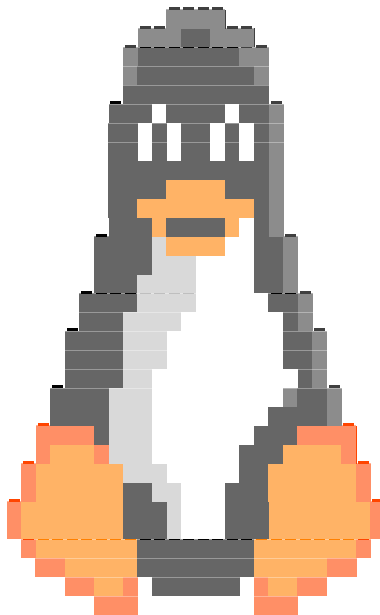

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΚΑΙ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΦΥΣ 140 Εισαγωγή στην Επιστημονική Χρήση Υπολογιστών (15821)
Χειμερινό Εξάμηνο 2023

Φώτης Πτωχός και Αλέξανδρος Αττίκης
Φροντιστήριο 4

3 Οκτωβρίου 2023
15:00 - 17:00



Φροντιστήριο 4

Παράδειγμα 1 Παράδειγμα για ανάγνωση και μετατροπή από *string* σε *int* και *float* αλλά και τη σωστή χρήση του τελεστή "+" για ένωση δύο αντικειμένων τύπου *string* (concatenation):

tutorial4/ex1.py

```
1  #!/usr/bin/python3
2  '''
3  USAGE:
4      chmod +x ex1.py
5      python3 ex1.py
6      script -q ex1.log python3 -i ex1.py
7
8
9  DESCRIPTION:
10 Introduction to strings (revisited)
11 '''
12
13 # Get user name and age, as strings!
14 name = input(">>> Enter your name: ")
15 age = input(">>> Enter your age: ")
16
17 # Concatenate the strings using the operator "+"
18 msg = " Hello, " + name + "!"
19
20
21 try:
22
23     # Convert age from string to an integer or float
24     age_int = int(age)
25     age_float = float(age)
26
27     # Print information to user. Convert ages from int/float back to str using
28     # concatenation
29     print("=== ", msg)
30     print("=== You are " + str(age_int) + " years old as an integer.")
31     print("=== You are " + str(age_float) + " years old as a float.")
32     print("=== In 10 years you will be " + str(int(age_int+10)) + " years old
33     as an integer.")
34     print("=== In 10 years you will be " + str(float(age_float+10)) + " years
35     old as a float.")
36 except: #ValueError:
37     print("=== Invalid age input ", age, ". Please enter a valid integer age!")
38
39 quit()
```

Αποτέλεσμα:

```
>>> Enter your name: John Bonham
```

```
>>> Enter your age: 3
=== Hello, John Bonham!
=== You are 3 years old as an integer.
=== You are 3.0 years old as a float.
=== In 10 years you will be 13 years old as an integer.
=== In 10 years you will be 13.0 years old as a float.
```

Στην Python, όταν προσθέτετε έναν ακέραιο σε ένα *float*, ο ακέραιος μετατρέπεται αυτόματα σε *float* και στη συνέχεια εκτελείται ο υπολογισμός. Το αποτέλεσμα θα είναι ένας αριθμός τύπου *float*.

Παράδειγμα 2 Παράδειγμα για τον έλεγχο χαρακτήρων μέσα σε ένα *string*:

tutorial4/ex2.py

```
1  #!/usr/bin/python3
2  '''
3  USAGE:
4      chmod +x ex2.py
5      python3 ex2.py
6      script -q ex2.log python3 -i ex2.py
7
8
9  DESCRIPTION:
10 Introduction to strings (revisited) - Demonstration that string is a tuple (
    immutable object)
11 '''
12
13 # Define a string to be examined in this example
14 msg = "Hello, World!"
15
16 # Access individual characters of the string using indexing
17 char1 = msg[0]
18 char2 = msg[-1]
19 msgLen = len(msg)
20
21 print("=== The first character is ", char1, ", the last character is", char2)
22
23 print("=== Loop over all characters in string using \"for\" and \"range()\": ")
24 for k in range(0, msgLen, +1):
25     m = "The character in position " + str(k) + " is " + msg[k]
26     print("\t", m)
27
28
29 print("=== Loop over all characters in string \"for\", \"in\" and \"enumerate\"
    : ")
30 for o, l in enumerate(msg, 0):
31     if o % 2==0:
32         m = "The character in position " + str(o) + " is " + msg[o]
33     else:
34         m = "The character in position " + str(o) + " is " + l
35     print("\t", m)
36
37 print("=== Loop over all characters in string \"for\" and \"in\": ")
38 for l in msg:
39     print(l)
40
41
42 # Attempt to change a character. This will raise an error!
43 try:
44     # Change the first character to something else (e.g. a lowercase 'h')
45     msg[0] = 'h'
46 except:
47     print("=== ERROR! Strings are immutable in Python! and do not support item
    assignment!")
```

```
48
49 quit()
```

