

# Ασκήσεις εργαστηρίου

---

## Υπενθύμιση θεωρίας

### Βιβλιοθήκες της Python

Ένα σημαντικό πλεονέκτημα της Python είναι η ικανότητα χρήσης ενός μεγάλου αριθμού βιβλιοθηκών που επεκτείνουν σημαντικά τις δυνατότητες της γλώσσας για διάφορες εφαρμογές οι οποίες είναι χρήσιμες για έναν μηχανολόγο.

Μία από τις πιο σημαντικές βιβλιοθήκες της Python με πολλές δυνατότητες για υπολογισμούς είναι η `numpy`.

Η `numpy` όπως και οι άλλες συναρτήσεις αρχικά πρέπει να εγκατασταθεί (εάν δεν είναι ήδη διαθέσιμη) και στη συνέχεια καλείται με την εντολή `import`, στην οποία βάζουμε και μία συντομογραφία του ονόματος της βιβλιοθήκης για την καλούμε πιο εύκολα:

```
import numpy as np
```

Συνήθως οι κλήσεις στις βιβλιοθήκες γίνονται στην αρχή του κώδικα και τουλάχιστον πρέπει να έχουν γίνει πριν καλέσουμε συναρτήσεις από αυτές. Η κλήση συναρτήσεων γίνεται με τη χρήση της συντομίας π.χ. `np`, μία τελεία και στη συνέχεια το όνομα της συνάρτησης π.χ. `np.sin(x)` για το ημίτονο.

Η βιβλιοθήκη `numpy` περιλαμβάνει ένα μεγάλο αριθμό μαθηματικών συναρτήσεων όπως:

`np.sin(x)`, `np.cos(x)`, `np.tan(x)`, `np.arcsin(x)`, `np.arccos(x)`, `np.arctan(x)`, `np.degrees(x)`, `np.radians(x)`, `np.exp(x)`, `np.log(x)`

Επίσης διαθέτει αρκετές συναρτήσεις για άλλες λειτουργίες όπως η στρογγυλοποίηση: `np.round()`, `np.fix()`, `np.floor()`, `np.ceil()`.

# Ασκήσεις εργαστηρίου

---

## Υπενθύμιση θεωρίας

### Βιβλιοθήκες της Python

Μία πολύ σημαντική δυνατότητα που μας δίνει η βιβλιοθήκη `numpy` είναι και ο ορισμός πινάκων (`array`). Οι πίνακες έχουν αρκετή ευελιξία σε σχέση με τις λίστες και με αυτούς μπορούμε να εκτελέσουμε πράξεις όπως στη Γραμμική Άλγεβρα.

Ο ορισμός του πίνακα γίνεται με την εντολή `np.array`. Μπορούμε να κατασκευάσουμε ένα πίνακα ορίζοντας τις τιμές των στοιχείων του μέσα σε αγκύλες όπως στις λίστες : `a = np.array([1,2,3,4,5,6])`

Επίσης μπορούμε να κατασκευάσουμε διάφορους «έτοιμους» πίνακες όπως πίνακες με μηδενικά ή στοιχεία ίσα με τη μονάδα:

`a = np.zeros(3)` → `[0,0,0]`

`b = np.ones(6)` → `[1,1,1,1,1,1]`

Ακόμα και πιο ειδικούς πίνακες όπως τον πίνακα `I` (“eye”), δηλαδή πίνακα με 2 διαστάσεις ο οποίος έχει στη διαγώνιο τα στοιχεία ίσα με 1 και όλα τα στοιχεία εκτός διαγωνίου μηδενικά. `a = np.eye(3)`

Αν θέλουμε να κατασκευάσουμε έναν πίνακα-διάνυσμα με μια συγκεκριμένη σειρά αριθμών έχουμε δύο επιλογές: την εντολή `linspace` και την `arange`:

`a = np.linspace(1,10,50)` → 50 αριθμοί από 1 έως 10 (ορίζουμε το πλήθος των αριθμών και όχι το βήμα)

`a = np.arange(1,10,2)` → αριθμοί από 1 έως 10 με βήμα 2 (ορίζουμε το βήμα και όχι το πλήθος των αριθμών)

# Ασκήσεις εργαστηρίου

---

## Υπενθύμιση θεωρίας

### Βιβλιοθήκες της Python

Υπάρχουν εντολές για να υπολογίσουμε το μέγεθος των πινάκων ή τη διάστασή τους: `np.size`, `np.ndim`, `np.shape`. Επίσης μπορούν να χρησιμοποιηθούν διάφορες συναρτήσεις από τη στατιστική όπως μέσος όρος, διάμεσος και τυπική απόκλιση για τους πίνακες: `np.mean`, `np.median`, `np.std`.

