

פייתון

שיעור 4: מחרוזות

הגדרת מחרוזת

- ▶ מחרוזת: אוסף של תווים כלשהם
 - אפשר להגדיר עם גרש יחיד או גרשיים
 - תו character - מחרוזת באורך 1

```
>>> greeting = 'Hi what is your name?'  
>>> greeting  
'Hi what is your name?'
```

המחרוזת '1234' לא שווה למספר 1234 ▶

```
>>> 1234 + 5678
```

```
6912
```

```
>>> '1234' + 5678
```

```
Traceback (most recent call last):
```

```
  File "<pyshell#22>", line 1, in <module>
```

```
    '1234' + 5678
```

```
TypeError: cannot concatenate 'str' and 'int' objects
```

ביצוע `format` ל-`print`

```
>>> greet = 'My age is: '  
>>> age = 20  
>>> greet + ' ' + str(age)  
'My age is:  20'
```

מסורבל: ▶

קל: ▶

```
>>> print '{} {}'.format(greet, age)  
My age is:  20
```

String slice



▶ אפשר לחתוך חלקים ממחרוזת באופן הבא:

```
>>> my_str[start:stop:delta]
```

▶ החיתוך דומה לפונקציה range אותה
הכרנו:

- Start : אינדקס התחלה (ברירת מחדל 0)
- Stop : אינדקס סיום (ברירת מחדל סוף המחרוזת)
- Delta : כמה אינדקסים קופצים (ברירת מחדל 1)
- אפשר לשים גם ערכים שליליים- האינדקס האחרון הוא 1-, או קפיצה שלילית (אחורה)

slice - דוגמאות

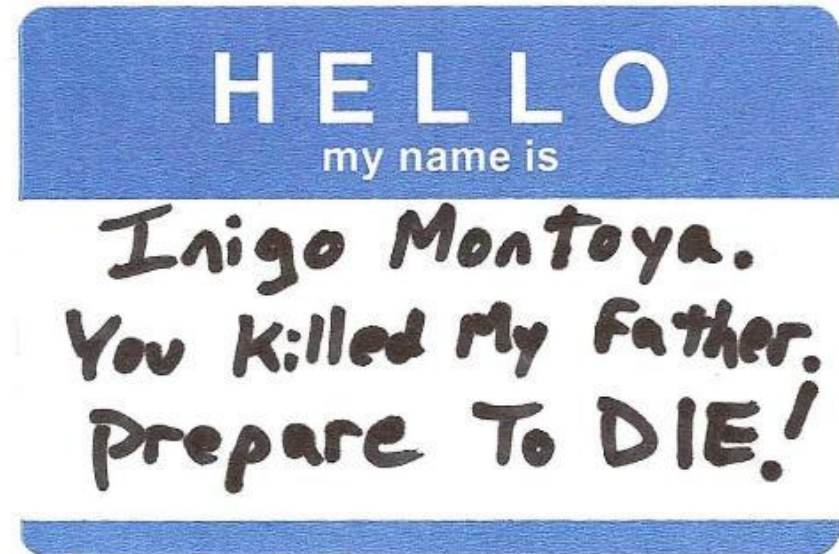
```
>>> name = 'Shrek'  
>>> name[1]  
'h'  
>>> name[1:3]  
'hr'  
>>> name[1:]  
'hrek'  
>>> name[1::2]  
'he'  
>>> name[:]  
'Shrek'  
>>> name[:-1]  
'Shre'
```



תרגיל - slicer

- ▶ כיתבו ב-psycharm תוכנית שמכילה משתנה מסוג string, בעל הערך "Hello, my name is Inigo Montoya".
- ▶ השתמשו רק ב-slice על המחרוזת והדפיסו את המחרוזות הבאות:

- "Hello"
- "my name"
- "Hlo ynm slioMnoa"
- "lo ynm sl"



◦ קרדיט: עומר רוזנבוים, שי סדובסקי

פקודות על מחרוזות- דוגמאות

מסקנה:
אם אנחנו צריכים
לבצע פעולה
כלשהי בפייתון,
לרוב נוכל למצוא
לכך פונקציה
קיימת

