

Data Analysis and Processing using Python 1

Silviu Ojog - <https://www.youtube.com/@SilviuOjog>

***LINK*Academy**



Tipuri de programare

- Imperativa
 - Procedurala
 - Orientata pe obiecte
- Declarativa
- Functionala



Tipuri de programare

- Procedurale:
 - Limbaje: C (exclusiv), Python(procedural+oop), Php
 - Se bazeaza pe proceduri in executia programului (functii)

Tipuri de programare

- Obiect orientate:
 - Limbaje: Python, Java(exclusiv), C++, Swift, Php
 - Folosesc obiecte



Tipuri de programare

- Obiect orientate:
 - Limbaje: Python, Java(exclusiv), C++, Swift, Php
 - Folosesc obiecte





Tipuri de programare

- Declarative
 - Limbaje: HTML, SQL, XML
 - ex: Afișează-mi cel mai bătrând utilizator
 - `select * from users order by age desc limit`
 - Desenează-mi un cadru roșu
 - `<div style='border:1px solid red'></div>`



Functional programming

- lambda
 - *o functie anonimă*
- map
 - *mapeaza(transforms) un tip de date (colectie, secventa sau iterator) in alt tip de date cu ajutorul unei functii lambda*
- filter
 - *filtreaza un anumit tip de data cu ajutorul unei functii lambda*
- reduce
 - *reduce tot un tip de date la o singura valoare*



Functional programming

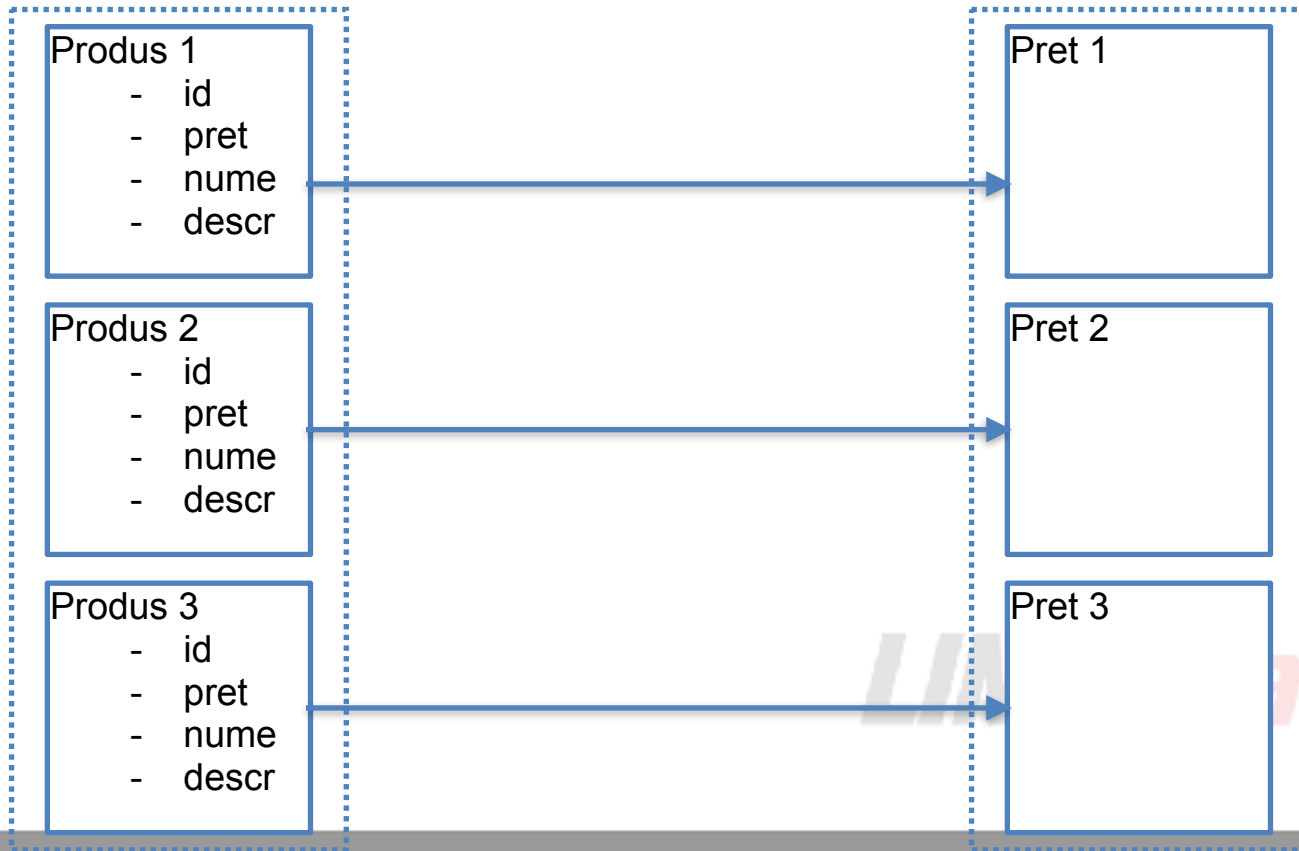
- lambda
 - *lambda arguments : expression*
- map
 - *map(function, iterable1, iterable2,...)*
- filter
 - *filter(function, iterable)*
- reduce
 - *reduce(function, iterable)*



