



חעי מימרן

1.6

מחרוזות

# מחרוזות (strings)

- ▶ מחרוזת היא סדרה של תווים (characters): אותיות קטנות וגדולות, ספרות, רווחים, וסימנים שונים שחלקם מופיעים על המקלדת.
  - ▶ בפייתון המחרוזות מוגדרות כאובייקטים מסוג `str`.
  - ▶ ניתן להציג מחרוזת בעזרת גרש או גרשיים, אבל היא צריכה להסתיים באותו הסימן שבו נפתחה:
- ▶ `'b'`
  - ▶ `"Happy Birthday $$$"`
  - ▶ `'''` - מחרוזת ריקה

# רווחים ותווים מיוחדים (Escape characters)

- ▶ בתוך מחרוזת יכולים להיות תווים מיוחדים:
- ▶ `\n` – סימן ירידה לתחילת השורה הבאה
- ▶ `\t` – מקש טאב, מסמן לרוב 8 רווחים או קפיצה לטור הבא
- ▶ `\b` – מקש Backspace, חזרה אחורה
- ▶ `\'` – תו גרש עצמו (כאשר הוא חלק מהמחרוזת)
- ▶ `\\` - מקש ה-Backslash עצמו

```
>>> print ("a\tb")  
a      b
```

```
>>> print ("ok\n\'ok\'")  
ok  
'ok'
```

▶ למשל:

# המרה בין מספרים למחרוזות

```
>>> a = "5"
>>> type (a)
<class 'str'>
>>> a1 = int(a)
>>> a1
5
>>> type (a1)
<class 'int'>
>>> b = "-4.6"
>>> b
'-4.6'
>>> float (b)
-4.6
```

- ▶ ניתן להמיר בין מספרים ומחרוזות
- ▶ אם הערך מופיע בגרשיים, זו מחרוזת
- ▶ הפונקציה type כותבת את סוג הערך
- ▶ הפונקציות int, float, str מחזירות ערך (אובייקט) מהטיפוס של שם הפונקציה

# בדיקה - מה מוכל במחרוזת

- ▶ fruit = "Red Apple"
- ▶ color = "Red"
- ▶ If color in fruit:
  - ▶ print ("Bingo – Red fruit")
- ▶ else:
  - ▶ print ("Not a red fruit")
- ▶ **Bingo – Red fruit**



# גישה לתווים בודדים או חלק ממחרוזת

- ▶ ניתן לגשת לתו בודד או לחלק ממחרוזת בעזרת סוגריים מרובעים ומספרים סידוריים. התו הראשון במחרוזת הוא במיקום 0, לאחריו תו 1 וכן הלאה.
- ▶ התו האחרון במחרוזת (לא משנה מה אורכה) יהיה במיקום -1, השני מהסוף -2 והלאה.

▶ >>> a = "Encyclopedia"

▶ >>> a[0]

▶ 'E'

▶ a[2:4]

▶ 'cy'

# הצגת קצה ימני של מחרוזת

ניתן להציג גם תווים מהקצה הימני של מחרוזת: ▶

- ▶ Name = "Hermione"
- ▶ print (Name[-1])
- ▶ 'e'
- ▶ print (Name[-3:-2])
- ▶ 'o'
- ▶ print (Name[-3:])
- ▶ 'one'

# בדיקת אורך של מחרוזת

ניתן לקבל אורך של מחרוזת בעזרת הפעלת הפונקציה `len` על המחרוזת (ערך קבוע או משתנה):

- ▶ `Name = "Hermione"`
- ▶ `len(Name)`
- ▶ **8**
- ▶ `len("")`
- ▶ **0**
- ▶ `len("Encyclopedia")`
- ▶ **12**



# חיבור או שרשור מחרוזות

ניתן לחבר מחרוזות אחת לשניה, זה נקרא שרשור מחרוזות, באמצעות סימן פלוס:

- ▶ `a, b = "Good", "Enough"`
- ▶ `print (a+b)`
- ▶ `GoodEnough`
- ▶ `a += b`
- ▶ `print (a)`
- ▶ `GoodEnough`

# הכפלת מחרוזות

▶ ניתן לחבר מחרוזת עם עצמה מספר פעמים, כלומר לחזור מספר פעמים ברצף על אותו ביטוי, בעזרת כפל בין מחרוזת למספר שלם:

- ▶  $Hi = \text{'Hi Guy'}$
- ▶  $Hi * 5$
- ▶  $\text{'Hi Guy Hi Guy Hi Guy Hi Guy Hi Guy'}$

# פורמט מחרוזות בעזרת F-strings

- ▶ ניזכר בפונקציה `print()`, היא יודעת להדפיס מרכיבים מטיפוסים שונים, מספרים, מחרוזות וביטויים אחרים, ולחבר אותם לרצף אחד, למעשה למחרוזת אחת שמודפסת על המסך. כמו בדוגמה הזו שראינו בסרטון 1.4:
- ▶ `print ("Your account balance is", num1, "dollars.")`
- ▶ נניח שנרצה לעשות בעצמנו דבר דומה למה שעושה `print`, לייצר מחרוזת שמורכבת מרצף של טקסט וערכים שמגיעים ממשתנים או ביטויים מסוג שונה. איך נוכל לעשות זאת?
- ▶ דרך אחת היא להפעיל את הפונקציה `str()` על ביטויים שהם לא מחרוזת, לקבל מחרוזות ואז לחבר אותן. אבל בפייתון מודרנית יש דרך נוחה יותר:

# F-strings (formatted string literal)

▶ דרך מודרנית להרכבת מחרוזות שנוספה בפייתון 3.6 נקראת F-strings.

▶ מחרוזות כאלה מזוהות באות F (קטנה או גדולה) לפני תחילת המחרוזת. דרך זו מאפשרת לשתול במחרוזת עצמה סוגריים מסולסלים שמוחלפים בערכים, כאשר הערכים מסוגים שונים רשומים בתוך הסוגריים. למשל:

```
>>> name, age = "Fred", 50
```

```
>>> f'My name is {name}, my age next year is {age+1}' ▶
```

```
'My name is Fred, my age next year is 51'
```

# F-strings – דוגמה נוספת

נראה להרכיב משפט בו הדובר מציג את שמו ואת שמות בנותיו, ולשמור אותו  
במשתנה introduction:

- ▶ name = 'Fred'
- ▶ girl1, girl2, girl3 = 'Sara', 'Dana', 'Tamar'
- ▶ introduction = f" My name is {name}, and my girls\' names are: \
- ▶ {girl1}, {girl2} and {girl3}."
- ▶ print (introduction)
- ▶ My name is Fred, and my girls' names are: Sara, Dana and Tamar.