סימן + :
חיבור מחרוזות

:len
אורך המחרוזת

:upper
הצגה באותיות
גדולות

שימו לב- המחרוזת
לא השתנתה עקב
upper

:find
מיקום התו במחרוזת

```
>>> message = 'hello' + ' world'
>>> print message
hello world
>>> len(message)
11
>>> message.upper()
'HELLO WORLD'
>>> message
'hello world'
>>> message.find('o')
4
```


- ▶ איך אפשר למצוא פקודות שאנחנו לא מכירים?
- ▶ `dir` - מחזירה את כל מה ש"שייך" למשתנה מסויים

```
>>> message = 'Hello World'
>>> dir(message)
['_add_', '_class_', '_contains_', '_delattr_', '_doc_', '_eq_', '_format_', '_ge_', '_getattribute_', '_getitem_', '_getnewargs_', '_getslice_', '_gt_', '_hash_', '_init_', '_le_', '_len_', '_lt_', '_mod_', '_mul_', '_ne_', '_new_', '_reduce_', '_reduce_ex_', '_repr_', '_rmod_', '_rmul_', '_setattr_', '_sizeof_', '_str_', '_subclasshook_', '_formatter_field_name_split', '_formatter_parser', 'capitalize', 'center', 'count', 'decode', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'index', 'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

- ▶ איך אפשר לדעת מה פקודת עושה?
- ▶ הפונקציה help

```
>>> help(message.count)
Help on built-in function count:
```

```
count(...)
    S.count(sub[, start[, end]]) -> int
```

Return the number of non-overlapping occurrences of substring sub in string S[start:end]. Optional arguments start and end are interpreted as in slice notation.

```
>>> message.count('el')
1
```

צירופי תווים מיוחדים

Escape Sequence	Meaning
<code>\newline</code>	Ignored
<code>\\</code>	Backslash (\)
<code>\'</code>	Single quote (')
<code>\"</code>	Double quote (")
<code>\a</code>	ASCII Bell (BEL)
<code>\b</code>	ASCII Backspace (BS)
<code>\f</code>	ASCII Formfeed (FF)
<code>\n</code>	ASCII Linefeed (LF)
<code>\r</code>	ASCII Carriage Return (CR)
<code>\t</code>	ASCII Horizontal Tab (TAB)
<code>\v</code>	ASCII Vertical Tab (VT)
<code>\ooo</code>	ASCII character with octal value <i>ooo</i>
<code>\xhh...</code>	ASCII character with hex value <i>hh...</i>

במחרוזות קיימים צירופים מיוחדים

- לדוגמה אם נזין למחרוזת את הצירוף `\n` לא יודפס התו `\n` ואחריו התו `n` אלא תבוצע ירידת שורה
- להלן טבלה של הצירופים המיוחדים
- לקריאה נוספת:
- <https://docs.python.org/2.0/ref/strings.html>

הגדרת מחרוזת- תרגיל

▶ ומה אם יש לנו טקסט שכולל צירוף מיוחד ואנחנו רוצים שהוא יודפס?

◦ לדוגמה שם של ספריה במחשב:

```
print 'c:\cyber\number1\b52'
```

```
c:\cyber
```

```
umber52
```

◦ תרגיל: העזרו בטבלה שבשקף הקודם ומיצאו איך להדפיס את השם נכון.

- ▶ כפי שראינו כאשר מגדירים מחרוזת צירופי תווים מיוחדים מקבלים פרשנות אחרת
- ▶ ההנחיה r (קיצור של raw) אומרת לפייתון "קח את התווים כמו שהם, תתעלם מצירופים מיוחדים"

```
>>> directory = r'cyber\number1\b52'  
>>> print directory  
cyber\number1\b52
```

בקישור הבא ניתן למצוא ריכוז של כל ה- string
:methods

<https://docs.python.org/2/library/stdtypes.html>

קבלת קלט מהמשתמש

▶ הפונקציה `raw_input` משמשת לקליטת קלט מהמשתמש והכנסתו למשתנה מסוג מחרוזת

```
username = raw_input('Please enter your name\n')  
print 'Hello {}'.format(username)
```