Map

O lista initiala

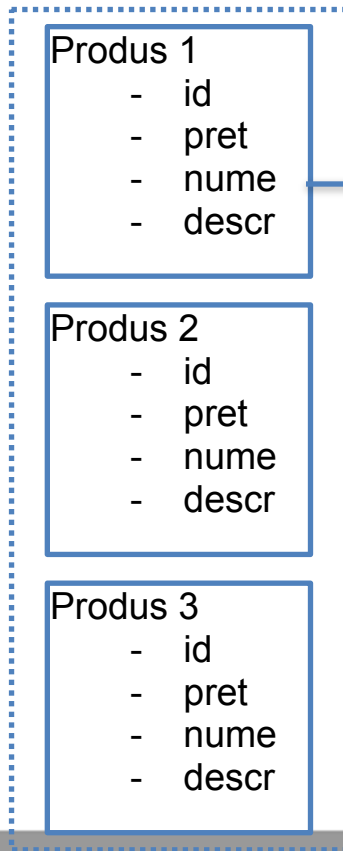
O lista Transformata



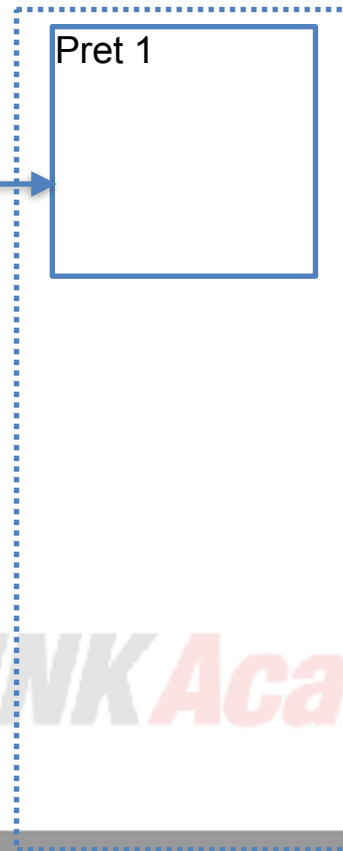


Map

O lista initiala



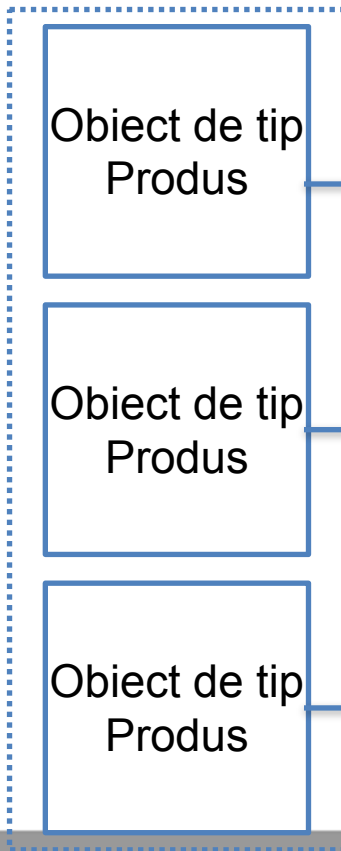
