

23 Avril 2025

- 1-) Return sur la séance 1.
- 2-) Programmation Python.
  - Structure Tert . ✓
  - Structure Bunch . ✓
- 3-) Séquence / Collections.

1) Introduction A53 → Collecte de données.  
↳ Des modèles prédition.

2) Processus de Dev modèles

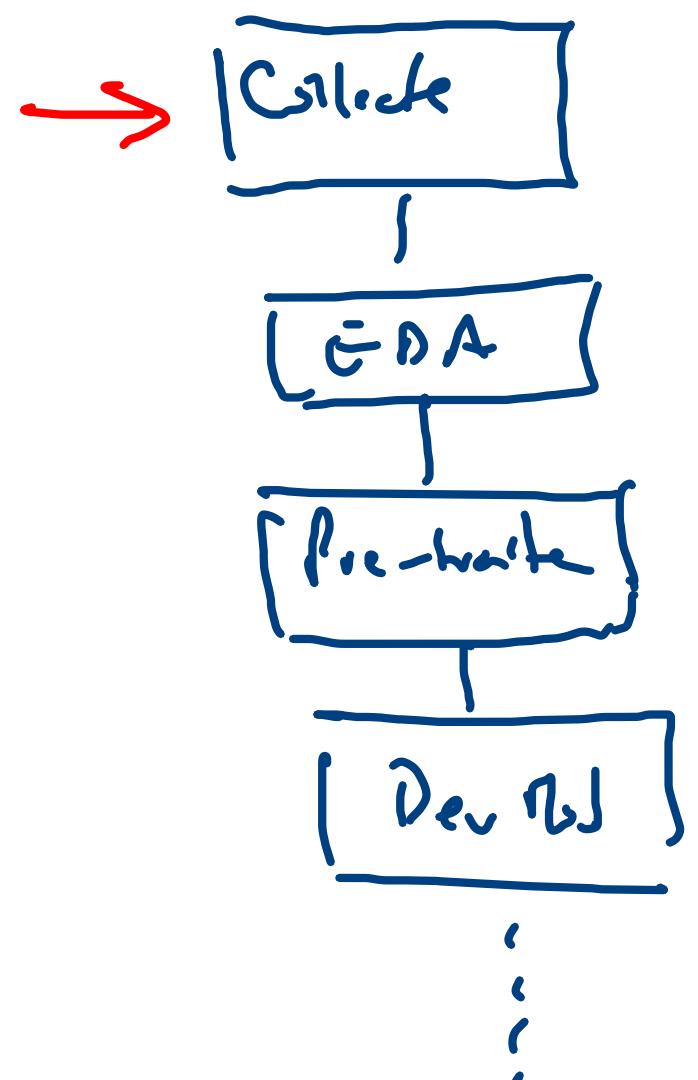
3) Info Python.

- Affectation variable.

- Lecture et Fonction variable.

⇒ IDE pycharm

Projet IMC.



Python Pas de notion de constante !!

if sum > MAX : MAX ↴  
{

MAX = 15

<

:

:

MAX ??

if one-way  
(une voie)

```
if condition :  
    ; inst à execute.  
    .  
    .  
    → ligne vide  
    ;
```

short circuit.

if two-way  
(deux voies)

```
if condition :  
    ; inst si True.  
    .  
    else :  
        ; inst si False.  
        .  
        .  
        → ligne vide.  
        .  
        .
```

CASE  
~~SWITCH~~

if multi-way  
(plusieurs voies)

```
if condition :  
    ;  
elif condition :  
    ;  
elif condition :  
    ;  
else :  
    ;  
    .  
    .  
    → ligne vide  
    ;
```

# PEP Python Enhancement Proposal

PEP 8: Convention over magic.

nom

Num X

NUM

sensible in  
a case.