Αποτέλεσμα:

```
=== The first character is H , the last character is !
=== Loop over all characters in string using "for" and "range()":
    The character in position 0 is H
    The character in position 1 is e
    The character in position 2 is l
    The character in position 3 is l
    The character in position 4 is o
    The character in position 5 is ,
    The character in position 6 is
    The character in position 7 is W
    The character in position 8 is o
    The character in position 9 is r
    The character in position 10 is l
    The character in position 11 is d
    The character in position 12 is !
=== Loop over all characters in string "for", "in" and "enumerate":
    The character in position 0 is H
    The character in position 1 is e
    The character in position 2 is l
    The character in position 3 is l
    The character in position 4 is o
    The character in position 5 is ,
    The character in position 6 is
    The character in position 7 is W
    The character in position 8 is o
    The character in position 9 is r
    The character in position 10 is l
    The character in position 11 is d
    The character in position 12 is !
=== Loop over all characters in string "for" and "in":
H
e
l
l
o
,

W
o
r
l
d
!
=== ERROR! Strings are immutable in Python! and do not support item
assignment!
```

Προσέξτε πως εάν προσπαθήσουμε να αλλάξουμε έναν χαρακτήρα σε συγκεκριμένη θέση μέσα σε μια μεταβλητή τύπου *string*, δημιουργείται ένα `TypeError`. Αυτό συμβαίνει επειδή, όπως είδαμε και στη Διάλεξη 05, τα *strings* είναι αμετάβλητα αντικείμενα (*immutable objects*) στην Python. Αυτό είναι σε αντίθεση με τις λίστες (*lists*) ή τις πλειάδες (*tuples*), που μπορούν να τροποποιήσουν τα στοιχεία τους. Οι συμβολοσειρές δεν μπορούν να αλλάξουν στη θέση τους - θα χρειαστεί να δημιουργήσετε μια νέα συμβολοσειρά με τις επιθυμητές τροποποιήσεις.

Παράδειγμα 3 Επαναληπτικές διαδικασίες με *strings*:

tutorial4/ex3.py

```
1  #!/usr/bin/python3
2  '''
3  USAGE:
4      chmod +x ex3.py
5      python3 ex3.py
6      script -q ex3.log python3 -i ex3.py
7
8
9  DESCRIPTION:
10 Introduction to strings (revisited) - Looping over all characters in a string
11 '''
12
13 # Define a string to be examined in this example
14 msg = "It's... Monty Python's Flying Circus"
15 index = 0
16
17 print("=== Examine all characters of a string object; \"while\"-loop")
18 while index < len(msg):
19     print("\t", msg[index])
20     index+= 1
21
22
23 print("=== Examine all characters of a string object; \"while\"-loop (reverse)")
24 index = len(msg)-1
25 while index >= 0:
26     print("\t", msg[index])
27     index-= 1
28
29 print("=== Get list of valid attributes (methods, functions, variables) for an
30       object of type int")
31 print(dir("a"))
32
33 print("=== Examine all characters of a string object; \"for\"-loop and \"in\"")
34 for letter in msg:
35     print("\t", letter.capitalize())
36
37 quit()
```

Αποτέλεσμα:

```
=== Examine all characters of a string object; "while"-loop
I
t
'
s
```

```
.  
.   
.   
  
M  
o  
n  
t  
y  
  
P  
y  
t  
h  
o  
n  
,  
s  
  
F  
l  
y  
i  
n  
g  
  
C  
i  
r  
c  
u  
s  
  
=== Examine all characters of a string object; "while"-loop (reverse  
s  
u  
c  
r  
i  
C  
  
g  
n  
i  
y  
l  
F  
  
s  
,  
n  
o
```



```

h
t
y
P

y
t
n
o
M

.
.
.
s
,
t
I
=== Get list of valid attributes (methods, functions, variables) for an
    object of type int
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__
', '__eq__', '__format__', '__ge__', '__getattr__', '__getitem__', '
__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '
__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__',
 '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '
__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '
capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', '
expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha
', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', '
isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', '
ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind',
'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines
', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', '
zfill']
=== Examine all characters of a string object; "for"-loop and "in"
I
T
,
S
.
.
.

M
O
N
T
Y

P
Y