O lista Transformata



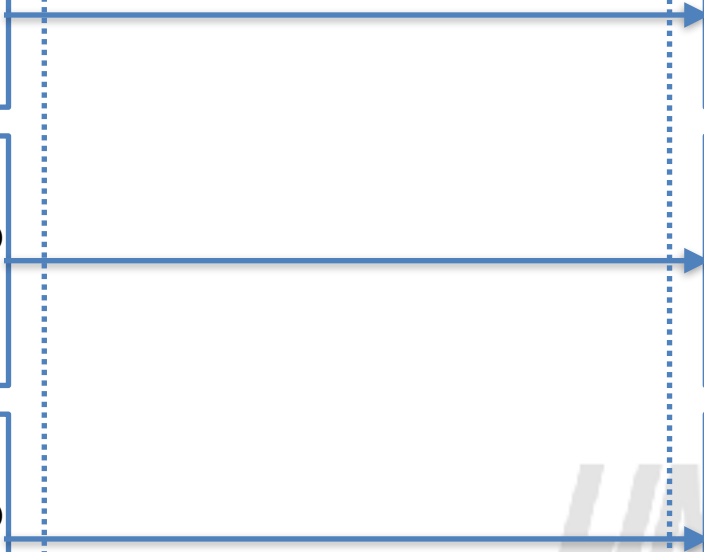
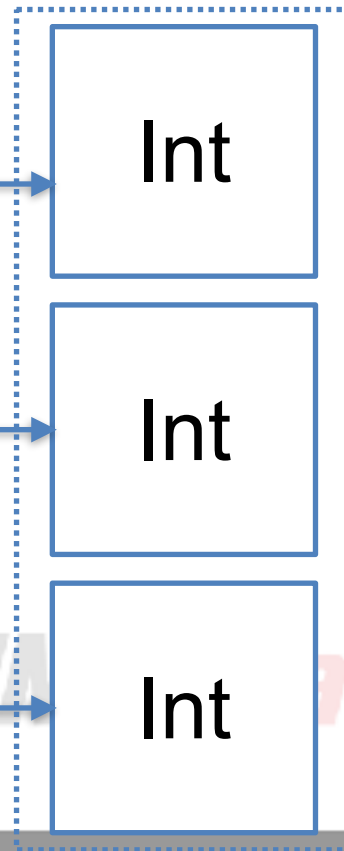


Map

O lista initiala

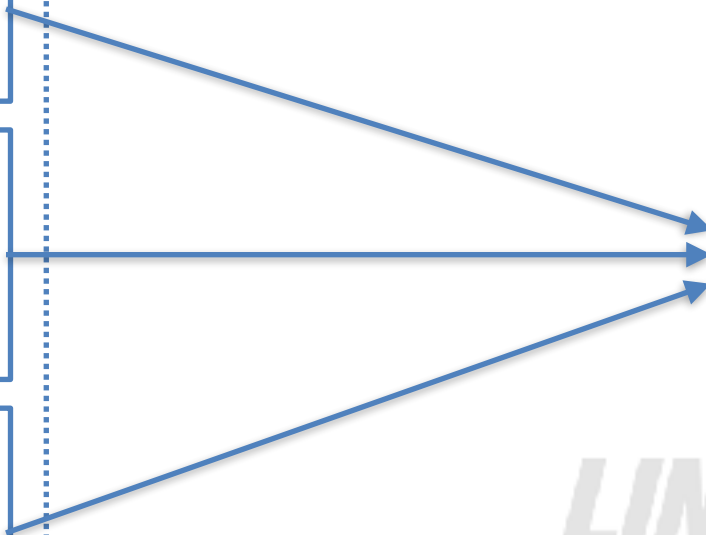
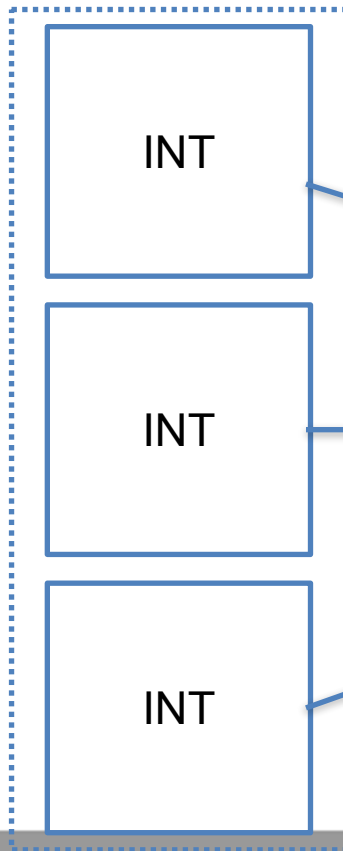


