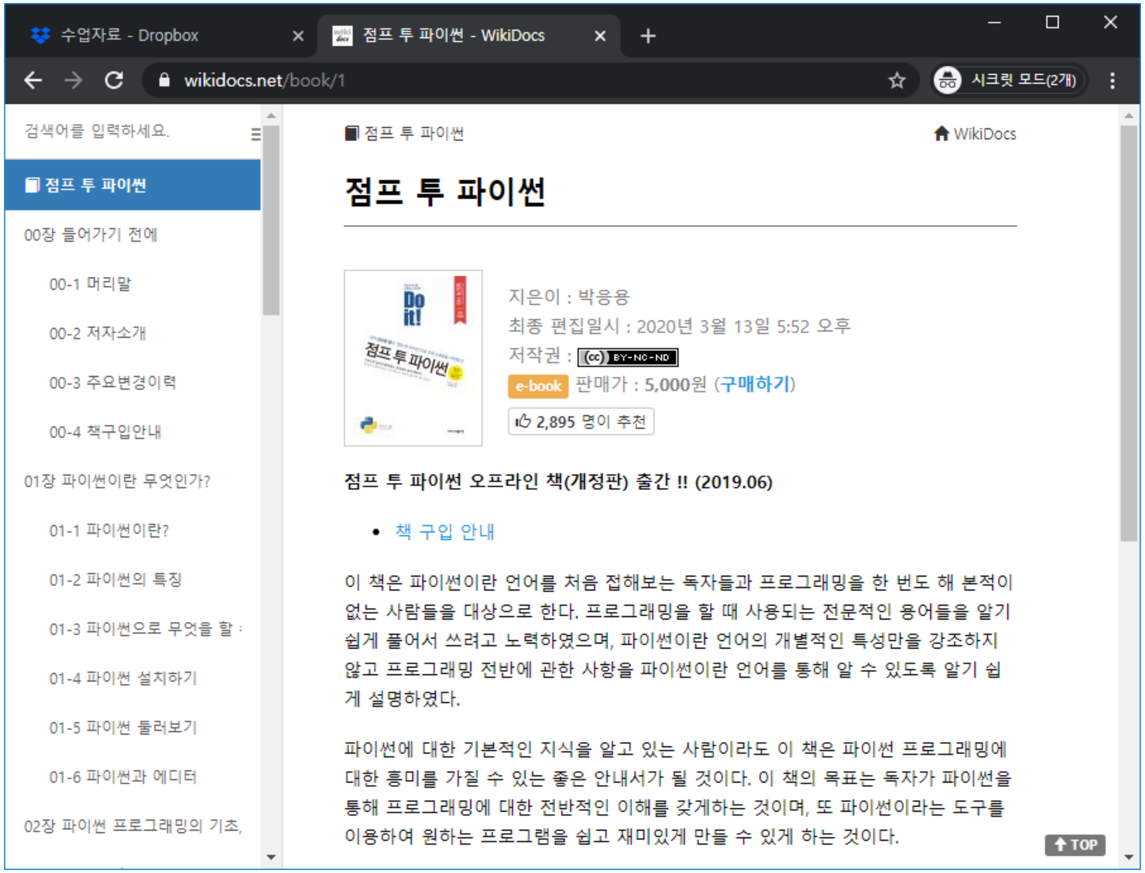
Financial Bigdata and Python

3. NumPy, Matplotlib

Mathematics, Jeonggyu Huh

Jump to Python

https://wikidocs.net/book/1



점프 투 파이썬

00장 들어가기 전에

00-1 머리말

00-2 저자소개

00-3 주요변경이력

00-4 책구입안내

01장 파이썬이란 무엇인가?

01-1 파이썬이란?

