



HELLO WORLD

이제 시작해 볼까요.

DATA VISUALIZATION IN PYTHON USING

BOKEH



수업 소개부터 하겠습니다.~^^

문제상황 해결 역량을 기르는
파이썬을 활용한 데이터 분석의 탄탄한 시작 -

파이썬을 활용한 데이터 분석 입문 CAMP

 Fast campus



- 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**

- 질문 환영 : 페이스북이든 메일이든

facebook : <https://www.facebook.com/hyunggi.min.16>

blog : <http://pinkwink.kr/>

mail : pinkwink.korea@gmail.com



한글로 된 날짜를 바꾸기...

 HYUNG GI MIN · 2018년 4월 1일 일요일

질문이 와서... 살짝 답변을 합니다.

가장 간단하게는

```
target_date = '2018년 04월 01일'
target_date = str.replace(target_date, '년 ', '-')
target_date = str.replace(target_date, '월 ', '-')
target_date = str.replace(target_date, '일', '')
target_date
```

뭐.. 이렇게 생각해 볼 수 있네요.. 그러면 결과가 2018-04-01 이라고 나옵니다. 그런데.. 안
이쁘네요...ㅠ ㅠ.

살짝 손을 조금 더 대면...

```
target_date = '2018년 04월 01일'
target_date = '.'.join(each_word[:-1] for each_word in target_date.split())
target_date
```

Part 1. 데이터 분석의 시작

① 데이터 분석을 시작해 보자 ▼

[실습] 서울시 인구현황 분석

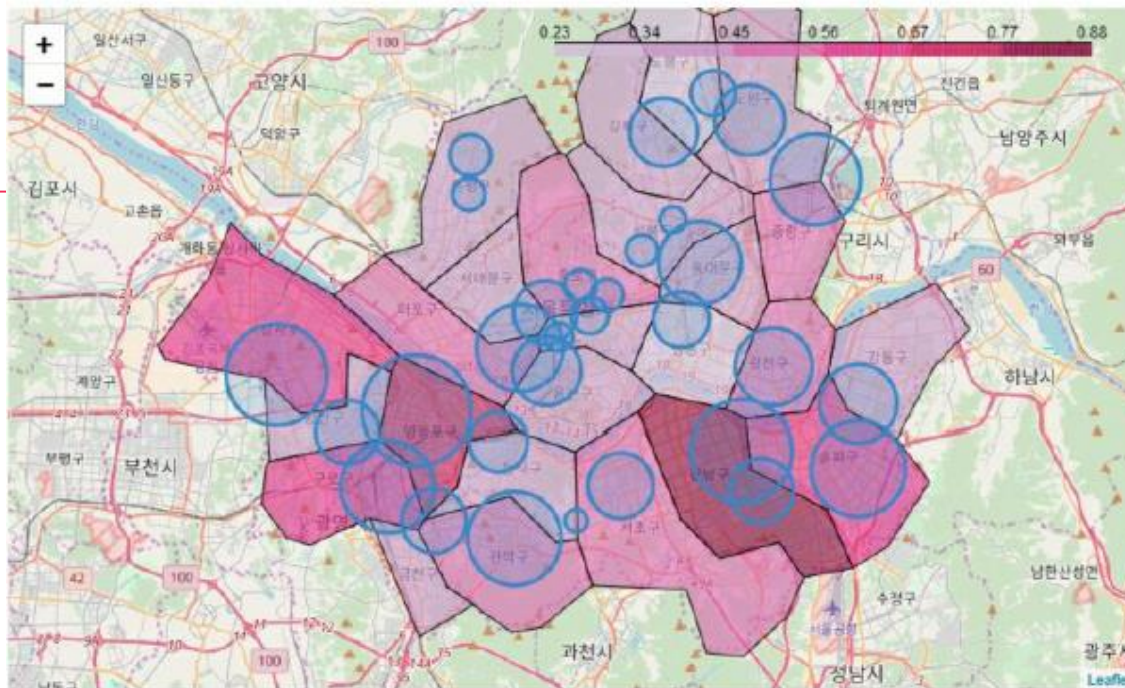
- 1) Anaconda 환경(conda env) 설정하기
- 2) Jupyter Notebook 환경에 익숙해지기
- 3) Markdown 문서 활용하기
- 4) 데이터 프레임의 정석, Pandas 익히기
- 5) Python 시각화 도구의 표준, Matplotlib 익히기
- 6) Python의 데이터 유형 익히기
- 7) Python에서 Excel이나 TXT 파일 읽기
- 8) 데이터를 정리하는 과정에 익숙해지기

- 당장 이번 주부터 3주간의 내용
- **Python, Pandas, Matplotlib**의 기초
- 지도 시각화 등의 광범위한 내용

② 데이터를 그려보자 ▼

[실습] 서울시 구 별 범죄 발생 현황 분석

- 1) Pandas의 고급 기능인 Pivot_Table 익히기
- 2) 주소 정보를 변환하는 Googlemaps API 익히기
- 3) 더 멋진 그래프를 그릴 수 있는 Seaborn 익히기
- 4) Pairplot/ Heatmap을 이용하여 하고 싶은 이야기를 그려보기
- 5) Folium을 이용한 지도 시각화 도전하기
- 6) Python의 조건문과 반복문을 이해하기



⊖ 웹 상에서의 데이터를 수집하여 분석해보자 ▼

[실습] 샌드위치 맛집 정보 분석하기

- 1) HTML의 기초와 스크래핑/ 크롤링의 개념 이해하기
- 2) 웹 페이지에서 원하는 정보를 얻기 위한 HTML 접근 방법
- 3) List와 반복문에 대해 깊이 있게 이해하기
- 4) 웹 페이지에서 접근 URL을 획득하고 페이지 탐색하기
- 5) 정규표현식(Regular Expression) 기초 익히기
- 6) Jupyter Notebook에서 반복문을 적용하기 편리한 TQDM 익히기

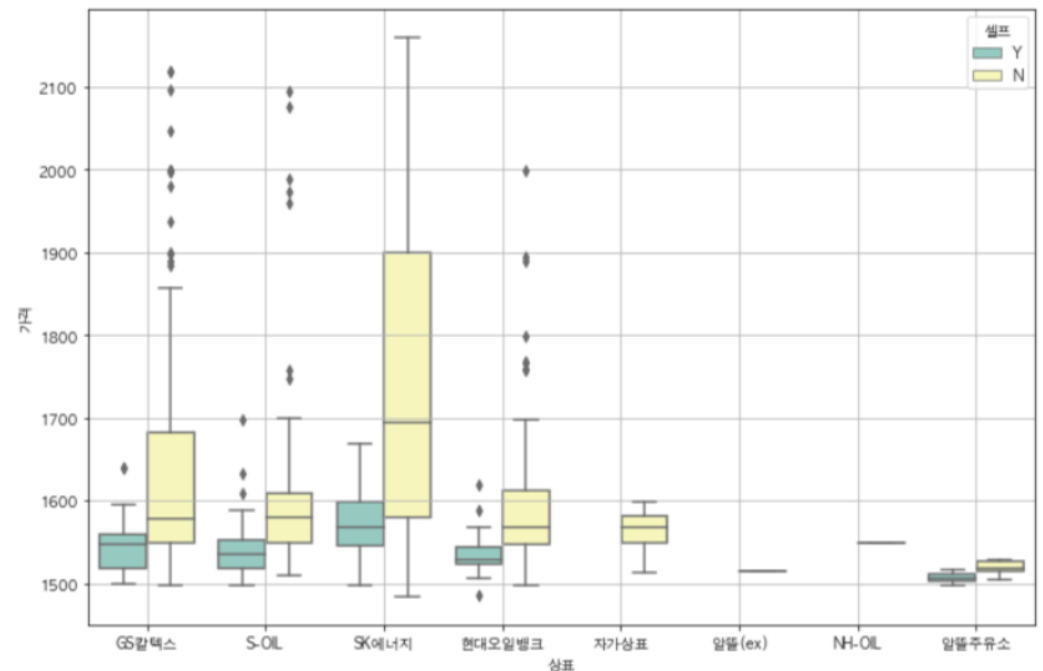
[실습] 네이버 영화 평점 시각화하기

- 1) 날짜 정보를 포함하는 데이터 이해하기
- 2) Pandas Datetime 데이터형 이해하기
- 3) 시간에 종속된 데이터 시각화하기

[실습] '셀프 주유소는 정말 가격이 저렴할까?' 확인하기

- 1) 오픈소스 플랫폼의 장점과 다양한 데이터 분석 사례 소개
- 2) 동적 환경에서 웹 데이터를 얻어올 수 있는 도구인 Selenium 익히기
- 3) Python 함수 (def) 개념 익히기
- 4) 내가 만든 함수 모듈로 사용해 보기
- 5) Python에서 여러 개의 엑셀 파일을 한 번에 읽거나 병합하는 방법 익히기
- 6) Boxplot의 개념 익히고 사용해 보기

- 인터넷의 정보를 얻어오는
웹 크롤링과 웹 스크래핑 기초
- 동적인 웹 환경에서 데이터를 얻어오는 방법



과거 데이터를 가지고 미래를 예측해 보자 ▼

[실습] 웹 트래픽 예측하기

- 1) Numpy 기초 익히기
- 2) Numpy를 이용하여 경향 분석하기
- 3) 선형 회귀분석의 개념을 이해하고 적용해 보기
- 4) Prophet 기초 익히기
- 5) Non-Stationary Seasonal Time Data에 대한 forecast 및 component 사용하기

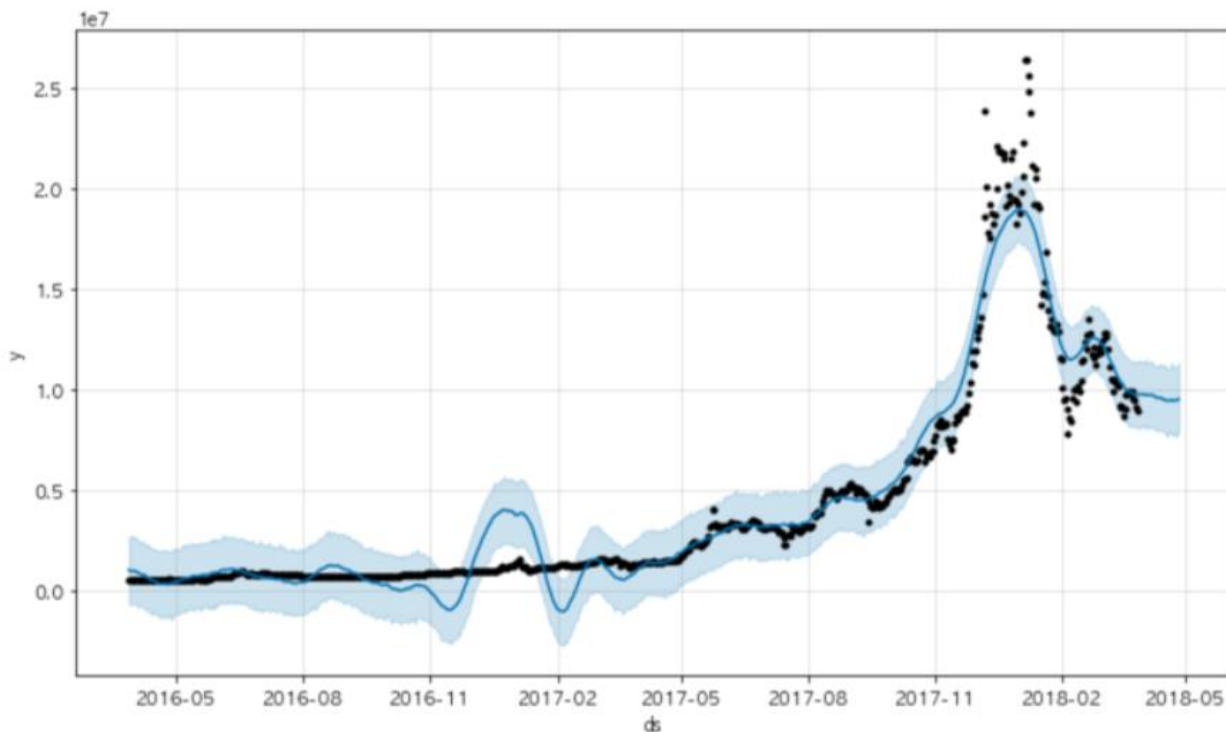
[실습] 주식 정보 확인하고 예측하기

- 1) 주가 정보를 웹페이지에서 스크래핑하기
- 2) 주가 정보를 pandas를 이용해서 얻어오기
- 3) 주가 정보에 대한 forecast 및 component 분석해 보기

[실습] 비트코인 시세 예측하기

- 1) 비트코인 시세 정보 스크래핑하기
- 2) Selenium에서 해당 요소가 나타날 때까지 기다리고 화면을 스크롤하는 기능 익히기
- 3) 비트코인 시세 정보에 대한 forecast 및 component 분석해 보기

- **Forecast** 모듈을 통한 예측
- 주가 정보, 비트코인 정보 예측 등등



② 텍스트 데이터를 분석해 보자 ▼

[실습] 비슷한 의미의 문장 고르기

- 1) 자연어 처리를 위한 모듈 설치법 확인하기
- 2) 간단한 형태소 분석을 통한 개념 이해하기
- 3) Wordcloud 학습하기
- 4) Naive Bayes 분류기 이해하기
- 5) 문장들 사이의 유사도 측정하기

[실습] 뉴스 검색 결과 분석하기

- 1) 특정 검색어를 통해 뉴스 분석하기
- 2) 뉴스 사이트 크롤링 방법 익히기
- 3) Try-Except 이해하기
- 4) 뉴스 검색을 통해 트렌드 분석하기
- 5) Gensim 적용하기

- 자연어 처리 기초
- 자연어 처리를 위한 모듈 사용법
- 웹 크롤링을 이용한 자연어 처리 예제



① 딥러닝을 이용해 데이터를 분석해 보자 ▼

[실습] 나이, 체중에 따른 혈중 지방 함량 문제 분석하기

- 1) 딥러닝을 위한 Python 모듈의 기본 사용법 익히기
- 2) 단층 신경망에 대해 이해하기

[실습] MNIST 손글씨 분석하기

- 1) 다층 신경망, CNN 등 코드로 딥러닝 이해하기

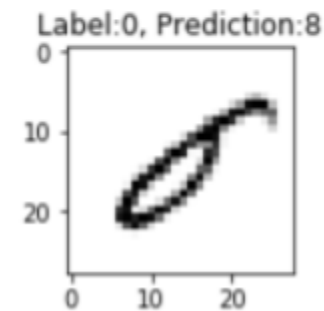
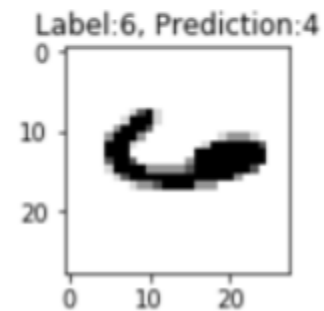
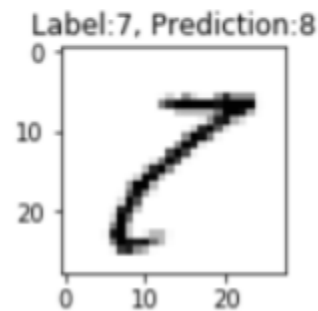
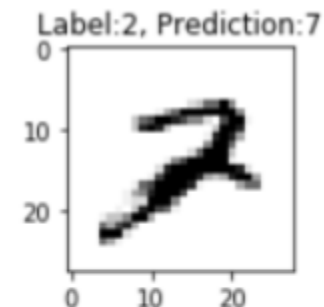
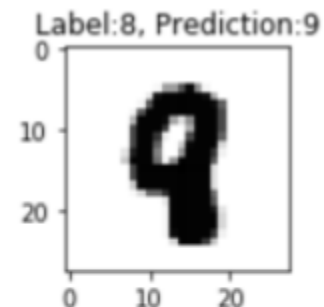
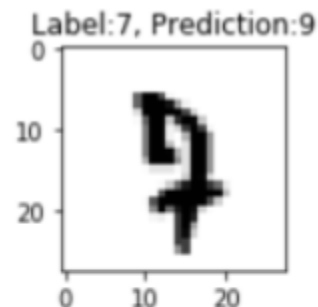
[실습] 내 손글씨 판독하기

- 1) 학습을 마친 딥러닝 적용해 보기

[실습] '디카프리오가 살아남을 수 없었을까?' 확인하기

- 1) 타이타닉호 생존자 예측하기

- 딥러닝 등의 머신러닝 기초
- 예제를 통해 머신러닝 맛보기



- 다루는 예제는 변동될 수 있음
- 상황에 따라 분량이 가변될 수 있음
- 강사도 직장인이라 한 주 정도 휴강 후 보강 상황이 발생할 수 있음
(최선을 다해 그런 일이 없도록 하겠습니다.~~~)
- 스피드~~~~

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](#)

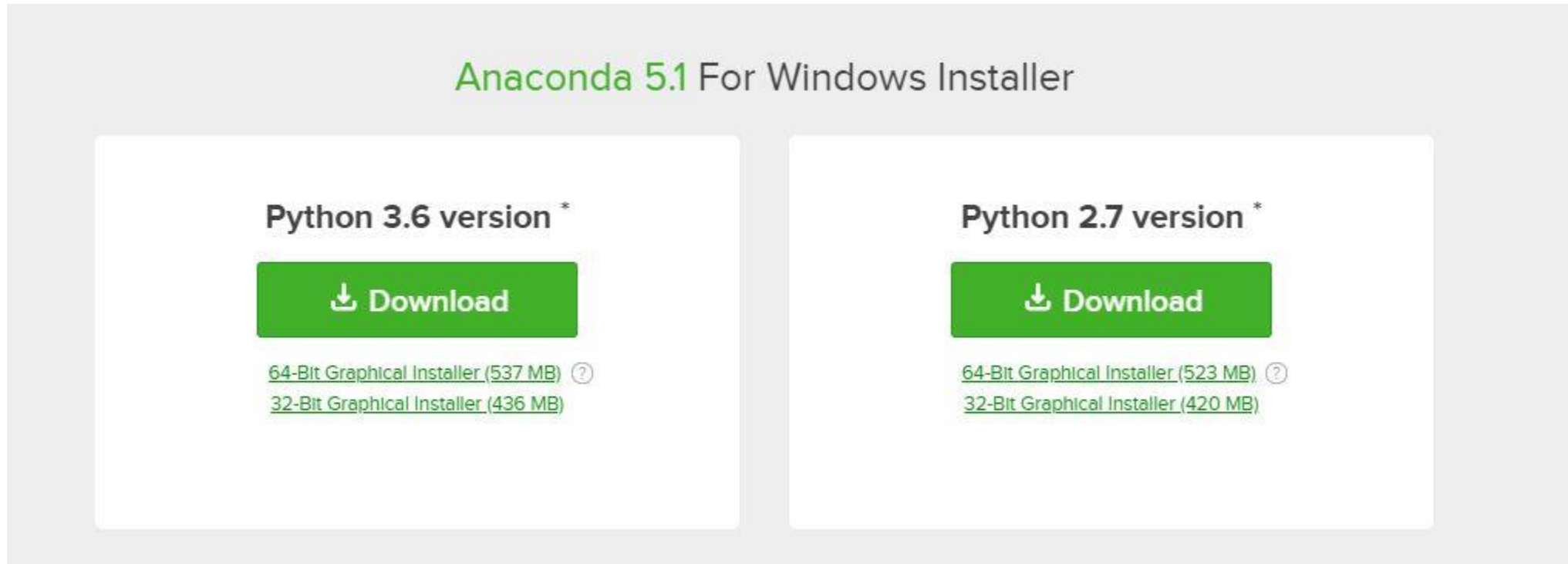
Portal to Data Science

Uncover insights in your data and create interactive visualizations

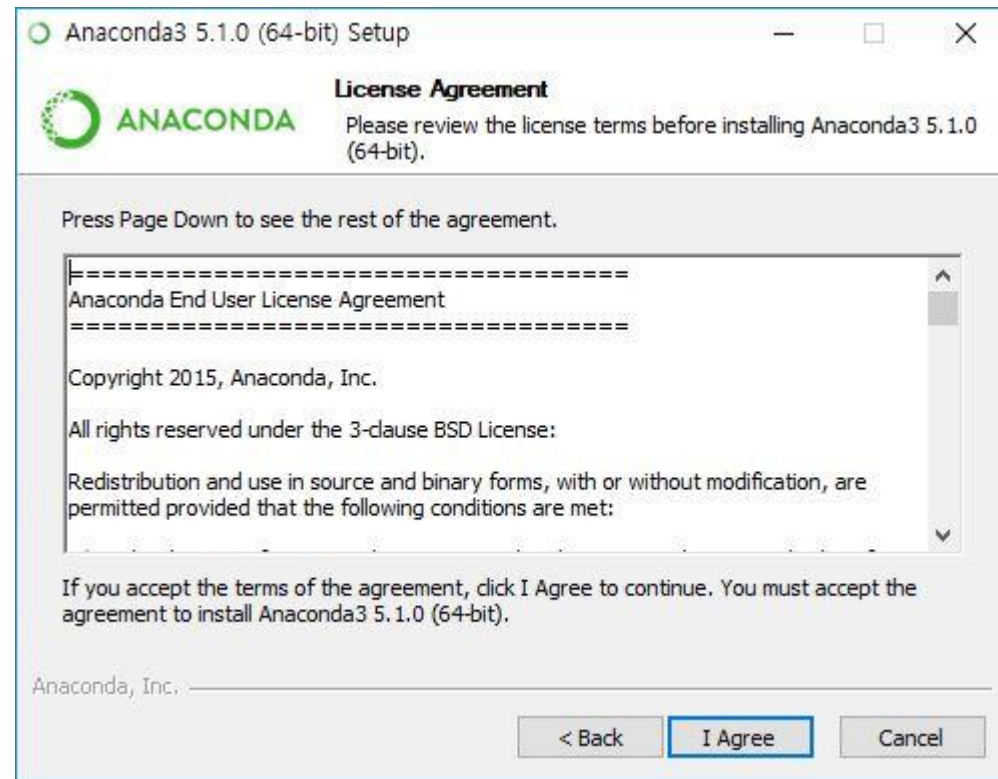
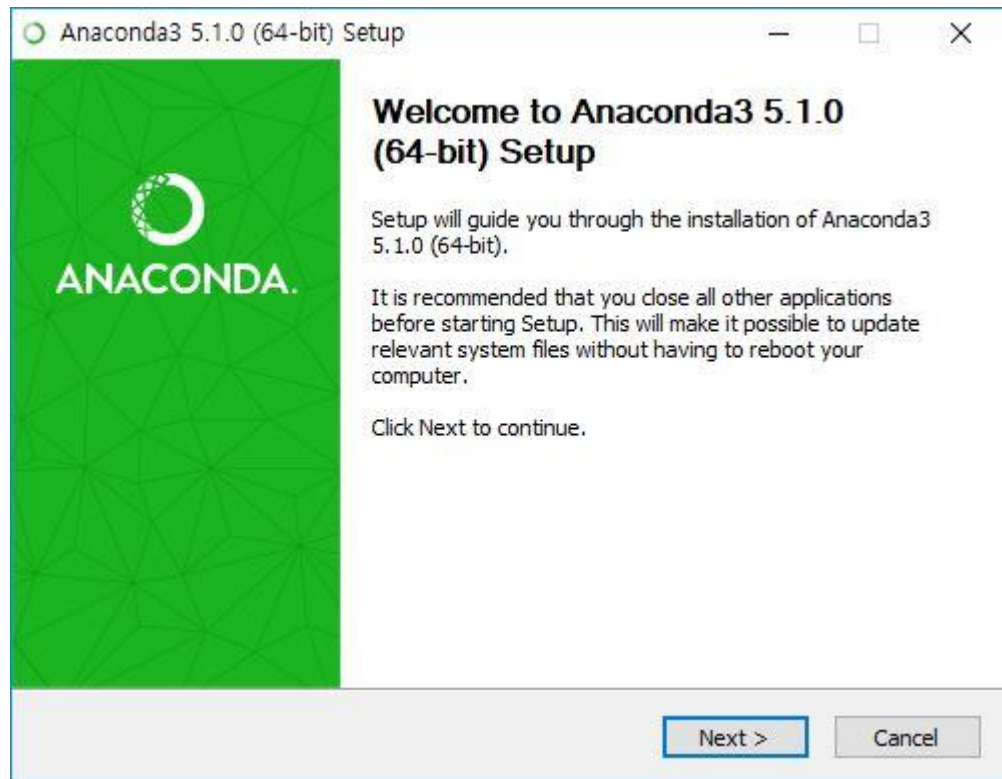
 Windows

