

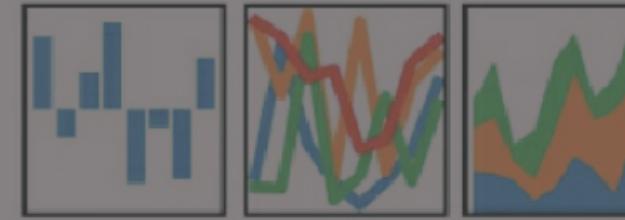


TensorFlow

python

NumPy

pandas  
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



이제 시작해 볼까요.

DATA VISUALIZATION IN PYTHON USING

BOKEH

BeautifulSoup



---

수업 소개부터 하겠습니다.~^^

- 기초만 순서대로 다루지 않는다
- 큰 주제를 두고 이를 완성해가는 과정을 소개
- 원하는 목표를 이루는 과정에서 필요한 기초를 함께 녹여서 진행
- 한 주 한 주 진행하면서  
기초의 양을 줄이고 보다 실전에 가까운 이야기를 진행
- 이야기의 흐름에 집중

---

# 먼저 설치 ~ Anaconda

- 하나가 아닌 **Python** 설치법
- 또 **Python**만 설치해서는 안되고 여러 모듈도 필요할 때마다 설치
  - 그 와중에 모듈간 버전과
  - **Python** 버전도 신경을 써야함
- 그래서 입문자에게는 참으로 좋은 통합 배포판
  - **Anaconda** 아나콘다 (**Python**이 포함되어 있음)

# Download Anaconda Distribution

Version 5.1 | Release Date: February 15, 2018

Download For:   

### High-Performance Distribution

Easily install 1,000+ [data science packages](#)

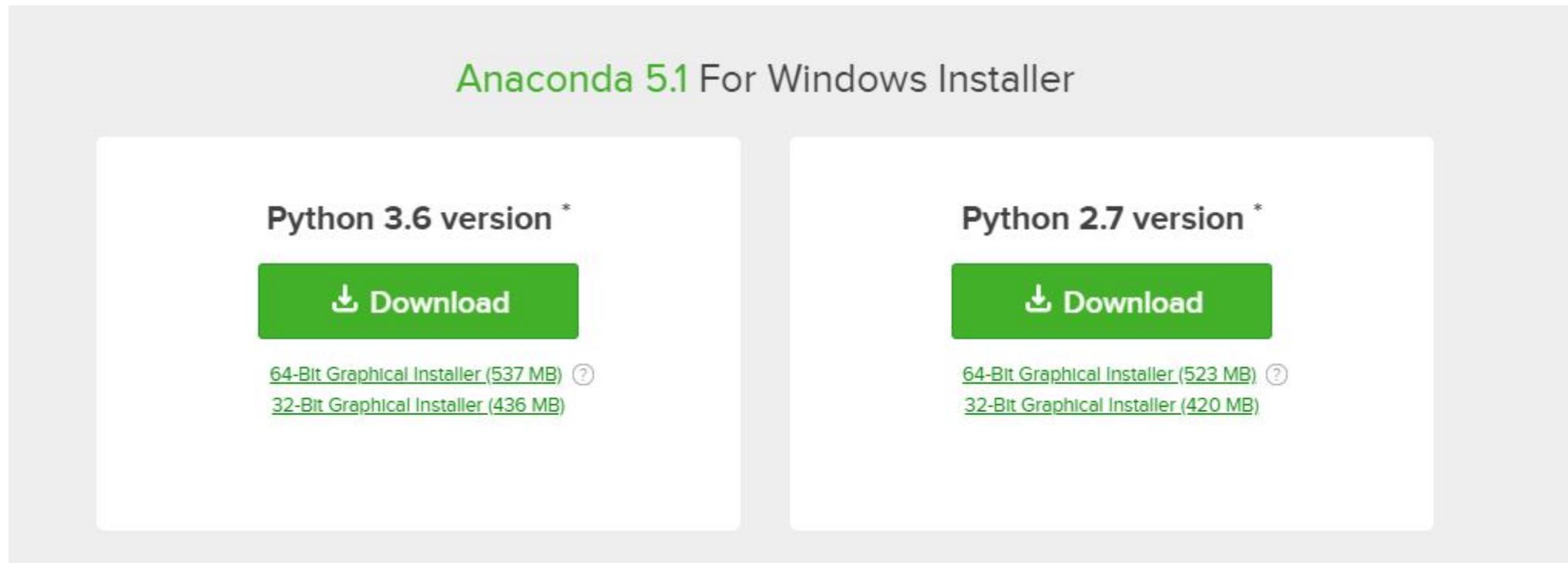
### Package Management

Manage packages, dependencies and environments with [conda](#)

### Portal to Data Science

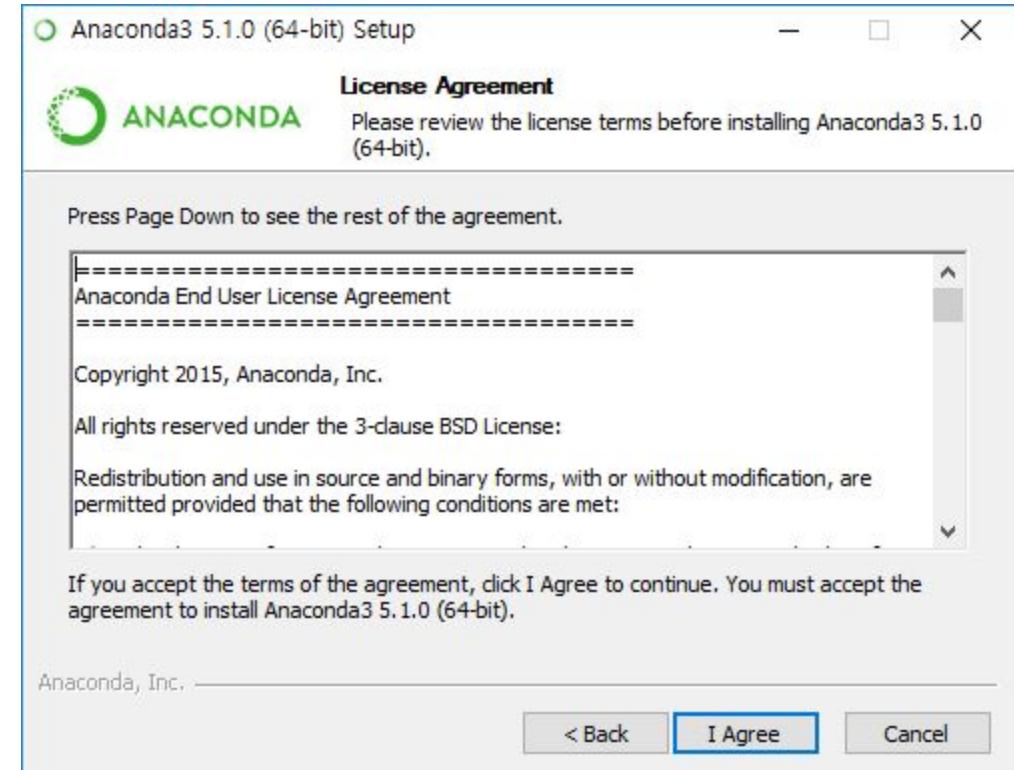
Uncover insights in your data and create interactive visualizations



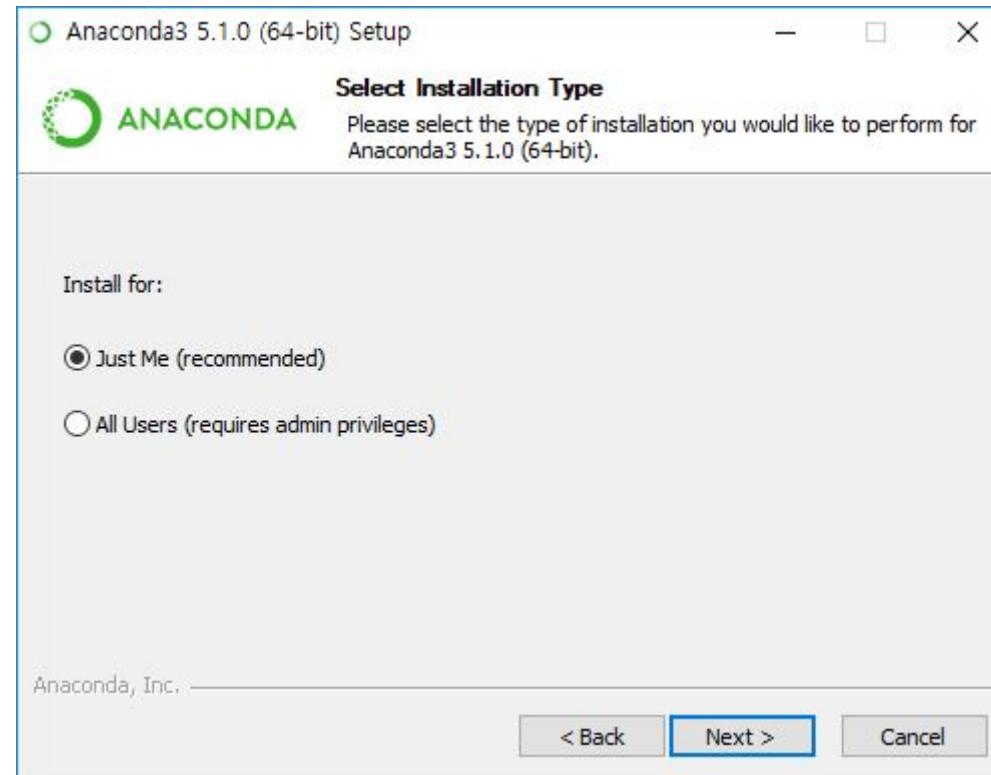


- 꼭~ Python 3.6으로 설치

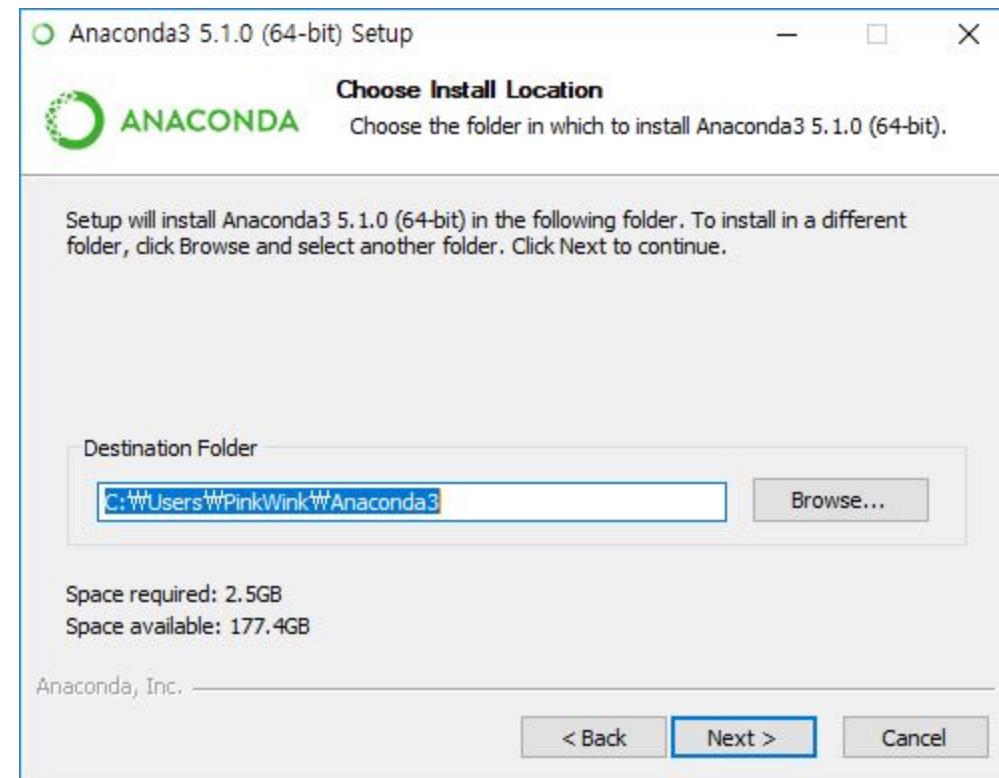
# Anaconda 설치



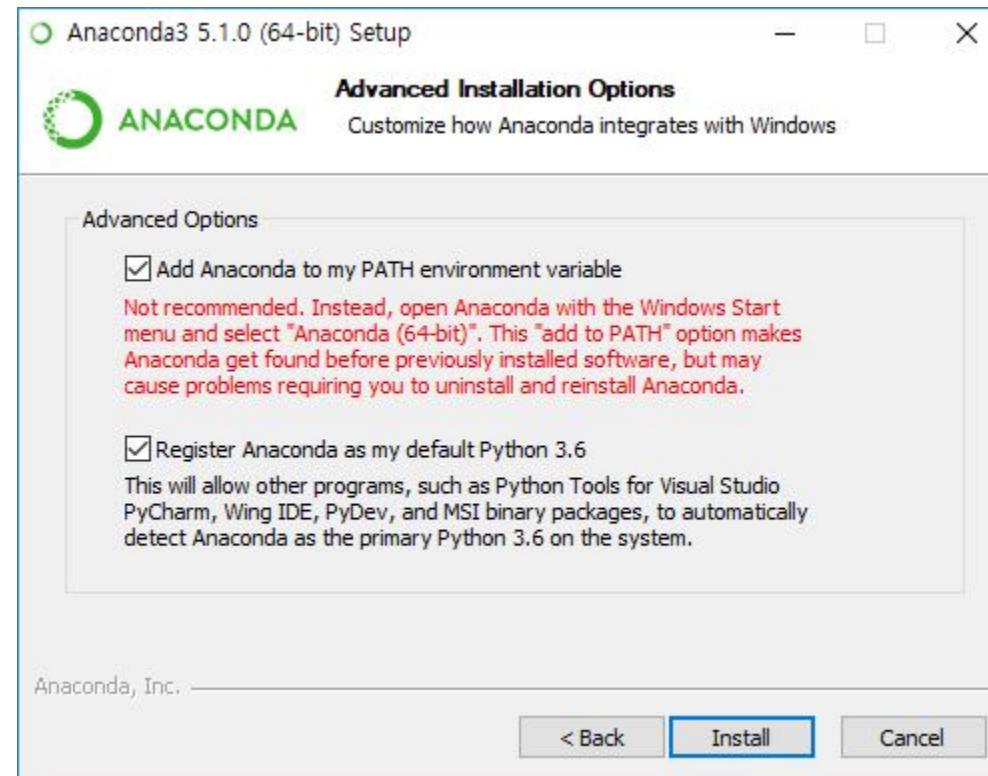
- 뭐 **Next**만 누르면 되는 편한 설치~



- 여기서 **Just Me**로 선택. (**All**로 하면 기본 시작위치가 조금 다른 곳으로 이동함)