01-2 파이썬의 특징

01-3 파이썬으로 무엇을 할 :

01-4 파이썬 설치하기

01-5 파이썬 둘러보기

01-6 파이썬과 에디터

02장 파이썬 프로그래밍의 기초,

점프 투 파이썬

점프 투 파이썬

지은이 : 박응용

최종 편집일시 : 2020년 3월 13일 5:52 오후

저작권 : (cc) BY-NC-ND

e-book 판매가 : 5,000원 (구매하기)

2,895 명이 추천

점프 투 파이썬 오프라인 책(개정판) 출간 !! (2019.06)

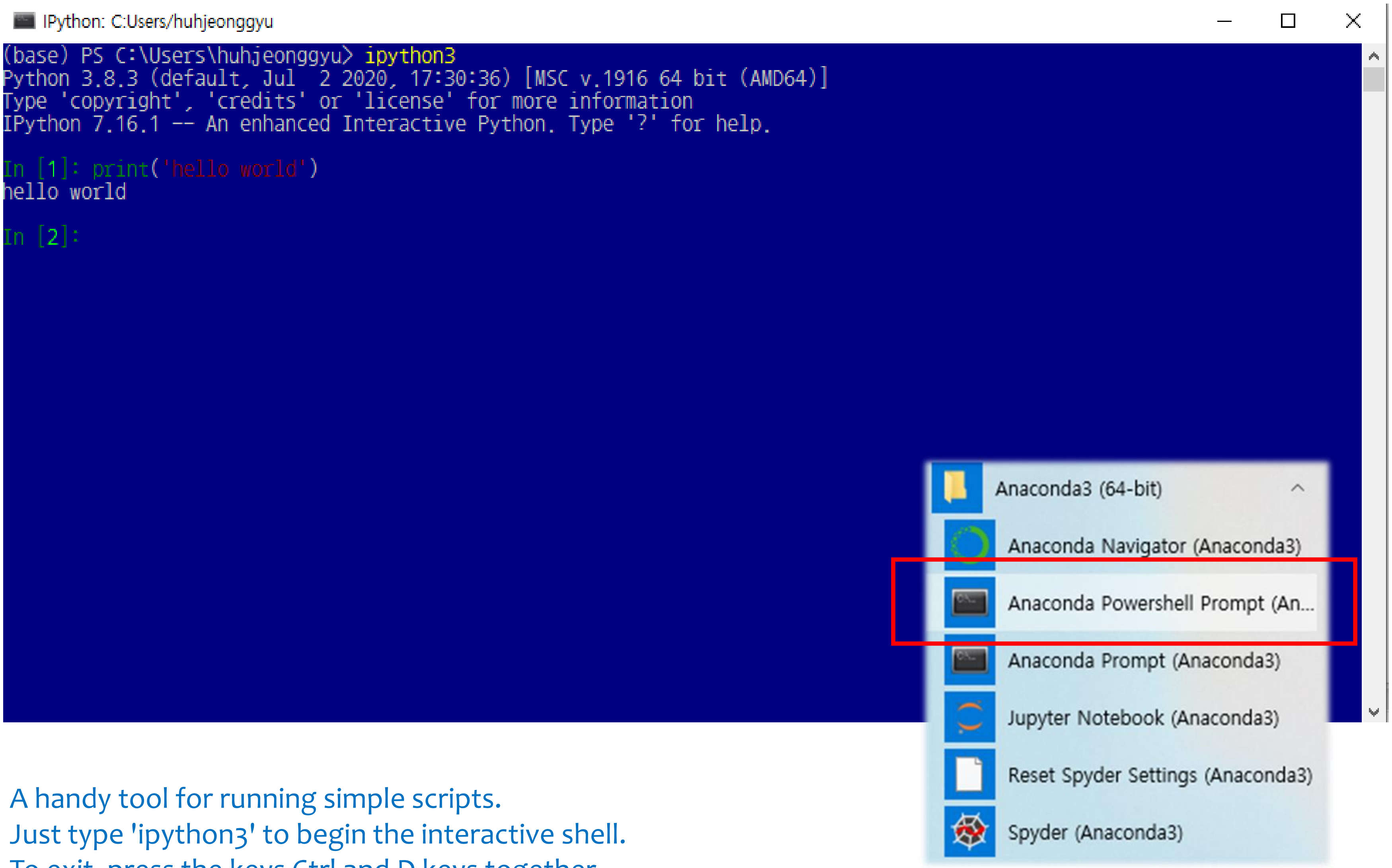
- 책 구입 안내

이 책은 파이썬이란 언어를 처음 접해보는 독자들과 프로그래밍을 한 번도 해 본적이 없는 사람들을 대상으로 한다. 프로그래밍을 할 때 사용되는 전문적인 용어들을 알기 쉽게 풀어서 쓰려고 노력하였으며, 파이썬이란 언어의 개별적인 특성만을 강조하지 않고 프로그래밍 전반에 관한 사항을 파이썬이란 언어를 통해 알 수 있도록 알기 쉽게 설명하였다.

파이썬에 대한 기본적인 지식을 알고 있는 사람이라도 이 책은 파이썬 프로그래밍에 대한 흥미를 가질 수 있는 좋은 안내서가 될 것이다. 이 책의 목표는 독자가 파이썬을 통해 프로그래밍에 대한 전반적인 이해를 갖게하는 것이며, 또 파이썬이라는 도구를 이용하여 원하는 프로그램을 쉽고 재미있게 만들 수 있게 하는 것이다.

Pahk Eungyong (pahkey@gmail.com)

An introduction to Python published on Wikidocs (https://wikidocs.net)



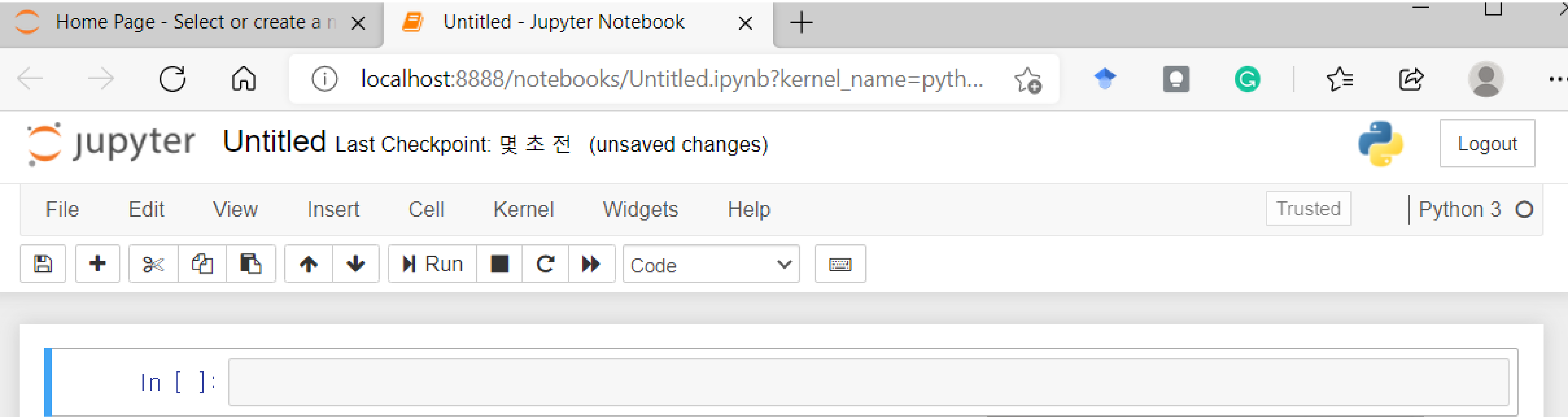
IPython: C:Users/huhjeonggyu

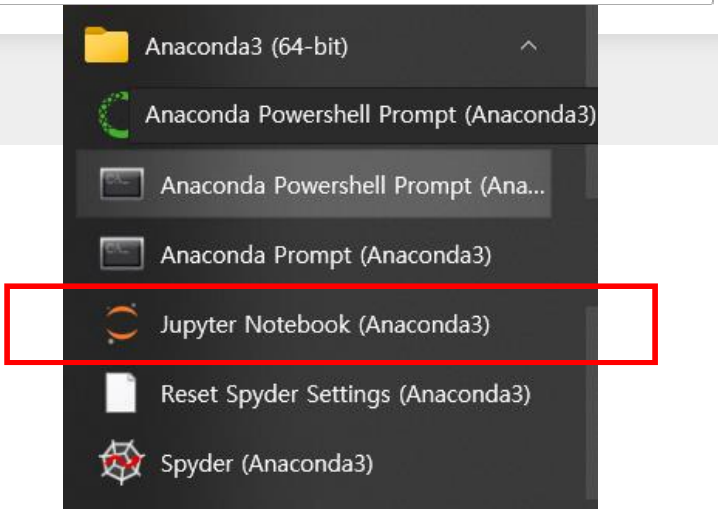
```
(base) PS C:\Users\huhjeonggyu> ipython3
Python 3.8.3 (default, Jul  2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.16.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print('hello world')
hello world

In [2]:
```

