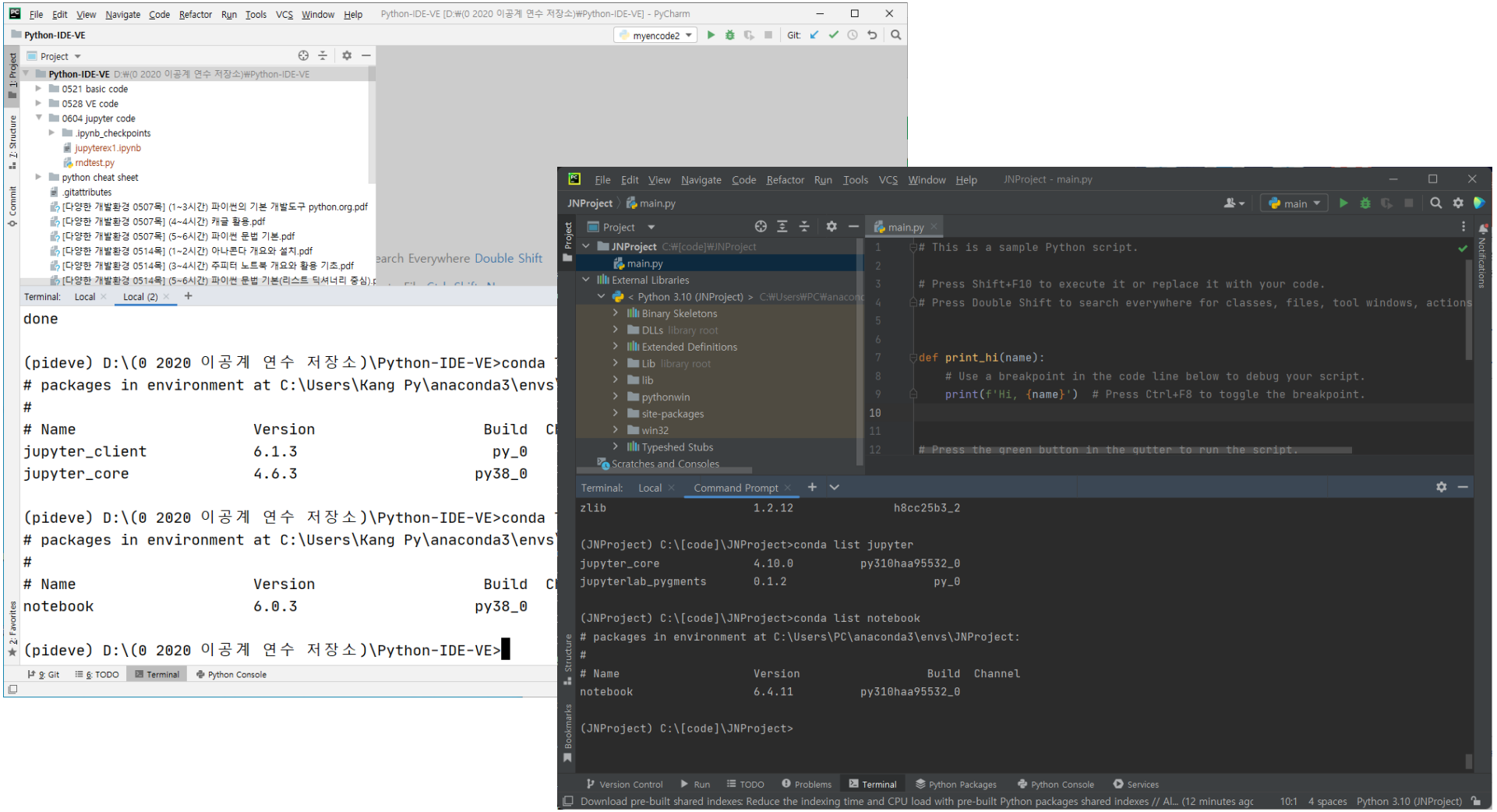
터미널 기본 실행 수정 방법

- 설정(ctrl + alt + s) 실행에서
  - Tools | Terminal



notebook 설치

- **Conda install notebook**
  - 가능한 conda로 설치, pip 설치 시 특히 notebook이 문제가 발생할 수 있음

