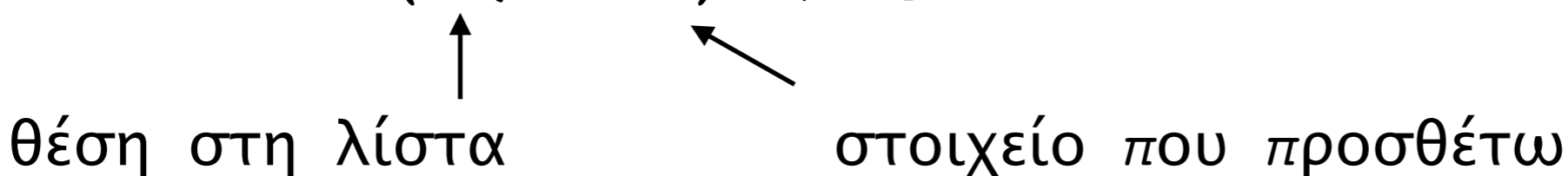


Λίστες

Λίστες

- `C=[-10,-5,0,10]`
- `C.append(15)` (`C=[-10,-5,0,10,15]`)
- `C=C+[20,25]` (`C=[-10,-5,0,10,15,20,25]`)
- `C.insert(0,-15)` (`C=[-15,-10,-5,0,10,15,20,25]`)


θέση στη λίστα στοιχείο που προσθέτω

Λίστες

- `del C[2]` (Διαγραφή 3ου στοιχείου)
`C=[-15,-10,0,10,15,20,25]`
- `len(C)` (Μήκος και για άλλες ακολουθιακές δομές)
- `10 in C` (εγκλεισμός)
- `C.pop()` (επιστρέφει το τελευταίο στοιχείο και το αφαιρεί)

Λίστες

- `C=range(-10,30,5)` (λίστα με βήμα 5)
- Η `range` μπορεί να κάνει ακέραιο βήμα

```
c=[]  
c_val=10  
c_max=12  
while c_val<=c_max  
    c.append(c_val)  
    c_val+=.2
```

Λίστες

- Θέλουμε να μεταβάλουμε τα στοιχεία μιας δοσμένης λίστας
- $C=[0,5,10,15]$
- Θέλουμε να γίνει $[2,7,12,17]$

```
for i in C:  
    i+=2
```