```

```
T  
H  
O  
N  
,  
S  
  
F  
L  
Y  
I  
N  
G  
  
C  
I  
R  
C  
U  
S
```

Όπως είδαμε και στη Διάλεξη 05, η χρήση ενός βρόχου *for* είναι καλύτερη αφού η μεταβλητή επανάληψης περιέχεται μέσα στην δομή του βρόχου μας. Υπενθυμίζεται επίσης πως η συνάρτηση `dir()` στην Python μπορεί να χρησιμοποιείται για τη λήψη λίστας με όλα τα έγκυρα χαρακτηριστικά ενός αντικειμένου (π.χ. μέθοδοι, συναρτήσεις, κλάσεις, μεταβλητές κ.λπ.).

Παράδειγμα 4 Χρήση υποακολουθίας (δηλ. τμήματος) σε συμβολοσειρές με τη τεχνική του τεμαχισμού (slicing):

tutorial4/ex4.py

```
1  #!/usr/bin/python3
2  '''
3  USAGE:
4      chmod +x ex4.py
5      python3 ex4.py
6      script -q ex4.log python3 -i ex4.py
7
8
9  DESCRIPTION:
10 Introduction to strings (revisited) - Looping over all characters in a string
11 '''
12
13 # Define a string to be examined in this example
14 print("=== The original string is:")
15 msg = "It's... Monty Python's Flying Circus"
16 print(msg)
17
18
19 print("\n=== All characters in msg. Thus, msg[:] is equivalent to msg:")
20 print(msg[:])
21
22
23 print("\n=== All characters in msg except last one:")
24 # The slice includes all elements up to, but not including, the element at this
   index.
25 print(msg[0:-1:+1])
26 print(msg[:-1])
27
28
29 print("\n=== All characters in range 0 to 2 (excluding) with step=1:")
30 print(msg[0:2])
31 print(msg[:2])
32
33
34 print("\n=== All characters in range 8 to -1 (excluding) with step=1:")
35 print(msg[8:-1])
36
37
38 print("\n=== All characters in range 8 to -1 (including) with step=1:")
39 print(msg[8:])
40
41
42 print("\n=== All second character starting from character at position 1:")
43 print(msg[1:-1:+2])
44
45
46 print("\n=== All characters after the first two words:")
47 print(msg[ len("It's... ") :-1:+1])
48
```

Παράδειγμα 4 συνεχίζεται...

```
49
50 print("\n=== All characters after the first five word:")
51 print(msg[len("It's... Monty Python's "):-1:+1])
52
53
54 print("\n=== All characters excluding the last character:")
55 print(msg[0:-1:+1])
56
57
58 print("\n=== All characters including the last character:")
59 print(msg[0:len(msg):+1])
60
61
62 print("\n=== All characters in reverse, excluding last (i.e. first) character:"
63       )
64 print(msg[-1:0:-1])
65
66
67 print("\n=== All characters in reverse, including last (i.e. first) character:"
68       )
69 print(msg[len(msg):0:-1])
70
71
72 print("\n=== All characters in reverse, including last (i.e. first) character
73       using string concaten")
74 print(msg[-1:0:-1] + msg[0])
75
76
77 print("\n===", "Trying to slice a string beyond the end of the string, Python
78       will not raise an error; it will slice the string as much as possible and
       stop at the end of the string:")
79
80 print(msg[0:len(msg):+1])
81
82 quit()
```

