

Python

2. 파이썬 설치

❖ 파이썬의 버전

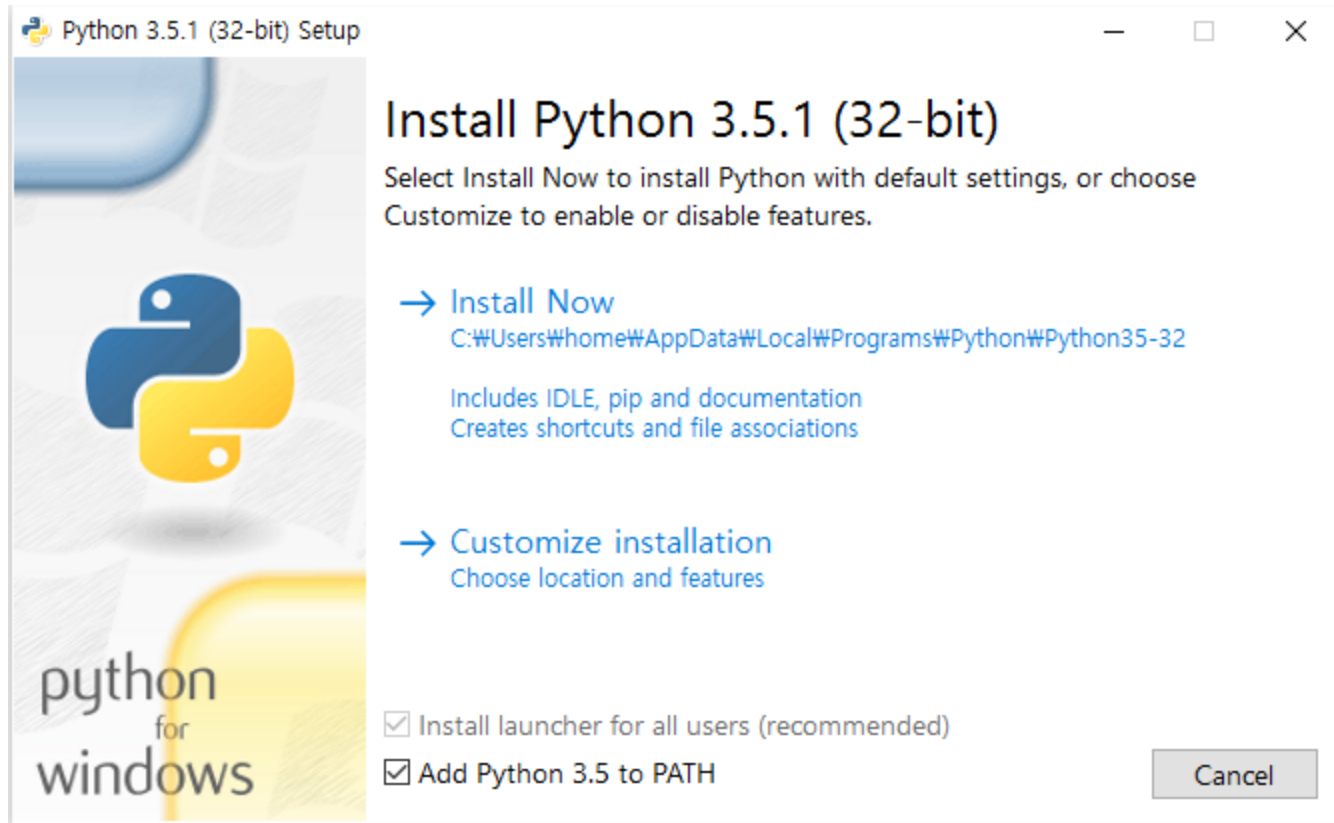
- ✓ 파이썬은 2.x 버전과 3.x 버전이 있는데 파이썬은 2.x버전 과 3.x버전 간의 호환성을 유지하고 있지 않습니다.
- ✓ 변화
 - ◆ print가 함수로 변경
 - ◆ long 형이 없어지고 정수는 int로 통일
 - ◆ int / int 의 결과는 float
 - ◆ 문자열이 string과 unicode로 구분되었는데 string 과 bytes로 구분됩니다.
- ✓ 2to3.py를 이용해서 2.x 버전을 3 버전으로 변환가능(파이썬 설치 디렉토리의 Tools\Scripts 디렉토리에 존재)

❖ 파이썬 다운로드 및 설치

<https://www.python.org/downloads/>

윈도우 용을 설치할 때 파이썬 명령어가 있는 디렉토리를 path에 추가하는 옵션을 체크해야 이클립스에서 파이썬 설정을 자동으로 할 수 있습니다.

2. 파이썬 설치



2. 파이썬 설치

파이썬 배포판 - 아나콘다: www.continuum.io/downloads
<https://repo.continuum.io/archive/.winzip>

- 파이썬 배포판: lpython, Jupyter notebook, Qt Console, Python 포함
- 데이터 분석, 수학, 과학 관련 다양한 라이브러리 포함

Get Superpowers with Anaconda

Anaconda is the leading open data science platform powered by Python. The open source version of Anaconda is a high performance distribution of Python and R and includes over 100 of the most popular **Python**, R and Scala packages for data science. Additionally, you'll have access to over 720 packages that can easily be installed with conda, our renowned package, dependency and environment manager, that is included in Anaconda. Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution. See [the packages included with Anaconda](#) and [the Anaconda changelog](#).

Which version should I download and install?

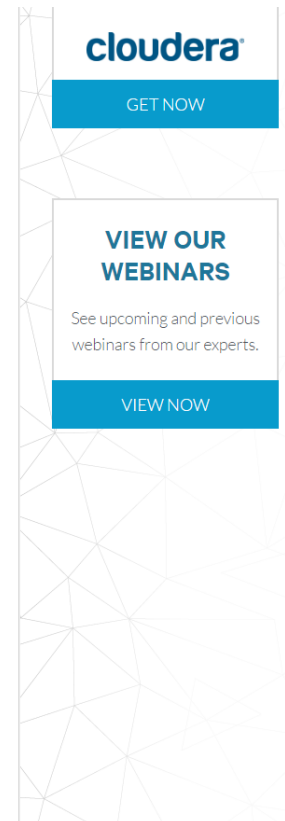
Because Anaconda includes installers for Python 2.7 and 3.5, either is fine. Using either version, you can use Python 3.4 with the conda command. You can create a 3.5 environment with the conda command if you've downloaded 2.7 — and vice versa.

If you don't have time or disk space for the entire distribution, try [Miniconda](#), which contains only conda and Python. Then install just the individual packages you want through the conda command.

Anaconda for Windows

| PYTHON 2.7 | PYTHON 3.5 |
|---|---|
| WINDOWS 64-BIT GRAPHICAL INSTALLER 341M | WINDOWS 64-BIT GRAPHICAL INSTALLER 352M |
| Windows 32-bit Graphical Installer 286M | Windows 32-bit Graphical Installer 293M |

Behind a firewall? Use these [zipped Windows installers](#).



