

מבוא למדעי המחשב – שפט פיתון (234128)
חו"ף 2025
תרגיל בית 5

מועד הגשה: **04/02/2025** בשעה 20:00

זה היום האחרון של סמסטר חורף

הגשה רק בבודדים

חשוב לשמור את קוד
אישור הגשה שמקבלים
המערכת כמשמעותם!!

המתרגל האחראי על תרגיל זה: **הדר מטלן**

מספר: טאוב 710

E-mail: hadar.m@campus.technion.ac.il

שעת קבלה רגילה: **יום ב' 14:00-15:00**, בזום בקישור [2269702002](https://technion.zoom.us/j/2269702002)
שעות קבלה מיוחדות לשאלות על התרגיל:
יום ה' (23.01.2025) 15:00-16:00 , בזום בקישור [2269702002](https://technion.zoom.us/j/2269702002)
וביום ד' (29.01.2025) 13:00-14:00, טאוב 219

מילואים:

- מילואים:** שיטת ההקלות של קורס פיתון מותאמת לסילבוס של הקורס. **אין להסתמך בשיטות בקורסים אחרים!**
- לטיפול בבקשות הארכה להגשה או ביטול הגשה או עזרה בחומר הקורס יש לפנות **ליינה** (enovbari@gmail.com), המתרגלת האחראית של הקורס.
- באופן רגיל למילואים לא מושכים כל יומ מילואים = יום דחייה. על מנת לקבל דחיה עבור ת"ב,0 יש לשלוח מייל בצווף אישור מילואים (או מסמך שקוול) למתרגלת האחראית- יינה.
- בונוס של 5 נקודות ניתן לסטודנטים/יות שmagic'ים אלקטונית דרך אתר הקורס, בזמן עד למועד הגשה. לא ניתן טיפול ידני בונגוס (לא קשור לנסיבות ואישורים).

הוראות הגשה:

- הגשה אלקטונית **בבודדים**.
- תרגיל שיוגש בזמן (עד למועד הגשה ולא בתקופת הארכה) יקבל אוטומטית **בונוס של 5 נקודות**.
- תאריך הגשת ערעורים יתפרסם עbor כל תרגיל הבית לחודש.
- ערעורים על הדפסות שגויות בפלט שלכם לא יתקבלו**. עליך לבדוק את תוכנית שלך עם DiffMerge לפני הגשה.

הנחיות:

- אנא קרא/י את השאלות בעיון לפני שתתחל/י בפתרון.
- ניתן להניח כי הקולט תקין, אלא אם כן נאמר אחרת.
- יש לבדוק את האתר ה-FAQ של התרגיל לעיתים תכופות על מנת לקבל עדכונים על התרגיל.

- הגשה אלקטרונית בבודדים היא דרך האתר הקורס. הגיש קובץ **hw5.zip** (zip ולא פורמט אחר (!) כגון rar). **שם** של קובץ חייב להיות **באנגלית ללא רווחים**. אין דרישות נוספת לשם הקובץ, כלומר גם zip.sharon.mkobel מבχינת המערכת. הקובץ יכול בתוכו **בל** את הקבצים הבאים (לא לעשות zip על התיקייה בה נמצאים הקבצים להגשה – במקרה זה הגשה לא תתקבל ע"י המערכת!):
 - קובץ **students.txt** עם שマーク באנגלית, מספר תעודה זהה וכתובת האימייל שלך.
 - קובץ פתרון **hw5q1.py** המכיל את הקוד לשאללה עם כל הסעיפים.
 - קובץ pdf.12345678 המכיל את הקוד וההდפסות שלכם (ראו עמוד אחרון).
 - **מiad חשוב שם קובץ יהיה ת"ז הסטודנט**.
- יש להקפיד להגיש את כל הקבצים בדיקן עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה **לא תתקבל ע"י המערכת!**

שימוש לב- זהו תרגיל יבש:

- הבדיקה היא ידנית!
- הבדיקה تستכל על הקוד שכחיתם ולא רק על הפלט.

