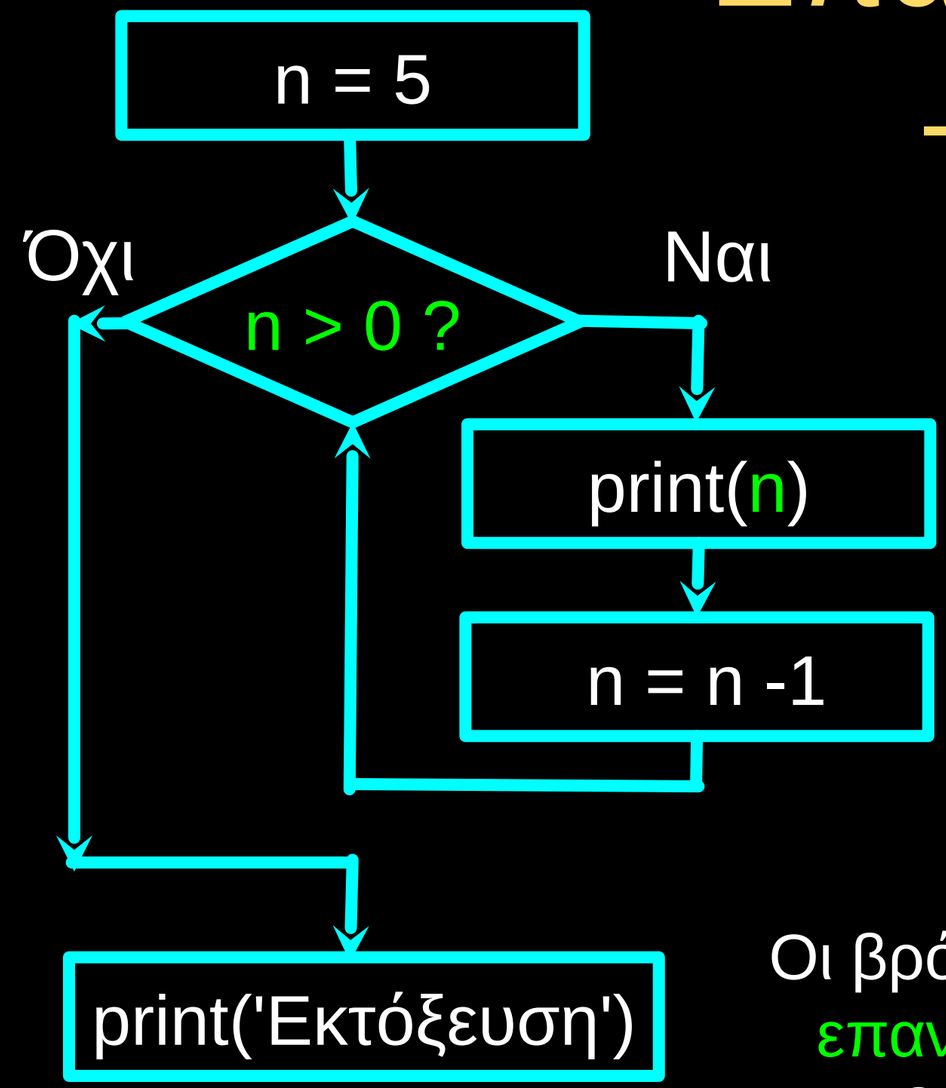
Βρόχοι και Επανάληψη

Κεφάλαιο 5

Python για Όλους

www.py4e.com

Επαναλαμβανόμενα βήματα – Δομή Επανάληψης



Program:

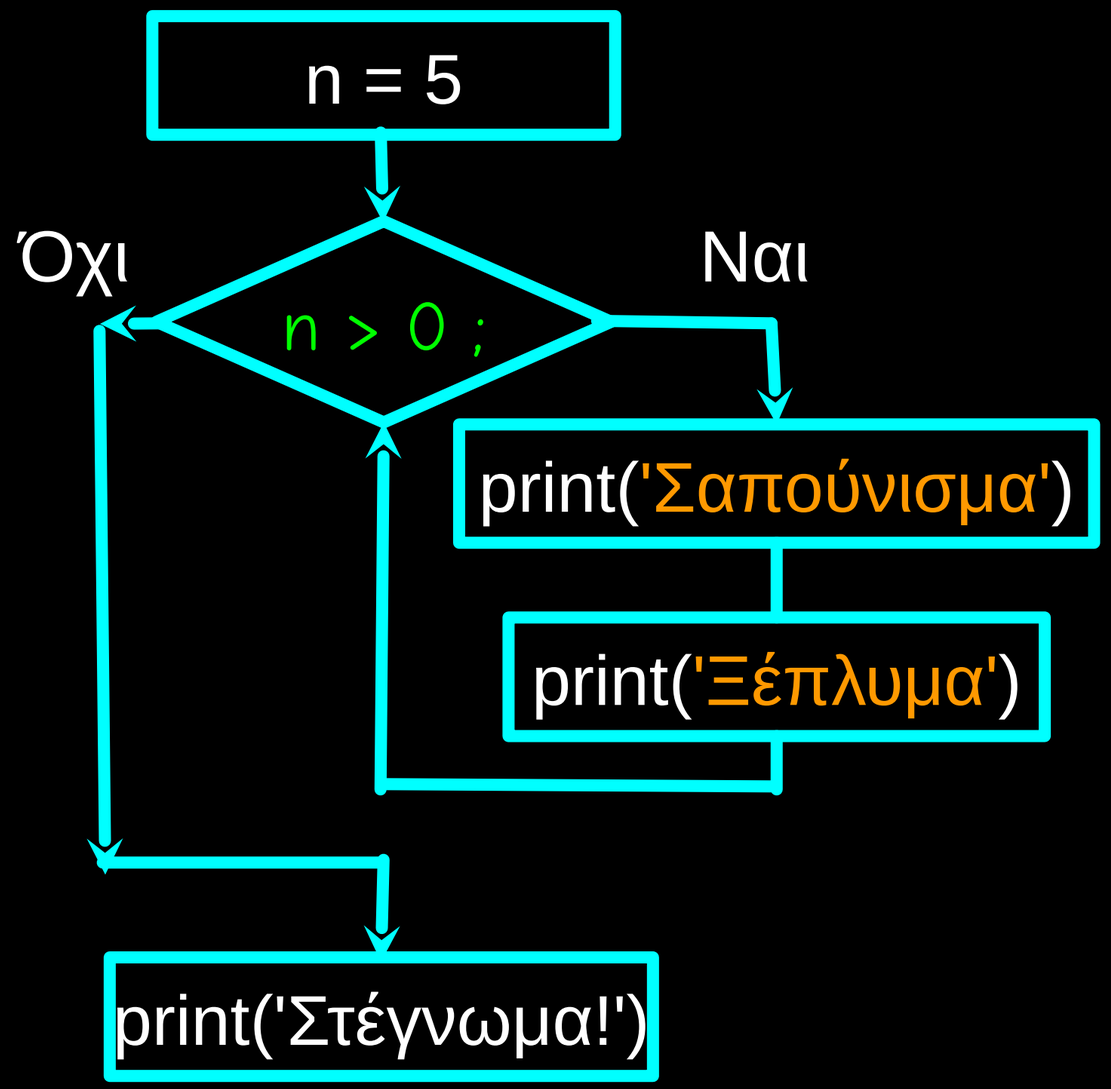
```
n = 5
while n > 0 :
    print(n)
    n = n - 1
print('Εκτόξευση!')
print(n)
```

