Εισαγωγή στην Python

Slides taken from Ionian University

[Informatics Department]

S.Sioutas

Μαθαίνοντας Προγραμματισμό με python

- https://sites.google.com/site/pythonlessons/Home

Πλεονεκτήματα της python

- Είναι μια απλή και συμπαγής γλώσσα προγραμματισμού.
- Είναι εύκολη στην εκμάθηση καθώς έχει απλούστατο συντακτικό.
- Ελεύθερο και ανοικτό λογισμικό.
- Ανεξαρτησία λειτουργικού συστήματος, όπως Linux, Windows, FreeBSD, Macintosh, Solaris κ.ά.
- Interpreted (Διερμηνευόμενη), δηλ. εκτελεί το πρόγραμμα απευθείας.
- Εκτεταμένες βιβλιοθήκες. Η Python Standard Library είναι τεράστια.

Εγκατάσταση της python

- Για να δούμε αν είναι εγκατεστημένη η python στο σύστημά μας ανοίγουμε ένα τερματικό ή πηγαίνουμε στην γραμμή εντολών των windows (C:\>) και γράφουμε python. Αν πάρουμε κάτι σαν το παρακάτω :
- *~$ python*
- *Python 2.5.2 (r252:60911, Jul 31 2008, 17:28:52) [GCC 4.2.3 (Ubuntu 4.2.3-2ubuntu7)] on linux2 Type "help", "copyright", "credits" or "license" for more information.*
  >>>
- η python είναι ήδη εγκατεστημένη στο σύστημά μας. Αν όχι πηγαίνουμε στην σελίδα του οργανισμού python http://www.python.org και την εγκαθιστούμε στο σύστημά μας.

Το Πρόγραμμα HelloWorld

- Στην προτροπή (prompt) >>> της pyhton γράφουμε :
- >>> *print('hello world - γεια σου κόσμε')*
- *hello world - γεια σου κόσμε*
- >>>
- Στην προτροπή >>> γράφουμε την εντολή μας και μόλις πατήσουμε enter η εντολή εκτελείται και στην παρακάτω γραμμή εμφανίζεται το αποτέλεσμα.
- Παρακάτω εμφανίζεται πάλι η προτροπή >>> και ο υπολογιστής περιμένει την επόμενη εντολή μας.

Οι Αριθμοί στην python

- Πρόσθεση : 2 + 1
- Αφαίρεση : 5 - 4
- Πολλαπλασιασμός : 7 * 3
- Διαίρεση : 10 / 2
- Υπόλοιπο διαίρεσης : 9 % 2
- Ύψωση σε δύναμη : 2**3
- >>> 7 / 2
- >>> 7 % 2

Η Προτεραιότητα των Πράξεων στην python

- Η python πρώτα εκτελεί τις πράξεις μέσα στις παρενθέσεις και μετά τις υπόλοιπες. Αν υπάρχουν πολλές πράξεις στη σειρά, η python θα τις εκτελέσει με την παρακάτω προτεραιότητα :
- παρενθέσεις ()
- ύψωση σε δύναμη **
- πολλαπλασιασμός *, διαίρεση / και υπόλοιπο διαίρεσης %
- πρόσθεση + και αφαίρεση -
- *>>> 1 + 2 * 3*
  7
- *>>> (1 + 2) * 3*
  9

Οι Μεταβλητές (Variables) – 1/2

- Το όνομα μιας μεταβλητής μπορεί να περιέχει γράμματα του λατινικού αλφάβητου και αριθμούς, πρέπει όμως υποχρεωτικά να αρχίζει από γράμμα :
- *>>> a = 5*
- *>>> b = 4*
- *>>> a + b*
- *9*
- Έχουμε τη δυνατότητα να δώσουμε τιμές σε πολλές μεταβλητές ταυτόχρονα, όπως :
- *>>> a = b = c = 10*
- *>>> a + b + c*
- *30*

Οι Μεταβλητές (Variables) – 2/2

- Τα κεφαλαία γράμματα επιτρέπονται στα ονόματα των μεταβλητών και η python τα ξεχωρίζει από τα μικρά (case sensitive).
- >>> $ax = 5$
- >>> $Ax = 3$
- >>> $ax + Ax$
- 8
- Μία μεταβλητή μπορεί να αποθηκεύσει και γράμματα ή λέξεις ή φράσεις εκτός από αριθμούς.

