

# 금융 데이터 분석을 위한 파이썬 개발 환경 만들기

이승준 [fb.com/plusjune](https://fb.com/plusjune)

# 강의 내용

---

- Scientific Python 배포판 설치
- 운영체제 별 설치 (Linux, OSX, Windows) 방법 및 차이
- 관리자 설치(전체), 사용자 설치(개인)
- 가상 클라우드 개발환경에 설치 ([c9.io](http://c9.io), [nitrous.io](http://nitrous.io))에 개발환경을 만들기
- 다양한 파이썬 버전과 패키지를 함께 사용하는 방법
- Jupyter Notebook을 서버 형태로 설치하여 어디서나 사용

# 금융 데이터 분석의 단계

---

수집 collect

- Question
- Collect data

처리 Prepare

- Organize
- Cleanse

분석 Analyze

- Pattern
- Filter
- Relationship
- Summarize

적용 Apply

- Visualize
- Report, Share
- Make decisions


# 주로 사용하는 도구

---

## 수집 Collect

- BeautifulSoup
- StringIO
- requests

## 저장 Prepare

- Csv 
- PySqlite (SQLite)
- PyMongo (MongoDB)
- 

## 분석 Analyze

- IPython
- NumPy
- pandas

## 적용 Apply

- Matplotlib 

# 파이썬 개발 환경 설치

---

- 온라인 학습 개발 환경
- 운영체제 별 설치 (Linux, OSX, Windows)
- 관리자 설치(전체), 사용자 설치(개인)
- 가상 클라우드 개발환경에 설치

# 간단한 온라인 실습과 테스트

---

- <https://www.pythonanywhere.com/try-ipython/>



# Scientific python 배포판

---

다양한 파이썬 패키지들을 묶은 설치본

- <https://continuum.io/downloads> ANACONDA
- <https://python-xy.github.io> Python(x,y)
- <https://enthought.com/products/canopy> canopy



## • 윈도우 (Python 3.4, 32-bit 버전) 설치

← → ↺ 🏠 <https://www.continuum.io/downloads>

It includes more than 300 of the most popular Python packages for science, math, engineering, and data analysis. [See the packages included with Anaconda.](#)

### Which version should I download and install?

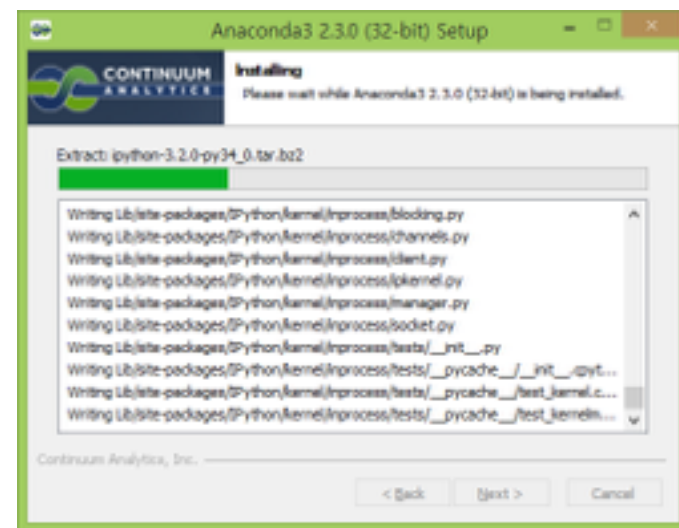
Because Anaconda includes installers for Python 2.7 and 3.4, either is fine. Using either version, you can use Python 2.6 and 3.3 with the conda command. You can create a 3.4 environment with the conda command if you've downloaded 2.7 — and vice versa.

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 3.4
<div>Windows 64-bit Graphical Installer</div> <div>336M</div>	<div>Windows 64-bit Graphical Installer</div> <div>336M</div>
<div>Windows 32-bit Graphical Installer</div> <div>281M</div>	<div>Windows 32-bit Graphical Installer</div> <div>277M</div>

Behind a firewall? Use these zipped Windows installers.