Anaconda3 (64-bit)
Anaconda Navigator (Anaconda3)
Anaconda Powershell Prompt (An...
Anaconda Prompt (Anaconda3)
Jupyter Notebook (Anaconda3)
Reset Spyder Settings (Anaconda3)
Spyder (Anaconda3)

A handy tool for running simple scripts.
Just type 'ipython3' to begin the interactive shell.
To exit, press the keys Ctrl and D keys together.

Anaconda Powershell


Home Page - Select or create a n
Untitled - Jupyter Notebook
localhost:8888/notebooks/Untitled.ipynb?kernel_name=pyth...
jupyter Untitled Last Checkpoint: 몇 초 전 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
In [ ]:


Anaconda3 (64-bit)
Anaconda Powershell Prompt (Anaconda3)
Anaconda Powershell Prompt (Ana...
Anaconda Prompt (Anaconda3)
Jupyter Notebook (Anaconda3)
Reset Spyder Settings (Anaconda3)
Spyder (Anaconda3)

Run Jupyter Notebook

If the script you want to run isn't in one sentence,
you're better off using a Jupyter Notebook rather than Anaconda PowerShell.

Anaconda PowerShell	Description
abs(3)	returns the absolute value of an input
abs(-3)	
abs(-1.2)	
	returns the power of an input
pow(2, 4)	pow(a,b) → a^b
pow(3, 3)	
round(4.6)	rounding function
round(4.2)	
round(5.678, 2)	specifies the number of decimal places to be rounded off.
max([1, 2, 3])	returns the maximum of the list
min([1, 2, 3])	returns the minimum of the list
sum([1, 2, 3])	returns the sum of the list
sorted([3, 1, 2])	sorts the input list
sorted(['a', 'c', 'b'])	

Exercise 1)

Let's find the sum, the maximum and minimum values of the following list.
[-8, 2, 7, 5, -3, 5, 0, 1]

Exercise 2)

Here is the result when 17/3 is output in Python:

```
>>> 17 / 3
5.666666666666667
```

Let's display the result 5.666666666666667 rounded up to 4 decimal places.

Anaconda PowerShell	Description
import numpy as np	imports the package NumPy 'np' is usually used as the abbreviation of NumPy.

a = np.array([1,2,3,4])
b = np.array([5,6,7,8])
	create arrays a and b of size 4

c = a+b
d = a-b
e = a*b
f = a/b
	Arithmetic operations between two matrices a and b of the same size (calculated elementwise between elements in the same position)

g = a+2
h = a-2
i= 2*a
j =a/2
	Arithmetic operations between an array and a scalar
	(Such an operation applies to all elements of the matrix)

How to implement matrix addition in python without NumPy?
 	
c = []
for e_a, e_b in zip(a,b) :
    e_c = e_a + e_b
    c.append(e_c)
	Very inefficient way.
	In other words, it's really slow.
	
	So if it is not for study or research purposes,
	do not reinvent the wheel!

Anaconda PowerShell	Description
A = np.array([[1,1,1],[1,1,1],[1,1,1],[1,1,1]])	creates a matrix of size 4x3
A = np.ones([4,3])	creates a matrix of size 4x3 where all elements are 1.
z = np.array([5,6])	creates a matrix of size 2
A+z	An error occurs because the matrices have different sizes.
a = np.array([1,2,3]) A+a	It is expected that an error occurs because the size of the matrix is different, but this code runs well in the way of sequentially adding the elements of 'a' to each row of 'A'.
	Even if it's annoying that a code doesn't run because of an error, it's not severe. However, it is rather dangerous if it is not performing as expected, but there is no error.