- 다른 곳을 선택해도 되지만, 수업 자료와의 통일감을 위해 기본 위치로 하자

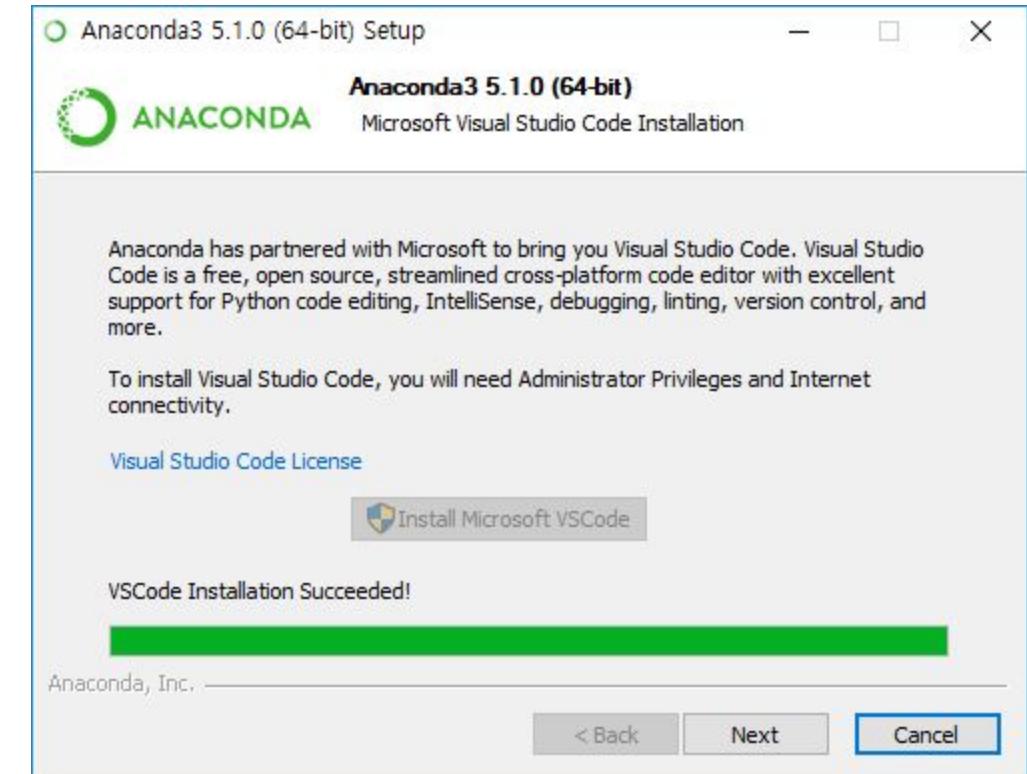


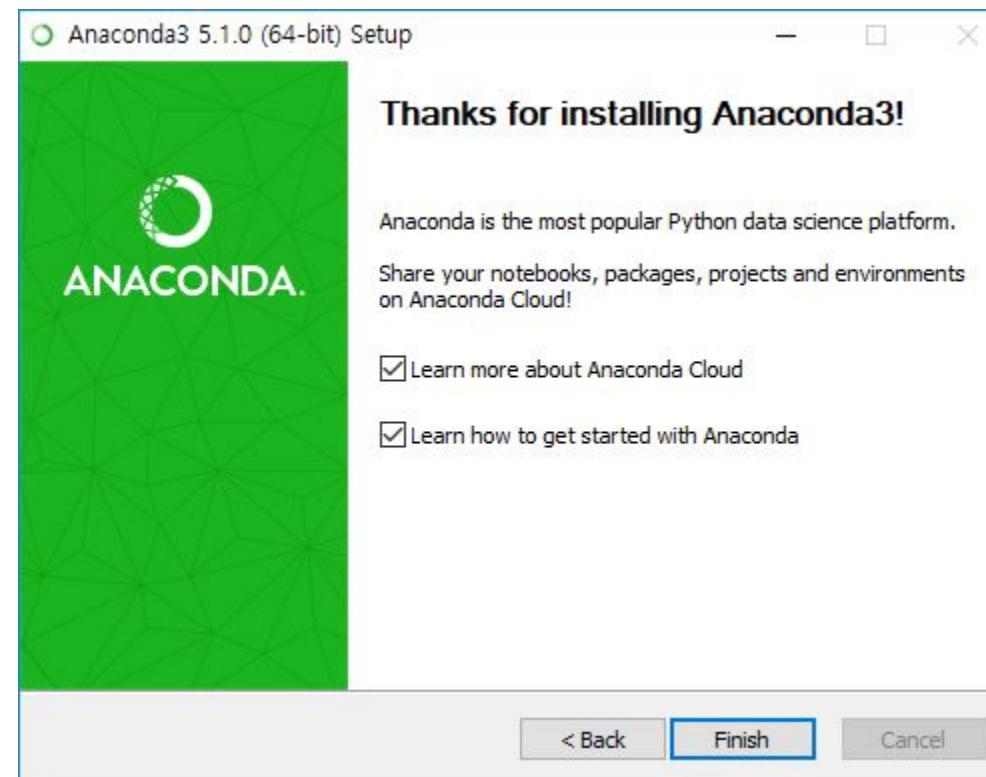
- 경로 설정... 다른 **Python**이 설치되어 있지 않다면 그냥 둘 다 체크하자



