

Εισαγωγή στην Python

Σημειώσεις μαθήματος

Ενότητα 3 – Έλεγχος ροής κώδικα

Δρ.Ν.Μανδέλλος, Δρ.Φ.Δογάνης

Εισαγωγή στην Python

Στόχοι ενότητας:

Λογικές
εκφράσεις

Έλεγχος
της ροής
κώδικα - if

Έλεγχος
ροής
κώδικα for
& while.

Λογικές εκφράσεις στην Python

Operators

Ισότητα ==

Ανισότητα !=

Μεγαλύτερο από >

Μικρότερο από <

Μεγαλύτερο ίσο >=

Μικρότερο ίσο <=

Operators

AND &

OR |

NOT not

```
✓ 0s ▶ x = -2
    if (x > 0.1) & (x < 0.5):
        print("not in 0.1<x<0.5 or x < -3")
    elif not((x < -3) | (x > 3)):
        print("not in ((x < -3) | (x > 3))")
    else:
        print("else")

🔍 not in ((x < -3) | (x > 3))
```

Δομές δεδομένων – Λίστες

Η if μας επιτρέπει να εκτελούμε κώδικα μόνο αν μια συγκεκριμένη συνθήκη είναι αληθής.

Σύνταξη:

if [συνθήκη]:

If block

Εντολές που εκτελούνται αν η συνθήκη είναι αληθής

elif [συνθήκη]:

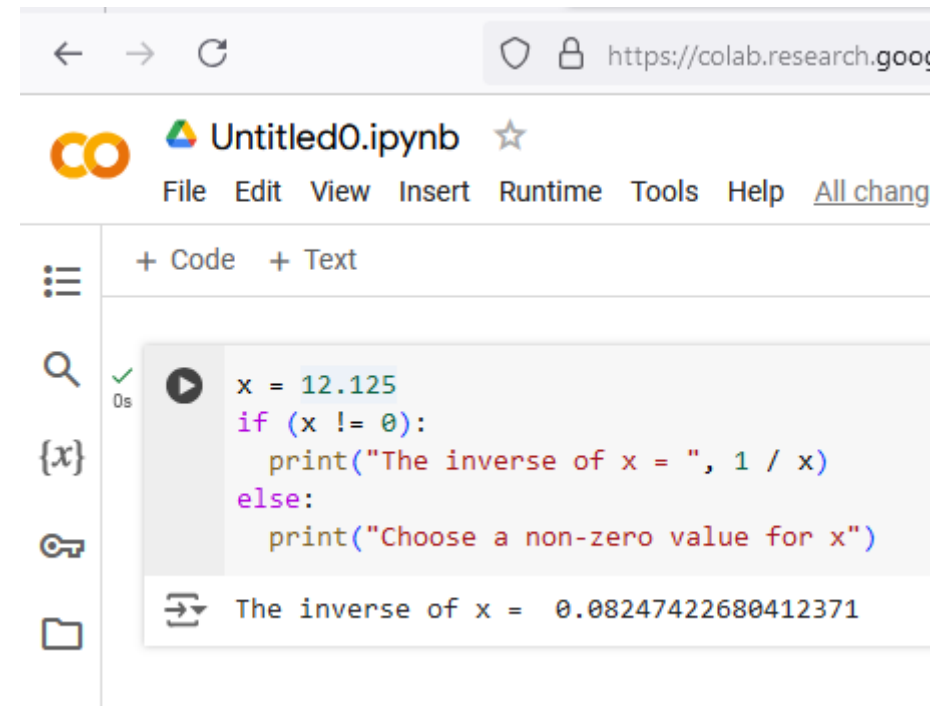
Else If block

Εντολές αν ισχύει η δεύτερη συνθήκη

else:

Else block

Εντολές αν καμία συνθήκη δεν είναι αληθής



The screenshot shows a Google Colab notebook interface. The browser address bar at the top displays 'https://colab.research.google.com'. The notebook title is 'Untitled0.ipynb'. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help', and 'All changes'. The left sidebar contains icons for file explorer, search, variable explorer, and output. The main code area shows a Python cell with the following code:

```
x = 12.125
if (x != 0):
    print("The inverse of x = ", 1 / x)
else:
    print("Choose a non-zero value for x")
```

Below the code cell, the output is displayed: 'The inverse of x = 0.08247422680412371'. The cell status is '0s' and '✓'.

Έλεγχος Ροής με if - Indent

Η σύνταξη της μεθόδου if ορίζει τα 'if' και 'else' blocks, τα οποία δεν οριοθετούνται με κάποιο σημείο 'end' όπως για παράδειγμα στην περίπτωση του Matlab.

Η οριοθέτηση στην Python επιτυγχάνεται με τα indents (διαστήματα) που δημιουργούνται πατώντας το πλήκτρο **TAB**, που προωθούν την επόμενη γραμμή μια θέση δεξιότερα από τη θέση που είναι τοποθετημένο το if.

if (συνθήκη):

→εντολή 1 *# η εντολή αυτή εκτελείται μέσα στο if*

→εντολή 2 *# η εντολή αυτή εκτελείται μέσα στο if*

εντολή 3 *# η εντολή αυτή ΔΕΝ εκτελείται μέσα στο if*

Το σύμβολο → ορίζει το αποτέλεσμα στην οθόνη αφού πατήσουμε το πλήκτρο TAB.

Έλεγχος Ροής με if - Indent

Το παρακάτω παράδειγμα είναι λάθος και μας επιστρέφει σφάλμα:



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: https://colab.research.google.com/drive/1I4IJWbfFn_SIN0uDmDEW4G. The notebook is titled "Untitled0.ipynb". The menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". The left sidebar contains icons for a menu, search, variables, keyboard shortcuts, and a file explorer. The main code area shows the following Python code:

```
x = 12.125
if (x != 0):
    print("The inverse of x = ", 1 / x)
else:
    print("Choose a non-zero value for x")
```

Below the code, an error message is displayed:

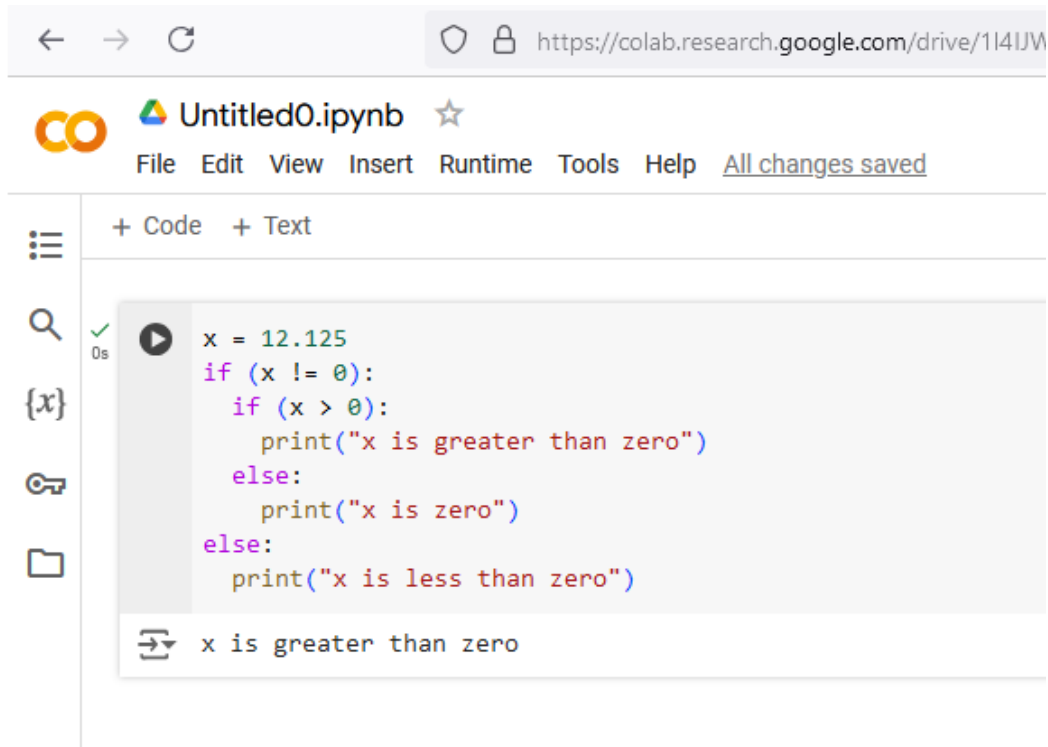
```
File "<ipython-input-7-855096f48312>", line 3
    print("The inverse of x = ", 1 / x)
    ^
IndentationError: expected an indented block after 'if' statement on line 2
```

