

פייתון

שיעור 3: משתנים, תנאים ולולאות

סוגי משתנים בפייתון

- ▶ קיימים מגוון סוגים של משתנים
 - הפקודה `type()` מציגה את סוג המשתנה
- ▶ נלמד מהקל אל הכבד

```
>>> type(True)
<type 'bool'>
>>> type(2)
<type 'int'>
>>> type(2.0)
<type 'float'>
>>> type('Hi')
<type 'str'>
```

```
>>> type([1, 2, 3])
<type 'list'>
>>> type((1, 2, 3))
<type 'tuple'>
>>> type({1:'a', 2:'b', 3:'c'})
<type 'dict'>
```



```
>>> True+1  
2
```

- ▶ משתנה בוליאני boolean יכול לקבל רק אחד משני ערכים:
 - True
 - False
- ▶ True מתורגם ל-1
 - False מתורגם ל-0
 - עם זאת נהוג להשוות דברים ל-True או False

int, float

```
>>> a=2
>>> b=2.0
>>> c=a+b
>>> c
4.0
>>> d=b/a
>>> d
1.0

>>> e = 2.3
>>> type(e)
<type 'float'>
```

- ▶ int - מספר שלם
- ▶ float - מספר שכולל נקודה עשרונית
- ▶ תוצאה של פעולות חשבון בין int ל-float תהיה float
- ▶ תוצאה של פעולות בין intים יכולה לעיתים להיות לא מדוייקת
◦ המרה ל-float הכרחית

```
>>> 8 / 3
2
>>> float(8) / float(3)
2.6666666666666665
```

אי דיוק ב-float

```
>>> a = 0.1
>>> a + a
0.2
>>> _ + a
0.30000000000000004
>>> _ + a
0.4
>>> _ + a
0.5
>>> _ + a
0.6
>>> _ + a
0.7
>>> _ + a
0.7999999999999999
>>> _ + a
0.8999999999999999
>>> _ + a
0.9999999999999999
```

- ▶ כיוון שהמחשב מייצג מספרים float על ידי כמות סופית של ביטים, לעיתים יש הבדלים זעירים בין הערך ששמור במחשב לבין תוצאת החישוב הנכונה
- השגיאות מתרבות אם המחשב סוכם float פעמים רבות

- ▶ if: לאחר תנאי if יבוא ביטוי. אם הביטוי הוא True אז יבוצע הקוד שבבלוק שלאחר ה-if.
- האופרטור '==' בודק אם יש שוויון
 - האופרטור '!=' בודק אי שוויון

```
x = 21
if x == 21:
    print 'Yay!'
```

```
x = 20
if x != 21:
    print 'Yay!'
```

- ▶ סימן הנקודותיים בסוף ה-if- מורה על פתיחת בלוק חדש
- ▶ איך פייתון יודע מתי נגמר הבלוק?
 - כל השורות שבאותו בלוק מתחילות עם אותו מספר רווחים
 - מקובל 4 תווי רווח

```
x = 14
if x > 10:
    print 'x is bigger than 10'
    x += 1    # x=x+1
    print x
```

ארבעה תווי רווח } בלוק

```
x is bigger than 10
15
```

else: אם התנאי if אינו מתקיים, אפשר להריץ קוד אחר
תחת ההנחיה else.

```
x = 5
if x > 10:
    print 'x is bigger than 10'
else:
    print 'x is NOT bigger than 10'
```

```
x is NOT bigger than 10
```


▶ elif: קיצור של else-if. התנאי if אינו מתקיים אך אנחנו רוצים לבדוק תנאי אחר לפני הגעה ל-else

```
friend_is_free = False
mother_is_home = True
if friend_is_free:
    print 'Going to the movies'
elif mother_is_home:
    print 'Eating apple pie'
else:
    print 'Watching "Friends" on TV'
```

אפשר לבדוק תנאים גם עם is ▶

```
x = True
if x is True:
    print 'The truth!'
```

```
x = 5
if x is not 5:
    print 'Not true!'
```