- 앗. **VSCode**를 설치하는 화면... 안 깔려 있다면 이 아이도 설치하자

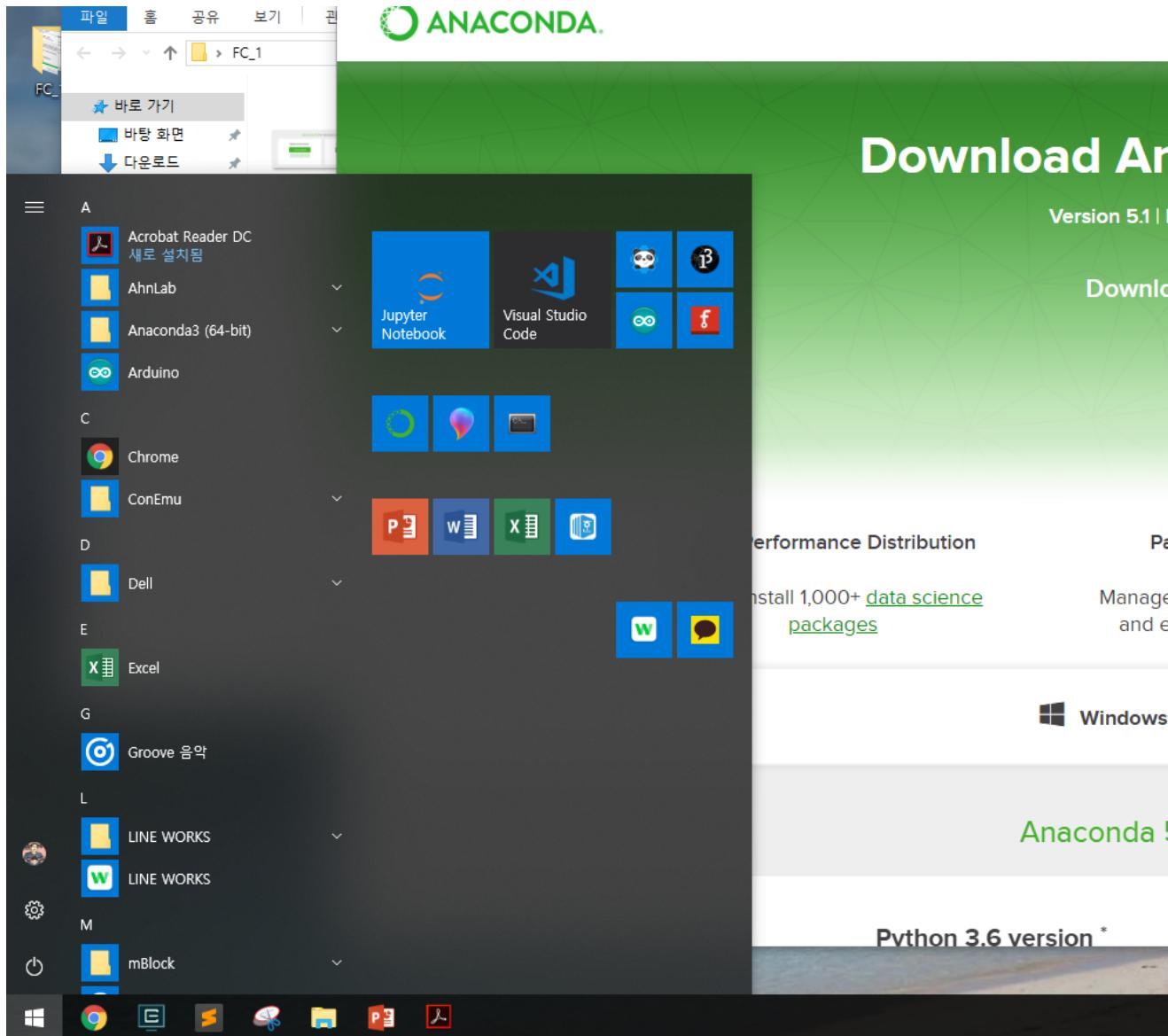
# Anaconda 설치





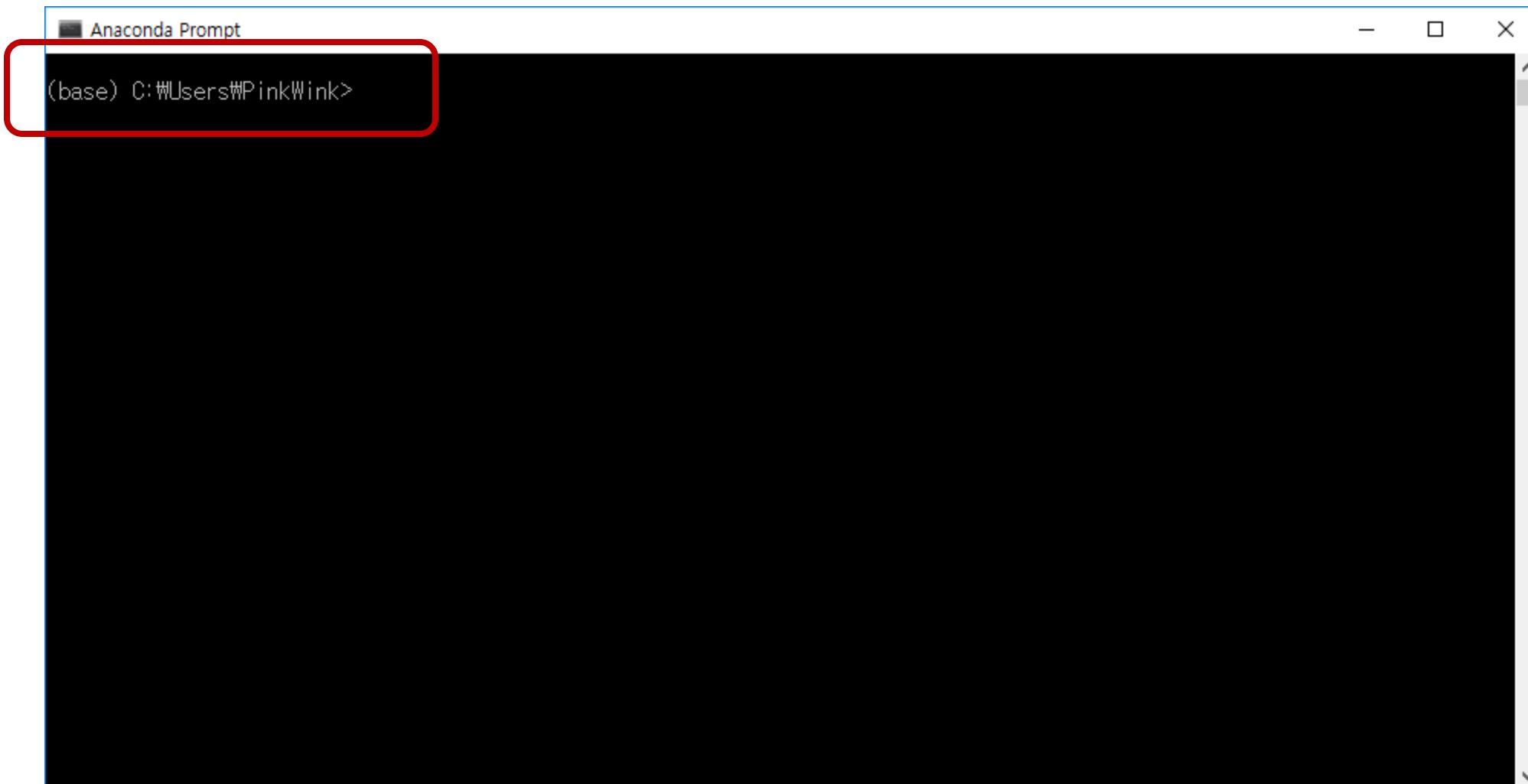
- 설치 끝 ~~~~

# Anaconda 설치

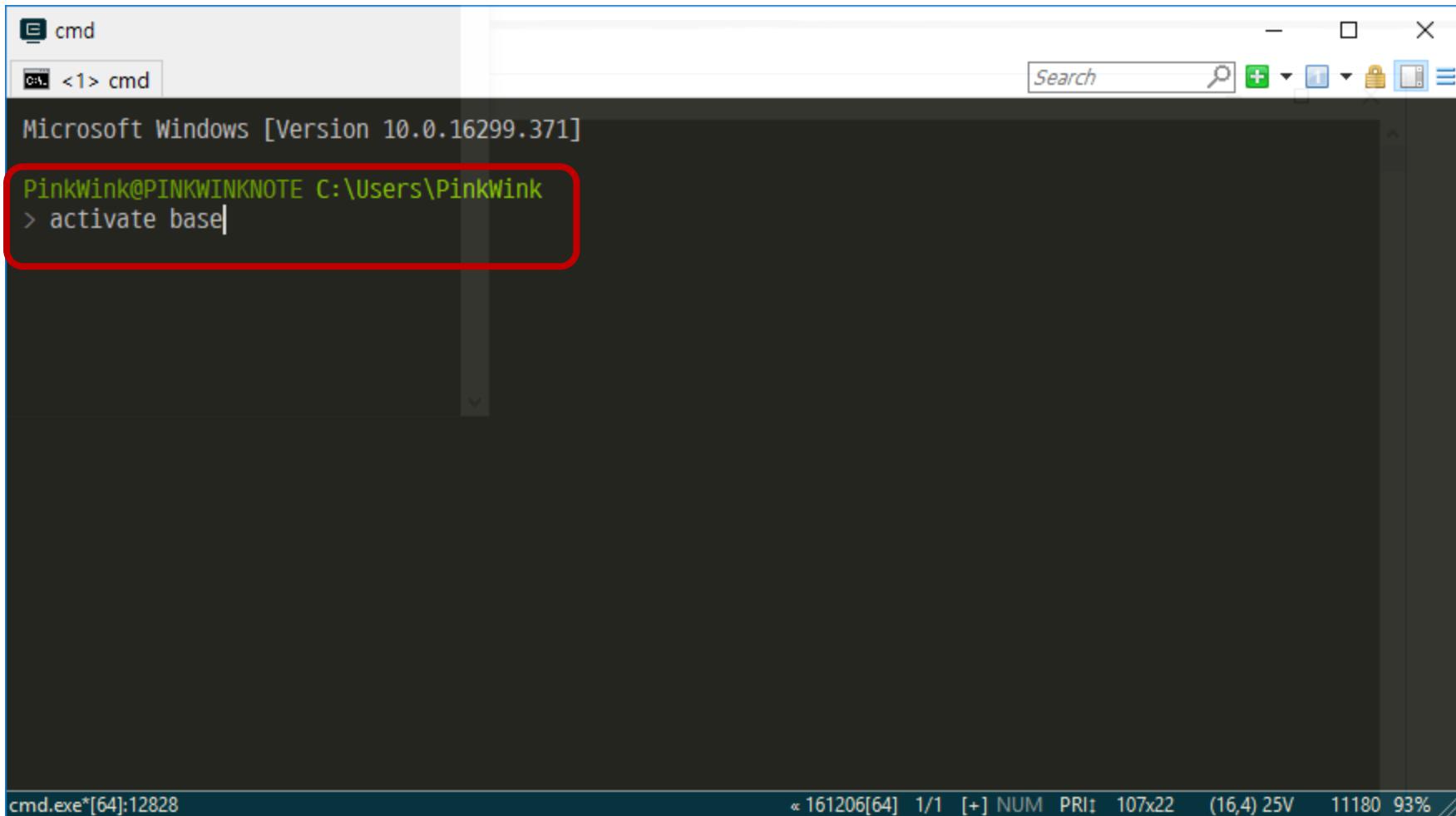


- **Jupyter Notebook**
- **Anacoda Prompt**
- **VSCode**

## Anaconda 설치



- Anaconda Prompt를 실행한 화면



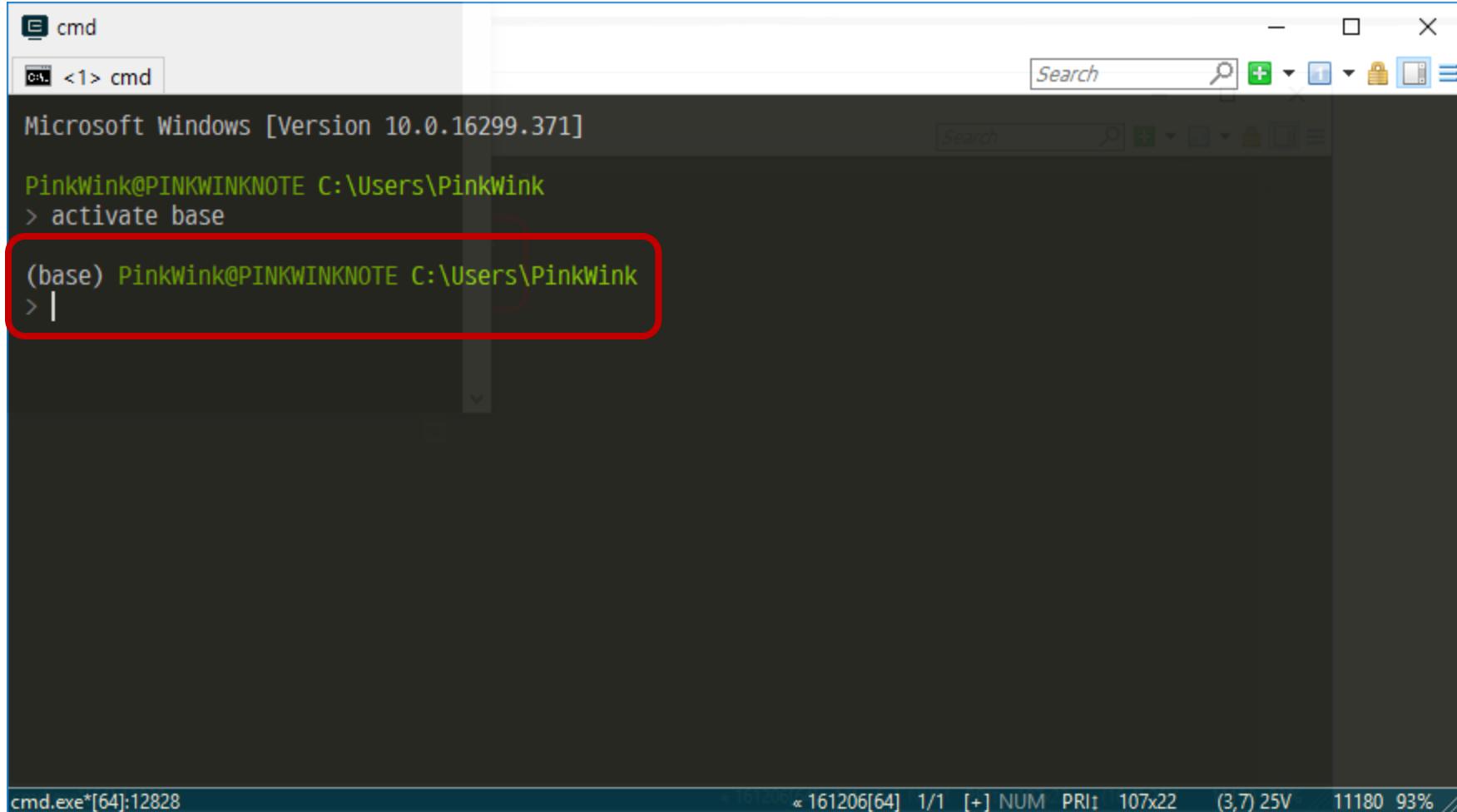
The screenshot shows a Microsoft Windows Command Prompt window titled "cmd". The title bar also displays "cmd <1> cmd". The window content shows the following text:

```
Microsoft Windows [Version 10.0.16299.371]
PinkWink@PINKWINKNOTE C:\Users\PinkWink
> activate base|
```

The line starting with "> activate base|" is highlighted with a red rectangle. The bottom status bar of the terminal window shows the process ID "cmd.exe\* [64]:12828" and various system metrics like CPU usage and battery level.

- 만약 다른 터미널 프로그램에서 접근할려고 하면 **activate base**라고 명령

## Anaconda 설치



The screenshot shows a Microsoft Windows Command Prompt window titled "cmd". The title bar also displays "cmd" and a tab labeled "<1> cmd". The window content shows the following text:

```
Microsoft Windows [Version 10.0.16299.371]
PinkWink@PINKWINKNOTE C:\Users\PinkWink
> activate base
(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
> |
```

A red rectangular box highlights the line "(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink". At the bottom of the window, there is a taskbar with the following information:

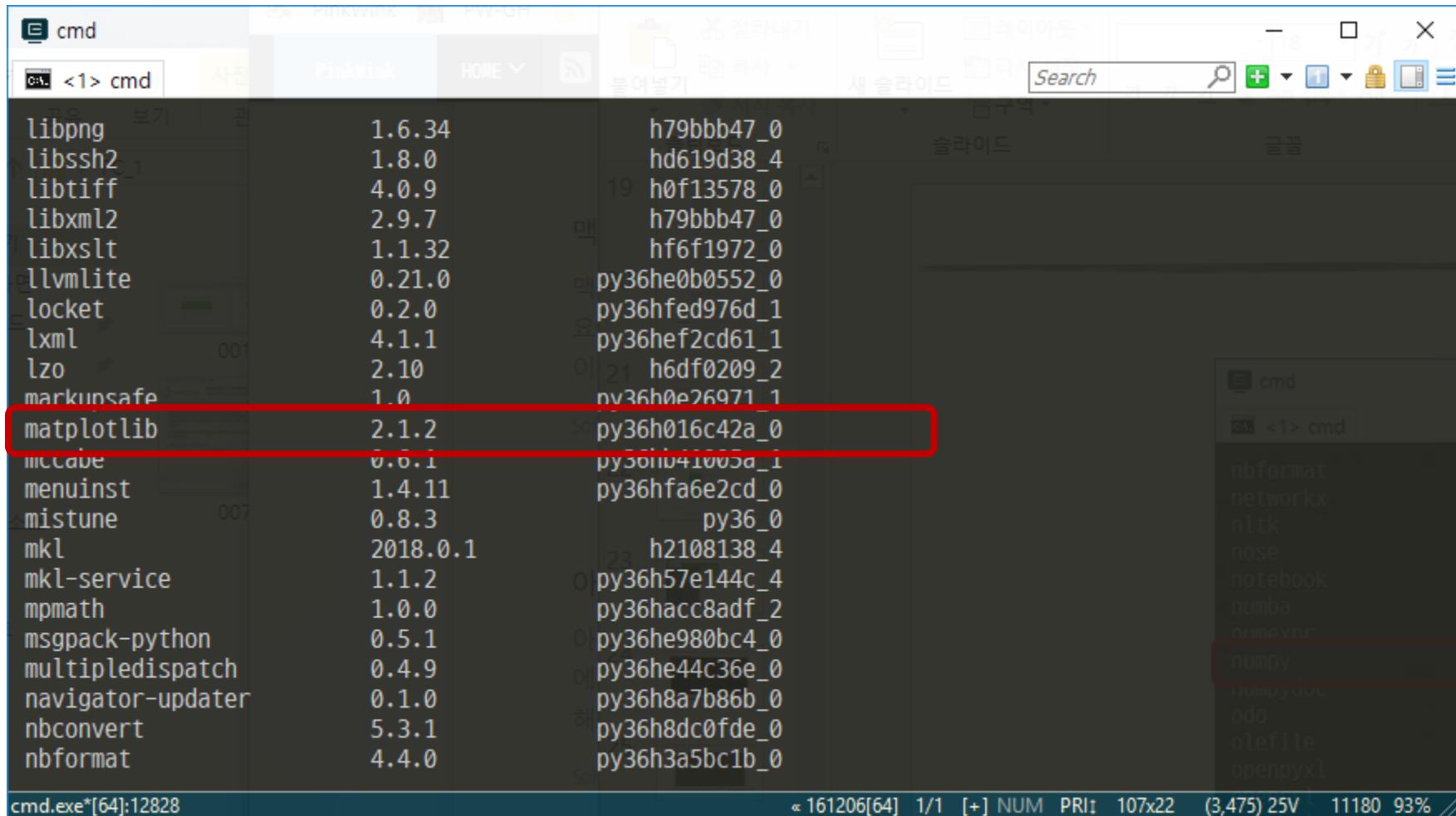
cmd.exe\*[64]:12828      \* 161206[64] 1/1 [+]- NUM PRI: 107x22 (3,7) 25V 11180 93%

- 아무튼 **(base)**라고 이름이 붙은 상태에서 시작하면 정상 (통상 **base**)

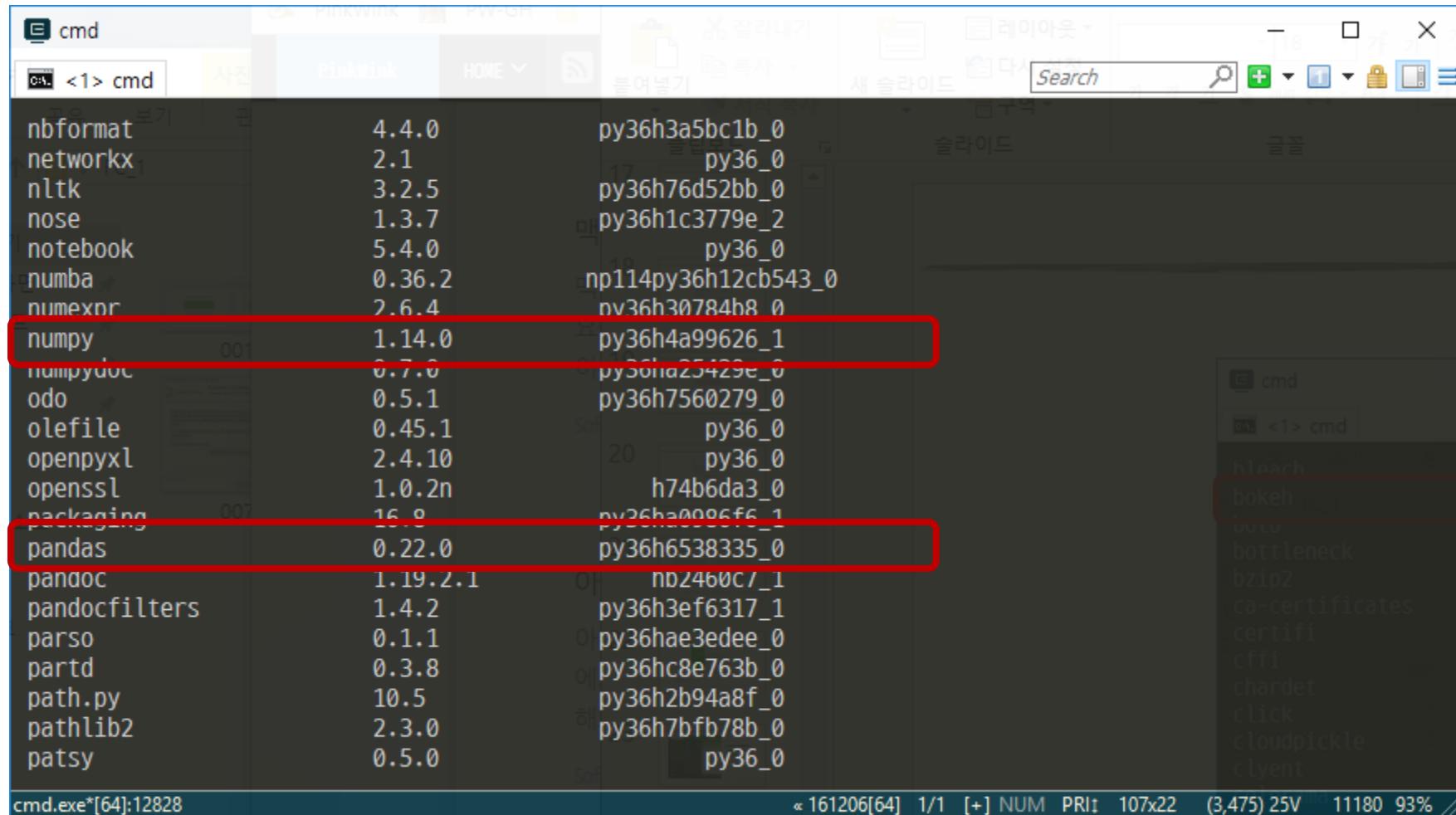
- **conda list**: 현재 설치된 모듈들의 목록을 알려줌

```
> conda list
# packages in environment at C:\Users\PinkWink\Anaconda3:
#
#           Name          Version      Build  Channel
_ipyw_jlab_nb_ext_conf    0.1.0       py36he6757f0_0
alabaster                  0.7.10      py36hcd07829_0
anaconda                   5.1.0        py36_2
anaconda-client              1.6.9        py36_0
anaconda-navigator            1.7.0        py36_0
anaconda-project              0.8.2      py36hfad2e28_0
asn1crypto                  0.24.0       py36_0
astroid                     1.6.1        py36_0
astropy                     2.0.3      py36hfa6e2cd_0
attrs                       17.4.0       py36_0
babel                      2.5.3        py36_0
backports                   1.0       py36h81696a8_1
backports.shutil.get_terminal_size 1.0.0
beautifulsoup4                4.6.0      py36hd4cc5e8_1
bitarray                     0.8.1      py36hffad0e2cd_0
bkcharts                     0.2       py36h7e685f7_0
blaze                      0.11.3      py36h8a29ca5_0
bleach                      2.1.2        py36_0
```

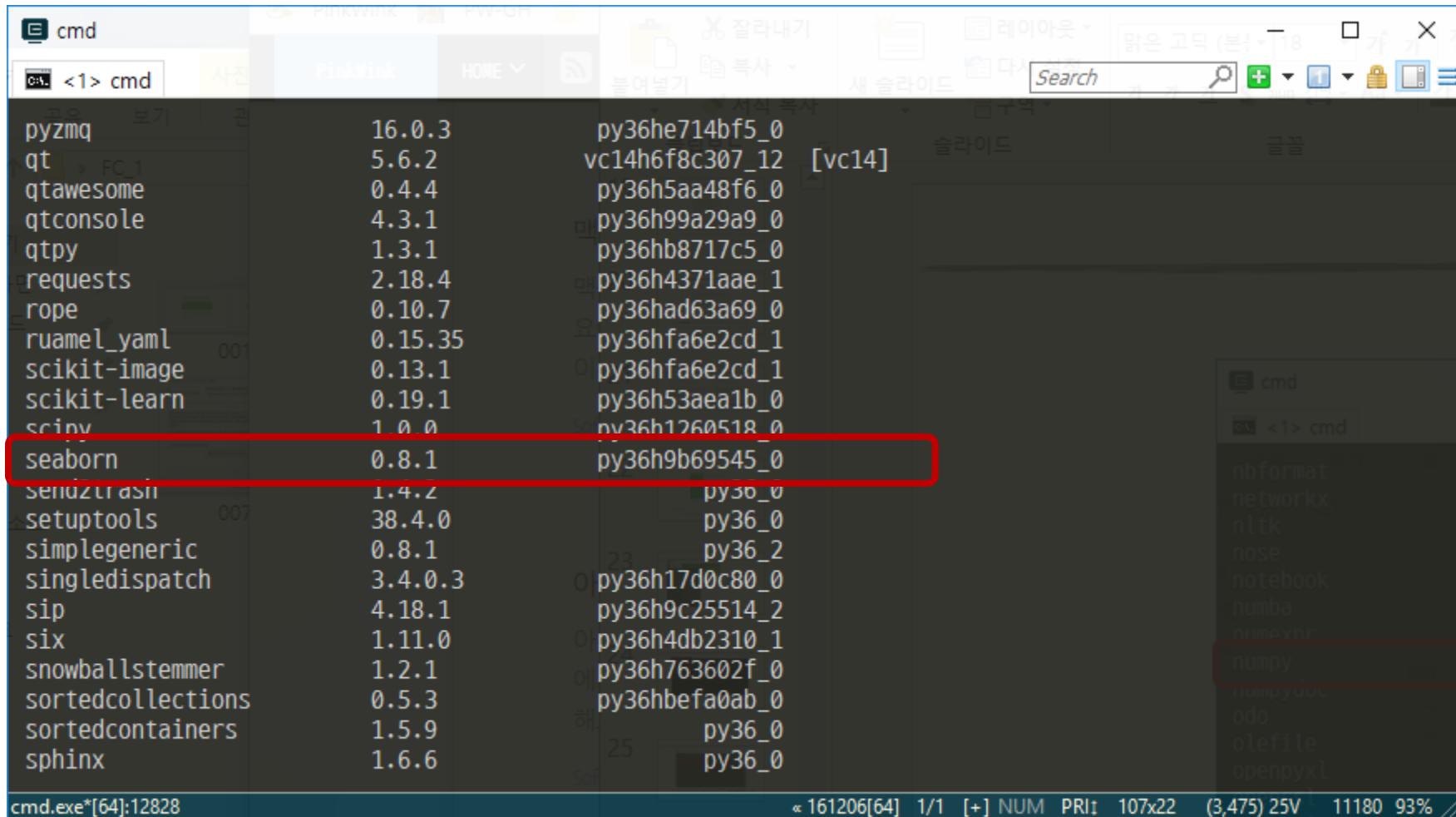
# Anaconda 설치



# Anaconda 설치



# Anaconda 설치

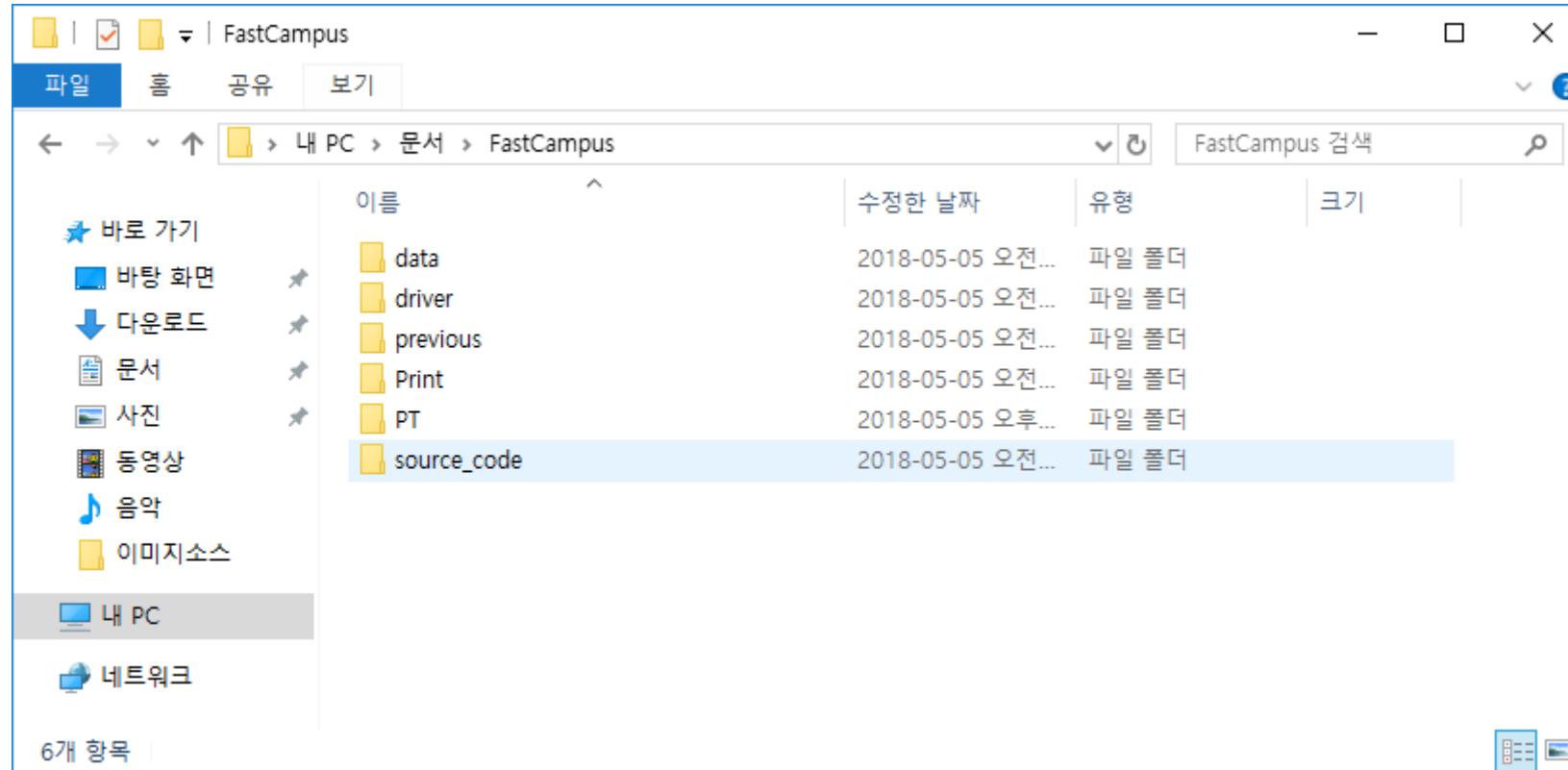


- Python은 모듈들의 버전 변화가 많은 언어
- 특히 인기있는 모듈들은 극심한 편
- 작업 버전업에도 명령어 자체가 변하는 경우도 많음

# Jupyter Notebook 실행

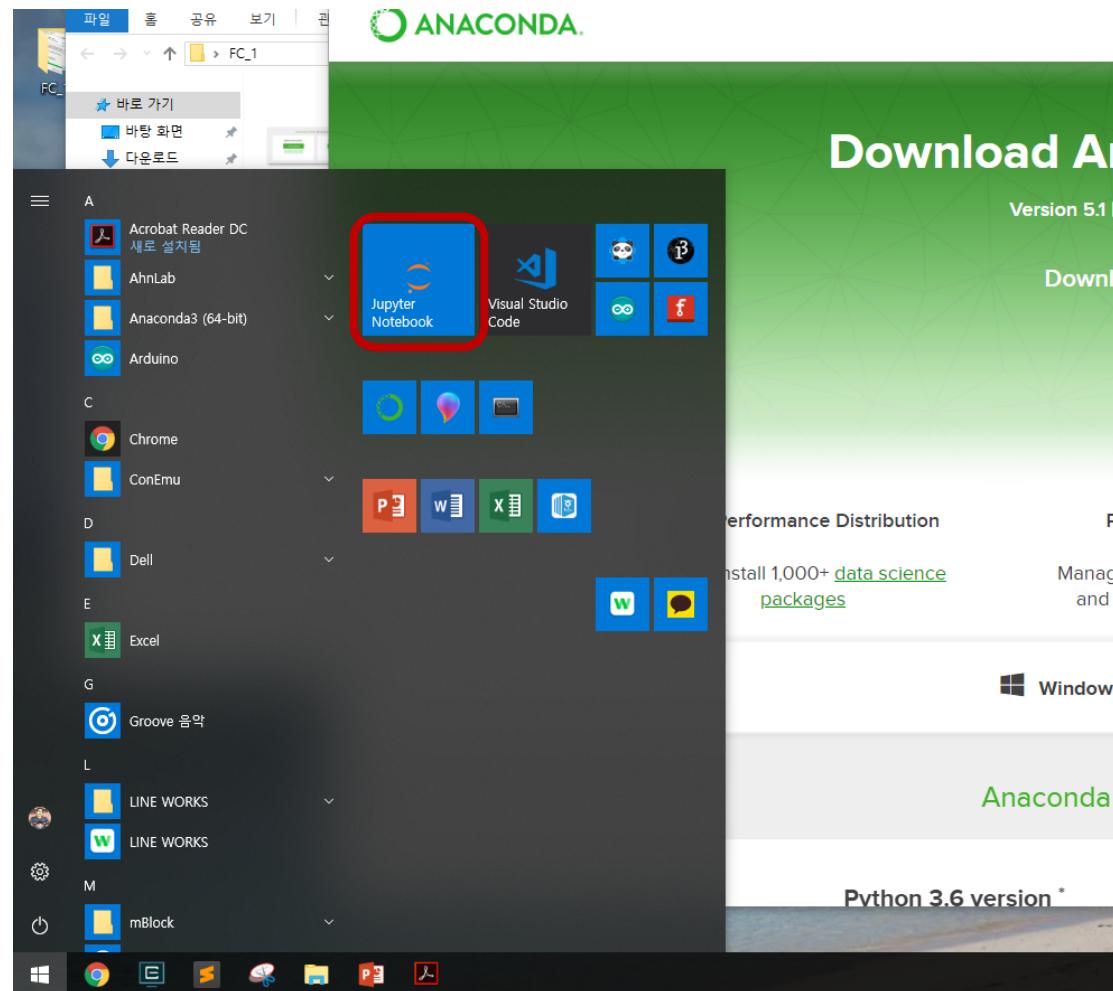
- 입문 수업의 특성상 폴더의 경로 설정 등에 문제가 있을 수 있음
- 시간의 효율성을 위해 폴더 이름과 순서를 맞춰 놓는 작업
- 우리 수업은 앞으로 **Jupyter Notebook**으로 진행

