

Προγραμματισμός Η/Υ

Τμήμα: Μηχανικών Χωροταξίας, Πολεοδομίας και Περιφερειακής Ανάπτυξης

Τίτλος Επιστημονικού Πεδίου: Πληροφορική και Ανάλυση Δεδομένων

Κωδικός Μαθήματος: ME0200

Τίτλος Μαθήματος: Προγραμματισμός Η/Υ

Κατηγορία Μαθήματος: Επιλογής

Εξάμηνο: Εαρινό

Περίγραμμα του μαθήματος

Το μάθημα εισάγει τους φοιτητές στις βασικές έννοιες και αρχές του προγραμματισμού και είναι προσαρμοσμένο στις ανάγκες και στο υπόβαθρο των χωροτακτών. Στόχος του μαθήματος είναι να αποκτήσουν οι φοιτητές τις αναγκαίες γνώσεις για την επεξεργασία και ανάλυση δεδομένων, την αυτοματοποίηση διαδικασιών και την σύνταξη σεναρίων (scripts) για την αναπαραγωγισιμότητα της ερευνάς τους. Πέρα από τις βασικές αρχές προγραμματισμού η διδασκαλία επεκτείνεται σε εξειδικευμένες θεματικές ενότητες που αφορούν τα γεωχωρικά δεδομένα και την ανάλυσή τους μέσω προγραμματισμού. Η διδασκαλία θα στηριχθεί στην [Python](#), μια σύγχρονη, ευρέως διαδεδομένη και υψηλού επιπέδου γλώσσα προγραμματισμού. Επιπλέον, όπου κριθεί αναγκαίο, θα επιδειχτούν συμπληρωματικά διαδικασίες με την γλώσσα προγραμματισμού [R](#).

Το πρόγραμμα των διαλέξεων καθώς και η προτεινόμενη βιβλιογραφία παρατίθενται στην συνέχεια.

2. Τιμές, τύποι και μεταβλητές. Συμβολοσειρές

Σταθερές (Constants)

Η Python δεν διαθέτει προκαθορισμένες *σταθερές* όπως άλλες γλώσσες προγραμματισμού. Όμως κατά σύμβαση και όχι κατά κανόνα έχει συμφωνηθεί οι *σταθερές* να ονοματίζονται με κεφαλαίους χαρακτήρες. Η αδυναμία της Python στην περίπτωση της δήλωσης *σταθερών* είναι ότι επιτρέπεται η αλλαγή των τιμών τους Παρακάτω παρατίθεται ένα παράδειγμα δήλωσης *σταθερών*.

```
RATIO_FEET_TO_METERS = 3.281
RATIO_LB_TO_KG = 2.205
PI = 3.14
```

Κυριολεκτικές σταθερές (literal constants)

Η κυριολεκτική *σταθερά* ή τιμή είναι ένας αριθμός, ή χαρακτήρας ή μιά συμβολοσειρά. Για παράδειγμα τα παρακάτω αποτελούν τιμές: 3.25 (στην python η υποδιαστολή ορίζεται με . και όχι ,), «ένα τυχαίο κείμενο», 5.25e-1. Αυτές οι τιμές δεν μεταβάλλονται κατά τη διάρκεια εκτέλεσης του προγράμματος γι' αυτό και λέγονται σταθερές. Μπορούν να εκχωρηθούν σε μεταβλητές και να χρησιμοποιηθούν σαν τελεστές σε λογικές εκφράσεις ή σαν παραμέτροι σε συναρτήσεις.

Τύποι δεδομένων

Οι τιμές ανήκουν σε τρεις τύπους δεδομένων (data types) ή κλάσεις (class):

- τους ακέραιους αριθμούς (integer) π.χ. το 15
- τους αριθμούς κινητής υποδιαστολής (floating point) π.χ. το 201.25)
- τις συμβολοσειρές (string) π.χ. το «Time is money»

Με την εντολή `type` ο διερμηνευτής μας απαντάει με τον τύπο της τιμής, όπως παρακάτω:

```
type("No news, good news.")
```

```
str
```

Η Python είναι *Dynamic typing* δηλαδή δεν ο τύπος των μεταβλητών δεν προκαθορίζεται κατά την συγγραφή αλλά κατά την εκτέλεση.