Οι Συμβολοσειρές (Strings) – 1/3

- String ονομάζουμε μια σειρά χαρακτήρων και ένα string μπορεί να περιβάλλεται από εισαγωγικά, μονά ή διπλά ή και τριπλά!
- *>>> a = 'good'*
- *>>> a*
- *'good'*
- Μπορώ να συνενώσω (concatenate) δύο strings :
- *>>> a = 'Καλή '*
- *>>> b = 'μέρα'*
- *>>> c = a + b*
- *>>> print (c)*
- *Καλή μέρα*

Οι Συμβολοσειρές (Strings) – 2/3

- Μπορώ να πολλαπλασιάσω ένα string με έναν αριθμό :
- *>>> a = ' Ηχώ '*
- *>>> b = a * 5*
- *>>> print (b)*
- *Ηχώ Ηχώ Ηχώ Ηχώ Ηχώ*
- Μπορώ να πάρω (απομονώσω) έναν χαρακτήρα ενός string :
- *>>> a = 'abcdefghi'*
- *>>> a[0]*
- *'a'*
- *>>> a[1]*
- *'b'*
- *>>> a[7]*
- *'h'*

Οι Συμβολοσειρές (Strings) – 3/3

- Μπορώ να πάρω (αποκόψω) ένα τμήμα ενός string :
- *>>> a = 'abcdefghi'*
- *>>> a[:4]*
- *'abcd'*
- *>>> a[4:]*
- *'efghi'*
- *>>> a[4:7]*
- *'efg'*

Σύνθεση Προγράμματος

- Ανοίγουμε έναν editor και γράφουμε το παρακάτω πρόγραμμα :
- *a = 3*
- *b = 7*
- *print (a, b, a**b)*
- Το αποθηκεύουμε με όνομα **first.py**. Στη συνέχεια πηγαίνουμε στο C\: και στον φάκελο του προγράμματος και γράφουμε :
- *python first.py*
- *3 7 2187*

Εντολές Εισόδου και Εξόδου – 1/2

- ***input()***
- *name = input('Δώσε το όνομά σου : ')*
- Η εντολή αυτή σταματάει την εκτέλεση του προγράμματος, αφού εμφανίσει το μήνυμα που υπάρχει μέσα στην παρένθεση, και περιμένει να γράψουμε κάτι και να πατήσουμε enter. Μετά καταχωρεί αυτό που γράψαμε στη μεταβλητή.

Εντολές Εισόδου και Εξόδου – 2/2

- ***print()***
- *print(μεταβλητή ή μήνυμα)*
- *print(a) ή print('Καλημέρα')*
- Η εντολή print εμφανίζει στην οθόνη ένα string.
- *print(a*(b+3*c))*
- *print('Το κεφάλαιο είναι : ' , k , ' και ο τόκος είναι : ' , t)*

Σχόλια (Comments) – 1/2

- Τα **σχόλια** δεν έχουν καμία επίδραση στην εκτέλεση του προγράμματος και σκοπό έχουν να εξηγήσουν και να σχολιάσουν τις εντολές του προγράμματος.
- Είναι απαραίτητα για να μπορεί ένας τρίτος σχετικά εύκολα να καταλάβει το πρόγραμμά μας αλλά κι εμείς οι ίδιοι αν χρειαστεί να το ξαναδούμε μετά από λίγο καιρό.

Σχόλια (Comments) – 2/2

- Τα σχόλια αρχίζουν με το σύμβολο # και συνεχίζουν μέχρι το τέλος της γραμμής.
  #!/usr/local/bin/python
  # -- coding: utf-8 -*-*
- *# Αυτό το πρόγραμμα υπολογίζει το ποσό που θα έχουμε σε έναν χρόνο*
- *# αν κάνουμε σήμερα μια κατάθεση …*

Λίστες (Lists) – 1/9

- Η λίστα ομαδοποιεί άλλα δεδομένα και είναι στην ουσία ένας κατάλογος τιμών που διαχωρίζονται με κόμματα και όλες μαζί βρίσκονται μέσα σε [ ].
- Δεν είναι υποχρεωτικό όλες οι τιμές να είναι του ίδιου τύπου.
- *a = ['london', 'rome', 1452, 9]*

Λίστες (Lists) – 2/9

- Κάθε μέλος μιας λίστας έχει το όνομα της λίστας και την θέση (index) του μέσα σ' αυτήν. Η θέση του πρώτου μέλους μιας λίστας είναι η θέση 0.
- *>>> a[0]*
- *'london'*
- *>>> a[2]*
- *1452*

Λίστες (Lists) – 3/9

- Τα μέλη μιας λίστας μπορεί να είναι διαφορετικού τύπου, η python όμως τα αντιμετωπίζει το καθένα με τον τύπο του.
- Στο προηγούμενο παράδειγμα, η λίστα a έχει τα δύο πρώτα μέλη της strings (london, rome) και τα δύο επόμενα ακέραιους (1452, 9) :
- *>>> a[0] + a[1]*
- *'londonrome'*
- *>>> a[2] + a[3]*
- *1461*

Λίστες (Lists) – 4/9

- Ο αριθμός που προσδιορίζει τη θέση ενός μέλους σε μια λίστα μπορεί να είναι αρνητικός, οπότε ξεκινάμε να υπολογίζουμε τη θέση του μετρώντας από το τέλος προς την αρχή.
- Η θέση του τελευταίου μέλους μπορεί να προσδιοριστεί και με το -1.
- *>>> a[3]*
- *9*
- *>>> a[-1]*
- *9*
- *>>> a[-3]*
- *'rome'*

a = ['london', 'rome', 1452, 9]

Λίστες (Lists) – 5/9

- Μπορούμε να πάρουμε ένα κομμάτι από μια λίστα :
- *>>> a[1:3]*
- *['rome', 1452]*
- Το a[1:3] σημαίνει ότι από την λίστα a παίρνω το μέλος a[1] και όλα τα επόμενα μέχρι το a[3] χωρίς να συμπεριλαμβάνεται το a[3], δηλ. 2 μέλη.

a = ['london', 'rome', 1452, 9]

a = ['london', 'rome', 1452, 9]

Λίστες (Lists) – 6/9

- Μπορώ να έχω και αρνητικούς αριθμούς χωρίς κανένα πρόβλημα :
- *>>> a[1:-1]*
- *['rome', 1452]*
- Αν λείπει ο πρώτος αριθμός σημαίνει ότι εννοώ την αρχή της λίστας, δηλαδή τον αριθμό 0.
- *>>> a[:3]*
- *['london', 'rome', 1452]*

Λίστες (Lists) – 7/9

- Αν λείπει ο δεύτερος αριθμός σημαίνει ότι εννοώ να συμπεριληφθεί στο κομμάτι μου μέχρι και το τελευταίο μέλος της λίστας.
- *>>> a[2:]*
- *[1452, 9]*
- *>>> a[2:4]*
- *[1452, 9]*

a = ['london', 'rome', 1452, 9]

a = ['london', 'rome', 1452, 9]

Λίστες (Lists) – 8/9

- Στο a[2:4] το 4 σημαίνει ότι το τελευταίο μέλος της λίστας που θα έχει το κομμάτι μου θα είναι αυτό με τη θέση 3, δηλ. το τελευταίο.
- Αν λείπουν και οι δύο αριθμοί σημαίνει ότι παίρνω όλα τα μέλη της λίστας στο κομμάτι μου.