- Useful array creation functions to know (array = matrix + vector + tensor)

zeros([n,m]): creates a matrix of size n×m where all elements are 0
ones([n,m]): creates a matrix of size n×m where all elements are 1.
arange(a,b): similar to the range function.
    ex) arange(0,5) → array([0,1,2,3,4])
linspace(a,b,n): creates a matrix where n evenly spaced numbers from 'a' to 'b'.
    ex) linspace(0,2,5) → array([0,0.5,1,1.5,2])

Anaconda PowerShell	Description
a = np.array([1,2,3,4])	
a.shape	the shape of a : (4,)
a.ndim	the dimension of a : 1
b = np.array([[1,2,3,4],[5,6,7,8]])	
b.shape	the shape of b : (2,4)
b.ndim	the dimension of b : 2
c = np.array([[1,2],[3,4]])	
c.shape	the shape of c : (2,2)
c.ndim	the dimension of c : 2
d = a.reshape([2,2])	changes the shape of a from (4,) to (2,2)
e = b.reshape(8)	changes the shape of b from (2,4) to (8,)
f = b.reshape(-1)	changes the shape of b from (2,4) to (8,)
g = a.reshape([2,-1])	changes the shape of a from (4,) to (2,2)
h = a.reshape([-1,4])	changes the shape of a from (4,) to (1,4)
np.vstack([a,b])	stacks matrices with the same number of columns top and bottom
np.hstack([b,c])	stacks a matrix with the same number of rows on both sides

Anaconda PowerShell	Description
a>2	the logical matrix generated by whether the element of 'a' is greater than 2
a[a>2]	the elements of 'a' greater than 2
b>3	
b[b>3]	
a[0]	accesses the 0th element of a
a[:3]	accesses 0-2 elements of a
a[-1]	accesses the last element of a
b[0,0]	accesses the 0th row, the 0th column element of b
b[0,:2]	accesses the 0th row, the 0-1 column of b
b[:,1:3]	accesses all the rows, the 1-2 columns of b
x = np.linspace(0,1,5)	creates a vector by dividing [0,1] into 4 equal parts
y = np.linspace(1,3,6)	creates a vector by dividing [1,3] into 5 equal parts
X, Y = np.meshgrid(x,y)	creates a 2D mesh based on x,y vectors
X	X is the x-coordinate of the mesh
Y	Y is the y-coordinate of the mesh

Anaconda PowerShell	Description
a = np.array([1,2,3,4])	
len(a)	the number of elements in a
np.sum(a)	the sum of all elements of a
np.mean(a)	the average value of a
np.var(a)	the variance of a
np.std(a)	the deviation value of a
np.max(a)	the maximum value of a
np.min(a)	the minimum value of a
np.argmax(a)	the maximum position of a
np.argmin(a)	the position of the minimum value of a
np.median(a)	the median of a
np.percentile(a,percentile)	the percentile of a. (ex) percentil(a,25) : 1st quartile

There are so many NumPy functions. It is impossible and undesirable to cover them all in this class.
(Like not memorizing an English dictionary during an English class..).

NumPy official user guide:
https://numpy.org/devdocs/user/index.html

Data Science School (by Kim Do-hyung) :
https://datascienceschool.net

Exercise 3)

Solve the problems for the following matrix.

```
m = np.array([[ 0, 1, 2, 3, 4],
              [ 5, 6, 7, 8, 9],
              [10, 11, 12, 13, 14]])
```

a) Index the value 7.
b) Index the value 14.
c) Slice the array [6, 7].
d) Slice the array [7, 12].
e) Slice the array [[3, 4], [8, 9]].

Exercise 4)

Create the following array using the functions introduced so far.

```
array([[   0.,    0.,    0.,   1.,     1.],
       [   0.,    0.,    0.,   1.,     1.],
       [   0.,    0.,    0.,   1.,     1.],
       [  10.,   20.,   30.,  40.,    50.],
       [  60.,   70.,   80.,  90.,   100.],
       [ 110.,  120.,  130.,  140.,  150.],
       [   0.,    0.,    0.,    1.,    1.],
       [   0.,    0.,    0.,    1.,    1.],
       [   0.,    0.,    0.,    1.,    1.],
       [  10.,   20.,   30.,  40.,    50.],
       [  60.,   70.,   80.,  90.,    100.],
       [ 110.,  120.,  130.,  140.,  150.]]).
```

Exercise 5)

Solve the problems for the following matrix.

```
x = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
              11, 12, 13, 14, 15, 16, 17, 18, 19, 20])
```

a) Find multiples of 3.
b) Find the numbers that when divided by 4 leaves 1 left.
c) Find the numbers among multiples of 3 that when divided by 4 leaves 1 left.