O lista Transformata



Reduce

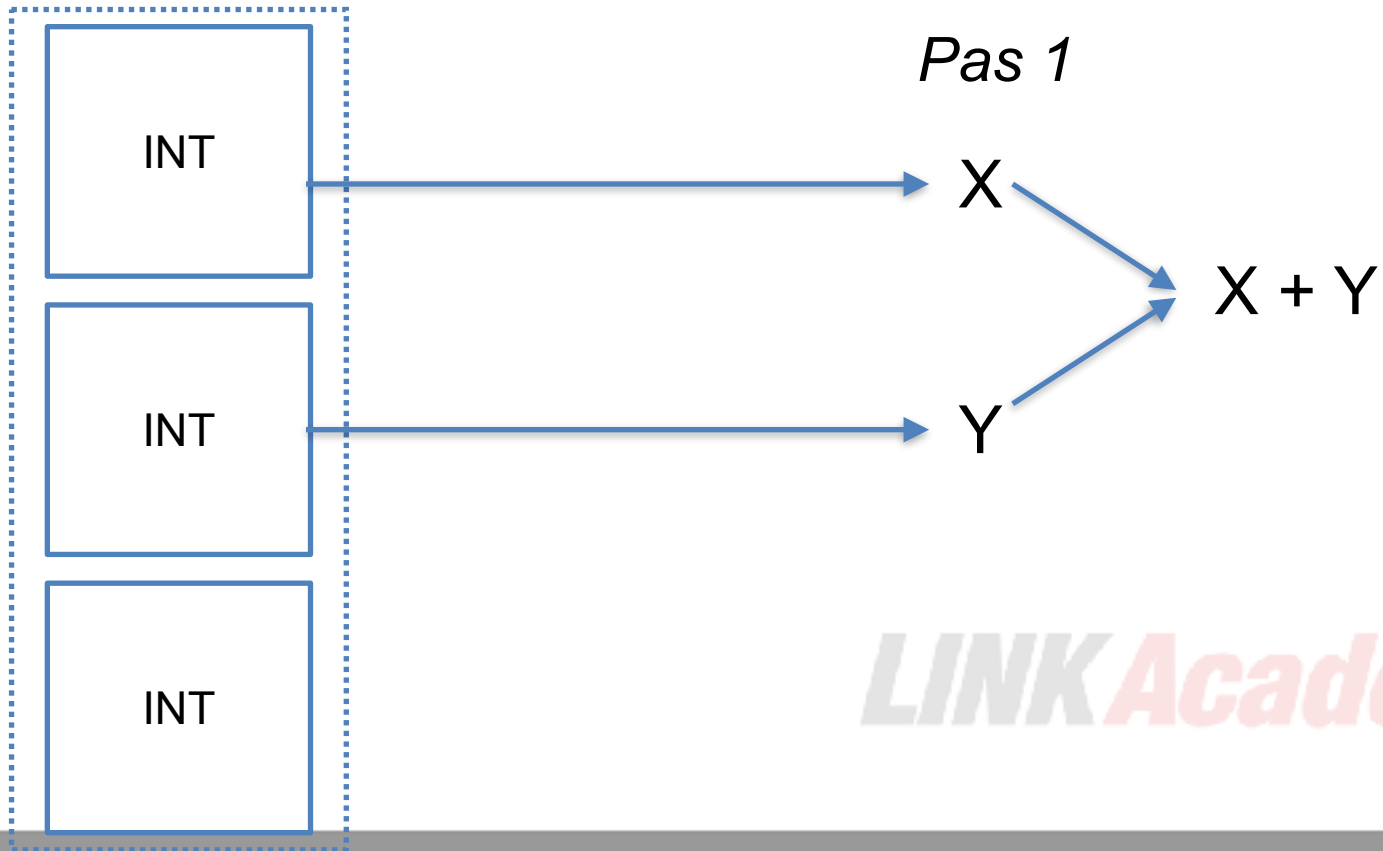
O lista initiala



O singura valoare
(suma lor)