At the bottom, there is a "Next steps:" section with a button labeled "Explain error".

Έλεγχος Ροής με if – Nested If

Σύνταξη ενός σχήματος nested if:

Προσέξτε τα διαστήματα που καθορίζουν το κάθε ένα 'If block'



The screenshot shows a Google Colab notebook interface. The browser address bar displays <https://colab.research.google.com/drive/114IJW>. The notebook is titled "Untitled0.ipynb" and has a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", "Help", and "All changes saved". On the left sidebar, there are icons for a search, a variable {x}, a key, and a folder. The main code area contains the following Python code:

```
x = 12.125
if (x != 0):
    if (x > 0):
        print("x is greater than zero")
    else:
        print("x is zero")
else:
    print("x is less than zero")
```

Below the code, the output is displayed: `x is greater than zero`.



This screenshot is identical to the one on the left, showing the same code and output. However, the code is formatted with vertical red dashed lines to highlight the indentation levels for each block of the nested if statement, illustrating the structure of the nested if logic.

Άσκηση

Δίνεται η τιμή του pH ενός διαλύματος.

Να γράψετε πρόγραμμα το οποίο να ελέγχει την τιμή του pH και να τυπώνει μήνυμα στον χρήστη ανάλογα με το αν είναι όξινο, βασικό ή ουδέτερο.

Άσκηση

```
# pH classification
pH= 3.5

if pH< 7:
    print("Acidic
          solution")
elif pH==7:
    print("Neutral
          solution")
else:
    print("Basic
          solution")
```

Έλεγχος Ροής με For

Η μέθοδος for χρησιμοποιείται για να επαναλαμβάνουμε μια σειρά εντολών για όσες φορές καθορίζει μια ακολουθία μεταβλητών, την τιμή των οποίων σε κάθε κύκλο παίρνει μια μεταβλητή.

Σύνταξη:

for [μεταβλητή] **in** [ακολουθία]:

for block

Εντολές που εκτελούνται για κάθε στοιχείο της ακολουθίας

Προσοχή: Το for block οριοθετείται με τον ίδιο τρόπο που οριοθετείται και στο if block, δηλαδή με τον ορισμό διαστήματος (αποτέλεσμα στην οθόνη του πλήκτρου TAB).

Άλλοι operators στην Python

Operators

Σύνθετοι τελεστές εκχώρησης

Αθροιστής	<code>+=</code>	Παράδειγμα: $a += 1 \Rightarrow a = a + 1$
Αρνητικός αθροιστής	<code>-=</code>	Παράδειγμα: $a -= 1 \Rightarrow a = a - 1$
Πολλαπλασιαστής	<code>*=</code>	Παράδειγμα: $a *= 5 \Rightarrow a = 5 * a$
Διαιρέτης	<code>/=</code>	Παράδειγμα: $a /= 5 \Rightarrow a = a / 5$

Λογικοί σύνθετοι τελεστές εκχώρησης

AND	<code>&=</code>	Παράδειγμα: $a \&= b \Rightarrow a = a \& b$
OR	<code> =</code>	Παράδειγμα: $a = b \Rightarrow a = a b$