```
Please enater your name  
Kalista  
Hello Kalista
```



תרגיל- אבניבי

‣ זוכרים את שפת הבית?

‣ כיתבו קוד שקולט משפט (באנגלית) מהמשתמש ומתרגם אותו לשפת הבית

‣ תזכורת: אחרי האותיות aeiou צריך להוסיף b ולהכפיל את האות לדוגמה: עבור הקלט ani ohev otach יודפס:

abanibi obohebev obotabach



תרגיל מודרך – donuts



- ▶ תרגיל מתוך google classes
- ▶ נתון הקובץ string1.py
- ▶ המשימה: לשכתב את הפונקציה donuts כך שהרצה שלה תיתן את התוצאות הנדרשות
 - החזרת ערך מפונקציה: הפקודה return ולאחריה הערך
- ▶ בתוך main מבוצעות קריאות לפונקציה, עם פרמטרים שונים, והתוצאות מושוות לתוצאות הנדרשות

```
# A. donuts
# Given an int count of a number of donuts, return a string
# of the form 'Number of donuts: <count>', where <count> is the number
# passed in. However, if the count is 10 or more, then use the word 'many'
# instead of the actual count.
# So donuts(5) returns 'Number of donuts: 5'
# and donuts(23) returns 'Number of donuts: many'
def donuts(count):
    # +++your code here+++
    return
```

פתרון - donuts

```
def donuts(count):  
    if count < 10:  
        return 'Number of donuts: ' + str(count)  
    else:  
        return 'Number of donuts: many'
```

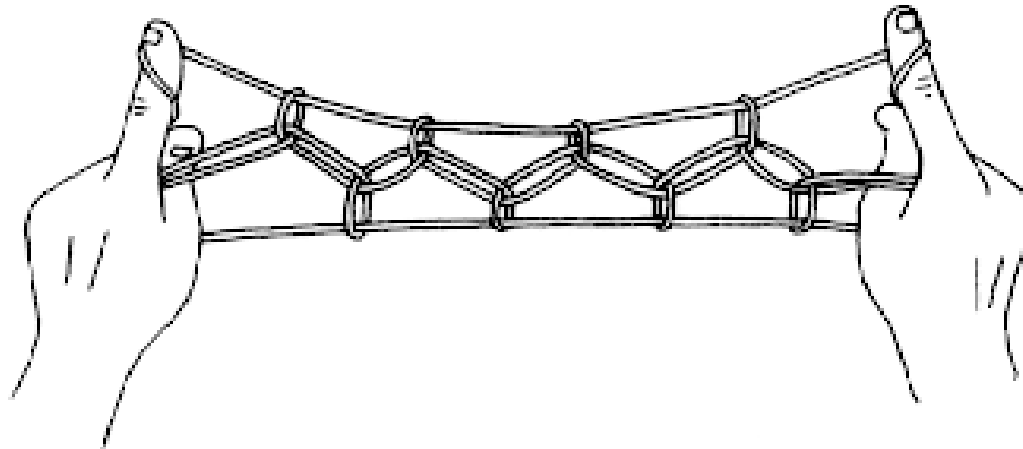
תוצאות ההרצה: ▶

```
donuts  
OK got: 'Number of donuts: 4' expected: 'Number of donuts: 4'  
OK got: 'Number of donuts: 9' expected: 'Number of donuts: 9'  
OK got: 'Number of donuts: many' expected: 'Number of donuts: many'  
OK got: 'Number of donuts: many' expected: 'Number of donuts: many'
```

Google

String1.py ▶

String2.py ▶



תרגיל מסכם- ז'אן ולז'אן



- ▶ כיתבו תוכנית שקולטת מהמשתמש מספר בעל חמש ספרות ומדפיסה:
 - את המספר עצמו
 - את ספרות המספר, כל ספרה בנפרד
 - את סכום הספרות של המספר

Please enter a 5 digit number

24601

You entered the number: 24601

The digits of this number are: 2,4,6,0,1

The sum of the digits is: 13

- ▶ הערות:
 - ניתן להניח שהמשתמש הכניס קלט תקין
 - שימוש בפונקציה `int`: הפקודה `int(number)` מחזירה את המספר שנמצא במחרוזת `number`

- ▶ קרדיט: עומר רוזנבוים, שי סדובסקי