Anaconda PowerShell	Description
import numpy as np	
a = np.array([1,2,3,4]) a	creates a vector
b = np.array([[1,2,3,4]]) b	creates a row vector (= a matrix with row length 1)
c = np.array([[1],[2],[3],[4]]) c	creates a column vector (= a matrix of column length 1)
a.ndim a.shape b.ndim b.shape c.ndim c.shape	ndim of scalars: 0 ndim of vectors: 1 ndim of matrices (column vector, row vector): 2 shape of scalars : () shape of vectors: (n,) shape of matrices (column vector, row vector): (n,m)

Anaconda PowerShell	Description
	* Method 1
a.reshape([4,1])	converts 'vector' to 'column vector' using the reshape function
a.reshape([1,4])	converts 'vector' to 'row vector'
b.reshape(4)	converts 'row vector' to 'vector'
b.reshape([4,1])	converts 'row vector' to 'column vector'
c.reshape(4)	converts 'column vector' to 'vector'
c.reshape([1,4])	converts 'column vector' to 'row vector'
	* Method 2
a[:,np.newaxis]	converts 'column vector' to 'vector' by adding axis 1
a[np.newaxis,:]	converts 'row vector' to 'vector' by adding axis 0
b.flatten()	converts 'row vector' to 'vector' using flatten function
b.transpose()	converts 'row vector' to 'column vecotor' by transposing it
c.flatten()	converts 'column vector' to 'vector'
c.transpose()	converts 'column vector' to 'row vector'

Anaconda PowerShell	Description

sa = a.sum()
sa
type(sa)
sa.ndim, sa.shape
	returns the sum of all elements of 'vector'
	
	Note that the return value is of type 'np.int', not of type 'int'.
	So you can still get the ndim and shape for the variable.
	Also notice that ndim is reduced from 1 to 0.

sa = a.sum(keepdims=True)
sa
type(sa)
sa.ndim, sa.shape
	sums all elements while preserving dimension.
	'a' is a vector of the shape (4,), so 'sa' has the shape (1,)
	(Note that () ≠ (1,))

sb = b.sum() sb sb.ndim, sb.shape	
sb = b.sum(axis=1) sb sb.ndim, sb.shape	sums all elements along the 1st (specific) axis 'b' has the shape (1,4), so 'sb' has the shape (1,) (the 1st axis has disappeared and the 0th axis has been left.) Meanwhile, ndim is reduced from 2 to 1.
sb = b.sum(axis=1,keepdims=True) sb sb.ndim, sb.shape	sums all elements along the 1st axis while preserving dimension In that case, the shape of 'sb' will be (1,1), not rather (1,). So ndim has been also kept at 2.
sb = b.sum(axis=0) sb sb.ndim, sb.shape	For a 'column vector' like 'b', summing the elements along the 0th axis changes nothing other than changing 'ndim'.

Anaconda PowerShell	Description
A = np.array([[1,2,3],[4,5,6]])	matrix of the shape (2,3)
A	
A.ndim, A.shape	
A.flatten()	flattens the matrix by row-major order
A.transpose()	swaps the 0th axis and 1st axis
A[np.newaxis,:,:].shape	A new axis is added as the 0th axis.
A[:,np.newaxis,:].shape	A new axis is added as the 1st axis.
A[:,:,np.newaxis].shape	A new axis is added as the 2nd axis.
sb = A.sum(axis=0)	sums all elements along the 0th axis
sb	
sb.ndim, sb.shape	
sb = A.sum(axis=1)	sums all elements along the 1st axis
sb	
sb.ndim, sb.shape	
sb = A.sum(axis=1,keepdims=True)	sums all elements along the 1st axis while preserving dimension.
sb	
sb.ndim, sb.shape	