Σχετικά με πράξεις μεταξύ πινάκων ενώ ορίζονται οι αντίστοιχες συναρτήσεις π.χ. `c=add(a,b)`, `c=multiply(a,b)`, υπάρχει η άμεση δυνατότητα να γίνουν πράξεις με τα γνωστά σύμβολα `+`, `-`, `*`, `/`, `**`. Έτσι η χρήση του `numpy` διευκολύνει τις πράξεις μεταξύ στοιχείων πινάκων σαν να ήταν πράξεις μεταξύ απλών μεταβλητών. Δηλαδή η εντολή `c = 2*a+b**2` με `a,b` πίνακες θα πολλαπλασιάσει τα στοιχεία του πίνακα `a` με `2`, θα υψώσει τα στοιχεία του πίνακα `b` στο τετράγωνο και μετά θα προσθέσει τους δύο νέους πίνακες και θα δημιουργήσει τον τελικό πίνακα `c`.

Εκτός από τη βιβλιοθήκη `numpy` μία άλλη ιδιαίτερα χρήσιμη βιβλιοθήκη είναι και η `matplotlib`, η οποία χρησιμοποιείται για γραφικές παραστάσεις. Η βιβλιοθήκη καλείται με την εντολή `import` ως εξής:

```
import matplotlib.pyplot as plt
```

Στη συνέχεια όλες οι εντολές καλούνται βάζοντας το `plt` στην αρχή.

Εάν θέλουμε να κάνουμε γραφική παράσταση σημείων χρησιμοποιούμε την εντολή `scatter` π.χ. `plt.scatter(x,y)` όπου `x,y` κατάλληλοι πίνακες-διανύσματα που πρέπει να έχουν οριστεί πριν καλέσουμε την εντολή `scatter`.

# Ασκήσεις εργαστηρίου

---

## Υπενθύμιση θεωρίας

### Βιβλιοθήκες της Python

Στη συνέχεια για να γίνει η γραφική παράσταση γράφουμε την εντολή `plt.show()`.

Επειδή όμως η γραφική παράσταση που προκύπτει δεν έχει τις απαιτούμενες πληροφορίες δηλαδή τα μεγέθη που απεικονίζονται στους δύο άξονες, τίτλο κ.α. πρέπει να χρησιμοποιήσουμε για αυτά επιπλέον συναρτήσεις:

`plt.xlabel(" ")` – μέσα στις παρενθέσεις γράφω το μέγεθος που απεικονίζεται στον x άξονα

`plt.ylabel(" ")` – μέσα στις παρενθέσεις γράφω το μέγεθος που απεικονίζεται στον y άξονα

`plt.title(" ")` – μέσα στις παρενθέσεις γράφω τον τίτλο

`plt.xlim( )` – γράφω τα όρια για τον άξονα x π.χ. 0,10

`plt.ylim( )` – γράφω τα όρια για τον άξονα y π.χ. 0,10

`plt.legend( )` – όταν έχουμε πάνω από μία καμπύλες χρησιμοποιούμε το `legend` για να υποδείξουμε την κάθε καμπύλη

`plt.grid("True")` – εμφανίζεται το πλέγμα πίσω από τη γραφική παράσταση

`plt.axis("equal")` – προσαρμόζονται οι άξονες σε σχέση με την πραγματική διάσταση

`plt.subplot( )` – ορίζουμε πολλά μικρότερα γραφήματα σε μια εικόνα π.χ. 2x2 οπότε γράφοντας 221 κάνουμε το πρώτο γράφημα, γράφοντας 222 το δεύτερο κτλ.

# Ασκήσεις εργαστηρίου

---

## Υπενθύμιση θεωρίας

### Βιβλιοθήκες της Python

Για γραφική παράσταση τύπου κατακόρυφης μπάρας χρησιμοποιούμε την εντολή `plt. bar` και για οριζόντιες μπάρες `plt.barh`.

ΠΡΟΣΟΧΗ: Στους κώδικες που καλούμε βιβλιοθήκες ισχύουν όλες οι υπόλοιπες εντολές και οι κανόνες της Python. Δηλαδή μπορούμε στον κώδικα να κάνουμε οποιουσδήποτε υπολογισμούς, να ορίσουμε συναρτήσεις κτλ. και στο τέλος να κάνουμε γραφικές παραστάσεις ή να χρησιμοποιούμε συναρτήσεις από τη βιβλιοθήκη `numpy`.