

Ερωτηματολόγιο προγραμματισμού Python

Απαντήσεις: Μάθημα 0 - Εισαγωγή, Μάθημα 1 - Τα βασικά

Revision: 0.2

Date: 2009-07-12

Author: Savvas Radevic <vicedar@gmail.com>

Copyright: © 2009 Savvas Radevic <vicedar@gmail.com>, © 2009 python-gr group <python-gr@googlegroups.com>

Credits: CREDITS.html

License: GNU FDL 1.3 <<http://www.gnu.org/licenses/fdl-1.3.txt>>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts. A copy of the license is included in the file entitled "COPYING-DOCS.txt".

Source: 01-basics-answers.tar.gz

PDF: 01-basics-answers.pdf

Περιεχόμενα

Γλωσσάρι λέξεων	1
Εισαγωγή	1
Απαντήσεις	1
Μάθημα 0	1
Μάθημα 1	3

Γλωσσάρι λέξεων

Ελληνικά	English
Συμβολοσειρά	String/Text string
Μεταβλητές	Variables
Τύποι δεδομένων	Data types
Αντικείμενα	Objects
Συνένωση	Concatenation
Εισαγωγικά	Quotes (π.χ. ",")

Εισαγωγή

Οποιοσδήποτε απορίες σχετικά με τις ασκήσεις / απαντήσεις, μπορείτε να τις εναποθέσετε στο θέμα συζήτησης: <http://forum.ubuntu-gr.org/viewtopic.php?f=55&t=5574>

Απαντήσεις

Μάθημα 0

1. Ποιος είναι ο δημιουργός της python;

Δημιουργήθηκε από τον Ολλανδό Guido van Rossum. Συνεχίζει να προσφέρει στο θαυμάσιο έργο του μαζί με άλλα μέλη της κοινότητας (κεντρική ομάδα ανάπτυξης / "core development team"). Η ανάπτυξη διευθύνεται από τον οργανισμό μη-κερδοσκοπικού χαρακτήρα Python Software Foundation.

Πηγές: 1. http://en.wikipedia.org/wiki/History_of_the_Python_programming_language 2. http://en.wikipedia.org/wiki/Python_Software_Foundation 3. [http://en.wikipedia.org/wiki/Python_\(programming_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))

2. Υπό ποιά άδεια διατίθεται η python;

Python Software Foundation License, άλλως "Python License":
<http://www.python.org/psf/license/>

3. Η python είναι στατική ή δυναμική γλώσσα προγραμματισμού; Αναφέρατε και άλλες γλώσσες προγραμματισμού με τα ίδια χαρακτηριστικά.

Είναι δυναμική γλώσσα: http://en.wikipedia.org/wiki/Dynamic_language Άλλες γλώσσες παρόμοιου τύπου: PHP, Perl, Ruby... http://en.wikipedia.org/wiki/Dynamic_language#Languages

4. Η python ακολουθεί το πρότυπο του δομικού (structured), διαδικαστικού (procedural) ή αντικειμενοστραφούς (object-oriented) προγραμματισμού; Αναφέρατε δύο παραδείγματα γλωσσών από κάθε κατηγορία.

Η python είναι γλώσσα η οποία αντιπροσωπεύει τον πολυ-παραδειγματικό τύπο προγραμματισμού (multi-paradigm)

http://en.wikipedia.org/wiki/Multi-paradigm_programming_language Επιτρέπει structured και OO (Object oriented) programming:

[http://en.wikipedia.org/wiki/Python_\(programming_language\)#Programming_philosophy](http://en.wikipedia.org/wiki/Python_(programming_language)#Programming_philosophy) Βρήκα και ένα μήνυμα όπου αναφέρουν πως δέχεται και procedural programming:
<http://mail.python.org/pipermail/tutor/1999-April/000122.html>

- Παραδείγματα γλωσσών που υποστηρίζουν Object-oriented προγραμματισμό: Python, Ruby, VB.NET, C# - http://en.wikipedia.org/wiki/Object-oriented_programming_language

- Παραδείγματα γλωσσών που υποστηρίζουν Structured προγραμματισμό: ALGOL, Pascal, PL/I, Ada - http://en.wikipedia.org/wiki/Structured_programming#Structured_programming_languages

- Παραδείγματα γλωσσών που υποστηρίζουν Procedural προγραμματισμό: PHP, Pascal, C++ - <http://www.dmoz.org/Computers/Programming/Languages/Procedural/>

5. Αναφέρατε 10 εφαρμογές που είναι γραμμένες σε python.