Τελεστής υπολοίπου ακέραιας διαίρεσης

Modulo	<code>%</code>	Παράδειγμα: $5 \% 2 = 1, 8 \% 2 = 0, 8 \% 5 = 3$
--------	----------------	--

Έλεγχος Ροής με For – Αριθμητικές ακολουθίες

Συνάρτηση **range**:

Η συνάρτηση *range(start, stop before, step)* μας επιστρέφει μια ακολουθία αριθμών που καθορίζεται από την τα ορίσματα:

start = πρώτη τιμή της ακολουθίας, τύπου **int**

stop = άνω φράγμα της ακολουθίας, τύπου **int**

step = βήμα, τύπου **int**

```
✓ 0s  for i in range(1, 10):  
    print(i)
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9
```

```
✓ 0s  for i in range(10, 1, -1):  
    print(i)
```

```
10  
9  
8  
7  
6  
5  
4  
3  
2
```

```
✓ 0s  for i in range(1, 10, 3):  
    print(i)
```

```
1  
4  
7
```

```
✓ 0s  for i in range(1, 10, -3):  
    print(i)
```

```
1  
-2  
-5
```

Έλεγχος Ροής με For – Enumerations

Enumerations: Ορίζεται ως ένα σύνολο από μεταβλητές (collection), τα οποία μπορούν να απαριθμηθούν. Παραδείγματα enumerations είναι λίστες (lists), ευρετήρια (dictionaries), πλειάδες (tuples).

Συνάρτηση **enumerate(collection, start=0)**, μας επιστρέφει μια αριθμημένη λίστα από μεταβλητές που ξεκινάει by default από τη θέση 0. Η πρώτη θέση μιας array στην Python είναι η θέση μηδέν.

Μπορούμε να χρησιμοποιήσουμε τη συνάρτηση for για να προσπελάσουμε οποιαδήποτε enumeration.

```
0s ▶ lst = ["eat", "sleep", "repeat"]  
    for counter, item in enumerate(lst):  
        print(counter, item)
```

⇒ 0 eat
1 sleep
2 repeat

```
0s ▶ lst = ["eat", "sleep", "repeat"]  
    for i in lst:  
        print(i)
```

⇒ eat
sleep
repeat

Άσκηση

Δίνονται πολλές τιμές θερμοκρασίας σε βαθμούς Κελσίου. Αποθηκεύστε τις σε κατάλληλο τύπο μεταβλητής. Να γράψετε πρόγραμμα το οποίο να υπολογίζει την αντίστοιχη τιμή θερμοκρασίας σε βαθμούς Kelvin.

.

Άσκηση

```
# convert multiple temperatures
temps_c = [25, 30, 35, 40]

for temp in temps_c:
    kelvin= temp+ 273.15
    print(f"{temp}°C = {kelvin} K")
```

Έλεγχος Ροής με While

Η μέθοδος while χρησιμοποιείται για να επαναλαμβάνουμε μια σειρά εντολών για τόσες φορές όσες ικανοποιείται μια συνθήκη. Το πλήθος των επαναλήψεων δεν είναι γνωστό εκ των προτέρων.

Σύνταξη:

while [συνθήκη] :

while block

Εντολές που εκτελούνται όσο ισχύει η συνθήκη

Προσοχή: Το *while block* οριοθετείται με τον ίδιο τρόπο που οριοθετείται και στο *if block*, δηλαδή με τον ορισμό διαστήματος (αποτέλεσμα στην οθόνη του πλήκτρου TAB).

Έλεγχος Ροής με While

Στα παρακάτω παραδείγματα έχουμε αριστερά έναν counter που αλλάζει τιμή όσο είναι μικρότερος του 5, και δεξιά έχουμε έναν counter που αλλάζει τιμή όσο είναι μικρότερος από 5, όμως έχουμε χρησιμοποιήσει τη συνάρτηση **break** για να διακόψουμε την ροή της while με βάση τη συνθήκη `count > 2`

+ Code + Text

✓
0s

```
count = 0
while count < 5:
    print("Count:", count)
    count += 1
```

```
Count: 0
Count: 1
Count: 2
Count: 3
Count: 4
```

+ Code + Text

✓
0s

```
count = 0
while count < 5:
    print("Count:", count)
    count += 1
    if (count > 2):
        break
```

```
Count: 0
Count: 1
Count: 2
```

Άσκηση

1. Υπολογίστε τον όρο **n** από τη σειρά των φυσικών αριθμών που ισχύει ότι το άθροισμα της σειράς των τετραγώνων μέχρι αυτό το στοιχείο δεν ξεπερνά τον αριθμό **limit**. Να εκτυπώσετε σαν απάντηση ως: «Ο όρος n για τον οποίο το άθροισμα της σειράς των τετραγώνων δεν ξεπερνάει το [limit], είναι ο [number]»
2. Σε μια βιομηχανική εγκατάσταση έχουμε έναν χημικό αντιδραστήρα, μια αποστακτική στήλη, έναν διαχωριστή και ένα ψυκτικό κύκλο. Φτιάξτε ένα λεξικό (dictionary) και καταχωρείστε τα στοιχεία αυτά. Στη συνέχεια ζητήστε από το χρήστη να εισάγει την τιμή κατανάλωσης / παραγωγής ενέργειας ανά ώρα για κάθε στοιχείο (αρνητική τιμή = παραγωγή ενέργειας).
 - a. Να υπολογιστεί το ισοζύγιο κατανάλωσης.
 - b. Να υπολογιστεί η πιο ενεργοβόρα συσκευή.
 - c. Να φτιαχτεί και να εκτυπωθεί μια λίστα με τις συσκευές που παράγουν ενέργεια

ΣΗΜΕΙΩΣΗ: Η εντολή `input` έχει σαν αποτέλεσμα `string`. Για να εισάγουμε αριθμητική τιμή θα πρέπει να κάνουμε 'Type casting', δηλαδή να δηλώσουμε ότι η μεταβλητή είναι τύπου `float`:

Π.χ. `consumption = float(input("Give consumption value"))`

Άσκηση

ΑΣΚΗΣΗ 1

Input

```
sum = 0 # κρατάω το άθροισμα της σειράς στη μεταβλητή αυτή
```

```
n = 0 # ο φυσικός αριθμός για τον οποίο ισχύει η συνθήκη  $\sum i < 200$ 
```

```
answer = input("Ποιόν αριθμός θέλετε να χρησιμοποιήσετε ως όριο;")
```

```
limit = int(answer) #στο python όλα είναι αντικείμενα, άρα μετατρέπουμε την απάντηση σε αριθμό
```

Processing

```
for i in range(1, limit):
```

```
    sum = sum + i*i
```

```
    if sum > limit:
```

```
        n = i - 1 # η προηγούμενη τιμή του i είναι αυτή που ισχύει  $\sum i < 200$ 
```

```
        break
```

Output

```
print(f"Ο όρος n για τον οποίο το άθροισμα της σειράς δεν ξεπερνάει το {limit}, είναι ο {n}")
```

Άσκηση

ΑΣΚΗΣΗ 2

Input

```
elements = {"Χημικός αντιδραστήρας": 0, "Αποστακτική στήλη": 0, "Διαχωριστής": 0, "Ψυκτικό κύκλο": 0}
```

```
for element in elements:
```

```
    consumption = float(input(f"Give consumption for {element}: \n"))
```

```
    elements[element] = consumption
```

Processing – Ερώτημα 1

```
sum = 0
```

```
for element in elements:
```

```
    sum = sum + elements[element]
```

```
print("Συνολική κατανάλωση: %5.0f" % sum)
```

Processing – Ερώτημα 2

```
maxValue = 0;
```

```
maxElement = "unknown"
```

```
for element in elements:
```

```
    if (maxValue < elements[element]):
```

```
        maxValue = elements[element]
```

```
        maxElement = element
```

```
print(f"Η πιο ενεργοβόρα συσκευή είναι: {maxElement} επειδή έχει τιμή {elements[maxElement]} \n");
```

