

# Les tuples et les listes



## 1. les tuples

Un objet de type tuple (p-uplet), est une suite ordonnée de p éléments qui peuvent être chacun de n'importe quel type.

| Code Python                         | Résultat affiché dans la console |
|-------------------------------------|----------------------------------|
| a=(1,2)                             | <class 'tuple'=""></class>       |
| print (type(a))                     | •                                |
| print(a)                            | (1, 2)                           |
| a=(1,'bonjour')                     |                                  |
| b=a*2                               | (1, 'bonjour', 1, 'bonjour')     |
| print(b)                            |                                  |
| tuple_1 = (5, 2, 25, 56)            |                                  |
| tuple_2 = ("jack","tony")           | (5, 2, 25, 56, 'jack', 'tony')   |
| tuple_3 = tuple_1 + tuple_2         | (3, 2, 23, 30, jack, torry)      |
| print(tuple_3)                      |                                  |
| couleur=(255,128,64)                |                                  |
| rouge=couleur[0]                    |                                  |
| vert=couleur[1]                     | 255 128 64                       |
| bleu=couleur[2]                     |                                  |
| print(rouge,vert,bleu)              |                                  |
| tuple=('a',1,5.6,45,'toto')         | longueur du tuple 5              |
| longueur=len(tuple)                 | а                                |
| print("longueur du tuple",longueur) | 1                                |
| for i in range(0,longueur):         | 5.6                              |
| print(tuple[i])                     | 45                               |
|                                     | toto                             |
| tuple=('a',1,5.6,45,'toto')         | longueur du tuple 5              |
| longueur=len(tuple)                 | а                                |
| print("longueur du tuple",longueur) | 1                                |
| for valeur in tuple:                | 5.6                              |
| print(valeur)                       | 45                               |
|                                     | toto                             |

Les parenthèses d'un tuple ne sont pas obligatoires mais conseillées.

| Code Python                  | Résultat affiché dans la console                           |
|------------------------------|--|
| a=(1,2) # ou a=1,2<br>a[0]=0 | TypeError: 'tuple' object does not support item assignment |

Un tuple est non mutable, c'est-à-dire qu'il ne peut pas être modifié après sa création.

| Code Python | Résultat affiché dans la console |
|-------------|----------------------------------|
| a=(1,2)     |                                  |
| b=(0,a[1])  | (1, 2) (0, 2)                    |
| print(a,b)  |                                  |

#### A quoi sert un tuple?

Le tuple permet une affectation multiple.

| Code Pytl               | non        | Résultat affiché dans la console |
|-------------------------|------------|----------------------------------|
| couleur=(255,128,64)    |            |                                  |
| rouge=couleur[0]        |            |                                  |
| vert=couleur[1]         |            | 255 128 64                       |
| bleu=couleur[2]         |            |                                  |
| print(rouge,vert,bleu)  |            |                                  |
| couleur=(255,128,64)    | #emballage |                                  |
| rouge,vert,bleu=couleur | #déballage | 255 128 64                       |
| print(rouge,vert,bleu)  |            |                                  |

Un tuple permet également de renvoyer plusieurs valeurs lors d'un appel d'une fonction.

| Code Python   | Résultat affiché dans la console |
|---|----------------------------------|
| <pre>def donne_moi_ton_nom():     tuple=("Roger", "Federer")     return tuple</pre> | ('Roger', 'Federer')             |
| print(donne_moi_ton_nom())  |                                  |

Prochainement, nous verrons les tuples nommés.

https://www.youtube.com/watch?v=n54Q-lccZEs

## 2. Les listes à une dimension

Un objet de type list, que nous appelons une liste, est un ensemble ordonné d'éléments avec des indices pour les repérer. Les éléments d'une liste sont séparés par des virgules et entourés de crochets. Une liste est mutable, c'est-à-dire que l'on peut changer le contenu de la liste.

| Code Python                            | Résultat affiché dans la console |
|--|----------------------------------|
| Ist=[45,12,4,78]                       | <class 'list'=""></class>        |
| print (type(lst))                      | [45, 12, 4, 78]                  |
| print(lst)                             | [43, 12, 4, 70]                  |
| lst=[45,12,4,78]                       |                                  |
| Ist[1]=0                               | [45, <b>0</b> , 4, 78]           |
| print(lst)                             |                                  |
| lst=[45,12,4,78]                       | longueur de la liste 4           |
| longueur=len(lst)                      | 45                               |
| print("longueur de la liste",longueur) | 12                               |
| for i in range(0,longueur):            | 4                                |
| print(lst[i])                          | 78                               |
| lst=[45,12,4,78]                       | 45                               |
| for valeur in lst:                     | 12                               |
| print(valeur)                          | 4                                |
|  | 78                               |

#### Initialisation de liste

| Code Python   | Résultat affiché dans la console   |
|---|--|
| Ist=[0] *10 print(lst) Ist[2]=5 print(lst) Ist=[i for i in range(0,10)] print(lst)                      | [0, 0, 0, 0, 0, 0, 0, 0, 0]<br>[0, 0, 5, 0, 0, 0, 0, 0, 0]<br>[0, 1, 2, 3, 4, 5, 6, 7, 8, 9] |
| Ist=[i*2 for i in range(0,10)] print(Ist)   | [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]  |
| #ajout d'un élément dans la liste  lst=[i for i in range(0,10)]  print(lst)  lst.append(50)  print(lst) | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]<br>[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 50]                         |