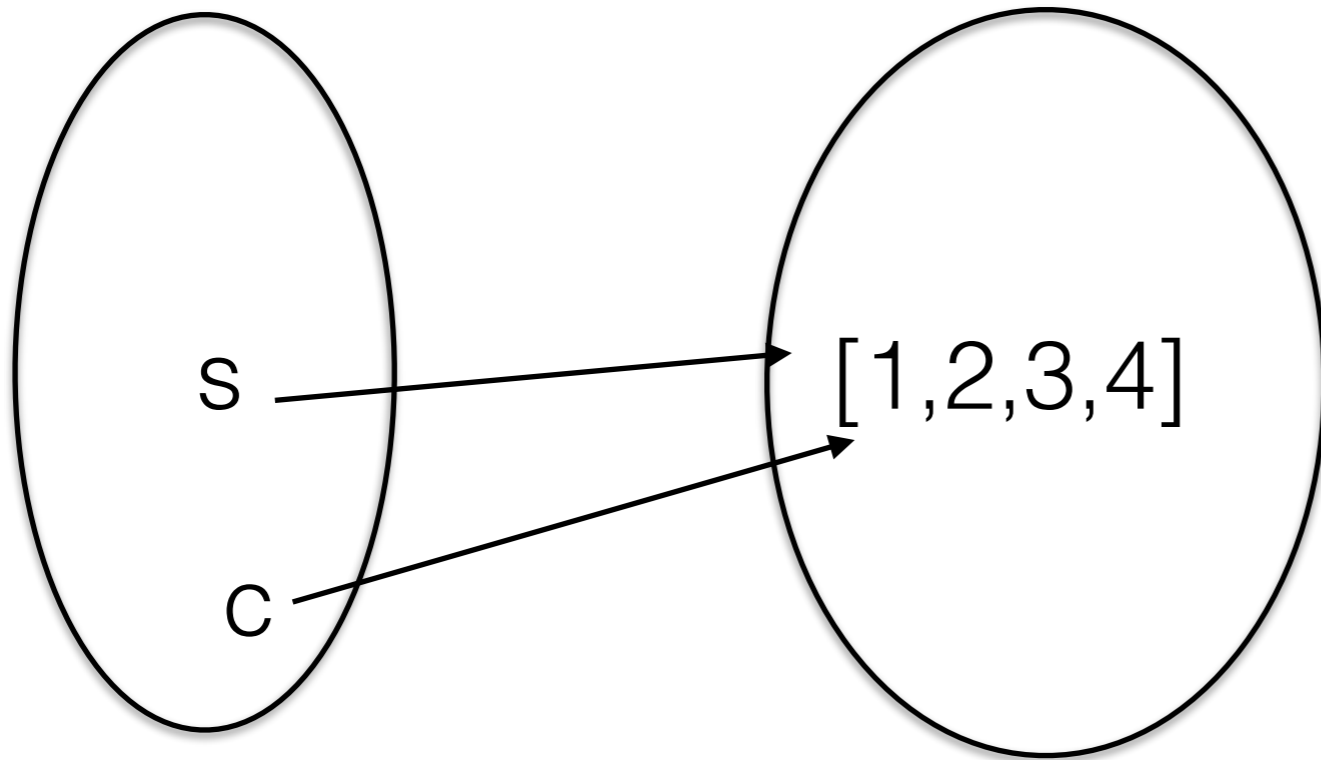
- Λάθος αντιμετώπιση

Λίστες

```
m=len(C)
for i in range(m):
    c[i]+=2
```

Αλλάζει τη λίστα με τον
επιθυμητό τρόπο

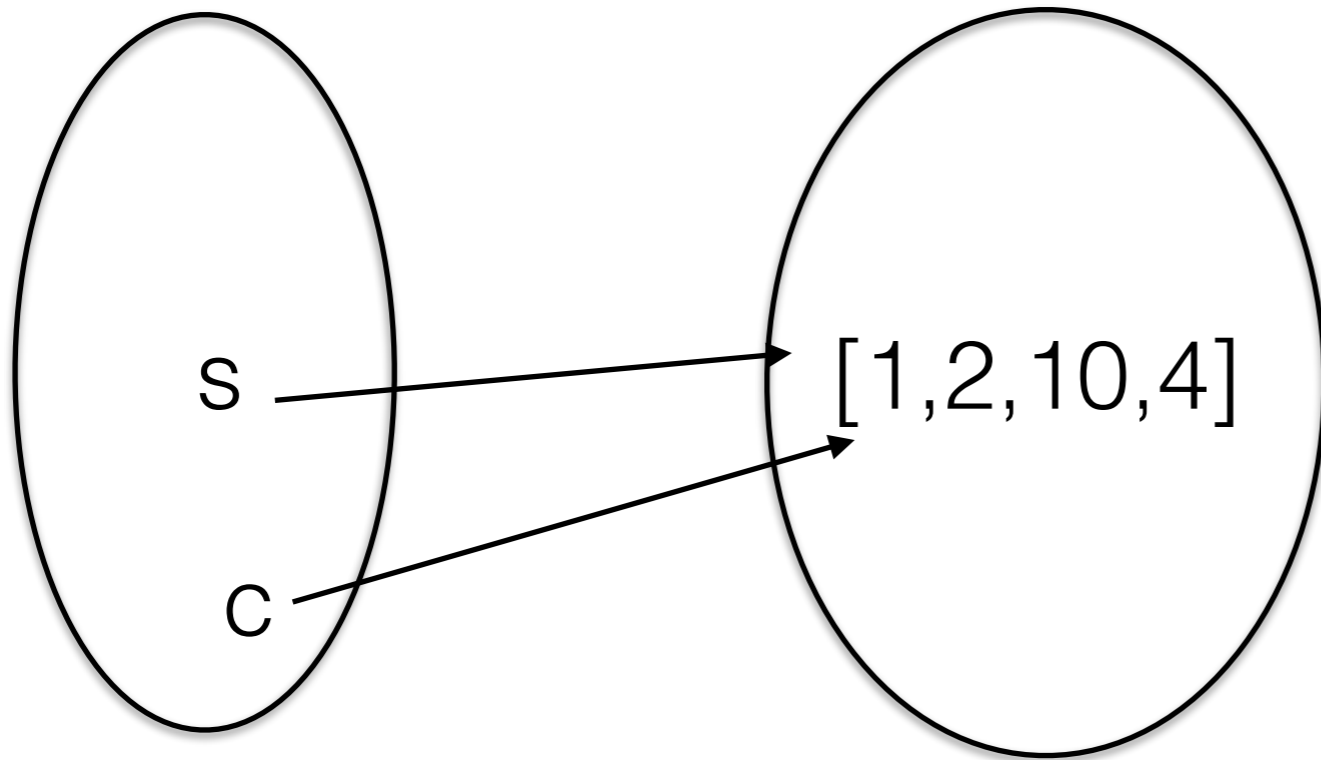
Λίστες



`s=[1,2,3,4]`

