Python – Εισαγωγή

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών

Η γλώσσα Python

Το 1989 ο Guido Van Rossum δημιούργησε τη γλώσσα Python

Η ονομασία της δεν προέρχεται από το φίδι πύθωνα αλλά από το γεγονός ότι ο Guido λάτρευε τους Monty Pythons

- Το 1994 παρουσιάστηκε η έκδοση 1.0 της Python
- Το 2000 παρουσιάστηκε η έκδοση 2.0 της Python
- Το 2008 παρουσιάστηκε η έκδοση 3.0 της Python
- Σήμερα χρησιμοποιούμε την έκδοση 3.11.3

Η γλώσσα Python

- Είναι μια γλώσσα με απλή σύνταξη που διαβάζεται εύκολα
- Ο κώδικας της είναι ανοικτός και υποστηρίζεται από μια μεγάλη κοινότητα προγραμματιστών παγκοσμίως
- Οι διαθέσιμες βιβλιοθήκες είναι χιλιάδες και εξυπηρετούν ερευνητικούς σκοπούς όπως ανάλυση δεδομένων αλλά και κατασκευή ειδικών προγραμμάτων και παιχνιδιών
- Υποστηρίζει διαφορετικά μοντέλα προγραμματισμού όπως διαδικαστικό αλλά και αντικειμενοστρεφές

Η γλώσσα Python

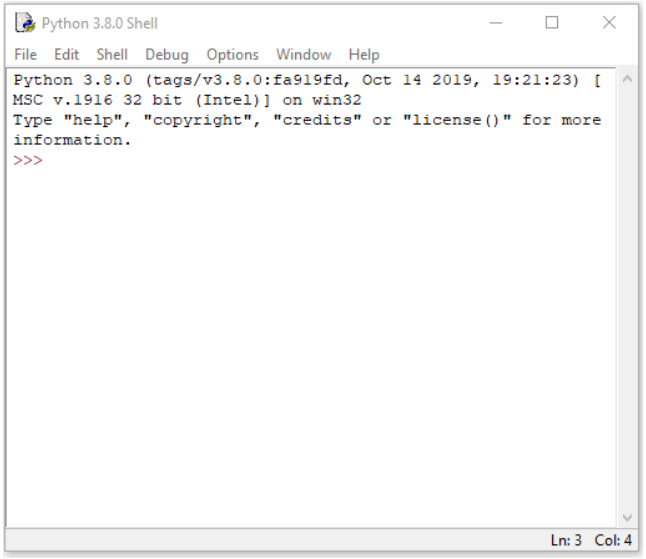
- Επίσημη ιστοσελίδα της Python
- Κεντρική βιβλιογραφία της Python
- Πληροφορίες για τη βασική βιβλιοθήκη της γλώσσας
- Λίστα με βιβλιοθήκες της Python

Εγκασταση της γλώσσας

Στο python.org επιλέγουμε την έκδοση 3 της Python

Το πακέτο που θα γίνει download περιλαμβάνει:

Το περιβάλλον ανάπτυξης **IDLE**



Τα εργαλεία για τη διαχείριση βιβλιοθηκών όπως το Pip3 μέσω του οποίου μπορούμε να εγκαταστήσουμε νέα python modules με την εντολή pip3 install

Μεταβλητές

Το όνομα μιας μεταβλητής:

- μπορεί να περιλαμβάνει λατινικούς χαρακτήρες ή αριθμούς (0…9) καθώς και το _
- αλλά **δεν** μπορεί να αρχίζει με αριθμητικό ψηφίο.

Παραδείγματα:

y = 7
firstname = “Νίκος“
pi = 3.14

Ποια είναι αποδεκτά ονόματα μεταβλητών;

1) b323 2) _33375a 3)1223_ass 4) _sddds

Μεταβλητές

Τύποι αριθμητικών μεταβλητών:

- **<int>** ακέραιος (integer)
- **<float>** δεκαδικός (floating point)
- **<str>** συμβολοσειρά (string)

Με την εντολή type(x) εμφανίζεται ο τύπος μιας μεταβλητής

Εκχώρηση τιμής σε μεταβλητή

Παραδείγματα:

```
b = 5
type(a)

z = z**3 + 10
x = “Κώστας ” - ”Νίκος”
```

Σύμβολα πράξεων:

+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
//	Ακέραια διαίρεση (πηλίκο)
%	modulo (υπόλοιπο διαίρεσης)
**	Ύψωση σε δύναμη