 macOS

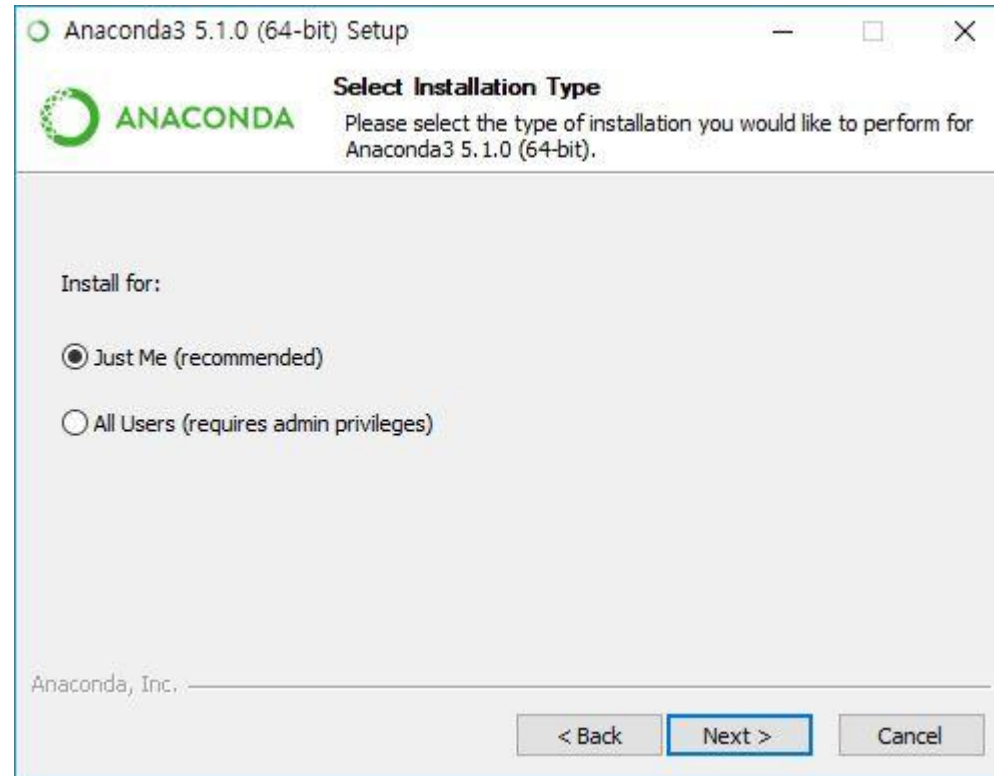
 Linux



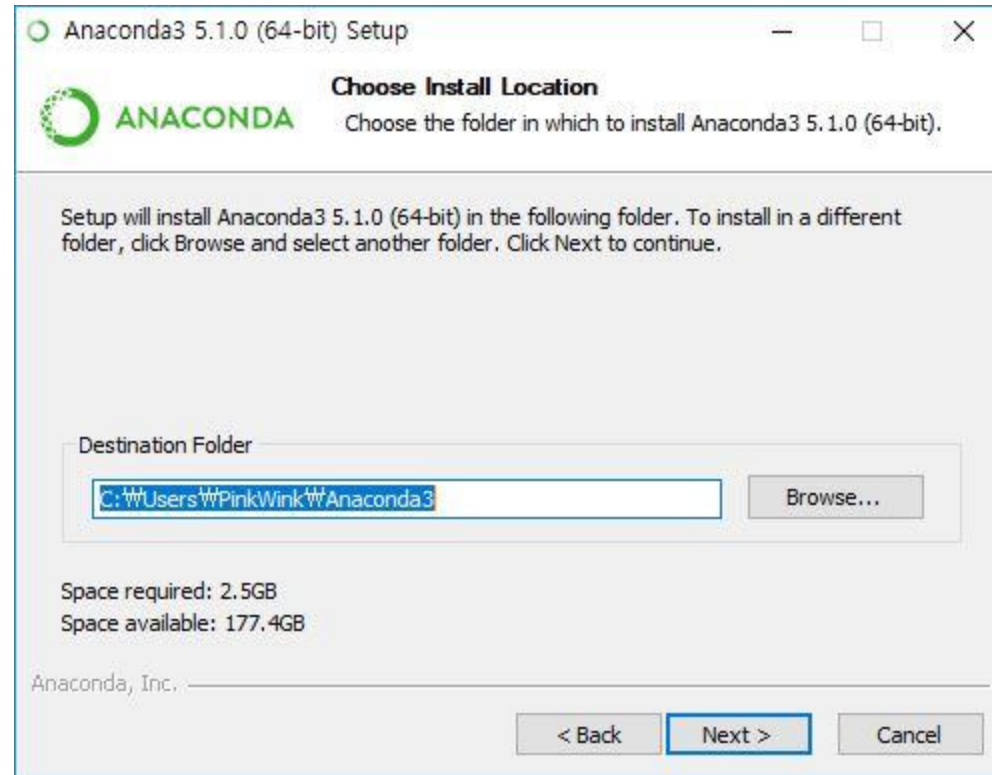
- 꼭~ Python 3.6으로 설치



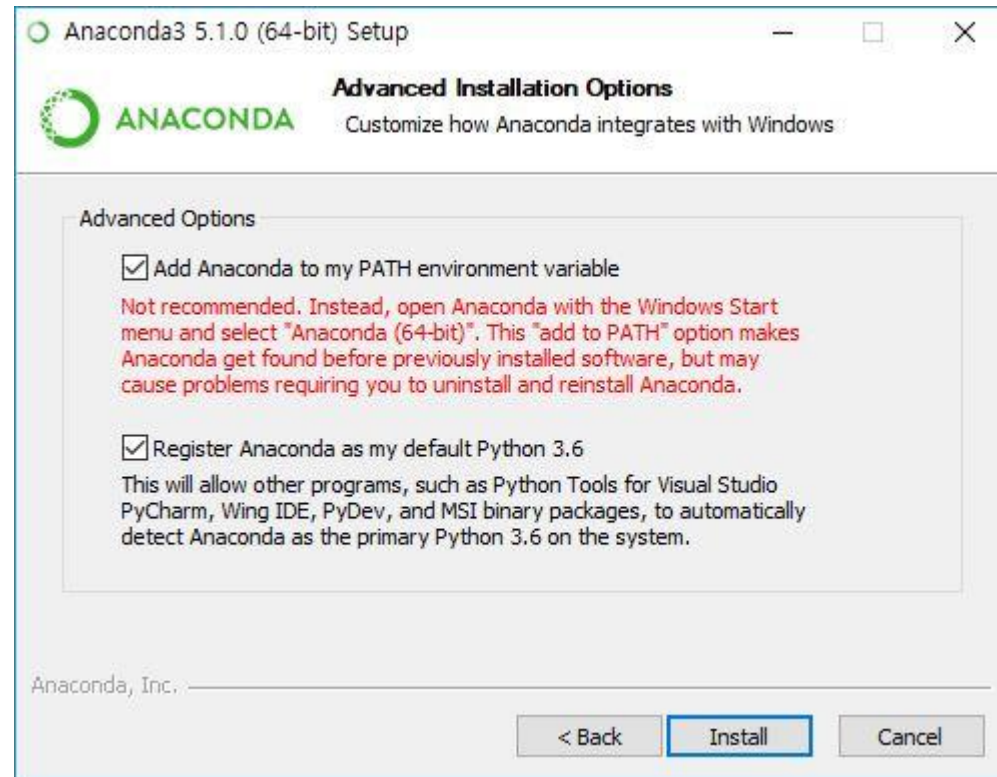
- 뭐 **Next**만 누르면 되는 편한 설치~



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



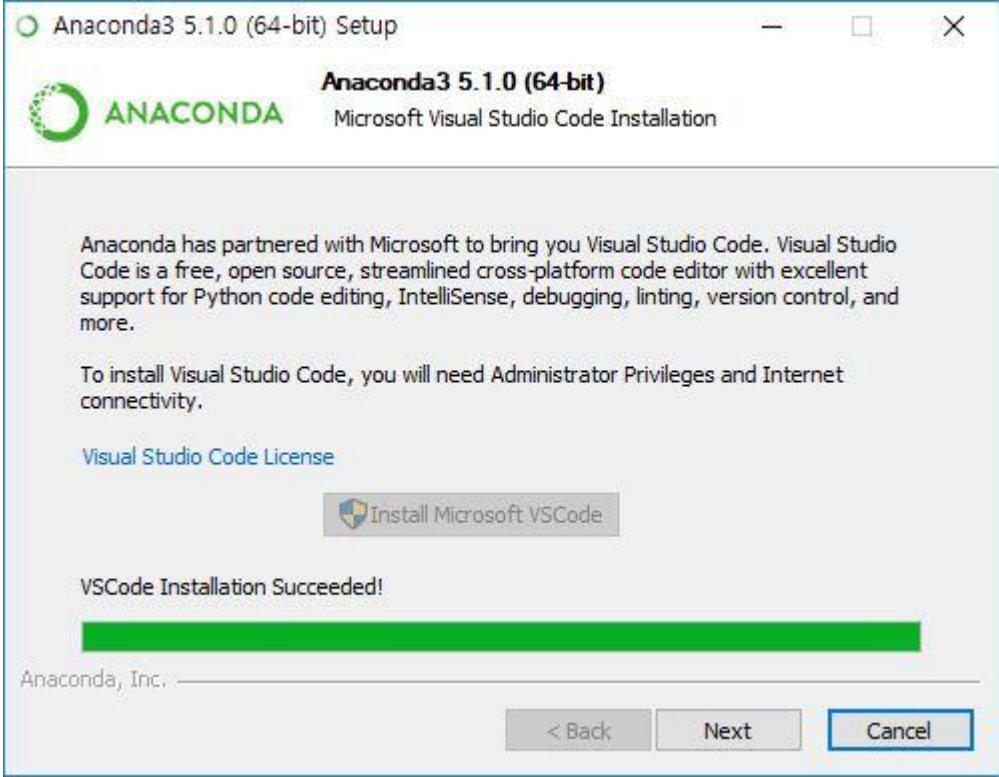
- 다른 곳을 선택해도 되지만, 수업 자료와의 통일감을 위해 기본 위치로 하자