Deluge BitTorrent Client, Nicotine-Plus, PyChess, PyScrabble, streamtuner, regina, kig, avogadro, exaile, minirok, labyrinth, gmail-notify, django, pylons, yum, viewvc... :-)

Ψάξτε για "python" στο Ubuntu (Applications Add/Remove) ή επισκεφτείτε την ιστοσελίδα

http://en.wikipedia.org/wiki/Python_software

6. Τι κατάληξη έχουν τα αρχεία της python;

Συνήθως .py για να μπορούν να αναγνωρίζονται από το Windows. Σε UNIX/Linux λειτουργικά συστήματα, ασχέτως της κατάληξης, μπορείτε να θέσετε σαν πρώτη γραμμή "#!/usr/bin/python", το λεγόμενο "hashbang": [http://en.wikipedia.org/wiki/Shebang_\(Unix\)](http://en.wikipedia.org/wiki/Shebang_(Unix))

7. Πως εκτελούμε ένα αρχείο python;

1. Μέσω κονσόλας: python myscript.py Αν το αρχείο είναι εκτελέσιμο, μπορεί να εκτελεστεί μ'αυτό τον τρόπο: ./myscript.py

2. Με διπλό κλικ (εαν το αρχείο είναι εκτελέσιμο).

3. Μέσω επιλογής σε κάποιο μενού.

8. Πού βρίσκουμε περισσότερες πληροφορίες για την python και για τα υπάρχοντα modules.

1. <http://docs.python.org>
 2. <http://docs.python.org/3.0>
9. Τι δυνατότητες υπάρχουν για Γραφικό Περιβάλλον Χρήστη (GUI / Graphical User interface); Αναφέρετε τουλάχιστον τρεις.
1. pyGTK - <http://en.wikipedia.org/wiki/PyGTK>
 2. PyQt - <http://en.wikipedia.org/wiki/PyQt>
 3. wxPython (python-wxgtk) - <http://en.wikipedia.org/wiki/WxPython>
10. Γράψτε το γνωστό **"Hello, world"**; πρόγραμμα σε python.
Κοιτάξτε το Μάθημα 0 - Εισαγωγή

Μάθημα 1

1. Πόσοι και ποιοί τύποι μεταβλητών υπάρχουν;
Απάντηση: 1. Αριθμοί:

- Ακέρατοι:

```
#!/usr/bin/python2.6
1; 23; 3493          # Decimal integers
01; 027; 06645      # Octal integers
0x1; 0x17; 0xDA5    # Hexadecimal integers

long(1); long(23); long(99999333493)      # Long decimal integers
long(01); long(027); long(01351033136165) # Long octal integers
long(0x1); long(0x17); long(0x17486CBC75) # Long hexadecimal integers
```

```
#!/usr/bin/python3
1; 23; 3493          # Decimal integers
# Στην python3 οι ακέραιοι τύπου octal χρειάζονται "0o" στην αρχή
0o1; 0o27; 0o6645    # Octal integers
0x1; 0x17; 0xDA5     # Hexadecimal integers

# Στην python3 οι ακέραιοι τύπου long έχουν συνενωθεί με τους κανονικούς ακέραιους int
# Δεν υπάρχει long(), είναι int() τώρα
```

Πηγή: <http://diveintopython3.org/porting-code-to-python-3-with-2to3.html>

2. Δεκαδικοί:

```
$ python3
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 1.0; 23.0222424; 3493
1.0
23.0222424
3493
```

3. Text strings:

```
$ python3
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> "Hello!"
'Hello!'
>>> "Τεστ τεστ"
'Τεστ τεστ'
>>> a = "Τεστ τεστ"
>>> print(a)
Τεστ τεστ
```

2. Δημιουργήστε τις ακόλουθες μεταβλητές:

- Μια μεταβλητή με το όνομα `metavliti21` χρησιμοποιώντας **διπλά** εισαγωγικά με την ακόλουθη τιμή:

```
- Έχεις καμιά ιδέα πού να πάμε;
```

Απάντηση:

```
metavliti21 = "- Έχεις καμιά ιδέα πού να πάμε;"
```

- Μια μεταβλητή με το όνομα `metavliti22` χρησιμοποιώντας **μονά** εισαγωγικά με την ακόλουθη τιμή:

```
- Ναι, ας πάμε στο Καφεωδείο "ΟΡΦΕΑΣ".
```

Απάντηση:

```
metavliti21 = '- Ναι, ας πάμε στο Καφεωδείο "ΟΡΦΕΑΣ".'
```

- Μια μεταβλητή με το όνομα `metavliti23` χρησιμοποιώντας **διπλά** εισαγωγικά με την ακόλουθη τιμή:

```
"Αυτό είναι "παράδειγμα" 'χρήσης' πολλών εισαγωγικών."
```

Απάντηση:

```
metavliti23 = "\"Αυτό είναι \"παράδειγμα\" 'χρήσης' πολλών εισαγωγικών.\""
```

3. Δημιουργήστε μια καινούργια μεταβλητή `metavliti31`.

Θέστε την τιμή της μεταβλητής `metavliti31` συνενώνοντας τις μεταβλητές `metavliti21` και `metavliti22` με μια **κενή γραμμή στο ενδιάμεσο**:

```
- Έχεις καμιά ιδέα πού να πάμε;
(κενή γραμμή)
- Ναι, ας πάμε στο Καφεωδείο "ΟΡΦΕΑΣ".
```

Απάντηση:

Ένας από τους παρακάτω τρόπους θα ήταν αποδεκτός:

```
#!/usr/bin/python3
metavliti31 = print(metavliti21 + "\n\n" + metavliti22)
metavliti31 = print("{0}\n\n{1}".format(metavliti21, metavliti22))
metavliti31 = print("%s\n\n%s" % (metavliti21, metavliti22))
```

4. Ποιοι χαρακτήρες μπορούν να χρησιμοποιηθούν για το **όνομα** μιας μεταβλητής και ποιοι για την **τιμή** μιας μεταβλητής;

Υπάρχει διαφορά για **python3.0** και **python2.6** σε αυτό το σημείο;

Απάντηση:

Για το **όνομα** μιας μεταβλητής:

- python3 - δέχεται χαρακτήρες από το σύνολο **Unicode** (ή/και UTF-8;) που συμβολίζουν γράμματα και αριθμούς (δεν είμαι απόλυτα σίγουρος, κάποιοι χαρακτήρες φαίνεται να μην υποστηρίζονται)
- python2.6 - δέχεται χαρακτήρες από το σύνολο **ASCII**

Για την **τιμή** μιας μεταβλητής:

- python3 - δέχεται χαρακτήρες από το σύνολο **Unicode** (ή/και UTF-8;)
- python2.6 - δέχεται χαρακτήρες από το σύνολο **Unicode** (ή/και UTF-8;) - για αντιπροσώπευση ως Unicode χρειάζεται η χρήση του u"":

```
$ python
Python 2.6.2 (release26-maint, Apr 19 2009, 01:58:18)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> u"Τρώω και μαγειρεύω"
u'\u03a4\u03c9\u03c9 \u03c9 \u03b1\u03b2\u03b3 \u03b4\u03b5\u03b6\u03c7\u03c8\u03c9\u03cd\u03ce'
>>> print(u"Τρώω και μαγειρεύω")
Τρώω και μαγειρεύω
>>> "Τρώω και μαγειρεύω"
'\xc4\xcf\x81\x8e\x89 \xba\xbb\xbc\xbd\xbe\xbf \xb3\xbd\xbe\xbf \xb5\xbd\xbf\x81\x8d\x89'
>>> print("Τρώω και μαγειρεύω")
Τρώω και μαγειρεύω
>>>
```

5. Θέστε δύο μεταβλητές:

```
ilikia=20
onoma="Ανδρέας"
```

Το πρώτο συμβολίζει την ηλικία του Ανδρέα, και το δεύτερο περιέχει το όνομα του.