Κανόνες ονοματοδοσίας μεταβλητών

Τα ονόματα των μεταβλητών στην Python υπακούουν στους παρακάτω κανόνες:

- Το όνομα μίας μεταβλητής μπορεί να ξεκινά από ένα γράμμα ή από κάτω πάλυα.
- Το όνομα μίας μεταβλητής δεν μπορεί με αριθμό.
- Το όνομα μίας μεταβλητής μπορεί να περιέχει μόνο αλφαριθμητικούς χαρακτήρες.
- Στα ονόματα των μεταβλητών γίνεται διάκριση ανάμεσα σε πεζά και κεφαλαία (case sensitive).
- Οι δεσμευμένες λέξεις της Python (keywords) δεν μπορούν να χρησιμοποιηθούν σε ονόματα μεταβλητών.

Συμβολοσειρές (Strings)

Μια συμβολοσειρά είναι μια ακολουθία από χαρακτήρες όπως το `"Το πεπρωμένον φυγείν αδύνατον."`. Μπορεί να είναι σε κάθε γλώσσα που υποστηρίζεται από το πρότυπο Unicode. Οι συμβολοσειρές περικλείονται σε μονά, διπλά ή τριπλά εισαγωγικά. Με τριπλά εισαγωγικά μπορούν να ενσωματωθούν με ευκολία συμβολοσειρές σε πολλές γραμμές και πολλαπλά εισαγωγικά εντός αυτών. Ακολουθούν παραδείγματα συμβολοσειρά.

Χαρακτήρες διαφυγής,κενά, νέες γραμμές

Μπορούμε να σπάσουμε μια συμβολοσειρά κατά την συγγραφή σε νέα γραμμή με τον χαρακτήρα `\` και κατά την εκτέλεση με τον χαρακτήρα `\n` π.χ.

```
message = 'There is no smoke \
without fire'
print(message)
```

```
There is no smoke without fire
```

```
message = 'There is no smoke \nwithout fire'
print(message)
```

```
There is no smoke
without fire
```

Ή να ορίσουμε κενά με το `\t`

```
message = 'There is no smoke \twithout fire'
print(message)
```

```
There is no smoke      without fire
```

Ο χαρακτήρας `\` είναι χαρακτήρας διαφυγής που απενεργοποιεί την ειδική λειτουργία των παραπάνω ή την παράθεση εισαγωγικών μέσα σε εισαγωγικά.

```
print('There is no smoke \n without fire')
```

```
There is no smoke \n without fire
```

```
print('Where there\'s a will, there\'s a way')
```

```
Where there's a will, there's a way
```

Ανεπεξέργαστες συμβολοσειρές (Raw Strings)

Παρόμοιο αποτέλεσμα με τα παραπάνω πετυχαίνουμε τις ανεπεξέργαστες συμβολοσειρές οι οποίες ορίζονται με ένα `r` σαν πρόθεμα

```
print(r"It was made by \n συνέχεια")
```

```
It was made by \n συνέχεια
```

Αφαίρεση κενών

Σε αρκετές περιπτώσεις οι συμβολοσειρές περιέχουν κενά είτε στην αρχή είτε στο τέλος. Για παράδειγμα οι παρακάτω συμβολοσειρές δεν είναι το ίδιες για την Python. Και επιβεβαιώνεται σε μέσω ελέγχου ισότητας.

```
departmentA='ΤΜΧΠΑ'  
departmentB = ' ΤΜΧΠΑ '  
print(departmentA == departmentB) #not equal
```

```
False
```

Για την αφαίρεση των κενών αριστερά, δεξιά ή ταυτόχρονα και στις δύο πλευρές της συμβολοσειρας χρησιμοποιούμε την μέθοδο `strip` και τις παραλλαγές της `rstrip` και `lstrip`

```
print(departmentB.rstrip())  
print(departmentB.lstrip())  
print(departmentB.strip())
```

```
ΤΜΧΠΑ  
ΤΜΧΠΑ  
ΤΜΧΠΑ
```

Συνένωση (Concatenation) συμβολοσειρών

Η απλή παράθεση συμβολοσειρών οδηγεί στην συνενωσή τους δηλ.

```
message = "Curiosity " "killed " 'the ' "'cat'"  
print(message)
```

```
Curiosity killed the cat
```

Συνένωση συμβολοσειρών και μεταβλητών

Η συνένωση μεταβλητών και συμβολοσειρών γίνεται με τον τελεστή `+`.

```
city='Βόλος'  
perifereia='Θεσσαλία'  
print('Ο '+city+' είναι πόλη της Ελλάδα στην ' +perifereia)
```

```
Ο Βόλος είναι πόλη της Ελλάδα στην Θεσσαλία
```

Η μέθοδος format

Άλλη μια πιο πρακτική μέθοδος κατά την συνένωση μεταβλητών και συμβολοσειρών είναι η μέθοδος `format`.

```
print('Ο {0} έχει υψόμετρο {1} μέτρα'.format("Όλυμπος", 2918))  
print('Ο {} έχει υψόμετρο {} μέτρα'.format("Όλυμπος", 2918))  
print('Ο {name} έχει υψόμετρο {height} μέτρα'.format(name="Σμόλικας", height= 2637))
```

```
O Όλυμπος έχει υψόμετρο 2918 μέτρα
O Όλυμπος έχει υψόμετρο 2918 μέτρα
O Σμόλικας έχει υψόμετρο 2637 μέτρα
```

Δεσμευμένες λέξεις (reserved words)

Ορισμένες λέξεις έχουν ιδιαίτερη σημασία για την Python και δεν μπορούν να χρησιμοποιηθούν σαν ονόματα μεταβλητών. Τα παρακάτω κομμάτια κώδικα θα εκδηλώσουν σφάλμα μεταγλώττισης.

Πρόκειται για 33 λέξεις στην τρέχουσα έκδοση της Python. Μπορούμε να δούμε ποιές είναι αυτές οι δεσμευμένες λέξεις με την παρακάτω εντολή:

```
help("keywords")
```

Here is a list of the Python keywords. Enter any keyword to get more help.

False	class	from	or
None	continue	global	pass
True	def	if	raise
and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield
break	for	not	

Η εντολή help

Γενικά με την εντολή `help` καλούμε για βοήθεια και πληροφορίες την Python:

```
help(print)
```

Help on built-in function print in module builtins:

```
print(...)
  print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

  Prints the values to a stream, or to sys.stdout by default.
  Optional keyword arguments:
  file: a file-like object (stream); defaults to the current sys.stdout.
  sep:   string inserted between values, default a space.
  end:   string appended after the last value, default a newline.
  flush: whether to forcibly flush the stream.
```

```
help(abs)
```

Help on built-in function abs in module builtins:

```
abs(x, /)
  Return the absolute value of the argument.
```

```
help(max)
```

Help on built-in function max in module builtins:

```
max(...)
  max(iterable, *[, default=obj, key=func]) -> value
  max(arg1, arg2, *args, *[, key=func]) -> value

  With a single iterable argument, return its biggest item. The
  default keyword-only argument specifies an object to return if
  the provided iterable is empty.
  With two or more arguments, return the largest argument.
```

Αλλαγή Πεζών Κεφαλαίων (Convert case)

Μπορούμε να κάνουμε αλλαγή ανάμεσα σε κεφαλαία και πεζά με τις παρακάτω μεθόδους συμβολοσειρών: `upper()`, `title()`, `lower()`. Αξίζει να σημειώσουμε ότι οι μέθοδοι αυτές δεν έχουν επίδραση στην μεταβλητή που τις καλούμε αλλά πρέπει να επαναεκχωρήσουμε το αποτέλεσμα της μεθόδου στην μεταβλητή με το ίδιο όνομα.

```
agios="άγιος νικόλαος"

print(agios.upper())
print(agios) # ο agios παραμένει "άγιος νικόλαος"

print(agios.title())
print('ΑΓΙΑ ΕΛΕΝΗ'.lower())

agios = agios.upper()
print(agios) # ο agios μετά την εκχώρηση στην ίδια μεταβλητή γίνεται ΑΓΙΟΣ
ΝΙΚΟΛΑΟΣ
```

```
ΑΓΙΟΣ ΝΙΚΟΛΑΟΣ
άγιος νικόλαος
Άγιος Νικόλαος
αγία ελένη
ΑΓΙΟΣ ΝΙΚΟΛΑΟΣ
```

