Python Workshop

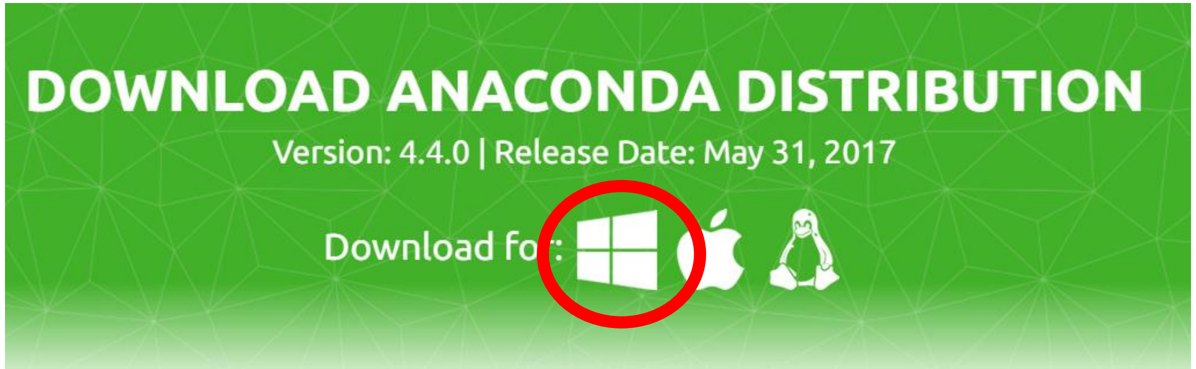
- Raspberry Pi -

2017년 8월

서강대학교

Anaconda

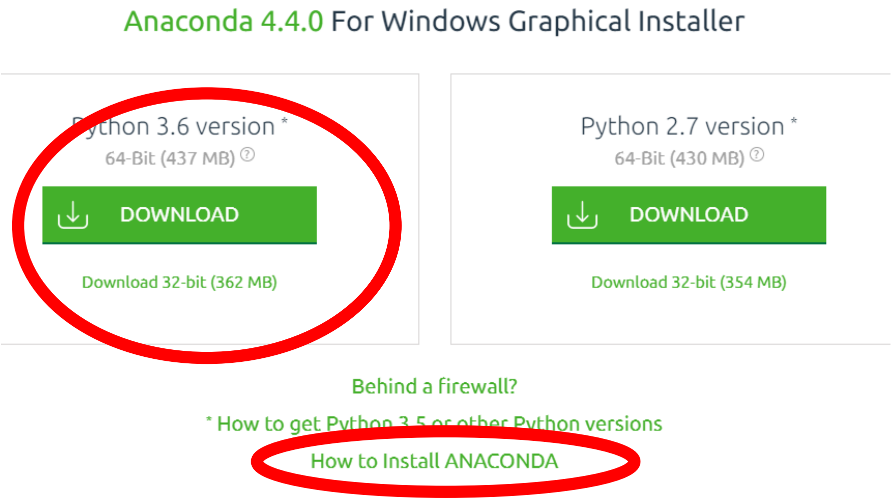
https://www.continuum.io/downloads



DOWNLOAD ANACONDA DISTRIBUTION

Version: 4.4.0 | Release Date: May 31, 2017

Download for:



Anaconda 4.4.0 For Windows Graphical Installer

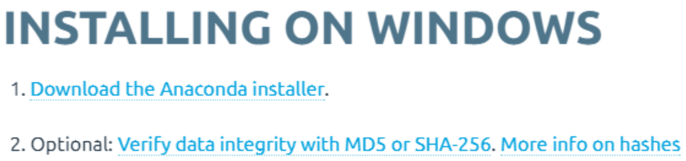
Python 3.6 version *
64-Bit (437 MB)
DOWNLOAD
Download 32-bit (362 MB)

Python 2.7 version *
64-Bit (430 MB)
DOWNLOAD
Download 32-bit (354 MB)

Behind a firewall?

