



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Μάθημα: ECE_Y106 Εισαγωγή στους Υπολογιστές

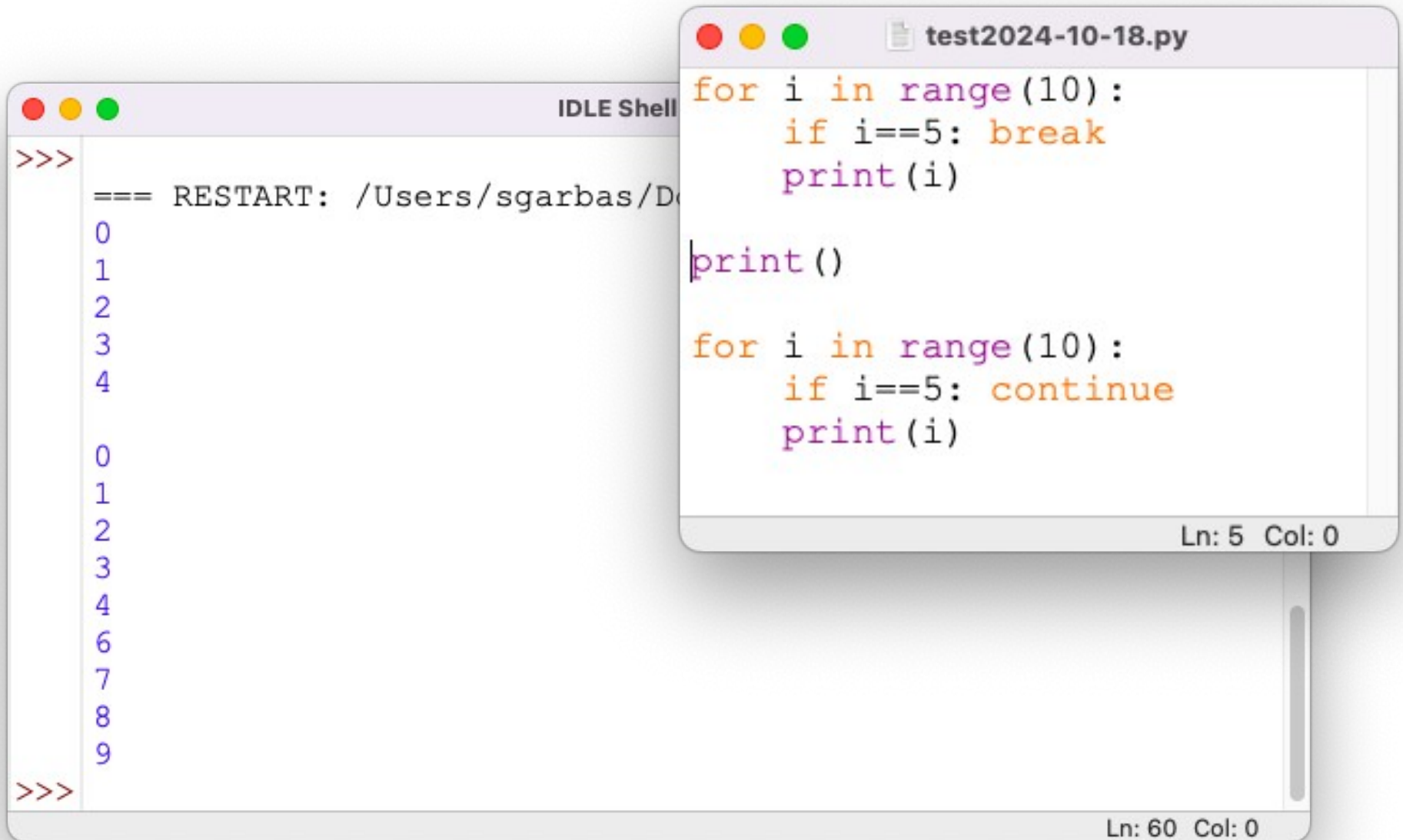
ECE_Y106: Εισαγωγή στους Υπολογιστές

ΔΙΑΛΕΞΗ #3

Ακολουθίες και Συναρτήσεις

Αν. Καθ. Κυριάκος Σγάρμπας

break / continue



The image shows two overlapping windows from the Python IDLE environment. The background window is the 'IDLE Shell' with a 'RESTART' message and a list of numbers from 0 to 9. The foreground window is a code editor for 'test2024-10-18.py' containing two Python code snippets. The first snippet uses a 'break' statement to stop a loop at i=5. The second snippet uses a 'continue' statement to skip i=5 in a loop.

```
test2024-10-18.py
for i in range(10):
    if i==5: break
    print(i)

print()

for i in range(10):
    if i==5: continue
    print(i)
```