## Jupyter Notebook 실행

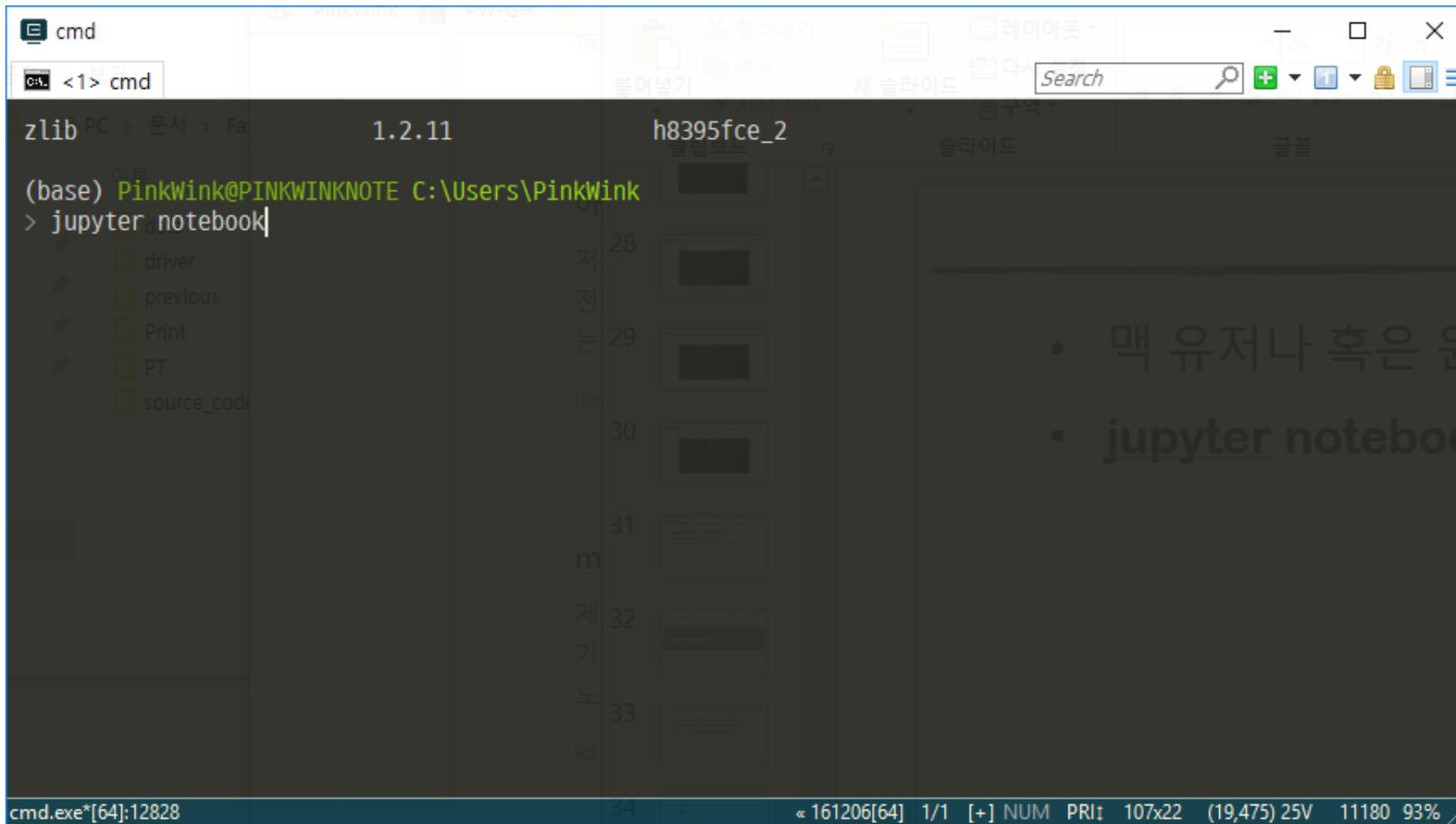


- **data, driver, source\_code** 폴더를 만들어 둠
- 앞으로 수업 자료 및 실습은 **source\_code** 폴더에서 진행하는 것으로 약속

- Windows User의 경우는 시작 버튼을 누르고 Jupyter Notebook 선택



- 맥 유저나 혹은 윈도우 유저의 다른 접근법은 터미널에서
- **jupyter notebook** 이라고 입력



# Jupyter Notebook 실행

The screenshot shows a Windows taskbar with several pinned icons. In the center, there is a window titled "cmd - jupyter notebook". The window contains a command-line interface where the user has run "jupyter notebook". The output shows that JupyterLab beta preview extension was loaded from C:\Users\PinkWink\Anaconda3\lib\site-packages\jupyterlab, and the application directory is C:\Users\PinkWink\Anaconda3\share\jupyter\lab. However, an error message indicates that the server extension jupyterlab failed to load due to an issue with loading nodejs. The error traceback shows the code path from the jupyterlab commands.py file through process.py and jlpmap.py, finally reaching a ValueError. The error message states that nodejs 5+ and npm need to be installed before continuing. The taskbar also shows other pinned icons for Google, Microsoft Edge, File Explorer, and File History.

```
(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
> jupyter notebook
[I 13:49:40.400 NotebookApp] JupyterLab beta preview extension loaded from C:\Users\PinkWink\Anaconda3\lib\site-packages\jupyterlab
[I 13:49:40.400 NotebookApp] JupyterLab application directory is C:\Users\PinkWink\Anaconda3\share\jupyter\lab
[W 13:49:40.416 NotebookApp] Error loading server extension jupyterlab
  Traceback (most recent call last):
    File "C:\Users\PinkWink\Anaconda3\lib\site-packages\jupyterlab\commands.py", line 321, in __init__
      self._run(['node', 'node-version-check.js'], cwd=HERE, quiet=True)
    File "C:\Users\PinkWink\Anaconda3\lib\site-packages\jupyterlab\commands.py", line 1165, in _run
      proc = Process(cmd, **kwargs)
    File "C:\Users\PinkWink\Anaconda3\lib\site-packages\jupyterlab\process.py", line 73, in __init__
      self.proc = self._create_process(cwd=cwd, env=env)
    File "C:\Users\PinkWink\Anaconda3\lib\site-packages\jupyterlab\process.py", line 131, in _create_process
      cmd[0] = which(cmd[0], kwargs.get('env'))
    File "C:\Users\PinkWink\Anaconda3\lib\site-packages\jupyterlab\jlpmap.py", line 59, in which
      raise ValueError(msg)
  ValueError: Please install nodejs 5+ and npm before continuing installation. nodejs may be installed using conda or directly from the nodejs website.
```

- 만약 이런 메시지가 나타난다면

# Jupyter Notebook 실행

The screenshot shows a Jupyter Notebook environment. On the left, a terminal window titled 'cmd - conda install -c conda-forge nodejs' displays the command being run and its progress. On the right, a package details view for 'nodejs' is shown, including its license, home page, download statistics, and installers section.

```
cmd - conda install -c conda-forge nodejs
<1> cmd - conda in...
(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
>
(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
>
(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
> conda install -c conda-forge nodejs
Solving environment: / |
```

Conda

Search

License: MIT  
Home: <https://nodejs.org/>  
105573 total downloads  
Last upload: 10 days and 6 hours ago

Installers

conda install ?

Platform	Version
linux-64	v9.11.1
win-32	v9.11.1
osx-64	v9.11.1
win-64	v9.11.1

To install this package with conda run:

```
conda install -c conda-forge nodejs
```

python.exe[64]:11864

\* 161206[64] 1/1 [+]- NUM PRI 107x22 (24,533) 25V 11180 93%

- **conda install –c conda-forge nodejs**

# Jupyter Notebook 실행

```
cmd - conda install -c conda-forge nodejs
<1> cmd - conda in...
## Package Plan ##
environment location: C:\Users\PinkWink\Anaconda3
added / updated specs:
- nodejs
The following packages will be downloaded:
package          | build
nodejs-9.11.1    | 0      12.6 MB  conda-forge
The following NEW packages will be INSTALLED:
nodejs: 9.11.1-0  conda-forge
Proceed ([y]/n)? y
python.exe*[64]:11864
```

- nodejs 모듈을 설치

# Jupyter Notebook 실행

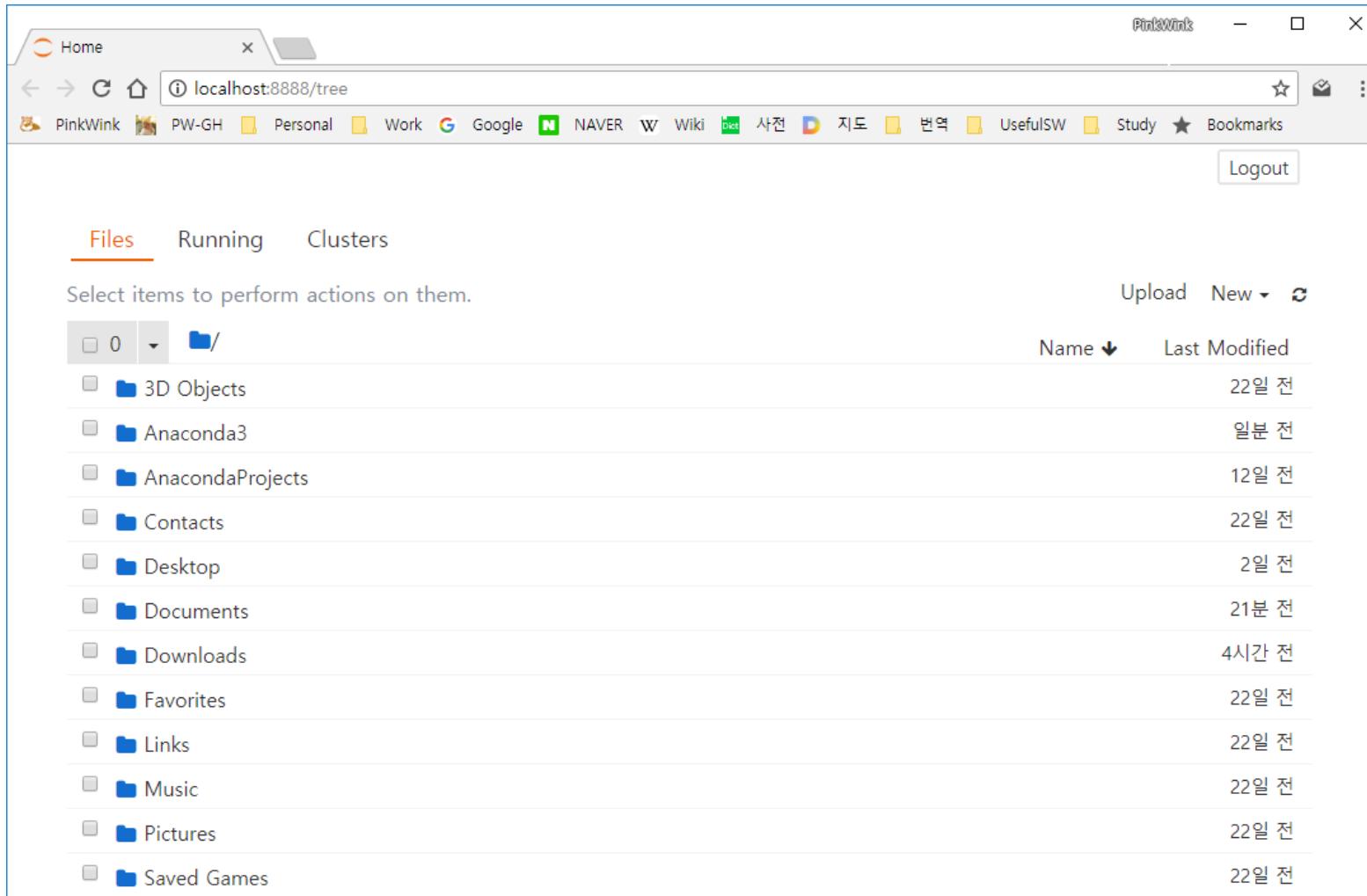
```
cmd
<1> cmd
▶ 내 PC > 문서 > Fa
The following packages will be downloaded:
  package          | build
  --              | --
  nodejs-9.11.1   | 30
  previous        | 31  0      12.6 MB  conda-forge
  Print PT
The following NEW packages will be INSTALLED:
  nodejs: 9.11.1-0 conda-forge
  Proceed ([y]/n)? y
  32
  33
  34
  35
  Downloading and Extracting Packages
  nodejs 9.11.1: #####
  Preparing transaction: done
  Verifying transaction: done
  Executing transaction: done
  (base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
> |
```

- 다시 jupyter notebook 실행

```
cmd - jupyter notebook
cmd <1> cmd - jupyter ...
nodejs-9.11.1
The following NEW packages will be INSTALLED:
  nodejs: 9.11.1-0 conda-forge
Proceed ([y]/n)? y
Downloading and Extracting Packages
nodejs 9.11.1: ###### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
> jupyter notebook
[I 13:54:09.952 NotebookApp] JupyterLab beta preview extension loaded from C:\Users\PinkWink\Anaconda3\lib\site-packages\jupyterlab
[I 13:54:09.952 NotebookApp] JupyterLab application directory is C:\Users\PinkWink\Anaconda3\share\jupyter\lab
|
python.exe[64]:13768
```

# Jupyter Notebook 실행



- 웹 브라우저에 **localhost:8888/tree**라는 주소에 접속한 것을 알 수 있음

- 여기서 잠깐...
- 우리 수업은 기본 웹 브라우저로 구글 크롬을 사용
  - 크롬을 사용하지 않으면 수업과 100% 호환되지 않을 수 있음
- 추가로 **sublime text**를 사용하는 것도 추천함

# Jupyter Notebook 실행

The screenshot shows a Jupyter Notebook interface with a top navigation bar containing links like PW-GH, Personal, Work, Google, NAVER, Wiki, 사전 (Dictionary), 지도 (Map), 번역 (Translation), UsefulSW, Study, and Bookmarks. A 'Logout' button is also present. Below the navigation bar, there are tabs for 'Files', 'Running', and 'Clusters', with 'Files' being the active tab. A message 'Select items to perform actions on them.' is displayed above a file list. The file list includes a dropdown menu for selecting multiple files, a folder icon, and a count of 0 items. The list itself contains several folders: '3D Objects', 'Anaconda3', 'AnacondaProjects', 'Contacts', 'Desktop', 'Documents', 'Downloads', and 'Favorites'. To the right of the file list are columns for 'Name' and 'Last Modified', with corresponding timestamps for each item.

Name	Last Modified
3D Objects	22일 전
Anaconda3	5분 전
AnacondaProjects	12일 전
Contacts	22일 전
Desktop	2일 전
Documents	24분 전
Downloads	4시간 전
Favorites	22일 전

- 아무튼 이 화면에서 우리가 만든 경로로 찾아가기

The screenshot shows a Jupyter Notebook interface. At the top right are 'Logout' and 'Logout' buttons. Below them is a navigation bar with 'Files' (highlighted in red), 'Running', and 'Clusters'. A message 'Select items to perform actions on them.' is displayed. On the left, there's a file browser with a folder icon, '0' files, and a dropdown menu. The path '/ Documents/ FastCampus/ source\_code' is highlighted with a red box. To the right are 'Upload' and 'New' buttons, also highlighted with a red box. Below these are sorting options 'Name' and 'Last Modified' with a dropdown arrow, and a date filter '몇초 전'. At the bottom center, it says 'The notebook list is empty.'

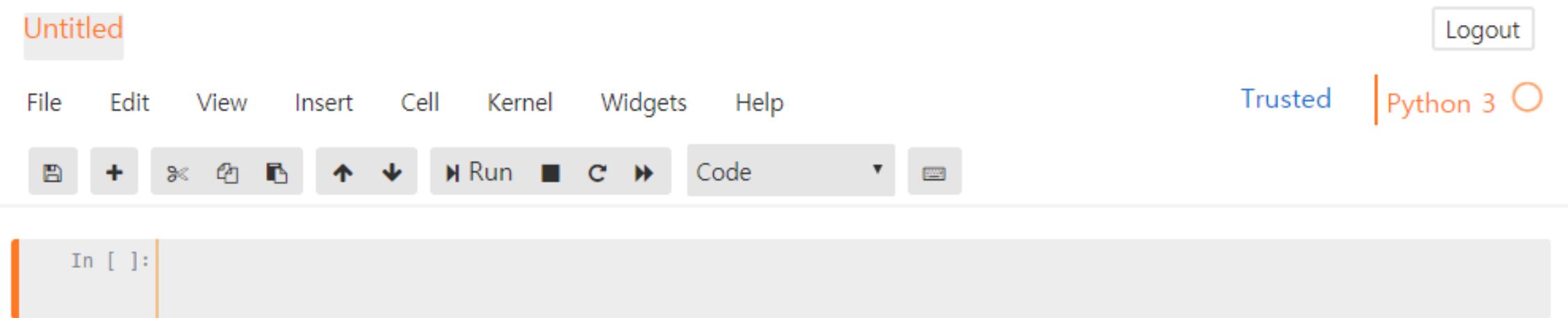
- 학습할 코드가 위치할 경로를 본능적으로 확인하고
- **New** 버튼 누르기

# Jupyter Notebook 실행

The screenshot shows the Jupyter Notebook interface. At the top, there are three tabs: 'Files' (highlighted in orange), 'Running', and 'Clusters'. Below the tabs, it says 'Select items to perform actions on them.' There is a file browser showing '0' files in the current directory ('Documents/ FastCampus/ source\_code'). A red box highlights the 'Python 3' option in the 'Notebook' section of the context menu, which is displayed on the right side of the screen. The menu also includes options for 'Text File', 'Folder', and 'Terminal'.

- Python3 선택

- Jupyter notebook의 전체 화면



---

# Hello World

---

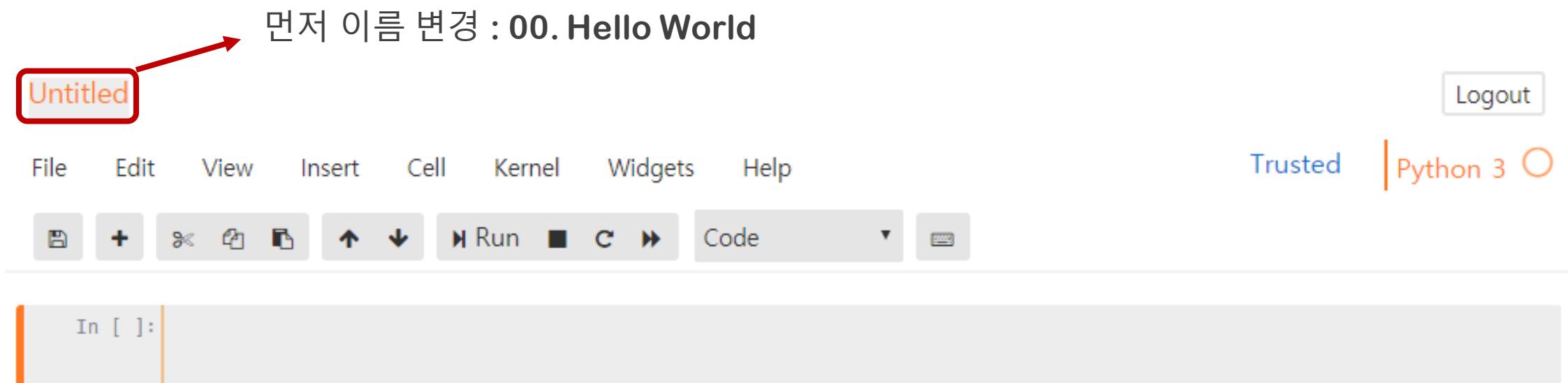
먼저 이름 변경 : 00. Hello World

Untitled

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 O

In [ ]:



The screenshot shows a Jupyter Notebook interface. At the top left, there is a title bar with the text "Hello World". Below it is a toolbar with various icons for file operations like saving, opening, and running cells. The main area is a code editor with the placeholder text "In [ ]:". In the top right corner, there are user authentication details: "Logout", "Trusted", and "Python 3 O". A red arrow points from the text "먼저 이름 변경" to the "Untitled" tab, which is highlighted with a red border.

## 00. Hello World

[Logout](#)

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3



In [ ]:

- 여기를 클릭하고

```
In [ ]: s1 = "Hello"  
         s2 = "World"  
  
         print(s1 + "!" + s2)
```

- 위와 같이 입력.
- 입력 후에 **Shift + Enter**

```
In [1]:  
s1 = "Hello"  
s2 = "World"  
  
print(s1 + "! " + s2)
```

```
Hello! World
```

- **Shift + Enter** 는 입력 셀에 있는 코드를 실행하라는 의미
- 실행되고 출력할 내용이 있으면 입력셀 바로 아래에 출력값이 나타남
- 실행되면 변수 내용이 변경됨

---

# Jupyter Notebook을 사용하는데 유용한 명령 소개

```
In [1]:  
s1 = "Hello"  
s2 = "World"  
  
print(s1 + "! " + s2)
```

Hello! World

```
In [ ]:  
s1  
In [ ]:  
Type: str  
String form: Hello  
Length: 5
```

- 커서가 **s1** 뒤에 있을 때 **Shift+Tab**을 입력하면 변수에 대한 정보가 나타남

```
In [2]: s1?  
In [ ]:
```

- 뒤에 ?를 붙여서 실행(**Shift+Enter**)하면 화면 하단에 정보가 나타남

```
Type: str
String form: Hello
Length: 5
Docstring:  
str(object='') -> str  
str(bytes_or_buffer[, encoding[, errors]]) -> str  
  
Create a new string object from the given object. If encoding or  
errors is specified, then the object must expose a data buffer  
that will be decoded using the given encoding and error handler.  
Otherwise, returns the result of object.__str__()  
or repr(object).  
encoding defaults to sys.setdefaultencoding().  
errors defaults to 'strict'.
```

In [5]:

```
%ls
```

C 드라이브의 볼륨에는 이름이 없습니다.

볼륨 일련 번호: D8C4-3FD9

C:\Users\PinkWink\Documents\FastCampus\source\_code 디렉터리

```
2018-05-05 오후 05:49    <DIR>        .
2018-05-05 오후 05:49    <DIR>        ..
2018-05-05 오후 02:08    <DIR>        .ipynb_checkpoints
2018-05-05 오후 05:49            1,001 00. Hello World.ipynb
                           1개 파일           1,001 바이트
                           3개 디렉터리   192,213,581,824 바이트 남음
```

- **%ls** : 현재 폴더의 파일 목록을 보여줌

In [6]:

```
%pwd
```

```
'C:\\\\Users\\\\PinkWink\\\\Documents\\\\FastCampus\\\\source_code'
```

- **%pwd**: 현재 경로를 보여줌

```
In [7]: %cd ..
```

```
C:\Users\PinkWink\Documents\FastCampus
```

```
In [9]: %pwd
```

```
'C:\\\\Users\\\\PinkWink\\\\Documents\\\\FastCampus'
```

- **%cd** : 폴더 이동을 의미
- **..** : 한단계 상위 폴더를 의미

In [ ]:

```
%cd source_code/
```

- 다 입력하지말고 , **%cd sou** 정도에서 **Tab** 키를 눌러보자
- 앞으로, 습관적으로 **Tab**을 눌러보면 자연스럽게 **tab**의 역할을 알게 된다.

```
In [10]:
```

```
%cd source_code/
```

```
C:\Users\PinkWink\Documents\FastCampus\source_code
```

```
In [11]:
```

```
%pwd
```

```
'C:\\\\Users\\\\PinkWink\\\\Documents\\\\FastCampus\\\\source_code'
```

- 다시 제자리로 ~

In [12]:

```
%whos
```

Variable	Type	Data/Info
-----		
s1	str	Hello
s2	str	World

- **%whos** : 현재 사용되는 변수 목록 확인

In [13]:

```
%history
```

```
s1 = "Hello"  
s2 = "World"  
  
print(s1 + " ! " + s2)  
s1?  
s1??  
print??  
%ls  
%pwd  
%cd ..  
%l -
```

- **%history** : 입력한 명령어 history를 보여줌

```
%pwd  
%whos  
%history
```

```
In [14]: %timeit
```

```
In [19]: %timeit for _ in range(1000): True
```

38.4  $\mu$ s  $\pm$  1.24  $\mu$ s per loop (mean  $\pm$  std. dev. of 7 runs, 10000 loops each)

- **%timeit** : 실행되는 시간을 표시

```
In [20]: %matplotlib inline
```

- **%matplotlib inline** : **matplotlib**의 결과를 아웃섹션에 그대로 표현

In [26]:

```
!conda list
```

```
# packages in environment at C:\Users\PinkWink\Anaconda3:  
#  
# Name          Version   Build  Channel  
_ipyw_jlab_nb_ext_conf    0.1.0    py36he6757f0_0  
alabaster           0.7.10   py36hcd07829_0  
anaconda            5.1.0     py36_2  
anaconda-client       1.6.9     py36_0  
anaconda-navigator    1.7.0     py36_0  
anaconda-project      0.8.2    py36hfad2e28_0  
asn1crypto           0.24.0    py36_0  
astroid              1.6.1     py36_0  
astropy              2.0.3    py36hfa6e2cd_0  
attrs                17.4.0    py36_0  
babel               2.5.3     py36_0
```

- !표를 이용해서 터미널 명령을 바로 입력할 수 있음

```
In [27]: !dir/w
```

C 드라이브의 볼륨에는 이름이 없습니다.

볼륨 일련 번호: D8C4-3FD9

C:\Users\PinkWink\Documents\FastCampus\source\_code 디렉터리

[.]

[..]

[.ipynb\_checkpoints]

00. Hello World.ipynb

1개 파일

23,023 바이트

3개 디렉터리 192,197,017,600 바이트 남음

# Mark Down 문서

– Jupyter를 아름답고 실용적으로 꾸며주는 도구

## 00. Hello World

[Logout](#)

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3



- 여기를 클릭하면, **Code**와 **Markdown**을 선택할 수 있다.
- **Code** : Python 코드
- **Markdown** : 해당 셀을 마크다운 형식의 문서를 작성할 수 있도록 함

## 마크다운

위키백과, 우리 모두의 백과사전.

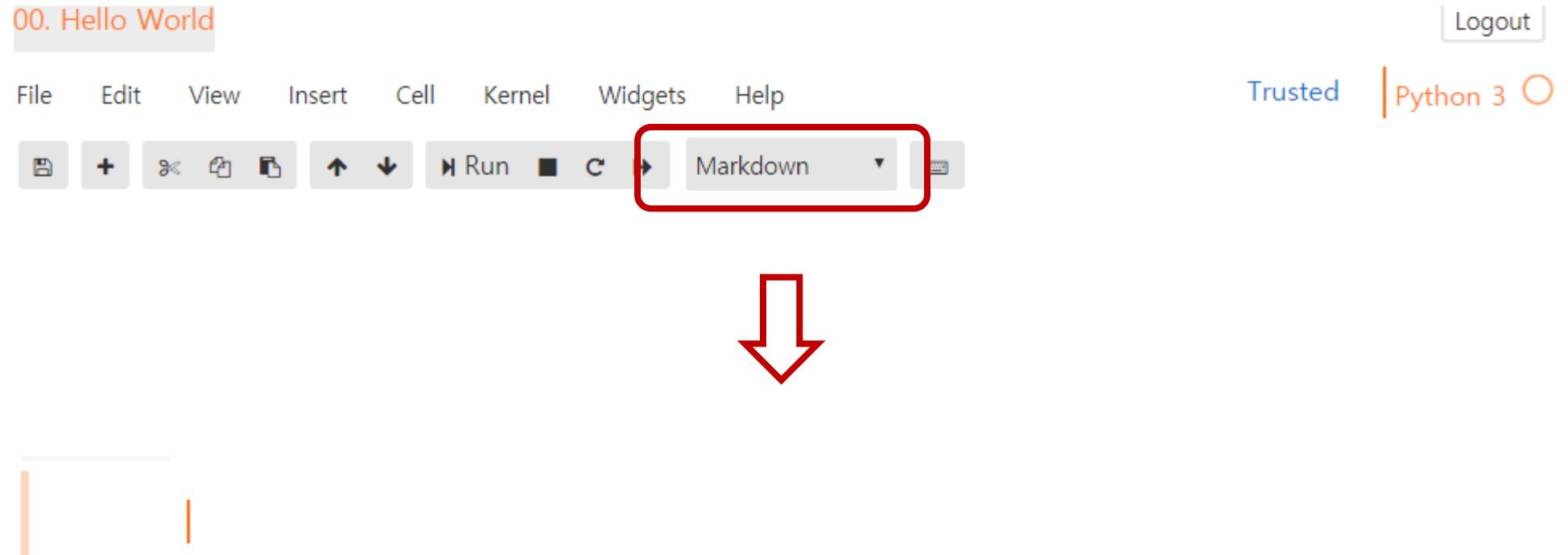
마크다운(markdown)은 일반 텍스트 문서의 양식을 편집하는 문법이다<sup>[1]</sup>. README 파일이나 온라인 문서, 혹은 일반 텍스트 편집기로 문서 양식을 편집할 때 쓰인다. 마크다운을 이용해 작성된 문서는 쉽게 HTML 등 다른 문서형태로 변환이 가능하다.