Αποτέλεσμα:

```
=== The original string is:
It's... Monty Python's Flying Circus

=== All characters in msg. Thus, msg[:] is equivalent to msg:
It's... Monty Python's Flying Circus

=== All characters in msg except last one:
It's... Monty Python's Flying Circu
It's... Monty Python's Flying Circu

=== All characters in range 0 to 2 (excluding) with step=1:
It
```

```
It

=== All characters in range 8 to -1 (excluding) with step=1:
Monty Python's Flying Circu

=== All characters in range 8 to -1 (including) with step=1:
Monty Python's Flying Circus

=== All second character starting from character at position 1:
ts. ot yhnsFyn ic

=== All characters after the first two words:
Monty Python's Flying Circu

=== All characters after the first five word:
Flying Circu

=== All characters excluding the last character:
It's... Monty Python's Flying Circu

=== All characters including the last character:
It's... Monty Python's Flying Circus

=== All characters in reverse, excluding last (i.e. first) character:
sucriC gniylF s'nohtyP ytnoM ...s't

=== All characters in reverse, including last (i.e. first) character:
sucriC gniylF s'nohtyP ytnoM ...s't

=== All characters in reverse, including last (i.e. first) character using
    string concaten
sucriC gniylF s'nohtyP ytnoM ...s'tI

=== Trying to slice a string beyond the end of the string, Python will not
    raise an error; it will slice the string as much as possible and stop at
    the end of the string:
It's... Monty Python's Flying Circus
```

Η βασική σύνταξη για τον τεμαχισμό είναι **ακολουθία[start:stop:step]** όπου:

- **start**: Ο δείκτης από το οποίο ξεκινά ο τεμαχισμός (συμπεριλαμβανομένου).
- **stop**: Ο δείκτης στον οποίο τελειώνει ο τεμαχισμός. Περιλαμβάνει όλα τα στοιχεία μέχρι, αλλά δεν περιλαμβάνει, το τελευταίο στοιχείο.
- **step**: Το βήμα (προαιρετικό). Αυτό είναι το μέγεθος βήματος (ή το διάστημα) μεταξύ των στοιχείων του τεμαχισμού. Καθορίζει πόσα στοιχεία θα παραβλεφθούν. Η προεπιλεγμένη τιμή είναι 1.

Αν παραλείψουμε την τιμή του κάτω (πάνω) ορίου τότε θεωρείται αυτόματα η τιμή της αρχικής (τελικής) θέσης της συμβολοσειράς.

Παράδειγμα 5 Χρήση της χαρακτηριστικής λέξης *in* ως λογικός τελεστής και μετατροπή ενός αντικειμένου *string* σε αντικείμενο τύπου *int*:

tutorial4/ex5.py

```
1  #!/usr/bin/python3
2  '''
3  USAGE:
4      chmod +x ex5.py
5      python3 ex5.py
6      script -q ex5.log python3 -i ex5.py
7
8
9  DESCRIPTION:
10 A look at the keyword "in" as a logical operator. Also casting of strings into
11 other variable types.
12 '''
13 # Define two string variables
14 phrase1 = "It's.... Monty Python's Flying Circus"
15 phrase2 = "And Now for Something Completely Different"
16
17 # More examples on string slicing
18 print("=== ", phrase1, end="\n")
19 print("=== ", phrase1[0:], end="\n")      # same. default last position is
20                                           # position of last character
21 print("=== ", phrase1[0:-1], end="\n")    # without last letter
22 print("=== ", phrase1[:-6], end="\n")     # without last word. default start
23                                           # position is 0
24 print("=== ", phrase1[9:], end="\n")      # without first word
25 print("=== ", phrase1[9:-16], end="\n")   # first two words
26
27
28 print("\n=== Casting string value to float: ")
29 num = "2.14159" # pi - 1
30 try:
31     print("\tPi = ", float(num) + 1.0)
32 except:
33     print("\t", "ERROR! Cannot convert string with decimal into float!")
34
35
36 print("\n=== Casting string value to int: ")
37 try:
38     print("\tPi = ", int(num) + 1.0)
39 except:
40     print("\t", "ERROR! Cannot convert string with decimal into integer!")
41
42
43 print("\n=== Use of 'in' and 'not in' operators on strings:")
44 print("." in phrase2)
45 print("." not in phrase2)
46
47
48 print("\n=== Use of 'in' operator and list of numbers:")
49 print(1 in [0, 2, 4, 6, 8, 10])
```

Παράδειγμα 5 συνεχίζεται...

```
47 print("\n=== Use of 'in' operators in list of strings:")
48 print("a" in ["Albert", "Eistein", "B"])
49 print("B" in ["Albert", "Eistein", "B"])
50 print("B" not in ["Albert", "Eistein", "B"])
51
52 quit()
```