שימוש לב- לאתר ה- **FAQ** של התרגיל, אשר מכיל עכדים ותשובות לשאלות נפוצות, ויעדכן מפעם לפעם. חשוב להתעדכן לעתים תכופות – זה יכול להשוך הרבה בעיות, ונΚודות שירדו בבדיקהuschela. בפרט, לפני פניה למתרגל האחראי בדקו את הכתוב על מנת להשוך לעצמכם זמן המתנה לתשובה, שבדרכו כלל תהיה הפניה ל- **FAQ** בכל מקרה. ניתן למצוא את ה- **FAQ** [בלינק](https://docs.google.com/document/d/1sulv0p9_ZQRvHc4E-d3qi0bnZavgBG0ENsV7S9mo-1o/edit?usp=sharing).

https://docs.google.com/document/d/1sulv0p9_ZQRvHc4E-d3qi0bnZavgBG0ENsV7S9mo-1o/edit?usp=sharing

! חובה להתעדכן בקובץ השאלות הנפוצות !

https://docs.google.com/document/d/1sulv0p9_ZQRvHc4E-d3qi0bnZavgBG0ENsV7S9mo-1o/edit?usp=sharing

ת"י ב 5 מבוסס על תרגולים 9,10 והריצאות 9,10.

אין להשתמש בחומר שנלמד אחריהם.

תרגיל בית 5

אורן חובב ספורט מנסה להכין אתר עם כל מיני נתונים על תחרויות ספורט אהובות עליו. עזרו לאורן לכתוב פונקציות נחוצות לאתר.

למטרת התרגיל, סיפקנו לכם קובץ csv בפורמט של הקבציםuai שアイتم האטר יעבד:

sports_events.csv - עליו תצרכו לעשות הרצאה הסופית לצורך ההגשה (ראו ההסבר בסוף).

ניתן להניח שבכל קבצי csv שアイם אורן עובד, אין מחרוזת עם יותר מ20 תווים.

לאורך כל התרגיל אפשר ומומלץ להשתמש בפונקציות שמיימות בסעיפים קודמים.

בנוסף – אפשר להניח בכל הסעיפים שימושות התחרויות ושמות האתלטים אינם עלויים על 20 אותיות כל אחד.

חשוב מאוד!!! – לנוחותכם סיפקנו לכם קובץ ג'ופיטר **HW5_structure.ipynb** ש מכיל בדיקת המבנה של ההגשה הסופית! כל מה שアイם צריך לעשות, זה למלא את הפונקציות עם המימושים שלכם, ואז להריץ הכל. **חייבים להשתמש במבנה זה להימנע מהגשה שגיהה.**

סעיף 1

הקובץ **sports_events.csv** מכיל את מאגר התחרויות. כשלב ראשוני, כתבו פונקציה

```
def read_event_data(file_path)
```

ש מקבלת נתיב לקובץ CSV **file_path** (ניתן להניח שהמסלול חוקי), טוענת אותו **ומחזרה** אותו (לא להדפיס!!)

סיבות זמן: (1)0

יש להניח ששיטות הזמן של פונקציית **read_csv** של pandas היא קבועה.

הערה:تطענו ותדפיסו את 5 השורות הראשונות בשביל עצמכם, ותנסו להבין את המבנה של המאגר וכי צד שמורים בו פרטי התחרויות לפני שימושיכים בפתרון. יש למחוק הדפסה זו לפני ההגשה.

תשימושו בקטע קוד זה לבדיקה עצמית (תמחקו לפני ההגשה):

```
file_path = 'sports_event.csv'  
events = read_event_data(file_path)  
print(events.head(10))
```

ותקבלו את הדפסה הבאה:

	Event	Date	Athlete	Country	Score
0	Shot Put	2024-03-07	Liam Taylor	IND	4.79
1	Shot Put	2024-03-11	John Doe	USA	3.90
2	Long Jump	2024-03-05	Michael Brown	GBR	7.35
3	Long Jump	2024-03-09	Jane Smith	JAM	8.15
4	Shot Put	2024-03-10	Jane Smith	JAM	5.06
5	Long Jump	2024-03-05	Anna Martinez	GER	4.34
6	Discus Throw	2024-03-10	Michael Brown	GBR	9.70
7	Discus Throw	2024-03-03	John Doe	USA	5.61
8	Discus Throw	2024-03-02	David Kim	CHN	8.63
9	Shot Put	2024-03-03	James Wilson	KEN	2.58

סעיף 2

משו את הפונקציה:

```
def top_athlete(events, event_name)
```

شمבלת קובץ CSV טען (כלומר הפלט של הסעיף הקודם) events , ושם של תחרות, ושם של האתלט עם השם score הכי גבוה בתחרות הנתונה.