# 설치 확인

```
명령 프롬프트 - ipython

C:\Users\Seung-June>ipython
Python 3.4.3 [Anaconda 2.3.0 (32-bit)] <default, Mar  6 2015, 12:08:17> [MSC v.1600 32 bit (Intel)]
Type "copyright", "credits" or "license" for more information.

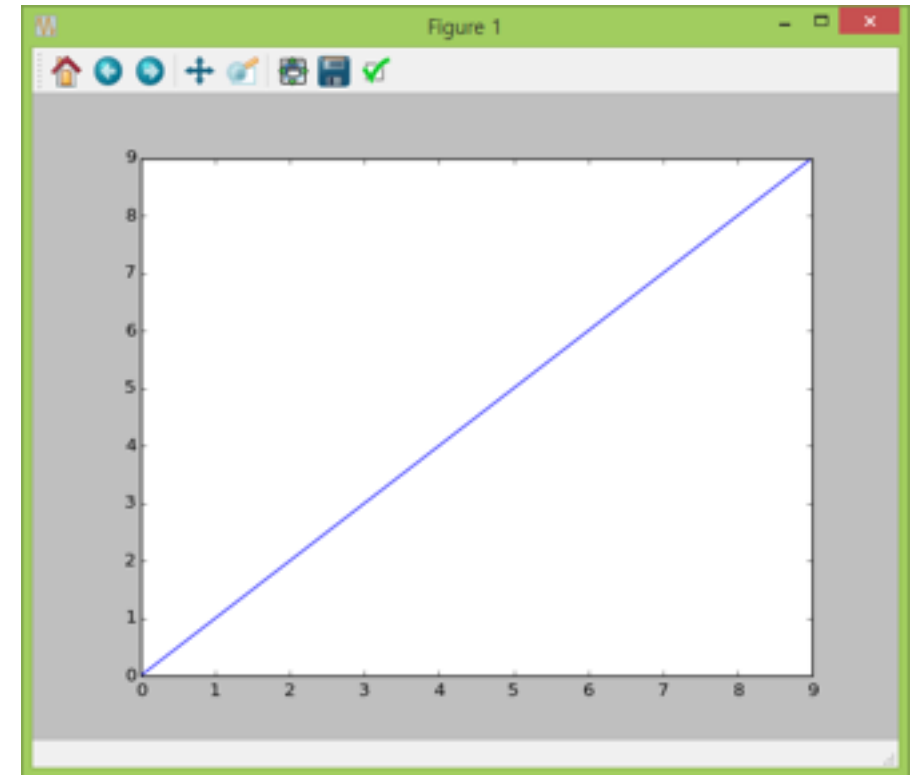
IPython 3.2.0 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: %pylab
Using matplotlib backend: Qt4Agg
Populating the interactive namespace from numpy and matplotlib

In [2]: import matplotlib.pyplot as plt

In [3]: plt.plot(range(0,10))
Out[3]: [<matplotlib.lines.Line2D at 0x5473330>]

In [4]:
```



# OSX

---

다운로드 하여 설치

Anaconda3-2.3.0-MacOSX-x86\_64.pkg

명령라인

```
$ bash Anaconda-2.3.0-MacOSX-x86_64.sh
```

# Linux에 설치

---

다운로드 후 bash 명령으로 설치

```
$ bash Anaconda-2.3.0-Linux-x86.sh
```

# 파이썬 로컬 컴파일 설치

---

파이썬 사용자 영역에 컴파일하여 설치 (리눅스, Unix, Mac OS)

```
# 다운로드
```

```
$ wget https://www.python.org/ftp/python/3.4.3/Python-3.4.3.tgz
```

```
$ tar xvf Python-3.4.3.tgz
```

```
$ cd xvf Python-3.4.3
```

```
# 설치
```

```
$ ./configure --prefix=$HOME/python34
```

```
$ make; make install
```

# virtualenv



독립적인 실행 환경을 만들기 위해 사용

```
$ virtualenv --python ~/python34/bin/python3 venv34
$ source venv34/bin/activate
(venv34) $
(venv34) $ deactivate
$
```

# 파이썬 로컬 설치

---

```
# install python packages
```

```
(venv34) $ pip install numpy
```

```
(venv34) $ pip install pandas
```

```
(venv34) $ pip install requests
```

```
# install matplotlib
```

```
(venv34) $ pip install matplotlib
```

```
# install ipython
```

```
(venv34) $ pip install ipython pyzmq tornado jinja2
```

# 클라우드 IDE 서비스들

---

Nitrous.io, Cloud9, Koding, Codeanywhere

- 다양한 기능의 온라인 IDE
- 가상화된 VM(가상 머신)을 제공하기도 한다
- 코딩을 배우기에 최적의 환경

# update-alternatives

---

## python3를 기본으로 사용하기

```
sudo update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1  
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.4 2
```



# update-alternatives

update-alternatives 명령으로 동일한 이름에 대해 우선순위를 지정할 수 있다.  
(숫자가 높을 수록 우선순위가 높다)

우선순위를 바꾸려면 아래와 같이 한다.

```
→ ~ sudo update-alternatives --config python
```

There are 2 choices for the alternative python (providing /usr/bin/python).