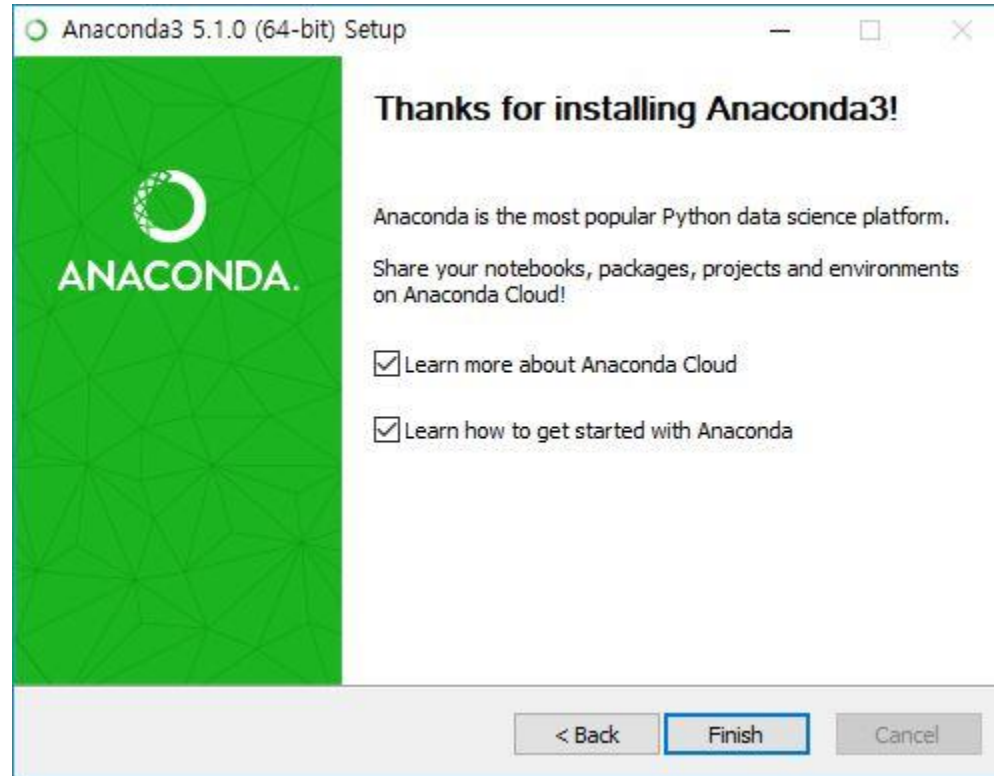


- 경로 설정... 다른 **Python**이 설치되어 있지 않다면 그냥 둘 다 체크하자

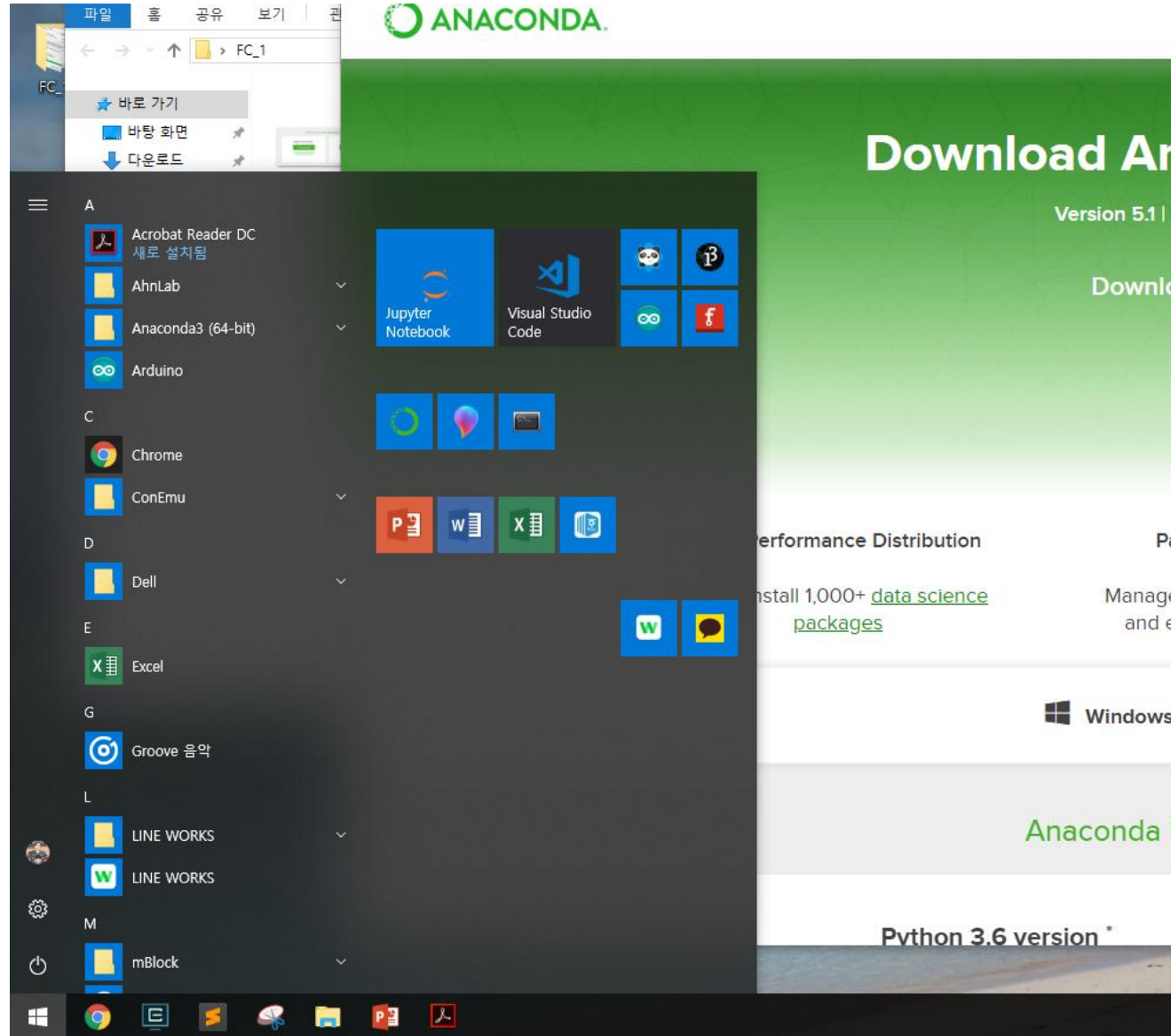


- 앳. **VSCode**를 설치하는 화면... 안 깔려 있다면 이 아이도 설치하자

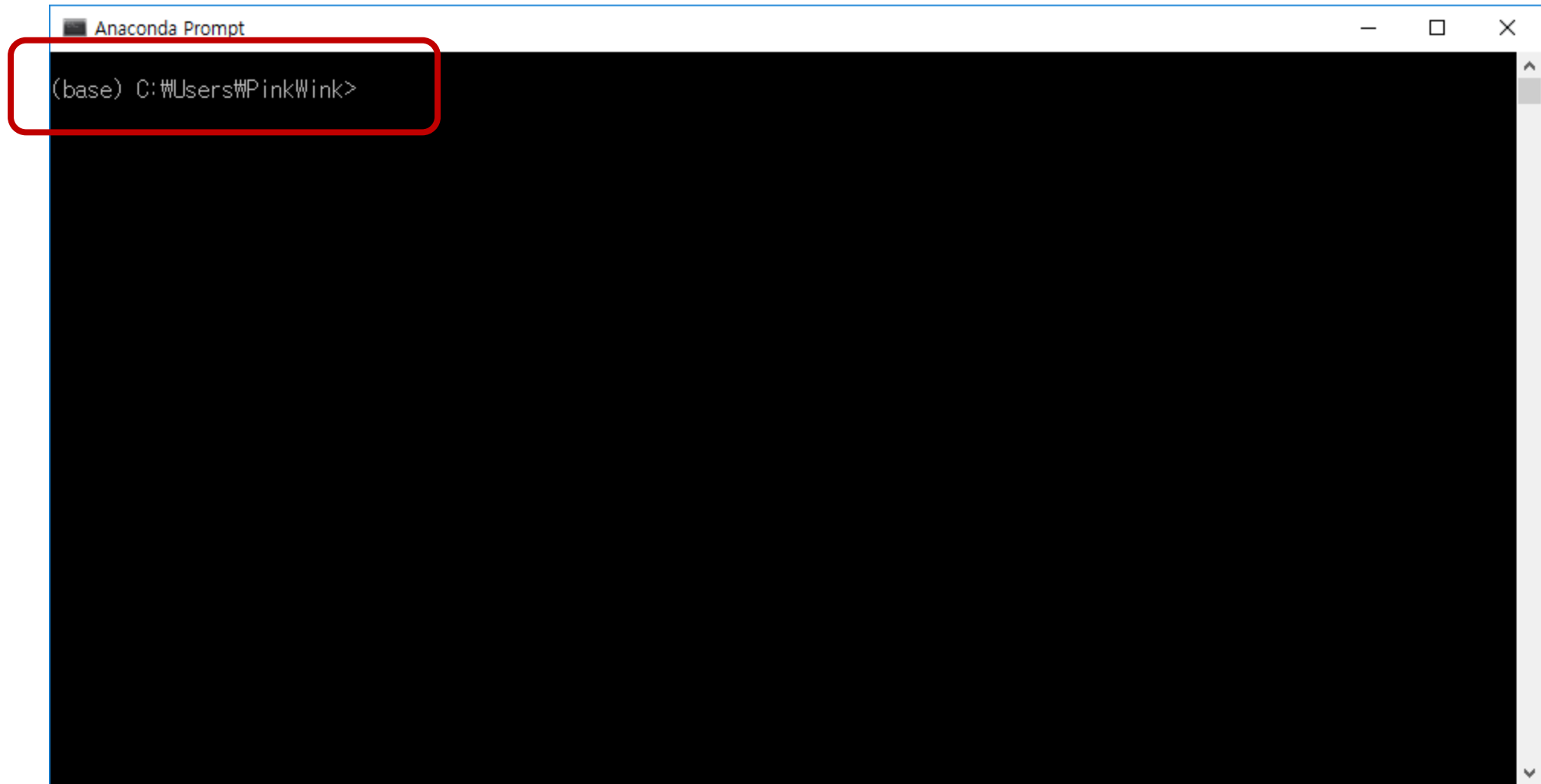




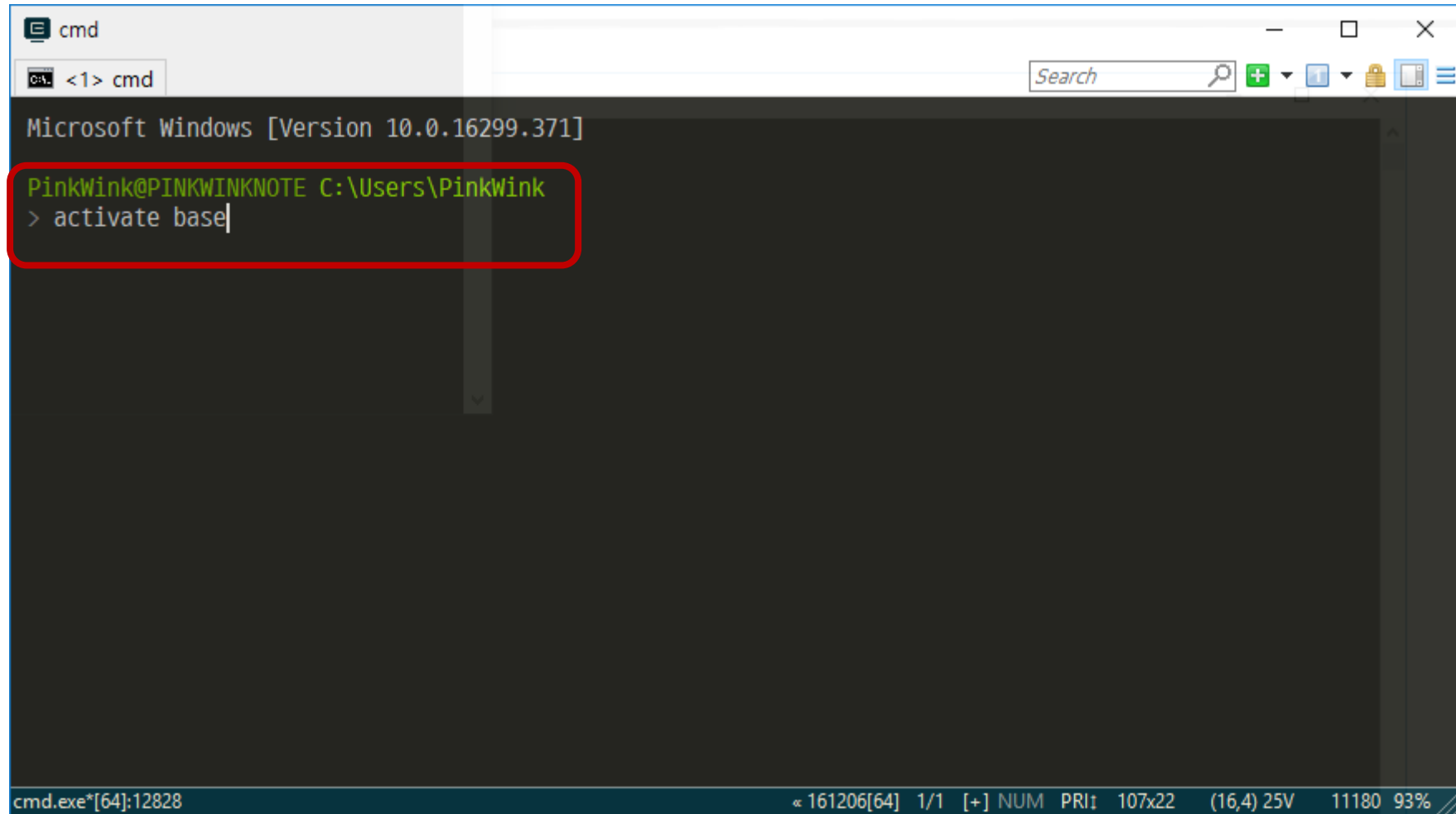
- 설치 끝 ~~~~



- Jupyter Notebook
- Anacoda Prompt
- VSCode



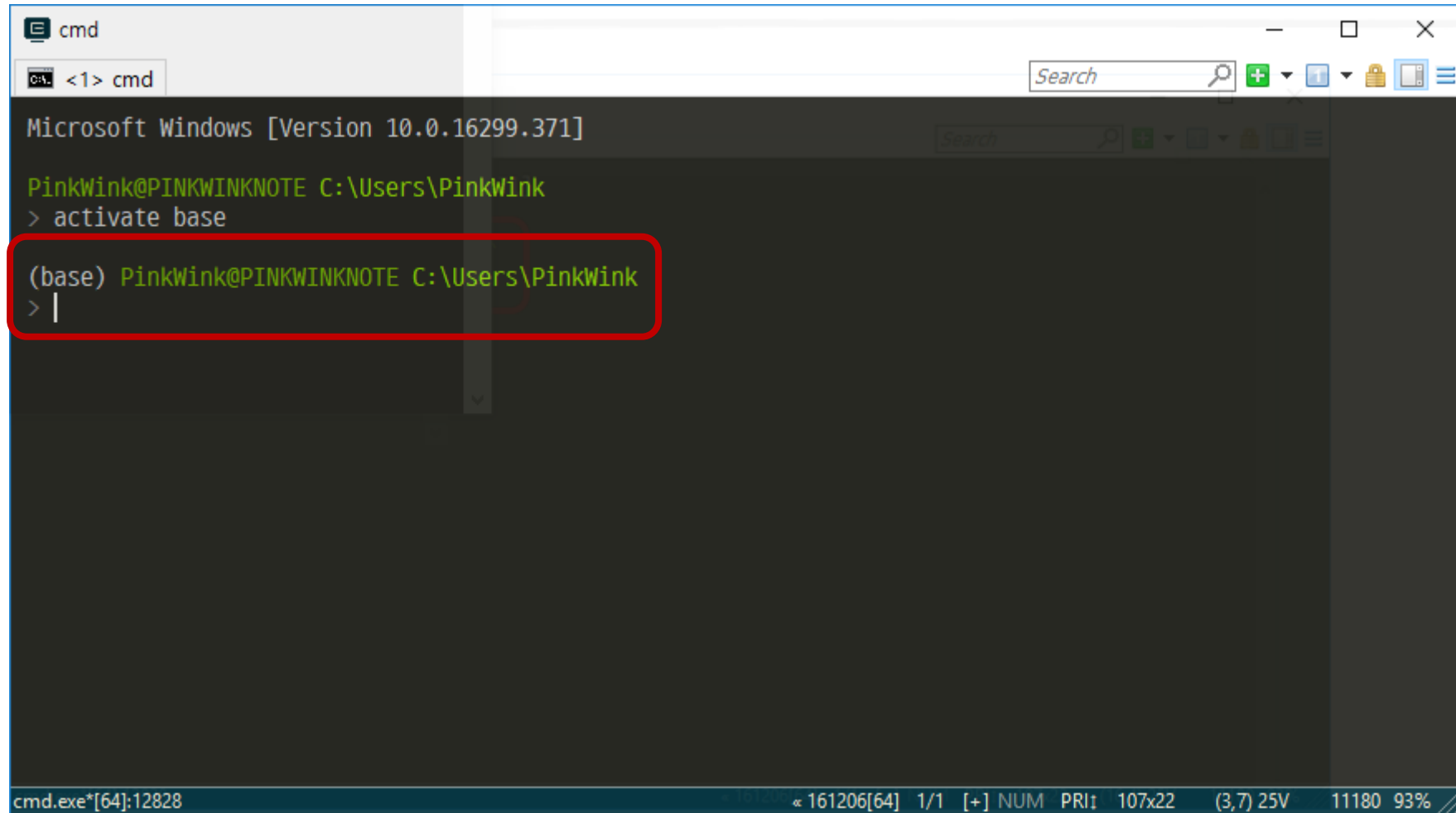
- Anaconda Prompt를 실행한 화면



The image shows a Windows Command Prompt window titled 'cmd'. The window has a search bar at the top right and a taskbar at the bottom. The command prompt displays the text 'Microsoft Windows [Version 10.0.16299.371]' followed by the prompt 'PinkWink@PINKWINKNOTE C:\Users\PinkWink'. The command '> activate base' is entered and highlighted with a red rectangle. The status bar at the bottom shows 'cmd.exe*[64]:12828' and system information.

```
cmd
<1> cmd
Microsoft Windows [Version 10.0.16299.371]
PinkWink@PINKWINKNOTE C:\Users\PinkWink
> activate base
```

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

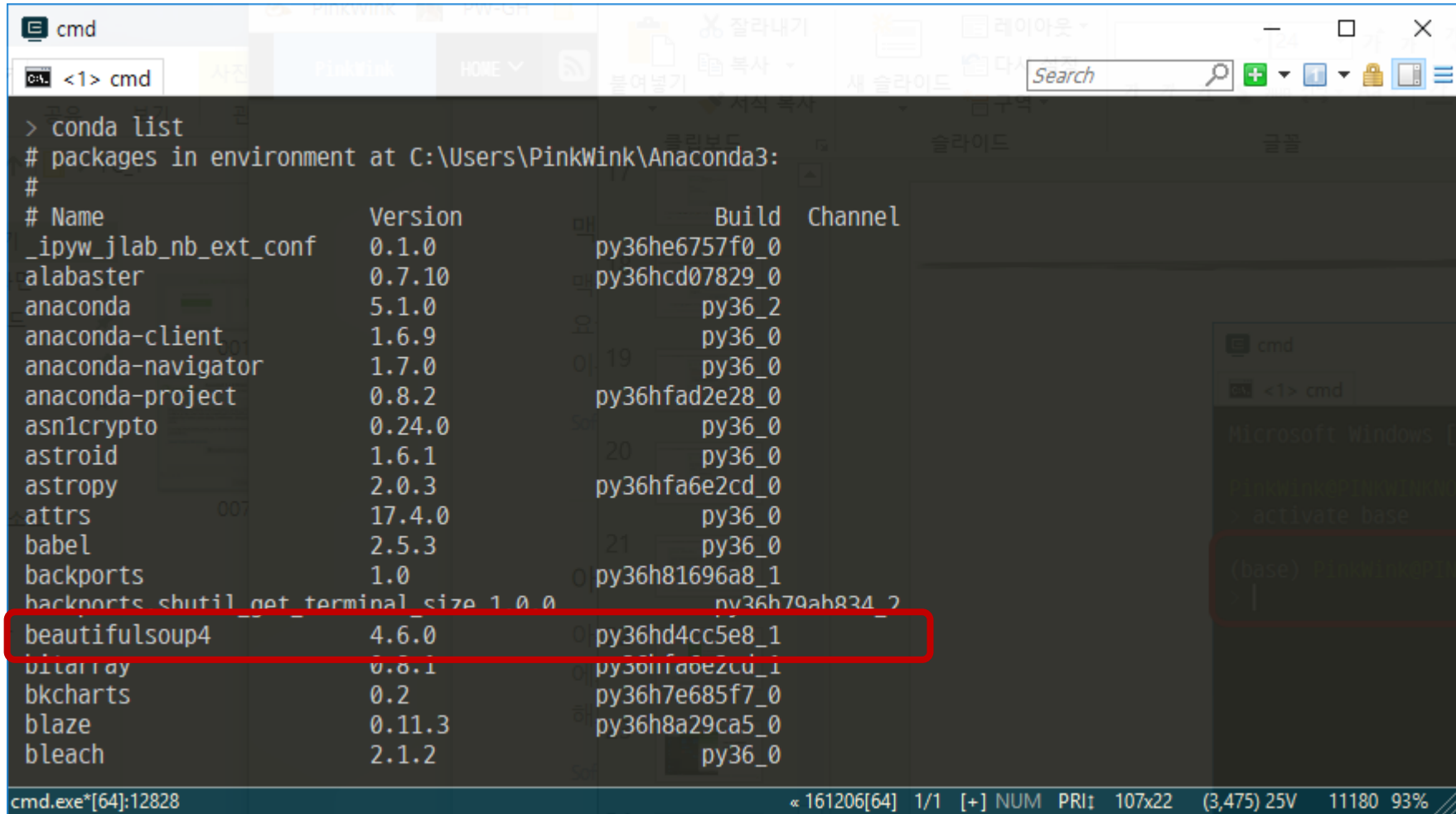


The screenshot shows a Windows Command Prompt window titled 'cmd'. The prompt is 'C:\>'. The user has entered 'activate base'. The prompt has changed to '(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink>'. The text '(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink>' is highlighted with a red rectangle. The status bar at the bottom shows 'cmd.exe*[64]:12828' and system information.

