

# Numpy

Βιβλιοθήκη Python για μαθηματικούς υπολογισμούς

# Numpy - module

- Είναι ένα πακέτο (σύνολο) συναρτήσεων ειδικά σχεδιασμένες για μαθηματικούς υπολογισμούς
- Κατασκευάζεται μια νέα δομή - κλάση αντικειμένων `array` δίνει τη δυνατότητα να ορισθούν διανύσματα και πίνακες
- Αποτελεσματικοί υπολογισμοί και πράξεις γραμμικής άλγεβρας (επίλυση γραμμικών συστημάτων, υπολογισμός ιδιοτιμών και ιδιοδιανυσμάτων)

# Numpy - array

- `array` είναι μια νέα δομή - κλάση που μοιάζει με τη λίστα
- σε αντίθεση με τη λίστα η διάσταση ενός `array` είναι ορισμένη από την αρχή και αποθηκεύει αντικείμενα που είναι μόνο του ίδου τύπου

# Numpy - array

- ```
import numpy as np  
a=np.array([10,20,30])
```

Δημιουργώ ένα αντικείμενο `array` που έχει διάσταση 1. Είναι το διάνυσμα (10,20,30)

# Numpy - array

- Διάσταση  
`a.ndim`  
Επιστρέφει  
`1`
- Μορφή του `array`  
`a.shape`  
Επιστρέφει  
`(3,)`
- Τύπος στοιχείων  
`a.dtype`  
Επιστρέφει  
`dtype('int64')`

# Numpy - arange

- `arange` - Δημιουργώ ένα `array` με αριθμούς όπως η `range` για λίστες
- `np.arange(10)`  
Επιστρέφει  
`array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])`
- ακολουθία αριθμών με βήμα 2  
`np.arange(3, 12, 2)`  
Επιστρέφει  
`array([ 3, 5, 7, 9, 11])`

# Numpy - arange

- `arange` - ακολουθία αριθμών με βήμα δεκαδικό
- `np.arange(1.2, 2.0, 0.1)`  
Επιστρέφει  
`array([ 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9])`
- `linspace` - δοσμένο πλήθος αριθμών  
`np.linspace(11., 12., 5)`  
`array([ 11., 11.25, 11.5, 11.75, 12.])`

# Numpy - zeros, ones

Δημιουργώ array με 0  
`np.zeros(10)`

```
array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.])
```

Δημιουργώ array με 1

```
np.ones(5)
```

```
array([ 1.,  1.,  1.,  1.,  1.])
```



# Πράξεις με array

- Πράξεις: +, -

```
x = np.array([1,2,3])
```

```
y = np.array([4,5,6])
```

```
x+y
```

Επιστρέφει

```
array([ 5, 7, 9])
```

- $x=[1,2,3]$

```
y=[4,5,6]
```

```
x+y
```

Επιστρέφει

```
[ 1, 2, 3, 4, 5, 6]
```

# Πράξεις με array

- Πράξεις: +, -

```
x = np.array([1,2,3])
```

```
y = np.array([4,5,6])
```

$x-y$

Επιστρέφει

```
array([-3, -3, -3])
```

- Στις λίστες όμως

```
x=[1,2,3];y=[4,5,6]
```

$x-y$

Επιστρέφει

Σφάλμα. Δεν γίνεται η πράξη

# Πράξεις με array

- Πράξεις: \*,/,\*\*: Γίνονται στοιχείο με στοιχείο

```
x = np.array([1,2,3])
```

```
y = np.array([4,5,6])
```

`x*y`

Επιστρέφει

```
array([ 1*4, 2*5, 3*6]) δηλ.
```

```
array([ 4, 10, 18])
```

- `y/x`

Επιστρέφει

```
array([ 4/1, 5/2, 6/3]) δηλ.
```

```
array([ 4.0, 2.5, 2.0])
```

`x**y`

Επιστρέφει

```
array([ 1, 32, 729])
```

# Πράξεις array με αριθμό

- Πράξεις: +, -, \*, /, \*\*: Γίνονται με κάθε στοιχείο

```
x = np.array([1,2,3])
```

```
y = np.array([4,5,6])
```

