יחידה 7

לולאות

שני פרקים, 12 דקות. while, break, continue, for ו-range.

מפת היחידה

12 חלקים ▲

פרק	דקה	מה קורה שם
7.1	0:31	"כל עוד": הרעיון של `while`
7.1	1:05	איטרציה, ומתי הלולאה נעצרת
7.1	1:36	מונה מ-0 עד 4
7.1	2:10	לולאה אינסופית
7.1	**2:40**	**הקלט שלא נגמר: שגיאה לוגית**
7.1	**4:49**	**`break` ו- `continue`**
7.2	0:30	שלוש פעמים אותו קוד
7.2	**1:32**	**`for`**
7.2	2:34	המשתנה אחרי הלולאה
7.2	2:50	`for` על מחרוזת, ועל רשימה של רשימות
7.2	3:42	אותו דבר ב- `while` וב- `for`
7.2	**4:14**	**`range`**

7.1 · לולאת while

`while` חוזרת על בלוק **כל עוד** תנאי מתקיים. כמו דוב בתרדמת חורף: כל עוד מזג האוויר חורפי, הוא ממשיך לישון. מספר הסבבים לא ידוע מראש.

```
>>> i = 0
>>> while i < 5:
...     print(i)
...     i += 1
...
```

```
0
1
2
3
4
```

- כל סבב נקרא **איטרציה**
- **לפני** כל איטרציה פייתון בודקת את התנאי. `False` : הלולאה נגמרת. `True` : הבלוק רץ, וחוזרים לבדוק
- `i += 1` הוא קיצור של `i = i + 1` . בלעדיו `i` נשאר 0, והלולאה לא נגמרת

לולאה אינסופית

```
while True:
    handle_next_request()
```

לפעמים בכוונה: שרת שאמור לעבוד כל הזמן. **הרבה יותר פעמים בטעות**, בגלל תנאי שלא יכול להפוך ל- `False` . מפסיקים תוכנית תקועה עם Ctrl+C.

הקלט שלא נגמר

התוכנית שואלת כמה חודשים בשנה דובי גריזלי ישנים, וממשיכה לשאול עד שעונים 5, 6 או 7:

```
user_input = input("Hibernation months? ")
while (user_input != 5 and user_input != 6
       and user_input != 7):
    print(user_input)
    user_input = input("Try again: ")
```

עונים 7, ושוב נשאלים. עונים 5, ושוב. **הלולאה לא נגמרת אף פעם**, כי `input` מחזיר מחרוזת, ו- `"7"` אף פעם לא שווה ל- `7` . התיקון:

```
user_input = int(input("Hibernation months? "))
while (user_input != 5 and user_input != 6
       and user_input != 7):
    print(user_input)
    user_input = int(input("Try again: "))
```

זו שגיאה לוגית. אין הודעת שגיאה ואין קריסה. התוכנית פשוט עושה משהו אחר ממה שהתכוונתם. ולכן היא הכי קשה למצוא.

`break` ו- `continue`

	מה עושה
`break`	יוצא מהלולאה **מיד**, בלי עוד איטרציות
`continue`	מדלג על **שאר האיטרציה הנוכחית**, וממשיך לבאה

```
>>> i = 0
>>> while i < 10:
...     i += 1
...     if i % 2 == 0:
...         break
...     print(i)
...
1
```

`%` הוא שארית חלוקה. `i % 2 == 0` פירושו "`i` זוגי". הלולאה הדפיסה `1` , ועצרה כש- `i` הגיע ל-2.

```
>>> i = 0
>>> while i < 10:
...     i += 1
...     if i % 2 == 0:
...         continue
...     print(i)
...
1
3
5
7
9
```

אותו קוד עם `continue` : במספר זוגי ה- `print` מדולג, והלולאה ממשיכה.

7.2 · לולאת for

הבעיה

```
animals = ["bear", "husky", "reindeer"]
animal = animals[0]
print(animal)
```

```
animal = animals[1]
print(animal)
animal = animals[2]
print(animal)
```

אותן שתי שורות שלוש פעמים, ורק המיקום משתנה. "עבור כל חיה ברשימה, הדפס אותה" כתוב בפייתון כמעט באותן מילים:

```
>>> animals = ["bear", "husky", "reindeer"]
>>> for animal in animals:
...     print(animal)
...
bear
husky
reindeer
```

`for` , שם משתנה, `in` , ורצף. **`in` כאן הוא חלק מהלולאה, ולא הבדיקה מיחידה 2**: בכל איטרציה הוא משים ל- `animal` את האיבר הבא.

המשתנה נשאר

```
>>> animal
'reindeer'
```

אחרי הלולאה המשתנה עדיין קיים, ומצביע על האיבר האחרון.

על כל רצף

מחרוזת היא רצף, ולכן `for` עובר עליה תו אחרי תו:

```
>>> for letter in "bear":
...     print(letter)
...
b
e
a
r
```

ורשימה של רשימות:

```
>>> animal_sounds = [["dog", "woof"],
...                  ["cat", "meow"],
...                  ["cow", "moo"]]
```

```
>>> for pair in animal_sounds:
...     print(pair[0], pair[1])
...
dog woof
cat meow
cow moo
```

אותו דבר, שלוש דרכים

```
>>> num = 0
>>> while num < 5:
...     print(num, end=" ")
...     num += 1
...
0 1 2 3 4
>>> for num in [0, 1, 2, 3, 4]:
...     print(num, end=" ")
...
0 1 2 3 4
>>> for num in range(5):
...     print(num, end=" ")
...
0 1 2 3 4
```

(`end=" "` מדפיס רווח במקום שורה חדשה, כדי שהפלט ייכנס בשורה אחת.)

`range`

`range` מייצרת סדרת מספרים, בלי לכתוב רשימה:

	פירוש	מתקבל
`range(5)`	מ-0 עד 5, **בלי 5**	0 1 2 3 4
`range(5, 10)`	מ-5 עד 10	5 6 7 8 9
`range(0, 10, 3)`	מ-0 עד 10, בקפיצות של 3	0 3 6 9
`range(5, 0, -1)`	קפיצה שלילית	5 4 3 2 1

```
>>> list(range(0, 10, 3))
[0, 3, 6, 9]
```

```
>>> list(range(5, 0, -1))
[5, 4, 3, 2, 1]
```

`range(start, stop, step)` : `start` ברירת מחדל 0, `step` ברירת מחדל 1, ו- `stop` **לא נכלל**. בדיוק כמו בחיתוך.

while או for

	while	for
מתי	לא יודעים מראש כמה פעמים	עוברים על רצף, או מספר פעמים ידוע
דוגמה	עד שהמשתמש יענה נכון	כל איבר ברשימה
סכנה	לולאה אינסופית	כמעט אין

בדיקה עצמית

1. מה ידפיס `for x in range(2, 8, 2): print(x)` ?
2. למה `while i < 5: print(i)` (בלי שום שורה נוספת) לא נגמרת?
3. מה ההבדל בין `break` ל- `continue` ?
4. אחרי `for c in "abc": pass` , על מה מצביע `c` ?
5. איך מדפיסים את המספרים מ-10 עד 1?

תשובות. 1: `2` , `4` , `6` . `8` לא נכלל. 2: `i` לא משתנה, ולכן התנאי נשאר `True` . 3: `break` יוצא מהלולאה. `continue` מדלג רק על האיטרציה הנוכחית. 4: `'c'` , האיבר האחרון. 5: `for n in range(10, 0, -1): print(n)` .

סיכומים אישיים מתוך סרטוני הקורס Self.py בקמפוס IL, ולא חומר רשמי של הקורס.