cloudera

GET NOW

VIEW OUR WEBINARS

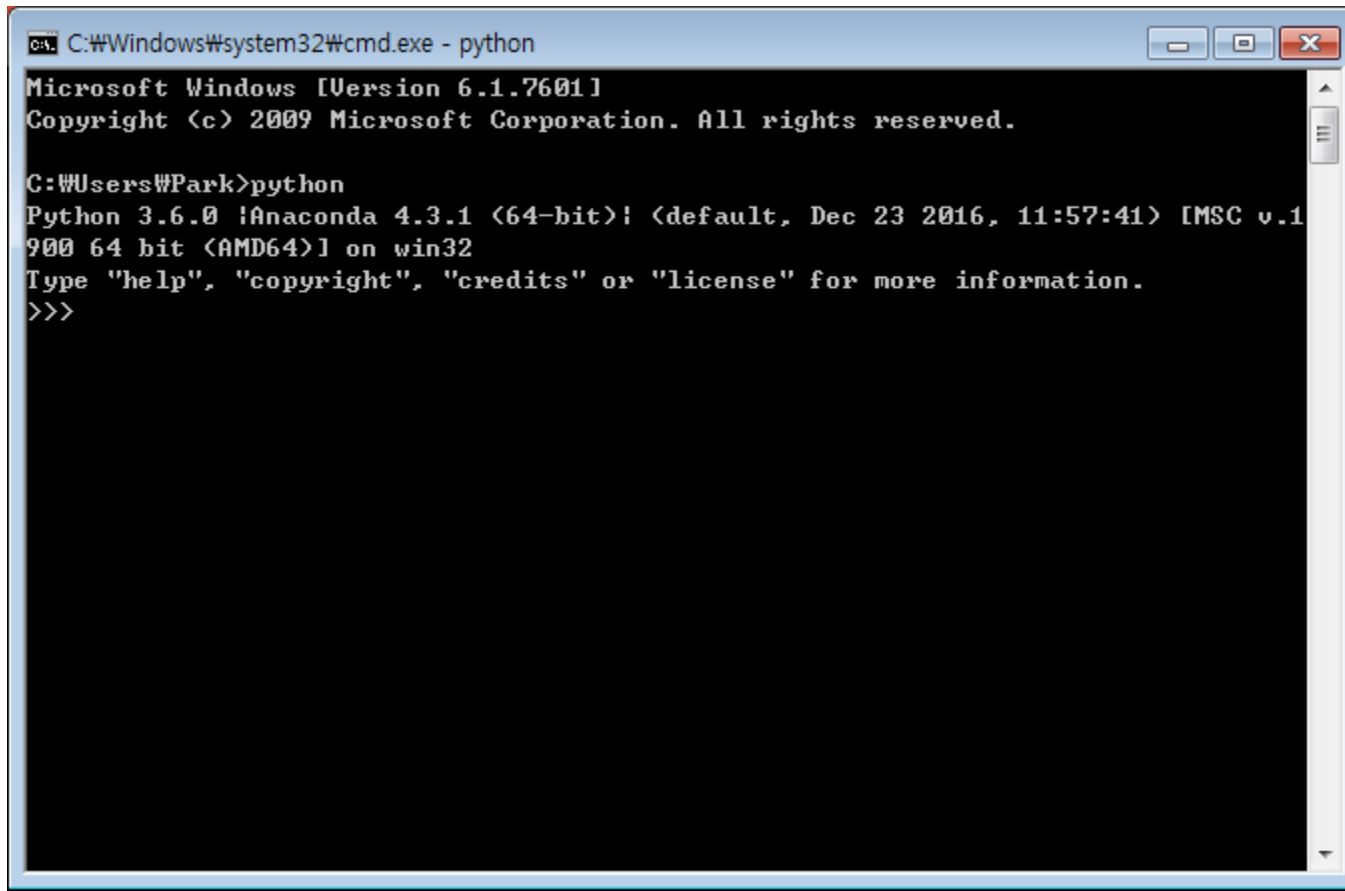
See upcoming and previous webinars from our experts.

VIEW NOW

2. 파이썬 설치

❖ 설치 확인

- ✓ cmd를 실행
- ✓ python 이라고 입력 한 후 버전 확인



```
CA: C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Park>python
Python 3.6.0 |Anaconda 4.3.1 (64-bit)| (default, Dec 23 2016, 11:57:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

3. IDE

| IDE

- ❖ ipython: 아나콘다를 설치하면 같이 설치
- ❖ PyDev 플러그 인을 이용한 이클립스
- ❖ Visual Studio의 파이썬 도구
- ❖ PyCharm



3. IDE

❖ anaconda와 ipython 업데이트

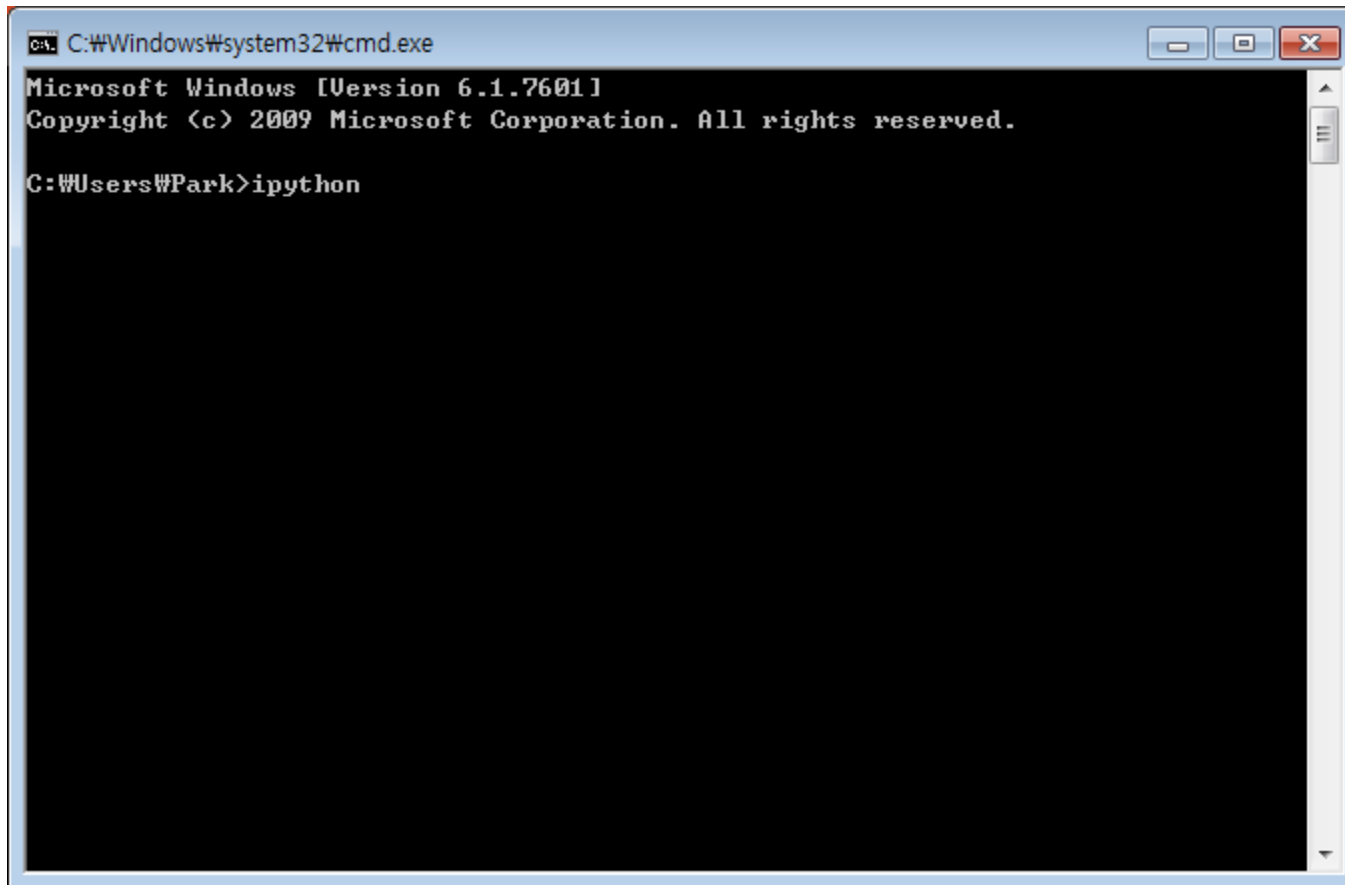
conda update conda

conda update ipython



3. IDE

- ❖ ipython을 이용한 코딩 및 실행

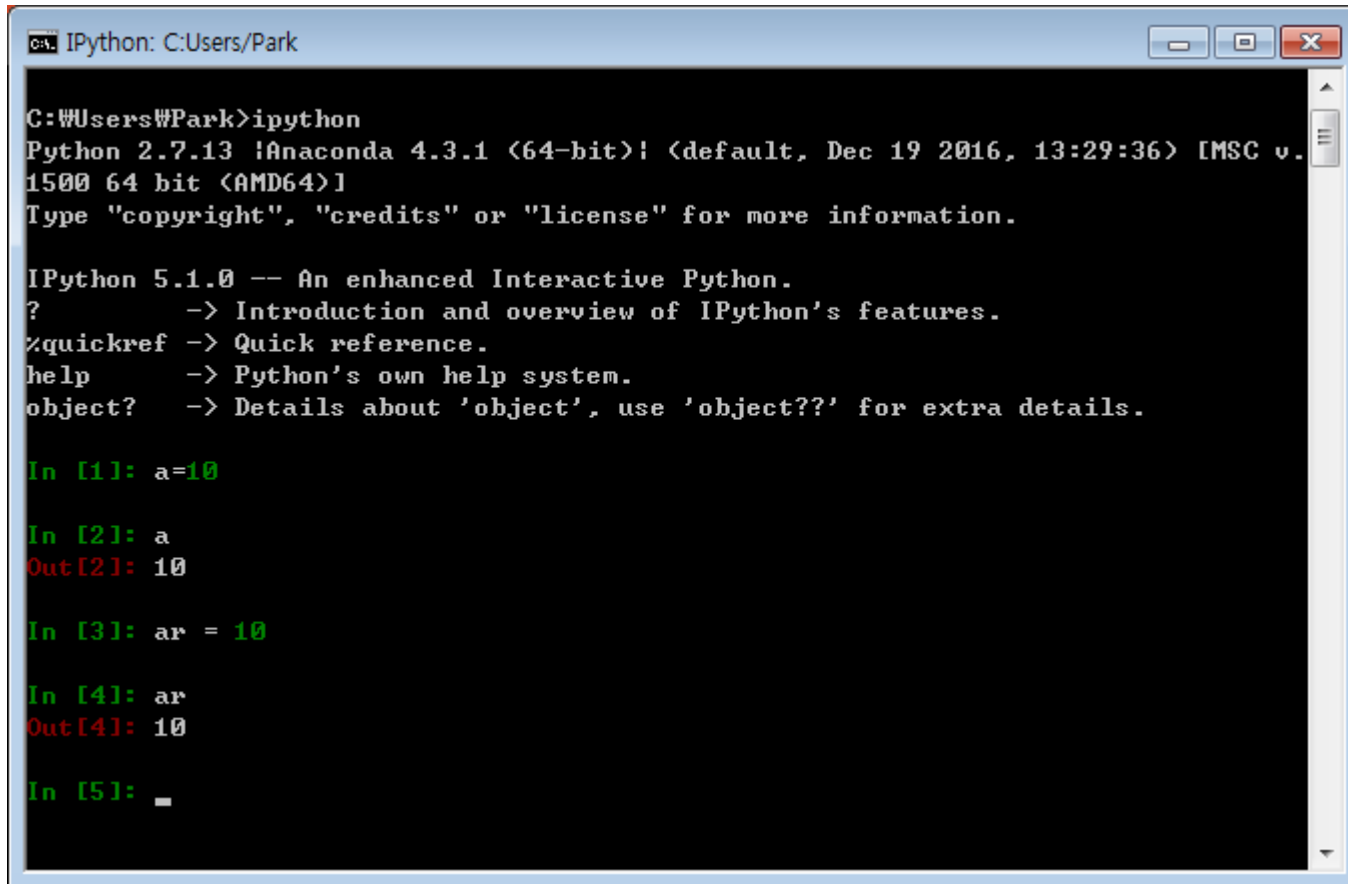