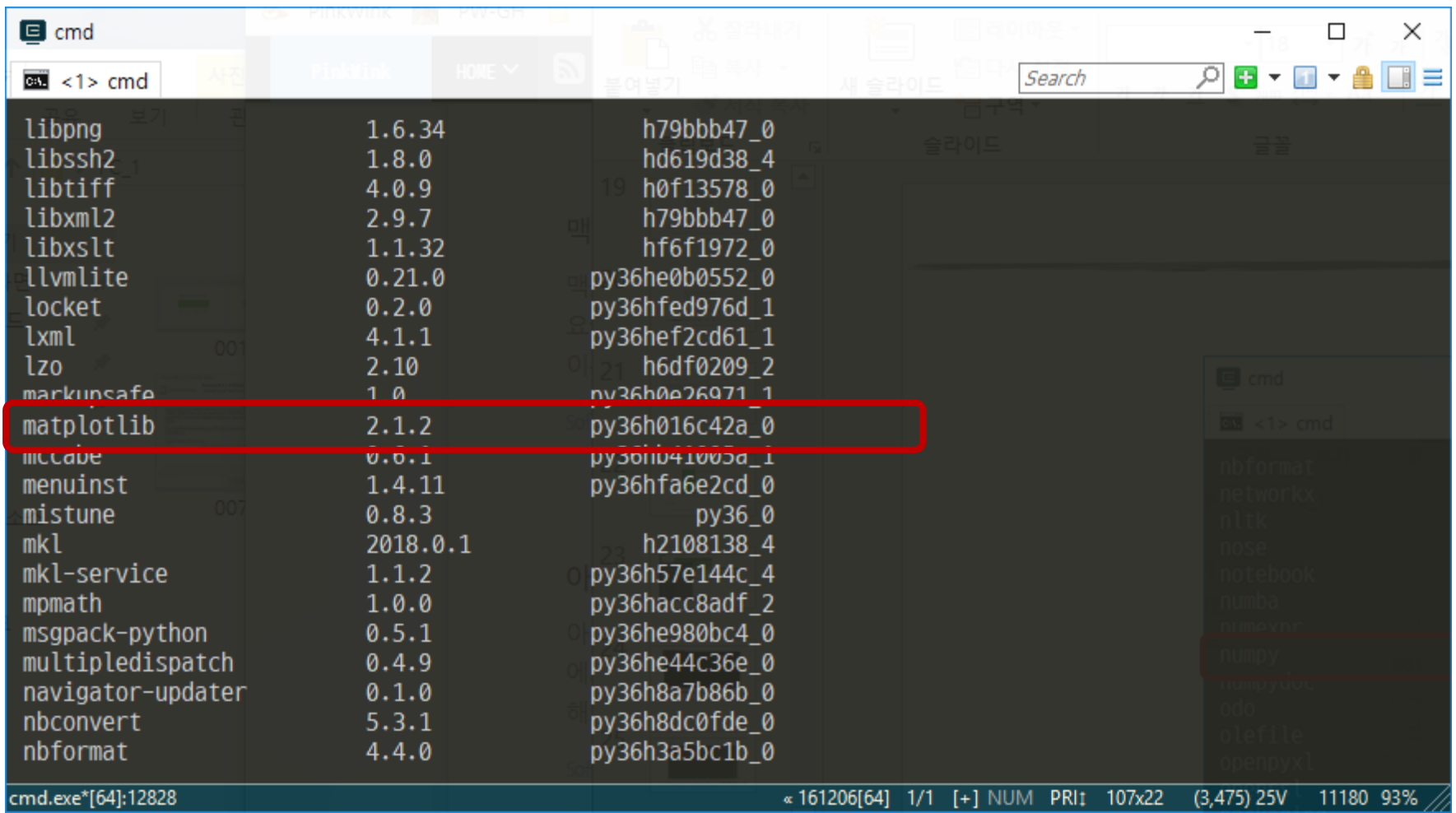
```
cmd
C:\> activate base
(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink>
```

- 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        py36h79ab834_2
beautifulsoup4                        4.6.0        py36hd4cc5e8_1
bitarray                              0.8.1        py36hfa6e2cd_1
bkcharts                              0.2          py36h7e685f7_0
blaze                                 0.11.3       py36h8a29ca5_0
bleach                                2.1.2        py36_0
```



```
cmd
<1> cmd
libpng 1.6.34 h79bbb47_0
libssh2 1.8.0 hd619d38_4
libtiff 4.0.9 h0f13578_0
libxml2 2.9.7 h79bbb47_0
libxslt 1.1.32 hf6f1972_0
llvmlite 0.21.0 py36he0b0552_0
loket 0.2.0 py36hfd976d_1
lxml 4.1.1 py36hef2cd61_1
lzo 2.10 h6df0209_2
markupsafe 1.0 py36h0e26071_1
matplotlib 2.1.2 py36h016c42a_0
mccabe 0.6.1 py36h041005a_1
menuinst 1.4.11 py36hfa6e2cd_0
mistune 0.8.3 py36_0
mkl 2018.0.1 h2108138_4
mkl-service 1.1.2 py36h57e144c_4
mpmath 1.0.0 py36hacc8adf_2
msgpack-python 0.5.1 py36he980bc4_0
multipledispatch 0.4.9 py36he44c36e_0
navigator-updater 0.1.0 py36h8a7b86b_0
nbconvert 5.3.1 py36h8dc0fde_0
nbformat 4.4.0 py36h3a5bc1b_0

cmd.exe*[64]:12828
< 161206[64] 1/1 [+] NUM PRI: 107x22 (3,475) 25V 11180 93%
```

Anaconda 설치

