Έλεγχος ροής - if

Σύνταξη

`if` (λογική συνθήκη):
    εντολή 1
    εντολή 2

    Το τέλος της `if` δηλώνετε με το τέλος της στοίχισης των εντολών

if - elif - else

Σύνταξη

```
if (λογική συνθήκη):
    εντολή 1
    εντολή 2
elif (λογική συνθήκη):
    εντολή 3
    εντολή 4
else:
  εντολή 3
  εντολή 4
```

Επαναλήψεις - Loops

while

Σύνταξη

`while` (λογική συνθήκη)**:**
εντολή 1
εντολή 2

Το τέλος της **`while`** δηλώνετε με το τέλος της στοίχισης των εντολών

while

Πίνακας θερμοκρασιών

```
print '----------------------'
C=-20
dC=5
while C<=40:
    F=(9.0/5)*C+32
    print C,F
    C=C+dC
print '----------------------'
```

Σύντομες εντολές

C=C+dC μπορεί να γίνει C+=dC

```
print '--------------------'
C=-20
dC=5
while C<=40:
    F=(9.0/5)*C+32
    print C,F
    C+=dC
print '--------------------'
```

ομοίως

- C-=dC # C=C-dC
- C*=dC # C=C*dC
- C/=dC # C=C/dC

Ανισότητα - Ισότητα

- `c==40` (ισότητα)
- `c!=40` ή `c<>0` (διάφορο)
- `c>=40` (μεγαλύτερο ή ίσο)
- `c>40` (γνήσια μεγαλύτερο)
- `not c==40` είναι ίδιο με το `c!=40`

Παραδείγματα με λογικές εκφράσεις

```
x=0;y=1.2;z=-1
x>0 and z>2 or y>0 ⟶ True (πρώτα το and και μετά το or)
x>0 and (z>2 or y>0) ⟶ False (πρώτα η πράξη της παρένθεσης)
not x>0 or z>-2 and y>0 ⟶ True
not (x>0 or z>-2) and y>0 ⟶ False
```

format

```
print '----------------------'
C=35
dC=1./7
while C<=40:
    F=(9.0/5)*C+32
    print 'C=%.8f,F=%.8f'%(C,F)
    C+=dC
print '----------------------'
```

(8 δεκαδικά)

```
print 'C={cel:.4f},F={far:.4e}'.format(cel=C,far=F)
```

(μέθοδος format)

format

`%s`	string
`%d`	integer
`%10.6f`	float (10 θέσεις με 6 δεκαδικά ψηφία)
`%g`	generic number
`%e ή %E`	Επιστημονική μορφή

Άθροισμα σειράς

$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Άθροισμα

$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

```
x = 1.2 # assign some value
N = 25 # maximum power in sum
k=1
s=x
sign = 1.0
import math
while k < N:
    sign = - sign
    k=k+2
    term = sign*x**k/math.factorial(k)
    s = s + term
print 'sin(%g) = %g (approximation with %d terms)' % (x, s, N)
```

For

Σύνταξη

```
for (μετρητής) in (ακολουθία):
  εντολή 1
  εντολή 2
```

Το τέλος της **for** δηλώνετε με το τέλος της στοίχισης των εντολών

for

- Εντολή `in:` Έλεγχος εγκλεισμού
- `1 in [1,2,3]` ⟶ `True`
- `4 in [1,2,3]` ⟶ `False`
- `4 not in [1,2,3]` ⟶ `True`
- `‘a’ in ‘abc’` ⟶ `True`

for - break

```
for x in range(20):
    if (x%2==0):
        print x
        if x==16:
            print ‘break’
            break
```

⟶ stop εκτέλεσης **for**

for - continue

```
for x in range(20):
    if (x%2==0):
        print '%d is even'%x
        continue
    print x
```

→ συνέχιση εκτέλεσης αύξηση μετρητή