שים לב שהפונקציה מחזירה את שני הערכים, ולא מדפסה כלום.

ניתן להניח שהevent_name הוא שם של event תקין שקיים במאגר.

ניתן להניח שבבדיקה אתלט אחד השיג את הציון הכי גבוה.

סיבות דמן: (n)O כאשר ח הוא מספר ה-events בקובץ csv.

תשתמשו בקוד הבא לבדיקה עצמית (למבחן לפני הגשה!):

```
file_path = 'sports_event.csv'
events = read_event_data(file_path)
print(top_athlete(events, "Long Jump"))
```

תקבלו את הדרישה הבאה:

('David Kim' , 9.67)

סעיף 3

משו את הפונקציה:

```
def event_scores(events, event_name)
```

شمבלת קובץ CSV טען (כלומר הפלט של סעיף 1) events , ושם של תחרות (event_name) , ושם של האתלט scores של כל המופיעים באותו event .

אורך הרשימה המוחזרת צריכה להיות שווה למספר המופיעים של אותה event בcsv .

סיבות דמן: (n)O כאשר ח הוא מספר ה-events בקובץ csv.

תשתמשו בקוד הבא לבדיקה עצמית (למחוק לפני הגשה!):

```
file_path = 'sports_event.csv'  
events = read_event_data(file_path)  
print(sorted(event_scores(events, "Long Jump")))
```

ותקבלו את ההפוסה:

[0.35, 0.42, 1.4, 2.14, 2.65, 4.34, 6.48, 7.35, 8.15, 8.97, 9.67]

סעיף 4

ממשו את הפונקציה

```
def unique_events(events)
```

שמקלט קובץ CSV טען (כזכור הפלט של סעיף 1), **ומחזירה** (לא להדפיס!) רשימה עם סוגי האירועים השונים.

סיבות דמן: (מהו כאשר ח הוא מספר ה-events בקובץ csv ו-m הוא מספר סוגי events השוניים.)

תשתמשו בקוד הבא לבדיקה עצמית (למחוק לפני הגשה!):

```
file_path = 'sports_event.csv'  
events = read_event_data(file_path)  
print(sorted(unique_events(events)))
```

ותקבלו את ההפוסה:

['Discus Throw', 'High Jump', 'Long Jump', 'Shot Put']

סעיף 5

ממשו את הפונקציה

```
def count_events(events)
```

שמקלט קובץ CSV טען (כזכור הפלט של סעיף 1), **ומחזירה** (לא להדפיס!) שתי רשימות – רשימה של התחרויות השונות במגרש, ורשימה שמקילה את מספר התחרויות שהתרחשו בכל תחרות ברשימה. רשיימה הראשונה. סדר ההופעתן של התחרויות ברשימה הראשונה לא משנה אף הערכים ברשימה השנייה. חייבם להיות בהתאם לרשימה הראשונה.

השתמשו בפונקציה `unique_events` שמיימשתם בסעיף הקודם.

סיבוכיות זמן: (מהו 0 כאשר ח' הוא מספר ה-events בקובץ `acs` ו- מ' הוא מספר סוגי events
השוניים).

תשתמשו בקוד הבא לבדיקה עצמית (למחוק לפני הגשה!):

```
file_path = 'sports_event.csv'  
events = read_event_data(file_path)  
print(count_events(events))
```

ותקבלו את ההדפסה (לא בהכרח באותו סדר):

```
(['Shot Put', 'Long Jump', 'Discus Throw', 'High Jump'], [18, 11, 8, 13])
```

בדוגמא לעיל אנו רואים שהevent 'Shot Put' הופיע 18 פעמים במאגר, הת'event 'Long Jump' הופיע 11 פעמים במאגר, הת'event 'Discus Throw' הופיע 8 פעמים במאגר והevent 'High Jump' הופיע 13 פעמים במאגר.

