



ΠΕΡΙΕΧΟΜΕΝΑ:

1. Τελεστές
2. Γενικευμένη εξαγωγή χαρακτηριστικών και μεθόδων

ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ:

1. Python Advanced: Μάθημα 6 - Λάμδα και Συν/κός Προγ/μός

ΜΑΘΗΜΑ 6.3: operator

1. Τελεστές

- Το **module operator** περιέχει συναρτήσεις για (σχεδόν) όλους τους built-in τελεστές της python
- Χρησιμεύει:
 - Για να διοχετεύσουμε έναν τελεστή ως όρισμα σε μία συνάρτηση
 - Είναι πιο γρήγορο από το να διοχετεύσουμε ένα λάμδα που υλοποιεί τον τελεστή
 - Και συντακτικά πιο όμορφο από το να ορίσουμε δική μας εκδοχή συνάρτησης και να τη διοχετεύσουμε ως όρισμα.

Αριθμητικοί Τελεστές:

Τελεστής	Συνάρτηση
+ a	pos(a)
a + b	add(a, b)
- a	neg(a)
a - b	sub(a, b)
a * b	mul(a, b)
a / b	truediv(a, b)
a // b	floordiv(a, b)
a % b	mod(a, b)
a ** b	pow(a, b)

Bitwise Τελεστές:

Τελεστής	Συνάρτηση
a & b	and_(a, b)
a b	or_(a, b)
~ a	invert(a)
a ^ b	xor(a, b)
a << b	lshift(a, b)
a >> b	rshift(a, b)

Σχεσιακοί Τελεστές:

Τελεστής	Συνάρτηση
a < b	lt(a, b)
a <= b	le(a, b)
a == b	eq(a, b)
a != b	ne(a, b)
a >= b	ge(a, b)
a > b	gt(a, b)

Λογικοί Τελεστές:

Τελεστής	Συνάρτηση
not a	not_(a)
a is True	truth(a)
a is b	is_(a, b)
a is not b	is_not(a, b)

Slicing:

s[i:j] = v	setitem(s, slice(i,j), v)
del s[i:j]	delitem(s, slice(i,j))
s[i,j]	getitem(s, slice(i,j))

Indexing:

a[b]	getitem(a,b)
del a[b]	delitem(a, b)
a[b] = c	setitem(a,b,c)

Ακολουθίες:

Τελεστής	Συνάρτηση
a + b	concat(a, b)
a in b	contains(b, a)
a.count(b)	countOf(a, b)
a.index(b)	indexOf(a, b)

Τελεστές Επαυξημένης Καταχώρησης:

Ορίζεται επίσης ένας τελεστής για τους αντίστοιχους αριθμητικούς ή bitwise τελεστές:

- π.χ. `iadd(a, b)` αντιστοιχεί στον τελεστή `a+=b`

Παράδειγμα 1: operators.py

```
from operator import add
from functools import reduce
from random import randrange
my_list = [randrange(10) for _ in range(10)]
print(my_list)
def my_add(a, b):
    return a+b
print(reduce(my_add, my_list, 0))
print(reduce(add, my_list, 0))
print(reduce(lambda x, y: x+y, my_list, 0))
```

- Οι παρακάτω συναρτήσεις εξάγουν χαρακτηριστικά και στοιχεία με γενικευμένο τρόπο:

Συνάρτηση	Επεξήγηση
attrgetter(attr)	Επιστρέφει αντικείμενο που είναι callable. Αυτό, με όρισμα ένα άλλο αντικείμενο x, επιστρέφει το χαρακτηριστικό του attr του x [Με παραπάνω ορίσματα, επιστρέφει το attr των επιμέρους ορισμάτων]
itemgetter(pos)	Επιστρέφει αντικείμενο που είναι callable. Αυτό, με όρισμα ένα άλλο αντικείμενο x, επιστρέφει το στοιχείο στη θέση pos του x [Με παραπάνω ορίσματα, επιστρέφει τα στοιχεία τους στη θέση pos]

Παράδειγμα 2: getters.py

```
a = A() # A: class {x, y}. getitem[0] == x, getitem[1] == y
b = B() # B: same
```

```
attr_x = attrgetter("x")
print(attr_x(a))
print(attr_x(b))
```

```
item = itemgetter(1)
print(item(a))
print(item(b))
```

- Ενώ η ακόλουθη συνάρτηση, ορίζει έναν γενικευμένο τρόπο για να καλούμε μία μέθοδο την οποία την ορίζουμε μέσω συμβολοσειράς ως μέθοδο κλάσης.

Συνάρτηση	Επεξήγηση
methodcaller(method, *args, **kwargs)	Επιστρέφει αντικείμενο που είναι callable. Αυτό, με όρισμα ένα άλλο αντικείμενο x, επιστρέφει το αποτέλεσμα της κλήσης της μεθόδου method του x με ορίσματα τα *args, **kwargs.

Παράδειγμα 3: method_caller.py

```
a = A() # A: class {x, y}. f(arg) = x + arg
b = B()
```

```
method_f = methodcaller("f", 100)
print(method_f(a))
print(method_f(b))
```

Παράδειγμα 3: sort by index.py

```
A = [(1,19,29), (5, 14, 24), (9, 11, 27)]
```

```
for i in range(3):
    print(sorted(A, key=itemgetter(i)))
```