

Εισαγωγή στην Γλώσσα Προγραμματισμού Python

Βασισμένο στην παρουσίαση της Python του Michael DiRamio - 2005

Διάφορες Γλώσσες Προγραμματισμού

- C or C++
- Java
- Perl
- Scheme
- Fortran
- Python
- Matlab

και πολλές άλλες...

Περίληψη Παρουσίασης

- Εκτέλεση προγραμμάτων Python και Output
- Τύποι Δεδομένων
- Input και I/O Αρχείων
- Έλεγχος Ροής
- Συναρτήσεις
- Python και Επιστημονικές Εφαρμογές
- Binary Διανομές, Scientific Python

Hello World

- Ανοίγουμε ένα τερματικό και πληκτρολογούμε “python”
- Σε γραφικό περιβάλλον χρησιμοποιούμε το IDLE
- Στο prompt πληκτρολογούμε ‘hello world!’

```
>>> 'hello world!'
```

```
'hello world!'
```

Python Overview

Από το βιβλίο “*Learning Python*”, 4th Edition
Του Mark Lutz, Εκδ. Οίκος O'Reilly:

- Τα προγράμματα είναι συνθέσεις από modules
- Τα Modules περιέχουν εντολές
- Οι εντολές περιέχουν εκφράσεις
- Οι εκφράσεις δημιουργούν και επεξεργάζονται αντικείμενα (objects)

Ο Interpreter της Python

- Η Python είναι γλώσσα interpreted
- Ο interpreter παρέχει ένα διαδραστικό περιβάλλον για την διεπαφή με την γλώσσα
- Τα αποτελέσματα των εκφράσεων εμφανίζονται στην οθόνη

```
>>> 3 + 7
10
>>> 3 < 15
True
>>> 'print me'
'print me'
>>> print('print me')
print me
>>>
```

Η εντολή print

- Στοιχεία χωρισμένα από κόμμα εμφανίζονται με ένα κενό μεταξύ τους
- Το named argument 'end' στο τέλος εντολής `print('hello',end='')` επιτρέπει να αποφύγουμε τον χαρακτήρα `newline`

```
>>> print('hello')
hello
>>> print('hello', 'there')
hello there
```

```
print('hello', end='')
print('there')
```

Τεκμηρίωση

Ο χαρακτήρας '#' ξεκινάει ένα σχόλιο

```
>>> 'this will print'
```

```
'this will print'
```

```
>>> #'this will not'
```

```
>>>
```

Μεταβλητές

- Δεν δηλώνονται, απλά εκχωρούνται
- Η μεταβλητή δημιουργείται την πρώτη φορά που της εκχωρείται μία τιμή
- Αποτελούν αναφορές σε αντικείμενα
- Η πληροφορία τύπου είναι στο αντικείμενο, όχι στην αναφορά
- Τα πάντα στην Python είναι αντικείμενα

Τα πάντα είναι αντικείμενα

- Όταν λέμε τα πάντα, εννοούμε τα πάντα, στα οποία συμπεριλαμβάνονται συναρτήσεις και κλάσεις (περισσότερα σχετικά αργότερα!)
- Ο τύπος δεδομένων αποτελεί **ιδιότητα του αντικειμένου** και όχι της μεταβλητής

```
>>> x = 7
>>> x
7
>>> x = 'hello'
>>> x
'hello'
>>>
```

Τύποι Δεδομένων

Τύπος Αντικειμένου	Παράδειγμα
Numbers	1234, 3.1415, 3+3j
Strings	'test', "John's"
Lists	[1, [2, 'three'], 4]
Dictionaries	{'food': 'spam', 'taste': 'yum'}
Tuples	(1, 'spam', 4, 'U')
Files	myfile=open('eggs', 'r')
Sets	set('abc'), {'a', 'b', 'c'}
Άλλοι βασικοί τύποι	Boolean

Τύποι Αριθμών

- **Integers** (αντίστοιχος τύπος με τον τύπο `long` της C, αλλά χωρίς περιορισμούς)
- **Floating point numbers** και
- **Complex numbers**

```
>>> 132224
132224
>>> 132323 ** 2
17509376329
>>> 13/6
2.1666666666666665
>>> 3+4j + 2 + 1j
(5+5j)
>>>
```

Αριθμοί: Floating Point

- `int(x)` μετατρέπει το `x` σε `integer`
- `float(x)` μετατρέπει το `x` σε `floating point`
- Ο interpreter εμφανίζει πολλά ψηφία

```
>>> 1.23232
•1.23232
•>>> print(1.23232)
•1.23232
•>>> 1.3E7
•13000000.0
•>>> int(2.0)
•2
•>>> float(2)
•2.0
```

