**Python**

לפני הגעתכם לקורס תתבקשו ללמוד את בסיס שפת התכנות הנקראת Python

## הוראות לחניך

הלמידה שלכם תתבצע בעזרת המדריך של אלעד פלג – מומחה אנליטיקה וBI, אותו תוכלו למצוא בכתובת: <https://www.youtube.com/playlist?list=PLVCiFzJMihGlmX4_FAELKaBNTZdp6b3v7>

בנוסף תוכלו למצוא סיכומים בחלק מן הנושאים באתר המדריך בכתובת: [Elad Peleg Python-guide](https://www.eladpeleg.co.il/python-guide)

שימו הלמידה מתבצעת מהמדריך בYouTube וההוראות מתייחסות אליו

סביבת העבודה שמשתמש המדריך נקראת Jupiter notebook, אתם תעבדו בסביבת עבודה שנקראת idle. הסבר על סביבת העבודה idle בה תעבדו בכתובת: [Idle Instructions.mp4](https://www.pitt.edu/~naraehan/python3/WIN-Python3.5-try.mp4)

את לימודי השפה בקורס נמשיך מנקודת הידע אליה תגיעו במדריך זה, לכן יש ללמוד היטב חומר זה על מנת להימנע מיצירת פער ביכולות ובידע שלכם בתחילת הקורס.

בתחילת הקורס **תיבחנו** על הנושאים השונים והשיעורים אותם תלמדו במהלך במדריך. הנושאים שעליכם ללמוד במדריך הם:

1. כתיבת ליטרלים (סרטון 3)
2. הפונקציה פרינט (סרטון 4)
3. משתנים + עבודה עם מספרים (סרטונים – 5, 6)
4. הפונקציה Input (סרטון 8)
5. המרות (סרטון 9)
6. מחרוזות + פעולות נוספות + חיתוך מחרוזות + string format (סרטונים – 10, 11, 12, 18)
7. ביטויים בוליאנים + תנאים + תנאים עם קלט + תנאים מורכבים (סרטונים – 13, 14, 15, 17)
8. וידוא תקינות נתונים (סרטון 16)
9. לולאות while + continue and break (סרטונים – 19, 20, 21)
10. פונקציית range (סרטון 22)
11. לולאות for + קינון (סרטונים – 23, 24, 25)
12. רשימות + פעולות נוספות + עריכת רשימות + העתקת רשימות (סרטונים – 26, 27, 28, 30)
13. טאפלים (סרטון 31)
14. סטים + פעולות על סטים (סרטונים – 37, 38)
15. מבוא למילונים + עריכת נתונים במילונים + ניהול ערכים במילונים (סרטונים – 33, 34, 35)

שימו סדר הסרטונים מעט שונה מן הסדר המקורי בYouTube

תרגולים

יש לציין שחשוב לתרגל כמה שיותר ולהתנסות עם השפה לשם הבנה.

לשם כך ישנם טאבים Types וLoops בכל אחד תרגילים ברמות שונות.

ניתן לחלק את התרגול לפי נושאים:

* Types חלק א לאחר נושאים 1-6
* Types חלק ב לאחר נושאים 7-8
* Loops חלק א' לאחר נושאים 9-11
* Types חלק ג' + Loops חלק ב' לאחר נושאים 12-15
* Dictionaries לאחר נושא 15

## Types & Conditions

**חלק א'**

1. כתבו תוכנית אשר מקבלת מחרוזת ומדפיסה אותה מהסוף להתחלה.
2. כתבו תוכנית אשר מקבלת שם פרטי ושם משפחה ומדפיסה כתובת מייל.

לדוגמא, עבור הקלט Oshri Avneri התקבל הפלט [Oshri.Avneri@mail.com](mailto:Oshri.Avneri@mail.com)

1. כתבו תוכנית המקבלת גובה ואורך הבסיס של משלוש שווה צלעות ומדפיסה:

* גובה ואורך בסיס
* היקף ושטח

1. כתבו תוכנית המקבלת את מספר השנים בהם אתה חי, התוכנית תדפיס:

* גיל
* גיל בימים
* גיל בשעות
* גיל בשניות

**חלק ב'**

1. כתבו תוכנית אשר מקבלת גיל מהמשתמש.

* אם הגיל הוא 18, התוכנית תדפיס ‘Congratulations’
* אם הגיל קטן מ 18, יודפס ‘You are young’
* אחרת יודפס ‘We love old people’

אם גיל המשתמש אינו מספר יש להדפיס הודעה בהתאם.

1. כתבו תוכנית אשר מקבלת שלושה מספרים ומחזירה את המספר האמצעי

(**גדול מהמספר הכי קטן, קטן מהמספר הכי גדול**)

במידה ואין כזה התוכנית תדפיס את כל המספרים

1. כתבו תוכנית אשר מקבלת טמפרטורה מהמשתמש.

התוכנית תמיר את הטמפרטורה מצלזיוס לפרנהייט.

לדוגמא התוכנית תקבל **C**24 ותחזיר 75.2**F**, תקבל **F**39.2 ותחזיר **C**4.

נוסחת ההמרה - (C x 1.8) + 32= F)

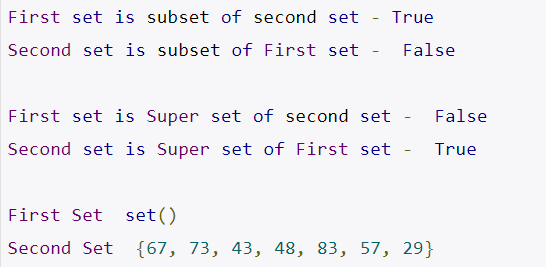
1. כתבו תוכנית אשר מקבלת מחרוזת, התוכנית תחזיר מחרוזת המורכבת מ:

* 2 האותיות ראשונות
* 2 האותיות האחרונות

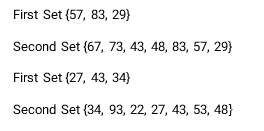
אם המחרוזת הקצרה מ2 תווים תוחזר הודעה "too short"

**חלק ג'**

1. כתבו תוכנית הבודקת אם סט אחד הוא חלק מ או מכיל את הסט השני ולהפך, אם נמצא שהוא חלק מהסט השני יש למחוק אותו. פלט התוכנית יראה כך:



בדקו את התוכנית על הזוגות הבאים:



1. כתבו תוכנית אשר תקבל המשתמש שני מספרים ופעולה חשבונית ותחזיר את התוצאה, התוכנית תחזיר תמיד מספר שלם, במידה והוכנס משהו שהוא לא מספר או פעולה חשבונית התוכנית תדפיס הודעה בהתאם.

פלט התוכנית יהיה כמו תרגיל חשבון, NUM1 [+|-|X|/] NUM2 = NUM3

1. כתבו תוכנית שתמצא את החיתוך בין שני סטים ותמחק את הערכים שבחיתוך מהסט הראשון, התוכנית תדפיס את שני הסטים, את החיתוך ואת הסט הראשון לאחר מחיקת החיתוך.

**Loops & Lists**

**חלק א'**

1. כתבו סקריפט שמוצא את כל המספרים בין 1500 ל5900 שמתחלקים ב7 והם כפולה של 5 ומחזיר אותם בסדר יורד
2. כתבו תוכנית אשר תקבל מספר מהמשתמש (ניתן להניח שהתקבל מספר) ותדפיס פירמידה בגובה המספר, **לדוגמה** - עבור הקלט 5 הפלט יראה כך:

\*   
\* \*   
\* \* \*   
\* \* \* \*   
\* \* \* \* \*   
\* \* \* \*   
\* \* \*   
\* \*   
\*

1. הדפיסו את כל המספרים בסדרת פיבונצ'י אשר ערכם קטן מ-10,000, חובה להשתמש בwhile True, תוכלו למצוא הסבר על סדרת פיבונצ'י בקישור הבא: <https://he.wikipedia.org/wiki/%D7%A1%D7%93%D7%A8%D7%AA_%D7%A4%D7%99%D7%91%D7%95%D7%A0%D7%90%D7%A6%27%D7%99>
2. 7 בום – הדפיסו למסך את כל המספרים בין 0 ל 1000 שמתחלקים ב -7 ללא שארית או מכילים את הספרה 7, לפי הסדר.

השתמשו רק בפעולות חשבון!

עזרה: פעולת מודולו – החזרת השארית מחלוקה, מתבצעת בעזרת סימן %.

לדוגמא: 14%4

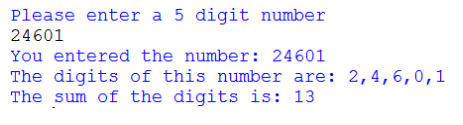
מחזיר 2 (14 לחלק ל4 שווה ל3 עם שארית 2).

**חלק ב'**

1. כתבו תוכנית אשר מקבלת מהמשתמש מספרים חיובים (ניתן להניח שמתקבל מספר) עד קבלת מספר שלילי ומדפיסה את הממוצע, מקסימום ומינימום
2. כתבו תוכנית שקולטת מהשתמש מספר בעל 5 ספרות ומדפיסה:

* את המספר עצמו
* את ספרות המספר כל ספרה בנפרד מופרדות על ידי פסיק (ללא פסיק אחרי הספרה האחרונה)
* את סכום ספרות המספר

דוגמא לריצה תקינה:



1. כתבו תוכנית שקולטת מהמשתמש שתי רשימות ובודקת אם ערכי הרשימה השנייה נמצאים בראשונה, במידה וכן תדפיס את הרשימה הראשונה ללא הערכים המשותפים ואת הרשימה השנייה, במידה ולא תדפיס את שתי הרשימות

## Dictionaries

**מצורף כאן סיכום לשימושכם על נושא המילונים, מוזמנים לקרוא אותו ולהיעזר בו בפתירת התרגילים:**

**מהו מילון?**

מילון (Dictionary) ב Python הוא טיפוס שמטרתו היא המרה בין מפתחות לבין ערכים.

מילון מחזיק מפתחות, ערכים ואת המיפוי ביניהם. כאשר המשתמש פונה למילון, הוא מציין מפתח – ומקבל עבורו ערך.

**הצורך במילון:**

מילון מאפשר לנו לגשת לערכים לא בעזרת מיקומים (index) ברשימה, אלא באמצעות כל דבר שנבחר, נוכל לפנות אליהם בעזרת אותיות בודדות, מחרוזות וגם מספרים לבחירתנו.

מה הכוונה?

אם עד היום התרגלנו לפנות למשתנים השמורים ברצף(למשל רשימה) לפי האינדקס המספרים שלהם, למשל:

List = ['a','b','c']

על מנת לפנות לאותיות נצטרך לפנות למיקום שלהם.

למשל כדי לפנות לאות 'a' נצטרך לפנות אליה כך: List[0]

כעת בעזרת מילונים נוכל להגדיר איך נרצה לפנות אליהם.

**הגדרת מילון:**

מילון נגדיר בעזרת סוגריים מסולסלים:

example **=** **{**'bob' **:** 10**,** 'bill' **:** 20**,** 'alice' **:** 3**0}**

המילון שלנו הוא example

על מנת להגדיר צמד של מפתח וערך נשתמש בנקודתיים.

מצד שמאל נמצא המפתח ומצד ימין נמצא הערך.

לאחר הגדרת המילון אם נרצה לפנות לערך מסוים נשתמש במפתח שהוגדר לו.

כלומר, אם נרצה לפנות למספר 10 נשתמש ב'bob' בתור אינדקס.

example['bob']

כך לא נצטרך להתעסק עם מיקומים אלא נוכל לפנות בעזרת כל דבר שנגדיר כמפתח.

דגש חשוב: מפתח חייב להיות immutable.

**הוספת מפתחות וערכים:**

על מנת להוסיף ערכים נשתמש בסוגריים מרובעים:

Example['morris'] = 40

כעת נוכל לפנות אל הערך 40 עם המפתח morris

זוהי הדרך שגם נשנה ערכים, כלומר אם המפתח morris היה קיים, אז הערך אליו ניתן לפנות עם המפתח morris היה משתנה ל40.

דרך נוספת להוסיף ערכים זה על ידי המתודה update:

מקבלת מילון ומשרשרת אותו אל המילון הקיים (מוסיפה את כל המפתחות והערכים של המילון הנוסף בהתאמה

example**.**update**({**'hello' : 1**,** 'world' : 2**})**

{'bob' : 10, 'bill' : 20, 'alice' : 30,'hello' : 1 , 'world': 2}

ואם פשוט נדפיס את המילון הוא יראה כך:

example

{'world': 2, 'bob': 10, 'bill': 20, 'alice': 30, 'hello': 1}

יתרונות של השיטה הראשונה: אינטואיטבי, פועל כמו השמה רגילה של כל משתנה.

יתרונות של המתודה update: ניתן להוסיף מספר איברים במקביל.

**מחיקת ערכים:**

על מנת למחוק ערכים נשתמש בdel ונכתוב את המילון המפתח המתאים.

del example['bob']

כעת הערך 1 נמחק ולא נוכל לפנות אליו יותר עם 'bob'

**פונקציות שימושיות:**

כדי לבדוק האם מפתח מסוים נמצא במילון נוכל להשתמש ב- in:

דגש: זה עובד רק עם מפתח ולא עם ערכים.