2+x

Επιστρέφει

```
array([ 2+1, 2+2, 2+3]) δηλ.
```

```
array([ 3, 4, 5])
```

- y-2

Επιστρέφει

```
array([ 4-2, 5-2, 6-2]) δηλ.
```

```
array([ 2, 3, 4])
```

2\*x

Επιστρέφει

```
array([ 2, 4, 6])
```

# Πράξεις array με αριθμό

- Πράξεις: +, -, \*, /, \*\*: Γίνονται με κάθε στοιχείο

```
x = np.array([1,2,3])
```

```
y = np.array([4,5,6])
```

```
y/2
```

Επιστρέφει

```
array([ 4/2, 5/2, 6/2]) δηλ.
```

```
array([ 2, 2, 3])
```

- $2/x$

Επιστρέφει

```
array([ 2/1, 2/2, 2/3]) δηλ.
```

```
array([ 2.0, 1.0, 0.66666667])
```

```
x**2
```

Επιστρέφει

```
array([ 1, 4, 9])
```

# array - “κομμάτιασμα”

```
x = np.array([1,2,3,4,5])
```

```
x[3]
```

Επιστρέφει

4

Το “κομμάτιασμα” ακολουθεί τους ίδιους κανόνες με αυτές στις λίστες

```
x[0:4]
```

Επιστρέφει

```
array([1, 2, 3, 4])
```

# πολυδιάστατα array

Πίνακας

|   |   |
|---|---|
| 1 | 2 |
| 3 | 4 |

```
x = np.array([[1.,2.],[3.,4.]])
```

`x.ndim`

Επιστρέφει

2

`x.shape`

Επιστρέφει

(2,2)

# πολυδιάστατα array

```
x = np.zeros((2,3))
```

Επιστρέφει

```
array([[ 0.,  0.,  0.],  
       [ 0.,  0.,  0.]])
```

```
x = np.ones((2,3))
```

Επιστρέφει

```
array([[ 1.,  1.,  1.],  
       [ 1.,  1.,  1.]])
```



# πολυδιάστατα array

```
x = np.eye(3)
```

Επιστρέφει

```
array([[ 1.,  0.,  0.],  
       [ 0.,  1.,  0.],  
       [ 0.,  0.,  1.]])
```

```
x = np.array([1,2,3])
```

```
y = np.diag(x)
```

Επιστρέφει

```
array([[1, 0, 0],  
       [0, 2, 0],  
       [0, 0, 3]])
```

# πολυδιάστατα array

```
x = np.array([[1,2,3],[4,5,6]])
```

Επιστρέφει

```
array([[1, 2, 3],  
       [4, 5, 6]])
```

```
x.reshape(3,2)
```

Επιστρέφει

```
array([[1, 2],  
       [3, 4],  
       [5, 6]])
```

# πολυδιάστατα array

- Πράξεις: +, -, \*, /, \*\*: Γίνονται στοιχείο με στοιχείο

```
x = np.array([[1,2],[3,4]])
```

```
y = np.array([[1,0],[0,1]])
```

`x*y`

Επιστρέφει

```
array([[1, 0],  
       [0, 4]])
```

Για να γίνει η πράξη πρέπει τα `arrays` να είναι της ίδιας μορφής (`x+np.arange(3)` δεν γίνεται η πράξη)

# πολυδιάστατα array

- Πράξεις: διανυσματικές

```
x = np.array([1, 2, 3, 4])
```

```
y = np.array([1, 0, 0, 0])
```

```
x*y
```

Επιστρέφει

```
array([1, 0, 0, 0])
```

Για να γίνει το εσωτερικό γινόμενο:

```
x.dot(y) ή np.dot(x, y). Επιστρέφει
```

```
1
```

# πολυδιάστατα array

- Πράξεις: διανυσματικές

```
x = np.array([[1],[2],[3]])
```

```
y = np.array([1,1,1])
```

```
np.dot(y,x)
```

Επιστρέφει

```
array([6])
```

Ενώ το `np.dot(x,y)` δεν γίνεται

Για να γίνει η παραπάνω διανυσματική πράξη :

```
y = np.array([[1,1,1]])
```