



Table des matières

Table des matières.....	I
1. Déclarations de conteneurs alternatives.....	I
1.1. Tuple.....	I
1.2. Listes.....	I
1.3. Dictionnaires.....	I
2. Listes avec compréhension.....	I
3. Listes à deux dimensions.....	2

1. Déclarations de conteneurs alternatives

1.1. Tuple

```
#création d'un tuple vide  
mon_nuplet = tuple()  
#équivalent  
mon_tuple=()  
#création d'un n-uplet avec un seul élément  
t1 = 'a', # la parenthèse indique que c'est un tuple
```

1.2. Listes

```
#création d'une liste vide  
ma_liste = []  
#équivalent  
ma_liste = list()
```

1.3. Dictionnaires

```
#création d'un dictionnaire vide  
mon_dictionnaire = {}  
#équivalent  
mon_dictionnaire = dict()
```

2. Listes avec compréhension



Une liste en compréhension est une expression littérale de liste équivalente à une boucle for qui construirait la même liste en utilisant la méthode append()

#listes définies en compréhension

<pre>A=[0 for i in range(5)]</pre>	<pre>A=list() for i in range(5): A.append(0)</pre>
<pre>print(A)</pre>	

```
#donne [0, 0, 0, 0, 0]
B=[x**2 for x in range(11)]
print(B)
#donne [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
C=[x for x in range(25) if x%3==0]
print(C)
#donne [0, 3, 6, 9, 12, 15, 18, 21, 24]
```

3. Listes à deux dimensions

On peut la créer de façon exhaustive :

```
D=[[0,1,2],[3,4,5],[6,7,8]]
print(D)
print("D[0][0] : {}".format(D[0][0]))
print("D[0][2] : {}".format(D[0][2]))
print("D[2][0] : {}".format(D[2][0]))
```

```
[[0, 1, 2], [3, 4, 5], [6, 7, 8]]
D[0][0] : 0
D[0][2] : 2
D[2][0] : 6
```

On peut aussi utiliser les boucles :

```
nb_lin = 3
nb_col = 3
mat = []
for i in range(nb_lin):
    row = []
    for j in range(nb_col):
        elt = int(input("Donner M {}/{}\t".format(i,j)))
        row.append(elt)
    mat.append(row)
print(mat)
```

```
Donner M 0/0 0
Donner M 0/1 1
Donner M 0/2 2
Donner M 1/0 3
Donner M 1/1 4
Donner M 1/2 5
Donner M 2/0 6
Donner M 2/1 7
Donner M 2/2 8
[[0, 1, 2], [3, 4, 5], [6, 7, 8]]
```

On peut utiliser des générateurs

```
A=[[i for i in range(3)],[3+j for j in range(3)],[6+j for j in range(3)]]
print(A)
print("A[0][0] : {}".format(A[0][0]))
print("A[0][2] : {}".format(A[0][2]))
print("A[2][0] : {}".format(A[1][0]))
```

```
[[0, 1, 2], [3, 4, 5], [6, 7, 8]]
A[0][0] : 0
A[0][2] : 2
A[2][0] : 6
```