Μιγαδικοί Αριθμοί

- Ενσωματωμένοι στην Python
- Υποστηρίζουν τις ίδιες πράξεις όπως οι integer και float

```
>>> x = 3 + 2j
>>> y = -1j
>>> x + y
(3+1j)
>>> x * y
(2-3j)
```

Οι αριθμοί είναι immutable

```
>>> x = 4.5
```

```
>>> y = x
```

```
>>> y += 3
```

```
>>> x
```

```
4.5
```

```
>>> y
```

```
7.5
```

x → 4.5

y ↗

x → 4.5

y → 7.5

String Literals

- Τα Strings είναι *immutable*
- Δεν υφίσταται char type όπως στη C, C++ ή Java
- Ο τελεστής + είναι overloaded για να συνδέει

```
>>> x = 'hello'  
>>> x = x + ' there'  
>>> x  
'hello there'
```

String Literals: Πολλά Είδη

- Μονά ή διπλά εισαγωγικά, και τρία διπλά εισαγωγικά για strings με πολλαπλές γραμμές

```
>>> 'I am a string'
'I am a string'
>>> "So am I!"
'So am I!'
>>> s = """And me too!
though I am much longer
than the others :)"""
'And me too!\nthough I am much longer\nthan the others :)'
>>> print(s)
And me too!
though I am much longer
than the others :)
```

Substrings and Methods

```
>>> s = '012345'  
>>> s[3]  
'3'  
>>> s[1:4]  
'123'  
>>> s[2:]  
'2345'  
>>> s[:4]  
'0123'  
>>> s[-2]  
'4'
```

- **len**(String) – επιστρέφει τον αριθμό των χαρακτήρων του String
- **str**(Object) – επιστρέφει ένα String για το Object

```
>>> len(x)  
6  
>>> str(10.3)  
'10.3'
```

String Formatting

- Μοιάζει με την printf της C
- <formatted string> % <elements to insert>
- Συχνά χρησιμοποιούμε απλά %s για τα πάντα, επειδή μετατρέπει το αντικείμενο στην String μορφή του.

```
>>> "One, %d, three" % 2
'One, 2, three'
>>> "%d, two, %s" % (1,3)
'1, two, 3'
>>> "%s two %s" % (1, 'three')
'1 two three'
>>>
```

Lists

- Ταξινομημένες συλλογές από δεδομένα
- Τα Data μπορούν να είναι διαφόρων τύπων
- Είναι *mutable*
- Ζητήματα με πολλαπλές αναφορές και mutability
- Same subset operations as Strings

```
>>> x = [1,'hello', (3 + 2j)]  
>>> x  
[1, 'hello', (3+2j)]  
>>> x[2]  
(3+2j)  
>>> x[0:2]  
[1, 'hello']
```

Lists: Τροποποιώντας το Περιεχόμενο

- $x[i] = a$ επανακαταχωρεί στο στοιχείο i τιμή a
- Επειδή το x και το y αναφέρονται στο ίδιο στοιχείο της λίστας, και τα δύο αλλάζουν
- Η μέθοδος **append** επίσης τροποποιεί την λίστα

```
>>> x = [1,2,3]
>>> y = x
>>> x[1] = 15
>>> x
[1, 15, 3]
>>> y
[1, 15, 3]
>>> x.append(12)
>>> y
[1, 15, 3, 12]
```

Lists: Τροποιώνοντας το Περιεχόμενο

- Η μέθοδος **append** τροποποιεί την λίστα και επιστρέφει **None**
- Το άθροισμα από lists (+) επιστρέφει νέο list

```
>>> x = [1,2,3]
>>> y = x
>>> z = x.append(12)
>>> z == None
True
>>> y
[1, 2, 3, 12]
>>> x = x + [9,10]
>>> x
[1, 2, 3, 12, 9, 10]
>>> y
[1, 2, 3, 12]
>>>
```

Tuples (πλειάδες)

- Τα tuples είναι *immutable* είδη list
- Προσοχή όταν έχουμε tuple με ένα στοιχείο:
‘,’ είναι απαραίτητο για να διαφοροποιείται από την αριθμητική έκφραση (344)

```
>>> x = (1,2,3)
>>> x[1:]
(2, 3)
>>> z=(344)
•>>> z
•344
•>>> z=(344,)
•>>> z
•(344,)
```

