

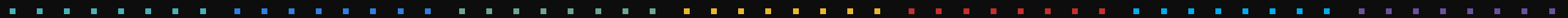
WMNBE-2203 | BACK-END DEVELOPMENT

Python #2

Basics

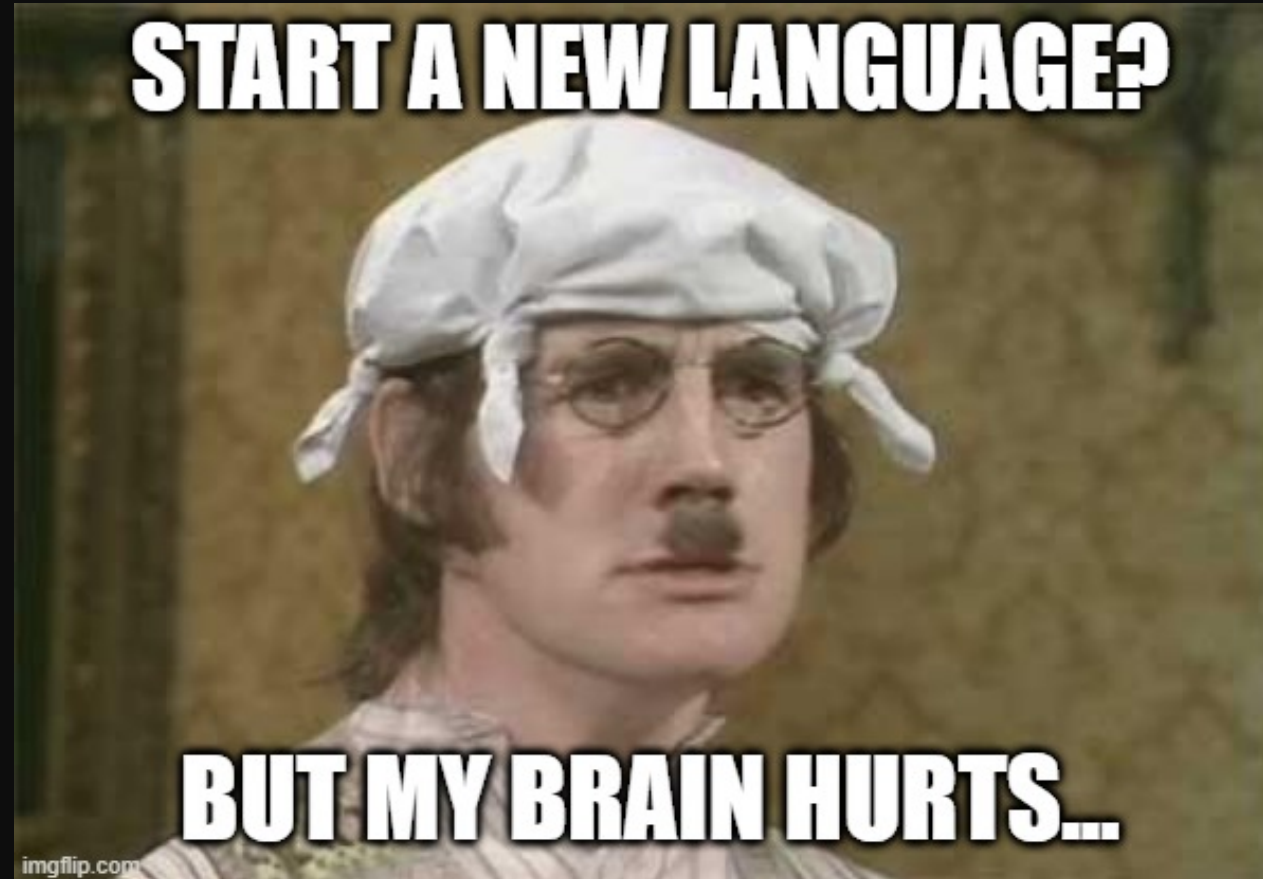


ΠΕΡΙΕΧΟΜΕΝΑ



Περιεχόμενα

- Βασικά στοιχεία
- Μεταβλητές
- Εκφράσεις
- Είσοδος / Έξοδος
- Συναρτήσεις
- Μορφοποίηση



ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ

Βασικά στοιχεία

Δομή

Η **Python** είναι μία γλώσσα με στόχο την απλότητα και την υψηλή αναγνωσιμότητα του κώδικα.