Processing – Ερώτημα 3

```
generatorDevices = []
```

```
for element in elements:
```

```
    if (elements[element] < 0):
```

```
        generatorDevices.append(element)
```

```
print("Συσκευές που παράγουν ενέργεια: ", generatorDevices);
```

Άσκηση

Για τα τμήματα A1, A2 & A3 μιας σχολής έχουμε την παρακάτω καρτέλα βαθμολογίας μορφοποιημένη σε πλειάδες:

```
student1 = ("AK", "A1", 7)
student2 = ("MN", "A2", 8)
student3 = ("TS", "A2", 9.5)
student4 = ("AP", "A1", 10)
student5 = ("PT", "A1", 5)
student6 = ("KE", "A3", 2)
student7 = ("OS", "A3", 3)
student8 = ("SS", "A2", 8)
student9 = ("AF", "A2", 5)
student10 = ("FX", "A1", 5.8)
student11 = ("KO", "A1", 6.5)
student12 = ("LP", "A2", 4)
student13 = ("TT", "A3", 2)
student14 = ("PP", "A1", 6)
student15 = ("SA", "A1", 6)
```

Ζητείται:

1. Να υπολογιστούν πόσα άτομα βρίσκονται σε κάθε τμήμα
2. Να υπολογιστεί ο μέσος όρος βαθμολογίας
3. Να υπολογιστεί ο μέσος όρος βαθμολογίας ανά τμήμα

Άσκηση

ΛΥΣΗ

INPUT DATA

```
student1 = ("AK", "A1", 7)
student2 = ("MN", "A2", 8)
student3 = ("TS", "A2", 9.5)
student4 = ("AP", "A1", 10)
student5 = ("PT", "A1", 5)
student6 = ("KE", "A3", 2)
student7 = ("OS", "A3", 3)
student8 = ("SS", "A2", 8)
student9 = ("AF", "A2", 5)
student10 = ("FX", "A1", 5.8)
student11 = ("KO", "A1", 6.5)
student12 = ("LP", "A2", 4)
student13 = ("TT", "A3", 2)
student14 = ("PP", "A1", 6)
student15 = ("SA", "A1", 6)
```

Φτιάχνω μια λίστα με τους φοιτητές

```
list = []
list.append(student1)
list.append(student2)
list.append(student3)
list.append(student4)
list.append(student5)
```

```
list.append(student6)
list.append(student7)
list.append(student8)
list.append(student9)
list.append(student10)
list.append(student11)
list.append(student12)
list.append(student13)
list.append(student14)
list.append(student15)
```

αρχικοποιώ το dictionary

```
dic = {}
```

προσθέτω τις πλειάδες σε λίστα ανά τμήμα

```
for i in range(0, len(list)):
    dept = list[i][1]
    if not(dept in dic):
        dic[dept] = [] # φτιάξε ένα νέο κλειδί με κενή λίστα
                        # επειδή δεν υπάρχει το τμήμα
```

```
dic[dept].append(list[i]) # αν το τμήμα του φοιτητή υπάρχει στο
                           # dic τότε πρόσθεσε τον στη λίστα
```

PROCESSING

Πλήθος φοιτητών ανά τμήμα

```
for key in dic:
    print(f'Φοιτητές στο {key} είναι: {len(dic[key])}')

# Υπολογισμός μέσου όρου
```

Υπολογισμός μέσου όρου

```
grandTotal = 0
totalLength = 0
for key in dic:
    sum = 0
    length = len(dic[key])
    for i in range(0, length):
        listItem = dic[key][i]
        grade = float(listItem[2])
        sum += grade
    grandTotal += sum
    totalLength += length
    average = sum / length
# OUTPUT
print(f'Μέσος όρος για το {key}: %.3f' % average)
```

```
average = grandTotal / totalLength
```

OUTPUT

```
print(f'Συνολικός Μέσος όρος: %.3f' % average)
```