Λίστες (Lists) – 9/9

- *>>> a[:]*
- *['london', 'rome', 1452, 9]*
- Το a[:] είναι μια νέα λίστα που τυχαίνει να έχει τα ίδια μέλη με την λίστα a.
- Αυτός είναι ένας τρόπος να δημιουργήσουμε ένα αντίγραφο μιας λίστας.

a = ['london', 'rome', 1452, 9]

Συναρτήσεις για Λίστες – 1/11

- ***append()***

- Η συνάρτηση αυτή προσθέτει ένα νέο μέλος στο τέλος της λίστας :
- *>>> a.append(12)*
- *>>> a*
- *['london', 'rome', 1452, 9, 12]*

Συναρτήσεις για Λίστες – 2/11

- ***insert()***

- Η συνάρτηση αυτή εισάγει ένα νέο μέλος σε μια συγκεκριμένη θέση της λίστας :
- *>>> a.insert(2, 'paris')*
- *>>> a*
- *['london', 'rome', 'paris', 1452, 9, 12]*

['london', 'rome', 1452, 9, 12]

Συναρτήσεις για Λίστες – 3/11

- ***extend()***

- Η συνάρτηση αυτή προσθέτει στο τέλος της λίστας όσα μέλη υπάρχουν μέσα στην παρένθεση :
- *>>> a.extend(['milano', 1812])*
- *>>> a*
- *['london', 'rome', 'paris', 1452, 9, 12, 'milano', 1812]*

['london', 'rome', 'paris', 1452, 9, 12]

Συναρτήσεις για Λίστες – 4/11

- ***remove()***

- Η συνάρτηση αυτή αφαιρεί από μια λίστα την πρώτη εμφάνιση μιας συγκεκριμένης τιμής :
- *>>> a = ['a', 1, 'b', 3, 'd', 1, 'a', 7]*
- *>>> a.remove(1)*
- *>>> a*
- *['a', 'b', 3, 'd', 1, 'a', 7]*

Συναρτήσεις για Λίστες – 5/11

- ***pop()***

- Η συνάρτηση αυτή αφαιρεί από τη λίστα το τελευταίο μέλος της και το επιστρέφει :
- *>>> c = a.pop()*
- *>>> c*
- *7*
- *>>> a*
- *['a', 'b', 3, 'd', 1, 'a']*

a = ['a', 1, 'b', 3, 'd', 1, 'a', 7]

Συναρτήσεις για Λίστες – 6/11

- ***index()***

- *>>> a*
- *['a', 'b', 3, 'd', 1, 'a']*
- *>>> a.index(3)*
- *2*
- Η συνάρτηση index() βρίσκει το πρώτο μέλος της λίστας που είναι ίδιο μ' αυτό που έχουμε βάλει μέσα στην παρένθεση και επιστρέφει τη θέση του μέσα στη λίστα.

Συναρτήσεις για Λίστες – 7/11

- Μπορώ να μάθω τον αριθμό (πλήθος) των μελών μιας λίστας με την ***len()*** :
- *>>> len(a)*
- *6*
- Η len() είναι μια συνάρτηση όλης της python και έχει εφαρμογή σ' όλα τα αντικείμενα της python.
- *>>> k = 'asdfghjkl'*
- *>>> len(k)*
- *9*

Συναρτήσεις για Λίστες – 8/11

- Μπορώ να πάρω το άθροισμα των μελών μιας λίστας (αν είναι αριθμοί) με τη συνάρτηση ***sum()*** :
- *>>> a = [1, 2, 3, 4, 5]*
- *>>> sum(a)*
- *15*

Συναρτήσεις για Λίστες – 9/11

- Η συνάρτηση ***range()*** επιστρέφει μια λίστα αριθμών :
- *>>> range(8)*
- *[0, 1, 2, 3, 4, 5, 6, 7]*
- Ο αριθμός μέσα στην παρένθεση σηματοδοτεί την τελευταία τιμή η οποία δεν συμπεριλαμβάνεται στη λίστα.
- *>>> range(5, 11)*
- *[5, 6, 7, 8, 9, 10]*

Συναρτήσεις για Λίστες – 10/11

- Αν δώσω 2 αριθμούς στην παρένθεση, ο 1ος δηλώνει τον πρώτο αριθμό της λίστας και ο 2ος τον πρώτο αριθμό που δεν θα περιλαμβάνει η λίστα.
- Αν δώσω 3 αριθμούς, οι δύο πρώτοι δηλώνουν την αρχή και το τέλος της λίστας και ο τρίτος την απόσταση (βήμα) ανάμεσα σε δύο αριθμούς της λίστας.
- *>>> range(0, 30, 5)*
- *[0, 5, 10, 15, 20, 25]*

Συναρτήσεις για Λίστες – 11/11

- *>>> range(100, 0, -10)*
- *[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]*
- Όπως φαίνεται από το παραπάνω παράδειγμα, η συνάρτηση range() μπορεί να δουλέψει και κατά φθίνουσα σειρά αν δώσουμε τον τρίτο αριθμό αρνητικό.

