

Laboratorul 6 - Multimi

Formatori:

Tutor: [Militaru Mihai-Adrian](#)  

Tutor: [Dragomir Titian-Cornel](#)  

+8

.....
⇩

Data de începere a cursului:

 25.09.2023

 [Utilizatori înscrși](#)

 [Calendar](#)

 [Note](#)

 [Cursurile mele](#) ▶ [S1-L-AC-CTIRO1-LSD](#) ▶ Săptămâna 6: Mulțimi, Relații, Dicționare ▶ [Laboratorul 6 - Multimi](#)

Laboratorul 6 - Multimi

Mulțimi (set)

Documentatia oficiala python: <https://docs.python.org/3/tutorial/datastructures.html#sets>.

O multime poate fi definita cu acolade {}:

```
set1 = {1, 2, 3}
print(set1)
```

Multimile au următoarele caracteristici:

- Sunt neordonate.
- Elementele multimilor sunt unice. Elementele duplicat nu sunt permise.
- O multime în sine poate fi modificata, dar elementele conținute în multime trebuie să fie de tip imuabil (obiectele imuabile sunt obiecte a căror valoare nu poate fi modificată după inițializare).

Multimile (sets) suportă operații matematice precum unirea, intersecția, diferența și diferența simetrică.

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
print(set1) # {1, 2, 3}
print(set2) # {3, 4, 5}

diferenta = set1 - set2 # elemente care exista in set1 dar nu si in set2
print(diferenta) # {1, 2}

unire = set1 | set2 # elemente care exista in set1 sau in set2
print(unire) # {1, 2, 3, 4, 5}

intersectie = set1 & set2 # elemente care exista in set1 si in set2
print(intersectie) # {3}

diferenta_simetrica = set1 ^ set2 # elemente care exista in set1 sau in set2, dar nu in ambele
print(diferenta_simetrica) # {1, 2, 4, 5}
```

Funcția len() returnează numărul de elemente dintr-o mulțime, iar operatorii *in* și *not in* pot fi folosiți pentru a testa apartenența:

```
multime1 = {'ana', 'maria', 'ion'}
multime_vida = {}

print('ana' in multime1) # TRUE
print('florin' in multime1) # FALSE
print('florin' not in multime1) # TRUE
```

Parcurgerea multimilor cu ajutorul funcției reduce():

```
import functools

def parcurgere_simpla(multime):
    functools.reduce(lambda acc, element: print(element), multime, None)

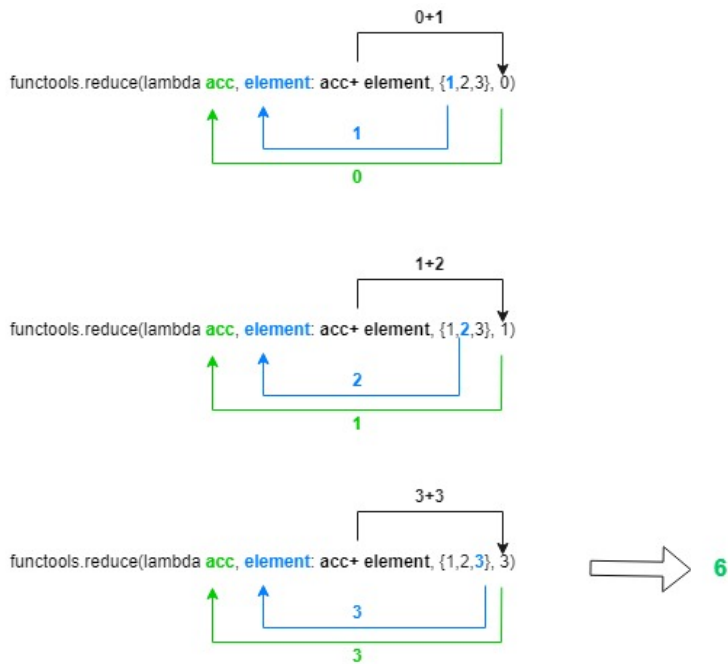
parcurgere_simpla({1, 2, 3, 4, 5})
```

Suma elementelor unei multimi de intregi:

```
import functools

def suma_multime_int(multime):
    return functools.reduce(lambda acc, element: acc + element, multime, 0)

print(suma_multime_int({1, 2, 3}))
```



Exerciții rezolvate cu Mulțimi:

1. Scrieți o funcție care returnează mulțimea tuturor divizorilor unui număr pozitiv n dat ca argument. Parcurgerile se vor realiza cu ajutorul funcțiilor recursive.

```
import math as m

def multimea_divizorilor(n, A=set(), i=2):
    if i > m.sqrt(n):
        return A
    else:
        if n % i == 0:
            A.update({i, int(n/i)})
        return multimea_divizorilor(n, A, i+1)

print(multimea_divizorilor(20))
```

2. Implementați funcția standard filter care ia ca parametri o funcție booleană (condiție, predicat) f și o mulțime s și returnează mulțimea elementelor din s care satisfac funcția f . Parcurgeți mulțimea inițială cu funcția `reduce()`.

```
import functools
def my_filter(f, A, B=set()):
    def functie(acc, elem):
        if(f(elem)):
            B.add(elem)
        return f(elem)
    functools.reduce(functie, A, 0)
    return B

def impar(x):
    return x % 2
B=set(my_filter(impar, {1, 2, 3, 4, 5, 6}))
print(B)
# {1, 3, 5}
```

◀ Codul din curs Dictionare- în Python

Sari la...

Exerciții - Săptămâna 6 - Multimi ▶

✉ Contactați serviciul de asistență

Sunteți conectat în calitate de Ciobanu Daria-Andreea (Delogare)

S1-L-AC-CTIRO1-LSD

Meniul meu

Profil

Preferinte

Calendar

🗺 ZOOM

Română (ro)

English (en)

Română (ro)

Rezumatul păstrării datelor

Politici utilizare site