Μετατροπή βαθμών Celsius σε Fahrenheit:

1) 10^{o} 2) 0^{o}

Εκχώρηση τιμής σε μεταβλητή

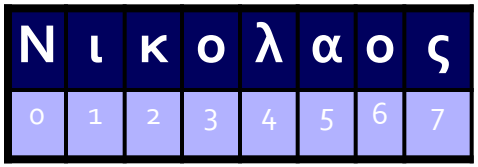
Συντομογραφίες πράξεων:

- **x += y** Πρόσθεση του y στην τιμή του x (x = x + y)
- **x -= y** Αφαίρεση του y από την τιμή του x (x = x – y)
- **x *= y** Πολλαπλασιασμός του y με την τιμή του x (x = x * y)
- **x /= y** Διαίρεση του y με την τιμή του x (x = x / y)
- **x //= y** Ακέραια διαίρεση του y με x (x = x // y)
- **x %= y.** Υπόλοιπο της διαίρεσης του y με x (x = x % y)
- **x **= y** ΄Υψωση του y στο x (x = x **y)

Δημιουργία συμβολοσειράς (string)

Μια συμβολοσειρά είναι μια ακολουθία χαρακτήρων που περικλείεται από μονά ή διπλά εισαγωγικά
Κάθε χαρακτήρας έχει τη θέση του στο αλφαριθμητικό οι οποίες ξεκινούν από τη θέση 0

Παραδείγματα:



name="Νικόλαος" (8 χαρακτήρες)
city= 'Πάτρα'

name[0] : N

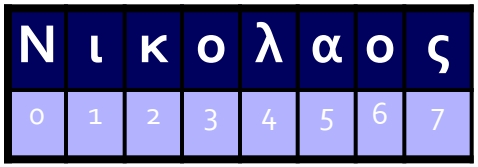
Οι χαρακτήρες αποθηκεύονται ως ακέραιοι μήκους από 1 έως 4 bytes
Η συνάρτηση ord(character) επιστρέφει τον κωδικό του χαρακτήρα
Η συνάρτηση chr(integer) επιστρέφει το χαρακτήρα που αντιστοιχεί ο κωδικός.

Τεμαχισμός συμβολοσειράς (slice)

Ένα τμήμα μιας συμβολοσειράς αποτελείται από το χαρακτήρα της αρχής μέχρι το χαρακτήρα τέλους, χωρίς να περιλαμβάνεται ο τελευταίος.

Παράδειγμα:

s="Νικόλαος" (8 χαρακτήρες)



s[2:6]
'κολα'

Αντίστοιχα τι θα εμφανιστεί για τα παρακάτω;
s[2:]
και
s[:2]

Πράξεις με συμβολοσειρές

Οι συμβολοσειρές αποτελούν αμετάβλητες ακολουθίες χαρακτήρων και **δεν επιτρέπεται η αλλαγή τους.**

Επιτρέπεται όμως η **πρόσθεση** συμβολοσειρών όπως και ο **πολλαπλασιασμός** συμβολοσειράς με ακέραιο.

τελεστής	**αποτέλεσμα**
<seq> + <seq>	συνένωση
<seq> * <int>	επανάληψη
<seq>[]	δείκτης
len(<seq>)	μήκος ακολουθίας
<seq>[:]	τεμαχισμός
for <var> in <seq>:	επανάληψη
<expr> in <seq>	συμμετοχή (Boolean)

Επιπλέον η συνάρτηση len(s) μας δίνει το πλήθος χαρακτήρων της συμβολοσειράς

Μέθοδοι συμβολοσειρών

Υπάρχουν πολλές μέθοδοι που μπορούν να εφαρμοστούν σε μια συμβολοσειρά string για αναζήτηση, μετατροπή, εξαγωγή λέξεων, έλεγχο αν το κείμενο αποτελείται από αριθμούς ή γράμματα κ.α.