* How to get Python 3.5 or other Python versions

How to Install ANACONDA



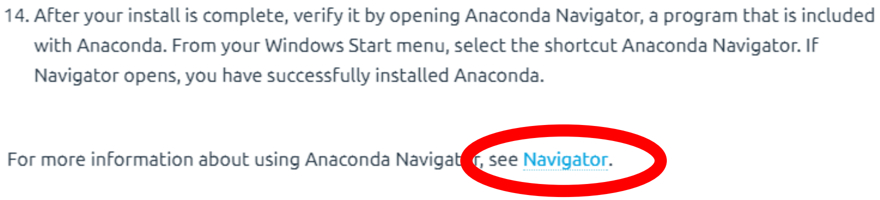
INSTALLING ON WINDOWS

1. Download the Anaconda installer.
2. Optional: Verify data integrity with MD5 or SHA-256. More info on hashes

.

.

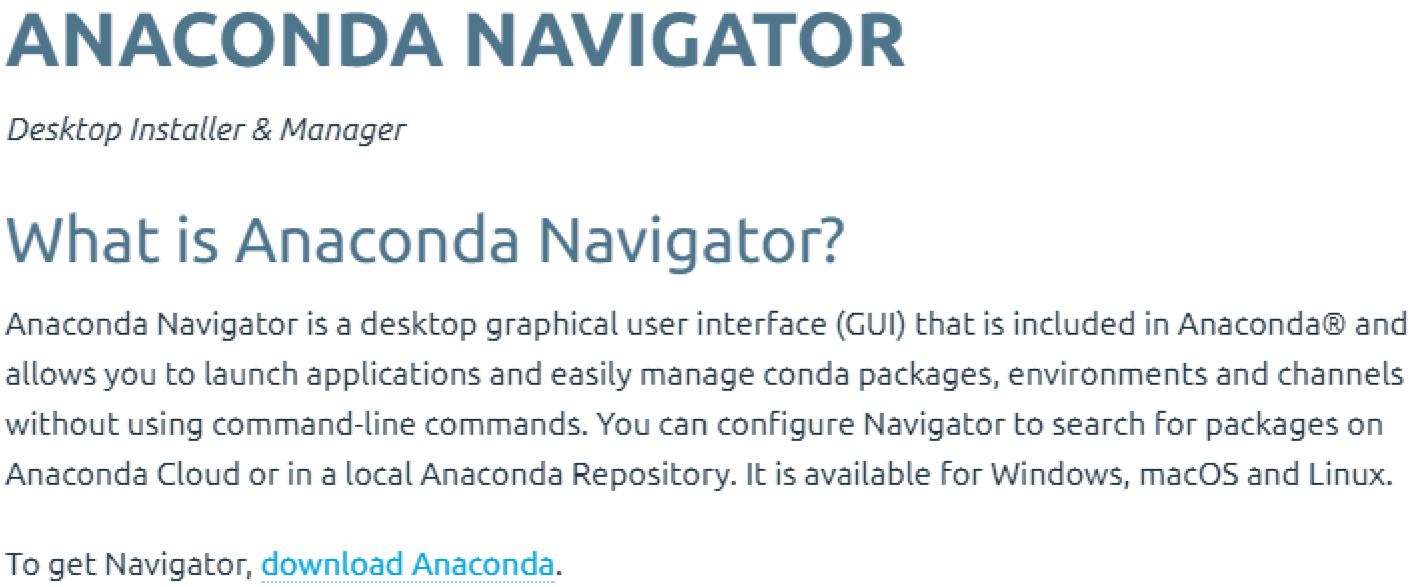
.



14. After your install is complete, verify it by opening Anaconda Navigator, a program that is included with Anaconda. From your Windows Start menu, select the shortcut Anaconda Navigator. If Navigator opens, you have successfully installed Anaconda.

For more information about using Anaconda Navigator, see Navigator.

Anaconda Navigator



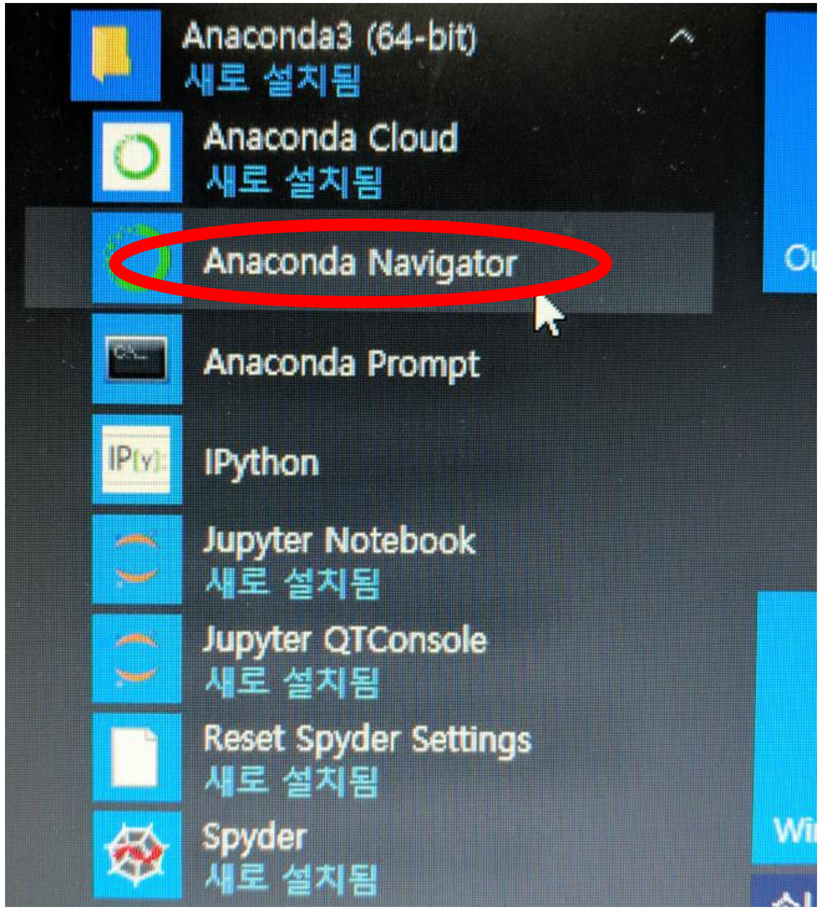
ANACONDA NAVIGATOR

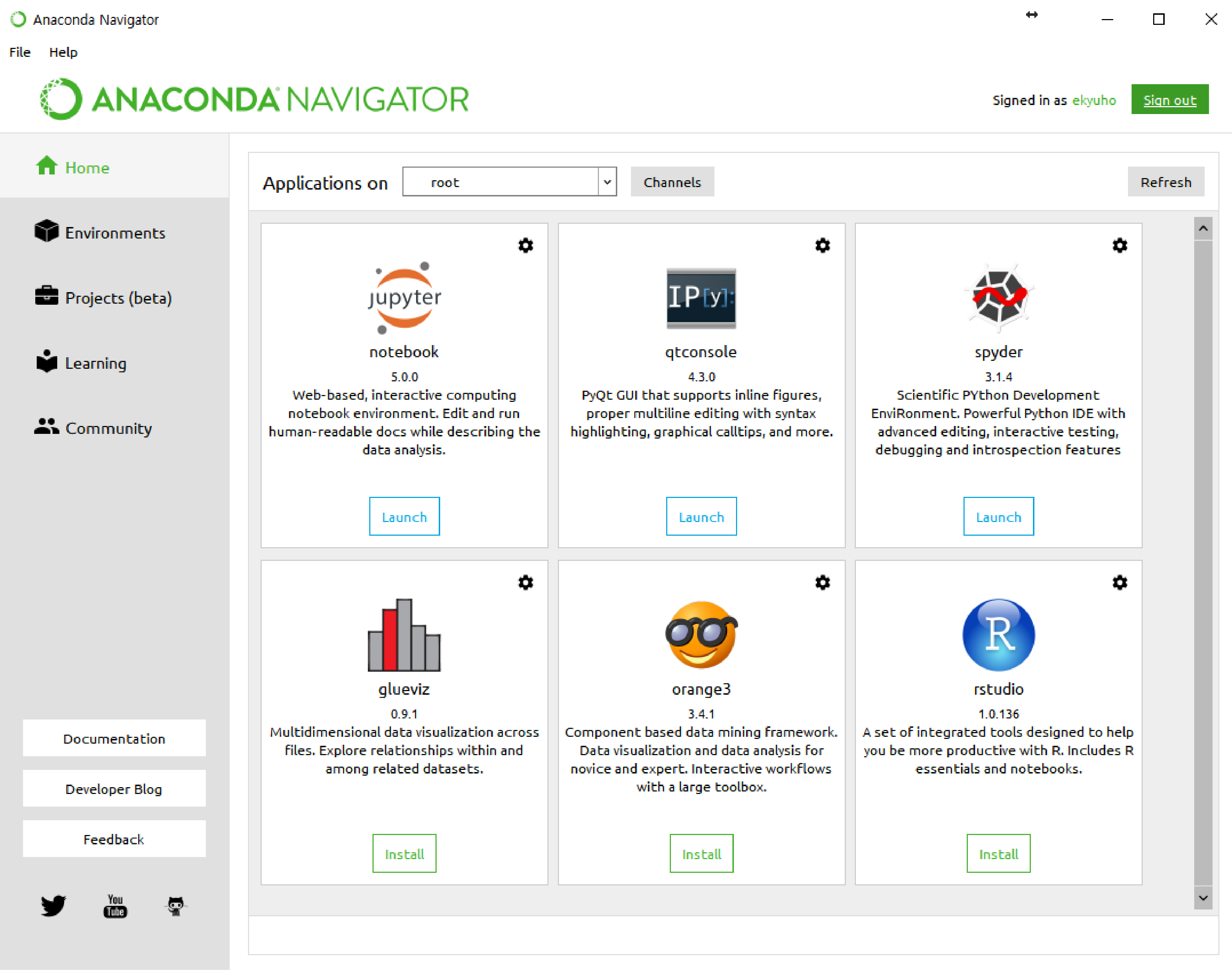
Desktop Installer & Manager

What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) that is included in Anaconda® and allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. You can configure Navigator to search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS and Linux.

To get Navigator, download Anaconda.





notebook
5.0.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

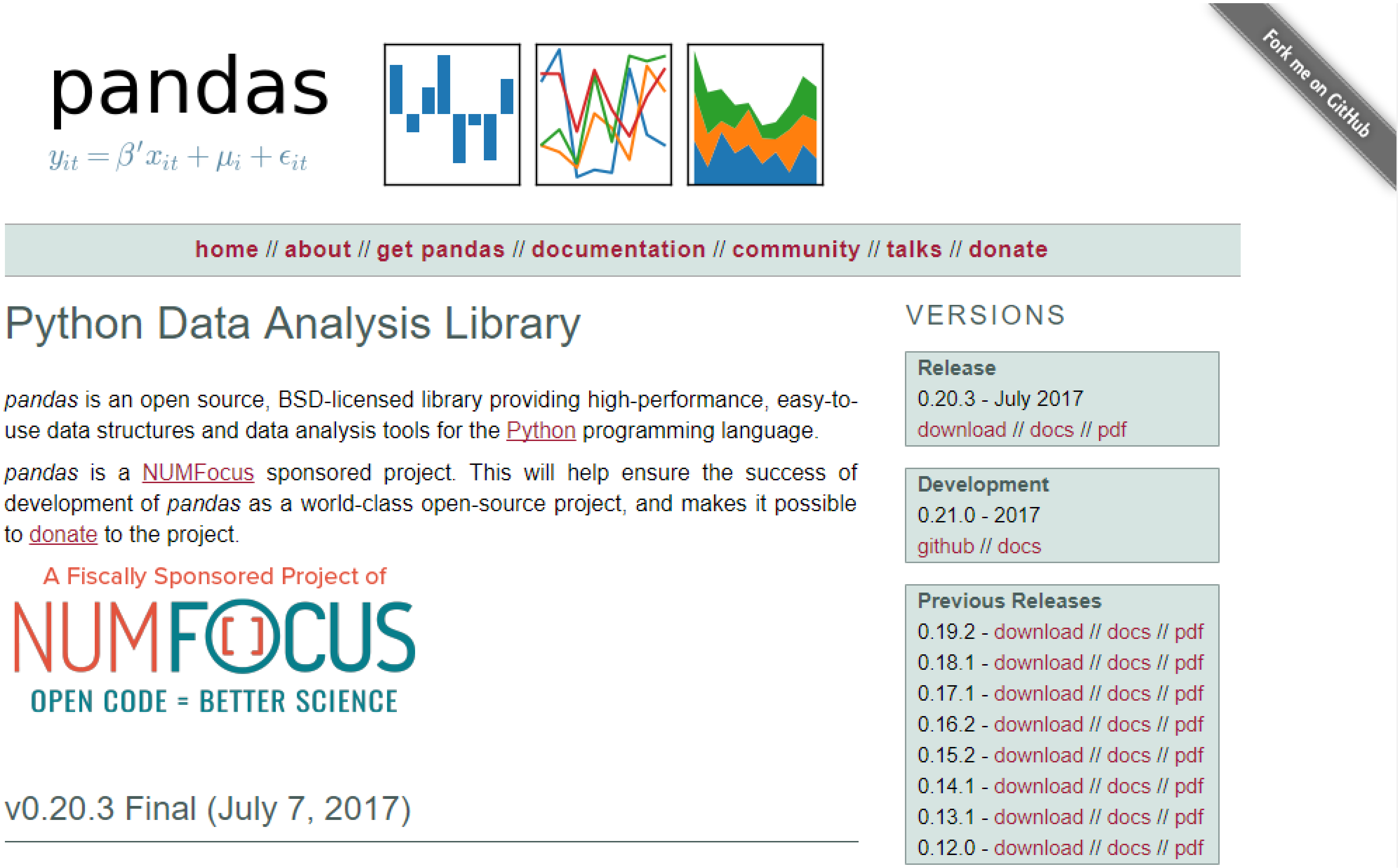
qtconsole
4.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