```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Park>ipython
```


3. IDE

- ❖ 객체의 이름만 입력하면 바로 출력



```
C:\Users\Park>ipython
Python 2.7.13 |Anaconda 4.3.1 (64-bit)| (default, Dec 19 2016, 13:29:36) [MSC v.
1500 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: a=10

In [2]: a
Out[2]: 10

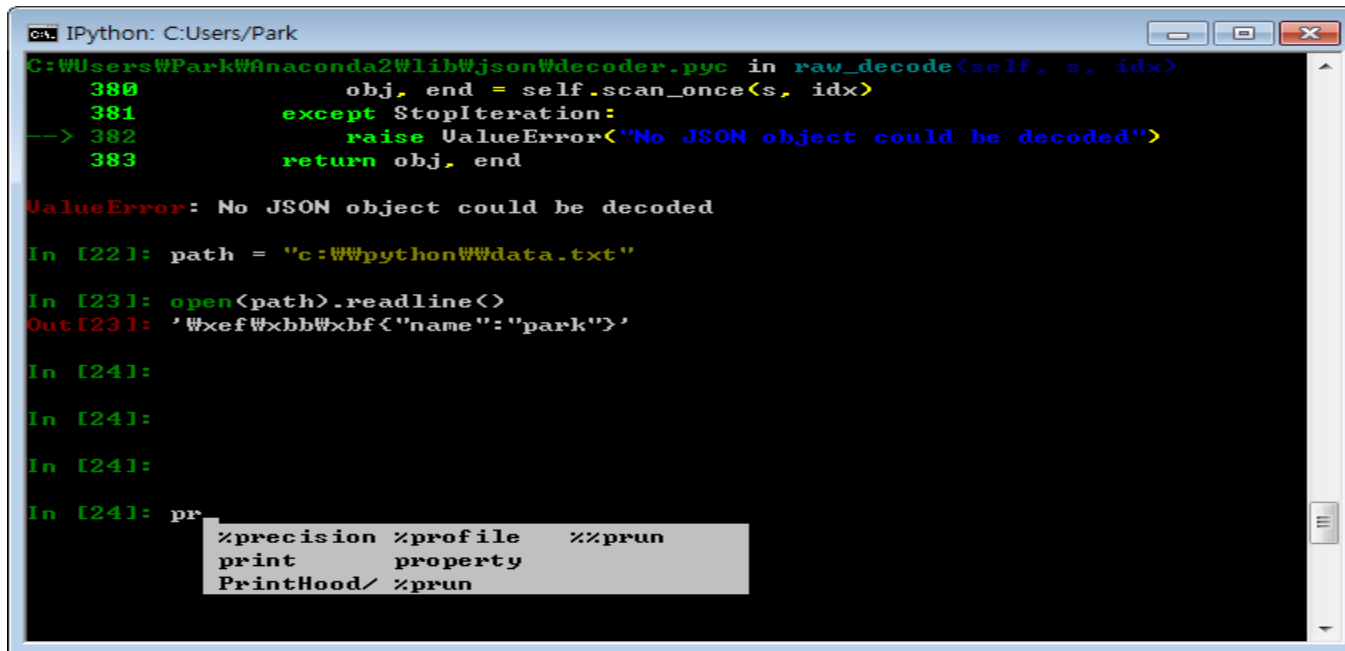
In [3]: ar = 10

In [4]: ar
Out[4]: 10

In [5]: _
```

4.ipython

- ❖ 탭을 이용한 자동완성 기능 제공: 명령을 입력하는 동안 Tab을 누르면 현재 위치에서 사용할 수 있는 변수를 화면에 출력
- ❖ 변수 이름 앞이나 뒤에 ?를 입력하면 변수에 대한 정보를 출력
- ❖ 함수 이름 뒤에 ??를 입력하면 함수의 소스 코드 출력



```
IPython: C:\Users\Park
C:\Users\Park\Anaconda2\lib\json\decoder.py in raw_decode(self, s, idx)
    380         obj, end = self.scan_once(s, idx)
    381     except StopIteration:
--> 382         raise ValueError("No JSON object could be decoded")
    383     return obj, end

ValueError: No JSON object could be decoded

In [22]: path = "c:\\python\\data.txt"

In [23]: open(path).readline()
Out[23]: 'b\xef\xbb\xbf{"name":"park"}'