Anaconda PowerShell	Description
a = np.array([1,2,3,4])	
a.shape	shape: (4,)
a.ndim	ndim: 1
b = np.array([[1,2,3,4],[5,6,7,8]])	
b.shape	shape: (2,4)
b.ndim	ndim: 2
c = np.array([[1,2],[3,4]])	
c.shape	shape: (2,2)
c.ndim	ndim: 2
np.vstack([a,b])	stacks matrices with the same number of columns top and bottom
np.hstack([b,c])	stacks matrices with the same number of rows on both sides
np.concatenate([a,b],axis=0)	a: (4,) / b: (2,4)
	: stacking the matrices on axis 0
	(An error occur because the dimensions of a and b are different)
a2 = a[np.newaxis,:]	a: (1,4) / b: (2,4) (= np.hstack([a,b]))
np.concatenate([a2,b],axis=0)	: stacking the matrices on axis 0
np.concatenate([b,c],axis=1)	b: (2,4) / c: (2,2) (= np.vstack([b,c]))
	: stacking the matrices on axis 1

Anaconda PowerShell	Description

x = np.arange(4)
print('x',x.shape); print(x)
	[0,1,2,3] (4,)

xx = x.reshape(4,1)
print('xx',xx.shape); print(xx)
	[[0], (4,1)
	[1],
	[2],
	[3]]

y = np.ones(5)
print('y',y.shape); print(y)
	[1,1,1,1,1] (5,)

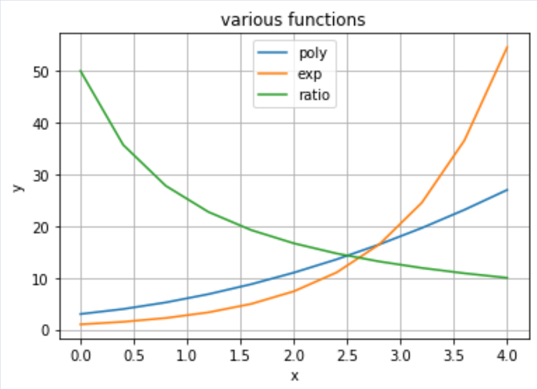
z = np.ones((3,4))
print('z',z.shape); print(z)
	[[1,1,1,1], (3,4)
	[1,1,1,1],
	[1,1,1,1]]

x+y	(4,) + (5,) Error!
	Broadcasting is applied to below examples
xx+y (xx+y).shape	(4,1) + (5,) = (4,1) + (1,5) = (4,5) + (4,5) [[0], + [1,1,1,1,1] = [[0], + [[1,1,1,1,1]] = [[0,0,0,0,0], + [[1,1,1,1,1], [1], [1], [1,1,1,1,1], [1,1,1,1,1], [2], [2], [2,2,2,2,2], [1,1,1,1,1], [3]] [3]] [3,3,3,3,3]] [1,1,1,1,1]]
x+z (x+z).shape	(4,) + (3,4) = (1,4) + (3,4) = (3,4) + (3,4) [0,1,2,3] + [[1,1,1,1], = [[0,1,2,3]] + [[1,1,1,1], = [[0,1,2,3], + [[1,1,1,1], [1,1,1,1], [1,1,1,1], [0,1,2,3], [1,1,1,1], [1,1,1,1]] [1,1,1,1]] [0,1,2,3]] [1,1,1,1]]

Exercise 6) Dimension reduction operations (sum, max, etc.)

Create a 5 x 6 matrix composed of real numbers and obtain the following values for the matrix.

1) The maximum value of the whole
2) Sum of each row
3) Maximum value in each row
4) Average of each column
5) Minimum in each column

Anaconda PowerShell	Description
import numpy as np	MatplotLib : popular package for drawing using Python
import matplotlib.pylab as plt	pylab : introduces MATLAB interface to MatplotLib
xarr = np.linspace(0,4,11)	
yarr1 = xarr**2 + 2*xarr + 3	
yarr2 = np.exp(xarr)	
yarr3 = 50/(1+xarr)	
plt.figure(figsize=(6,4))	creates a figure window with size (6,4)
plt.plot(xarr,yarr1)	plots the graph of (xarr,yarr1)
plt.plot(xarr,yarr2)	plots the graph of (xarr,yarr2)
plt.plot(xarr,yarr3)	plots the graph of (xarr,yarr3)
plt.xlabel('x')	x-axis labeling
plt.ylabel('y')	y-axis labeling
plt.title('various functions')	titles the graphs
plt.legend(['poly','exp','ratio'])	makes legends
plt.grid()	makes the grid appear
plt.show()	shows the graphs
	