Μερικές από αυτές είναι οι παρακάτω:

islower() - αληθές αν το str έχει μόνο πεζά
isdigit () - αληθές αν το str έχει μόνο αριθμούς
strip([chars]) – αφαιρεί τους χαρακτήρες από την αρχή και το τέλος του str
upper() - μετατρέπει τα κεφαλαία σε μικρά
lower() – μετατρέπει τα πεζά σε κεφαλαία
count() – μέτρηση του αριθμού εμφάνισης ενός χαρακτήρα σε ένα str
isalpha() – αληθές αν περιέχονται μόνο χαρακτήρες

Εντολές output

print()

print(ορίσματα, sep =‘δ’, end = ‘ε’)

δ : διαχωριστής

ε : τερματικός χαρακτήρας

Ποιο θα είναι το αποτέλεσμα των:

print(1,2, sep="*")

print(1, end=',')

Εντολές output

```
>>> x = 124.456

>>> print("η τιμή του χ είναι:", x)

η τιμή του χ είναι: 124.456

>>> print("η τιμή του χ είναι %f" % (x))

η τιμή του χ είναι 124.456000

>>> print("η τιμή του χ είναι %1.2f" % (x))

η τιμή του χ είναι 124.46
```

Εντολές input

input()

x=input(“Δώσε μια τιμή:”)

Προσοχή: Η input() επιστρέφει συμβολοσειρά

Τι είναι λάθος στο παρακάτω παράδειγμα:

x = input(“Δωσε 1ο αριθμό”)

y = input(“Δωσε 2ο αριθμό”)

s=x+y

Δομή δεδομένων Λεξικού (dictionary)

Ορισμός λεξικού ακολουθία από ζευγάρια τύπου:
dict={κλειδί: τιμή, …} μέσα σε { }

katalogos = {“Κώστας”:123456,”Σπύρος”:122344}

Τα κλειδιά πρέπει να είναι **μοναδικά** - επιτρέπονται αμετάβλητοι τύποι δεδομένων όπως αριθμοί, string, tuples, αλλά όχι list, dictionaries

Μέθοδοι λεξικών:
keys() – επιστρέφει τα κλειδιά
values() – επιστρέφει τις τιμές των στοιχείων
items() – επιστρέφει μια λίστα από τα αντικείμενα

Δομή επανάληψης for

for μέλος in αντικείμενο :

ομάδα εντολών 1

else:

ομάδα εντολών 2

(η ομάδα εντολών 2 δεν εκτελείται αν υπάρχει break)

for/break/continue

```
for μέλος in αντικείμενο :
        ομάδα εντολών 1

        if συνθήκη :
                continue
        if συνθήκη :
                break

else:

        ομάδα εντολών 2
```

range: επανάληψη for με απαρίθμηση

for i **in** range(start, end, step) : ομάδα εντολών 1

Συνοπτικές λίστες (list comprehension)

Συνοπτικός τρόπος για τη δημιουργία λίστών χωρίς να γίνει χρήση χρήση της δομής επανάληψης for

[εκφρ(x) for x in Obj συνθήκη]

Παράγει λίστα με στοιχεία τις εκφρ(x) για κάθε τιμή του x που ανήκει στο αντικείμενο Obj που ικανοποιεί τη συνθήκη

Εμβέλεια μεταβλητών

- Εμβέλεια ορίζεται η περιοχή του προγράμματος στην οποία είναι γνωστή μια μεταβλητή
- Οι **τοπικές** μεταβλητές έχουν εμβέλεια μόνο στη συνάρτηση ορισμού τους και ορίζονται μέσα σε συναρτήσεις
- Οι μεταβλητές λέγονται καθολικές (**global**): όταν ορίζονται εκτός συναρτήσεων και έχουν εμβέλεια σε ολόκληρο το πρόγραμμα
- Ορισμός τοπικής μεταβλητής ως **global**: οι αλλαγές της επηρεάζουν την συνώνυμη καθολική μεταβλητή

Βιβλιοθήκες / modules

- **import** module_name
- **from** module_name **import** *

Η βιβλιοθήκη module_name είναι ένα αρχείο python, με όνομα module_name.py το οποίο φορτώνεται ώστε τα στοιχεία που περιέχει να μπορούν να χρησιμοποιηθούν στο πρόγραμμά μας.

Επιπλέον μπορούμε να δημιουργήσουμε τα δικά μας modules και να τα καλέσουμε από το πρόγραμμά μας.

Python modules

random	δημιουργία τυχαίων αριθμών
math	μαθηματικές συναρτήσεις
os os.path	διεπαφή λειτουργικού συστήματος
urllib.request	ανάκτηση ιστοσελίδων
sqlite3	βάση δεδομένων SQLite
csv	διαχείριση αρχείων csv
socket	διαδικτυακή επικοινωνία διεργασιών
cgi	Common Gateway Interface
wsgiref	υλοποίηση διεπαφής WSGI

try / except / finally

Exception : η κατάσταση σφάλματος στην οποία εισέρχεται ένα πρόγραμμα.