Αποτέλεσμα:

```
=== It's.... Monty Python's Flying Circus
=== It's.... Monty Python's Flying Circus
=== It's.... Monty Python's Flying Circu
=== It's.... Monty Python's Flying
=== Monty Python's Flying Circus
=== Monty Python

=== Casting string value to float:
Pi = 3.14159

=== Casting string value to int:
ERROR! Cannot convert string with decimal into integer!

=== Use of 'in' and 'not in' operators on strings:
False
True

=== Use of 'in' operator and list of numbers:
False

=== Use of 'in' operators in list of strings:
False
True
False
```

Παράδειγμα 6 Παραδείγματα χρήσης της μεθόδου *format* και του τελεστή % για μορφοποίηση:

tutorial4/ex6.py

```
1  #!/usr/bin/python3
2  '''
3  USAGE:
4      chmod +x ex9.py
5      python3 ex9.py
6      script -q ex9.log python3 -i ex9.py
7
8
9  DESCRIPTION:
10 Introduction to string formatting.
11
12
13 NOTES:
14 The type can be used with format codes:
15 'd' for integers
16 'f' for floating-point numbers
17 'b' for binary numbers
18 'o' for octal numbers
19 'x' for octal hexadecimal numbers
20 'h' for hexadecimal numbers
21 's' for string
22 'e' for floating-point in an exponent format
23
24
25 LINKS:
26 https://docs.python.org/3/library/string.html
27 https://www.w3schools.com/python/ref\_string\_format.asp
28 '''
29 import numpy as np
30
31 msg = "1) The value of pi is " + str(np.pi)
32 print("===", msg)
33
34
35 # The empty braces {} interpreted as the next item in the format list
36 msg = "{} The value of {} is {}".format("2)", "pi", np.pi)
37 print("===", msg)
38
39
40 # Print pi with only 3 decimals places
41 msg = "{}) The value of {} is {:.3f}".format("3", "pi", np.pi)
42 print("===", msg)
43
44
45 # Explicit use of the format list item to use by the use of the list index
46 msg = "4) The {0} of {4} is {5:.4f} (4 decimal places)".format("value", "JOHN"
47     , "CLEESE", "also not selected", "pi", np.pi, "blah", "blooh")
48 print("===", msg)
49
```


Παράδειγμα 6 συνεχίζεται...

```
50 msg = "5) The {1} of {2} is {0:.5f}".format(1.12345, "value", "my number", "led", "zeppelin", np.pi)
51 print("===", msg)
52
53
54 a = 4
55 b = 8
56 msg = "6) '{1:d}' times '{0:d}' equals '{2:d}'".format(b, a, a*b)
57 print("===", msg)
58
59
60 msg = "7) '{1:d}' times '{0:d}' equals '{2:d}'".format(a, b, a*b)
61 print("===", msg)
62
63
64 msg = "8) For integer use {0:d}, for string use {1:s}, for binary use {2:b} or {3:b}, for exp use {4:e}".format(8, "three", True, False, 100*100)
65 print("===", msg)
66
67
68 # "{:g}" is a formatting specifier that represents the number in a format that is "general,"
69 # i.e. it chooses a format that is suitable for the number, avoiding unnecessary trailing zeros.
70 msg = "9) The value of pi is {:g}".format(np.pi)
71 print("===", msg)
72
73
74 myList=[str(i) for i in range(11)]
75 msg = "10) The element of the list with index 2 has value {0[2]:s}, and that with index 5 has value {0[5]:s}!".format(myList, myList)
76 print("===", msg)
77
78
79 msg = "11) %s has won the Nobel Prize %d times; in %s and %s.\n\tShe represents %.3f (%.2f %%) of 118 Nobel prizes in Physics awarded between 1901 and 2022." % ("Marie Curie", 2, "Phyics", "Chemistry", float(1/118), float(1/118)*(100.0) )
80 print("===", msg)
81
82 quit()
```

Αποτέλεσμα:

```
=== 1) The value of pi is 3.141592653589793
=== 2) The value of pi is 3.141592653589793
=== 3) The value of pi is 3.142
=== 4) The value of pi is 3.1416 (4 decimal places)
=== 5) The value of my number is 1.12345
=== 6) '4' times '8' equals '32'
```

```
=== 7) '8' times '4' equals '32'
=== 8) For integer use 8, for string use three, for binary use 1 or 0, for
    exp use 1.000000e+04
=== 9) The value of pi is 3.14159
=== 10) The element of the list with index 2 has value 2, and that with
    index 5 has value 5!
=== 11) Marie Curie has won the Nobel Prize 2 times; in Physics and Chemistry
    .
    She represents 0.008 (0.85 %) of 118 Nobel prizes in Physics awarded
    between 1901 and 2022.
```

