

- Variables simples (int, float) - Variables Collections "séquences" (Listes, String, Tuples, set, dictionnaires)

In [2]:

```
x = 1256
y = 1000
z = x ** y
print(z)
```

```
97642614840045931933566539510234405727780914277077907000105989060303225733621027156466112
56594293242077193276184749507312304064425744471555317350326782247207469296705070841890773
48037600683030231192272974091433867707063297192284678855846845608785853738196722853444761
81550434885089028380540828881710762655455265346664449005483521245916787627863788246902304
48881032006647920818431785861969774665628058215703689241822853980276840281564074633447809
37028405943956089806960897148418158298666553564815085142706821193886719821398701229630125
04055928744890207161943881425533741902788612886455400602592812914687884440716946489863431
78476198561736542531197257313354734845619973084898776542729098931589222601879424883467648
53077308834095930684214246662887735240310550268358668908422862048349580653139966471027886
64875364620245066312694493547547301756188242374579281064664444256810787941179723701362485
25431083815516416280193063469759968751817358172926449945492023878425383393217909815179538
43505985010068957876360513671376750891007437733462601216158170141999828854834050348729921
57897771431916005590887298581151305736929587499884147579926759693453000167581606147799646
51979819117442892621994517411563790383954399605212844502255588742531459506992831923673642
80509217472113529140417992957864320691087336588838934960570773309474074662412678926106366
67946157882926701011164499817200208260820375134002485227633384292796610777961449891356771
61408531825107881808437433452073845811413352407735760181559100924452381760535561816294168
70076908710852132314530851347803509289416859587511938555881497247751008283232873991351489
14811376781581005811242196172507059305438987452831676442820577290602161449441944007751993
69924915002131226066025626952070025986659901544331581871526275145246497027491662370222252
85721871150760866785447702635958094065457977105409651828555178030487317352436812251666279
16180053485979406657872560319801194105497455268931416724484220814162030838892360255512113
04306186486266881497792033766417453103439392127065020674943366464754986534800620281947542
64250025296527803369577924919895277364540454038938275250551963346235766959948453420706605
55461331926483931810751942000356446249738947439081372225526634397462057925694940558455787
82521061364500230471825055583978578995207107322073719046378263813156951595722230443513931
35102038597844964277205682872158551338089600668511391547034513209216115806592323196121523
92317403474174260265247905175230479200255804955454632760626640744289023356166973379621639
15613657970093093808683666746863530320943746028360381469060657174917413487989969205874244
44737249108607559897661403425709255945162335570313066044770740212933674077851987505245932
10865702098775181290456088980742073711235572189815159962135366082748061721064126392890774
59456134470931409271414751154077615669471615249017369846640809726127558463078624610910455
19425284544537912582525291437947059685352354662083057575739977894598620488486733848970649
25200956225703355641042906818590934840501923733850287882225799348545118789321465901385020
6523605371259675289629256225477443989734280215941108226200562030530789376
```

Tuples, Set.

A- Les tuples sont des listes immutables (non modifiables) sont délimités par (). Exemple : T1 = (1,2,6,9,11,"a")
print(T1) T1 = (6,9,10) print(T1) cette opération est possible car elle concerne une affectation et non un modification (mutabilité), la mutabilité est la modification d'une partie de la valeur. les chaines de caractères sont aussi immuable: **S = "Ceci est un essai" S = "Bonjour" print(S[0]) S[0] = "a"**

In [3]:

```
T1 = (1,2,6,9,11,"a")
print(T1)
T1 = (6,9,10)
print(T1)
```

```
(1, 2, 6, 9, 11, 'a')
(6, 9, 10)
```

Exemple : Créer un tuple qui contient les valeurs de 1000 à 10000 avec un saut de 1000 et le trier.