Output:

```
5
4
3
2
1
Εκτόξευση!
0
```

Οι βρόχοι (επαναλαμβανόμενα βήματα) έχουν μεταβλητές επανάληψης που αλλάζουν κάθε φορά που εκτελείτε ο βρόχος. Συχνά αυτές οι μεταβλητές επανάληψης διατρέχουν μια ακολουθία αριθμών

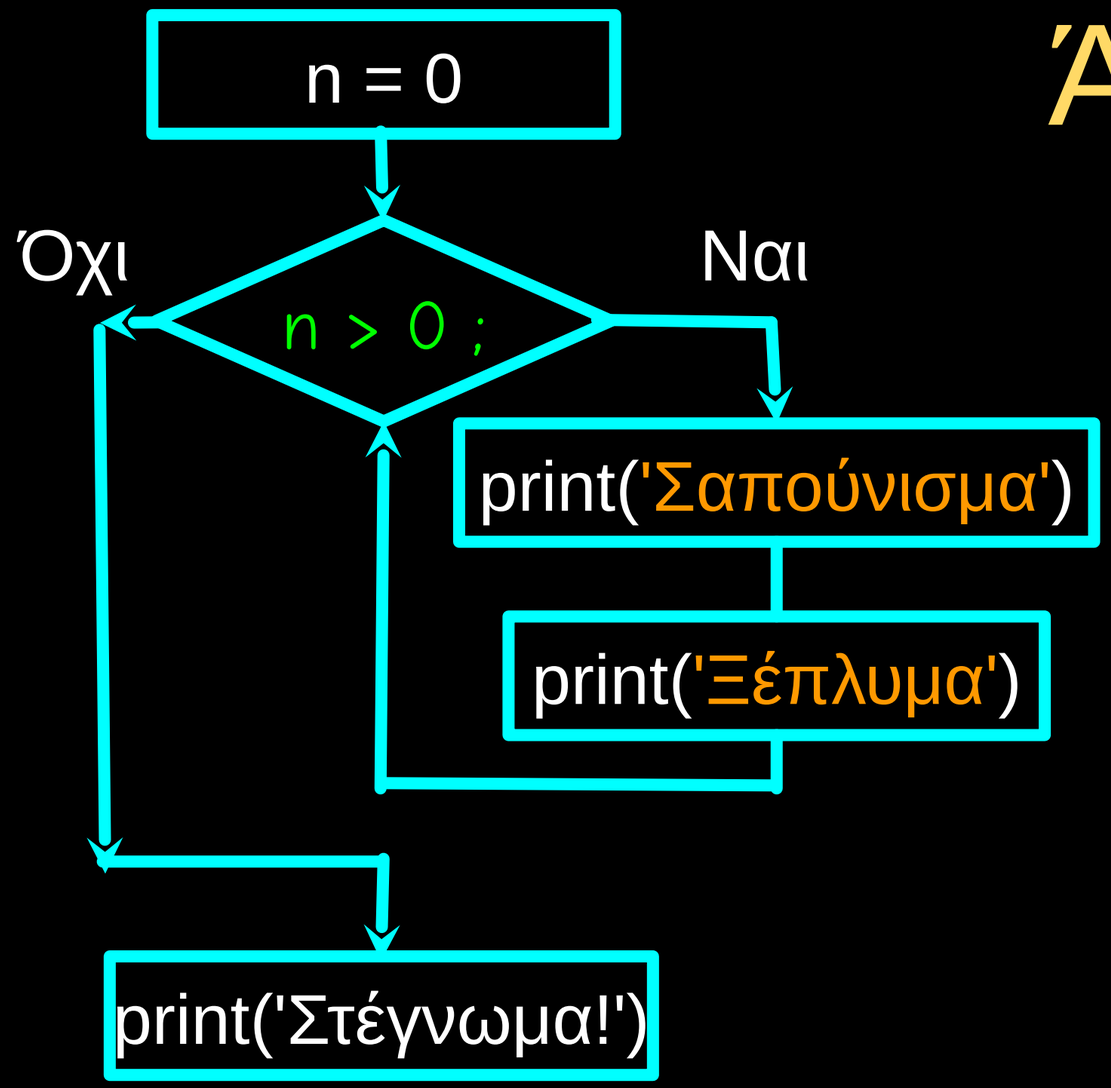
Ένας Ατέρμων Βρόχος




```
n = 5
while n > 0 :
    print('Σαπούνισμα')
    print('Ξέπλυμα')
print('Στέγνωμα!')
```

Τι λάθος υπάρχει σε αυτό το βρόχο;

Άλλος ένας Βρόχος




```
n = 0
while n > 0 :
    print('Σαπούνισμα')
    print('Ξέπλυμα')
print('Στέγνωμα!')
```

Τι κάνει αυτός ο βρόχος;

Διαφυγή από το Βρόχο

- Η δήλωση break διακόπτει την εκτέλεση του τρέχοντα βρόχο και μεταβαίνει στην πρώτη εντολή αμέσως μετά τον βρόχο
- Είναι σαν μια δοκιμή βρόχου που μπορεί να τοποθετηθεί οπουδήποτε στο σώμα του βρόχου

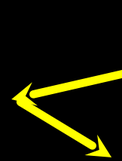
```
while True:
    γραμμή = input('> ')
    if γραμμή == 'τέλος' :
        break
    print(γραμμή)
print('Τέλος!')
```

```
> Γειά σας
Γειά σας
> τελείωσε
τελείωσε
> τέλος
Τέλος!
```

Διαφυγή από το Βρόχο

- Η δήλωση break διακόπτει την εκτέλεση του τρέχοντα βρόχο και μεταβαίνει στην πρώτη εντολή αμέσως μετά τον βρόχο
- Είναι σαν μια δοκιμή βρόχου που μπορεί να τοποθετηθεί οπουδήποτε στο σώμα του βρόχου

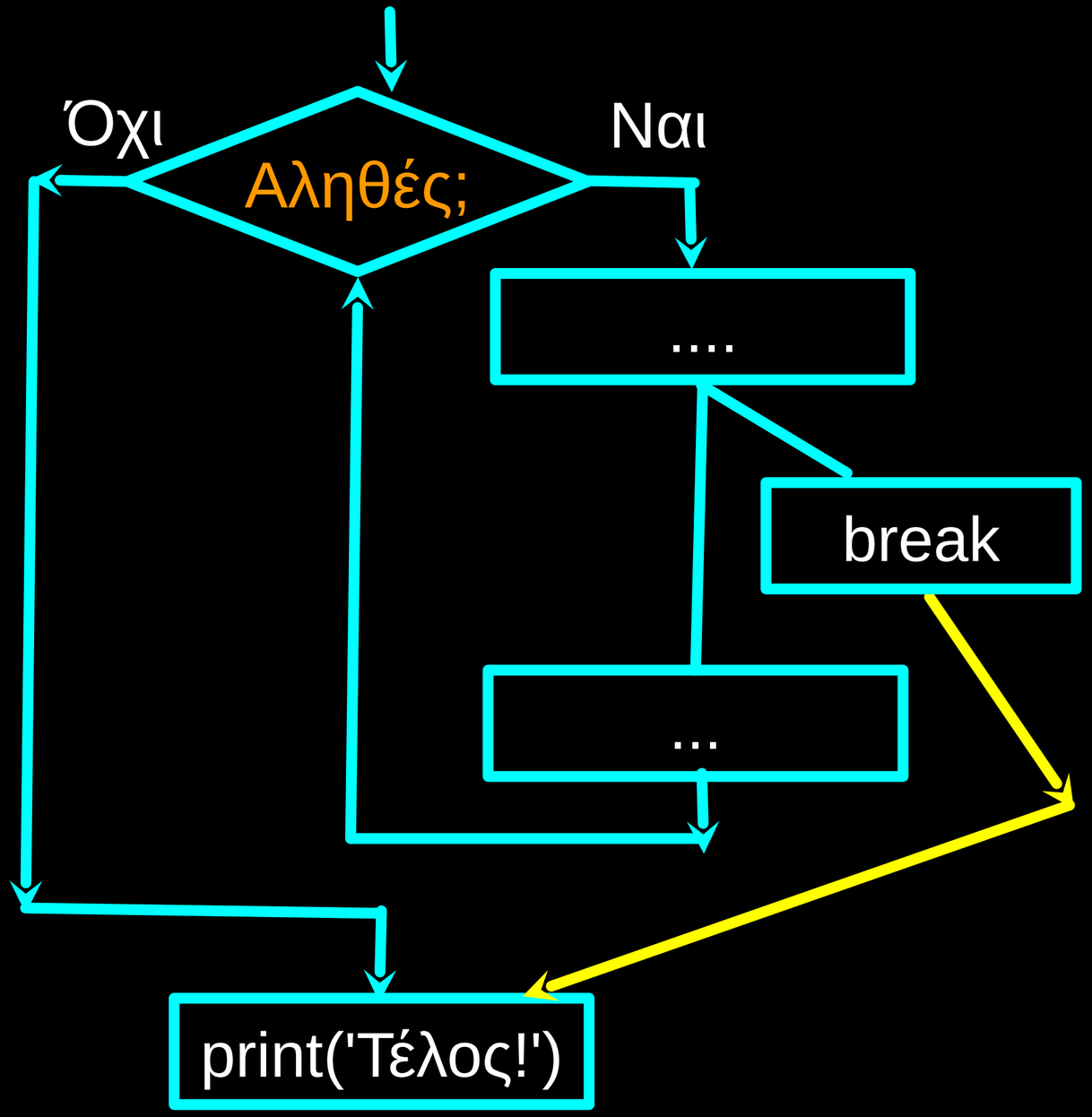
```
while True:
    γραμμή = input('> ')
    if γραμμή == 'τέλος' :
        break
    print(γραμμή)
print('Τέλος!')
```



```
> Γειά σας
Γειά σας
> τελείωσε
τελείωσε
> τέλος
Τέλος!
```

```
while True:
    γραμμή = input('> ')
    if γραμμή == 'τέλος' :
        break
    print(γραμμή)
print('Τέλος!')
```

http://en.wikipedia.org/wiki/Transporter_(Star_Trek)



Ολοκληρώνοντας μια Επανάληψη με continue

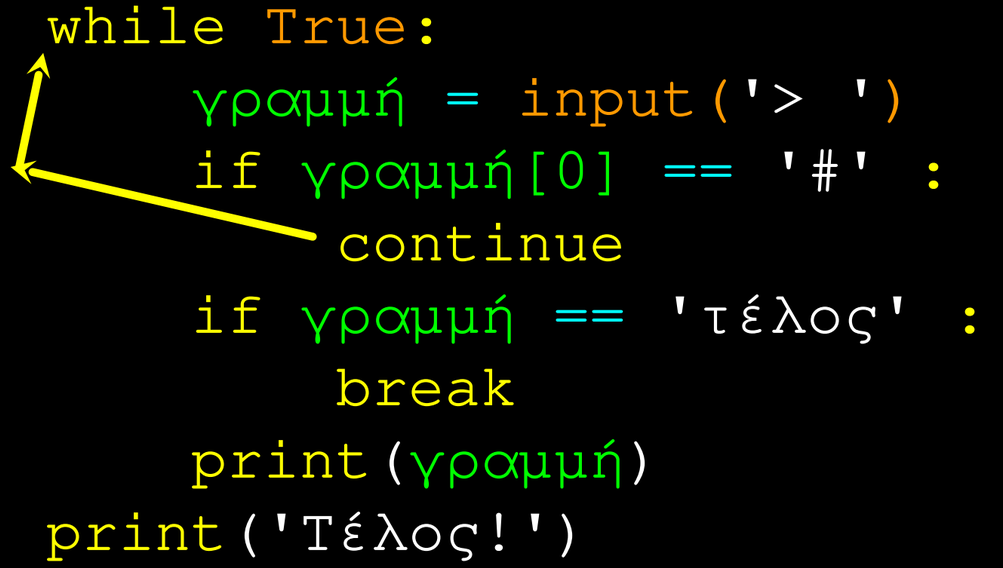
Η δήλωση continue τερματίζει την τρέχουσα επανάληψη, μεταβαίνει στην κορυφή του βρόχου και ξεκινά την επόμενη επανάληψη

```
while True:
    γραμμή = input('> ')
    if γραμμή[0] == '#' :
        continue
    if γραμμή == 'τέλος' :
        break
    print(γραμμή)
print('Τέλος!')
```

```
> Γειά σας
Γειά σας
> # μην το εκτυπώσεις αυτό
> εκτύπωσε αυτό!
εκτύπωσε αυτό!
> τέλος
Τέλος!
```

