

דו"ח הכנה - ניסוי 118

נדב אשכנזי – 313326811

איתי סלפיאן – 316296482

שאלה 1

הפעילו פייתון באופן אינטראקטיבי, ובצעו את החישובים להלן. שרטטו טבלא ורשמו לכל חישוב (1) את הביטוי שכתבתם, שימו לב שלעתים אתם נדרשים לסוג משתנה ולסוג הצבה מסויימים, (2) את התוצאה שחישב המחשב, (3) ואת הטיפוס של התוצאה.

1. שלוש וחצי (נקודה צפה) ועוד ארבע (שלם).
2. 15_8 (אוקטאלי) ועוד 255_{16} (הקסדצימאלי).
3. 999999999 (תשע תשעיות) כפול עשר.
4. $(1+2j) * 2$
5. $1 + 2j * 2$

סוג	תוצאה	ביטוי
float	7.5	$3.5 + 4$
int	610	$(0o15 + 0x255)$
Py3 – int py2- long	Py3 -9999999990 Py2 - 9999999990L	$999999999*10$
complex	$2+4j$	$(1+2j)*2$
complex	$1+4j$	$1+2j*2$

שאלה 2:

1. מה השגיאה המתקבלת במקרה הקודם בו ניסינו לשנות את המחרוזת?
2. מה התוצאה שידפיס print בדוגמא האחרונה?
3. מה קורה אם רושמים מספר שלילי בטווח? בידקו.

1.

```
>>> S = "grail"
>>> S[4] = "n"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
```

2. grain

3. מספר שלילי מייצג אינדקס מהסוף, אם כותבים לדוג [0:-1] נקבל את כל המילה ללא האות האחרונה :

```
>>> S[0:-1]
'grai'
```

```
>>> count
[1, 2, "O'Clock", 4]
>>> count[5] = 9
```

תקלה!

שאלה 3

מה השגיאה המתקבלת בדוגמא האחרונה?

```
>>> count = [1, 2, "O'Clock", 4]
>>> count[5] = 9
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list assignment index out of range
```

שאלה 4

רשמו סדרת פקודות כדלקמן: (1) הצבת מטריצה 3×3 בעלת הערכים 1-9 במשתנה M; (2) הצגת המטריצה; (3) החלפת האיבר האמצעי במלה "Robin"; (4) הצגת התוצאה.

```
>>> M = [[1,2,3],[4,5,6],[7,8,9]]
>>> print(M)
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>> M[1] = "Robin"
>>> print(M)
[[1, 2, 3], 'Robin', [7, 8, 9]]
```

שאלה 5

בנו רשימה המכילה שני מילונים, כל אחד בעבור מכונית אחרת, עם השדות הנ"ל.

```
>>> cars = [{'Model': 'Fabia', 'Maker': 'Skoda', 'Year': 2006}, {'Color': 'Gray', 'Model': 'Fabia', 'Maker': 'Skoda', 'Year': 2006}]
>>> print(cars)
[{'Model': 'Fabia', 'Maker': 'Skoda', 'Year': 2006}, {'Color': 'Gray', 'Model': 'Fabia', 'Maker': 'Skoda', 'Year': 2006}]
```

```
>>> count = (1, 2, 3, "O'Clock")
>>> count
(1, 2, 3, "O'Clock")
>>> len(count)
4
>>> count + (4, "O'Clock")
(1, 2, 3, "O'Clock", 4, "O'Clock")
>>> count[2] = 5
```

תקלה!

שאלה 6

מה השגיאה המתקבלת בדוגמא האחרונה?

```
>>>
>>>
>>> count = (1, 2, 3, "O'Clock")
>>> len(count)
4
>>> count + (4, "O'Clock")
(1, 2, 3, "O'Clock", 4, "O'Clock")
>>> count[2] = 5
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
```

```
x = 15
if x < 0:
    print "x is negative."
elif x < 10:
    print "x is a small number."
elif x < 20:
    print "mmm..."
    print "Maybe x is not so small."
else:
    print "x is large!"
```

שאלה 7

מה תדפיס התוכנית הנ"ל? ומה אם נציב בשורה הראשונה 100 במקום 15?

1.

mmm...

Maybe x is not so small.

2.

x is large!

דוגמא:

```
x = range(7)
while x:                                # While x is not empty
    if x[0] > 4:
        break
    x = x[1:]
else:
    print "All values are not greater than 4."
print "Done."
```

שאלה 8

כמה שאלות:

1. מה מדפיסה התוכנית הנ"ל?
2. מה מדפיסה התוכנית אם בשורה הראשונה נחליף את 7 ב-4?
3. מה מדפיסה התוכנית אם בשורה הראשונה נחליף את 7 ב-5?
4. מה עושה הפונקציה range?