Δομές Ελέγχου στην python

- Υπάρχουν δύο είδη δομών ελέγχου, οι ***δομές επανάληψης*** *(loops)* και οι ***δομές επιλογής*** ή ***διακλάδωσης*** *(branches)*.
- Οι δομές επανάληψης επιτρέπουν σε μια ομάδα εντολών να επαναλαμβάνονται ξανά και ξανά, ενώ οι δομές επιλογής επιτρέπουν στον υπολογιστή να επιλέξει ανάμεσα σε δύο ή περισσότερες ομάδες εντολών εξετάζοντας τις συνθήκες που προκύπτουν κατά την εκτέλεση του προγράμματος.

Δομές Επανάληψης – 1/4

- ***Εντολή for***
- Με την εντολή for μπορώ να σαρώσω μια λίστα και να πάρω διαδοχικά ένα-ένα τα μέλη της.
- *>>> li = [1, 3, 5, 7, 9]*
- *>>> for mel in li:print(mel)*
- *1*
- *3*
- *5*
- *7*
- *9*

Δομές Επανάληψης – 2/4

- *>>> for mel in li :print(mel, li.index(mel))*
- *1 0*
- *3 1*
- *5 2*
- *7 3*
- *9 4*

Δομές Επανάληψης – 3/4

- *>>> onoma = ['John', 'Roger', 'Natalie', 'Tamara']*
- *>>> for a in onoma :print(a, len(a))*
- *John 4*
- *Roger 5*
- *Natalie 7*
- *Tamara 6*

Δομές Επανάληψης – 4/4

Εντολή while

```
# break the loop as soon it sees 'e'
# or 's'
i = 0
a = 'geeksforgeeks'

while i < len(a):
  if a[i] == 'e' or a[i] == 's':
    i += 1
    continue

  print('Current Letter :', a[i])
  i += 1
```

```
Current Letter : g
Current Letter : k
Current Letter : f
Current Letter : o
Current Letter : r
Current Letter : g
Current Letter : k
```

Δομές Επιλογής

- ***Εντολή if***
- *x = input("Παρακαλώ δώστε έναν αριθμό : ")*
- *if x < 0 :*
- *print('Αρνητικός αριθμός ')*
- *elif x == 0 :*
- *print('Μηδέν ')*
- *else :*
- *print('Θετικός')*

Συναρτήσεις (Functions) – 1/2

```
def even_or_odd(x) :
if x % 2 == 0 :
    return 'ζυγός'
else:
    return 'μονός'
a = int (input('Δώσε έναν αριθμό : '))
print('Ο αριθμός ' + str( a ) + ' είναι ' + even_or_odd(a))
```

Συναρτήσεις (Functions) – 2/2

- *import math*
- *a = int(input ('Δώσε έναν αριθμό '))*
- *print('1 x 2 x ... x ' + str(a) + ' = ' + str(math.factorial(a)))*
- Στο παραπάνω πρόγραμμα ζητήσαμε από την python να ενσωματώσει τις συναρτήσεις που βρίσκονται στην βιβλιοθήκη της python με όνομα **math**.
- Αυτή, όπως λέει και το όνομά της, περιέχει συναρτήσεις σχετικές με τα μαθηματικά.
- Εμείς χρησιμοποιήσαμε την συνάρτηση **factorial()** που δέχεται έναν ακέραιο και επιστρέφει το γινόμενο 1 x 2 x 3 x ... μέχρι τον αριθμό που δώσαμε (παραγοντικό).