Ολοκληρώνοντας μια Επανάληψη με continue

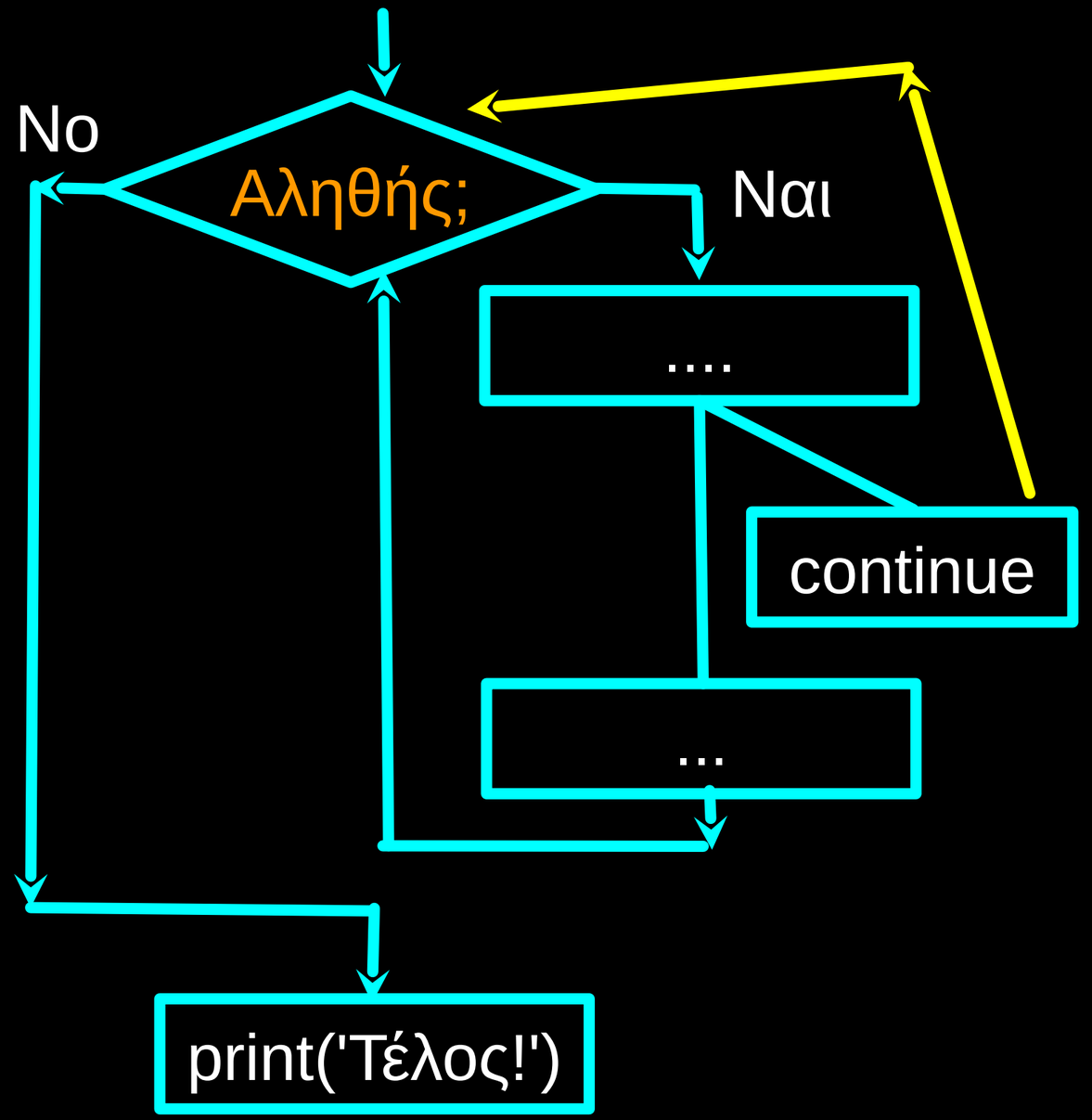
Η δήλωση continue τερματίζει την τρέχουσα επανάληψη, μεταβαίνει στην κορυφή του βρόχου και ξεκινά την επόμενη επανάληψη



```
while True:
    γραμμή = input('> ')
    if γραμμή[0] == '#' :
        continue
    if γραμμή == 'τέλος' :
        break
    print(γραμμή)
print('Τέλος!')
```

> Γειά σας
Γειά σας
> # μην το εκτυπώσεις αυτό
> εκτύπωσε αυτό!
εκτύπωσε αυτό!
> τέλος
Τέλος!

```
while True:
    γραμμή = raw_input('> ')
    if γραμμή[0] == '#' :
        continue
    if γραμμή == 'τέλος' :
        break
    print(γραμμή)
print('Τέλος!')
```



Ατέρμονες Βρόχοι

- Οι βρόχοι while ονομάζονται «ατέρμονες βρόχοι» επειδή συνεχίζουν να εκτελούνται μέχρι μια λογική συνθήκη να γίνει False/Ψευδής
- Οι βρόχοι που έχουμε δει μέχρι τώρα είναι αρκετά εύκολο να εξεταστούν για να δούμε αν θα τερματιστούν ή αν θα είναι «ατέρμονες βρόχοι»
- Κάποιες φορές είναι λίγο πιο δύσκολο να βεβαιωθούμε εάν ένας βρόχος θα τερματιστεί

Καθορισμένοι Βρόχοι

Επανάληψη σε ένα σύνολο στοιχείων…

Καθορισμένοι Βρόχοι

- Πολύ συχνά έχουμε μια λίστα με στοιχεία τις γραμμές ενός αρχείου - ουσιαστικά ένα πεπερασμένο σύνολο πραγμάτων
- Μπορούμε να γράψουμε έναν βρόχο που θα εκτελείτε μία φορά για καθένα από τα στοιχεία ενός συνόλου χρησιμοποιώντας την εντολή for της Python
- Αυτοί οι βρόχοι ονομάζονται «καθορισμένοι βρόχοι» επειδή εκτελούν έναν ακριβή αριθμό φορών
- Λέμε ότι οι «καθορισμένοι βρόχοι επαναλαμβάνονται επί των μελών ενός συνόλου»

Ένας Απλός Καθορισμένος Βρόχος

```
for i in [5, 4, 3, 2, 1] :
    print(i)
print('Εκτόξευση!')
```

```
5
4
3
2
1
Εκτόξευση!
```