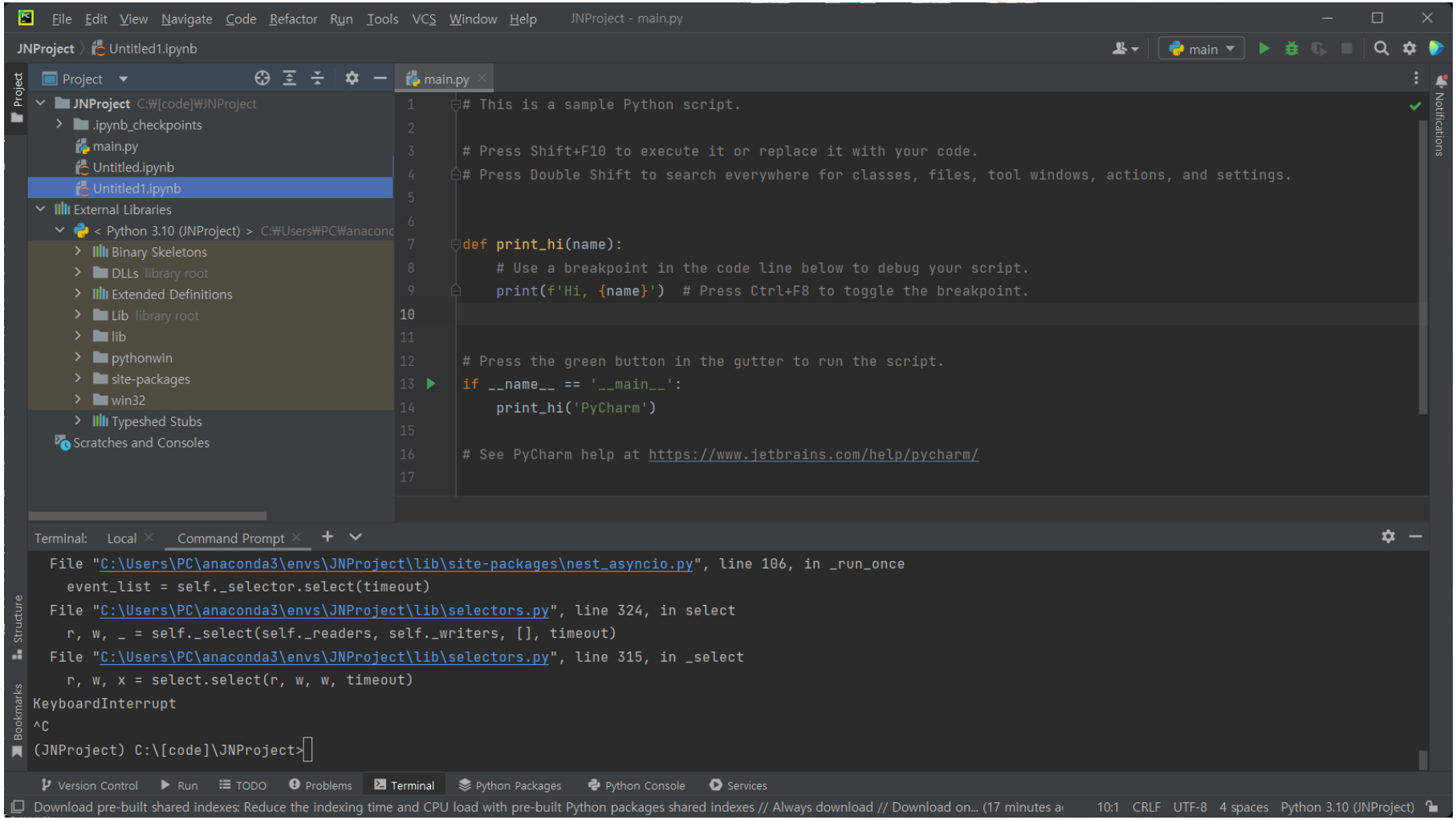


주피터 노트북 설치와 실행

- 설치 2가지
  - conda install jupyter
  - conda install notebook
- 실행 2가지
  - jupyter notebook
  - python –m notebook

```
(pideve) D:\(0 2020 이공계 연수 저장소)\Python-IDE-VE>python -m notebook
[I 15:33:32.161 NotebookApp] The port 8888 is already in use, trying another port.
[I 15:33:32.162 NotebookApp] The port 8889 is already in use, trying another port.
[I 15:33:32.172 NotebookApp] Serving notebooks from local directory: D:\(0 2020 이공계 연수 저장소
)\Python-IDE-VE
[I 15:33:32.172 NotebookApp] The Jupyter Notebook is running at:
[I 15:33:32.173 NotebookApp] http://localhost:8890/?token=f17bc3176370ba801958df09b1decf2322886b8b
07700edc
[I 15:33:32.173 NotebookApp]  or http://127.0.0.1:8890/?token=f17bc3176370ba801958df09b1decf232288
6b8b07700edc
[I 15:33:32.174 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to
 skip confirmation).
[C 15:33:32.238 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/Kang%20Py/AppData/Roaming/jupyter/runtime/nbserver-60092-open.html
```