Εστω και μία απλή εντολή σε **Python** μπορεί να εκτελεστεί ως αυτόνομο πρόγραμμα, ένα χαρακτηριστικό γενικά των διερμηνευόμενων γλωσσών.

Βασικά στοιχεία

Δομή

Ακολουθεί τον κανόνα *off-side* (football run), σύμφωνα με τον οποίο τα blocks καθορίζονται από τη στοίχιση (*indentation*) του κώδικα.

Τα σχόλια στην **Python** ορίζονται με το χαρακτήρα **#**.

ΜΕΤΑΒΛΗΤΕΣ

Μεταβλητές

Τύποι δεδομένων / μεταβλητών

Οι μεταβλητές, σε ένα πρόγραμμα, αναφέρονται σε περιοχές στη μνήμη, όπου το πρόγραμμά μας αποθηκεύει τα δεδομένα του.

Οι βασικοί τύποι μεταβλητών στην **Python** είναι οι εξής:

- **int** - ακέραιες τιμές
- **float** - αριθμοί κινητής υποδιαστολής
- **str** - αλφαριθμητικά
- **bool** - λογικές τιμές

Μεταβλητές

Ονοματολογία

Το όνομα μιας μεταβλητής:

- αποτελείται μόνο από γράμματα (**a-z** & **A-Z**), ψηφία (**0-9**) και την κάτω παύλα (**_**).
- είναι *case sensitive*.
- δεν μπορεί να ξεκινά με ψηφίο (**0-9**)
- δεν μπορεί να είναι δεσμευμένη λέξη

Μεταβλητές

Εκχώρηση / Ανάθεση

Η **Python** είναι μία δυναμική γλώσσα. Ο τύπος μιας μεταβλητής δε δηλώνεται πριν χρησιμοποιηθεί και μπορεί να αλλάξει κατά τη διάρκεια του προγράμματος.

Η εκχώρηση / ανάθεση τιμής γίνεται με το σύμβολο της ισότητας **=**.

Μπορεί να γίνει ταυτόχρονη ανάθεση, σε παραπάνω από μία μεταβλητές

```
var1 = <value>
var1 = var2 = ... = varn = <value>
```

Μεταβλητές

Παράδειγμα

```
a = 3                # int
a = 3.14            # float

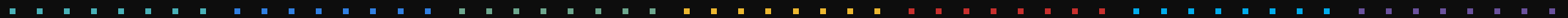
b = 'Python'        # str
b = "Python's great" # str

b = """This is
a multiline
string"""           # str

c = True            # bool

e = f = 'We have the same value'
```

ΕΚΦΡΑΣΕΙΣ



Εκφράσεις

Αριθμητικοί Τελεστές

Τελεστής	Πράξη
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
//	Ακέραια διαίρεση
%	Modulo
**	Ύψωση σε δύναμη

Εκφράσεις

Τελεστές και **str**

Τελεστής	Λειτουργία	Παράδειγμα
+	Συνένωση	'a' + 'b' => 'ab'
*	Επανάληψη	'a' * 3 => 'aaa'

Εκφράσεις

Ανάθεση & Τελεστής

Τελεστής	Χρήση	Ισοδύναμα
<code>+=</code>	<code>x += y</code>	<code>x = x + y</code>
<code>-=</code>	<code>x -= y</code>	<code>x = x - y</code>
<code>*=</code>	<code>x *= y</code>	<code>x = x * y</code>
<code>/=</code>	<code>x /= y</code>	<code>x = x / y</code>
<code>//=</code>	<code>x //= y</code>	<code>x = x // y</code>
<code>%=</code>	<code>x %= y</code>	<code>x = x % y</code>
<code>**=</code>	<code>x **= y</code>	<code>x = x ** y</code>