목차 [숨기기]

- 1 역사
- 2 문법
- 3 같이 보기
- 4 각주
- 5 외부 링크



출처 : 위키백과



- 셀 번호가 사라진다.

## 가속도 센서를 이용하여 위치를 구하는 과정에 대한 학습

2016.07.13 민형기 책임연구원 작성

- 본 문서는 최근 작성한 가속도 센서를 두 번 적분하여 위치를 구하는 과정을 다시 정리하고, 시퀀셜한 절차로 마지막에 작성해서 코드의 분량을 확인하고자 한다.

### 1. 가속도 센서의 파형 관찰 및 High Pass Filter와 Discrimination Window 적용

- Sample Rate이 4ms 이므로 데이터의 길이에 맞춰 시간(t)을 미리 정의해 둔다

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as patches
%matplotlib inline

rawData = np.genfromtxt('rawDataForExEstimationPositionUsingAccelerometer.csv', delimiter=',')

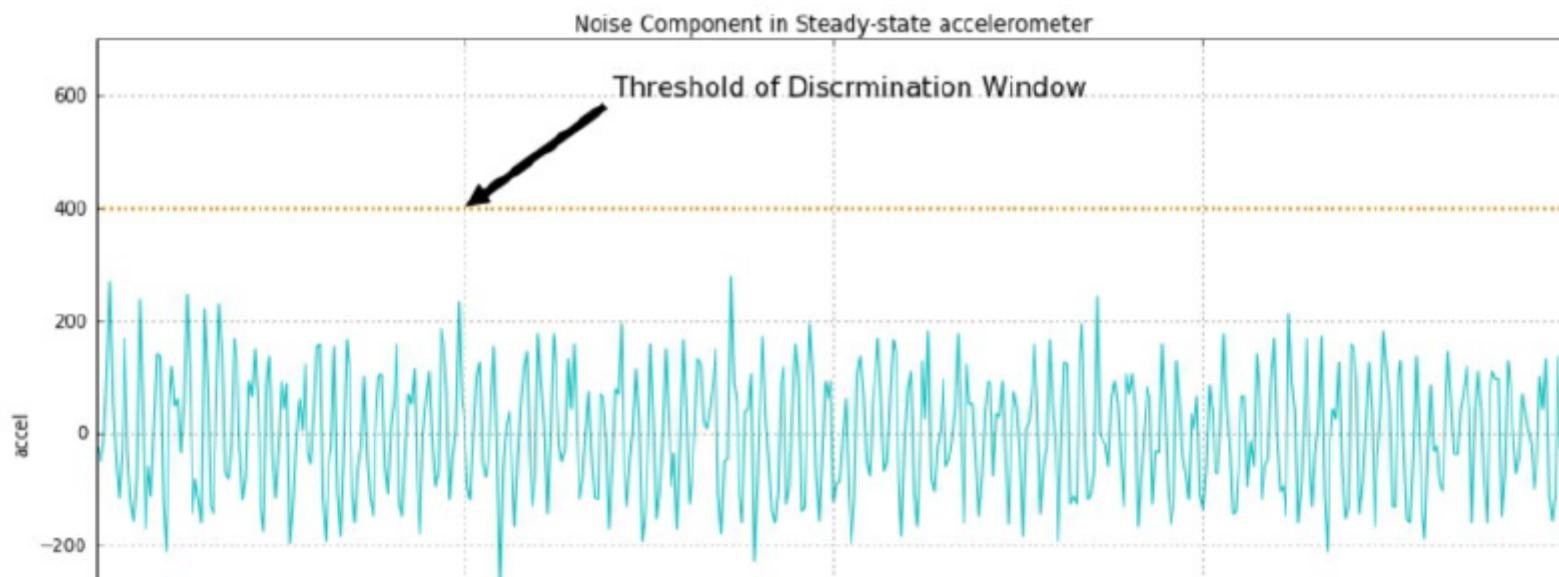
acc_rawdata = rawData[:,0]
acc_rawdata = acc_rawdata.astype(float)

Ts = 0.004
Fs = 1/Ts

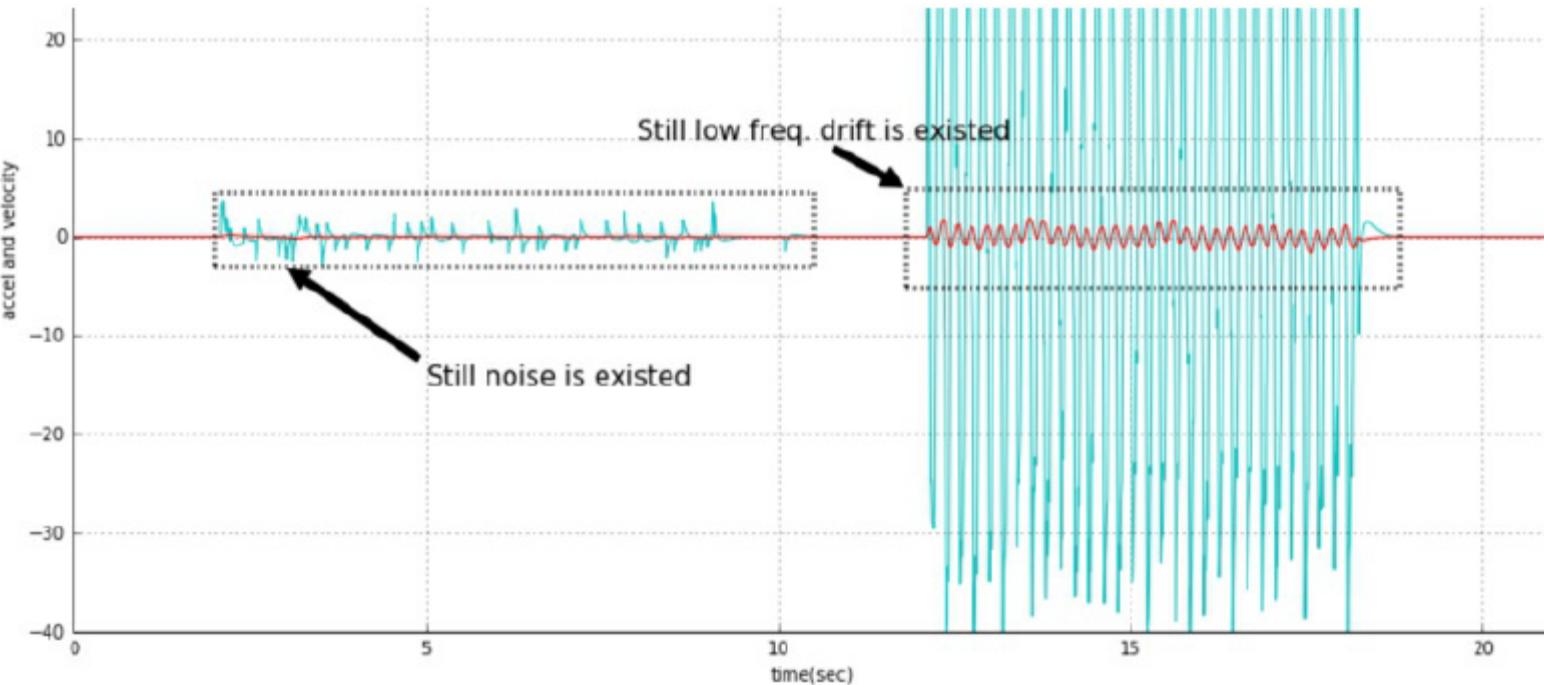
testLength = float(len(acc_rawdata))
t = np.arange(0, testLength*Ts, Ts)
```

출처 : 강사의 회사내 보고서

```
In [7]: plt.figure(figsize=(15,8))
plt.plot(t, acc, 'c')
plt.grid(True)
plt.xlabel('time(sec)')
plt.ylabel('accel')
plt.title('Noise Component in Steady-state accelerometer')
plt.axis([0, 2, -700, 700])
plt.hlines(400,0,2, linewidth=3, linestyle='dotted', color='orange')
plt.hlines(-400,0,2, linewidth=3, linestyle='dotted', color='orange')
plt.annotate('Threshold of Discrimination Window', xy=(0.5,400), xytext=(0.7,600), size=15, arrowprops=dict(facecolor='black', shrink=0.02))
plt.show()
```



출처 : 강사의 회사내 보고서

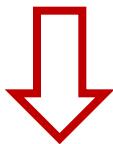


- 위 그래프에서는 두 가지 문제가 관찰되는데, 하나는 첫 부분의 속도입장에서 노이즈처럼 보이는 구간이다.
- 이 부분은 실험 전 실제로 움직인 부분인지 혹은 노이즈인지 데이터만 봐서는 판단하기 어려워 추후 다시 실험할 때 확인하도록 한다
- 두 번째는 저주파 성분의 스윙이 아직도 관찰된다는 것이다
- 이 경우는 이전의 HPF의 계수를 조정하는 것과, 구한 위치 결과에 다시 HPF를 적용하는 것으로 방법을 나눠볼 수 있다.
- 본 문서에서는 다시 HPF를 적용하는 것으로 한다

출처 : 강사의 회사내 보고서

### # Markdown 문서

- \* 별표를 사용하면 목록형으로 기술 가능
- \* 마크 다운은 Jupyter Notebook이 문서로서의 가치를 가지도록 하는 중요한 도구임



### Markdown 문서

- 별표를 사용하면 목록형으로 기술 가능
- 마크 다운은 Jupyter Notebook이 문서로서의 가치를 가지도록 하는 중요한 도구임

# 제목

## 소제목

### 더 작은 소제목

\* 번호 없는 목록

\* \*\*두 개면 굵게\*\*, \*하나면 기울이기\*, \*\*\*세 개면 굵은 기울임\*\*\* 이렇게



## 제목

### 소제목

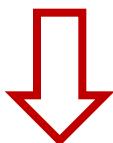
#### 더 작은 소제목

- 번호 없는 목록
- 두 개면 굵게, 하나면 기울이기, 세 개면 굵은 기울임 이렇게

## ## 가로 구분선

---

\* 이렇게 (---) 를 사용해서 가로 구분선을 사용할 수 있음



## 가로 구분선

---

- 이렇게 (---) 를 사용해서 가로 구분선을 사용할 수 있음

## ## 인용문

- \* Pandas : 단일 머신에 맞는 데이터셋으로 작업할 때 Pandas는 최고의 도구이며 스팀팩을 맞은 마린, 스테로이드 맞은 엑셀처럼 강력하다
  - > 언어와 상관없이 사용할 수 있는 가장 강력하고 유연한 오픈소스 데이터 분석/처리 도구가 되는 것이 목표
- \* 이런 Pandas는 Anaconda에 기본으로 포함되어 배포되고 있다



## 인용문

- Pandas : 단일 머신에 맞는 데이터셋으로 작업할 때 Pandas는 최고의 도구이며 스팀팩을 맞은 마린, 스테로이드 맞은 엑셀처럼 강력하다
  - 언어와 상관없이 사용할 수 있는 가장 강력하고 유연한 오픈소스 데이터 분석/처리 도구가 되는 것이 목표
- 이런 Pandas는 Anaconda에 기본으로 포함되어 배포되고 있다

## ## 수식

```
\begin{align}
\dot{x} &= \sigma(y-x) \\
\dot{y} &= \rho x - y - xz \\
\dot{z} &= -\beta z + xy
\end{align}

$$\begin{eqnarray}
x' &=& \sin\phi &+& z \cos\phi \\
z' &=& - & \cos\phi &+& z \sin\phi \\
\end{eqnarray}$$
```

- \* 달러 기호 하나를 이용해서 수식을 감싸면 문장내에서  $\sqrt{3x-1}+(1+x)^2$  이렇게 사용가능

## 수식

$$\dot{x} = \sigma(y - x)$$

$$\dot{y} = \rho x - y - xz$$

$$\dot{z} = -\beta z + xy$$

$$x' = x \sin \phi + z \cos \phi$$

$$z' = -x \cos \phi + z \sin \phi$$

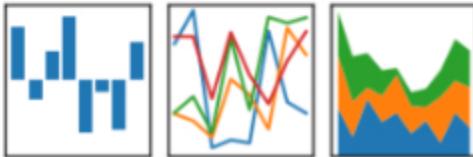
- 달러 기호 하나를 이용해서 수식을 감싸면 문장내에서  $\sqrt{3x - 1} + (1 + x)^2$  이렇게 사용가능

- 우리 수업과는 조금 멀어지게 되어서 수식 입력에 대한 부분은 생략...

## Pandas로 CSV, 엑셀 파일 읽기

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



[overview](#) // [get pandas](#) // [documentation](#) // [community](#) // [talks](#) // [donate](#)

pandas: powerful Python data analysis toolkit

- Python에서 R 만큼의 강력한 데이터 핸들링 성능을 제공하는 모듈
- 단일 프로세스에서는 최대 효율
- 코딩 가능하고 응용 가능한 엑셀로 받아들여도 됨
- 누군가는 스테로이드를 맞은 엑셀로 표현함

- 이렇게 만들어준 코드는?

### ## Pandas로 CSV, 엑셀 파일 읽기