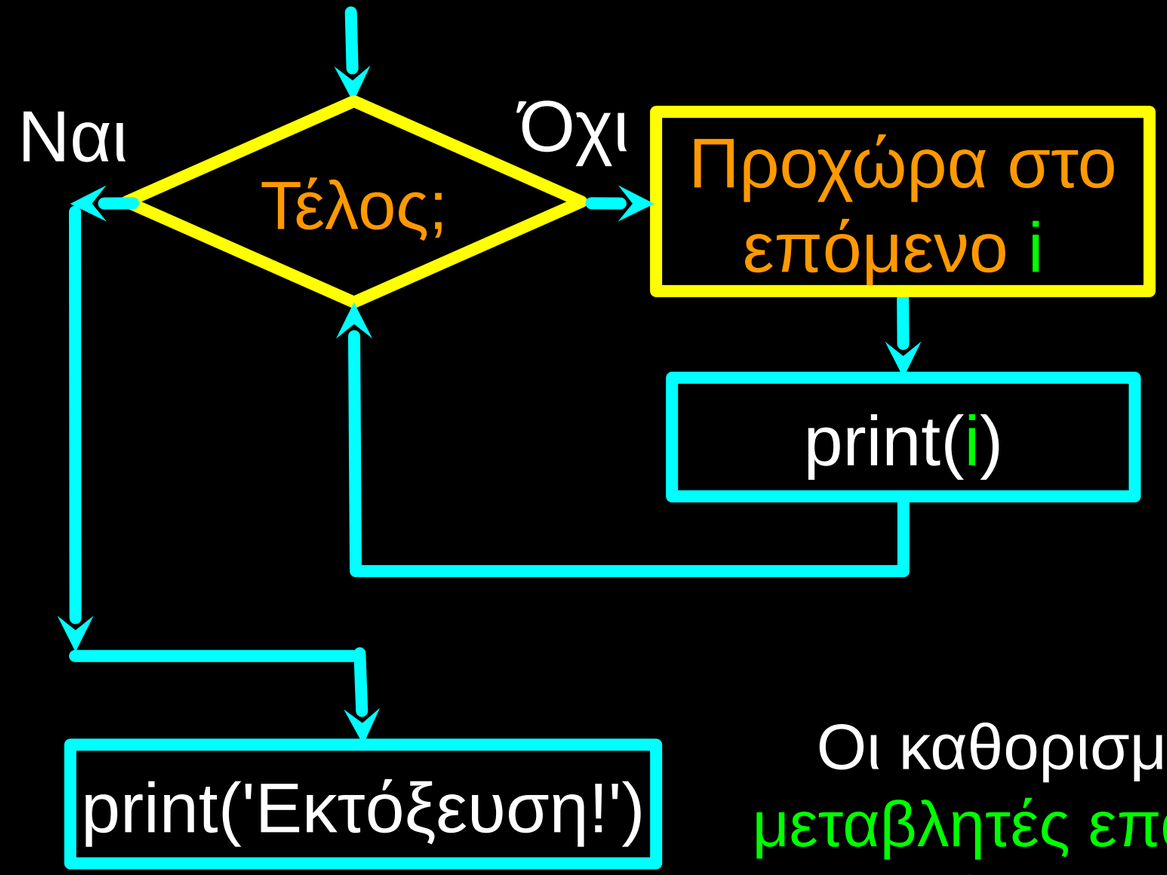
Ένας Καθορισμένος Βρόχος με Συμβολοσειρές

```
φίλοι = ['Δημήτρης', 'Σοφία', 'Άρης']
for φίλος in φίλοι :
    print('Καλή Χρονιά:', φίλος)
print('Τέλος!')
```

Καλή Χρονιά: Δημήτρης
Καλή Χρονιά : Σοφία
Καλή Χρονιά : Άρης

Τέλος!

Ένας Απλός Καθορισμένος Βρόχος



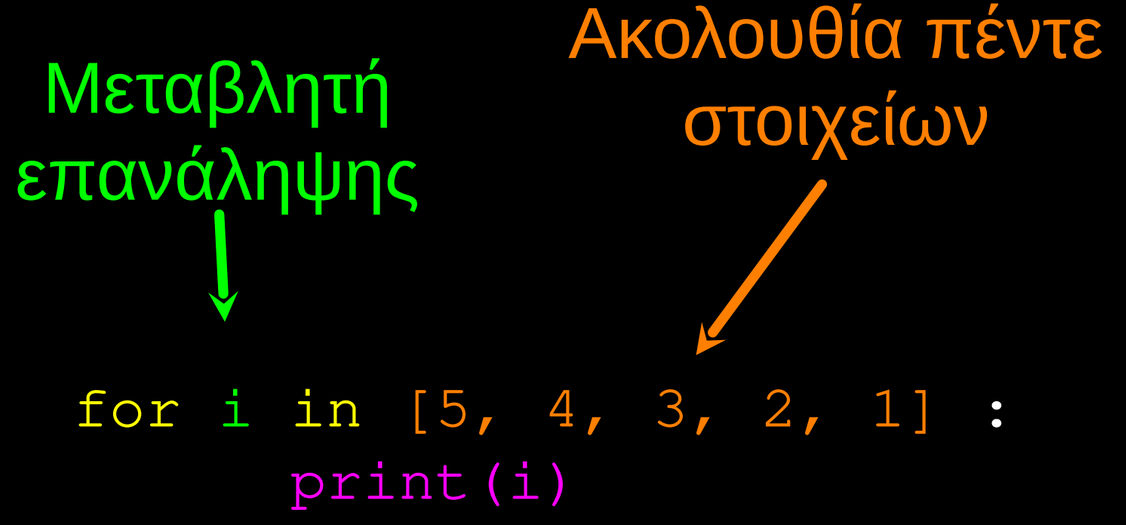

```
for i in [5, 4, 3, 2, 1] :
    print(i)
print('Εκτόξευση!')
```

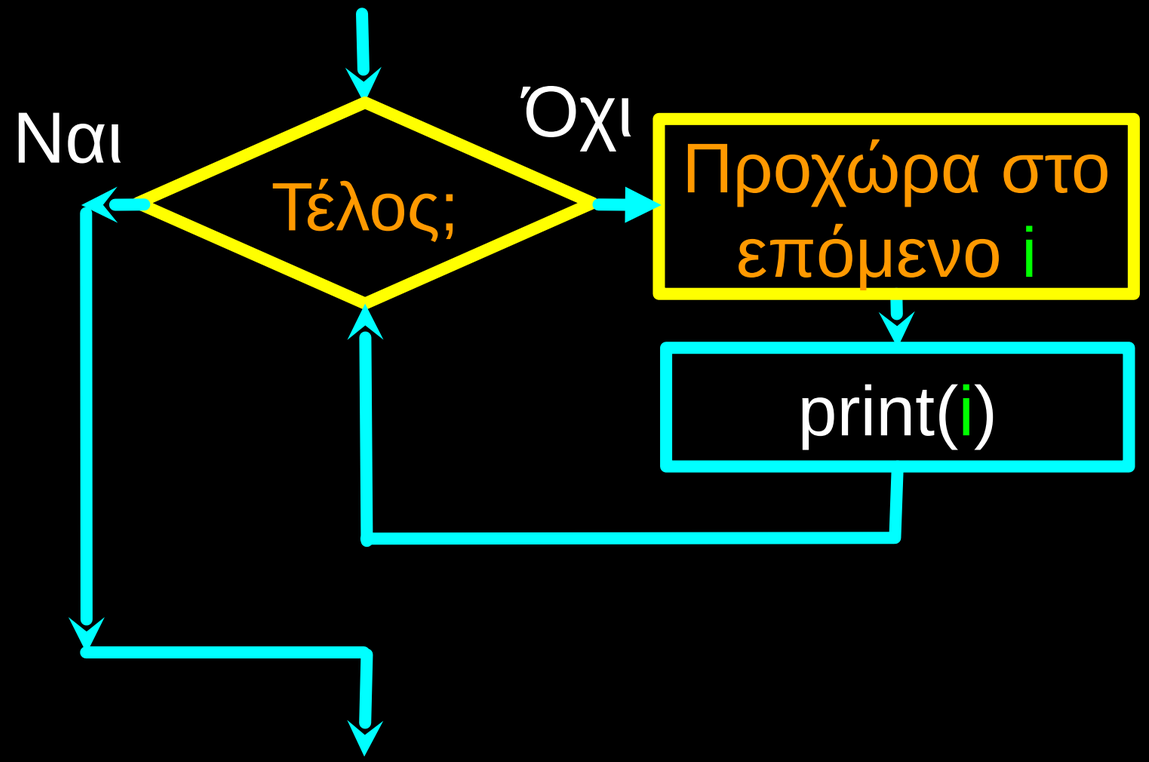
```
5
4
3
2
1
Εκτόξευση!
```

Οι καθορισμένοι βρόχοι (βρόχος for) χρησιμοποιούν μεταβλητές επανάληψης που αλλάζουν κάθε φορά μέσω του βρόχου. Αυτές οι μεταβλητές επανάληψης παίρνουν τιμές από την ακολουθία ή το σύνολο που θα ορίσουμε

Ας δούμε το in...

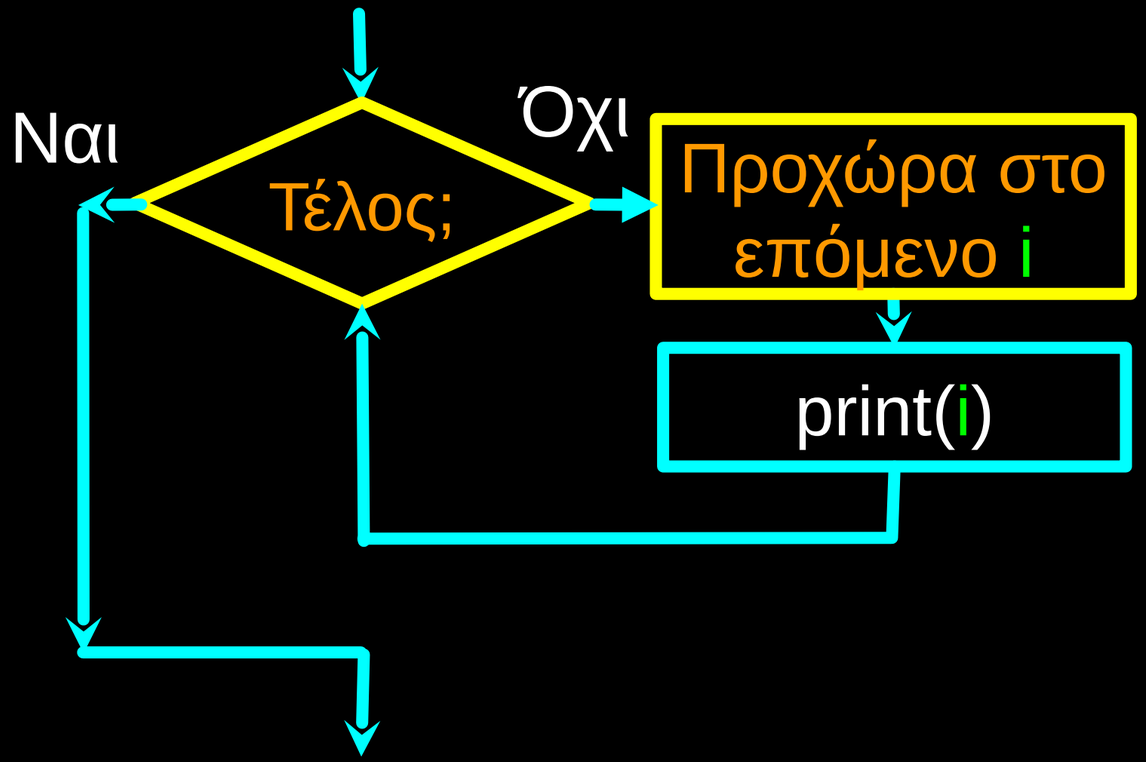
- Η μεταβλητή επανάληψης "επαναλαμβάνεται" επί της ακολουθίας (ταξινομημένο σύνολο)
- Το μπλοκ (σώμα) του κώδικα εκτελείται μία φορά για κάθε τιμή in (μέσα) στην ακολουθία
- Η μεταβλητή επανάληψης διατρέχει όλες τις τιμές in (μέσα) στην ακολουθία



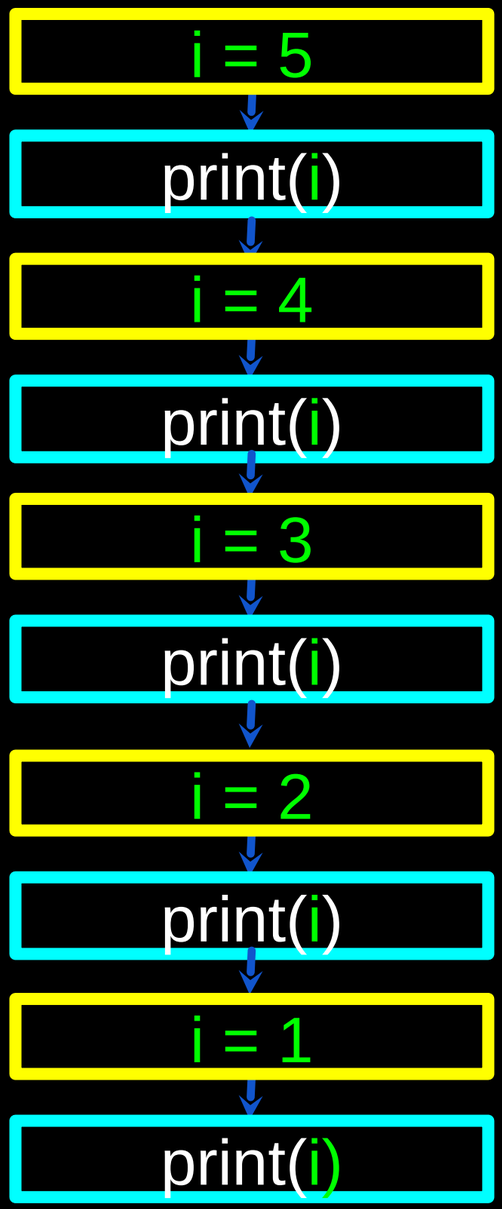



```
for i in [5, 4, 3, 2, 1] :
    print(i)
```

- Η μεταβλητή επανάληψης "επαναλαμβάνεται" επί της ακολουθίας (ταξινομημένο σύνολο)
- Το μπλοκ (σώμα) του κώδικα εκτελείται μία φορά για κάθε τιμή in (μέσα) στην ακολουθία
- Η μεταβλητή επανάληψης διατρέχει όλες τις τιμές in (μέσα) στην ακολουθία




```
for i in [5, 4, 3, 2, 1] :
    print(i)
```



Ιδιωματισμοί Βρόχου:
Τι Κάνουμε σε Βρόχους

Σημείωση: Παρόλο που αυτά τα παραδείγματα είναι απλά, τα μοτίβα ισχύουν για όλα τα είδη βρόχων

Κατασκευή «έξυπνων» βρόχων

Το κόλπο είναι να «γνωρίζετε» κάτι για ολόκληρο τον βρόχο όταν είστε κολλημένοι γράφοντας κώδικα που βλέπει μόνο μία καταχώριση τη φορά

Εκχώρηση αρχικών τιμών στις μεταβλητές

for κάτι in δεδομένα:

Αναζητήστε κάτι ή κάντε κάτι σε κάθε καταχώριση ξεχωριστά, ενημερώνοντας μια μεταβλητή

Προβάλετε τις μεταβλητές

Βρόχος που Διατρέχει Σύνολο

```
print('Πριν')
for κάτι in [9, 41, 12, 3, 74, 15] :
     print(κάτι)
print('Μετά')
```

```
$ python basicloop.py
Πριν
9
41
12
3
74
15
Μετά
```

Ποιος είναι ο Μεγαλύτερος Αριθμός;

Ποιος είναι ο Μεγαλύτερος Αριθμός;

3

Ποιος είναι ο Μεγαλύτερος Αριθμός;

41

Ποιος είναι ο Μεγαλύτερος Αριθμός;

12

Ποιος είναι ο Μεγαλύτερος Αριθμός;

9

Ποιος είναι ο Μεγαλύτερος Αριθμός;

74

Ποιος είναι ο Μεγαλύτερος Αριθμός;

15

Ποιος είναι ο Μεγαλύτερος Αριθμός;

Ποιος είναι ο Μεγαλύτερος Αριθμός;

3 41 12 9 74 15

Ποιος είναι ο Μεγαλύτερος Αριθμός;

μεγαλύτερος_μέχρι_στιγμής | -1

Ποιος είναι ο Μεγαλύτερος Αριθμός;

3

μεγαλύτερος_μέχρι_στιγμής 3

Ποιος είναι ο Μεγαλύτερος Αριθμός;

41

μεγαλύτερος_μέχρι_στιγμής | 41

Ποιος είναι ο Μεγαλύτερος Αριθμός;

12

μεγαλύτερος_μέχρι_στιγμής 41

Ποιος είναι ο Μεγαλύτερος Αριθμός;

9

μεγαλύτερος_μέχρι_στιγμής 41

Ποιος είναι ο Μεγαλύτερος Αριθμός;

74

μεγαλύτερος_μέχρι_στιγμής | 74

Ποιος είναι ο Μεγαλύτερος Αριθμός;

15

74

Ποιος είναι ο Μεγαλύτερος Αριθμός;

3 41 12 9 74 15

74

Εντοπίζοντας την Μεγαλύτερη Τιμή

```
μεγαλύτερος_μέχρι_στιγμής = -1
print('Πριν', μεγαλύτερος_μέχρι_στιγμής)
for αριθμό in [9, 41, 12, 3, 74, 15] :
    if αριθμό > μεγαλύτερος_μέχρι_στιγμής :
        μεγαλύτερος_μέχρι_στιγμής = αριθμό
    print(μεγαλύτερος_μέχρι_στιγμής, αριθμό)

print('Μετά', μεγαλύτερος_μέχρι_στιγμής)
```

```
$ python largest.py
Πριν -1
9 9
41 41
41 12
41 3
74 74
74 15
Μετά 74
```

Δημιουργούμε μια μεταβλητή που περιέχει τη μεγαλύτερη τιμή που έχουμε δει μέχρι τώρα. Εάν ο τρέχων αριθμός που εξετάζουμε είναι μεγαλύτερος, τότε αυτός είναι η νέα μεγαλύτερη τιμή που έχουμε δει μέχρι τώρα.

Περισσότερα Πρότυπα Βρόχων...

Καταμέτρηση σε έναν Βρόχο

```
zork = 0
print('Πριν', zork)
for thing in [9, 41, 12, 3, 74, 15] :
    zork = zork + 1
    print(zork, thing)
print('Μετά', zork)
```

```
$ python countloop.py
Πριν 0
1 9
2 41
3 12
4 3
5 74
6 15
Μετά 6
```

Για να μετρήσουμε πόσες φορές εκτελούμε έναν βρόχο, εισάγουμε μια μεταβλητή μετρητή που ξεκινά από το 0 και προσθέτουμε ένα σε αυτήν κάθε φορά που εκτελείτε ο βρόχος.

Αθροίζοντας σε έναν Βρόχο

```
zork = 0
print('Πριν', zork)
for thing in [9, 41, 12, 3, 74, 15] :
    zork = zork + thing
    print(zork, thing)
print('Μετά', zork)
```

```
$ python countloop.py
Πριν 0
9 9
50 41
62 12
65 3
139 74
154 15
Μετά 154
```

Για να προσθέσουμε μια τιμή που συναντάμε σε έναν βρόχο, εισάγουμε μια μεταβλητή αθροίσματος που ξεκινά από το 0 και προσθέτουμε την τιμή στη μεταβλητή αθροίσματος κάθε φορά μέσω του βρόχου.

Εύρεση Μέσου Όρου σε έναν Βρόχο

```
πλήθος = 0
άθροισμα = 0
print('Πριν', πλήθος, άθροισμα)
for τιμή in [9, 41, 12, 3, 74, 15] :
    πλήθος = πλήθος + 1
    άθροισμα = άθροισμα + τιμή
    print(πλήθος, άθροισμα, τιμή)
print('Μετά', πλήθος, άθροισμα, άθροισμα / πλήθος)
```

```
$ python averageloop.py
Πριν 0 0
1 9 9
2 50 41
3 62 12
4 65 3
5 139 74
6 154 15
Μετά 6 154 25.666
```

Ο μέσος όρος απλά συνδυάζει τα πρότυπα καταμέτρησης και αθροίσματος και διαιρεί όταν τελειώσει ο βρόχος.

Έλεγχος σε έναν Βρόχο

```
print('Πριν')
for τιμή in [9, 41, 12, 3, 74, 15] :
    if τιμή > 20:
        print('Μεγάλος αριθμός', τιμή)
print('Μετά')
```

```
$ python search1.py
Πριν
Μεγάλος αριθμός 41
Μεγάλος αριθμός 74
Μετά
```

Χρησιμοποιούμε μια εντολή if στον βρόχο για να εντοπίσουμε / φιλτράρουμε τις τιμές που ψάχνουμε.

Αναζήτηση με Χρήση Λογικής Μεταβλητής

```
βρέθηκε = False
print('Πριν', βρέθηκε)
for τιμή in [9, 41, 12, 3, 74, 15] :
    if τιμή == 3 :
        βρέθηκε = True
    print(βρέθηκε, τιμή)
print('Μετά', βρέθηκε)
```

```
$ python search1.py
Πριν False
False 9
False 41
False 12
True 3
True 74
True 15
Μετά True
```

Εάν θέλουμε απλώς να αναζητήσουμε και να γνωρίζουμε αν βρέθηκε μια τιμή, χρησιμοποιούμε μια μεταβλητή που αρχικοποιείται με False και παίρνει την τιμή True μόλις βρούμε αυτό που ψάχνουμε.

Πώς να Εντοπίσουμε την Μικρότερη Τιμή

```
μεγαλύτερος_μέχρι_στιγμής = -1
print('Πριν', μεγαλύτερος_μέχρι_στιγμής )
for τιμή in [9, 41, 12, 3, 74, 15] :
    if τιμή > μεγαλύτερος_μέχρι_στιγμής :
        μεγαλύτερος_μέχρι_στιγμής = τιμή
    print(μεγαλύτερος_μέχρι_στιγμής , τιμή)

print('Μετά', μεγαλύτερος_μέχρι_στιγμής )
```

```
$ python largest.py
Πριν -1
9  9
41  41
41 12
41  3
74  74
74  15
Μετά 74
```

Πώς μπορούμε να το αλλάξουμε για να βρει τη μικρότερη τιμή στη λίστα;

Εντοπίζοντας την Μικρότερη Τιμή

```
μικρότερος_μέχρι_στιγμή = -1
print('Πριν', μικρότερος_μέχρι_στιγμή)
for τιμή in [9, 41, 12, 3, 74, 15] :
    if τιμή < μικρότερος_μέχρι_στιγμή :
        μικρότερος_μέχρι_στιγμή = τιμή
    print(μικρότερος_μέχρι_στιγμή, τιμή)

print('Μετά', μικρότερος_μέχρι_στιγμή)
```

Αλλάξαμε το όνομα της μεταβλητής σε smallest_so_far και αλλάξαμε το > σε <

Εντοπίζοντας την Μικρότερη Τιμή

```
μικρότερος_μέχρι_στιγμή = -1
print('Πριν', μικρότερος_μέχρι_στιγμή)
for τιμή in [9, 41, 12, 3, 74, 15] :
    if τιμή < μικρότερος_μέχρι_στιγμή :
        μικρότερος_μέχρι_στιγμή = τιμή
    print(μικρότερος_μέχρι_στιγμή, τιμή)

print('Μετά', μικρότερος_μέχρι_στιγμή)
```

```
$ python smallbad.py
Πριν -1
-1 9
-1 41
-1 12
-1 3
-1 74
-1 15
Μετά -1
```

Αλλάξαμε το όνομα της μεταβλητής σε `μικρότερος_μέχρι_στιγμή` και αλλάξαμε το > σε <

Εντοπίζοντας την Μικρότερη Τιμή

```
μικρότερος = None
print('Πριν')
for τιμή in [9, 41, 12, 3, 74, 15] :
    if μικρότερος is None :
        μικρότερος = τιμή
    elif value < smallest :
        μικρότερος = τιμή
    print(μικρότερος, τιμή)
print('Μετά', μικρότερος)
```

```
$ python smallest.py
Πριν
9 9
9 41
9 12
3 3
3 74
3 15
Μετά 3
```

Έχουμε και πάλι μια μεταβλητή που η τιμή της είναι η μικρότερη μέχρι τώρα. Την πρώτη φορά που διατρέχουμε τον βρόχο η μικρότερη έχει τιμή None (Καμία), οπότε παίρνουμε την πρώτη τιμή ως τη μικρότερη.

Οι τελεστές is και is not

```
μικρότερος = None
print('Πριν')
for τιμή in [3, 41, 12, 9, 74, 15] :
    if μικρότερος is None :
        μικρότερος = τιμή
    elif τιμή < μικρότερος :
        μικρότερος = τιμή
    print(μικρότερος, τιμή)

print('Μετά', μικρότερος)
```

- Η Python έχει τον τελεστή is που μπορεί να χρησιμοποιηθεί σε λογικές εκφράσεις
- Παρόμοιο, αλλά ισχυρότερο από το ==
- is not είναι επίσης ένας λογικός τελεστής

Σύνοψη

- Βρόχοι while (αόριστοι)
- Ατέρμονες Βρόχοι
- Χρήση του break
- Χρήση του continue
- Σταθερές και Μεταβλητές None
- Βρόχοι For (καθορισμένοι)
- Μεταβλητές επανάληψης
- Ιδιωματισμοί Βρόχου
- Μεγαλύτερο ή Μικρότερο

Ευχαριστίες / Συνεισφορές

Αρχική ανάπτυξη : Charles Severance, University of Michigan School of Information

Απόδοση στα Ελληνικά: Κιουρτίδου Δ. Κωνσταντία

… Εισαγάγετε νέους Μεταφραστές και άτομα που έχουν συνεισφέρει εδώ

Συνέχεια…