חלק יבש - ענו על השאלות הבאות

- (1) הסבירו בקצרה מדוע השימוש שלהם עומדת בסיבוכיות הנדרשת?
- (2) איך סיבוכיות הזמן תשתנה במידה ולא יהיה נתון שם של תחרות לא עולה על 20 אותיות?

סעיף 6

ידעו שבכל התחרויות שבכל התחרויות `scores` המקיים מל' הוא 10.

ממשו את הפונקציה

```
def analyze_scores(events, event_name)
```

שמתקבל קובץ CSV טוון (כלומר הפלט של סעיף 1)(`events`) ושם של תחרות (`event_name`) ושם של התחרות (`event`).
מחזירה (לא להדפיס!) היסטוגרמָה המציגת כמה פעמים השיגו `score` בכל טווח של `scores` מ-0 ועד 10 בקטגוריות של אחד.

השתמשו בפונקציה `event_scores` שמיימשתם בסעיף 3.

לדוגמא:

עבור המאגר עם עשר שורות:

	Event	Date	Athlete	Country	Score
0	Shot Put	2024-03-07	Liam Taylor	IND	4.79
1	Shot Put	2024-03-11	John Doe	USA	3.90
2	Long Jump	2024-03-05	Michael Brown	GBR	7.35
3	Long Jump	2024-03-09	Jane Smith	JAM	8.15
4	Shot Put	2024-03-10	Jane Smith	JAM	5.06
5	Long Jump	2024-03-05	Anna Martinez	GER	4.34
6	Discus Throw	2024-03-10	Michael Brown	GBR	9.70
7	Discus Throw	2024-03-03	John Doe	USA	5.61
8	Discus Throw	2024-03-02	David Kim	CHN	8.63
9	Shot Put	2024-03-03	James Wilson	KEN	2.58

עבור הקריאה לפונקציה

```
analyze_scores(events, "Long Jump")
```

נקבל את הפלט:

[0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0]

כasher התא הראשון ברשימה מייצג את מספר התחרויות באותו event score שקיבלו בין 0 ל-1 (לא כולל) התא השני ברשימה מייצג את מספר התחרויות באותו event score שקיבלו בין 1 ל-2 (לא כולל וכו'). כלומר הערכים ברשימה מתאימים לscores:

[0-1, 1-2, 2-3, 3-4, 4-5, 5-6, 6-7, 7-8, 8-9, 9-10, 10]

בדוגמא לעיל

'Long Jump' מופיע 3 פעמים במאגר. פעם אחת עם ציון בין 5-4, פעם אחת עם ציון בין 7-8 ופעם אחת עם ציון בין 9-8. לכן הסטטוגרמה המוחזרת היא:

[0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0]

סיבוכיות זמן: (ח) כasher ח הוא מספר ה-events בקובץ csv

תשימושו בקוד הבא לבדיקה עצמית (למחוק לפני הגשה!):

```
file_path = 'sports_event.csv'
events = read_event_data(file_path)
print(analyze_scores(events, "Long Jump"))
```

ותקבלו את ההדפסה:

[2, 1, 2, 0, 1, 0, 1, 1, 2, 1, 0]

חלק יבש - ענו על השאלות הבאות

- (1) הסבירו בקצרה מדוע השימוש שלכם עומד בסיבוכיות הנדרשת?
- (2) הסבירו את ההבדל בסיבוכיות של הפונקציה הזאת לעומת הפונקציה בסעיף הקודם. למה ניתן להשיג סיבוכיות יותר טובה בפונקציה הזאת?
- (3) בהנחה שסיבוכיות הזמן של הפונקציה event_scores נשרת (ח(0), האם הסיבוכיות של הפונקציה analyze_scores תושפע אם לא יהיה נתון שאורך שם של תחרות/שם של אתלט לא יכול לעלות על 20 תווים?)

סעיף 7

משמעות הפונקציה

```
def analyze_event(file_path, event_name)
```

המקבלת מחרוזת של מסלול לקובץ CSV (ניתן להניח שהמסלול חוקי), ושם של event. הפונקציה מבצעת את הפעולות הבאות (ראו גם את פורתט הפלט בדוגמה):

- טוענת את קובץ ה- CSV שנמצא ב- file_path (ניתן להניח שמסימן חוקי).
- מדפיסה למסך את שם האתלט שהשיג את הערך score הכי גבוה ב- event הנתון. ניתן להניח שבדיק אתלט אחד השיג את הציון הכי גבוה.
- משרטת למסך את היסטוגרמה scores של אותה event.