In [8]:

```
T1 = tuple(range(10000, 999, -1000))
print(T1)
L1 = list(T1)
```

```
L1.sort()
T1 = tuple(L1)
print(T1)
```

```
(10000, 9000, 8000, 7000, 6000, 5000, 4000, 3000, 2000, 1000)
(1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000)
```

In [11]:

```
T1 = tuple(range(10000, 999, -1000))
print(T1)
T1 = tuple(sorted(T1))
print(T1)
```

```
(10000, 9000, 8000, 7000, 6000, 5000, 4000, 3000, 2000, 1000)
(1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000)
```

Destruction (destructuring)

In [16]:

```
T1 = (16,8,9)
a,b = T1
print(b)
```

8

B - Set : une série de valeurs non dupliquées, délimités par {} Exemple : S1 = {1,2,1,2,3,16,-7,3} print(S1)

In [20]:

```
S1 = {1,2,1,2,3,16,-7,3}
print(S1)
```

```
{16, 1, 2, 3, -7}
```

Exemple 2 : consiérant la liste suivante : L1 = list(range(1,11)) L2 = list(range(5,20)) L3 = L1 + L2 le L3 contiendrait les valeurs de L1 et de L3, du coup, plusieurs valeurs seront dupliquées, utiliser set pour nettoyer les valeurs doubles.

In [23]:

```
L1 = list(range(1,11))
L2 = list(range(5,20))
L3 = list(set(L1 + L2))
print(L3)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

Travail à faire :

Développer les différentes méthodes des SETs.

In [24]:

```
S1 = set(range(1,10))
S1.
```

Dictionnaires

un dictionnaire estv un ensemble d'information concernant une occurrence (stagiaire, article,), correspond aux données structures. Exemple : nous désirons gérer les données d'un stagiaire (matricule, nom, prénom, age), les valeurs sont accédées par nom du champs et non l'indice

stagiaire = {"matricule" : 1, "nom" : "BOUJAOUANE", "pren" : "AZIZ", "age" : 18} print(stagiaire["age"]

In [28]:

```
stagiaire = {"matricule" : 1, "nom" : "BOUJAOUANE", "pren" : "AZIZ", "age" : 18}
print(stagiaire["age"])
stagiaire["age"] = 17
```

```
print(stagiaire["age"])
stagiaire["fil"] = "DEV102"
print(stagiaire)
```

```
18
17
{'matricule': 1, 'nom': 'BOUJAOUANE', 'pren': 'AZIZ', 'age': 17, 'fil': 'DEV102'}
```

Exercice : Ecrire un programme qui permet de saisir les informations pour plusieurs stagiaires.

In [32]:

```
stagiaire = []

while True :
    mat = int(input("Saisir le matricule :"))
    nom = input("Saisir le nom :")
    prenom = input("Saisir le prénom :")
    age = int(input("Saisir l'age :"))
    st = {"matricule" : mat, "nom" : nom, "pren" : prenom, "age" : age}
    stagiaire.append(st)
    choix = input("Voulez vous continuer (o/n) :")
    if choix == "n" :
        break

print(stagiaire)
```

```
Saisir le matricule :1
Saisir le nom :ALOUANE
Saisir le prénom :AMINE
Saisir l'age :18
Voulez vous continuer (o/n) :o
Saisir le matricule :2
Saisir le nom :MAARASS
Saisir le prénom :SAYF
Saisir l'age :23
Voulez vous continuer (o/n) :o
Saisir le matricule :3
Saisir le nom :SALK
Saisir le prénom :REDA
Saisir l'age :18
Voulez vous continuer (o/n) :O
Saisir le matricule :4
Saisir le nom :KARCHAOU
Saisir le prénom :MED
Saisir l'age :19
Voulez vous continuer (o/n) :n
[{'matricule': 1, 'nom': 'ALOUANE', 'pren': 'AMINE', 'age': 18}, {'matricule': 2, 'nom':
'MAARASS', 'pren': 'SAYF', 'age': 23}, {'matricule': 3, 'nom': 'SALK', 'pren': 'REDA', 'a
ge': 18}, {'matricule': 4, 'nom': 'KARCHAOU', 'pren': 'MED', 'age': 19}]
```

In [33]:

```
L1 = [1,1,9,-6, 90]

for i in range(len(L1)) :
    print(L1[i])
```

```
1
1
9
-6
90
```

In [34]:

```
for x in L1 :
    print(x)
```

```
1
1
9
-6
```

90

In []: