

Εισαγωγή στην Python

Σημειώσεις μαθήματος

ΜΑΘΗΜΑ 3

Δρ.Ν.Μανδέλλος, Δρ.Φ.Δογάνης

Μάθημα 3 – Έλεγχος ροής κώδικα

Στόχοι Μαθήματος:

- 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	<=

in statement: χρησιμοποιείται για έλεγχο εαν η τιμή μιας μεταβλητής εμπεριέχεται σε μια συλλογή (λίστα, πλειάδα, λεξικό). (*if x in list: ..*)

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))

Έλεγχος Ροής με if

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] :

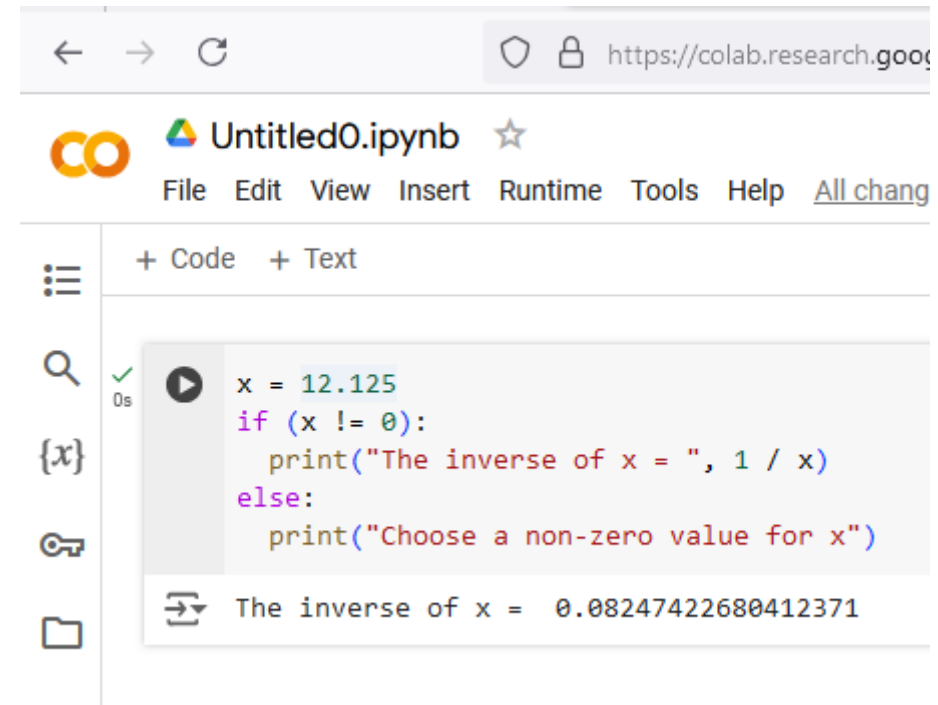
Else If block

Εντολές αν ισχύει η δεύτερη συνθήκη

else:

Else block

Εντολές αν καμία συνθήκη δεν είναι αληθής



The screenshot shows a web browser window with the URL <https://colab.research.google.com>. The notebook is titled "Untitled0.ipynb". The code cell contains 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")
```

The output of the code cell is:

```
The inverse of x = 0.08247422680412371
```

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 γυρίζει σφάλμα:



The screenshot shows a web browser window with the URL `https://colab.research.google.com/drive/1I4IJWbfFn_SIN0uDmDEW4G`. The page title is "Untitled0.ipynb". The notebook interface shows a code cell with 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")
```

The code is highlighted in a light gray box. Below the code, an error message is displayed:

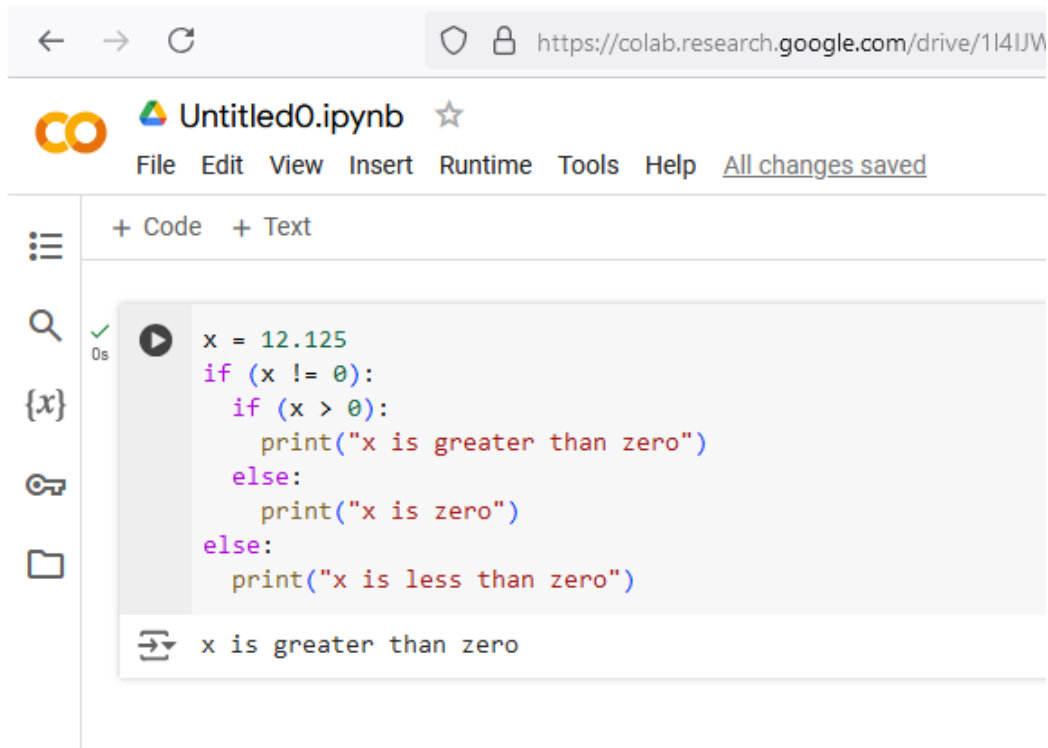
```
File "<ipython-input-7-855096f48312>", line 3
    print("The inverse of x = ", 1 / x)
    ^
IndentationError: expected an indented block after 'if' statement on line 2
```

At the bottom of the error message, 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'



```
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")
```

x is greater than zero



```
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")
```

x is greater than zero

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).

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)
```

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

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 μέχρι αυτό το στοιχείο δεν ξεπερνά το 200. Να εκτυπώσετε σαν απάντηση ως: «Ο όρος n για τον οποίο το άθροισμα της σειράς δεν ξεπερνάει το 200, είναι ο [ΑΡΙΘΜΟΣ]»
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 1

```
sum = 0 # κρατάω το άθροισμα της σειράς στη μεταβλητή αυτή
n = 0 # ο φυσικός αριθμός για τον οποίο ισχύει η συνθήκη  $\Sigma i < 200$ 
for i in range(1, 200):
    sum = sum + i
    if sum > 200:
        n = i - 1 # η προηγούμενη τιμή του  $i$  είναι αυτή που ισχύει  $\Sigma i < 200$ 
        break

# εκτύπωση του αποτελέσματος
print(f"Ο όρος  $n$  για τον οποίο το άθροισμα της σειράς δεν ξεπερνάει το 200, είναι ο  $\{n\}$ ")
```

Άσκηση 2

✓
27s

```
elements = {"Χημικός αντιδραστήρας": 0, "Αποστακτική στήλη": 0, "Διαχωριστής": 0, "Ψυκτικό κύκλο": 0}
for element in elements:
    consumption = float(input("Give consumption for " + element + ": "))
    elements[element] = consumption

# Ερώτημα 1
sum = 0;
for element in elements:
    sum = sum + elements[element]

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

# Ερώτημα 2
maxValue = 0;
for element in elements:
    if (maxValue < elements[element]):
        maxValue = elements[element]

print("Μέγιστη τιμή: %5.0f" % maxValue)

# Ερώτημα 3
generatorDevices = []
for element in elements:
    if (elements[element] < 0):
        generatorDevices.append(element)

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

```
⇒ Give consumption for Χημικός αντιδραστήρας: -5
Give consumption for Αποστακτική στήλη: 2
Give consumption for Διαχωριστής: -0.2
Give consumption for Ψυκτικό κύκλο: 3
Συνολική κατανάλωση: -0
Μέγιστη τιμή: 3
Συσκευές που παράγουν ενέργεια: ['Χημικός αντιδραστήρας', 'Διαχωριστής']
```