Η δομή **try / except** μας επιτρέπει να αντιμετωπίσουμε τα σφάλματα έτσι ώστε να μην διακοπεί η λειτουργία του προγράμματος.

```
try:
εντολές
…
except <τύπος σφάλματος 1> :
εντολές
except <τύπος σφάλματος 2> :
εντολές
finally :
εντολές
```

Διαχείριση αρχείων

Στα αρχεία αποθηκεύονται δεδομένα που μπορούμε να διαβάσουμε. Επιπλέον μπορούμε να γράψουμε δεδομένα σε αρχείο που υπάρχει ήδη, αλλά και να το δημιουργήσουμε (αν δεν υπάρχει).

Υπάρχουν δυο είδη αρχείων: κειμένου ή αρχεία bytes

Αρχεία κειμένου

fobj = **open**(όνομα, κατάσταση, encoding= 'utf-8')
Κατάσταση = '**w**' (write), '**r**' (read), '**a**' (append)
fobj.**name** # the name of the file
fobj.**read**() # read content of file
fobj.**seek**(0) # set current position at start
fobj.**close**() # close file

Σύνδεση με λειτουργικό σύστημα

Οι βιβλιοθήκες **os**, **os.path**, επιτρέπουν τη διεπαφή με το λειτουργικό σύστημα, τη δημιουργία και τη διαγραφή φακέλων, καθώς και την προβολή του περιεχομένου τους.

Επίσης η συνάρτηση **os.system()** μας επιτρέπει να εκτελούμε εντολές της γραμμής εντολών.

Η βιβλιοθήκη os

os.**getcwd**() # current folder
os.**chdir**(d) # change folder
os.**listdir**(d) # content of folder
os.**rename**("t1.txt", "t2.txt") # rename file
os.**remove**(file_name) # delete file
os.**mkdir**("newdir") # create folder
os.**rmdir**(dirname) # delete folder
os.**system**('mkdir newdir') # command line
os.**walk**(d) # walk the generated file tree
os.**sep** # separate character for folders (/)

Η βιβλιοθήκη os.path

os.path.**isdir**(filename) # ελέγχει αν είναι φάκελος
os.path.**split**(filename) # χωρίζει φάκελο-αρχείο
os.path.**splitext**(filename) # χωρίζει επέκταση
os.path.**dirname**(filename) # επιστρέφει φάκελο
os.path.**basename**(filename) # επιστρέφει αρχείο
os.path.**join**(dir, f) # ενώνει φάκελο+αρχείο

Regular expressions

Η βιβλιοθήκη re επιτρέπει τη χρήση της γλώσσας αναγνώρισης προτύπων σε κείμενο Regular Expressions (regex).

Μέσω της re μπορεί να γίνει αναζήτηση προτύπων σε κείμενα, καθώς και εκτέλεση πράξεων όπως ανάκτηση, μετατροπή κ.α. βάσει αυτών των προτύπων.

Regular expressions - Συναρτήσεις

re.**search**(pattern, string): True/False (αναζήτηση)

re.**sub**(pat1,pat2, string, max=0): τροποποιημένο string (αντικατάσταση)

re.**findall**(pattern, string): λίστα ευρημάτων (εύρεση)

re.**split**(pattern, string): λίστα από string (διαχωρισμός)

re.**compile**(pattern): αντικείμενο pattern* (μετάφραση προτύπου)

Regular expressions – Κλάσεις χαρακτήρων

\w Οποιοσδήποτε αλφαριθμητικός χαρακτήρας [a-zA-Z0-9_]
\W Οτιδήποτε εκτός αλφαριθμητικών χαρακτήρων [^a-zA-Z0-9_]
\d Αριθμός, [0-9]
\D Οτιδήποτε εκτός αριθμού, [^0-9]
\s Κενό [\n\f\t\v\r]
\S Οτιδήποτε εκτός του κενού [^[\n\f\t\v\r]

\b Όριο μεταξύ αλφαριθμητικού και μη-αλφαριθμητικού (αρχή ή τέλος λέξης)

Ειδικοί χαρακτήρες: . ^ $ * + ? { } [ ] \ | ( )

Ανάκτηση δεδομένων από το διαδίκτυο

Βιβλιοθήκη urllib.request

```
req_object = urllib.request.Request(διαδ. πόρος)
mypage = urllib.request.urlopen(req_object)
html = mypage.read().decode()
```