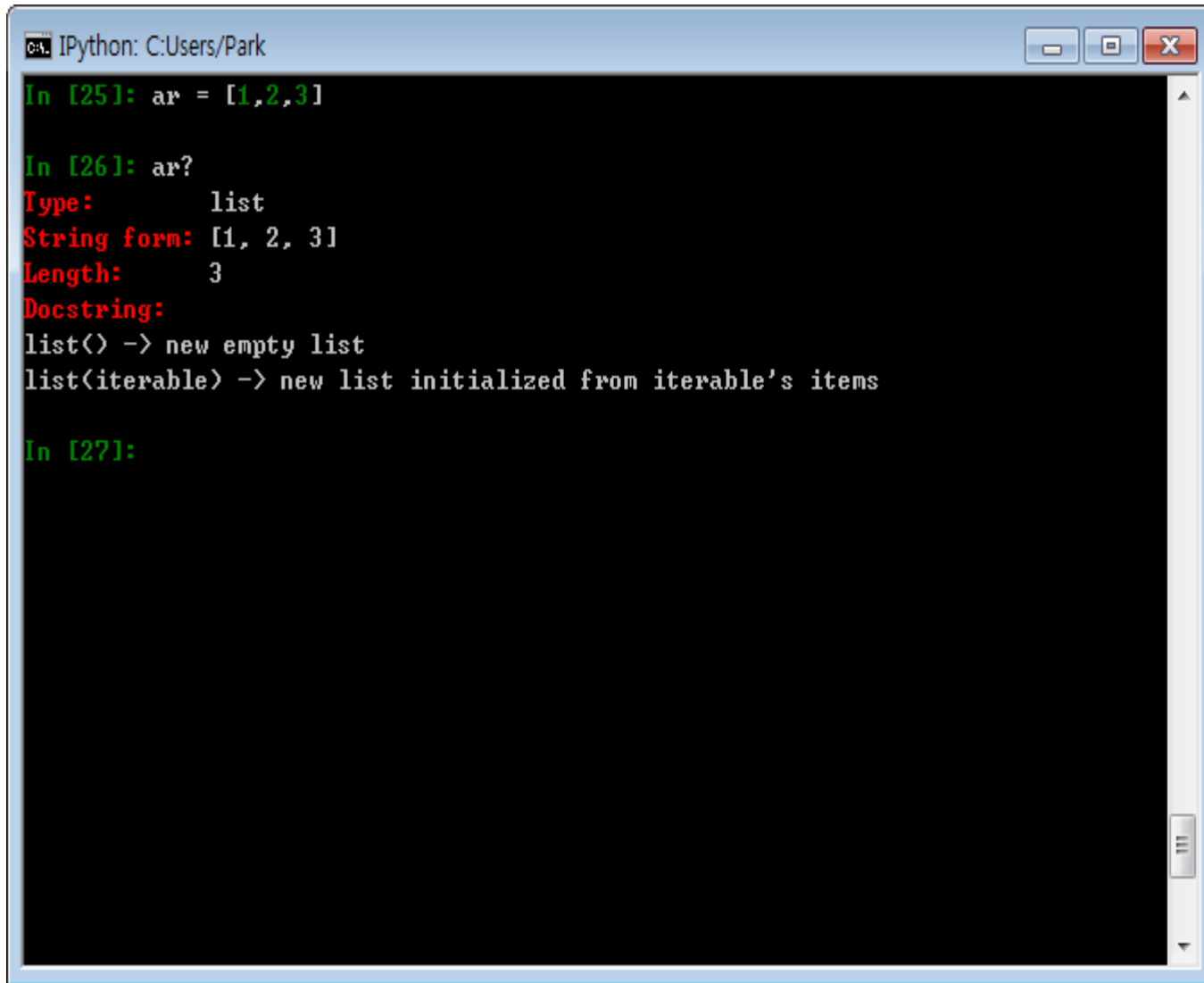
In [24]:

In [24]:

In [24]:

In [24]: pr
%precision %profile %%prun
print      property
PrintHood/ %prun
```

4.ipython



The screenshot shows an IPython terminal window with the title bar "IPython: C:\Users\Park". The terminal has a black background with green text for input and red text for output. The user enters three commands: `ar = [1,2,3]`, `ar?`, and an empty prompt. The output for `ar?` displays the type, string form, length, and docstring for the list object.

```
IPython: C:\Users\Park

In [25]: ar = [1,2,3]

In [26]: ar?
Type:      list
String form: [1, 2, 3]
Length:    3
Docstring:
list() -> new empty list
list(iterable) -> new list initialized from iterable's items

In [27]:
```

4.ipython

❖ 작업 디렉토리 확인

In [1]: `import os`

In [2]: `os.getcwd()`

Out[2]: 'C:\Users\WWPark'

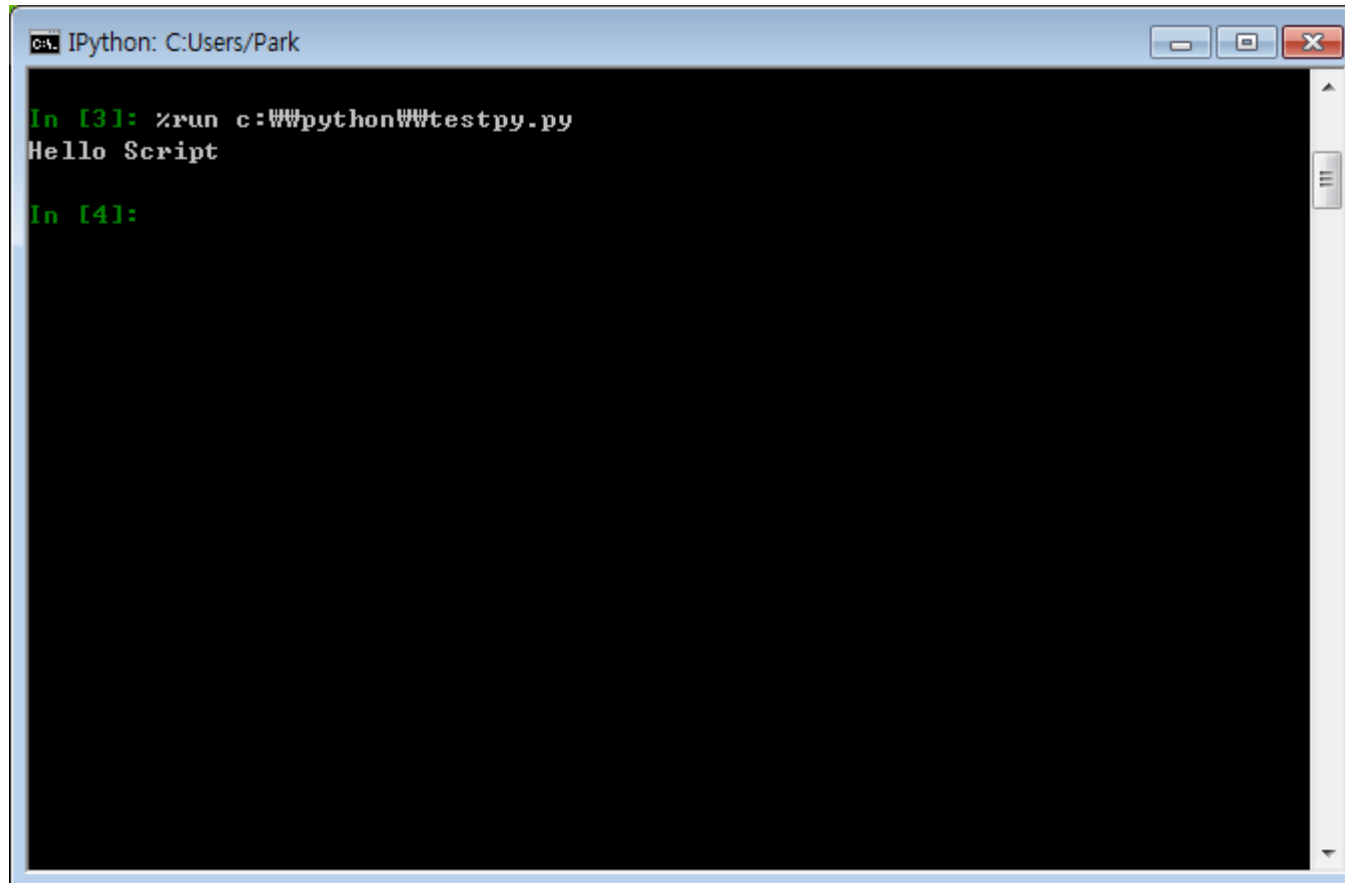
❖ 작업 디렉토리 변경

In [4]: `os.chdir('c:\python')`



4.ipython

- ❖ %run 파일경로: 파일 경로에 해당하는 파이썬 파일 실행
- ❖ 아래 코드를 갖는 파이썬 파일을 작성하고 실행
`print("Hello Script")`



The screenshot shows an IPython terminal window with the title bar "C:\ IPython: C:\Users\Park". The terminal has a black background with green text. The first prompt is "In [3]:", followed by the command "%run c:\www\python\www\testpy.py". The output "Hello Script" is displayed on the next line. The second prompt is "In [4]:".