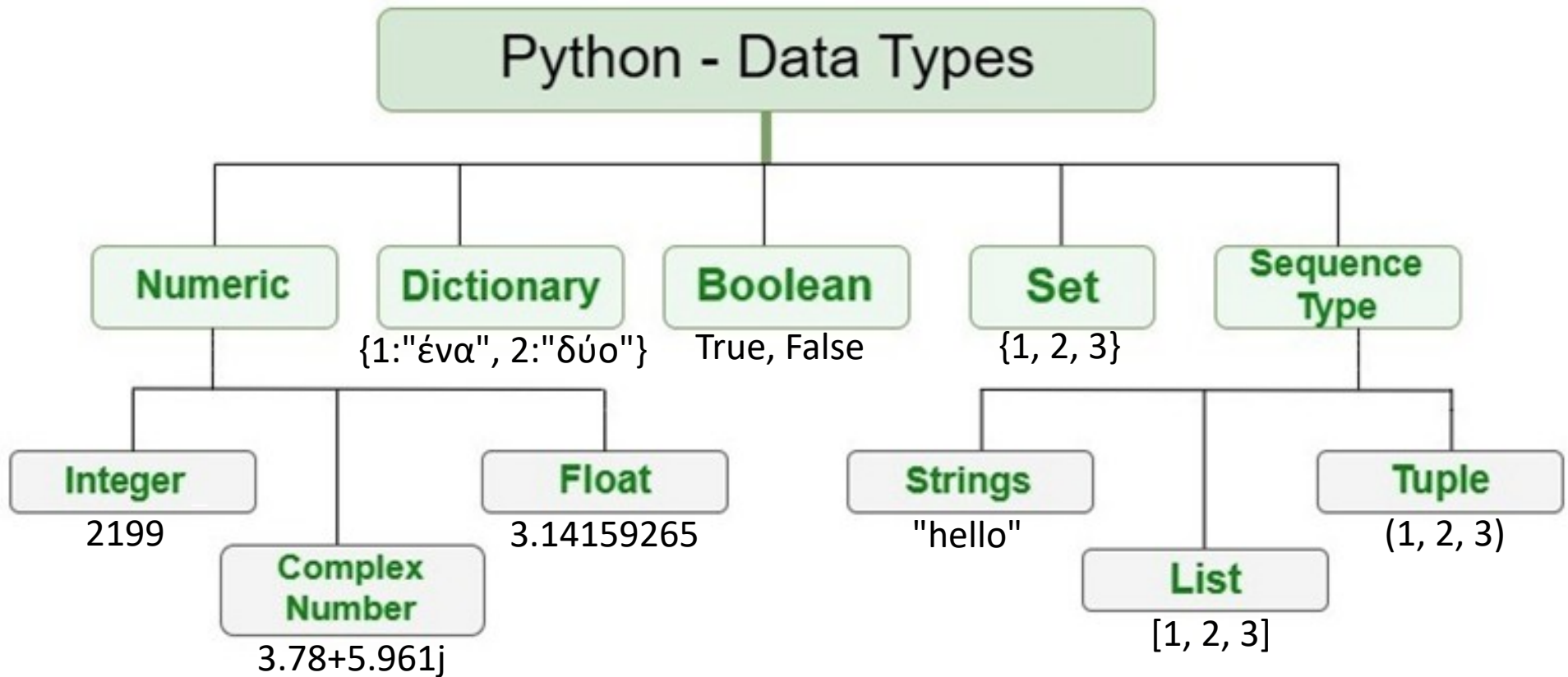
Ln: 5 Col: 0

IDLE Shell

```
>>>
=== RESTART: /Users/sgarbas/D
0
1
2
3
4
0
1
2
3
4
6
7
8
9
>>>
```

Ln: 60 Col: 0

Τύποι Δεδομένων



sequences (ακολουθίες)

```
IDLE Shell 3.12.6
>>> s = "abcdefgh"
>>> L = ["a","b","c","d","e","f","g","h"]
>>> t = ("a","b","c","d","e","f","g","h")
>>>
>>> len(s)
8
>>> s[3]
'd'
>>> L[0]
'a'
>>> t[-1]
'h'
>>> s[2:6]
'cdef'
>>> L[:5]
['a', 'b', 'c', 'd', 'e']
>>> t[4:]
('e', 'f', 'g', 'h')
>>>
>>> s[0:7:2]
'aceg'
>>> list(s[0:7:2])
['a', 'c', 'e', 'g']
>>> tuple(s[0:7:2])
('a', 'c', 'e', 'g')
>>> L[:7:2]
['a', 'c', 'e', 'g']
>>> tuple(L[:7:2])
('a', 'c', 'e', 'g')
>>> str(L[:7:2])
"['a', 'c', 'e', 'g']"
>>> |
```

Ln: 91 Col: 0

```
IDLE Shell 3.12.6
>>> s
'abcdefgh'
>>> s[0]
'a'
>>> s[0] = "k"
Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    s[0] = "k"
TypeError: 'str' object does not support item assignment
>>> L[0]
'a'
>>> L[0] = "k"
>>> L
['k', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
>>> t[0] = "k"
Traceback (most recent call last):
  File "<pyshell#32>", line 1, in <module>
    t[0] = "k"
TypeError: 'tuple' object does not support item assignment
>>>
>>> L = [1, 2.11, "e", "sdfg", (1,2), [3,4,5]]
>>> t = tuple(L)
>>> t
(1, 2.11, 'e', 'sdfg', (1, 2), [3, 4, 5])
>>> t[-1]
[3, 4, 5]
>>> t[-1][0]
3
>>>
>>> L = [1,2,3]
>>> M = L
>>> M[0] = 100
>>> print(M,L)
[100, 2, 3] [100, 2, 3]
>>> id(M)
4536559360
>>> id(L)
4536559360
```

Ln: 143 Col: 0

strings, lists, tuples

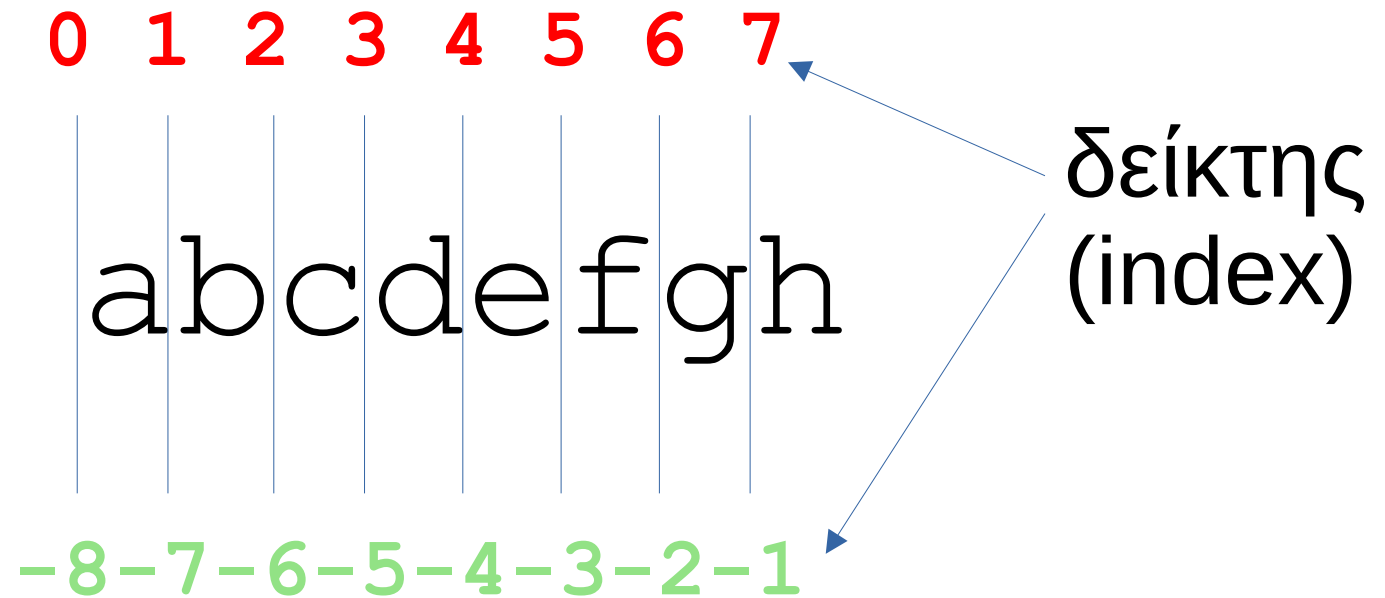
- ομοιότητες:

- len()
- indexing
- slicing
- +, *

- διαφορές:

- τύποι περιεχομένων
- mutable/
immutable
- μέθοδοι

slicing (τεμαχισμός)



Μέθοδοι

```

IDLE Shell 3.12.6
>>> s = "καλημέρα"
>>> s.upper()
'ΚΑΛΗΜΕΡΑ'
>>>
>>> s = "καλημέρα"
>>> a = s.upper()
>>> a
'ΚΑΛΗΜΕΡΑ'
>>> b = a.lower()
>>> b
'καλημέρα'
>>>
>>> L = [3,1,6,5,4,8,1,3]
>>> L.sort()
>>> L
[1, 1, 3, 3, 4, 5, 6, 8]
>>> L.append(2)
>>> L
[1, 1, 3, 3, 4, 5, 6, 8, 2]
>>> L.reverse()
>>> L
[2, 8, 6, 5, 4, 3, 3, 1, 1]
>>> sorted(L)
[1, 1, 2, 3, 3, 4, 5, 6, 8]
>>> L
[2, 8, 6, 5, 4, 3, 3, 1, 1]
>>>
Ln: 177 Col: 0

```

- String.upper()
- String.lower()
- List.append()
- List.pop()
- List.remove()
- List.sort()
- List.reverse()

List Comprehensions

```

IDLE Shell 3.12.6
>>>
>>> L = [1,2,3,4]
>>> M = [2*x for x in L]
>>> M
[2, 4, 6, 8]
>>> N = [3*x for x in L if x%2==1]
>>> N
[3, 9]
>>> P = L+M+N
>>> P
[1, 2, 3, 4, 2, 4, 6, 8, 3, 9]
>>> from math import log
>>> Q = [log(x/2) for x in P]
>>> Q
[-0.6931471805599453, 0.0, 0.4054651081081644, 0.69
31471805599453, 0.0, 0.6931471805599453, 1.09861228
86681098, 1.3862943611198906, 0.4054651081081644, 1
.5040773967762742]
>>> |
Ln: 205 Col: 0

