Συμπεριλήψη λίστας (list comprehension)

Μιχάλης Δρακόπουλος

Απρίλιος 2023

Είναι ένας *σύντομος* και *συμπαγής* τρόπος κατασκευής μια *νέας* λίστας από τα στοιχεία κάποιου αντικειμένου το οποίο πρέπει να *ακολουθεί το πρωτόκολλο επανάληψης*.

[ ]:
```
L = [1, 2, 3, 4, 5]
```

Η κατασκευή μιας λίστας `res` με τα στοιχεία της `L` αυξημένα κατά 10, μπορεί να γίνει ως εξής:

[ ]:
```
res = []
for x in L:
    res.append(x+10)
res
```

ή διαφορετικά με *συμπερίληψη*

[ ]:
```
res = [x + 10 for x in L]
res
```

Συμπερίληψη με συνθήκη Η νέα λίστα μπορεί να *συμπεριλαμβάνει* μόνο τα στοιχεία εκείνα που επαληθεύουν κάποια συνθήκη

[ ]:
```
even_squares = []
for x in range(12):
    if x%2 == 0:
        even_squares.append(x**2)
even_squares
```

[ ]:
```
even_squares = [x**2 for x in range(12) if x%2 == 0]
even_squares
```

Συμπερίληψη με διπλό `for`

[ ]:
```
[x + y for x in 'abc' for y in '123']
```

που είναι σαν να γράφαμε ισοδύναμα αλλά με περισσότερο κώδικα:

[ ]:
```
res = []
for x in 'abc':
    for y in '123':
        res.append(x+y)
res
```

Σύνθετες συμπεριλήψεις Χρήσιμες για επεξεργασία λίστας με στοιχεία λίστες.

Παράδειγμα: Επεξεργασία δισδιάστατων πινάκων Στην Python αν θέλουμε να δουλέψουμε με πίνακες γραμμικής άλγεβρας, χωρίς να χρησιμοποιήσουμε κάποια εξειδικευμένη βιβλιοθήκη (π.χ. NumPy), τους αποθηκεύουμε ως λίστες με στοιχεία λίστες.

Εστω ο πίνακας

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \end{bmatrix}$$

Μπορούμε να τον κατασκευάσουμε χωρίς συμπερίληψη

[ ]:
```
A = [[0, 1, 2, 3],
     [4, 5, 6, 7],
     [8, 9, 10, 11]
    ]
A
```

Η με *συμπερίληψη*

[ ]:
```
m = 3
n = 4
```

Μια γραμμή του A, έστω η 2 είναι:

[ ]:
```
i = 1
r2 = [n*i+j for j in range(n)]
r2
```

Ο A αποτελείται από 3 τέτοιες γραμμές:

[ ]:
```
A = [[n*i+j for j in range(n)] for i in range(m)]
A
```

[ ]:
```
A[2]
```

Για να πάρουμε τα στοιχεία μιας στήλης, έστω τη 2η ($j = 1$)

[ ]:
```
for row in A:
    print(row)
```

[ ]:
```
j = 1
c2 = [row[j] for row in A]
c2
```

Ο ανάστροφος του A

[ ]:
```
At = [[row[j] for row in A] for j in range(n)]
At
```

Με συμπερίληψη μόνο για τις στήλες, θα είχαμε ισοδύναμα:

[ ]:
```
At = []
for j in range(n):
    At.append([row[j] for row in A])
At
```

Και *χωρίς* συμπερίληψη

[ ]:
```
At = []
for j in range(n):
    col = []
    for row in A:
        col.append(row[j])
    At.append(col)
At
```