Anaconda PowerShell

```
plt.figure(figsize=(6,4))

plt.plot(xarr,yarr1,'-o')
plt.plot(xarr,yarr2,'--X',ms=10)
plt.plot(xarr,yarr3,':D',lw=2)

plt.xlabel('x')
plt.ylabel('y')
plt.title('various functions')
plt.legend(['poly','exp','ratio'])
plt.grid()
plt.show()
```

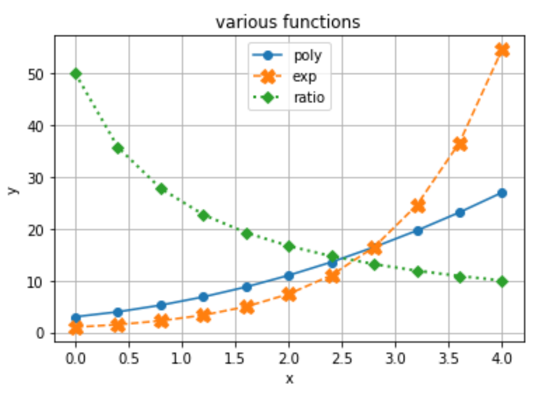
※ line style

-	solid line
--	dashed line
-.	dash-dot line
:	dotted lined

Description

※ plot function option

Option	Abbreviation	Description
color	c	line color
linewidth	lw	line width
linestyle	ls	line style
marker		marker type
markersize	ms	marker size
markeredgecolor	mec	marker line color
markeredgewidth	mew	marker line width
markerfacecolor	mfc	marker inner color



※ color http://Matplotlib.org/examples/color/named_colors.html

color	abbreviation
blue	b
green	g
red	r
cyan	c
magenta	m
yellow	y
black	k
white	w

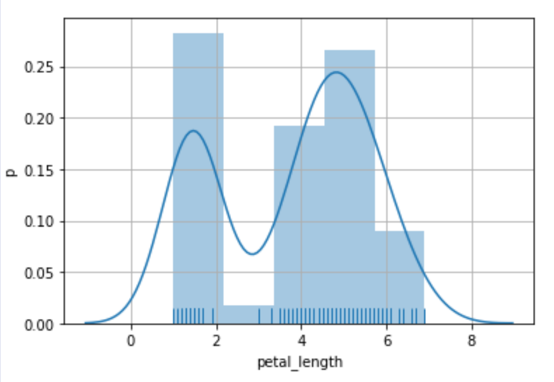
※ marker https://matplotlib.org/3.2.1/api/markers_api.html

marker type	
o	circle
^	triangle
s	square
P	cross
*	star
X	X
D	diamond
$...$ (ex. f)	desired character

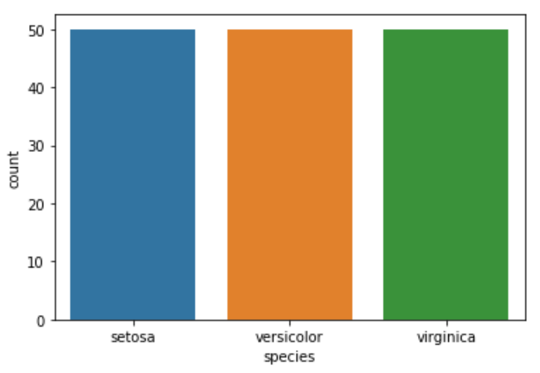
Anaconda PowerShell	Description

```
import matplotlib.pylab as plt
import seaborn as sns

iris = sns.load_dataset('iris')
iris


fig=plt.figure(figsize=(6,4))
sns.displot(data=iris,
            x='petal_length',
            kde=True, rug=True)
plt.ylabel('p')
plt.grid()
plt.show()

fig=plt.figure(figsize=(6,4))
sns.countplot(data=iris,
              x='species')
plt.show()
```

seaborn: an extension of MatplotLib for statistical visualization



displot() function is useful for drawing histogram, density, and rugs.



countplot() function visualizes how many data each class has