Η Python μπορεί να αναπαραστήσει αριθμούς ή συμβολοσειρές χρησιμοποιώντας τη μέθοδο *format* και του τελεστή μορφοποίησης % ως ακολούθως:

- d** για ακέραιους αριθμούς
- f** για αριθμούς κινητής υποδιαστολής
- b** για δυαδικούς αριθμούς
- o** για οκταδικούς αριθμούς
- x** για οκταδικούς δεκαεξαδικούς αριθμούς
- h** για δεκαεξαδικούς αριθμούς
- s** για συμβολοσειρά
- e** για κινητή υποδιαστολή σε μορφή εκθέτη

Παράδειγμα 7 Παραδείγματα χρήσης συναρτήσεων string:

tutorial4/ex7.py

```
1  #!/usr/bin/python3
2  '''
3  USAGE:
4      chmod +x ex7.py
5      python3 ex7.py
6      script -q ex7.log python3 -i ex7.py
7
8
9  DESCRIPTION:
10 Methods available for string manipulation (find, replace)
11 '''
12 import numpy as np
13
14 # Define two string variables
15 phrase1 = "It's.... Monty Python's Flying Circus"
16
17
18 print("=== Return the index of FIRST match; start from 0, return -1 if no match
19       found:")
19 print("\t", phrase1.find("."))
20 print("\t", phrase1.find("!"))
21
22
23 print("\n=== Lower/Upper-case letters:")
24 print(phrase1.lower())
25 print(phrase1.upper())
26
27
28 print("\n=== The methods return a COPY of the original string (which remains
29       UNCHANGED):")
29 print(phrase1)
30
31
32 # Case-insensitive searches
33 match = []
34 for l in phrase1:
35     if l.lower() == "y":
36         match.append(l)
37 print("\n=== Found %d matches for letter 'Y' or 'y' in input '%s'" % (len(
38     match), phrase1) )
39
40
41 print("\n=== Usage example of \"replace\" function ('/' with '-')")
42 dateOld = "04/10/2022"
43 dateNew = dateOld.replace("/", "-")
44 print("\tThe old date is {0:s} and the new date is {1:s}".format(dateOld,
45     dateNew))
46 print("\tThe old date is %s and the new date is %s".format(dateOld, dateNew))
47
48
```

Παράδειγμα 7 συνεχίζεται...

```
47 print("\n=== Showing the usage of methods lstrip(), rstrip(), strip(), and
    startswith()")
48 phrase2 = "    --> And Now for Something Completely Different<--    "
49 print("\tphrase2 is \"%s\" % (phrase2) )
50 print("\tphrase2.lstrip() is \"%s\" % (phrase2.lstrip()) )
51 print("\tphrase2.rstrip() is \"%s\" % (phrase2.rstrip()) )
52 print("\tphrase2.strip() is \"%s\" % (phrase2.strip()) )
53
54
55 name = "Albert Einstein"
56 print("\n=== Demonstration of the functions startswith() and endswith():")
57 print(name.startswith("A"))
58 print(name.startswith("a"))
59 print(name.endswith("ein"))
60
61 quit()
```

Αποτέλεσμα:

```
=== Return the index of FIRST match; start from 0, return -1 if no match
found:
4
-1

=== Lower/Upper-case letters:
it's.... monty python's flying circus
IT'S.... MONTY PYTHON'S FLYING CIRCUS

=== The methods return a COPY of the original string (which remains
UNCHANGED):
It's.... Monty Python's Flying Circus

=== Found 3 matches for letter 'Y' or 'y' in input 'It's.... Monty Python's
Flying Circus'

=== Usage example of "replace" function ('/' with '-')
The old date is 04/10/2022 and the new date is 04-10-2022
The old date is %s and the new date is %s

=== Showing the usage of methods lstrip(), rstrip(), strip(), and startswith
()
phrase2 is "    --> And Now for Something Completely Different<--    "
phrase2.lstrip() is "--> And Now for Something Completely Different<--
"
phrase2.rstrip() is "    --> And Now for Something Completely Different
<--"
phrase2.strip() is "--> And Now for Something Completely Different<--"

=== Demonstration of the functions startswith() and endswith():
True
```

Παράδειγμα 7 συνεχίζεται . .

```
False  
True
```