Selection	Path	Priority	Status
* 0	/usr/bin/python2	2	auto mode
1	/usr/bin/python2	2	manual mode
2	/usr/bin/python3	1	manual mode

# pip, virtualenv 설치

---

```
# pip
```

```
sudo apt-get update
```

```
# pip
```

```
sudo apt-get install python3-pip
```

```
sudo apt-get install python3-setuptools
```

```
# virtualenv
```

```
sudo pip install virtualenv
```

# 패키지 설치

```
# numpy, pandas
```

```
sudo pip install numpy
```

```
sudo pip install pandas
```

```
# ipython, jupyter, matplotlib
```

```
sudo pip install ipython pyzmq tornado matplotlib jinja2 pygments
```

```
sudo pip install jupyter
```

```
# requests, BeautifulSoup4
```

```
sudo pip install requests
```

```
sudo pip install beautifulsoup4
```

```
➔ ~ pip --version  
pip 7.1.2 from /usr/local/lib/python3.4/dist-packages (python 3.4)  
➔ ~  
➔ ~ pip --version # 버전확인  
pip 7.1.2 from /usr/local/lib/python3.4/dist-packages (python 3.4)  
➔ ~  
➔ ~ pip install numpy  
Collecting numpy  
Downloading numpy-1.10.1.tar.gz (4.0MB)  
58% |██████████| 2.4MB 264kB/s eta 0:00:07
```

# 설치 확인

---

→ ~ pip list



beautifulsoup4 (4.4.1)

chardet (2.2.1)

colorama (0.2.5)

decorator (4.0.4)

...

# Notebook 서버 - 프로파일 생성

---

```
$ ipython profile create nbserve
```

~/.ipython/profile\_nbserve/ 디렉토리가 생기고, 하위에 프로파일 생성된다.

# Notebook 서버 – 비번 설정

---

```
In [1]: from IPython.lib import passwd
```

```
In [2]: passwd()
```

```
Enter password:
```

```
Verify password:
```

```
Out[2]: 'sha1:171e7d6cfbca:6591a11e293b48709e6385e71ee531083cccc2c7'
```

~/.ipython/profile\_nbserve/ipython\_notebook\_config.py 파일의 내용중 비번 부분을 수정

```
c = get_config()
```

```
c.NotebookApp.ip = '0.0.0.0'
```

```
c.NotebookApp.open_browser = False
```

```
c.NotebookApp.port = 8888
```

```
c.NotebookApp.password = u'sha1:171e7d6cfbca:6591a11e293b48709e6385e71ee531083cccc2c7'
```

# Notebook 서버 - 프로파일 생성

---

```
$ ipython notebook --profile="nbserv"
```

지정한 프로파일을 사용하여 노트북을 실행

# 설치 요약 (우분투)

---

# 시스템 전체에 걸쳐 Python 3.4가 기본이 되도록 설정한다.

```
sudo update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1
```

```
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.4 2
```

# pip 설치

```
sudo apt-get install python3-pip
```

```
sudo apt-get install python3-setuptools
```

# numpy, pandas 등 각종 패키지 설치

```
sudo pip install numpy
```

```
sudo pip install pandas
```

```
sudo pip install ipython pyzmq tornado matplotlib jinja2 pygments
```

```
sudo pip install jupyter
```

```
sudo pip install requests
```

```
sudo pip install beautifulsoup4
```



# 실행 요약

---

# ipython notebook 실행

sudo ipython notebook --ip=0.0.0.0 

# Review

---

- Pythonanywhere – 온라인 간단한 테스트, 코드 확인
- Scientific Python 배포판 – 한방 처방
- Virtualenv – 버전 혼선없는 독립적인 환경
- 클라우드 Jupyter notebook 서버 – 어디서나 접근 가능한 파이썬 개발 환경