

Σημειώσεις

Προηγούμενο

Επόμενο

Γλώσσα προγραμματισμού python

Η γλώσσα αυτή γράφτηκε από τον Ολλανδό προγραμματιστή **Guido van Rossum** στα τέλη της δεκαετίας 1980-90. Η έκδοση 2.0 δημοσιεύτηκε στις 16 Οκτωβρίου 2000 και η έκδοση 3.0, η οποία δεν είναι, εν γένει, σύμφωνη (compatible) με τις προηγούμενες εκδόσεις, στις 3 Δεκεμβρίου 2008. Για τις ανάγκες του μαθήματος θα χρησιμοποιήσουμε την έκδοση 3.5.

Η python είναι μία γλώσσα <<υψηλού επιπέδου>> (άλλα παραδείγματα τέτοιων γλωσσών είναι η FORTRAN, C, Java κλπ). Ο κώδικας μία τέτοιας γλώσσας πρέπει να μετατραπεί σε <<γλώσσα μηχανής>> ώστε να εκτελεστεί από τον Η/Υ. Η επεξεργασία αυτή γίνεται από διερμηνευτές (interpreters) και μεταγλωττιστές (compilers). Στην περίπτωση της python η επεξεργασία γίνεται από διερμηνευτή.

Η Python είναι μια γλώσσα προγραμματισμού με απλό συντακτικό, εξαιρετική αναγνωσιμότητα, φορητότητα (portability) και μοντέρνα χαρακτηριστικά που την κάνουν κατάλληλη ως πρώτη γλώσσα προγραμματισμού. Η επιτυχία της οφείλεται σε μεγάλο βαθμό στο γεγονός ότι είναι ερμηνευόμενη γλώσσα (interpreted language). Αυτό σημαίνει ότι ο μεταγλωττιστής της Python παράγει ενδιάμεσο κώδικα (bytecode) ο οποίος μπορεί να εκτελεστεί από τον διερμηνέα (interpreter) σε πολλά διαφορετικά υπολογιστικά περιβάλλοντα. Διερμηνείς για τη γλώσσα Python υπάρχουν για όλα τα δημοφιλή λειτουργικά συστήματα (Windows, Linux, MacOS). Αν ο προσωπικός σας υπολογιστής χρησιμοποιεί το λειτουργικό σύστημα Linux ή το MacOS τότε η Python είναι ήδη εγκατεστημένη (βεβαιωθείτε όμως ότι έχετε τον διερμηνέα για την Python 3 γράφοντας python3 στη γραμμή εντολών). Διαφορετικά, μπορείτε να εγκαταστήσετε το περιβάλλον εργασίας Python με το όνομα Canopy από την ιστοσελίδα της **Enthought**.

Οι ελεύθερα προσβάσιμοι υπολογιστές του Τμήματος χρησιμοποιούν το λειτουργικό σύστημα Linux και έχουν, φυσικά, εγκατεστημένο το διερμηνέα για την Python 3. Περιέχουν επίσης, το κέλυφος **IDLE**, δηλαδή έναν διαδραστικό διερμηνέα της Python με ενσωματωμένο κειμενογράφο. Τα εργαστήρια του μαθήματος θα γίνονται σε αυτά τα υπολογιστικά συστήματα, αλλά μπορείτε να χρησιμοποιείτε τον δικό σας φορητό υπολογιστή.

Έχουμε επιλέξει την python για να μάθουμε προγραμματισμό διότι είναι:

- Γλώσσα ανοικτού κώδικα (διαθέσιμη στη διεύθυνση <http://python.org>).
- Γενικής χρήσης, υψηλού επιπέδου γλώσσα προγραμματισμού.
- Εύκολη στην εκμάθηση (απλό συντακτικό).
- Με δυνατότητες σε διάφορες κατευθύνσεις (ισχυρή γλώσσα προγραμματισμού).
- Κατάλληλη για αρχάριους και για έμπειρους προγραμματιστές.
- Αντικειμενοστραφής.
- Υπάρχουν αρκετά πακέτα υποστήριξης (βιβλιοθήκες).