spyder
3.1.4
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

glueviz
0.9.1
Multidimensional data visualization across files. Explore relationships within and among related datasets.

orange3
3.4.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

rstudio
1.0.136
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.



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

home // about // get pandas // documentation // community // talks // donate

Python Data Analysis Library

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

pandas is a NUMFocus sponsored project. This will help ensure the success of development of *pandas* as a world-class open-source project, and makes it possible to donate to the project.

A Fiscally Sponsored Project of
NUMFOCUS
OPEN CODE = BETTER SCIENCE

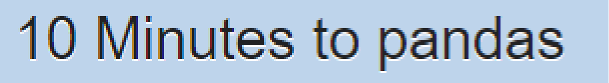
v0.20.3 Final (July 7, 2017)

VERSIONS

Release
0.20.3 - July 2017
download // docs // pdf

Development
0.21.0 - 2017
github // docs

Previous Releases
0.19.2 - download // docs // pdf
0.18.1 - download // docs // pdf
0.17.1 - download // docs // pdf
0.16.2 - download // docs // pdf
0.15.2 - download // docs // pdf
0.14.1 - download // docs // pdf
0.13.1 - download // docs // pdf
0.12.0 - download // docs // pdf



10 Minutes to pandas

http://pandas.pydata.org/pandas-docs/stable/10min.html

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
ts = pd.Series(np.random.randn(1000), index=pd.date_range('1/1/2000', periods=1000))
ts = ts.cumsum()
ts.plot()
```

```
df = pd.DataFrame(np.random.randn(1000, 4), index=ts.index, columns=['A', 'B', 'C', 'D'])
ts = ts.cumsum()
plt.figure();
df.plot();
plt.legend(loc='best')
```