Εκφράσεις

Συγκριτικοί / Σχεσιακοί Τελεστές

Τελεστής	Πράξη
<code>==</code>	Ίσο με
<code>!=</code>	Διάφορο από
<code>></code>	Μεγαλύτερο από
<code><</code>	Μικρότερο από
<code>>=</code>	Μεγαλύτερο ή ίσο με
<code><=</code>	Μικρότερο ή ίσο με

Εκφράσεις

Λογικοί Τελεστές

A	B	A and B	A or B	not A
True	True	True	True	<i>False</i>
True	<i>False</i>	<i>False</i>	True	<i>False</i>
<i>False</i>	True	<i>False</i>	True	True
<i>False</i>	<i>False</i>	<i>False</i>	<i>False</i>	True

Εκφράσεις

Ιεραρχία

1. Αριθμητικοί

1. ******

2. ***, /, //, %**

3. **+, -**

2. Συγκριτικοί

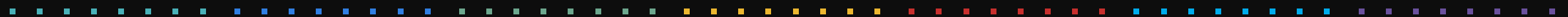
3. Λογικοί

1. **not**

2. **and**

3. **or**

ΕΙΣΟΔΟΣ / ΕΞΟΔΟΣ



Είσοδος / Έξοδος

Είσοδος

Για την είσοδο δεδομένων από το πληκτρολόγιο χρησιμοποιούμε τη συνάρτηση **input()**, η οποία παίρνει ως όρισμα το μήνυμα προτροπής που θα εμφανιστεί στην οθόνη.

Ανεξάρτητα από το τι θα πληκτρολογηθεί, η συνάρτηση επιστρέφει πάντα συμβολοσειρά **str** (στην έκδοση **3.x**). Αν θέλουμε να μετατραπεί σε κάτι άλλο, η μετατροπή αυτή πρέπει να γίνει με τις συναρτήσεις **int()**, **float()**, **bool()**.

Για λόγους πληρότητας, να αναφέρουμε και τη **str()**, που είναι χρήσιμη σε άλλες περιπτώσεις.

Είσοδος / Έξοδος

Παράδειγμα

```
>>> x = input('Please enter a value: ')
Please enter a value: 10
>>> x
'10'
>>> x = int(x)
>>> x
10
>>> y = float(input('Amount: '))
>>> z = input('Enter your name: ')
```

Είσοδος / Έξοδος

Έξοδος

Για την εμφάνιση ενός ή περισσότερων στοιχείων στην οθόνη, χρησιμοποιούμε τη συνάρτηση **print()** (έκδοση 3.x). Η εμφάνιση των στοιχείων μπορεί να παραμετροποιηθεί με τα ορίσματα **sep** (*separator char*) και **end** (*end char*).

Σύνταξη

```
print(στοιχεία, sep=' ', end='\n')
```

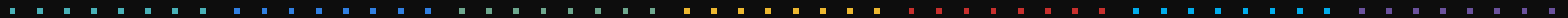
sep: διαχωριστής, **end**: τερματικός χαρακτήρας

Είσοδος / Έξοδος

Παράδειγμα

```
>>> a = 3
>>> b = 8
>>> print(a)
3
>>> print(a, b)
3 8
>>> print(a, b, sep=', ', end='.')
3, 8.
```


ΣΥΝΑΡΤΗΣΕΙΣ



Συναρτήσεις

Βασικές ενσωματωμένες συναρτήσεις

Συνάρτηση	Λειτουργία
abs(x)	Απόλυτη τιμή του x
pow(a,n)	Ύψωση του a στη n
divmod(x,y)	Ακέραια διαίρεση & modulo (πλειάδα)
round(a,n)	Στρογγυλοποίηση του a με ακρίβεια n

