

## סיכום שיעור 2

### תזכורת מהשיעור הקודם

#### פונקציית הקלט

בפייתון אנו יכולים לקלוט ישירות לתוך משתנה בצורה הבאה:

<pre>1 x = input("Please enter a number: ") 2 print("x is: ", x) 3</pre>	<pre>Please enter a number: 12 x is: 12 &gt;</pre>
--	--

בדוגמא הזו תודפס למסך המחרוזת: *Please enter a number:* ואז התוכנית תחכה למשתמש שיכניס קלט, הקלט שהמשתמש יכניס יכנס לתוך המשתנה *x*, לאחר מכן יודפס הקלט בעזרת הפונקציה *print*.

#### מחרוזות

בשפת C שלמדנו ישנו טיפוס הנקרא *char*, טיפוס זה יכול להכיל **תו אחד בלבד**. בפייתון אין את הטיפוס *char* אך יש טיפוס "חזק" ממנו והוא טיפוס המחרוזת (*string*) או בקיצור בפייתון (*str*). *str* היא טיפוס של משתנה המכיל **מחרוזת** של תווים, לדוגמא:

```
s = "Hello world"
```

בדוגמא זו אנו רואים הצהרה על משתנה בשם *s* מסוג *str* המכיל בתוכו את המחרוזת *Hello world*.

#### חשוב לזכור

- ניתן להצהיר על מחרוזת על ידי גרשיים כפולות ועל ידי גרשיים בודדים.
- המקבילה ל *char* בפייתון היא בעצם *string* המכילה מחרוזת המורכבת מתו בודד.

כדי לדעת את **אורך המחרוזת** נשתמש בפונקציה *len* בצורה הבאה:

<pre>1 s = "hello world" 2 print(len(s))</pre>	<pre>11 &gt;</pre>
--	--------------------

בדוגמא זו אנו רואים הצהרה על משתנה בשם *s* מסוג *str* המכיל בתוכו את המחרוזת *hello world*, לאחר מכן אנו קוראים לפונקציה *len* ומכניסים לתוכה את *s*, הפונקציה מוצאת את אורך המחרוזת (אורך המחרוזת הוא 11 כיוון שישנם 10 אותיות ותו רווח אחד), לבסוף הפונקציה *print* מדפיסה את האורך.

כמו שלמדנו בשפת C במחרוזת לכל אות מיקום משלה, האות הראשונה נמצאת בתא ה-0. כדי לגשת לאיבר במחרוזת נרשום את שם המחרוזת ובסוגריים מרובעים את המיקום שאליו נרצה לגשת, לדוגמא:

<pre>1 s = "Hello world" 2 print("The first char of the string is: ", s[0]) 3 print("The second char of the string is: ", s[1]) 4</pre>	<pre>The first char of the string is: H The second char of the string is: e &gt;</pre>
---	--

## פונקציית *type*

ניתן לדעת את טיפוס המשתנה על ידי שימוש בפונקציה *type*, לדוגמא:

<pre>1 x = 10 2 print("The type of x is", type(x)) 3 y = 12.6 4 print("The type of y is", type(y)) 5 s = "Hello world" 6 print("The type of s is", type(s)) 7</pre>	<pre>The type of x is &lt;class 'int'&gt; The type of y is &lt;class 'float'&gt; The type of s is &lt;class 'str'&gt; &gt;</pre>
---	--

בדוגמא זו אנו רואים הצהרה על משתנה *x* ואני יכולים לראות שהסוג שלו הוא *int* מכיוון שהוא מכיל את המספר השלם 10, בשורה הבאה אנו מדפיסים את סוג המשתנה של *x*. (ישנן שורות נוספות גם ל *float* ול-*str*).

## פונקציית הקלט – המשך

כברירת מחדל, מה שחוזר מפעולת הקליטה (**סוג המשתנה שנכנס למשתנה שאליו אנו קולטים**) הוא **מחרוזת**!

אז מה נעשה אם נרצה להתייחס אל המחרוזת שקיבלנו כאל מספר שלם לדוגמא?

נצטרך לבצע **המרה**. המרה היא בעצם החלפה של טיפוס משתנה לטיפוס משתנה אחר, לדוגמא:

<pre>1 x = input("Please enter a number: ") 2 print("The type of x is: ", type(x)) 3 y = int(input("Please enter a number: ")) 4 print("The type of y is: ", type(y)) 5</pre>	<pre>Please enter a number: 2 The type of x is: &lt;class 'str'&gt; Please enter a number: 3 The type of y is: &lt;class 'int'&gt; &gt;</pre>
---	---

בדוגמא זו אנו קולטים מהמשתמש מחרוזת לתוך משתנה *x* ולאחר מכן אנו מדפיסים את סוג המשתנה של *x* ואנו רואים כי הוא אכן מכיל **מחרוזת**. לאחר מכן אנו קולטים מהמשתמש מחרוזת נוספת אך מבצעים **המרה שלה ל *int*** ולכן כאשר אנו מדפיסים את סוג המשתנה של *y* אנו רואים כי הוא ***int***!

## תנאים

מילת ההתניה היא *if* כמו בשפת C. לתנאי יכולים להיות שני ערכים, אמת או שקר – *True* או *False*.

אם התנאי שכתבנו **כן מתקיים** – ערך התנאי הוא **אמת – *True***.  
אם התנאי שכתבנו **לא מתקיים** – ערך התנאי הוא **שקר – *False***.

אם התנאי מתקיים, ההוראות שיבוצעו יהיו ההוראות אחרי ה *if* אם התנאי לא מתקיים ההוראות שיבוצעו יהיו ההוראות שאחרי ה *else*. נראה דוגמא לתנאי בפיתון:

```
1 x = 1
2 if x == 2:
3     print("x = 2")
4 else:
5     print("x != 2")
6
```

x != 2  
> |

בדוגמא אנו רואים הצהרה על משתנה בשם *x* המכיל את המספר השלם 1.  
לאחר מכן אנו בודקים אם (*if*) התנאי *x == 2* מתקיים, הוא לא מתקיים ולכן הודפסה המחרוזת *x != 2*.

### חשוב לזכור:

- בסיום *if* וגם בסיום *else* יש נקודתיים.
- כל ההוראות לביצוע יכתבו **בהזחה** של *tab*.
- בשונה משפת C, בפיתון **לא** מכניסים את התנאי לתוך סוגריים.