1. Εκτυπώστε (**print()**), σε μια γραμμή, ένα οποιοδήποτε κείμενο της αρεσκείας σας, χρησιμοποιώντας τις μεταβλητές **ilikia** και **onoma** και την εντολή **format()**.

Απάντηση:

```
#!/usr/bin/python3
ilikia=20
onoma="Ανδρέας"
print("0 {0}, είναι τώρα {1} χρονών, γεροδεμένο παλικάρι.".format(ilikia, onoma))
```

6. **Με ποιόν τελεστή** ελέγχουμε αν ο αριθμός **5** ανήκει στο σύνολο **(1, 2, 3, 4, 5)**.

Απάντηση:

Με το **in**:

```

$ python3
Python 3.0.1+ (r301:69556, Apr 15 2009, 17:25:52)
[GCC 4.3.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 5 in (1, 2, 3, 4, 5)
True
>>> if 5 in (1, 2, 3, 4, 5):
...     print("NAI - ανήκει στο σύνολο")
... else:
...     print("OXI - Δεν ανήκει στο σύνολο (1,2,3,4,5) !")
...
NAI - ανήκει στο σύνολο

```

7. Πέντε αγόρια μαλλώνανε για το ποιος μπορεί να πηδήξει πιο μακριά στο άλμα εις μήκος. Ο πρώτος μπορεί να πηδήσει 1/4 (ένα τέταρτο) του μέτρου **λιγότερο** από τον δεύτερο. Ο τρίτος μπορεί να πηδήσει ένα μέτρο **περισσότερο** του **ενός τρίτου** της απόστασης του **αθροίσματος του πρώτου και δεύτερου**. Ο τέταρτος μπορεί να πηδήσει δύο μέτρα **περισσότερο** της απόστασης της **διαφοράς του τρίτου πλην την απόσταση του πρώτου** αγοριού.
- Αν ο **δεύτερος** μπορεί να πηδήσει **1.75** μέτρα, πόσο μπορούν να πηδήσουν τα πρώτα τέσσερα αγόρια;
 - Αν η **ρίζα** του **αθροίσματος της απόστασης που πηδάνε όλα τα αγόρια μαζί** είναι **3.5**, πόσο μακριά μπορεί να πηδήσει το πέμπτο αγόρι;
 - Παρουσιάστε το αποτέλεσμα για το κάθε αγόρι, καθώς και τα βήματα ολόκληρης της διαδικασίας που πήρατε για να βρείτε το αποτέλεσμα σε γλώσσα προγραμματισμού python (να φαίνεται η χρήση των παραπάνω δεδομένων).

Τοποθετήστε τα **αποτελέσματα** στις ακόλουθες μεταβλητές:

Πρώτο αγόρι: protos, δεύτερο αγόρι: deuterios, τρίτο αγόρι: tritos, τέταρτο αγόρι: tetartos, πέμπτο αγόρι: pemptos

Σημείωση:

- Οι υπολογισμένες τιμές είναι πάνω από ένα μέτρο (> 1.0 m).
- Χρησιμοποιήστε την τελεία "." αντί του κόμματος "," για δεκαδικά ψηφία.

Απάντηση:

Επειδή πρέπει να χρησιμοποιούμε μεταβλητές που έχουμε ήδη θέσει, αρχίζουμε με το deuterios που είναι δεδομένο. Επίσης, στην **python2.6** στη διαίρεση για να πάρουμε τα δεκαδικά χρησιμοποιούμε ".0", π.χ. **3.0** στο διαιρέτη (Εγώ τα χρησιμοποιώ σε όλους τους ακέραιους αριθμούς, ασχέτως αν είναι python3 ή python2.6. :-))

```

#!/usr/bin/python3
deuterios = 1.75
protos = deuterios - 0.25
tritos = (protos + deuterios) / 3.0 + 1.0
tetartos = (tritos - protos) + 2.0
pemptos = 3.5 ** 2 - (protos + deuterios + tritos + tetartos)
print("Πρώτος: {0}\nΔεύτερος: {1} Τρίτος: {2} Τέταρτος: {3} Πέμπτος: {4}".format(protos, deuterios, tritos, tetartos, pemptos))

```