```
C:\ IPython: C:\Users\Park
In [3]: %run c:\www\python\www\testpy.py
Hello Script
In [4]:
```

4.ipython

❖ 단축키

- Ctrl+P 또는 위 화살표 키: 명령어 히스토리를 역순으로 검색하기
- Ctrl+N 또는 아래 화살표 키: 명령어 히스토리에서 최근 순으로 검색하기
- Ctrl+R: readline 명령어 형식의 히스토리 검색(부분 매칭)하기
- Ctrl+Shift+V: 클립보드에서 텍스트 붙여넣기
- Ctrl+C: 현재 실행 중인 코드 중단하기
- Ctrl+A: 커서의 줄의 처음으로 이동하기
- Ctrl+E: 커서의 줄의 끝으로 이동하기
- Ctrl+K: 커서가 놓은 곳부터 줄의 끝까지 텍스트 삭제하기
- Ctrl+U: 현재 입력된 모든 텍스트 지우기
- Ctrl+F: 커서를 앞으로 한 글자씩 이동하기
- Ctrl+B: 커서를 뒤로 한 글자씩 이동하기
- Ctrl+L: 화면 지우기

4.ipython

❖ 매직 명령어: ipython이 제공하는 특수 명령어

%quickref: ipython의 빠른 도움말 표시

%magic: 모든 매직 함수에 대한 상세 도움말 출력

%debug: 최근 예외 트레이스백의 하단에서 대화형 디버거로 진입

%hist: 명령어 입력(그리고 선택적 출력) history 출력

%pdb: 예외가 발생하면 자동으로 디버거로 진입

%paste: 클립보드에서 들여쓰기가 된 채로 파이썬 코드 가져오기

%cpaste: 실행 파이썬 코드를 수동으로 붙여 넣을 수 있는 프롬프트 표시

%reset: 대화형 네임스페이스에서 정의된 모든 변수와 이름을 삭제

%page OBJECT: 객체를 pager를 통해 보기 좋게 출력

%run script.py: ipython 내에서 파이썬 스크립트 실행

%prun statement: cProfile을 통해 statement를 실행하고 프로파일링 결과를 출력

%time statement: 단일 statement 실행 시간을 출력

%timeit statement: statement를 여러차례 실행한 후 평균 실행 시간을 출력. 매우 짧은 시간 안에 끝나는 코드의 시간을 측정할 때 유용

%who, %who_ls, %whos: 대화형 네임스페이스 내에서 정의된 변수를 다양한 방법으로 표시

%xdel variable: variable을 삭제하고 ipython 내부적으로 해당 객체에 대한 모든 참조를 제거

%%writefile filename: filename인 파일을 생성

4.ipython

```
In [10]: import numpy as np
```

```
In [11]: a = np.random.randn(100,100)
```

```
In [12]: %time np.dot(a, a)
```

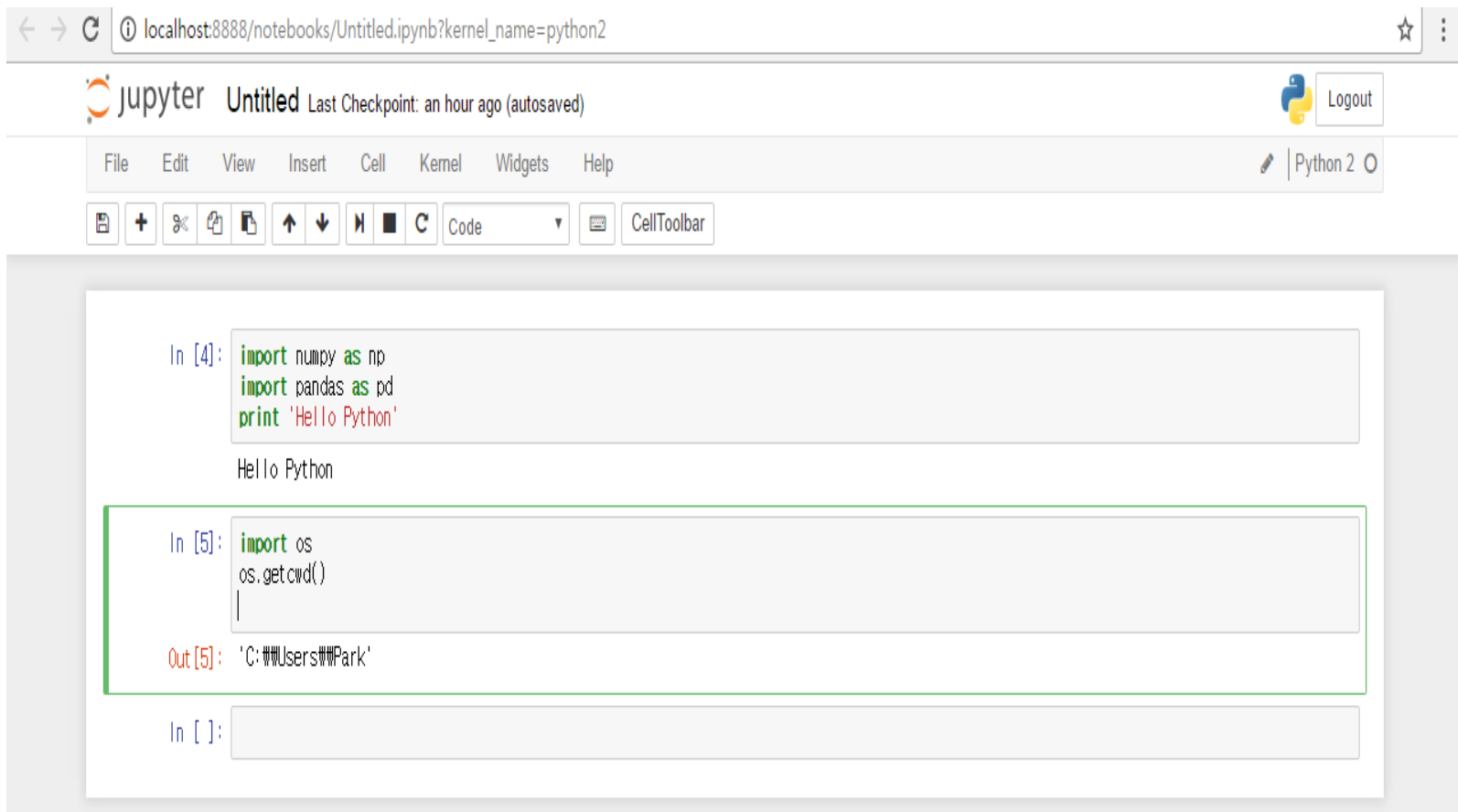
Wall time: 710 ms