Συναρτήσεις

Παράδειγμα

```
>>> abs(-9)
9
>>> pow(3, 3)
27
>>> divmod(15, 6)
(2, 3)
>>> round(3.5, 0)
4.0
>>> round(4.5, 0)
4.0
```

Συναρτήσεις

Μαθηματικές συναρτήσεις

Άλλες μαθηματικές συναρτήσεις είναι διαθέσιμες μέσω της βιβλιοθήκης **math**.

```
>>> import math
>>> math.pi
3.141592653589793
>>> math.e
2.718281828459045
>>> math.floor(4.9)
4
>>> math.ceil(4.9)
5
>>> math.sqrt(9)
3.0
```

Συναρτήσεις

Μαθηματικές βιβλιοθήκες

Επίσης χρήσιμη είναι και η βιβλιοθήκη **random**.

```
>>> import random
>>> random.randint(1, 8)
7
>>> random.random()
0.3031134844827079
```

Συναρτήσεις

Συναρτήσεις συμβολοσειρών

Συνάρτηση	Λειτουργία
len(s)	Μήκος συμβολοσειράς
ord(a)	Μετατροπή του χαρακτήρα a σε <i>unicode</i>
chr(n)	"Αντίστροφη" λειτουργία από τη ord

Συναρτήσεις

Συναρτήσεις συμβολοσειρών

Συνάρτηση	Λειτουργία
s.isalpha()	Αποτελείται μόνο από γράμματα;
s.isdigit()	Αποτελείται μόνο από ψηφία;
s.islower()	Αποτελείται μόνο από πεζά;
s.upper()	Μετατροπή σε κεφαλαία
s.lower()	Μετατροπή σε πεζά
s.capitalize()	Μετατροπή του πρώτου χαρακτήρα σε κεφαλαίο

Συναρτήσεις

Συναρτήσεις συμβολοσειρών

Συνάρτηση	Λειτουργία
<code>s.count(key,start=0,end=len(s))</code>	Πόσες φορές υπάρχει το key
<code>s.find(key,start=0,end=len(s))</code>	Αναζήτηση για το key
<code>s.startswith(key,start=0,end=len(s))</code>	Έλεγχος αν ξεκινά με key
<code>s.replace(old,new[,max])</code>	Αντικατάσταση
<code>s.strip(chr=' ')</code>	Αφαίρεση χαρακτήρων από τα άκρα

Συναρτήσεις

Συναρτήσεις συμβολοσειρών

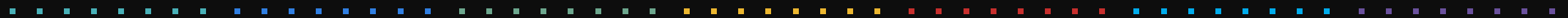
Συνάρτηση	Λειτουργία
<code>s.decode(encoding='UTF-8', errors='strict')</code>	Κωδικοποίηση
<code>s.encode(encoding='UTF-8', errors='strict')</code>	Αποκωδικοποίηση

Συναρτήσεις

Συναρτήσεις συμβολοσειρών

Συνάρτηση	Λειτουργία
<code>sep.join(s)</code>	Συνένωση
<code>s.split(sep="", max=s.count(sep))</code>	Διαχωρισμός

ΜΟΡΦΟΠΟΙΗΣΗ



Μορφοποίηση

string % operator

Μια παλαιότερη τεχνική, ήταν η μορφοποίηση με τη χρήση του τελεστή %

```
>>> print('Bought %d %s for €%.2f' % (2, 'BTC', 50102.24))  
Bought 2 BTC for €50102.24
```

Θυμίζει C και θεωρείται, πλέον, παρωχημένος τρόπος μορφοποίησης.