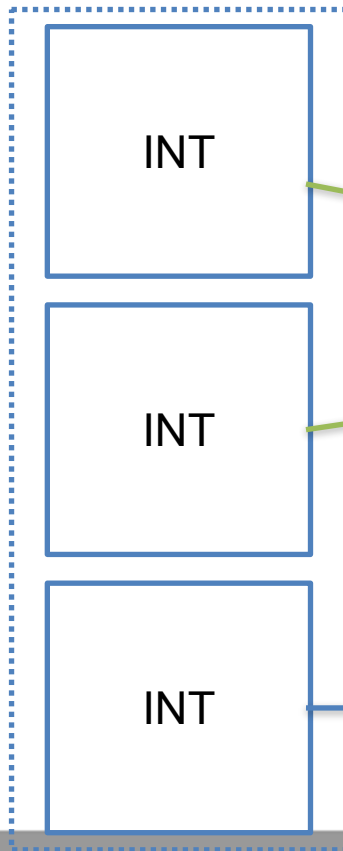
Reduce

O lista initiala

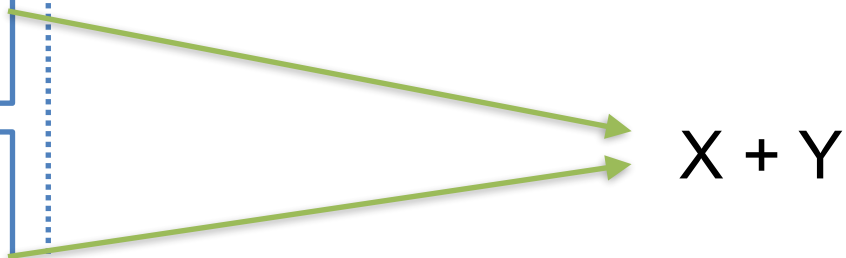


Reduce

O lista initiala

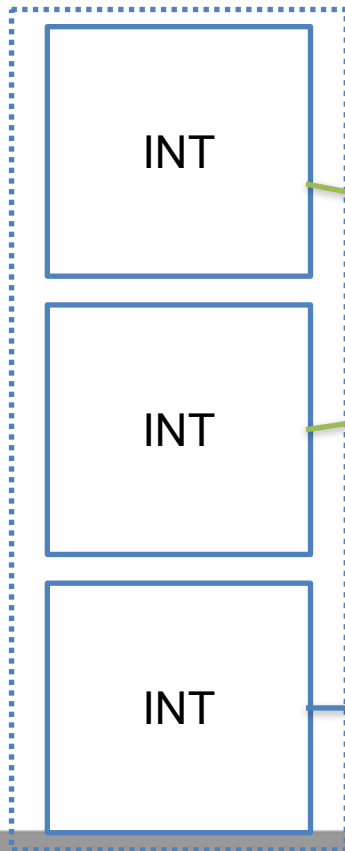


Pas 2

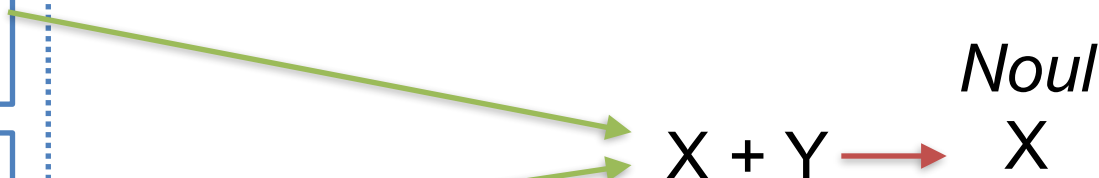


Reduce

O lista initiala



Pas 2

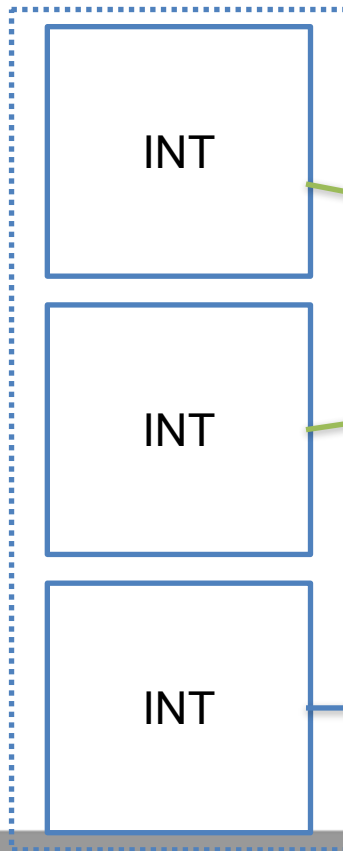


Silviu Ojog

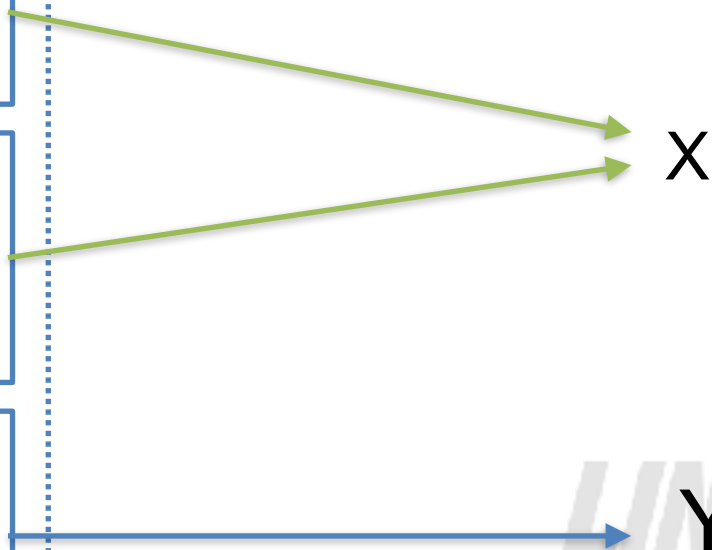
LINK Academy

Reduce

O lista initiala

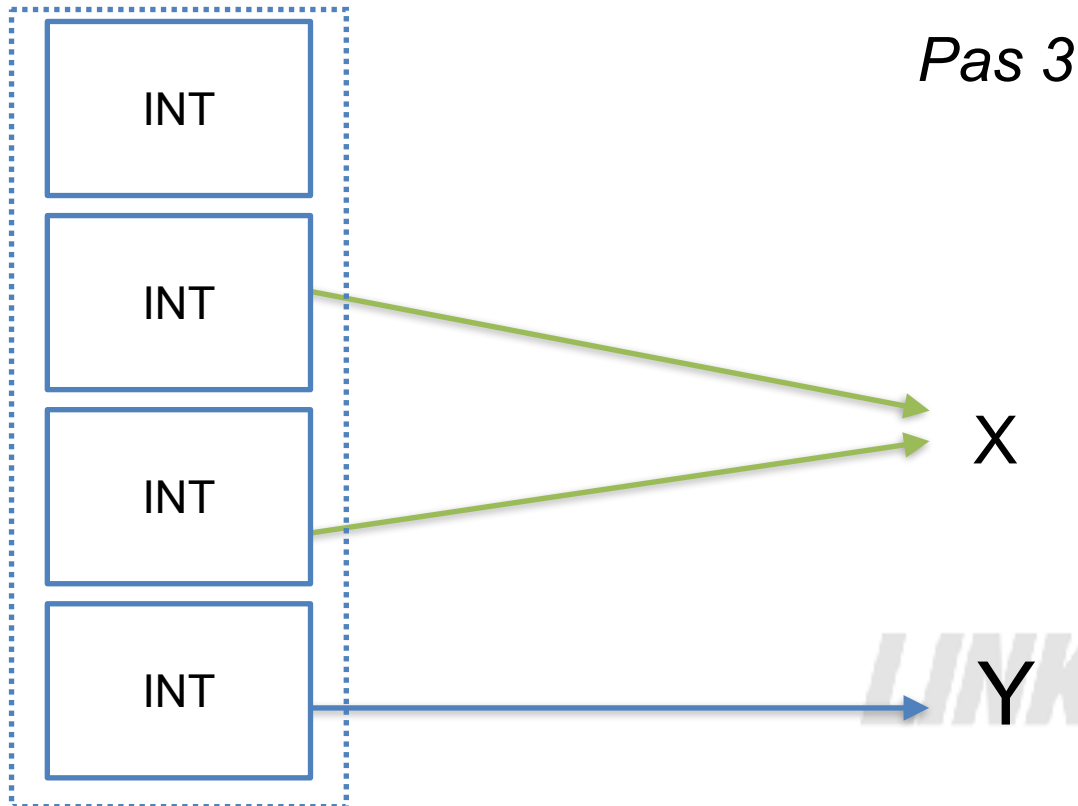


Pas 2



Reduce

O lista initiala



Silviu Ojog

Exercițiul 1

Creați o funcție Python care primește o listă de întregi ca argument și returnează o nouă listă care conține doar valori mai mari de 5.

lista = [10, 2, 30, 50, 300, 10]

Exercițiul 1

Creați o funcție Python care primește o listă de șiruri ca argument și returnează o nouă listă care conține doar șirurile mai lungi decât 5 caractere

lista = [10, 2, 30, 50, 300, 10]

Exercițiul 1

Creați o funcție Python care primește o listă de șiruri ca argument și returnează o nouă listă care conține doar șirurile mai lungi decât 5 caractere

lista = [10, 2, 30, 50, 300, 10]

R: *[x for x in lista if x > 50]*

Exercițiul 2

Definiți o funcție Python care primește un string ca argument și returnează un nou string în care toate vocalele au fost eliminate

```
vocale = "aeiouAEIOU"
```

```
input_string = "Salutare, ce mai faci?"
```

Exercițiul 2

Definiți o funcție Python care primește un string ca argument și returnează un nou string în care toate vocalele au fost eliminate

```
vocale = "aeiouAEIOU"
```

```
input_string = "Salutare, ce mai faci?"
```

```
R: "".join(char for char in input_string if char not in  
vocale)
```

Exercițiul 3

Creați următoarea listă într-o singură linie de comandă

```
my_list = ["1", "2", "3", "4"]
```