'Japan' **in** example

False

'bob' **in** example

True

**items**

מחזיר אובייקט מסוג dict\_items שמציג את המפתחות והערכים בזוגות.

example**.**items**()**

**[(**'bob'**,** 1**),** **(**'bill'**,** 2**),** **(**'alice'**,** 3**)]**

**keys**

מחזירה אובייקט מסוג dict\_keys של כל המפתחות.

example**.**keys**()**

**[**'bob'**,** 'bill'**,** 'alice'**]**

**values**

מחזירה אובייקט מסוג dict\_values של כל הערכים.

example**.**values**()**

**[**1**,** 2**,** 3**]**

* + - 1. כתבו תכנית אשר מקבלת אורך מילון כקלט (n), ויוצרת מילון לפי האורך בפורמט הבא:

Dictionary = {x : x\*x}, עבור n = 3 המילון יראה כך – dictionary = {1 : 1, 2 : 4, 3 : 9}

* + - 1. מתוחזק מילון אשר מכיל את ממוצע הציונים של כל תלמיד בכיתה. הממוצעים של התלמידים השתנו ונוסף תלמיד חדש, לכן צריך לעדכן את המילון!

1. מצא שתי דרכים לעדכן את המילון –

grades = {'yoav' : 98, 'dana' : 37, 'ohad' : 70, 'tal' : 100, 'ariel' : 90, 'itay' : 66, 'maya' : 88}

new\_grades = {'yoav' : 90, 'ohad' : 76, 'maya' : 80, 'ido' : 80}

1. המורה מעוניינת לראות גם את הממוצע הכיתתי של התלמידים לאחר העדכון, חשבו והדפיסו אותו.
   * + 1. כתבו תכנית שמנהלת חנות, מנהל החנות רוצה לבצע חמש פעולות בזו אחר זו (שימו לב להוסיף לו הודעה מתאימה עבור כל פעולה, וקלטו ממנו בכל פעם את הערכים הנדרשים להשלמת הפעולה – למשל שם מוצר או כמות יחידות):
2. לראות כמה יחידות יש ממוצר מסוים לפי שם מוצר
3. למכור מוצר (יורדות יחידות מהמוצר הקיים)
4. לקנות מוצר (עולות יחידות למוצר הקיים)
5. להוסיף מוצר חדש לחנות
6. להוציא מוצר מהמלאי בחנות

יש להשתמש במילון לצורך ביצוע המשימה, צרו מלאי של חנות לבחירתכם ועבדו עליו.

**תרגול מתגלגל**

ספק מכוניות רוצה לבנות מערכת שתקבל מסוכנויות הזמנות באופן אוטומטי ואתם נבחרתם לעזור לו! בנו תוכנית הקולטת תחילה תקציב מהסוכנות.

לאחר מכן התוכנית תציג לסוכנות את תקציבה ואת המכוניות כשליד כל מכונית יוצג המחיר שלה

הסוכנות תוכל להוסיף מכונית אחת כל פעם להזמנה והתוכנית תמשיך להציג לה את תקציבה המעודכן ואת המכוניות אותן תוכל לקנות עד שאין עוד מכוניות במלאי של הספק או שלסוכנות אין מספיק תקציב לאף מכונית נוספת.

הוסיפו לסוכנות אופציה לסיום ההזמנה אם תרצה ללא קשר לתקציב או למלאי המכוניות.

שימו לב – אין להציג לסוכנות מכונית שעולה יותר מהתקציב הנוכחי או מכונית שאזלה במלאי.

בסיום ההזמנה התוכנית תציג לסוכנות:

* מה התקציב שנשאר בסיום הקנייה
* כמה מכוניות קנתה מכל סוג
* כמה מכוניות נותרו במלאי מכל סוג

המכוניות אותן הספק מוכר ומחירן:

'F' עבור פרארי שמחירה 600

'P' עבור פורש שמחירה 200

'L' עבור למבורגיני שמחירה 400

ניתן לצאת מנקודת הנחה שהקלט מהמשתמש תקין

ממשו את התוכנית עבור מלאי של 5 פרארי, 4 פורש ו3 למבורגיני.

בונוס – בנו את אותה תוכנית אך הפעם קחו כקלט מהספק את סוגי המכוניות, המחיר של כל מכונית וכמה מכוניות מכל סוג יש במלאי.

**Extra**

ענו על השאלות הבאות:

1. מה הם סוגי הנתונים שקיימים ב – ?Python פרטו והסבירו על כל אחד מהם.
2. מה ההבדל בין mutable ל– ?immutable הסבירו בהרחבה.

תנו 3 דוגמאות לטיפוסי נתונים שהם mutable ו3 דוגמאות לטיפוסי נתונים שהם immutable.

1. מתי נשתמש בwhile ומתי בfor?
2. מה זה conditional expression?