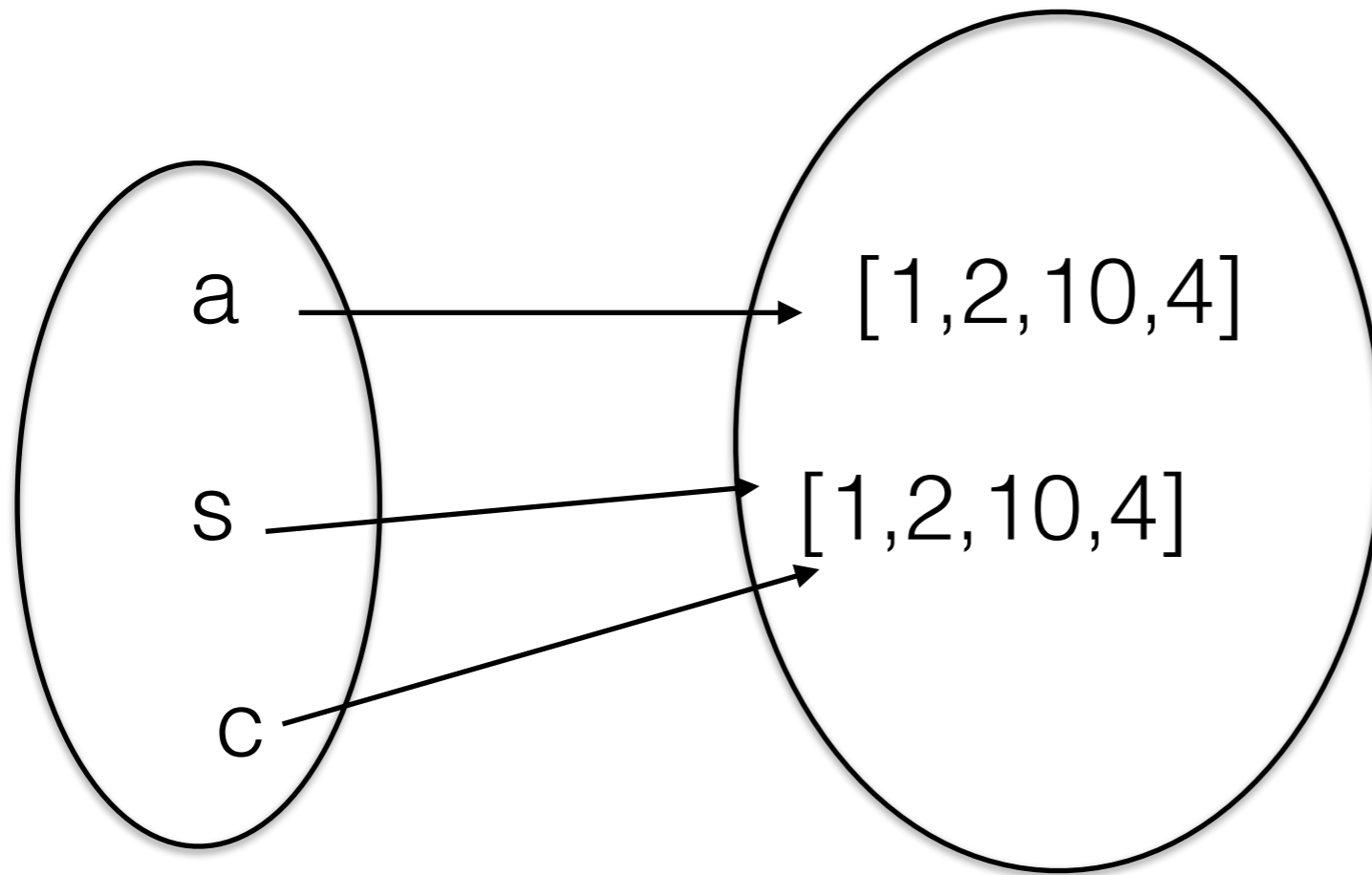
`c=s`

Λίστες



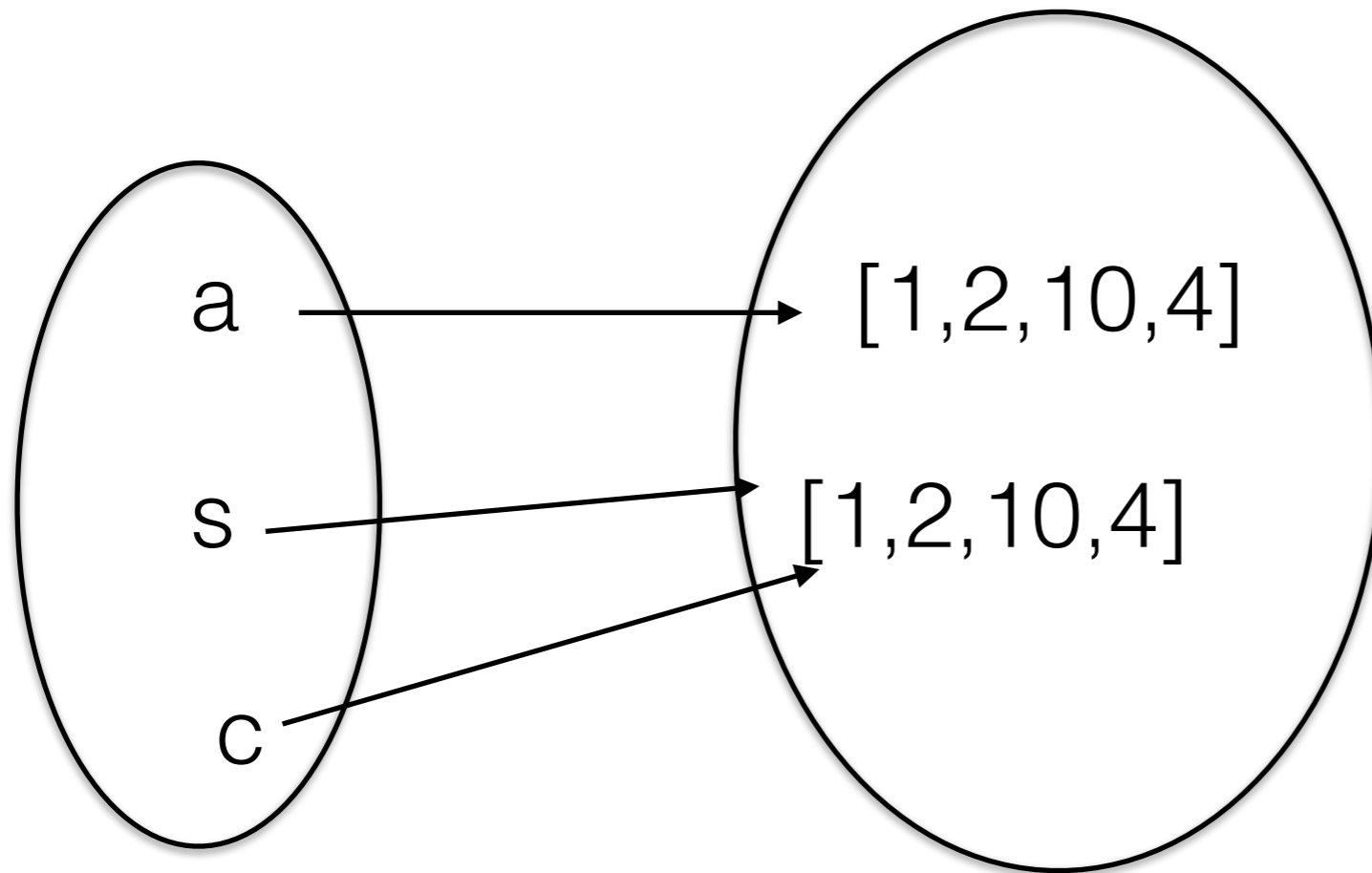
`c[2]=10`

Λίστες



`a=c[:]`

Λίστες



`s is c` → **True**
`a is c` → **False**
`a==c` → **True**

Λίστες - comprehension

```
Cdegrees = [-5 + i*0.5 for i in range(n)]
```

Φτιάχνει τη λίστα `[-5, -4.5, -4, -3.5, -3]` για `n=5`

```
Fdegrees = [(9.0/5)*C + 32 for C in Cdegrees]
```

Η λίστα με τις τιμές Fahrenheit που αντιστοιχούν στις τιμές της `Cdegrees`

```
C_plus_5 = [C+5 for C in Cdegrees]
```

Προσθέτουμε τον αριθμό 5 σε κάθε στοιχείο της `Cdegrees`

tuples

- Δομή παρόμοια με τις λίστες
- Δεν μεταβάλλονται
- `c=(2,1,'Hello')`
- Δεν μπορώ να διαγράψω, προσθέσω ή μεταβάλω στοιχείο της c