דוגמה ופורטט הפלט

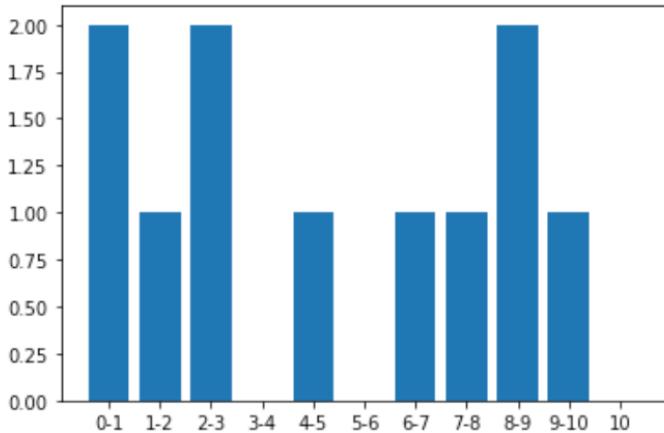
למשל עבור sports_event.csv שמסופק עם התרגיל:

ההרצה של הקוד:

```
file_path = 'sports_event.csv'  
analyze_event(file_path, "Long Jump")
```

תדפיס למסך את הפלט בפורטט הבא:

Top athlete in Long Jump is David Kim with score: 9.67



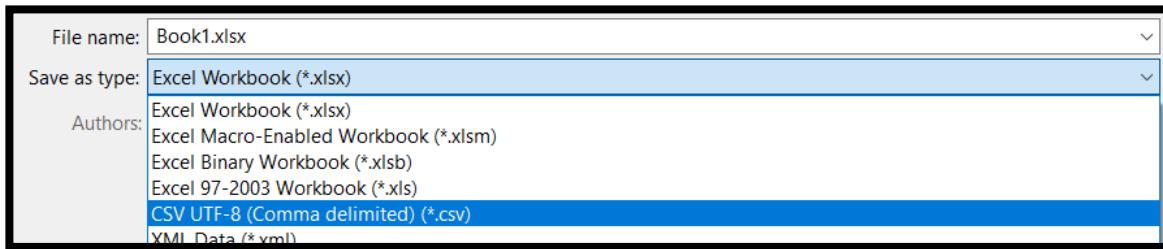
הfonkzia רק מדפסה פלטים למסך והוא לא מוחזירה שם ערך.

הרצות

אחרי שימושם את הפונקציות בכל הסעיפים, והשתמשתם בקודי הקוד שננתנו לבדיקה עצמית, אתם נדרשים להציג את הקוד שלכם, עם הרצות על המאגר שסיפקנו – sports_event.csv .

בנוסף, אתם מתבקשים בסעיף האחרון לבנות דוגמה משלכם – ליצור קובץ מאגר חדש בשם my_events.csv , ולהריץ את הפונקציה analyze_event עליו.

כדי להכין קובץ csv משלכם, תיצרו קובץ excel, תמלאו אותו, ותשמרו אותו עם סימנת csv .



חשוב – לשים את קבצי ה- csv באותה תיקיה של הקוד לפני ההרצות.

לאחר המימוש והרצות אתם יכולים להמיר את המחברת לקובץ pdf . רואו הסבר למטה.

חשוב מאוד!!! - לנו חוויתכם סיפקנו לכם קובץ ג'ופיטר HW5_structure.ipynb שמכיל בדיקת המבנה של ההגשה הסופית! כל מה שאתם צריך לעשות, זה מלא הפונקציות עם המימושים שלכם, ואז להריץ הכל. **חייבים להשתמש** במבנה זה להימנע מהגשה שגוייה.

הוראות הגשה

אתם מתבקשים להציג קובץ ZIP עם הקבצים הבאים:

- קובץ **students.txt** עם שמר באנגלית, מסpter תעודת הזהות וכתובת האימייל שלך.
- קובץ פתרון **hw5.py** המכיל את הפונקציות הנדרשות.
- קובץ **12345678.pdf** המכיל את הקוד, ה הדפסות ותשובה לשאלות המילוליות (שם הקובץ הוא מסטר תעודת הזהות).