```
Out[12]:
```

```
array([[ 19.58800137,  2.62837613, -8.28386112, ..., -4.0982812 ,
        -13.99239531,  5.67487065],
       [-5.55024439, 10.85218715,  6.14092077, ..., -13.73882312,
        -0.72756494,  2.39935928],
       [-3.60873191, -15.89902742,  8.33705822, ..., 10.37198751,
        -2.66928403, -4.6852677 ],
       ...,
       [-1.53904249, -1.2630197 , 25.4836311 , ...,  7.55673741,
         1.06642476,  3.21558587],
       [ 8.5091311 , -14.67747325,  9.04744578, ..., -9.93647063,
         4.96999403, -8.40284447],
       [-6.94798702, -5.71814232, 11.46917944, ...,  5.43738466,
        -8.7403922 , -6.61607408]])
```


5.ipython notebook

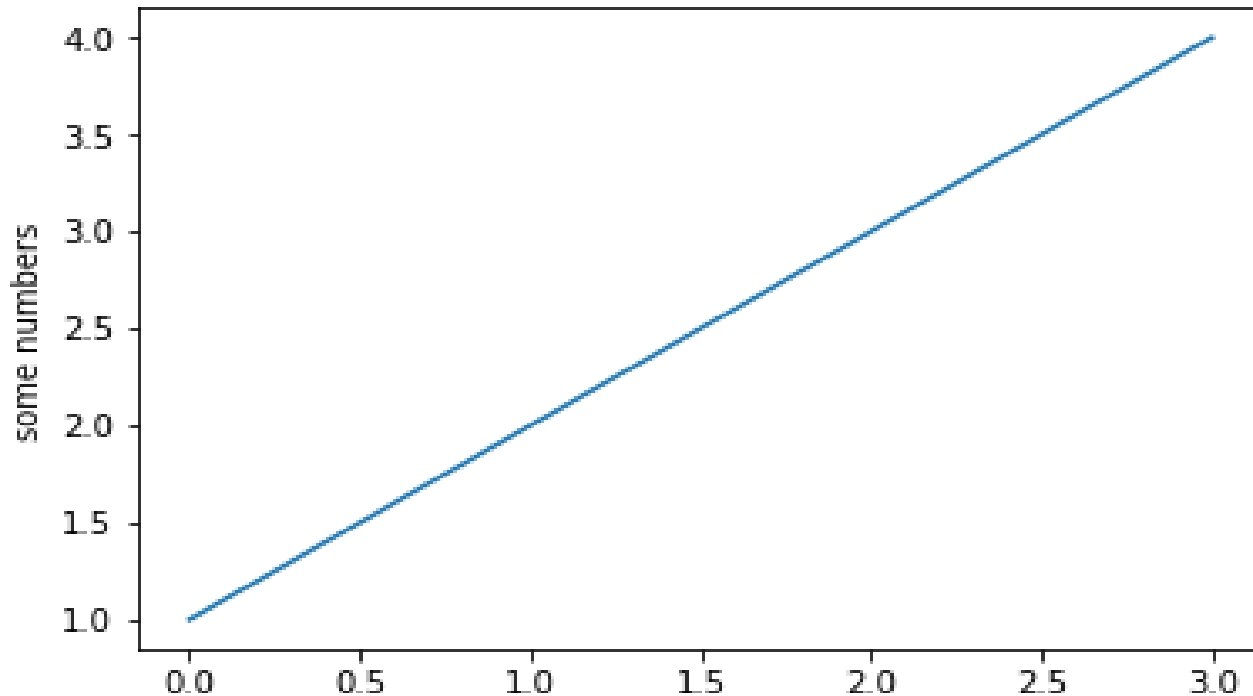
C:\Users\Park>>jupyter notebook



The screenshot displays the Jupyter Notebook web interface in a browser. The address bar shows the URL `localhost:8888/notebooks/Untitled.ipynb?kernel_name=python2`. The notebook title is "Untitled" with a subtext "Last Checkpoint: an hour ago (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, creating new cells, and running code. The code area contains two input cells. The first cell, labeled "In [4]:", contains the following code: `import numpy as np`, `import pandas as pd`, and `print 'Hello Python'`. Below this code, the output "Hello Python" is displayed. The second cell, labeled "In [5]:", contains the code `import os` and `os.getcwd()`. Below this code, the output "Out [5]: 'C:\\Users\\Park'" is displayed. A third input cell, labeled "In []:", is currently empty.

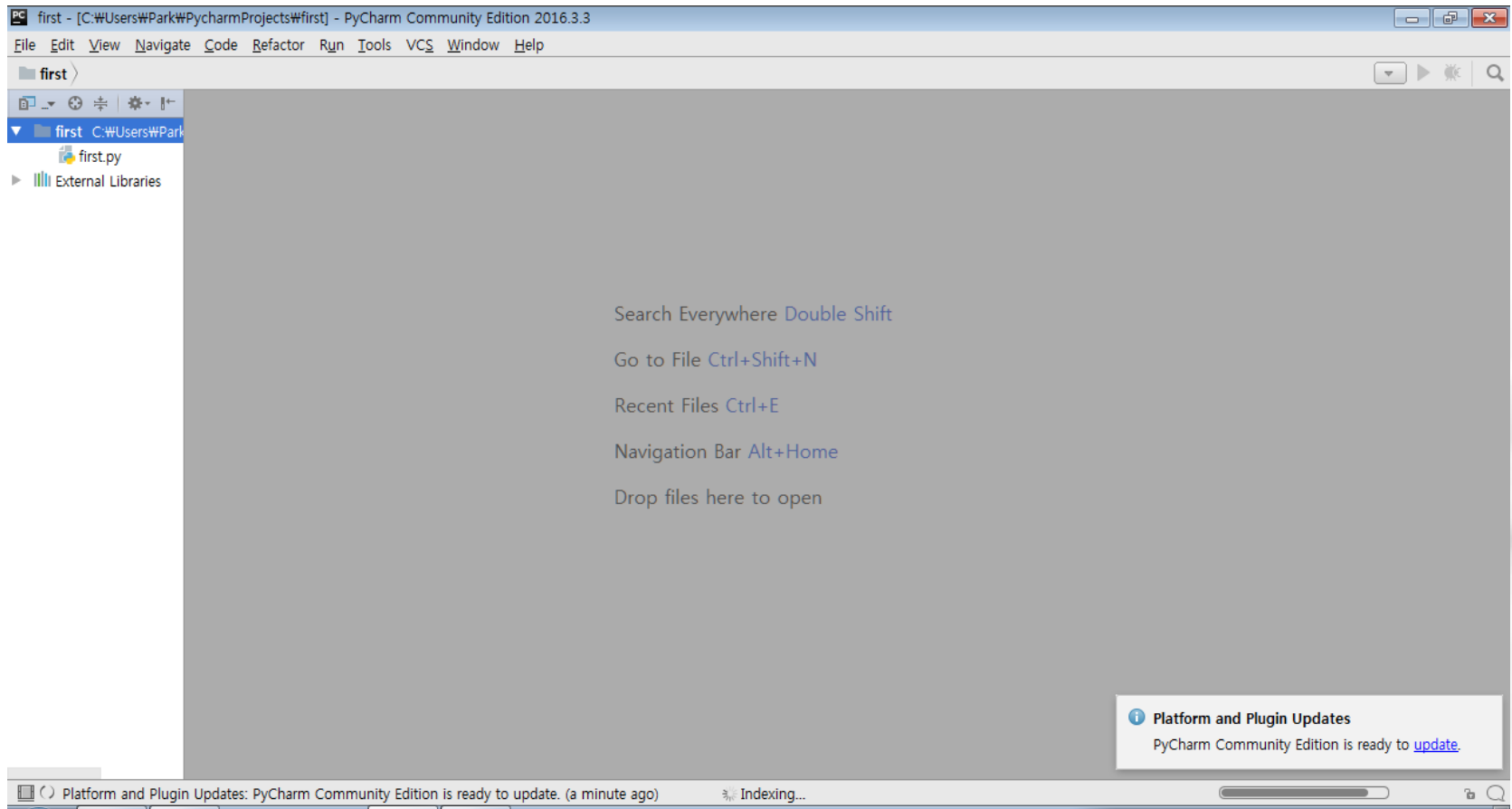
5.ipython notebook