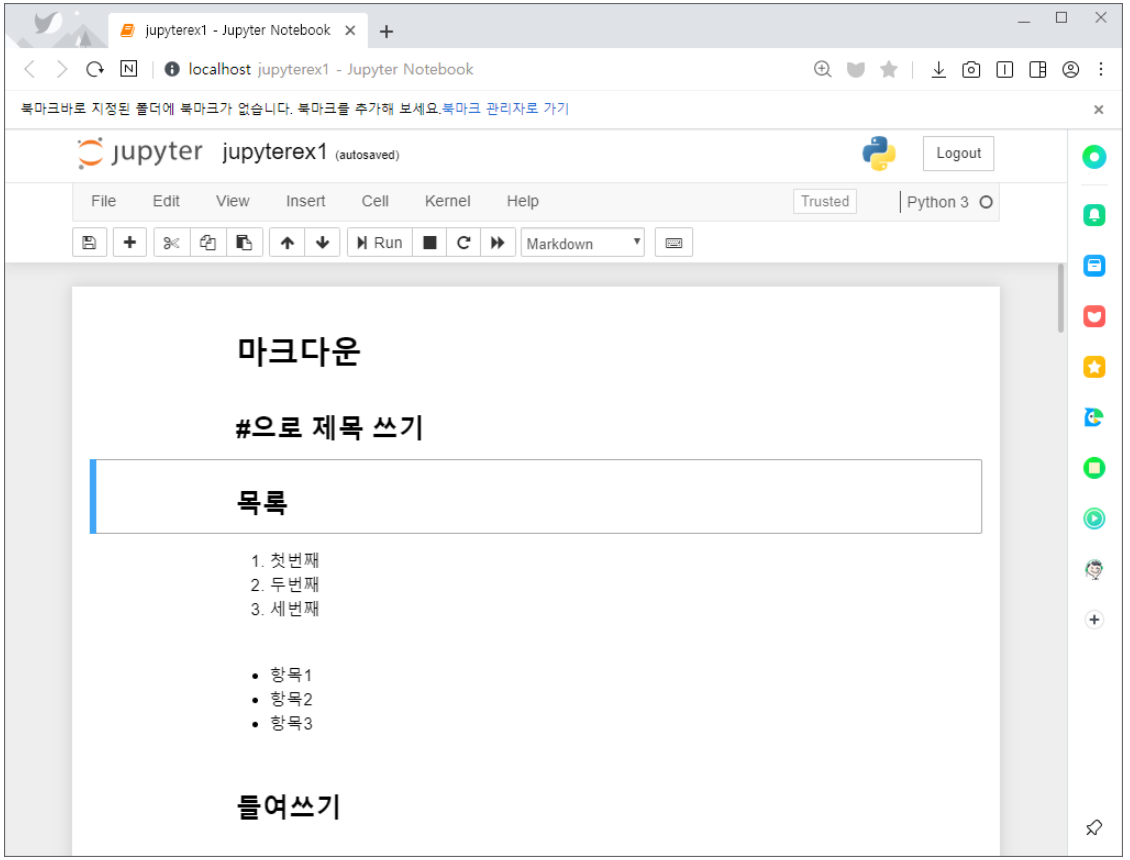
주피터 노트북 파일 생성 후 화면




주피터 노트북
마크다운 편집

실습 파일

- 주피터 노트북 실행
  - 터미널 열기
    - python –m notebook
    - jupyter notebook
- 파일 열기
  - jupyterex1.ipynb



마크다운 개요

- **마크다운은 텍스트 기반의 마크업 언어로 html으로 변환이 가능**
  - 마크다운에서도 6가지의 제목을 나타내는 문법이 있는데 #을 사용
  - 아래처럼 각각 개수만큼 각 숫자에 맞는 h 태그 <h1> 태그

```
# 첫 번째 큰 제목
## 두 번째 큰 제목
### 세 번째 큰 제목
#### 네 번째 제목
##### 다섯 번째 제목
###### 여섯 번째 제목
#######
```

제목

첫 번째 큰 제목

두 번째 큰 제목

세 번째 큰 제목

네 번째 제목

다섯 번째 제목

여섯 번째 제목

#######

실행 결과

목록

- 숫자와 함께 쓰면 순서가 있는 목록
- 글머리 기호와 함께 쓰면 순서가 없는 목록

```
1. 첫번째
2. 두번째
3. 세번째
```

```
+ 첫번째
+ 두번째
+ 세번째

- 첫번째
- 두번째
- 세번째

* 첫번째
* 두번째
* 세번째
```

순서가 있는 목록, 순서가 없는 목록

1. 첫번째
2. 두번째
3. 세번째

- 첫번째
- 두번째
- 세번째
- 첫번째
- 두번째
- 세번째
- 첫번째
- 두번째
- 세번째

순서가 있는 목록, 순서가 없는 목록 결과

들여 쓰기

- 목록 안의 목록을 쓰고 싶다면 tab을 한 번 해주고 사용

```
+ 첫번째
    + 두번째
        + 세번째

- 첫번째
    - 두번째
        - 세번째

* 첫번째
    * 두번째
        * 세번째
```

tab을 각각 해주면 색이 바뀌는 것을 볼 수 있다.

- 첫번째
  - 두번째
    - 세번째
- 첫번째
  - 두번째
    - 세번째
- 첫번째
  - 두번째
    - 세번째

tab에 따라 목록 기호가 바뀌어 있다.

수식 출력

- $와 $ 사이에 기술

$$y = 2n^2$$

```
$y=2n^2$
```

$$c = \sqrt{a^2 + b^2}$$

```
$c=#sqrt{a^2 + b^2}$
```

주피터 노트북
매직 명령어 %, %%

%로 시작하는 명령어

- 보통 %문자로 시작
  - 만약 사용자 변수가 선언되지 않았다면 %문자 없이도 기능을 수행
  - 예를 들어 cls라는 변수가 선언되어 있다면 %cls라고 입력해야 하지만
    - cls라는 변수가 없다면 그냥 cls라고 입력해도 동작을 수행

