

TP 3

Numpy

Objective : Use the numpy library

1 Matrix creation

Exercise 1 : 1D Matrix

```
import numpy as np

A = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
A2 = np.array([[1, 2, 3, 4, 5, 6, 7, 8, 9]])
B = np.array(range(10))
C = np.linspace(0, 10, 10)
D = np.arange(0, 10, 1)
```

- For each of the following expression (A to D), print the value of the numpy array and its shape (using `.shape()`).
 - What does the functions **np.array**, **np.linspace**, **np.arange** do ?
 - What the difference between A and A2 ? Why ?
 - How to change the expression of C, in order to have C equal to D ?
- Use one of the previous functions to create :
 - The array of the integer between 1 and 20 (excluded)
 - A array containg 20 times the integer 3.
 - A array containing multiples of 3 in decreasing order from 30 to 9 (both included)

Exercise 2 : 2D Matrix

```
A = np.eye(3)
B = np.zeros((3, 3))
C = np.ones((3, 3))
D = np.array(range(9)).reshape(3, 3)
E = np.diag([1, 1, 1])
F = np.array([[0, 1, 2], [3, 4, 5], [6, 7, 9]])
G = np.ones((3, 3)) - np.eye(3)
H = np.diag(D)
I = np.diag(np.diag(D))
```

- For each of the following expression try to predict the value and the shape of the resulting array, before printing it
- Which of the arrays are equal ?
- What do the functions `np.ones`, `np.eye`, `np.diag`, `np.zeros` and `np.reshape` do ?

Exercise 3 : Random Matrix

- For each of the following functions, try to understand what the functions does. What arguments do you pass to theses functions ? Print the result.
 - `np.random.randn`
 - `np.random.randint`
 - `np.random.rand`
- Create the following array :
 - A random array of shape (5, 3, 2) with gaussian distribution of average value 10 and standard deviation 5
 - A random array of shape (3, 1) with a uniform distribution between -1 and 1
 - A random array of shape (100) with random integers between 0 and 100 (excluded).

2 Operations on matrix

Exercise 4 : Comparison with list

- Create the list `L = [1, 2, 3]` and the array `A = np.array([1, 2, 3])`
- On both L and A, apply the following operation :

– <code>L + L</code>	– <code>A + A</code>
– <code>L + 1</code>	– <code>A + 1</code>
– <code>3*L</code>	– <code>3*A</code>
– <code>L*L</code>	– <code>A*A</code>
– <code>L**L</code>	– <code>A**A</code>
– <code>L/L</code>	– <code>A/A</code>
- Which operation is working on A ? On L ? What are the differences between the operation on list and array ?

3 Matrix manipulation

Exercise 5 : Indexing

- Create the array : $A = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix}$
- Using **only** indexing of the array A, create the following array :

– $u = (0 \ 1 \ 2)$	– $D = \begin{pmatrix} 0 & 2 \\ 3 & 5 \\ 6 & 8 \end{pmatrix}$
– $v = (1 \ 4 \ 7)$	
– $w = (0 \ 8)$	– $E = \begin{pmatrix} 3 & 5 \\ 0 & 2 \end{pmatrix}$
– $x = (2 \ 4 \ 6)$	
– $C = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{pmatrix}$	– $F = \begin{pmatrix} 8 & 7 & 6 \\ 5 & 4 & 3 \\ 2 & 1 & 0 \end{pmatrix}$

Exercise 6 : Multiplication

- Create the following matrix : $A = \begin{pmatrix} 7 & 0 \\ 3 & 2 \\ 3 & -4 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 2 \\ 0 & -3 \end{pmatrix}$, $C = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $D = (5 \ -3)$,
- Using the function `np.dot` and the transposition (`.T`) , try to find which of the following matrix multiplication are possible. If not, what is the error message given by Python ? Print the results of the multiplication as the shape.

- $A * B$
- $A^T * B$
- $B * A$
- $B * A^T$
- $C * A$
- $A * C$
- $D * C$
- $D * B * C$
- $A^T * A$
- $A * A^T$

Help : `np.dot`, `np.array`,

Exercise 7 : Boolean indexing

- Create the following `A = np.random.randint(0, 9, (3,3))`
- Without using numpy, find the indexes of the values in A superior to 5.
- Print the result of the following expressions. Look at their shape and type.

- `A > 5`
- `A[A > 5]`
- `np.where(A>5)`
- `A[np.where(A>5)]`

- How to use the previous expression to put to 0 to all the values inferior to 2 and superior to 7 ?

4 Functions

Exercise 8 : Basic functions

- Create the array : $A = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix}$
- Apply each on the following function to the array A and see what the result are :

- `np.log`
- `np.cos`
- `np.exp`
- `np.sin`
- `np.sqrt`

Exercise 9 : Statistics functions

All the following functions have an `axis` property : `np.mean`, `np.max`, `np.min`, `np.sum`, `np.cumsum`, `np.median` and `np.std`. By default they calculate the function on the whole array, but if you specify the `axis`, it will calculate it only following properties only along the axis.

- Create the following array : `A = np.random.randn(5, 6, 6, 4)`
- For each of the following expressions, try to predict the shape of the results before applying the function and check it after.

- `np.mean(A)`
- `np.mean(A, axis=3)`
- `np.mean(A, axis=(0,1,3))`
- `np.mean(A, axis=0)`
- `np.mean(A, axis=(2,3))`

Exercise 9 : Concatenation functions

- Create the array : $A = \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{pmatrix}$
- For each of the following expressions, try to predict the shape of the results before applying the function and check it after.

- `np.concatenate((A, A, A), axis=0)`
- `np.concatenate((A, A, A), axis=1)`
- `np.stack((A, A, A), axis=0)`

- What is the differences between `concatenate` and `stack` ?

5 Réponses

6 Source

- http://perso.numericable.fr/jules.svartz/prepa/IPT_sup/archives_TP_sup/TP3_MPSI.pdf
- http://s15847115.domainepardefaut.fr/moodle/pluginfile.php/1352/mod_resource/content/1/TP_numpy.pdf
- http://www.armelmartin.mon-site-a-moi.fr/doc/info/tp2_numpy_matplotlib.pdf