1.

Done.

2.

All values are not greater than 4.

Done.

3.

All values are not greater than 4.

Done.

4. מספקת אובייקט מסוג range שמייצג טווח ערכים, כיון שסיפקנו פרמטר אחד זה יהיה סוף הטווח (לא כולל), התחלתו היא 0 בדיפולט וערך הקפיצה הדיפולטי הוא 1.

```
>>> type(range(10))
<class 'range'>
```

קיים סינטקס range(a,b,c) כאשר a האיבר הראשון בטווח b סוף הטווח (לא כולל) וc ערך הקפיצה. ע"י המרה לlist ניתן לראות את ערכי הטווח:

```
>>> list(range(1,10,2))
[1, 3, 5, 7, 9]
```

```
for x in [1, 2, 3, 4, 5, 6, 7]:
    if x%2 == 0:
        print x, " is even."
    else:
        print x, " is odd."
```

שאלה 9

מה מדפיסה התוכנית לעיל?

1 is odd.
2 is even.
3 is odd.
4 is even.
5 is odd.
6 is even.
7 is odd.

התבוננו בפקודות הבאות:

```
>>> txt = open("stam.txt")
>>> lines = txt.readlines()
>>> lines = [line.upper() for line in lines]
>>> txt.close()
>>> txt = open("upStam.txt", "w")
>>> txt.writelines(lines)
>>> txt.close()
```

שאלה 10

הסבירו מה עושה כל שורה בתוכנית הנ"ל. מה מכיל הקובץ upStam.txt עם סיום ריצתה?

```
1 txt = open("stam.txt") # opens stam.txt file and provides reference to file object
2 lines = txt.readlines() # separates the text to a list of lines (strings) and assigns it into <lines> list object
3 lines = [line.upper() for line in lines] # uppercases each string in <lines> list
4 txt.close() # closes the file that was opened in line 1 and deletes the references provided
5 txt = open("upStam.txt", "w") # opens "upStam.txt" file with write privileges (or creates one if not existing) and provides reference to file object
6 txt.writelines(lines) # write all strings in <lines> list object in order into upStam.txt file
7 txt.close() # closes upStam.txt file
```

בסוף התוכנית בקובץ upStam.txt הוא להיות אותו תוכן של stam.txt אבל כולו באותיות גדולות.

שאלה 11

איזה תת מחרוזות מתאימות לביטויים הבאים עבור המחרוזת הנ"ל: (רשמו טבלא כמו הטבלא לעיל)

1. an
2. \ban
3. \Ban

an	This is <u>an</u> island
\ban	This is <u>an</u> island
\Ban	This is an island

שאלה 12

רשמו ביטויים רגולריים למציאת המחרוזות הבאות:

1. מספר טלפון עם או בלי קידומת, כאשר מקף מפריד בין הקידומת לטלפון, וללא מקפים נוספים, כלומר 04-1234567 או 1234567.
2. כתובת אינטרנט (URL) המתחילה ב-http://, ואחריו שניים או יותר שמות מופרדים בנקודות, כאשר כל שם מורכב מאותיות abc בלבד.
כך: http://name1.name2.name3.name4.com
כך: http://name1.name2
לא: http://name1..name2
לא: http://name1
לא: name1.name2.name3

1. $\{([0-9]\{2\})-([0-9]\{7\})\}$

2. $^{\wedge}\text{http:} // ([a-z] + \.) + [a-z] +$

שאלה 13

1. השתמשו בכלים שבידכם כעת כדי לרשום את הביטוי הנכון למציאת טקסט המוקף בתגים מתאימים. עבור הדוגמא לעיל התוצאה אמורה להיות (3 התאמות, כל אחת מסומנת בקו תחתי):
cut down the mightiest tree **in the forest** with...
a **herring**
2. בידנו מסד נתונים עם רשומות שקריות. נרצה לנצל את הידע החדש שלנו בביטויים רגולריים כדי לזהות רשומות חשודות.
 - 2.1. רשמו ביטוי לזיהוי כתובת אימייל בה המחרוזות לפני ואחרי הכרוכית (שטרודל) זהות.
 - 2.2. רשמו ביטוי לזיהוי שם מלא (ראשון + שני + משפחה) ששלושת חלקיו זהים.
 - 2.3. רשמו ביטוי לזיהוי שם מלא (ראשון + שני + משפחה) בו שם המשפחה זהה לשם הראשון או לשני.

1. $<(\wedge)>.\wedge?<\wedge1>$

2. $(\wedge w+)\@ \wedge 1$

3. $(\wedge w+)\ \wedge 1\ \wedge 1$

4. $(\wedge w+)\ (\wedge w+)\ (\wedge 1\wedge 2)$