매직명령어	기능
%magic %lsmagic	모든 매직 명령어의 도움말 출력 매직 명령어 리스트
%automagic	매직함수를 %없이도 실행하게끔 함(default) 또는 %를 붙여야만 실행하게끔 함(실행할 때마다 전환됨)
%pprint	pretty print 모드의 on/off 전환
%exit %quit	물어보지 않고 IPython을 종료시킨다.
%cls	화면 클리어
%who %who_ls %whos	변수의 리스트를 보여준다. 변수 리스트를 파이썬 리스트로 반환한다. 변수명 뿐만 아니라 변수 값도 보여준다.
%reset %reset -f	작업공간을 초기화 시킨다. 수행 여부를 묻지 않고 초기화 시킨다.
%run file.py	file.py 파일을 실행시킨다.
%paste	클립보드의 코드를 실행한다.
%cpaste %edit (or %ed)	코드 블럭을 직접 입력한 후 실행한다. 텍스트에디터를 실행시킨 후 거기에 입력한 코드를 실행한다.
%time %timeit	실행시간을 측정해서 보여준다. 여러 번 실행한 후 실행시간을 분석한다.
%hist	과거 명령어 리스트(history) 출력

주요 명령어

명령어	설명
%pwd, %cd	현재 위치 및 다른 디렉토리로 이동
%history	명령어 히스토리 출력
%reset	모든 정의된 변수 삭제
%%capture	실행되는 명령에 대한 정보의 결과를 저장
%whos	현재 정의된 변수 표시
%pdoc, %psource	Help 기능 실행
%timeit	평균 실행 시간을 출력
%bookmark	디렉토리에 대한 별칭을 저장하고 쉽게 이동할 수 있게 해줌
%%writefile	현재 디렉토리에 파일 생성
%load	디렉토리에 있는 파일을 셀에 로딩
%run	py 프록램 파일을 실행
%matplotlib inline	matplotlib을 내부 셀에서 실행하기

소스 파일 생성 및 실행

- %%매직명령어
  - 셀 전체에 영향, 셀 매직
  - %%writefile 파일이름

- %매직명령어
  - 한 줄의 명령, 라인 매직
  - %run 파일이름

- 다음으로 파이썬 *.py 코드
  - 편집과 실행이 가능

```
In [18]:
1 %%writefile rndtest.py
2 import random as rd
3
4 print(rd.random())
5 print(rd.randint(1, 3))
6 print(rd.choice(range(10)))

Writing rndtest.py
```

```
In [19]:
1 %run rndtest.py

0.9395107979835833
2
3
```

소스파일 셀에 올려 실행 준비

- %load 파일명
- %loadpy 파일명

In [ ]:
```
%load rndtest.py
```

In [ ]:
```
# %load rndtest.py
import random as rd

print(rd.random())
print(rd.randint(1, 3))
print(rd.choice(range(10)))

print(rd.choices(range(30), k=10))
print(rd.choices(range(30), k=10))
print(rd.choices(range(30), k=10))
print(rd.choices(range(30), k=10))
print(rd.choices(range(30), k=10))
```

실행하면 아래와 같이 결과가 표시

In [ ]:
```
%loadpy rndtest.py
```

In [ ]:
```
# %load rndtest.py
import random as rd

print(rd.random())
print(rd.randint(1, 3))
print(rd.choice(range(10)))

print(rd.choices(range(30), k=10))
print(rd.choices(range(30), k=10))
print(rd.choices(range(30), k=10))
print(rd.choices(range(30), k=10))
print(rd.choices(range(30), k=10))
```

실행하면 아래와 같이 결과가 표시

매직 명령어 전체 보기

- %lsmagic

```
In [32]: 1 %lsmagic

Out[32]: Available line magics:
%alias  %alias_magic  %autoawait  %autocall  %automagic  %autosave  %b
ookmark  %cd  %clear  %cls  %colors  %conda  %config  %connect_info  %
copy  %ddir  %debug  %dhist  %dirs  %doctest_mode  %echo  %ed  %edit
%env  %gui  %hist  %history  %killbgscripts  %ldir  %less  %load  %loa
d_ext  %loadpy  %logoff  %logon  %logstart  %logstate  %logstop  %ls
%lsmagic  %macro  %magic  %matplotlib  %mkdir  %more  %notebook  %page
%pastebin  %pdb  %pdef  %pdoc  %pfile  %pinfo  %pinfo2  %pip  %popd  %
pprint  %precision  %prun  %psearch  %psource  %pushd  %pwd  %pycat  %
pylab  %qtconsole  %quickref  %recall  %rehashx  %reload_ext  %ren  %r
ep  %rerun  %reset  %reset_selective  %rmdir  %run  %save  %sc  %set_e
nv  %store  %sx  %system  %tb  %time  %timeit  %unalias  %unload_ext
%who  %who_ls  %whos  %xdel  %xmode

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

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

실행 시간

- %timeit

- ms(밀리/초 : milli second) : 10^{-3}
- μs(마이크로/초: micro second) : 10^{-6}
- ns(나노/초 : nano second) : 10^{-9}
- ps(피코/초 : pico second) : 10^{-12}
- fs(펨토/초 : femto second) : 10^{-15}
- as(아토/초 : atto second) : 10^{-18}