מסמך PDF

במסמך ה- **pdf** שאתם מגישים צריכים להיות הדברים הבאים:

1. הפונקציות שנדרשתם לכתוב והרצות הדרשות:

- זה קובץ **ch5.ipynb**, אחרי שהשלמתם את המקומות החסרים עם מימושי הפונקציות והרצתם הכל.
- מותר לכם להוסיף תאים לצורך מימוש פונקציות עזר משלכם בכל הסעיפים.

את המחברת הזאת אתם מmirrim לקובץ **pdf**. **הוראות ליצירת PDF מהnotebook jupyter** מופיעים בהמשך.

2. תשובה לשאלות היבשות יכולות להיות רשומות באחת מה דרכים הבאות:

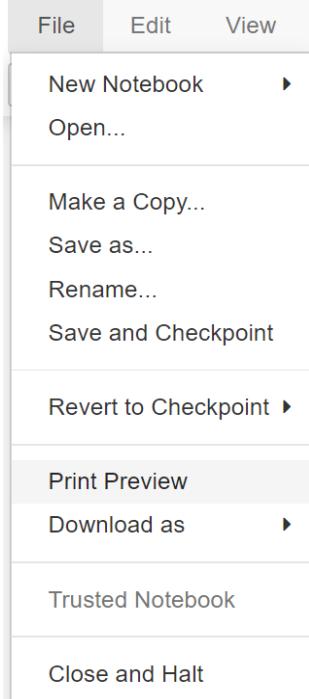
- מסמך WORD או TEXT כלשהו שהומר לPDF וחובר למסמך עם הפונקציות והרצות
- תא נפרד בnotebook notebook jupyter שם כתובות התשובות (ראו מבנה לדוגמא - **(HW5_structure)**.)
- סריקה של התשובות הכתובות על דף בכתב קרייא וברור

שים לב! אתם מגישים קובץ PDF **אחד בלבד!**

ישנם המונ אטרים חינמיים המאפשרים איחוד של קבצי PDF. אתר לדוגמה:

https://www.ilovepdf.com/merge_pdf

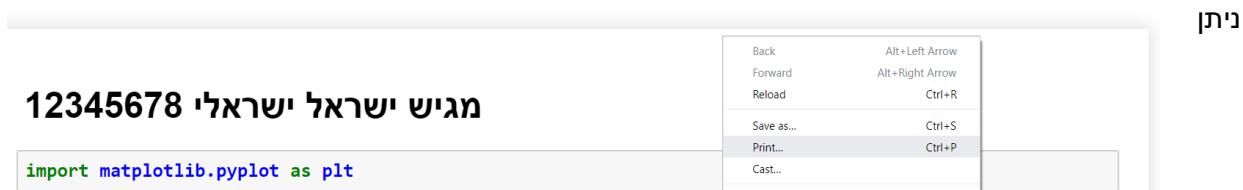
יצירת PDF מ-Jupyter notebook



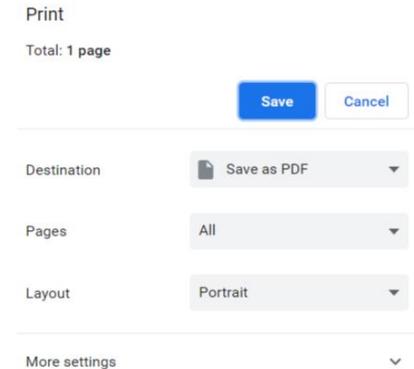
לחצו על File:

ואז בחרו ב- Print Preview כדי לראות את
air תראה המחברת ג'ופיטר שלכם
כשהיא תודפס (או נשמר כקובץ PDF)

כשאתם מוכנים, לחצו על הכפתור הימני בעבר וatz ב- print או לחצו P+ctrl:



להעביר את העבודה לפקד ע"י בחירה באפשרות - Save as PDF



מאל חשוב שם קובץ PDF יהיה ת"ז הסטודנט.

בצלחה!
😊