zip

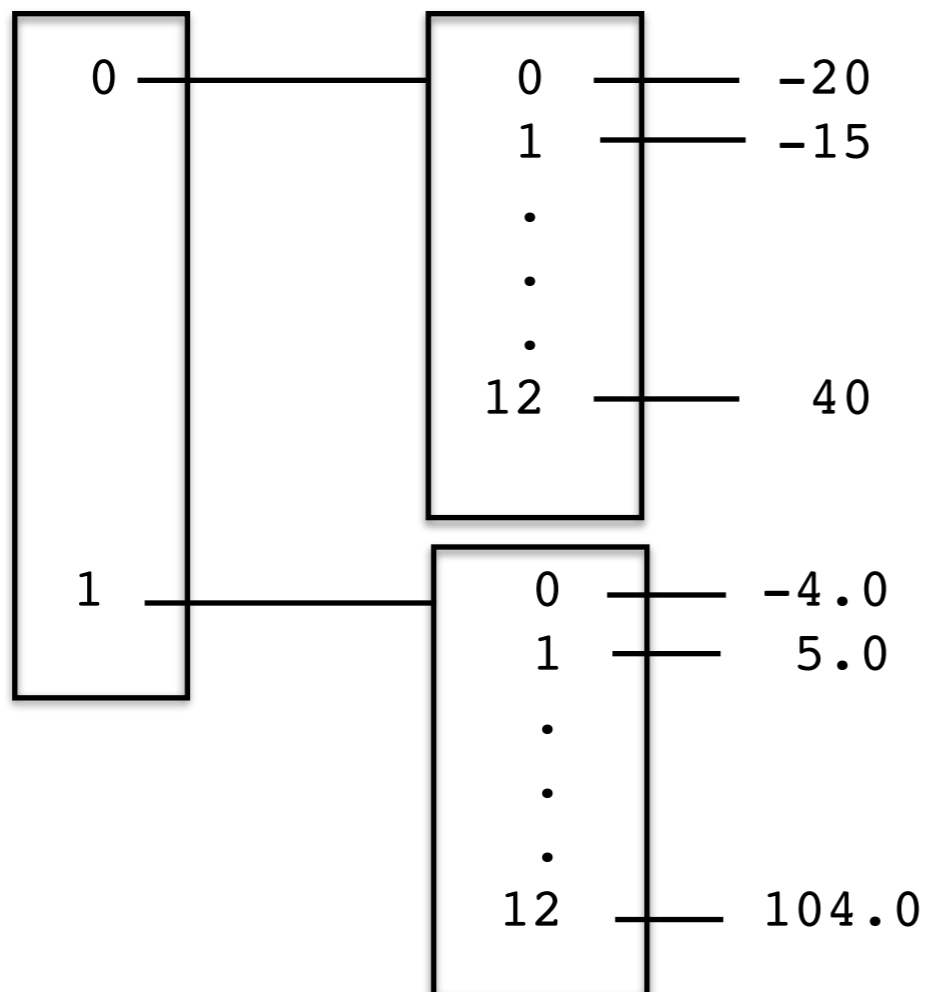
- C, F : Δύο λίστες
- $z = \text{zip}(C, F)$ (“ένωση”-“ζευγάρισμα” των C, F)
- $C = [1, 2, 3]; F = ['a', 'b', 'c']$
- $z = \text{zip}(C, F)$
- $z = [(1, 'a'), (2, 'b'), (3, 'c')]$
- z είναι λίστα και κάθε στοιχείο της είναι tuple

zip

```
for c,f in z:  
    print c,f
```

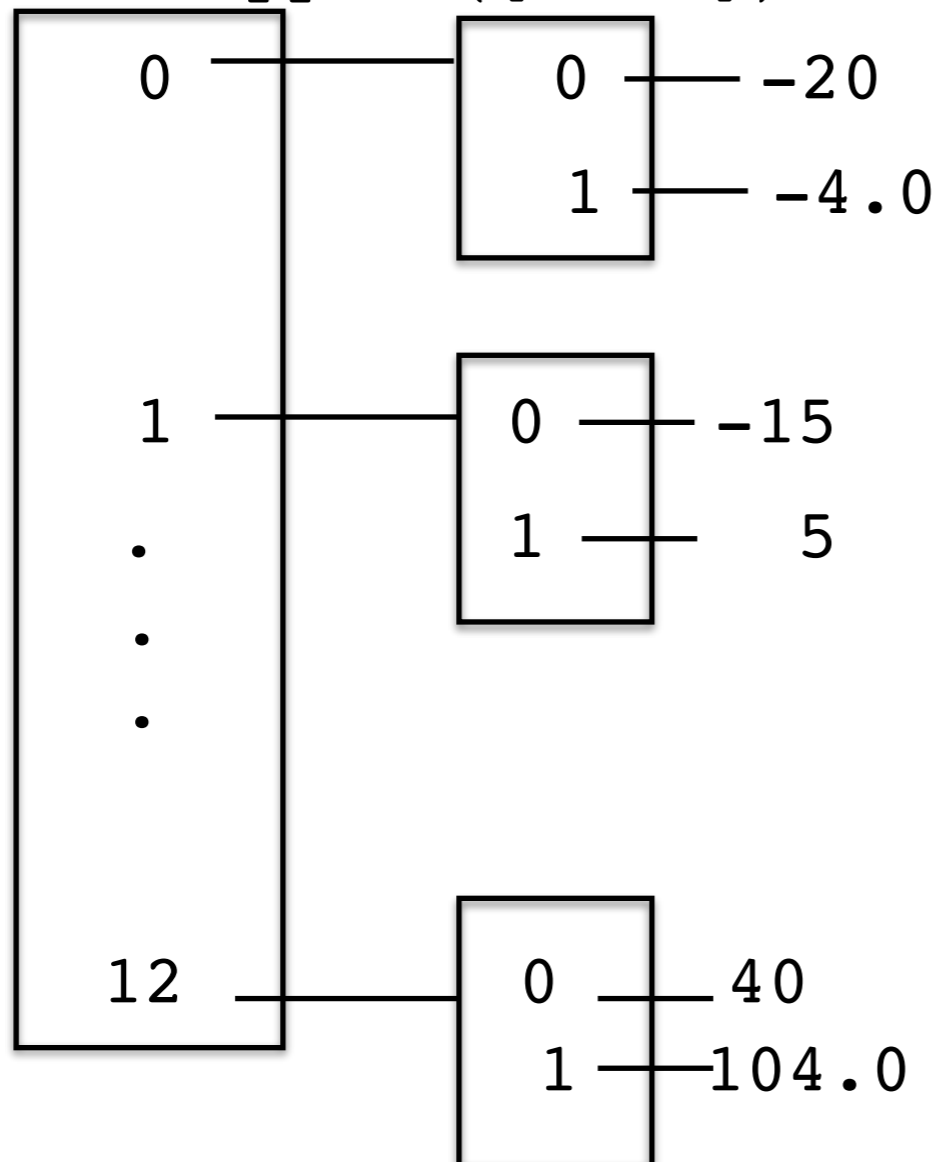
πίνακας

```
Cdegrees = [-20 + i*5 for i in range(n)]  
Fdegrees = [(9.0/5)*C + 32 for C in Cdegrees]  
table=[Cdegrees, Fdegrees]
```