Μορφοποίηση

.format() method

Νεότερη τεχνική είναι η χρήση της μεθόδου **.format()** του **str**, είτε με *positional arguments*...

```
>>> print('Bought {0} {1} for €{2:,.2f}'
        .format(2, 'BTC', 100204.48))
Bought 2 BTC for €100,204.48
```

Προσοχή στην προσθήκη του , στο **{2:,.2f}**

Μη βασίζεστε στην στρογγυλοποίηση μέσω του **{2:,.2f}**!

Μορφοποίηση

.format() method

... είτε με *keyword arguments*.

```
>>> print('Bought {n} {coin} for €{cost:,.2f}'
      .format(n=2, coin='BTC', cost=100204.48))
Bought 2 BTC for €100,204.48
```

Μορφοποίηση

f-strings

Από την έκδοση **3.6** και μετά, υπάρχει και η δυνατότητα μορφοποίησης με *formatted string literal* ή αλλιώς *f-strings*.

```
>>> n = 2
>>> coin = 'BTC'
>>> cost = 100204.48
>>> print(f'Bought {n} {coin} for €{cost:,.2f}')
Bought 2 BTC for €100,204.48
```

Μορφοποίηση

f-strings

Στα **f-strings** τα *placeholders* δεν είναι απλά *keyword arguments*, αλλά μπορούν να δεχθούν οποιαδήποτε έκφραση.

```
>>> n = 2
>>> coin = 'BTC'
>>> cost_each = 50102.24
>>> print(f'Bought {n} {coin} for €{(cost_each * n):.2f}')
Bought 2 BTC for €100204.48
```


HOMework

Homework #1

Στοιχεία κύκλου

Γράψτε ένα πρόγραμμα σε **Python** το οποίο να ζητά την ακτίνα ενός κύκλου και να εμφανίζει την περίμετρο και το εμβαδόν του.

Περίμετρος: $circumference = 2 \cdot \pi \cdot R$

Εμβαδόν: $area = \pi \cdot R^2$

Homework #2

Μετατροπή νομίσματος

Σε κάποιο σύστημα τα ποσά εμφανίζονται σε λάθος μορφή, λόγω κάποιου software bug. Γράψτε ένα πρόγραμμα σε **Python** το οποίο να μετατρέπει ένα ποσό, από τη μορφή **\$12,345.67** στη μορφή **€12.345,67**, με τα εξής βήματα:

1. να αφαιρεί τυχόν κενά από την αρχή και το τέλος του **string**
2. να αντικαθιστά την **.** με **,** και το ανάποδο (προσοχή, είναι λίγο δυσκολότερο από ότι φαίνεται αρχικά).
3. να αντικαθιστά το **\$** με **€**.


Χρήσιμα links

 3.x Documentation
<https://docs.python.org/3/>

 The Python Standard Library
<https://docs.python.org/3/library/>

 Python Data Types
https://www.w3schools.com/python/python_datatyp...

 Built-in Functions
<https://docs.python.org/3/library/functions.html#bool>

 Numeric and Mathematical Modules
<https://docs.python.org/3/library/numeric.html>

 Strings — Python in a Hurry
<https://pyhurry.readthedocs.io/en/latest/strings.html>

 Python - Strings - Tutorialspoint
https://www.tutorialspoint.com/python/python_strings...

 A Guide to the Newer Python String Format Techniques – Real Python
<https://realpython.com/python-formatted-output/>


Extra info

 Off-side rule - Wikipedia


https://en.wikipedia.org/wiki/Off-side_rule

 Why doesn't $0.1 + 0.2 - 0.3$ equal 0.0 in Python? - Quora

<https://www.quora.com/Why-doesnt-0-1-0-2-0-3-equ...>

 Exponentials in python: $x^{**}y$ vs `math.pow(x, y)` - Stack Overflow

<https://stackoverflow.com/questions/20969773/expon...>

 Python `round()` Function - Learn By Example

<https://www.learnbyexample.org/python-round-funct...>

 How to Format Number as Currency String in Python

<https://stackabuse.com/format-number-as-currency-...>

THANK YOU!