```
<div align="center"></img></div>
```

- \* Python에서 R 만큼의 강력한 데이터 핸들링 성능을 제공하는 모듈
- \* 단일 프로세스에서는 최대 효율
- \* 코딩 가능하고 응용 가능한 엑셀로 받아들여도 됨
- \* 누군가는 **\*\*스테로이드를 맞은 엑셀\*\***로 표현함

## 데이터 얻으러 가기

Google 서울시 관서별 5대 범죄 현황

전체 이미지 뉴스 동영상 지도 더보기 설정 도구

검색 결과 약 16,400개 (0.60초)

서울특별시 관서별 5대 범죄 현황 - 공공데이터포털  
<https://www.data.go.kr/dataset/3075835/fileData.do> ▾

서울특별시 관서별 5대 범죄 발생 - 경거 현황(2010년). 매체유형 : 기타 / 텍스트 파일, 링크 건수 : 2 전체  
행 수 : N/A 확장자 : CSV 다운로드 횟수(바로가기 횟수) : 349.  
이 페이지를 여러 번 방문했습니다. 최근 방문 날짜: 17. 11. 25

서울시 관서별 5대 범죄 현황 관련 이미지

→ 서울시 관서별 5대 범죄 현황에 대한 이미지 더보기 이미지 신고

서울특별시 관서별 청소년 범죄 현황 ENGLISH - 공공데이터포털  
<https://www.data.go.kr/dataset/3075889/fileData.do> ▾

서울특별시 지방경찰청 관서별 청소년 5대 범죄 현황 매체유형 : 기타 파일, 링크 건수 : 1 전체 행 수 : N/A 확장자 : CSV 다운로드 횟수(바로가기 횟수) : 346 전체

- 구글에서 서울시 관서별 5대 범죄 현황이라고 검색하자
- 기사 원문 <http://news1.kr/articles/?1911504>
- 데이터 다운로드 경로 <https://www.data.go.kr/dataset/3075835/fileData.do>

## ## 데이터 얻으러 가기

```
<div align="center"></img></div>
```

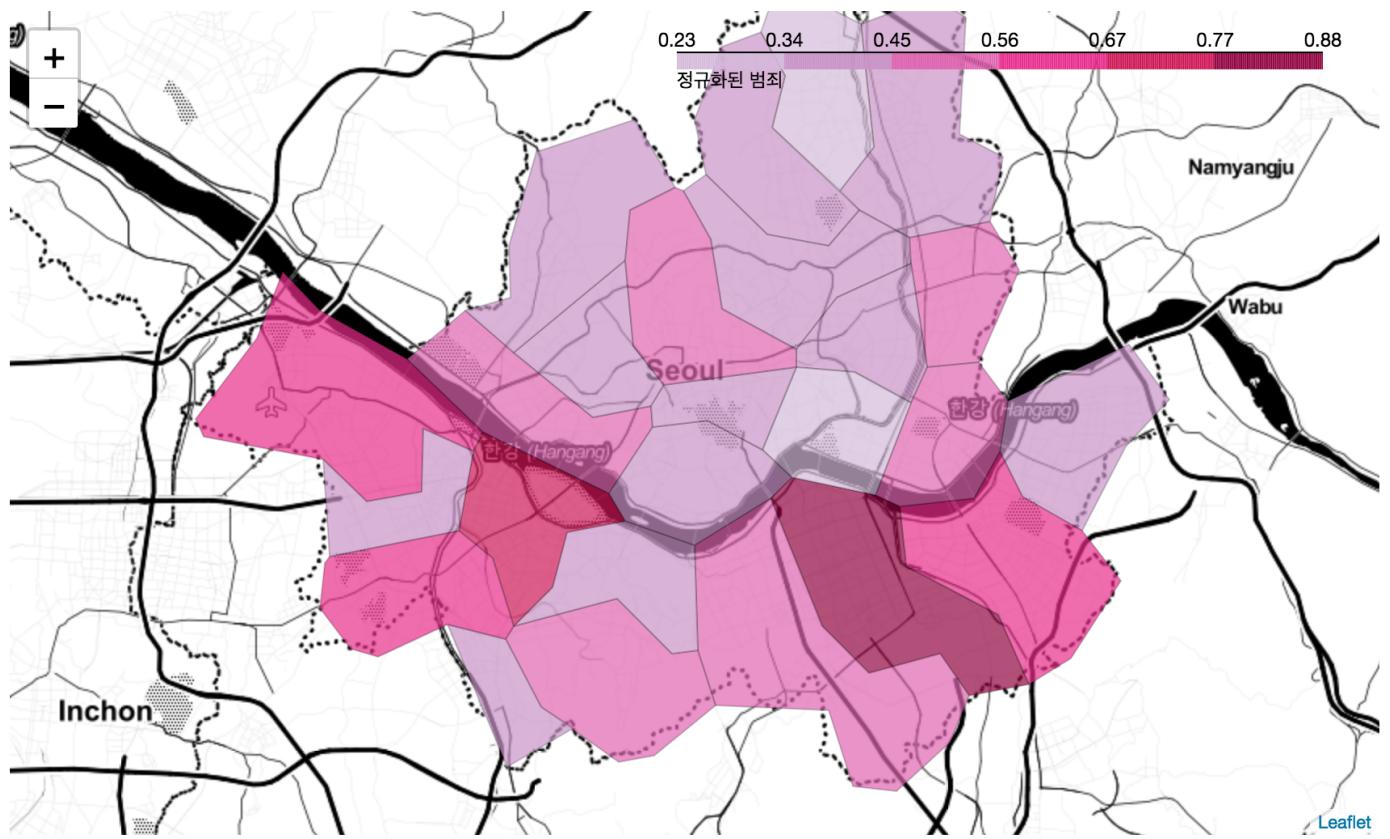
- \* 구글에서 \*\*서울시 관서별 5대 범죄 현황\*\*이라고 검색하자
- \* 기사 원문 <http://news1.kr/articles/?1911504>
- \* 데이터 다운로드 경로

<https://www.data.go.kr/dataset/3075835/fileData.do>

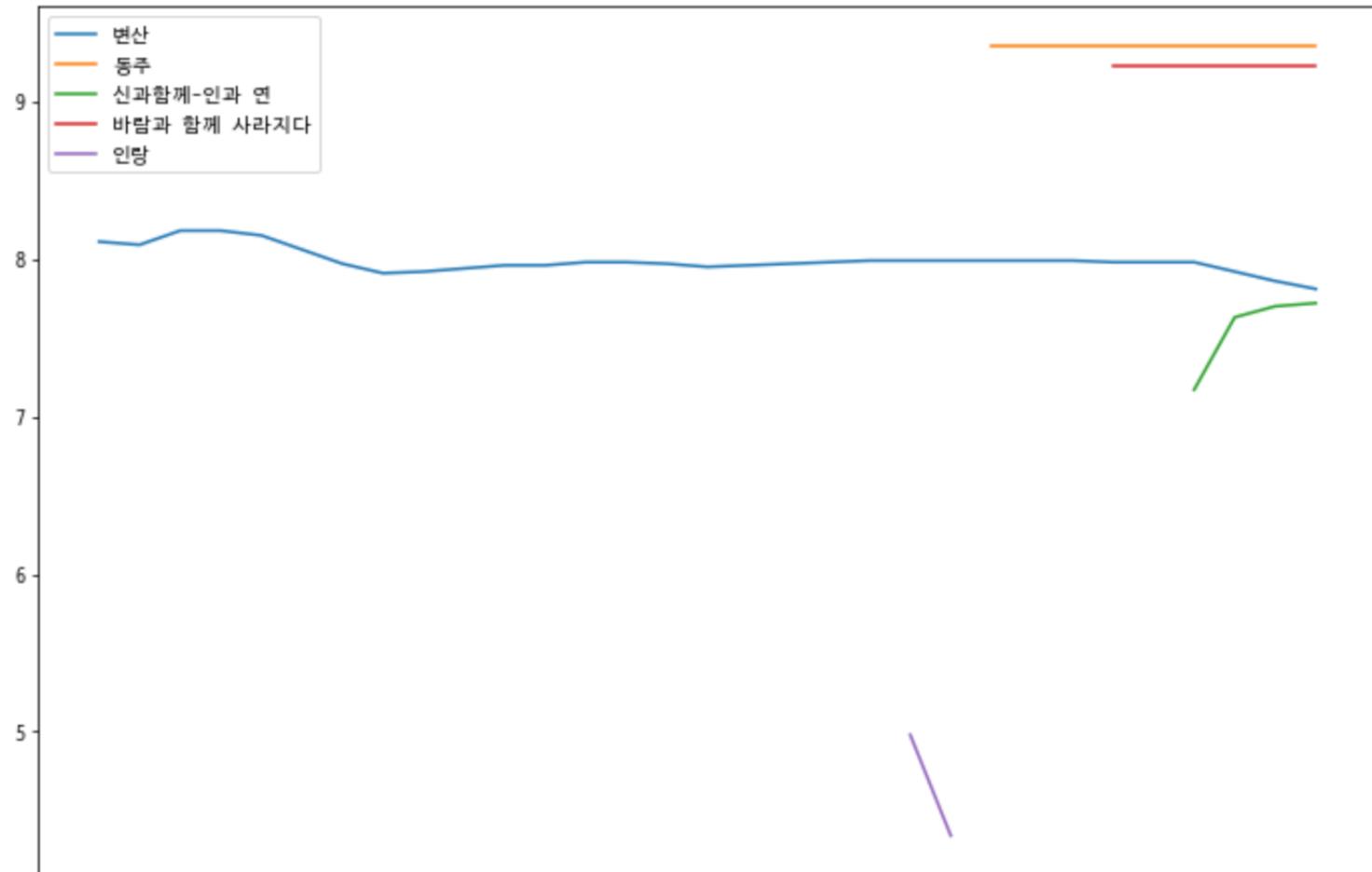
- 
- **Jupyter Notebook**은 그 자체로
    - 코드의 흐름을 설명하고
    - 실행 결과를 재실행없이 확인할 수 있어서
    - 문서로서의 가치를 가짐

# 이번 수업의 구성

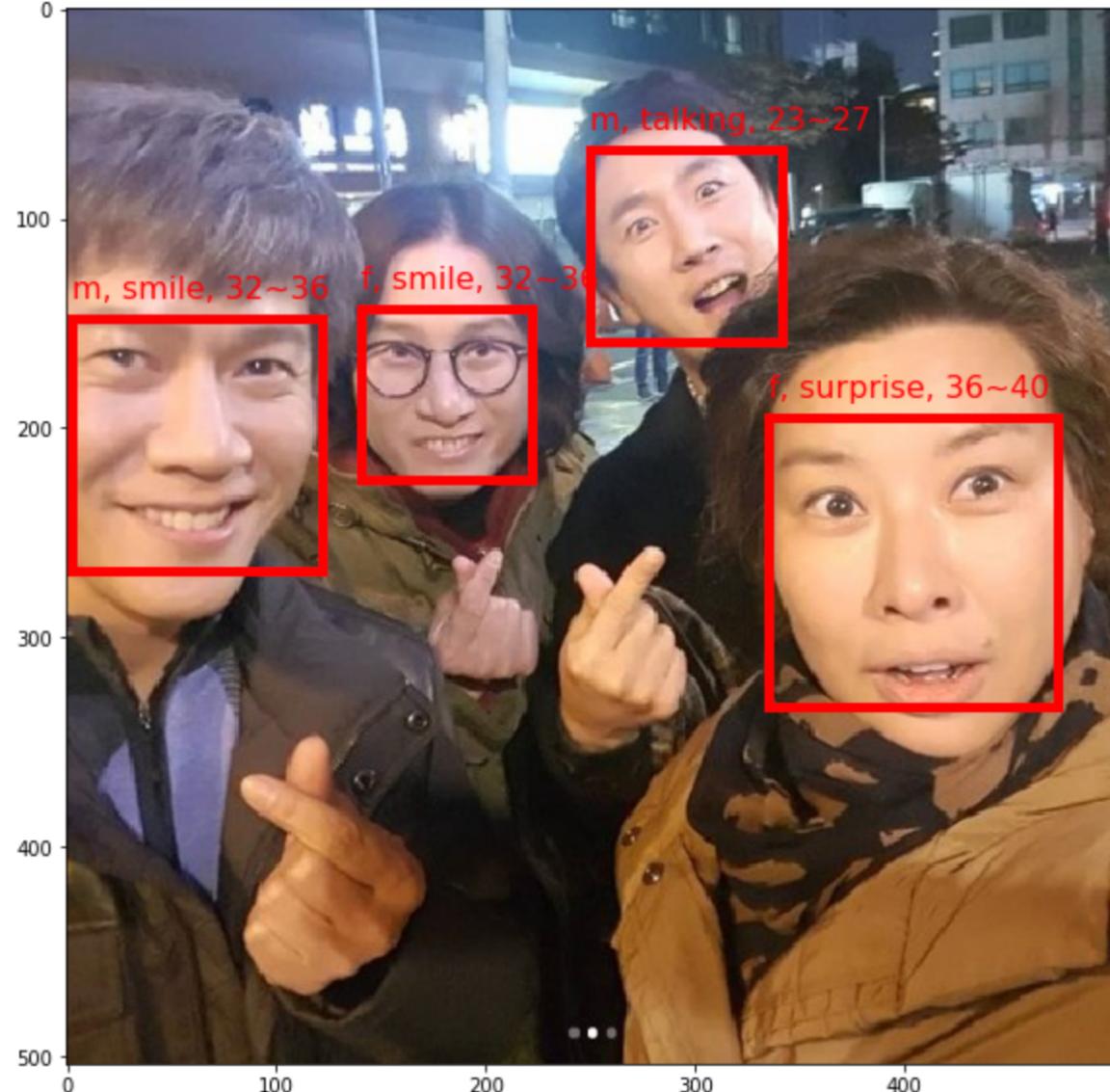
- 1일차 : 서울시 범죄 현황 분석
  - Python과 친해지기
  - Pandas 기초 응용
  - 데이터 시각화
  - 지도 시각화



- 2일차 : 영화 평점 시각화
  - 인터넷의 자료를 가져오기
  - 웹 스크래핑/크롤링 기초
  - 정규식의 간편 사용



- 3일차 : API 사용해보기
  - 네이버 API 사용
  - 다양한 검색 설정 사용해보기
  - 얼굴 인식



- 4일차 : 머신러닝(딥러닝) 구경해보기
  - Linear Regression
  - Decision Tree
  - MNIST 손글씨 분류하기
  - 타이타닉 생존자 분석하기



---

그럼 ~ 달려야죠 ~~~ !