Μεταβλητές

Πράξεις με αριθμούς

Θα δώσουμε πολλαπλασιασμό και διαίρεση αριθμών και θα πάρουμε το αποτέλεσμα:

```
>>> 3*4
12
>>> 3.0*4.0
12.0
>>> 3.0*4
12.0
>>> 1/2
0.5
>>> 4/2
2.0
>>> 2/3
0.6666666666666666
```

Τύποι μεταβλητών

1. `int` (ακέραιος)
2. `float` (κινητής υποδιαστολής)
3. `bool` (λογική μεταβλητή: παίρνει τις τιμές `True` ή `False`)
4. `NoneType` (παίρνει την τιμή `None`, δηλώνει απουσία τιμής)

Παράδειγμα. Θα δώσουμε τιμές σε μεταβλητές (κινητής υποδιαστολής) με ονόματα: `pi`, `a`.

```
>>> pi = 3.14
```

```
>>> a = 10.0
```

Κάνουμε πράξεις με μεταβλητές και παίρνουμε το αποτέλεσμα σε νέα μεταβλητή.

```
>>> area = pi*a**2
```

Βρήκαμε την επιφάνεια δίσκου με ακτίνα ίση με 10. Χρειάστηκε να υψώσουμε στο τετράγωνο ($a**2$) και να κάνουμε πολλαπλασιασμό (με το σύμβολο $*$). Τελικά εκχωρήσαμε το αποτέλεσμα του πa^2 στη νέα μεταβλητή `area`.

Τελεστές.

- $10+4$ --> 14 (Πρόσθεση)
- $10-4$ --> 6 (Αφαίρεση)
- $10*4$ --> 40 (Πολλαπλασιασμός)
- $10**4$ --> 10000 (Ύψωση σε δύναμη)
- $10/4$ --> 2.5 (Διαίρεση)
- $10//4$ --> 2 (Ακέραια διαίρεση)
- $10\%4$ --> 2 (Υπόλοιπο ακέραιας διαίρεσης)

Παράδειγμα. Δίνουμε τιμές σε μεταβλητές:

```
>>> a = 3
>>> b = 3.14
>>> d = True
>>> n = None
```

Με την εντολή `type` μπορούμε να ελέγξουμε τον τύπο των μεταβλητών οι οποίες ήδη έχουν ορισθεί (έχουν τιμές).

```
>>> type(a)
<class 'int'>
>>> type(b)
<class 'float'>
>>> type(d)
<class 'bool'>
>>> type(n)
```

```
<class 'NoneType'>
```

Άσκηση. Έχουμε 800 εγγεγραμμένους φοιτητές στο μάθημα του προγραμματισμού και έρχονται στις εξετάσεις 300 από αυτούς. Αν οι βαθμολογίες καταχωρηθούν με την βοήθεια της `python`, τι τύπων μεταβλητές χρειαζόμαστε για να καλύψουμε την βαθμολογία και των 800 φοιτητών;

Άσκηση. Αναφέρετε περιπτώσεις όπου ο τύπος `NoneType` θα ήταν χρήσιμος.

Ονόματα μεταβλητών

- Μπορούν να περιέχουν γράμματα και αριθμούς (αλφαριθμητικούς χαρακτήρες),
- αλλά πρέπει να ξεκινούν με ένα γράμμα.
- Τα πεζά διακρίνονται από τα κεφαλαία.
- Μπορεί να περιέχεται το underscore (`_`).

Παραδείγματα

```
>>> natural_number = 3
>>> to_pi = 3.14
>>> greeting = 'hallo there!'
>>> protasi = True
>>> bathmos = None
>>> Bathmos = 10
```

Μη αποδεκτά ονόματα μεταβλητών:

- `1number` - αρχίζει με αριθμό.
- `number!` - περιέχει το `!`
- `class` - είναι λέξη-κλειδί.