πίνακας

```
table = []  
for C, F in zip(Cdegrees, Fdegrees):  
    table.append([C, F])
```



Λίστες - χρήσιμες εντολές

```
a=[1,20,9,40,'b',40,'a']  
a.remove('a') → [1,20,9,40,'b',40]  
a.index(20) → 1  
a.count(40) → 2  
len(a) → 6  
min(a) → 1  
max(a) → 'b'  
sum(a) →  
a.reverse(a) → [40,'b',40,9,20,1]  
a.sort() → [1,9,20,40,40,'b']
```

Συναρτήσεις

Συναρτήσεις

```
def F(C):  
    Fvalue=(9.0/5)*C+32  
    print 'Inside F: C=%g, Fvalue=%g, r=%g'%(C,Fvalue,r)  
    return '%g degrees C is %g degrees F'%(C,Fvalue)
```

```
r=21  
print F(r)  
print C  
print Fvalue
```

→ Σφάλμα - δεν έχει ορισθεί εκτός συνάρτησης

→ Σφάλμα - δεν έχει ορισθεί εκτός συνάρτησης

καθολική - τοπική μεταβλητή

```
print sum → Είναι built-in συνάρτηση της Python (δεν είναι μεταβλητή)  
sum = 500 → Ξαναορίζουμε την αντιστοιχία του ονόματος sum με έναν ακέραιο  
print sum → Η sum είναι μια καθολική μεταβλητή
```

```
def myfunc(n):  
    sum = n + 1  
    print sum → Η sum είναι μια τοπική μεταβλητή  
    return sum
```

```
sum = myfunc(2) + 1 # Νέα τιμή στην καθολική μεταβλητή sum  
print sum
```

καθολική - τοπική μεταβλητή

```
a = 20; b = -2.5
```

—————> καθολικές μεταβλητές

```
def f1(x):
```

```
    a = 21
```

—————> νέα τοπική μεταβλητή

```
    return a*x + b
```

—————> $21*x - 2.5$

```
print a
```

—————> 20

```
def f2(x):
```

```
    global a
```

—————> Η a ορίζεται ως καθολική μεταβλητή

```
    a = 21
```

—————> νέα τιμή της καθολικής μεταβλητής a

```
    return a*x + b
```

—————> $21*x - 2.5$

```
f1(3); print a
```

—————> 20

```
f2(3); print a
```

—————> 21