```
from pandas import DataFrame, read_csv
import pandas as pd

data = r‘temp.log'
df = pd.read_csv(data)
df.plot()
```

‘temp.log’

https://1drv.ms/u/s!AtwGHpDGDtFazWsFQy2Qn3lVWtqn https://1drv.ms/u/s!AtwGHpDGDtFazWcRD3IvLkhla18v

temperature.csv

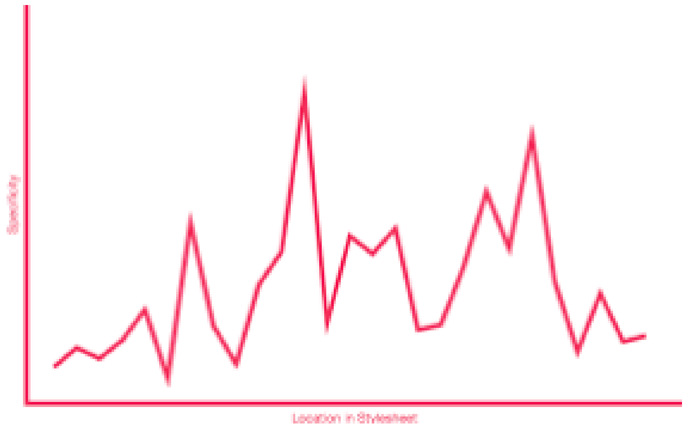
animation example code: simple_anim.py

https://matplotlib.org/examples/animation/simple_anim.html

```
"""
A simple example of an animated plot
"""
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation

fig, ax = plt.subplots()

x = np.arange(0, 2*np.pi, 0.01)
line, = ax.plot(x, np.sin(x))


def animate(i):
    line.set_ydata(np.sin(x + i/10.0))  # update the data
    return line,


# Init only required for blitting to give a clean slate.
def init():
    line.set_ydata(np.ma.array(x, mask=True))
    return line,