Exercițiul 3

Creați următoarea listă într-o singură linie de comandă

```
my_list = ["1", "2", "3", "4"]
```

```
R: [str(x) for x in range(1,5)]
```


Exercițiul 4

Scrieți o funcție Python care primește o listă de numere ca argument și returnează media acestora

lista = [10, 2, 30, 50, 300, 10]

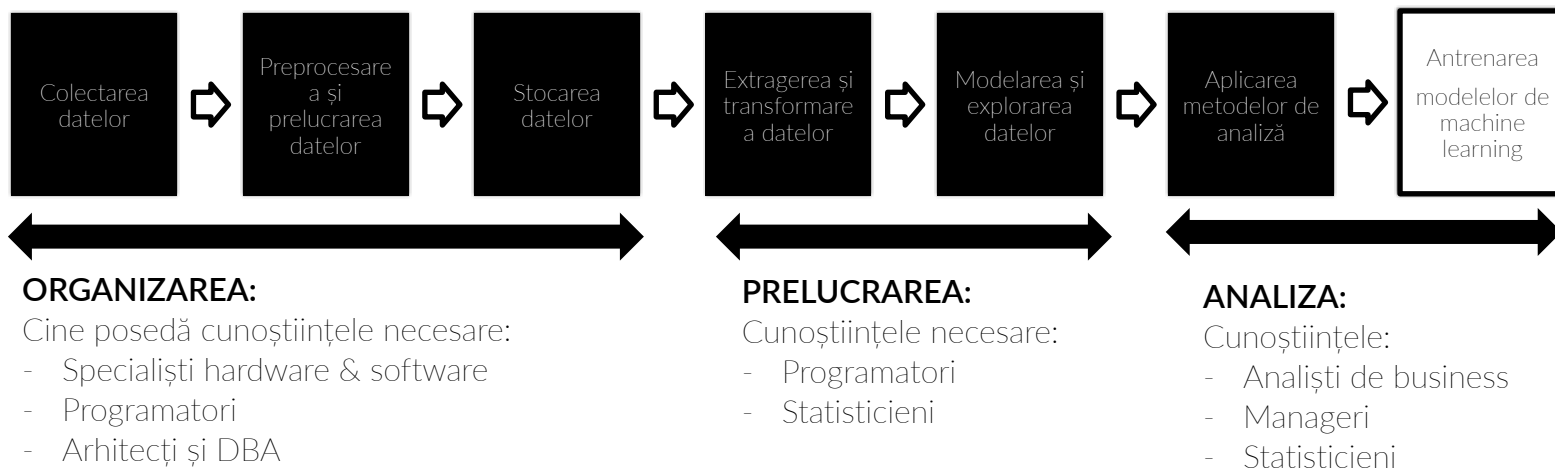
Exercițiul 4

Scrieți o funcție Python care primește o listă de numere ca argument și returnează media acestora

lista = [10, 2, 30, 50, 300, 10]

sum(lista) / len(lista)

ETAPELE DE ANALIZĂ A DATELOR



Tipuri de secvențe

- gama (**range**)
- string (**string**)
- lista (**list**)
- tuplul (**tuple**)
- șirul de bytes (**byte array**)
- secvența de bytes (**byte sequence**)

Biblioteca numpy

Biblioteca numpy

- NumPy este o bibliotecă Python pentru manipularea avansată a șirurilor/matricelor.
- Este o componentă indispensabilă a unui pachet de date
 - <https://numpy.org/devdocs/user/basics.types.html>

Biblioteca numpy

- Instalare librăriei
 - `pip install numpy`
- Folosirea librăriei
 - `import numpy as np`

Biblioteca numpy

- Instalare librăriei
 - `pip install numpy`
- Folosirea librăriei
 - `python`
 - `>>> import numpy as np`

Crearea șirului numpy

- NumPy funcționează în cea mai mare parte cu obiecte ndarray.
- Ndarray este un șir (array) multidimensional.
- Generarea unui obiect ndarray se face apelând funcția array.
- Funcția array este transmisă unui tip enumerabil sau un șir este generat de funcția arrange sau de o transformare a secvenței Python.

```
>>> arr = np.array([1,2,3,4,5])
```

```
>>> arr = np.arange(1,5)
```

LINK Academy

Tipizarea șirului numpy

- Șirurile numpy sunt strict tipizate. (toate elementele din vector au aceeași valoare)
- Tipul de șir se determină automat.
 - `arr = np.array([1,2,3,4,5])` #Tipul este int64
 - `arr = np.array([1.,2,3,4,5])` #Tipul este float64
 - `arr = np.array([1,2,3,4,5], dtype=bool)` #Tipul este bool

Crearea șirului numpy

- Array-urile
 - `arr = np.array([1,2,3,4,5])`
 - `arr = np.arange(1,5)`
- Obiectul obținut a fost tipizat și tratat ca o matrice.
- Fiecare șir (rând) al matricei se numește axă (**axis**)
 - `array([1, 2, 3, 4])`

LINK Academy

Tipizarea șirului numpy

- `>>> arr = np.arange(1,5)`
- `>>> arr`
- `array([1, 2, 3, 4])`
- `>>> type(arr)`
- `<class 'numpy.ndarray'>`
- `>>> arr.dtype`
- `dtype('int64')`

Tipuri de date

Numpy type	Description
np.int8	Byte (-128 to 127)
np.int16	Integer (-32768 to 32767)
np.int32	Integer (-2147483648 to 2147483647)
np.int64	Integer (-9223372036854775808 to 9223372036854775807)
np.uint8	Unsigned integer (0 to 255)
np.uint16	Unsigned integer (0 to 65535)
np.uint32	Unsigned integer (0 to 4294967295)
np.uint64	Unsigned integer (0 to 18446744073709551615)
np.intp	Integer used for indexing, typically the same as ssize_t
np.uintp	Integer large enough to hold a pointer
np.float32	Note that this matches the precision of the builtin python float
np.float64 / np.float_	
np.complex64	Complex number, represented by two 32-bit floats (real and imaginary components)
np.complex128 / np.complex_	

Vectori multidimensionali

- Numpy este capabil să detecteze șiruri multidimensionale și susține diverse operații aritmetice pe ele.

```
>>> arr = np.array([[1,2,3],[4,5,6]])  
array([[1, 2, 3], [4, 5, 6]])
```

Vectori multidimensionali

- Operatii

```
>>> arr = np.array([[1,2,3],[4,5,6]])
```

```
array([[1, 2, 3], [4, 5, 6]])
```

```
>>> print(arr+2)
```

```
[[3 4 5] [6 7 8]]
```

```
>>> print(arr*2)
```

```
[[ 2  4  6] [ 8 10 12]]
```

```
>>> print(arr*arr)
```

```
[[ 1  4  9] [16 25 36]]
```

Operatii pe array-uri

- Numpy aplică operații aritmetice pe șiruri dacă dimensiunile se pot potrivi:

```
>>> a = np.array([[1,2,3],[4,5,6]])  
>>> b = np.array([[10,10,10],[10,10,10]])  
>>> print(a*b)  
[[10 20 30] [40 50 60]]
```

- Dacă nu se potrivesc apare eroare:

```
>>> c = np.array([10,10])  
>>> print(a*c)
```

ValueError: operands could not be broadcast together with shape

Remodelarea șirurilor

- Numpy poate remodela șirurile multidimensionale și în acest caz trebuie să vă asigurați ca forma țintă să corespundă conținutului curent.

```
>>> arr = np.array([[1,2,3,4],[5,6,7,8]])
```

```
>>> arr.reshape(4,2)    #forma acceptant  
array([[1, 2],[3, 4],[5, 6],[7, 8]])
```

```
>>> arr.reshape(3,2)    #forma neacceptant
```

```
ValueError: cannot reshape array of size 8 into shape (3,2)
```

Compresia șirurilor

- Metoda squeeze reduce șirul la valorile șirurilor unidimensionale.

```
>>> arr = [[ [1],[2],[3] ]]
```

```
>>> print(np.squeeze(arr))
```

Şirurile structurale

- Numpy permite simularea structurii datelor prin şiruri multidimensionale definite special.

```
>>>tp = [('name', 'U10'), ('price', 'float')]
```

```
>>>x = np.array([("Phone", 125.9), ("TV", 82.22)], tp)
```

```
>>>print(x[0]["name"])
```

```
Phone
```

```
>>>print(x[0]["price"])
```

```
125.9
```

Exercițiul cu anii

- În program sunt definite variabilele startDate și endDate care reprezintă anul inițial și cel final.
- Trebuie creat un program care, pe baza acestor două valori, va afișa lista anilor.
- Aplicația trebuie să aibă o ieșire identică celei de mai jos:

```
***** Allowed years *****
2010
2011
2012
2013
2014
2015
*****
```

Exercițiul cu anii

```
***** Allowed years *****  
2010  
2011  
2012  
2013  
2014  
2015  
*****
```

-
- Notă:
 - Comanda range() se poate folosi cu parametri. De exemplu:
 - range(1,10)
 - Generează colecția de la numărul 1 la numărul 9