

## היכרות עם Numpy

במעבדה זו נבצע היכרות ראשונית עם הספרייה Numpy הפופולרית, בדגש על היכרות עם המערכים של הספרייה. בנוסף נממש את הנוסחה למכפלה פנימית (כולל מכפלה פנימית של מספרים מרוכבים)

## הוראות

עבור כל פונקציה, יש להשלים את הקוד בהתאם להנחיות המצורפות בדף זה.

## קבצים המצורפים למעבדה

- `lab2_numpy.py`
- זהו הקובץ אותו אתם נדרשים להגיש, אין לשנות לקובץ את השם, אין לדחוס את הקובץ (אי עמידה בהנחיות ההגשה תגרור הורדת ציון משמעותי)

### משימה 1: מטריצת יחידה

- השלימו את הפונקציה `create_identity_matrix` המקבלת מספר שלם ( $n$ ) ומחזירה מטריצה  $n \times n$  שאלכסונה הראשי הוא 1 ויתר האיברים הם 0.

### משימה 2: הכפלת איברים במערך

- השלימו את הפונקציה `scale_array` המקבלת מערך (או מטריצה) ומספר (שלם או עשרוני) ומכפילה את כל האיברים במערך במספר שהתקבל.

### משימה 3: סכום איברי עמודה

- השלימו את הפונקציה `get_cloumn_sum` המקבלת מטריצה ומספר עמודה ומחזירה את סכום כל האיברים בעמודה.

### משימה 4: בדיקת חיוביות

- השלימו את הפונקציה `check_positive` המקבלת מערך ומחזירה מערך בוליאני המציין אם כל איבר במערך המתקבל גדול מאפס או לא.

### משימה 5: חישוב ממוצע שורות במטריצה

- השלימו את הפונקציה `calculate_mean_rows` המקבלת מערך (מטריצה) ומחזירה מערך המכיל את הממוצע של כל שורה במטריצה (רמז: קראו על המשתנה `axis`)

### משימה 6: חישוב צמוד של מספר מרוכב

- השלימו את הפונקציה `complex_conjugate_vector` המקבלת מערך (מטריצה) ומחזירה מערך (מטריצה) עם הצמודים של כל איבר במערך שהתקבל.
  - תזכורת הצמוד של  $a+bi$  הוא  $a-bi$

### משימה 7: גודל מוחלט של מספר מרוכב

- השלימו את הפונקציה `get_complex_magnitude` המקבלת מערך ומחזירה מערך עם הגודל המוחלט של כל איבר
  - תזכורת הגודל המוחלט של מספר מרוכב  $a+bi$  שווה לה  $\sqrt{a^2 + b^2}$

## משימה 8: מטריצה הרימיתית

- השלימו את הפונקציה `check_hermitian` המקבלת מטריצה ומחזירה True אם המטריצה הרימיתית אחרת False
  - מטריצה הרימיתית היא מטריצה השווה לטרנספוז הצמוד שלה
  - השתמשו ב `np.transpose`, `np.conjugate`, `np.allclose` לפתרון התרגיל.

## משימה 9: מכפלה פנימית

- השלימו את הפונקציה `calculate_dot_product` המקבלת שני Iterables ומחזירה את המכפלה הפנימית ביניהם.
  - הנוסחה למכפלה פנימית נמצאת במצגת 1 שקף 18
  - רמז – כדאי להמיר את המשתנים המתקבלים למערכים של numpy
  - אם המערכים אינם באורך זהה זרקו חריגה מסוג Value Error
  - רמז 2 – קראו על הפונקציות `np.dot` ו `np.vdot` ותראו איזו מהן יכולה לסייע לכם בפתרון...

## משימה 10: השוואת ביצועים - מכפלה פנימית

- השלימו את הפונקציה `compare_perfomence_dot_product` המקבלת מספר שלם n ומחזירה tuple המכיל את זמני הריצה של הפונקציה.
  - צרו שני מערכי numpy באורך n עם ערכים אקראיים
  - צרו שתי רשימות עם אותם הערכים
  - מדדו את זמן חישוב המכפלה הפנימית על הרשימות באמצעות List Comprehension
  - מדדו את זמן חישוב המכפלה הפנימית על המערכים באמצעות `np.dot`
  - הפונקציה תחזיר tuple המכיל את הזמנים (`list_time`, `np_time`)

## משימה 11: השוואת ביצועים – הכפלת מערכים בסקלר

- השלימו את הפונקציה `compare_perfomence_scalar` המקבלת מספר שלם n ומחזירה tuple המכיל את זמני הריצה של הפונקציה.
  - צרו מערך numpy באורך n עם ערכים אקראיים
  - צרו רשימה עם אותם הערכים
  - מדדו את זמן הכפלת כל האיברים ברשימה במספר 3
  - מדדו את זמן חישוב הכפלת כל האיברים במערך במספר 3
  - הפונקציה תחזיר tuple המכיל את הזמנים (`list_time`, `np_time`)