Οι συμβολοσειρές είναι μη μεταβαλλόμενη δομή δεδομένων

Οι συμβολοσειρές αποτελούνται από ακολουθίες χαρακτήρων με σταθερό μέγεθος και μη μεταβαλλόμενα περιεχόμενα. Αυτό σημαίνει ότι δεν είναι δυνατόν να προστίθενται ή να αφαιρούνται χαρακτήρες, ούτε να τροποποιούνται τα περιεχόμενα του αλφαριθμητικού. Πρόκειται για μια μη μεταβαλλόμενη (immutable) δομή της Python. Η αρίθμηση των χαρακτήρων σε ένα αλφαριθμητικό ξεκινάει από το 0.

Έτσι στην συμβολοσειρά `country = Ελλάδα` έχουμε:

`country[0]` → Ε (η αρίθμηση ξεκινά από το 0)

`country[1]` → λ

`country[2]` → λ

`country[3]` → ά

`country[4]` → δ

`country[5]` → α

Η παραπάνω συμβολοσειρά έχει μήκος 6 χαρακτήρες.

Μήκος συμβολοσειράς

Μέσω της συνάρτησης `len` η Python μας επιστρέφει το μήκος συμβολοσειράς δηλαδή το πλήθος των χαρακτήρων (μαζί με τα κενά) από τους οποίους αποτελείται.

```
message = 'Η τώρα ή ποτέ.'
len(message)
```

14

Η μέθοδος find

Η μέθοδος `find` μας επιτρέπει να αναζητήσουμε μια συμβολοσειρά μέσα σε μια άλλη συμβολοσειρά. Η μέθοδος μας επιστρέφει την τοποθεσία από την ξεκινάει η αναζητούμενη συμβολοσειρά δηλαδή τον δείκτη (index) στην οποία εντοπίζεται ο πρώτος χαρακτήρας της αναζητούμενης συμβολοσειράς μέσα στα περιεχόμενα της αρχικής συμβολοσειράς. Στην παρακάτω συμβολοσειρά θα αναζητήσουμε την λέξη `ποτέ`.

```
stixos = 'Η Ελλάδα ποτέ δεν πεθαίνει'
index = stixos.find('ποτέ')
```

Κανονικά αν πάμε στον χαρακτήρα με ευρετήριο (index) 9 πρέπει να εντοπίσουμε τον πρώτο χαρακτήρα της συμβολοσειράς που είναι το **π**. Πράγματι:

```
stixos[index]
```

```
'π'
```

Αν δεν εντοπιστεί η λέξη που αναζητούμε στην συμβολοσειρά η Python θα επιστρέψει: **-1**

```
stixos.find('πάντα')
```

```
-1
```

Η αναζήτηση είναι case sensitive δηλαδή γίνεται διάκριση ανάμεσα σε πεζά και κεφαλαία.

```
stixos.find('Ελλάδα') # επιστρέφει τον δείκτη 2 γιατί εντοπίστηκε η λέξη κλειδί
```

```
2
```

```
stixos.find('ΕΛΛΑΔΑ') # επιστρέφει -1 γιατί δεν εντοπίστηκε η λέξη κλειδί
```

```
-1
```

Μια άλλη σημαντική μέθοδος των συμβολοσειρών είναι η μέθοδος **replace** κατά την οποία μπορούμε να αντικαταστήσουμε τα περιεχόμενα μιας συμβολοσειράς. Στην πρώτη παράμετρο ορίζουμε την συμβολοσειρά που θέλουμε να αντικαταστήσουμε με την δεύτερη παράμετρο.

```
stixos.replace('ποτέ', 'πάντα')
```

```
'Η Ελλάδα πάντα δεν πεθαίνει'
```

Αξιολόγηση

Η αξιολόγηση γίνεται:

- 50% από ατομικές ασκήσεις στην διάρκεια του εξαμήνου.
- 50% από τελικές εξετάσεις.

Με Λεωνίδα Λιάκος

© Πνευματική ιδιοκτησία 2021.