Λέξεις κλειδιά

Είναι δεσμευμένες λέξεις με ειδική σημασία. Για να τις δούμε όλες:

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class',
 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for',
```

```
'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal',  
'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with',  
'yield']
```

Εντολές

Έχουμε ήδη δει μερικές εντολές της python: `print`, `type`, `int`, `float`, `str`.

Παραδείγματα:

```
>>> print(a)  
3  
>>> print(d)  
True  
>>> print(a,b)  
3 3.14  
>>> int(b)  
3  
>>> float(a)  
3.0  
>>> str(3.14159)  
'3.14159'  
>>> print(greeting)  
hallo there!
```

Άλλα παραδείγματα:

```
>>> print(a,b,greeting)  
3 3.14 hallo there!
```

Άσκηση Γράψτε εντολές με τις οποίες θα δίνονται τιμές σε δύο μεταβλητές (π.χ. 3 και 3.14 αντίστοιχα) και μετά θα τυπώνεται το εξής

```
The value is: 3
```

```
The value is: 3.14
```

Πακέτα

Έχουμε δει επίσης την εντολή `import` η οποία μας επιτρέπει πρόσβαση σε προκαθορισμένες μεταβλητές, συναρτήσεις κλπ. Το πακέτο `math` περιέχει χρήσιμες μαθηματικές σταθερές και μαθηματικές συναρτήσεις.

```
>>> import math
>>> print(math.pi)
3.141592653589793
>>> math.sqrt(2.0)
1.4142135623730951
>>> math.sqrt(2)
1.4142135623730951
>>> math.exp(1)
2.718281828459045
```

Σφάλματα

Συντακτικό σφάλμα. Ας δούμε τα εξής

```
>>> print c
SyntaxError: Missing parentheses in call to 'print'
>>> print(c,Nick)
Traceback (most recent call last):
  File "", line 1, in
    print(c,Nick)
NameError: name 'Nick' is not defined
```

Η πρώτη εντολή έχει **συντακτικό** λάθος το οποίο η Python ονομάζει `SyntaxError`. Η δεύτερη εντολή έχει ένα λάθος το οποίο η Python ονομάζει `NameError`. Ο διερνηνευτής (interpreter) μας δίνει πληροφορία σχετικά με το κάθε λάθος.

Σφάλμα σημαντικής. Ας δούμε επίσης

```
>>> print(3,2)
3 2
```

Τυπώσαμε δύο αριθμούς. Δεν φαίνεται να υπάρχει κανένα λάθος στο παραπάνω. Εάν όμως ο σκοπός μας ήταν να τυπώσουμε τον αριθμό 3.2 τότε έχουμε κάνει ένα **λάθος σημαντικής** (semantic error). Δηλαδή, η εντολή δίνει μεν ένα αποτέλεσμα, αλλά δεν είναι αυτό που θέλαμε.

Μελέτη

Τοπικές ιστοσελίδες

1. [Ιστοσελίδα μαθήματος "Γλώσσα Προγραμματισμού I", 2016 \(Μιχ. Πλεξουσάκης\)](#)
2. [Ιστοσελίδα μαθήματος "Γλώσσα Προγραμματισμού I", 2015](#)

Βιβλιογραφία

1. [An Informal Introduction to Python: variables, strings](#)
2. Δ. Καρολίδης, *Μαθαίνετε εύκολα python*.
3. Κ. Μαγκούτης, Χ. Νικολάου, [Εισαγωγή στον αντικειμενοστραφή προγραμματισμό με Python](#), (Αποθετήριο "Κάλλιπος", 2016) - Δείτε τα κεφάλαια [1. Εισαγωγή στην Python](#), [2. Τιμές, τύποι, μεταβλητές, λέξεις κλειδιά, εντολές](#), [3. Εκφράσεις, τελεστές, σχόλια](#).

Εγκαταστήστε την python στον Η/Υ σας

- python.org
- enthought.com

Εργαλεία

- [Cloud coding environment](#).
- Μπορείτε να τρέξετε python στον φυλλομετρητή: [Codeskulptor](#).
- Μπορείτε να τρέξετε python στον φυλλομετρητή: [CodeAcademy](#).
- idle3: Είναι ένα ολοκληρωμένο περιβάλλον για την ανάπτυξη και εκτέλεση προγραμμάτων python.
- Μπορείτε να τρέξετε python στο κινητό σας τηλέφωνο (εφαρμογή QPython3).

**Γράψτε τι άλλο θέλετε
από αυτές τις σημειώσεις**

[Προηγούμενο](#)

[Επόμενο](#)