```
import matplotlib.pyplot as plt  
plt.plot([1,2,3,4])  
plt.ylabel('some numbers')  
plt.show()
```

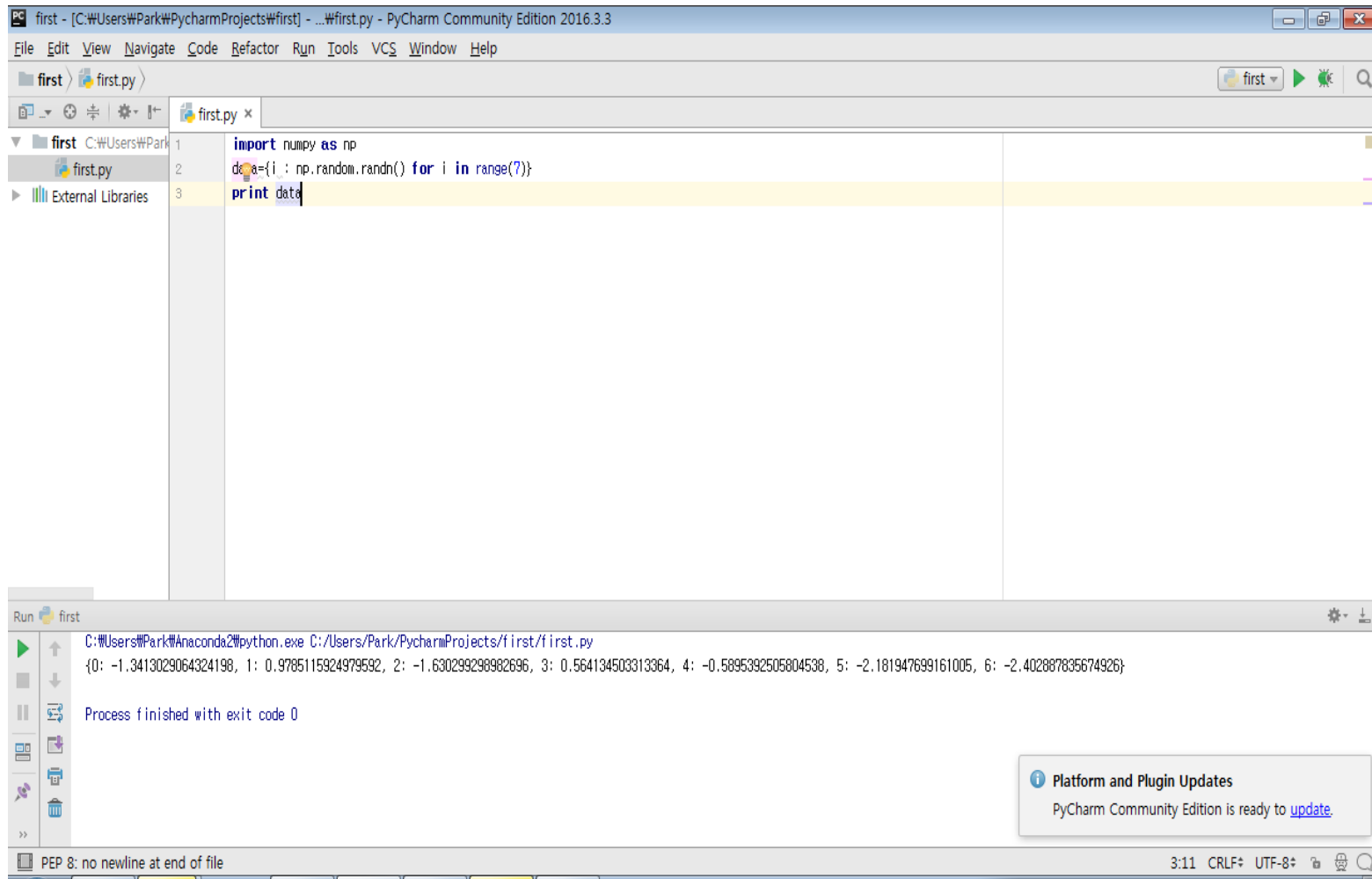


6. pycharm

- ❖ Pycharm은 IntelliJ, CLion 등을 제공하는 JetBRAINS에서 배포하는 프로그램입니다.
- ❖ <http://www.jetbrains.com/pycharm/download/#section=windows> 에서 Community 버전을 선택해서 다운로드 후 설치



6. pycharm

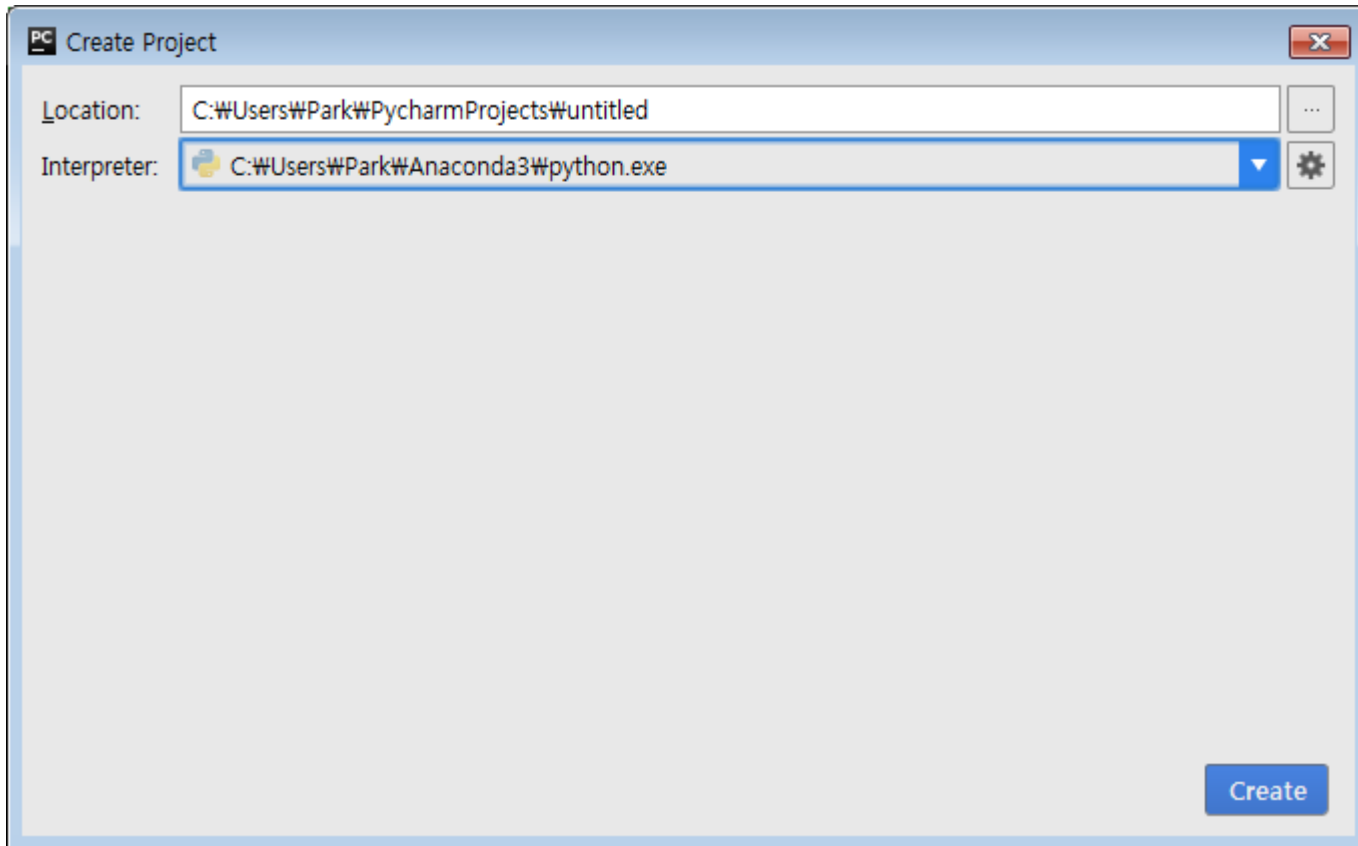


6. pycharm

- ❖ 가상 환경(Virtual Environment): 사용자가 정한 임의의 디렉토리 밑에 Python과 관련 패키지 등을 함께 넣어 그 안에서 독립적인 파이썬 개발 환경을 할 수 있도록 한 것으로 Lightweight, Self-contained 파이썬 개발 환경으로서 필요한 경우 한 개발 머신 안에 여러 개의 가상 환경을 만들고 각 가상 환경에서 다른 파이썬 버전이나 다양한 패키지들을 독립적으로 설치 사용할 수 있는 가상적 개발 환경
- ❖ Python 3 (3.4+)는 기본적으로 가상환경을 생성하는 유틸리티를 포함하고 있습니다.
