

תרגיל בית 5 חלק יבש:

סעיף 1:

- הסבר מה הקוד עושה:

הקוד יוצר רשימה חדשה בשם `my_list` וממיר כל תו במחרוזת `my_str` במקום `index` לערך ה-Unicode שלה על ידי שימוש בפונקציה `ord()` ומוסיף את הערך לרשימה החדשה במקום ה-`index`. כלומר `my_list[index] = ord(my_str[index])`.

- Oneliner שקול:

```
my_list = [ord(my_str[index]) for index in range(len(my_str))]
```

סעיף 2:

- הסבר מה הקוד עושה:

שורה ראשונה: `my_dictionary =`
הקוד יוצר מילון בשם `my_dictionary` על ידי שימוש בסוגרים מסולסלים,

שורה שניה: `for x in [1,2,3,4,5,6,7,8,9,10][::-1]`
הקוד עובר על כל ערך `x` ברשימה `[1,2,3,4,5,6,7,8,9,10]` בסדר הפוך (מ-10 עד 1) באמצעות חיתוך רשימה (*slicing*) עם `[::-1]`.

שורה שלישית הוספת ערכים למילון: `f"{x}^{x} is {x ** x}\n"`
לכל מספר `x` מהלולאה, הקוד מוסיף ערך למילון `my_dictionary` כך שהמפתח (*key*) הוא `x`, והערך (*value*) שממופה למפתח הוא מחרוזת בפורמט הבא: `f"{x}^{x} is {x ** x}\n"`. המחרוזת נבנית על ידי שימוש ב-*f-string* שמחליף את הערכים של `x` בתוך המחרוזת המתאימה.

ובשורה האחרונה הדפסת ערכי המילון: `print(*my_dictionary.values())`
הקוד מדפיס כל מחרוזת מה- (*values*) של המילון `my_dictionary` בשורה נפרדת כי כל מחרוזת מסתיימת ב-`\n`.
השימוש בפונקציה `values()` מחזיר את כל הערכים במילון.
אופרטור הפיזור `*` משמש לפריסת הערכים כאילו כל אחד מהם הוא פרמטר נפרד לפונקציה `print()`, מה שמאפשר הדפסה של כל הערכים בפורמט הנכון.

- Oneliner שקול:

```
print(*{x: f"{x}^{x} is {x ** x}\n" for x in range(10,0,-1)}.values())
```

סעיף 3:

- הסבר מה הקוד עושה:

שורה ראשונה: `for i in range(ord('A'), ord('z') + 1, 2):`
הקוד עובר על כל הערכים המספריים בטווח המתחיל מהערך ה-Unicode של האות 'A' (65) עד לערך ה-Unicode של האות 'z' (122) בתוספת 1, וזאת בקפיצות של 2 בכל איטרציה.
הפונקציה `ord()` מחזירה את ערך ה-Unicode של תו נתון, כך שהערך של `i` בכל איטרציה מייצג מספר Unicode.

שורה שניה בדיקת תנאי האם התו הוא אלפביתי: `if chr(i).isalpha():`
השורה הזו בודקת האם התו המיוצג על ידי הערך `i` (באמצעות `chr(i)`) הוא תו אלפביתי. הפונקציה `chr(i)` מחזירה את התו המתאים לערך ה-`i` Unicode. הפונקציה `isalpha()` מחזירה `True` אם התו הוא תו אלפביתי (`A-Z, a-z`), אחרת היא מחזירה `False`.

שורה שלישית הדפסה בפורמט מסוים:
`print(f"The ASCII number {i} represent the char {chr(i)}")`
`,"The ASCII number {i} represent the char {chr(i)}"`
כאשר `{i}` מוחלף בערך המספרי של `i` ו-`{chr(i)}` מוחלף בתו המתאים לערך זה בטבלת Unicode על ידי שימוש ב-`f-string`.

• Oneliner שקול:

```
print(
* [f"The ASCII number {i} represent the char {chr(i)}" for i in range(ord('A'), ord('z')
+ 1, 2) if chr(i).isalpha()], sep = '\n')
```

סעיף 4:

• הסבר מה הקוד עושה:

שורה ראשונה: `tmp_chr = ""` יצירת מחרוזת ריקה:
הקוד יוצר מחרוזת ריקה בשם `tmp_chr` שישמש לאגירת התווים.

שורה שניה: `for num in list_c:` לולאת `for` לעבור על כל מספר ברשימה:
הקוד עובר על כל מספר `num` ברשימה `list_c`. הרשימה מכילה ערכים שלמים שמייצגים קודים של תווים בטבלת Unicode.

שורה שלישית: `tmp_chr += chr(num)` שרשור התו המתאים לערך המספרי למחרוזת בכל איטרציה של הלולאה, התו המתאים לערך המספרי `num` מתווסף (=+) למחרוזת `tmp_chr`.

שורה אחרונה: `print(tmp_chr)` הדפסת המחרוזת שנוצרה:
הקוד מדפיס את המחרוזת `tmp_chr` שנבנתה במהלך הלולאה, שהיא מחרוזת של תווים שנוצרו מהערכים המספריים ברשימה `list_c`.

• Oneliner שקול:

```
print(''.join(map(chr, my_list)))
```