```
cmd
<1> cmd
nbformat 4.4.0 py36h3a5bc1b_0
networkx 2.1 py36_0
nltk 3.2.5 py36h76d52bb_0
nose 1.3.7 py36h1c3779e_2
notebook 5.4.0 py36_0
numba 0.36.2 np114py36h12cb543_0
numexpr 2.6.4 py36h30784b8_0
numpy 1.14.0 py36h4a99626_1
numpydoc 0.7.0 py36ha23429e_0
odo 0.5.1 py36h7560279_0
olefile 0.45.1 py36_0
openpyxl 2.4.10 py36_0
openssl 1.0.2n h74b6da3_0
packaging 16.8 py36ha0086f6_1
pandas 0.22.0 py36h6538335_0
pandoc 1.19.2.1 nb2460c/_1
pandocfilters 1.4.2 py36h3ef6317_1
parso 0.1.1 py36hae3edee_0
partd 0.3.8 py36hc8e763b_0
path.py 10.5 py36h2b94a8f_0
pathlib2 2.3.0 py36h7bfb78b_0
patsy 0.5.0 py36_0
cmd.exe*[64]:12828
< 161206[64] 1/1 [+] NUM PRI: 107x22 (3,475) 25V 11180 93%
```

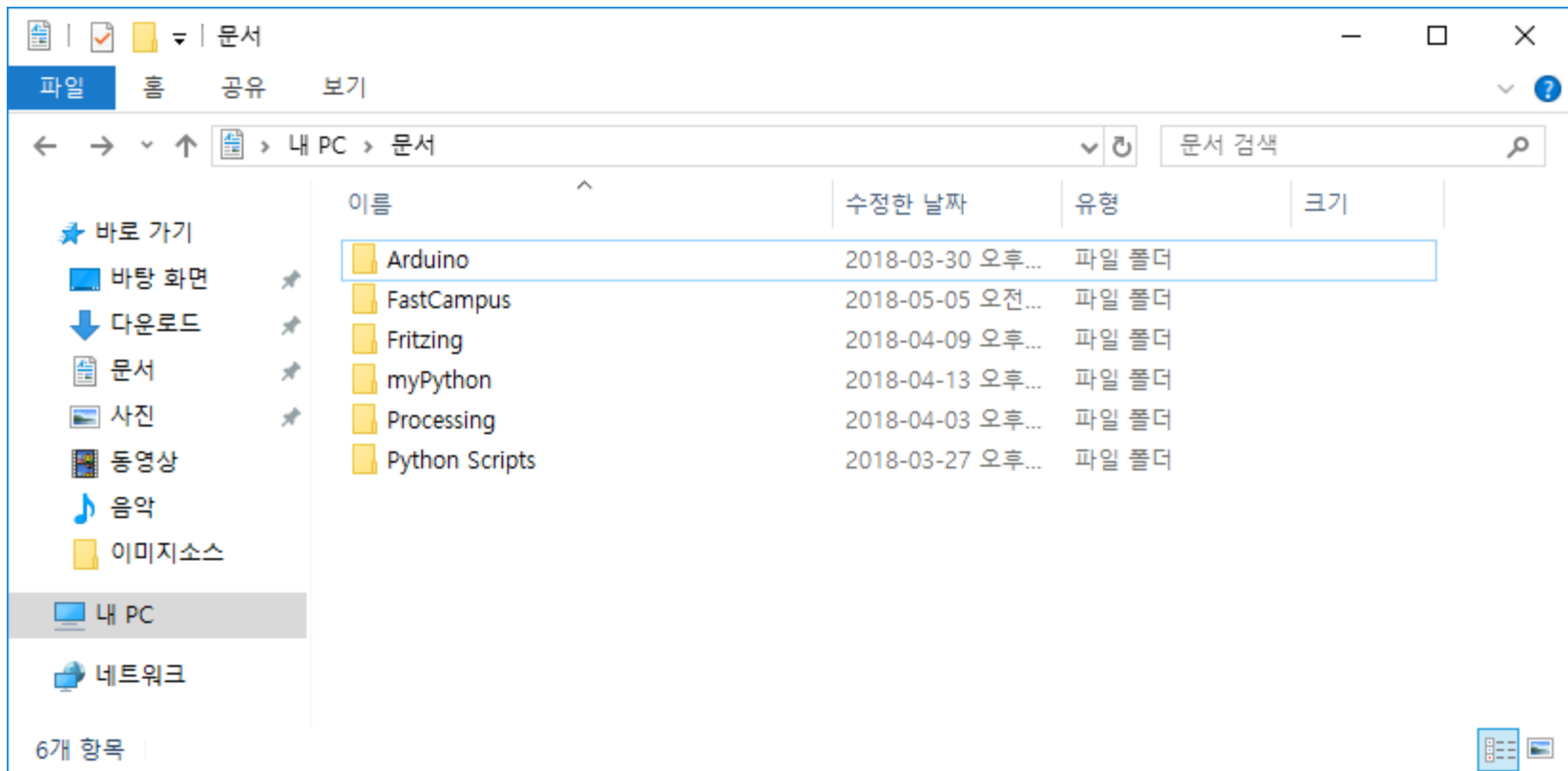

The screenshot shows a Windows command prompt window titled 'cmd'. The window displays a list of installed Anaconda packages in a three-column format. The 'seaborn' package is highlighted with a red box. The status bar at the bottom shows 'cmd.exe*[64]:12828' and system information.

Package Name	Version	Hash
pyzmq	16.0.3	py36he714bf5_0
qt	5.6.2	vc14h6f8c307_12 [vc14]
qtawesome	0.4.4	py36h5aa48f6_0
qtconsole	4.3.1	py36h99a29a9_0
qtpy	1.3.1	py36hb8717c5_0
requests	2.18.4	py36h4371aae_1
rope	0.10.7	py36had63a69_0
ruamel_yaml	0.15.35	py36hfa6e2cd_1
scikit-image	0.13.1	py36hfa6e2cd_1
scikit-learn	0.19.1	py36h53aea1b_0
scipy	1.0.0	py36h1260518_0
seaborn	0.8.1	py36h9b69545_0
send2trash	1.4.2	py36_0
setuptools	38.4.0	py36_0
simplegeneric	0.8.1	py36_2
singledispatch	3.4.0.3	py36h17d0c80_0
sip	4.18.1	py36h9c25514_2
six	1.11.0	py36h4db2310_1
snowballstemmer	1.2.1	py36h763602f_0
sortedcollections	0.5.3	py36hbefa0ab_0
sortedcontainers	1.5.9	py36_0
sphinx	1.6.6	py36_0

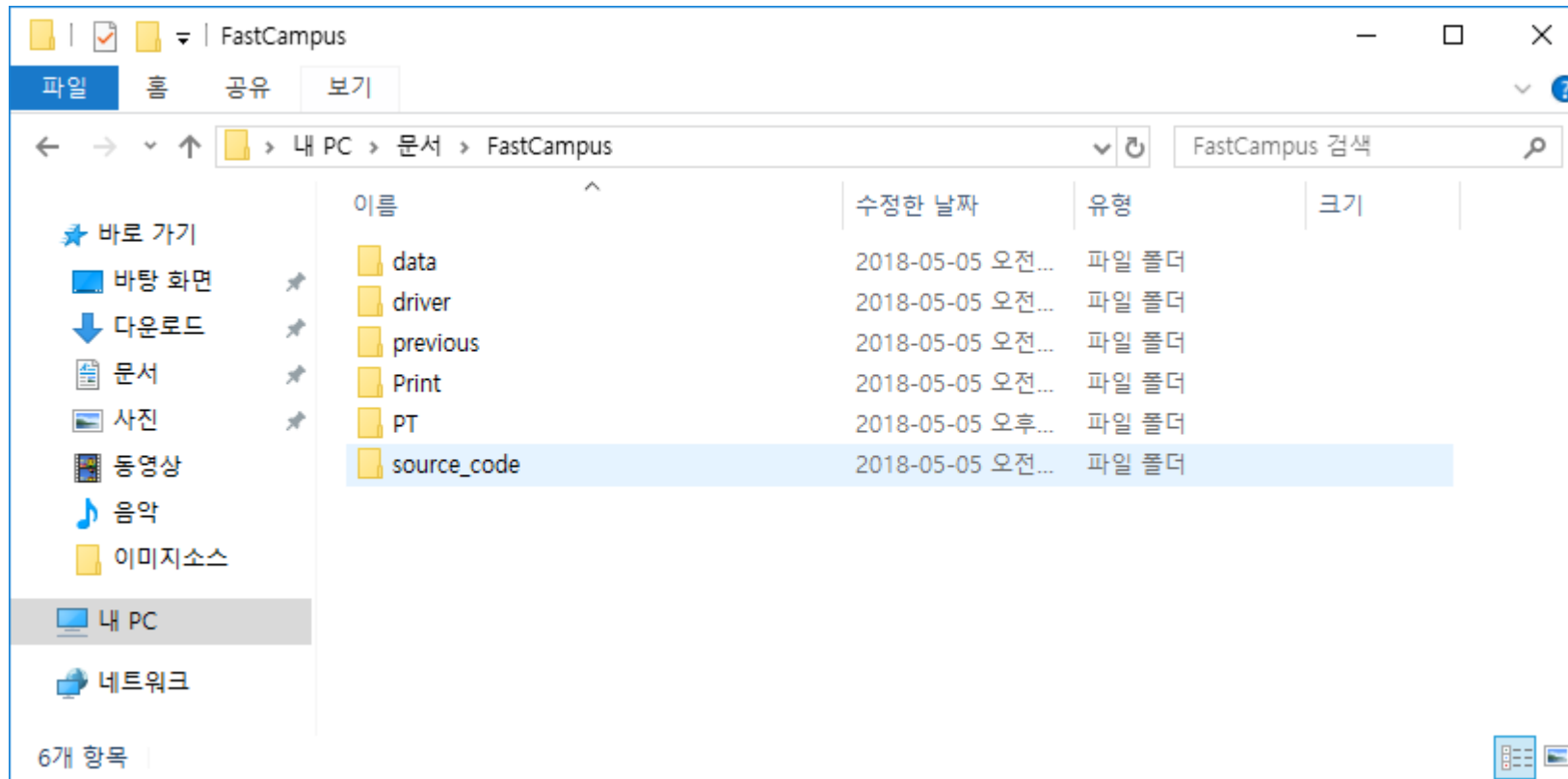
- **Python**은 모듈들의 버전 변화가 많은 언어
- 특히 인기있는 모듈들은 극심한 편
- 작업 버전업에도 명령어 자체가 변하는 경우도 많음

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

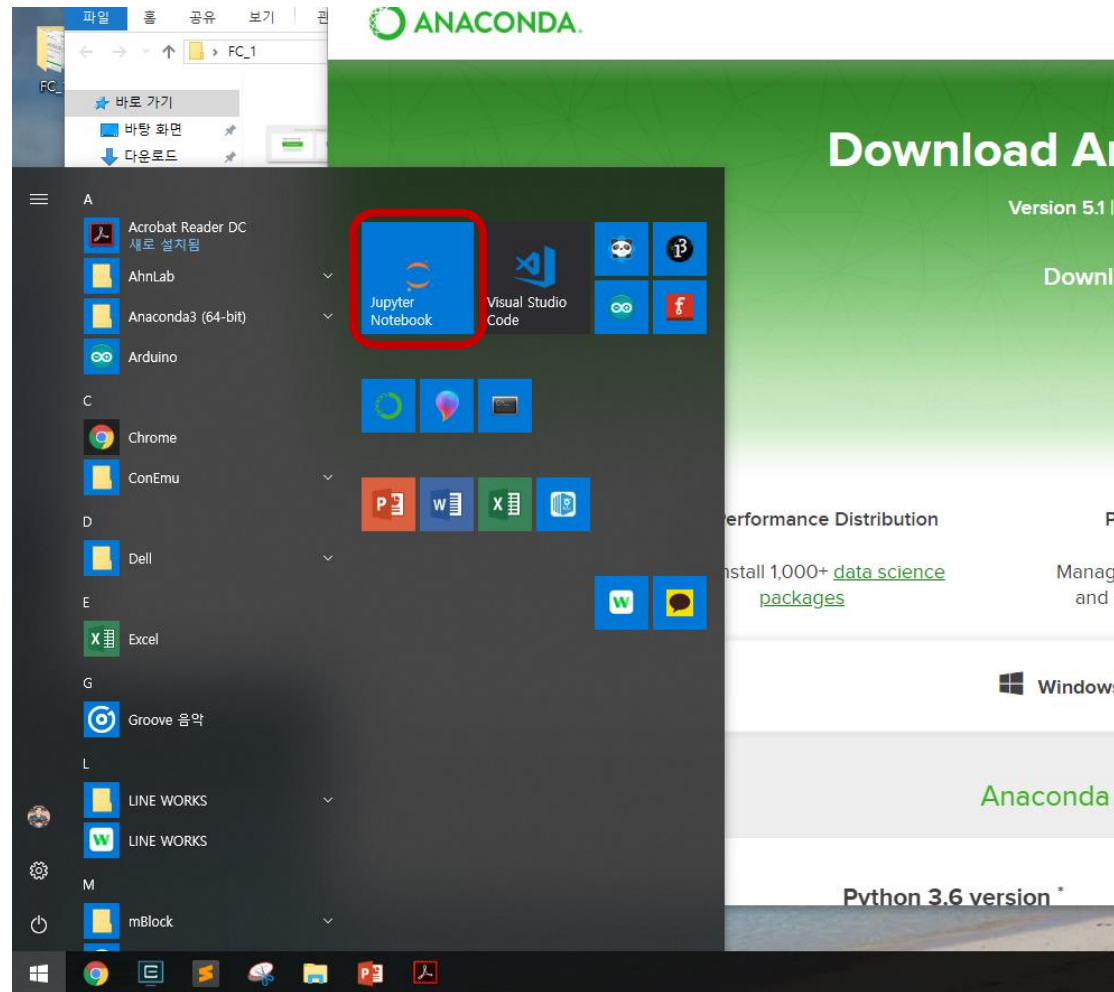


- 내문서 밑에 **FastCampus** 폴더를 만들고

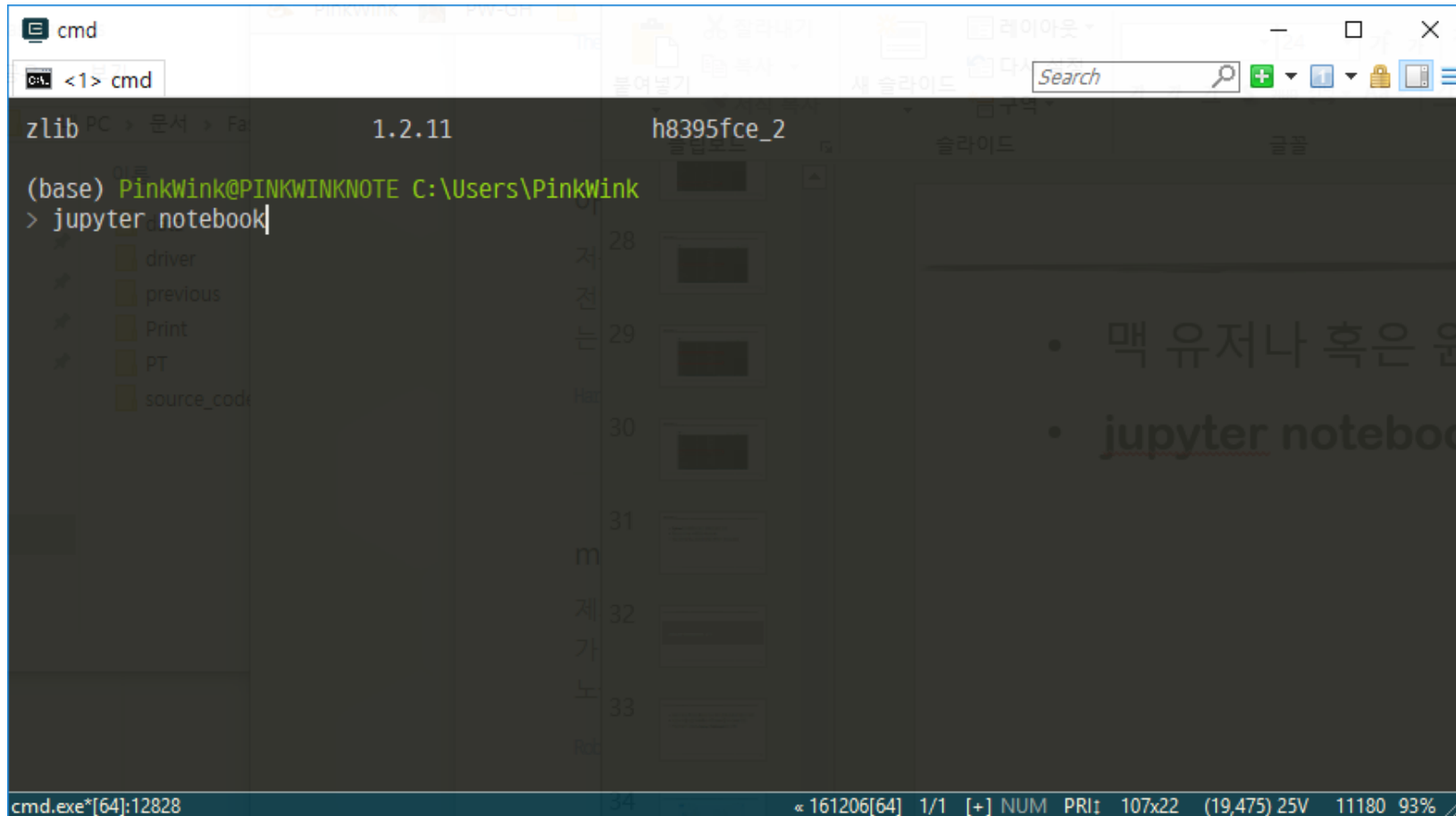


- 그 안에, **data**, **driver**, **source_code** 폴더를 만들어 둬
- 앞으로 수업 자료 및 실습은 **source_code** 폴더에서 진행하는 것으로 약속

- Windows User의 경우는 시작 버튼을 누르고 Jupyter Notebook 선택



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



The screenshot shows a Windows Command Prompt window titled 'cmd'. The command prompt shows the user is in the 'C:\Users\PinkWink' directory. The user has entered the command 'jupyter notebook' and the prompt is waiting for a response. The background of the Command Prompt window is dark, and the text is light green. The window title bar shows 'cmd' and standard Windows window controls. The taskbar at the bottom shows the Command Prompt icon and the text 'cmd.exe*[64]:12828'.

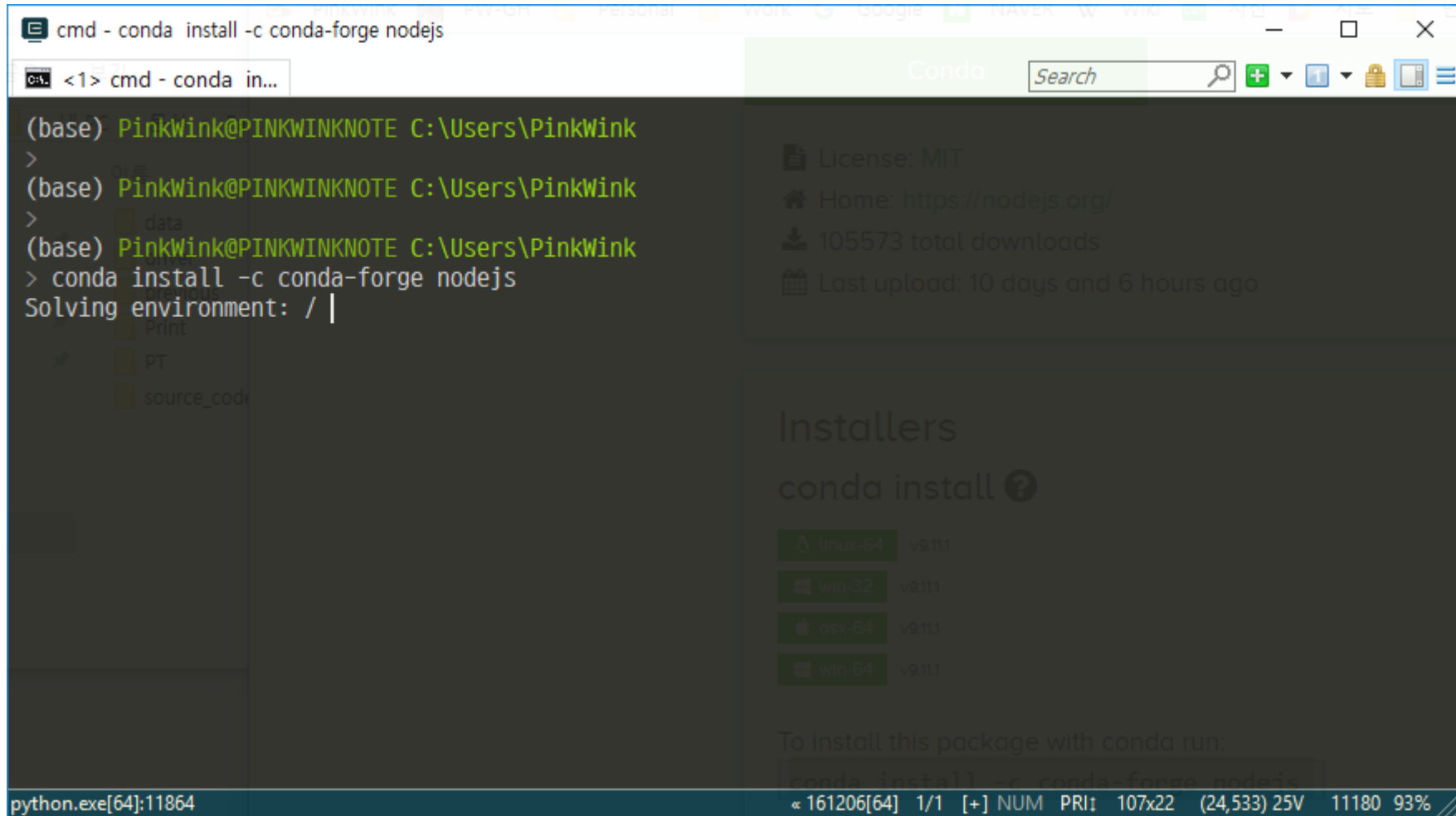
```
cmd
C:\Users\PinkWink
(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
> jupyter notebook
```

```
cmd - jupyter notebook

(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_proc
    cmd[0] = which(cmd[0], kwargs.get('env'))
  File "C:\Users\PinkWink\Anaconda3\lib\site-packages\jupyterlab\jupyterlab.py", line 59, in which
    raise ValueError(msg)
ValueError: Please install nodejs 5+ and npm before continuing installation. nodejs may be installed us
ing conda or directly from the nodejs website.

python.exe[64]:5848
```

- 만약 이런 메시지가 나타난다면



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

```
(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: / |
```

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

Installers

conda install ?

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

- `conda install -c conda-forge nodejs`

```
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      size
-----
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  α 161206[64] 1/1 [+] NUM PRI: 107x22 (19,564) 25V 11180 93%
```

- nodejs 모듈을 설치

```
cmd
C:\> conda install -c conda-forge nodejs

The following packages will be downloaded:

package | build | size | channel
-----|-----|-----|-----
nodejs-9.11.1 | 0 | 12.6 MB | conda-forge

The following NEW packages will be INSTALLED:

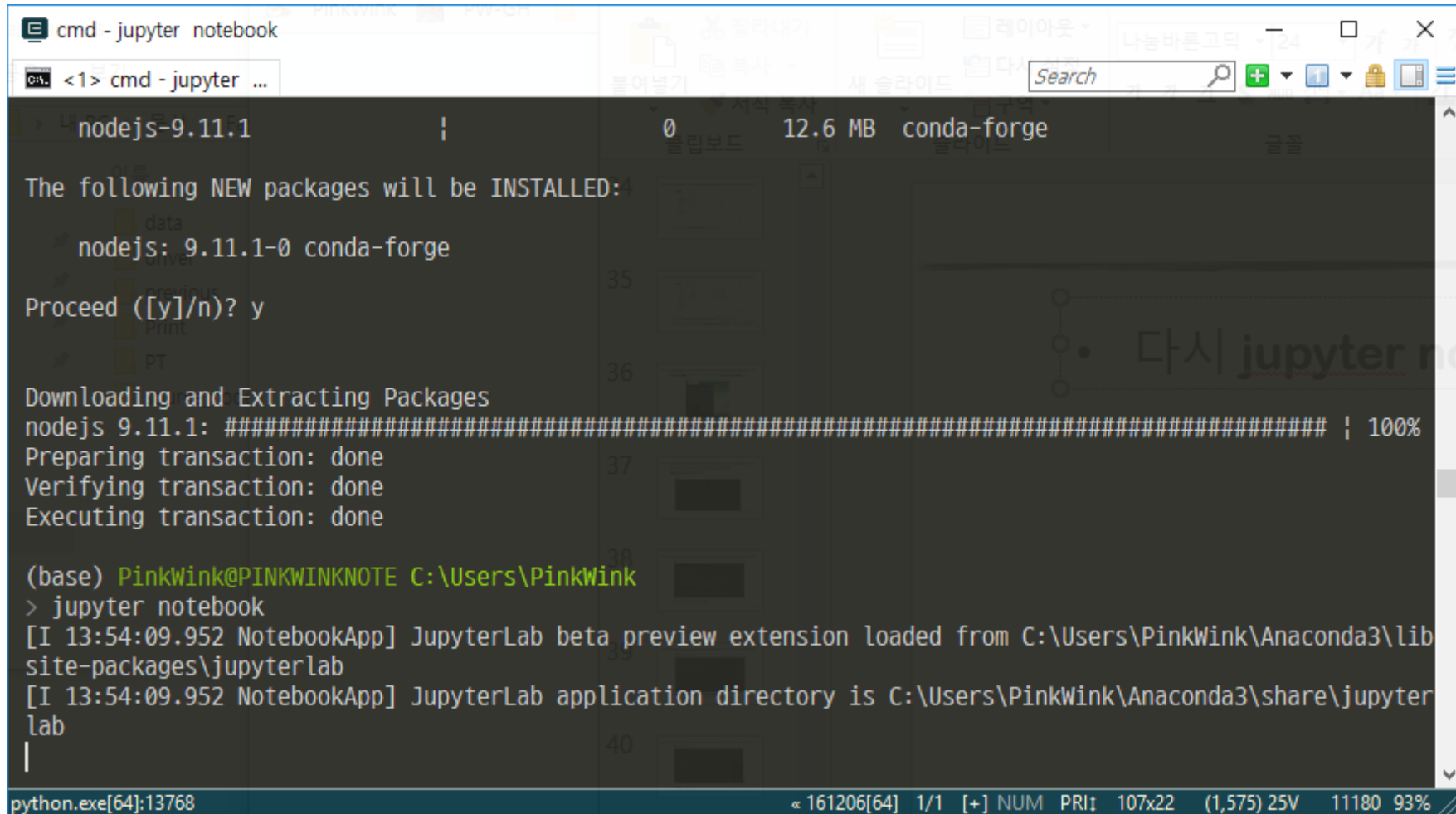
nodejs: 9.11.1-0 conda-forge

Proceed ([y]/n)? y

Downloading and Extracting Packages
nodejs 9.11.1: ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
> |
```

- 다시 jupyter notebook 실행



```
cmd - jupyter notebook
<1> cmd - jupyter ...

nodejs-9.11.1 | 0 12.6 MB conda-forge

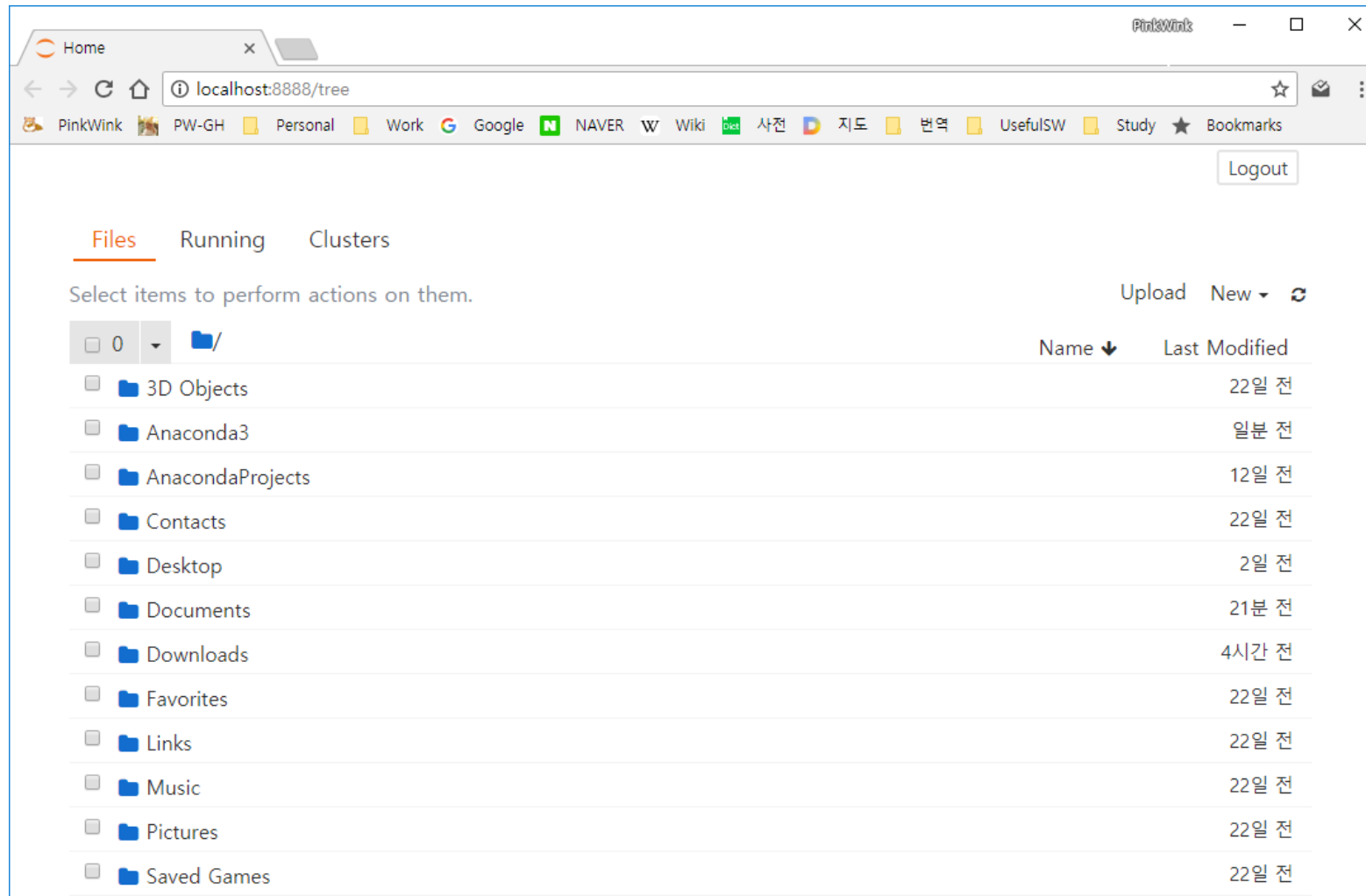
The following NEW packages will be INSTALLED:
  nodejs: 9.11.1-0 conda-forge

Proceed ([y]/n)? y

Downloading and Extracting Packages
nodejs 9.11.1: ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

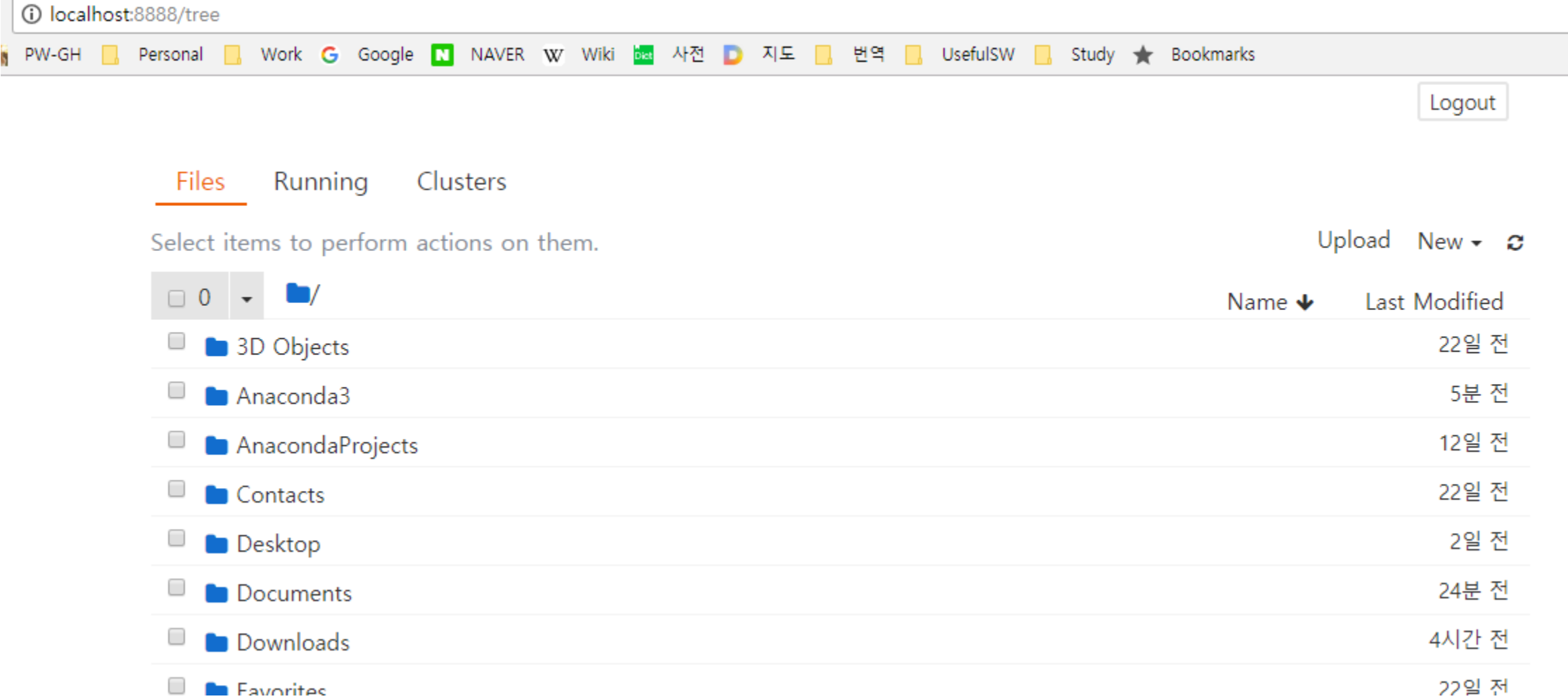
(base) PinkWink@PINKWINKNOTE C:\Users\PinkWink
> jupyter notebook
[I 13:54:09.952 NotebookApp] JupyterLab beta preview extension loaded from C:\Users\PinkWink\Anaconda3\lib
site-packages\jupyterlab
[I 13:54:09.952 NotebookApp] JupyterLab application directory is C:\Users\PinkWink\Anaconda3\share\jupyter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