```
In [33]:  1 %timeit?

In [41]:  1 import random as rd
          2
          3 %timeit rd.choices(list(range(10000)), k = 100000)

22.1 ms ± 190 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

```
Docstring:
Time execution of a Python statement or expression

Usage, in line mode:
  %timeit [-n<N> -r<R> [-t|-c] -q -p<P> -o] statement
or in cell mode:
  %%timeit [-n<N> -r<R> [-t|-c] -q -p<P> -o] setup_code
  code
  code...

Time execution of a Python statement or expression using the timeit
module.  This function can be used both as a line and cell magic:

- In line mode you can time a single-line statement (though multiple
  ones can be chained with using semicolons).

- In cell mode, the statement in the first line is used as setup code
  (executed but not timed) and the body of the cell is timed.  The cell
  body has access to any variables created in the setup code.
```

변수 관리

- %whos
- %reset

In [41]:
```
import random as rd

%timeit rd.choices(list(range(10000)), k = 100000)
```

```
22.1 ms ± 190 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

In [42]:
```
%whos
```

```
Variable   Type      Data/Info
-----------------------------
rd         module    <module 'random' from 'd:<...>yve₩₩nb₩₩lib₩₩random.py'>
```

In [43]:
```
%reset
%whos
```

```
Once deleted, variables cannot be recovered. Proceed (y/[n])? y
Interactive namespace is empty.
```

In [44]:
```
year = 2020
s = 'python'
n = list('java')
%whos
```

```
Variable   Type    Data/Info
--------------------------
n          list    n=4
s          str     python
year       int     2020
```

이전 내용이 모두 출력

- %history

```
In [45]: 1 %history
print('Hello World!')
import random as rd

print(rd.random())
print(rd.randint(1, 3))
import random as rd

print(rd.random())
print(rd.randint(1, 3))
import random as rd

print(rd.random())
print(rd.randint(1, 3))
print(rd.choice(range(10)))
list?
```

레이텍 수식 활용

- %%latex
  - ₩[시작 ₩] 로 종료

In [26]:
```
%%latex
₩[x^n + y^n = z^n₩]
```

$$x^n + y^n = z^n$$

주피터 노트북 쉘 명령어 !

!로 시작하는 쉘 명령어

- 명령 프롬프트에서의 명령어

shell	magic	설명
!pwd, !cd	%pwd, %cd	현재 위치 및 다른 디렉토리로 이동
!env	%env	컴퓨터 환경정보 보기
!echo	%echo	메시지 출력하기
!cp	%cp	카피하기
!ls	%ls	현재 디렉토리의 리스트
!mkdir	%mkdir	디렉토리 생성
!rmdir	%rmdir	디렉토리 삭제
!mv	%mv	파일 이동
!rm	%rm	파일 삭제

Notepad로 파일 만들기와 실행

- 2 셀을 코딩 후 노트패드를 종료하면 바로 두 셀이 실행됨

```
In [31]: !type rndtest.py

import random as rd
print(rd.random())
print(rd.randint(1, 3))
print(rd.choice(range(10)))

In [*]: !notepad myhello.py

In [*]: %run myhello.py
```

myhello.py - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
a = list('python')
print(a)
```

Ln 2, Col 9 100% Windows (CRLF) UTF-8