Sucre cas.

variable { salaire  
          { taux\_interet

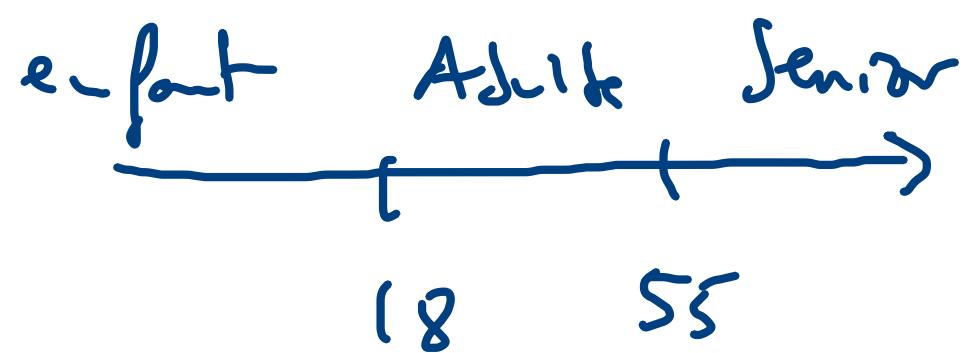
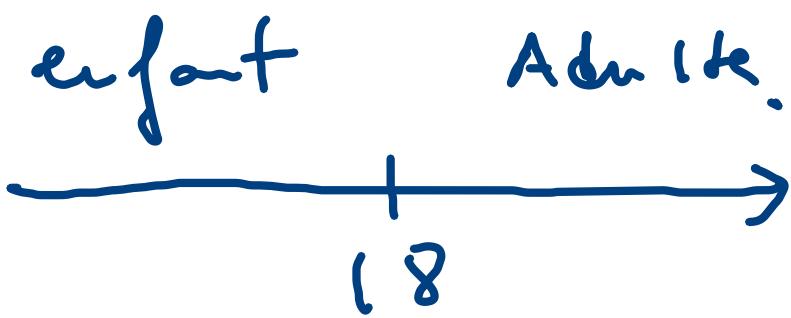
constante { TAUX  
          { TAUX\_INTERET

fct { calculer()  
      { Calculer\_taux()

classe { Patient  
          { Patient\_Celebre  
          { .

Test

Age



1

## Sequences, Collections (Mémoire)

list

[ , ..., ... ]

 $\rightarrow 0$  $-1 \leftarrow$ 

- MUTABLE ✓
- Doublovs ✓
- index [ ] / slicing
- méthodes : append()  
insert()

Tuple 2 ( , ..., ... )  $\leftarrow$  option

- IMMUTABLE ✓

- Doublovs

- index / slicing

Méthods.

Tuple  $\rightarrow$  constante

Set : { ---, ----, --- }

- MUTABLE mais dans la taille
- Unicité : pas de doublon
- pas d'indexation.
- méthodes : • add

?

3

dict : JSON

{ K:v, K:v, ... }

- MUTABLE

- Unicité sur la clé

- Indexation sur la clé

- méthodes: items()  
values()  
keys()

{ K ; V }

1 eleet

K: "contacts"

V: [ e<sub>1</sub>,

e<sub>2</sub>,

.

(

)

:

]

e<sub>1</sub>\_dict

e<sub>2</sub>  
e<sub>3</sub>

Contact 1

if: —

nor: —

;

---

Contact 2

,

,

,

,

---

product

{ description  
prix  
qte

prod-xml.

<listo>

<product>  
<description> Bonbon</description>  
<prix> 1.95</prix>  
<qte> 100</qte>  
</product>

<product>  
<description> Pain</description>  
<prix> 0.65</prix>  
<qte> 45</qte>  
</product>

</listo>

```
Prod. jsun
{
  "liste": [
    {
      "description": "Barbu",
      "prix": 1.45,
      "qte": 100
    },
    {
      "description": "Pain",
      "prix": 0.69,
      "qte": 45
    }
  ]
}
```

listo: [  $e_1, e_2, e_3 \dots,$  ]

for tmp in listo:  
print(tmp)

tmp: variable  
temporaria.

Bunch.

while i <sup>tmp.</sup> :

  ;   inst  
  ;   True  
  
→ ligne vide.

~~Do {  
} while ()~~

for tmp in iterable :

  ;   acc à b  
  ;   inst  
  
→ ligne vide

---

Java   for(int i=0; i<10; i++) {  
         }  
      ;   pyth.  
      for i in range(0, 10):  
         ;  
         ;