מה ההבדל בין == ל- is? ▶

- == בודק אם הערכים זהים
- is בודק אם מצביעים לאותו מקום בזיכרון

https://www.youtube.com/watch?v=0_dQpUtcubM

תנאים לוגיים and, or, not

▶ בנוסף לתנאי if רגילים, אפשר ליצור גם תנאים לוגיים

and ◦

or ◦

not ◦

```
>>> x=3
>>> if (x>6) or (x==3):
        print 'something'
```

```
something
```

לולאת while

- ▶ כל עוד התנאי מתקיים, הלולאה תמשיך להתבצע
 - יתבצעו כל השורות בבלוק
 - בסוף הבלוק- חזרה להוראת ה-while

```
while condition:  
    # do something  
    # return to the while condition
```

▶ תרגיל:

- כיתבו לולאת while שמדפיסה את המספרים מ-0 עד 5 (כולל)

- ▶ אינה דומה ללולאות for בשפות אחרות
- ▶ מקבלת אוסף של איברים
- ▶ כל איטרציה (מעבר על הלולאה) מתייחסת רק לאיבר אחד באוסף, לפי הסדר

```
for i in [0, 1, 2]:  
    print i*2
```

```
0  
2  
4
```

range, xrange

```
>>> range(3, 8, 1)
[3, 4, 5, 6, 7]
>>> range(3, 8, 2)
[3, 5, 7]
>>> range(5)
[0, 1, 2, 3, 4]
```

- ▶ מה אם רוצים לעשות לולאת for שעוברת על סדרת מספרים?
- ▶ הפקודה range מקבלת התחלה, סוף וקפיצה ויוצרת סדרת מספרים
- ▶ הפקודה xrange היא זהה, אך לא שומרת את הסדרה בזיכרון - חסכוני

תרגיל 1 Baby



▶ כיתבו לולאת for שמדפיסה את
המספרים מ-1 עד 40 (כולל)
▶ השתמשו ב-xrange

▶ קרדיט: עומר רוזנבוים, שי סדובסקי

תרגיל 7-Boom

- ▶ הדפיסו למסך את כל המספרים בין 0 ל-100 שמתחלקים ב-7 ללא שארית או שמכילים את הספרה 7, לפי הסדר. השתמשו רק בפעולות חשבון!
 - טיפ: חלוקה- הסימן '/', שארית - הסימן '%'
- ▶ קרדיט: עומר רוזנבוים, שי סדובסקי



- ▶ לעיתים נרצה שהלולאה שלנו לא תבצע כלום
- ▶ הביטוי Pass משמעותו "אל תבצע כלום"
- ▶ תרגיל: הדפיסו את המספרים מ-1 עד 10, חוץ מאשר 7

```
for i in range(1,11):  
    if i == 7:  
        pass  
    else:  
        print i
```

- ▶ שימושי בתהליך כתיבת קוד – כתיבת פונקציות "ריקות", רק לצורך תכנון הקוד

▶ מה יקרה אם נריץ את הלולאה הבאה?

```
while True:  
    print 'Hi'
```

- ▶ הביטוי Break מאפשר לנו לקטוע ריצה של לולאה
- ▶ שימושי כאשר אנחנו לא יודעים מראש אחרי כמה איטרציות נרצה לעצור

תרגיל – Take a Break



- ▶ הדפיסו את כל המספרים בסדרת פיבונצ'י אשר קטנים מ-10,000. חובה להשתמש ב-`While True`.

תרגיל Baby Pampers



- ▶ הדפיסו למסך את כל המספרים מ-0 עד 5, בקפיצות של 0.1
- ▶ הדפיסו את המספרים השלמים ללא נקודה עשרונית, כמו בדוגמה שבצד.
- ▶ שימו לב: פקודת range אינה מקבלת צעד שאינו מספר שלם
- ▶ כאשר מחברים 0.1 שוב ושוב, לעיתים התוצאה אינה מדוייקת
- ▶ קרדיט: עומר רוזנבוים, שי סדובסקי

0

0.1

0.2

...

1

1.1

...

...

5