ani = animation.FuncAnimation(fig, animate, np.arange(1, 200), init_func=init,
                              interval=25, blit=True)
plt.show()
```

Connected Temperature Monitor Station

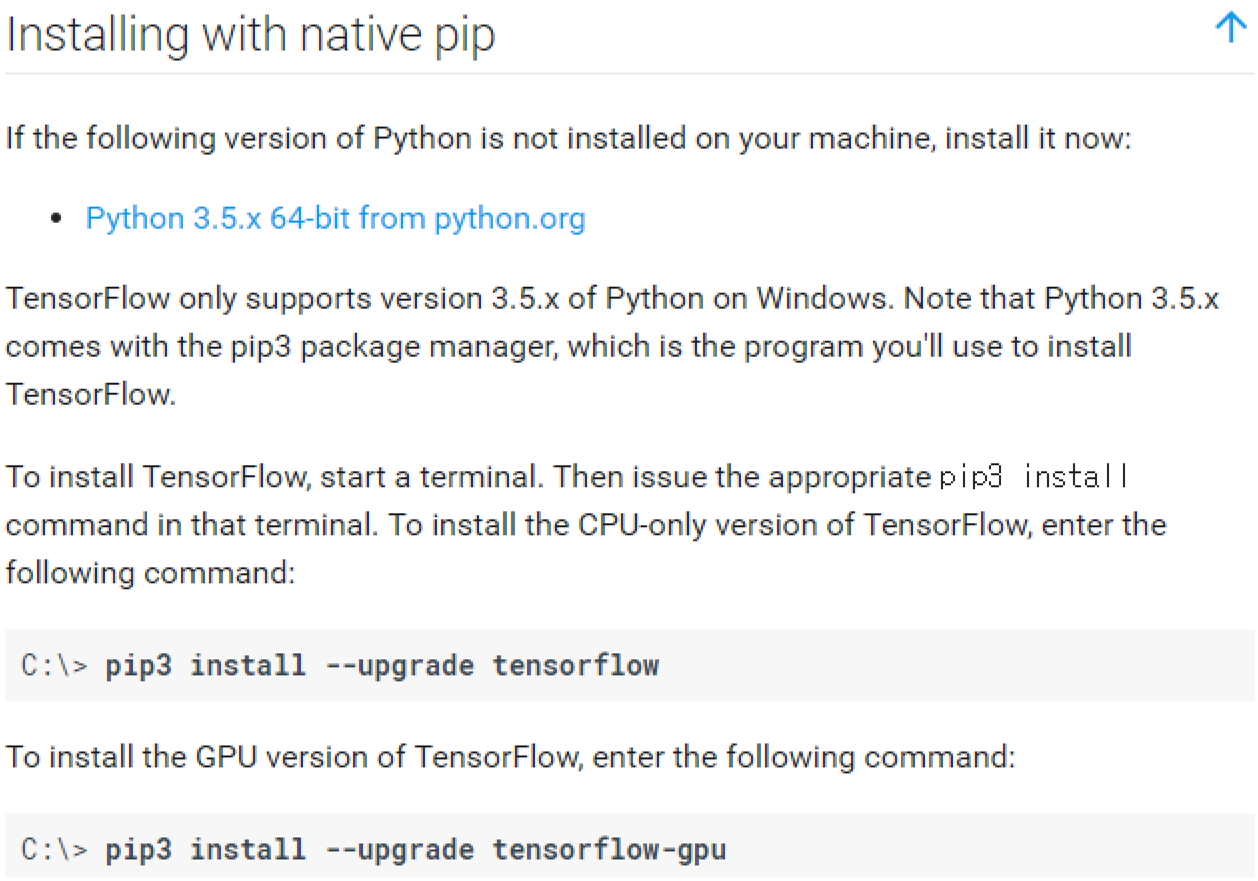
https://github.com/szazo/DHT11_Python



TensorFlow™ Install

Create tensorflow@anaconda
C> Activate tensorflow

C> pip install –upgrade tensorflow



Installing with native pip

If the following version of Python is not installed on your machine, install it now:

- Python 3.5.x 64-bit from python.org

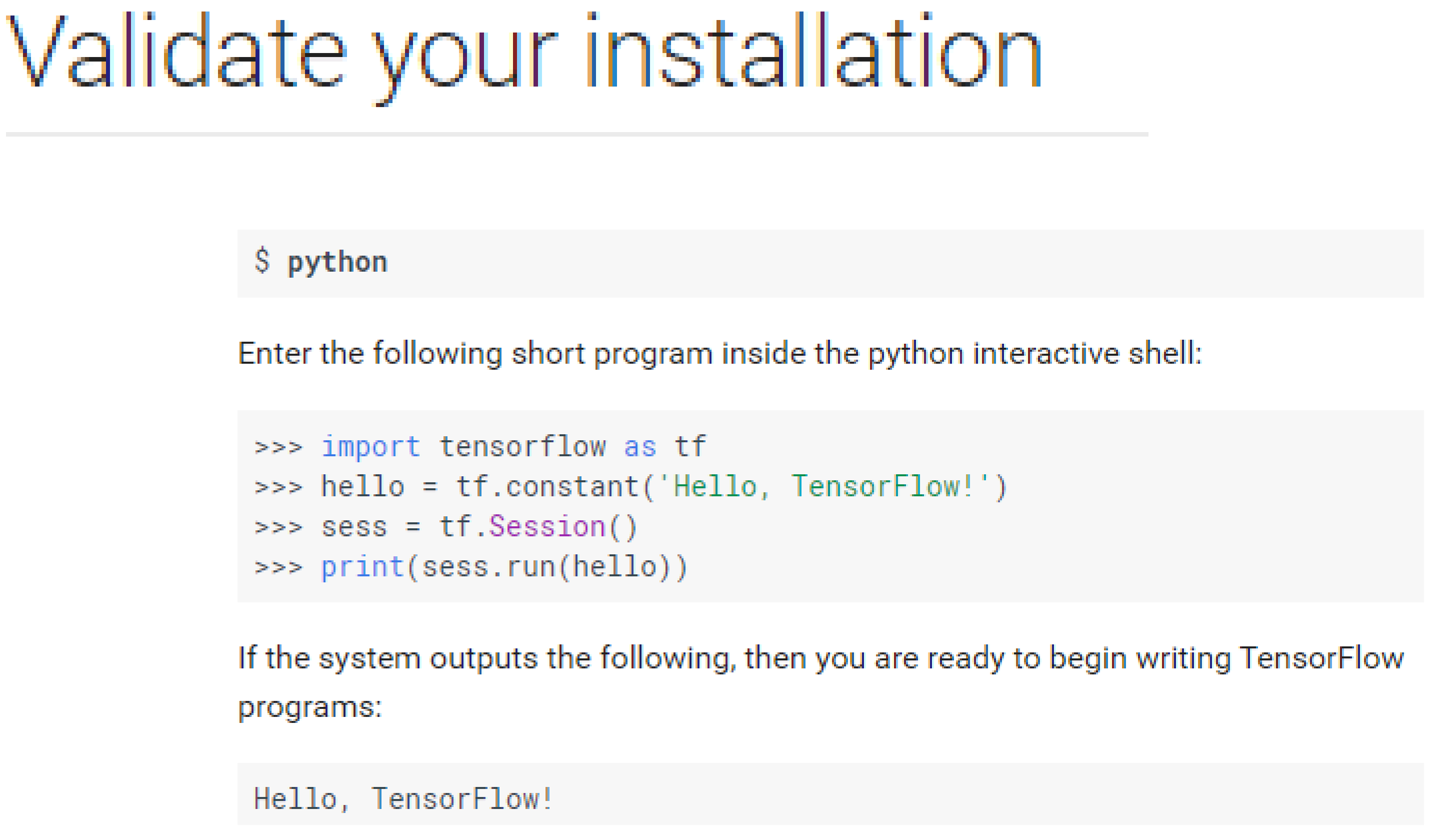
TensorFlow only supports version 3.5.x of Python on Windows. Note that Python 3.5.x comes with the pip3 package manager, which is the program you'll use to install TensorFlow.

To install TensorFlow, start a terminal. Then issue the appropriate `pip3 install` command in that terminal. To install the CPU-only version of TensorFlow, enter the following command:

```
C:\> pip3 install --upgrade tensorflow
```

To install the GPU version of TensorFlow, enter the following command:

```
C:\> pip3 install --upgrade tensorflow-gpu
```



Validate your installation

```
$ python
```

Enter the following short program inside the python interactive shell:

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))
```

If the system outputs the following, then you are ready to begin writing TensorFlow programs:

```
Hello, TensorFlow!
```

Learn Machine Learning

http://hunkim.github.io/ml/

https://github.com/hunkim/DeepLearningZeroToAll

https://www.youtube.com/watch?v=BQEhUD2XTaA

Regular Expression

- 특정 패턴을 추출

```
import re
p = re.compile('(₩d{8})')

f=open("x.txt", "r")
d = f.readlines()
f.close()

for k in d:
    k = k.replace('₩n', '')
    m = p.search(k)
    if (m):
        print(m.group(1))
    else:
        print(k)

f.close()
```

```
Abklekfh49874433alkdjhfieeer
Dkfle8rldfj98464387oeirfjlaskdfa
Laskdflasdkf;a3457823sldkdkfja;s
owdjf23987465halksjdfnas;df
As,kdfhias;odjfhalslskdf28764987
```

Regular Expression

- 패턴 정의
  - . -- matches any character
  - ^ -- matches the start of the string
  - $ -- matches the end of string
  - * -- 0 or more repetitions
  - + -- 1 or more repetitions
  - ? -- 0 or 1 repetitions
  - {m} – m copies
  - [] – individual listing   [ake] a or k or e
  - ₩d – matches a decimal digit
  - ₩w – matches an alpha-numeric ([a-zA-Z0-9])
  - ₩s – matches a space character

- Online Regular Expression Tester
  - https://regex101.com/

기상청 데이터 가져오기

- http://www.kma.go.kr/wid/queryDFSRSS.jsp?zone=1144063000

```
import datetime, time
import urllib.request
import re

f = urllib.request.urlopen('http://www.kma.go.kr/wid/queryDFSRSS.jsp?zone=1144060000')
data = str(f.read())

pat = re.compile(r"<tm>(₩d*)<₩/tm>")  #<tm>숫자들</tm> 형태의 패턴을 찾는다
                                            # (  )로 둘러싼 부분을 추출해준다
m = pat.search(data)
if m:
    print(m.group(0)) # (0)는 매칭 성공한 라인 전체 (1)~ 부터는 ( )로 둘러싼 부분에 매칭된 항목들
    dd = m.group(1)
    d2 = dd    #A
else:
    dd = "no data"

pat = re.compile(r"<hour>(₩d*)<₩/hour>₩₩n₩s*<day>(₩d*)<₩/day>₩₩n₩s*<temp>([-₩.₩d]*)<₩/temp>")
m = pat.search(data)
if not m:
    print("no data")
else:
    print(m.group(0))
    hh = m.group(1) + ":00"
    temp = m.group(3)

    print("예보시간은 ", d2)
    print(hh, "시의 예측온도는 ", temp)
```