- ❖ 가상 환경을 만들기 위해서는 pyvenv 라는 유틸리티를 사용하는데, 각 OS 마다 (윈도우즈, Mac, 리눅스) 사용법이 약간씩 다릅니다.
- ❖ Pycharm에서는 pyvenv 유틸리티가 번들로 포함되어 있으므로 가상환경 생성이 쉽습니다.

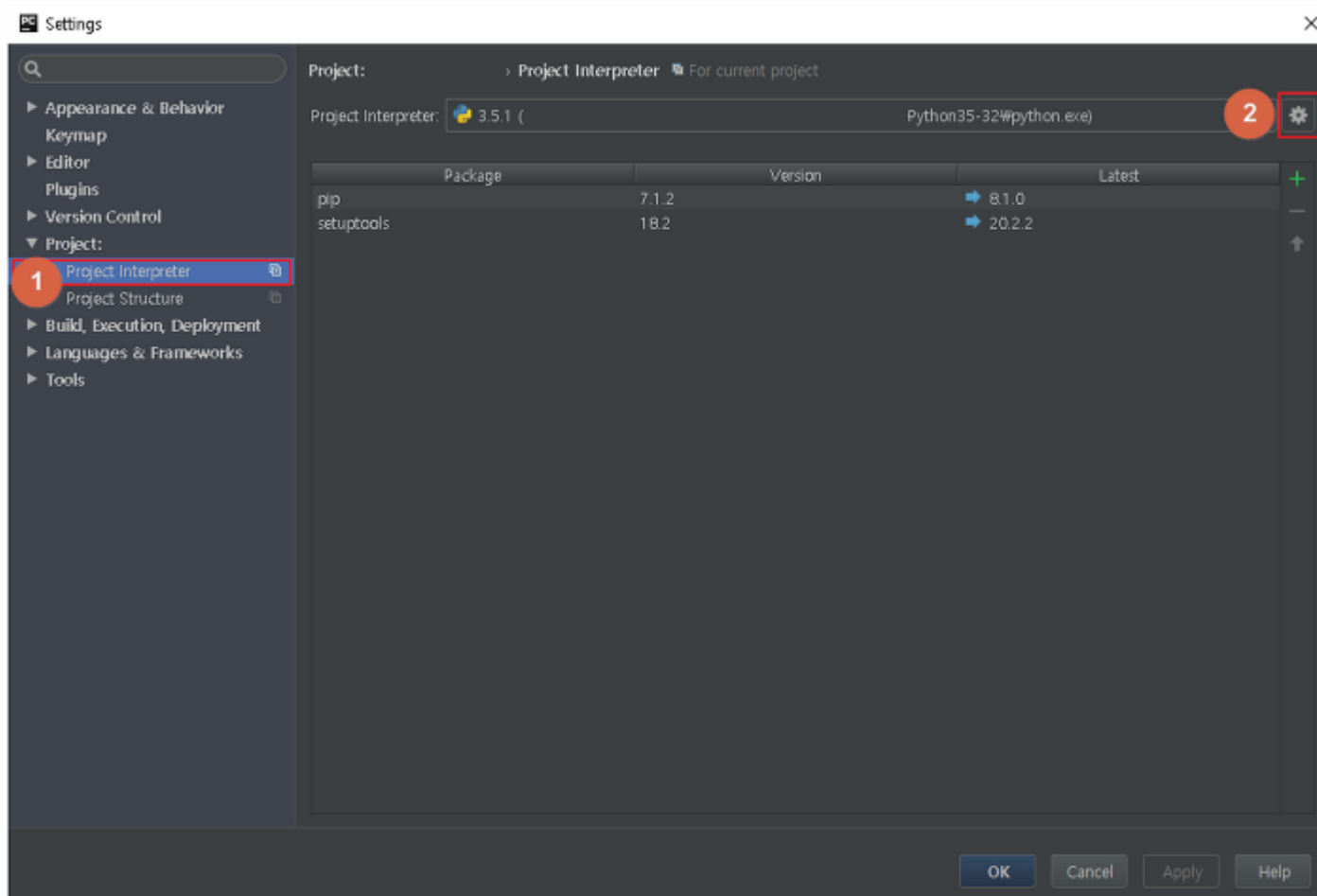
7. Pycharm 가상환경 구축

- ❖ 프로젝트 생성 [File] – [New Project]



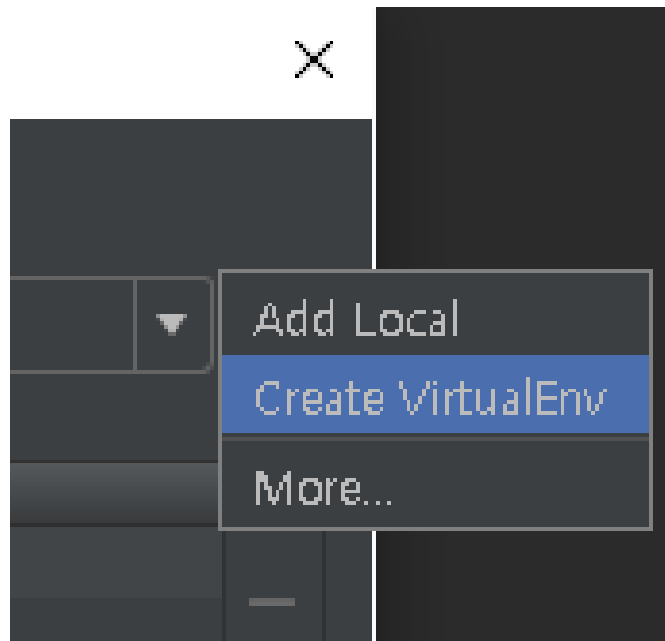
7. Pycharm 가상환경 구축

❖ 프로젝트 생성 [File] – [settings]



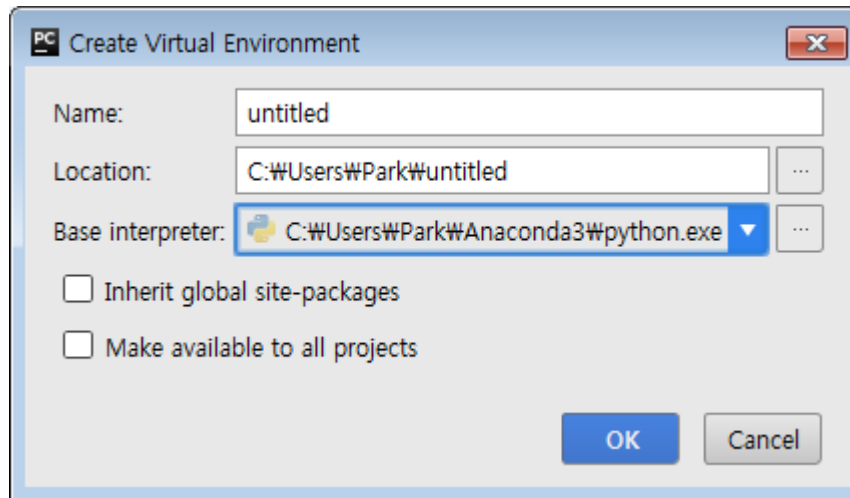
7. Pycharm 가상환경 구축

❖ Create VirtualEnv 선택



7. Pycharm 가상환경 구축

❖ 옵션 설정



- ✓ Name: 가상 환경 이름
- ✓ Location: 가상 환경이 생성될 경로
- ✓ Base interpreter: 사용할 파이썬 버전
- ✓ Inherit global site-packages: 기본 인터프리터의 site-packages를 상속받아 사용할 경우 체크
- ✓ Make available to all projects: 모든 프로젝트에서 사용할 경우 체크