```

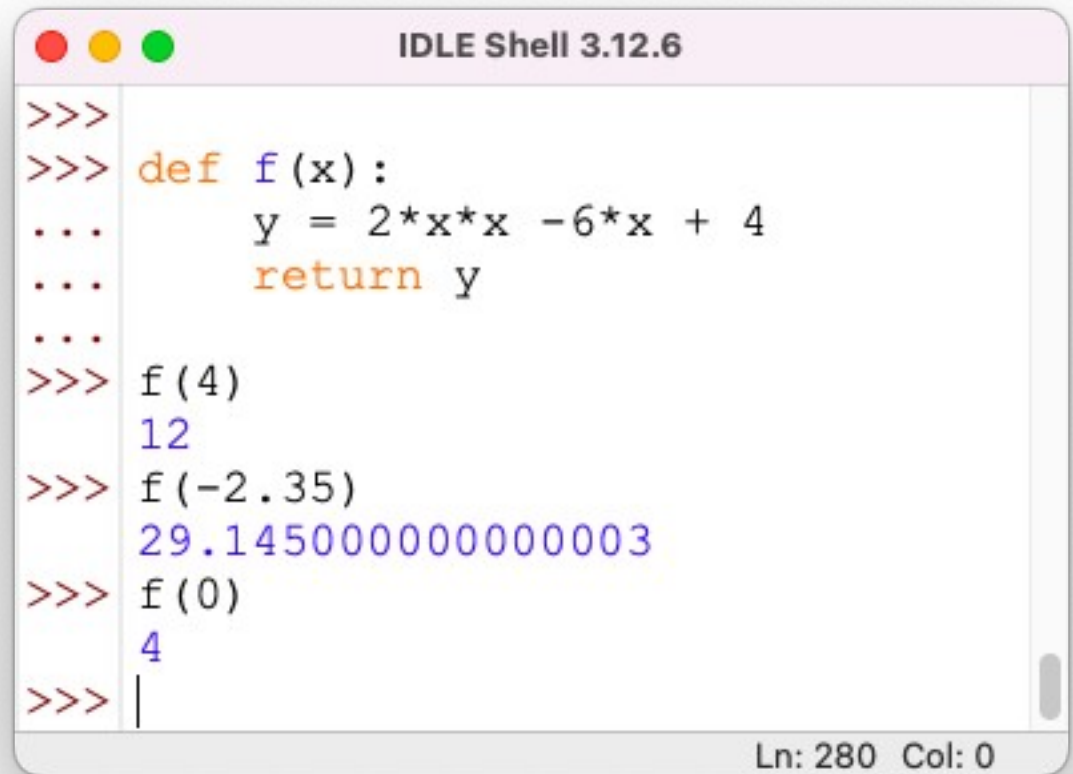

Αφανείς Πλειάδες

```

IDLE Shell 3.12.6
>>> a,b,c = 1,2,3
>>> print(a,b,c)
1 2 3
>>>
>>> a,b,c = 1,2,3
>>> a
1
>>> b
2
>>> c
3
>>> (a,b,c) = (1,2,3)
>>> a
1
>>> b
2
>>> c
3
>>>
>>> a,b = b,a
>>> a
2
>>> b
1
>>>
>>> (a,b) = (b,a)
>>>
>>> a
1
>>> b
2
>>>
Ln: 234 Col: 0
```

Ορισμός Συναρτήσεων

```
def function_name(arg1, arg2, ...):  
    εντολή  
    εντολή  
    ...  
    εντολή  
  
    return val1, val2, ...
```



```
IDLE Shell 3.12.6  
>>>  
>>> def f(x):  
...     y = 2*x*x - 6*x + 4  
...     return y  
...  
>>> f(4)  
12  
>>> f(-2.35)  
29.145000000000003  
>>> f(0)  
4  
>>> |  
Ln: 280 Col: 0
```

Εύρεση Πρώτων Αριθμών

```
test2024-10-18.py
def is_prime(x):
    if x<2: return False
    if x==2: return True
    if x%2==0: return False
    for i in range(3, int(x**0.5)+1, 2):
        if x%i==0: return False
    return True

for x in range(1000):
    if is_prime(x): print(x,end=" ")

Ln: 11 Col: 0

IDLE
>>>
= RESTART: /Users/sgarb
2 3 5 7 11 13 17 19 23
1 73 79 83 89 97 101 10
49 151 157 163 167 173
227 229 233 239 241 251
3 307 311 313 317 331 3
383 389 397 401 409 419
1 463 467 479 487 491 499 503 509 521 523 541 547 557
563 569 571 577 587 593 599 601 607 613 617 619 631 64
1 643 647 653 659 661 673 677 683 691 701 709 719 727
733 739 743 751 757 761 769 773 787 797 809 811 821 82
3 827 829 839 853 857 859 863 877 881 883 887 907 911
919 929 937 941 947 953 967 971 977 983 991 997

>>>

Ln: 269 Col: 0
```

Από που διαβάζουμε

- Αγγελιδάκης, έως σελ. 103
- Μανής, σελ. 61-69, 104-139
- Swaroop, σελ. 37-47, 54-58
- Περάκης, σελ. 45-59, 63-66
- Python Tutorial, έως 5.3

Φροντιστηριακή Άσκηση #3