python.exe[64]:13768  α 161206[64] 1/1 [+] NUM PRI: 107x22 (1,575) 25V 11180 93%
```

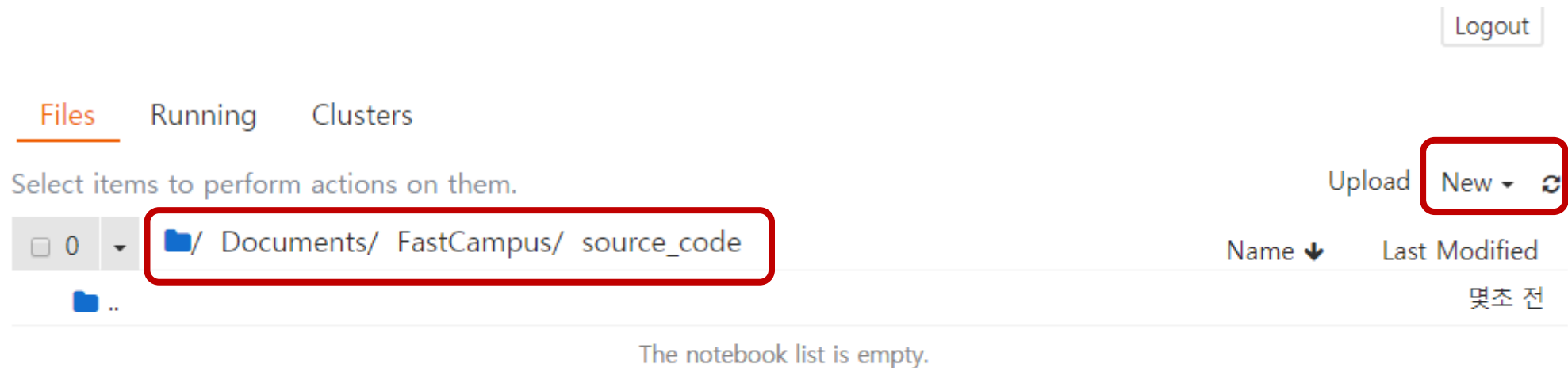


- 웹 브라우저에 `localhost:8888/tree`라는 주소에 접속한 것을 알 수 있음

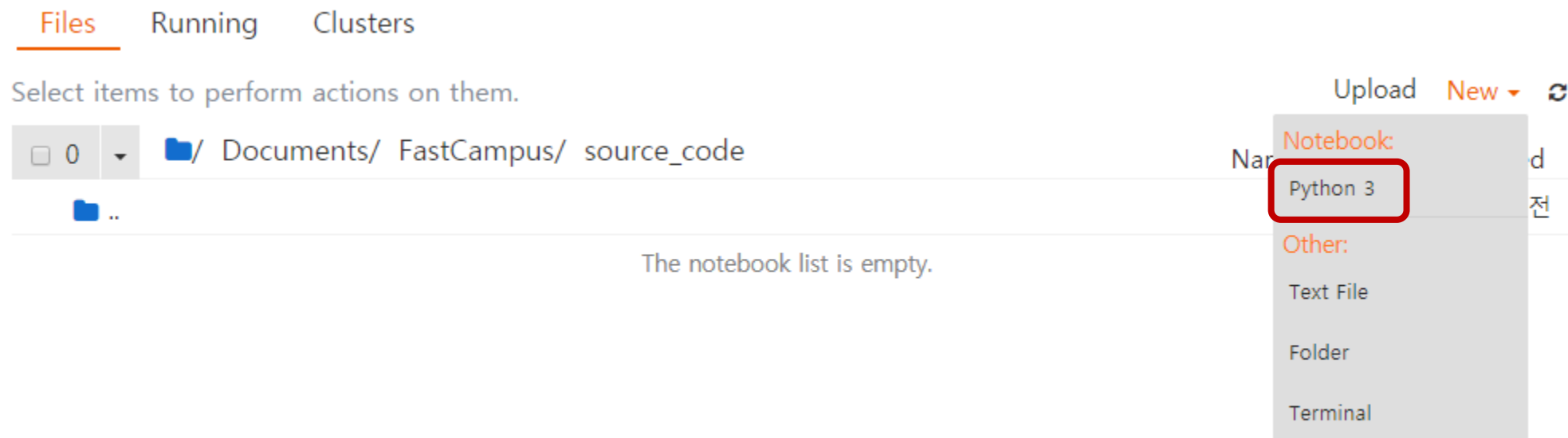
- 여기서 잠깐...
- 우리 수업은 기본 웹 브라우저로 구글 크롬을 사용
 - 크롬을 사용하지 않으면 수업과 **100%** 호환되지 않을 수 있음
- 추가로 **sublime text**를 사용하는 것도 추천함



- 아무튼 이 화면에서 우리가 만든 경로로 찾아가기

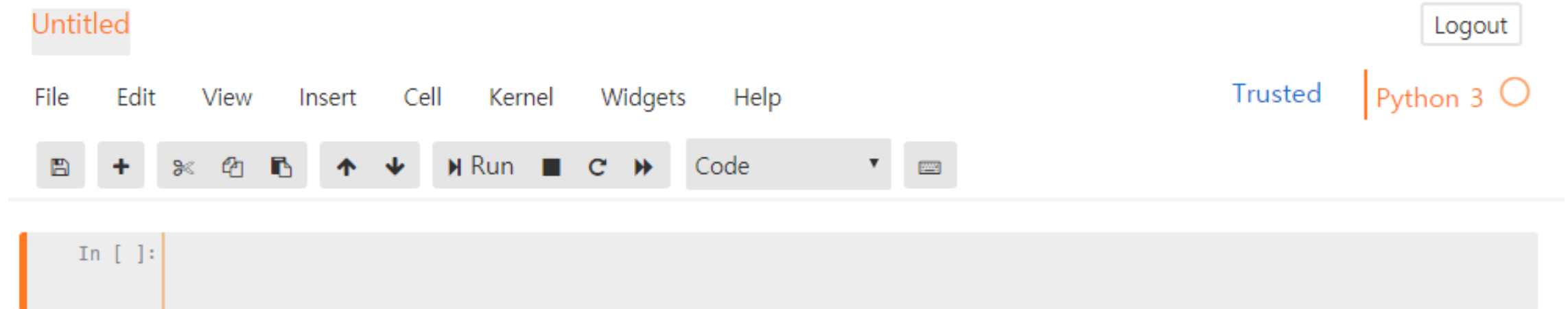


- 학습할 코드가 위치할 경로를 본능적으로 확인하고
- **New** 버튼 누르기



- 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

Icons for file operations and execution

In []:

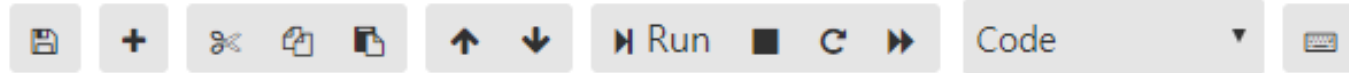
00. Hello World

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

^ + x

- 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__() (if defined)
or repr(object).
encoding defaults to sys.getdefaultencoding().
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 입력하지 말고 , %cd sou 정도에서 **Tab** 키를 눌러보자
- 앞으론, 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 ..
```

```
s1 =
```

- **%history** : 입력한 명령어 **history**를 보여줌

```
%cd source_code/
```

```
%pwd
```

```
%whos
```

```
%history
```

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

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

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

- **%timeit** : 실행되는 시간을 표시

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

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

```
In [21]: %pdb
```

```
Automatic pdb calling has been turned ON
```

- **%pdb** : 디버그 모드 사용

```
In [*]: for _ in xrange(10): True
```

```
-----  
NameError                                Traceback (most recent call last)
```

```
<ipython-input-23-4c554ab4a6f8> in <module>()  
----> 1 for _ in xrange(10): True
```

```
NameError: name 'xrange' is not defined
```

```
ipdb>
```

```
> <ipython-input-23-4c554ab4a6f8>(1)<module>()  
----> 1 for _ in xrange(10): True
```

- `%pdb` : 디버그 모드가 사용된 예

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

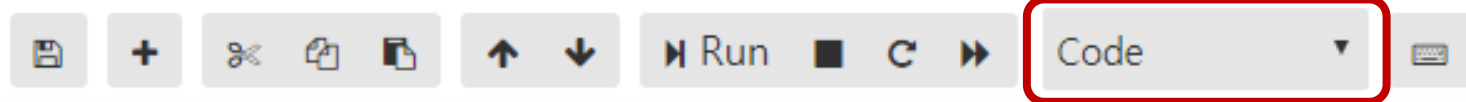
00. Hello World

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^[1]. 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

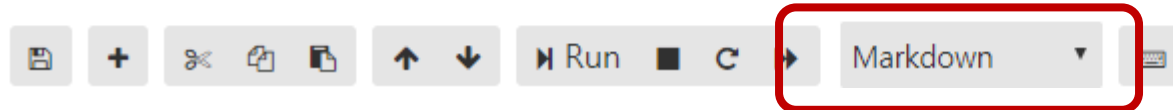


출처 : 위키백과

00. Hello World

Logout

File Edit View Insert Cell Kernel Widgets Help



Trusted

Python 3



- 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 4ms이므로 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=',')

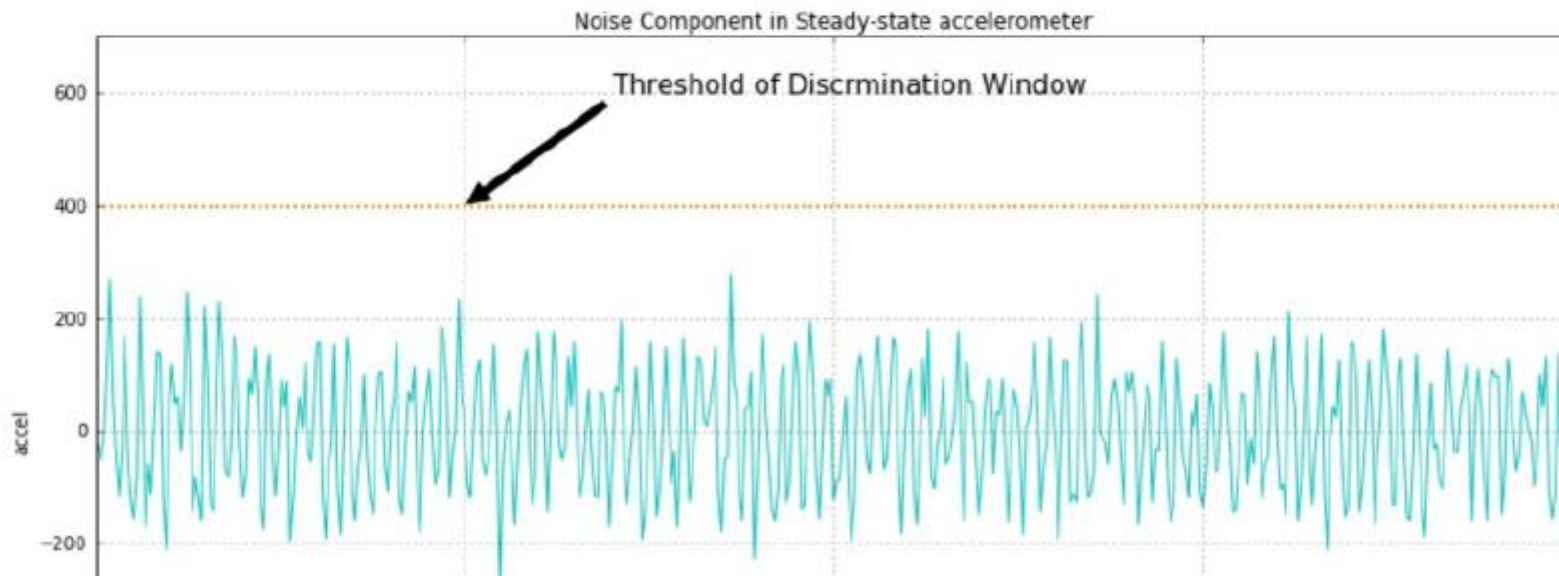
acc_rawdata = rawData[:,0]
acc_rawdata = acc_rawdata.astype(float)

Ts = 0.004
Fs = 1/Ts

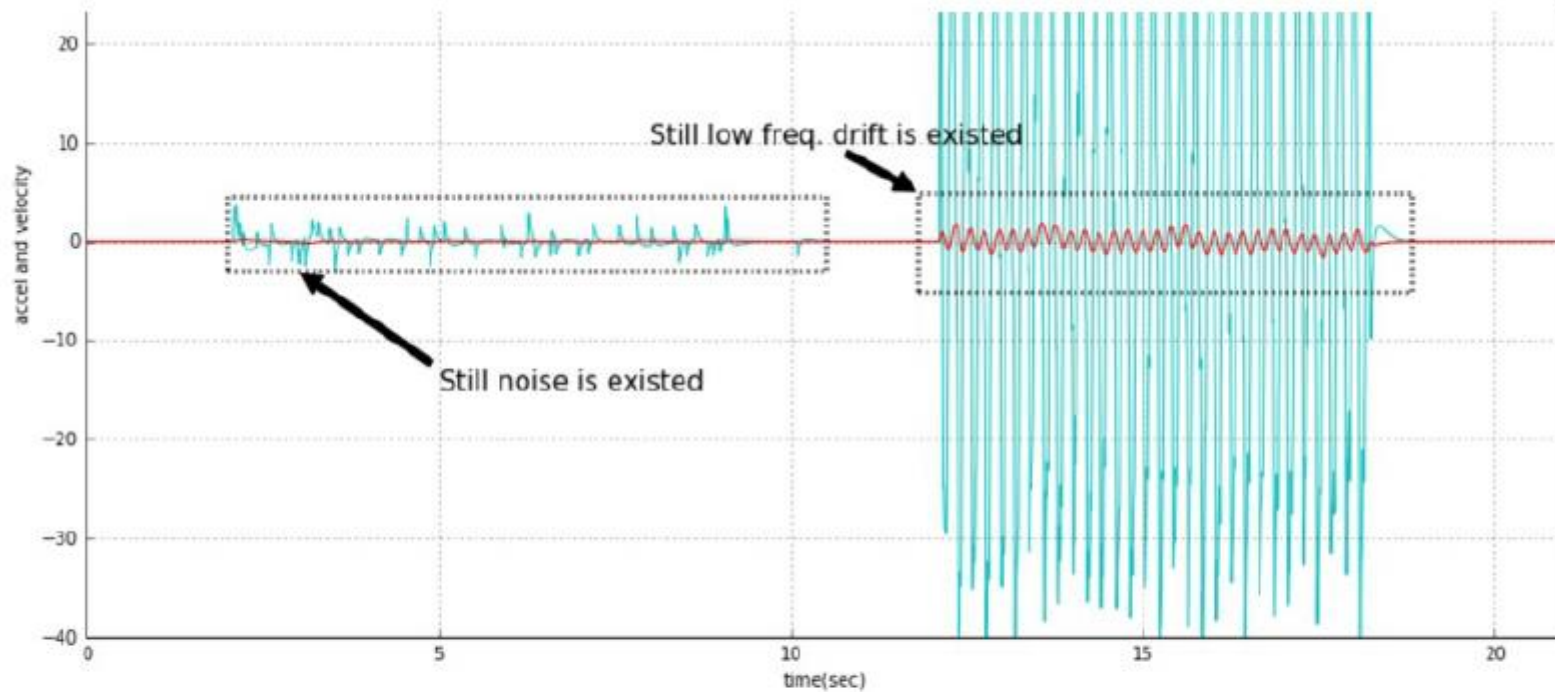
testLength = float(len(acc_rawdata))
t = np.arange(0, testLength*Ts, Ts)
```

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, lw=3, linestyle='dotted', color='orange')
plt.hlines(-400,0,2, lw=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

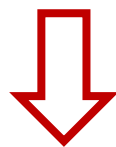


- 위 그래프에서는 두 가지 문제가 관찰되는데, 하나는 첫 부분의 속도입장에서 노이즈처럼 보이는 구간이다.
- 이 부분은 실험 전 실제로 움직인 부분인지 혹은 노이즈인지 데이터만 봐서는 판단하기 어려워 추후 다시 실험할때 확인하도록 한다
- 두 번째는 저주파 성분의 스윙이 아직도 관찰된다는 것이다
- 이 경우는 이전의 HPF의 계수를 조정하는 것과, 구한 위치 결과에 다시 HPF를 적용하는 것으로 방법을 나뉘볼 수 있다.
- 본 문서에서는 다시 HPF를 적용하는 것으로 한다

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



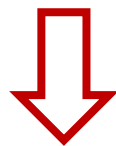
제목

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 **Pandas**는 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 **Pandas**는 **Anaconda**에 기본으로 포함되어 배포되고 있다



인용문

- **Pandas** : 단일 머신에 맞는 데이터셋으로 작업할 때 **Pandas**는 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 **Pandas**는 **Anaconda**에 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}
```

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

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

수식

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

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

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

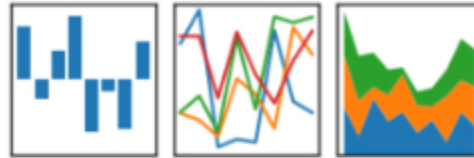
$$x' = x \sin \phi + z \cos \phi$$

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

- 달리 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"></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

검색결과 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 - 집계 현황(2010년). 매체유형 : 기타 / 텍스트 파일, 링크 건수 : 2 건
 형 수 : N/A 확장자 : CSV 다운로드 횟수(바로그기 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

서울특별시 관서별 청소년 범죄 현황 ENGLISH - 공공데이터포털
<https://www.data.go.kr/dataset/3075889/fileData.do>
 서울특별시 지방경찰청 관서별 청소년 5대범죄 현황 매체유형 : 기타 파일, 링크 건수 : 1 건
 형 수 : N/A 확장자 : CSV 다운로드 횟수(바로그기 횟수) : 346 건

- 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