Document ressource 1NSI - Tuples - Listes

### Accès à plusieurs données dans une liste.

| Code Python  | Résultat affiché dans la console                           |
|--|--|
| #affiche les derniers éléments en partant de l'indice 2  Ist=[i for i in range(0,10)]  print(Ist)  print(Ist[2:])      | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]<br>[2, 3, 4, 5, 6, 7, 8, 9] |
| #affiche les 3 premiers éléments  lst=[i for i in range(0,10)]  print(lst)  print(lst[:3])                             | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]<br>[0, 1, 2]                |
| #affiche le 2ème élément en partant de la fin lst=[i for i in range(0,10)] print(lst) print(lst[-2])                   | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]                             |
| #affiche les éléments entre l'indice 2 et l'indice 6 (exclu)  lst=[i for i in range(0,10)]  print(lst) print(lst[2:6]) | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]<br>[2, 3, 4, 5]             |

## Supprimer un élément

| Code Python                  | Résultat affiché dans la console |
|------------------------------|----------------------------------|
| lst=[i for i in range(0,10)] |                                  |
| print(lst)                   | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]   |
| lst.remove(5)                | [0, 1, 2, 3, 4, 6, 7, 8, 9]      |
| print(lst)                   |                                  |

### Insérer un élément

| Code Python                  | Résultat affiché dans la console   |
|------------------------------|------------------------------------|
| lst=[i for i in range(0,10)] | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]     |
| print(lst)                   | [0, 1, 2, 3, 4, 5, 6, 12, 7, 8, 9] |
| lst.insert(7,12)             | [0, 1, 2, 3, 4, 3, 0, 12, 7, 0, 3] |
| print(lst)                   | Indice 7                           |

https://www.youtube.com/watch?v=AFRdL2hge0o

### 3. Les listes à deux dimensions

| Code Python                                 | Résultat affiché dans la console             |
|---|--|
| lst=[[1,2,3],                               |  |
| [4,5,6],                                    | [[1, 2, 3], [4, 5, 6], [7, 8, 9]]            |
| [7,8,9]]                                    | [[1, 2, 3], [4, 3, 0], [7, 0, 3]]            |
| print(lst)                                  |  |
| #accès à un élément du tableau              |  |
| lst=[[1,2,3],                               |  |
| [4,5,6],                                    |  |
| [7,8,9]]                                    | 6  |
| ligne=1                                     |  |
| colonne=2                                   |  |
| print(lst[ligne][colonne])                  |  |
| #affichage sous forme de tableau            |  |
| lst=[[1,2,3],                               |  |
| [4,5,6],                                    | 123  |
| [7,8,9]]                                    | 456  |
| for ligne in range(0,3):                    | 789  |
| for colonne in range(0,3):                  |  |
| print(lst[ligne][colonne],end=' ')          |  |
| print()                                     |  |
| #initialisation d'un tableau à 2 dimensions |  |
| ligne = 3                                   |  |
| colonne = 4                                 | [[0, 0, 0], [0, 2, 0], [0, 0, 0], [0, 0, 0]] |
| lst = [[0] * ligne for i in range(colonne)] |  |
| lst[1][1]=2                                 |  |
| print(lst)                                  |  |

#### Listes et tuples

Liste → suite indexée et modifiable d'éléments de tout type Attention : l'indexation commence à 0

```
NomListe = [élément1, élément2, élément3,...]
NomListe[i] → élément d'index i (lecture ou écriture)
NomListe[ 0 ] → premier élément
NomListe [ -1] → dernier élément
```

Principales fonctions:

Longueur : len() → renvoie le nombre d'éléments Ajout: Nomliste.append $(x) \rightarrow$  ajoute x en fin de liste Insertion: Nomliste.insert(i, x)  $\rightarrow$  insert  $x \ge l$ 'index i Suppression: Nomliste.pop() →supprime le dernier élémt Nomliste.pop(i) → supprime élément d'indexe i

- Tuples → un tuple est une liste non modifiable NomTuple = (élément1, élément2, élément3,..) NomTuple[i] → élément d'indexe i en lecture seule
- Liste de listes → une liste peut contenir des listes !! NomListe[i][j] désigne l'élément d'index j de la liste d'index i

```
L=[5,8,"Julie"] # Liste de 3 éléments
a=L[0] # lecture: a vaut 5
L[1] =10 # Ecriture: L vaut [5,10,"Julie"]
b=L[-1] # b vaut 'Julie'
x= len(L) # x vaut 3
c=L[3] # erreur dépassement
L.append(3) # L vaut [5,10, 'Julie',3]
L.insert(2,"P") # L vaut [5,10,'P','Julie',3]
L.pop() # L vaut [5,10,'P', Julie']
L.pop(2) # L vaut [5,10, 'Julie']
```

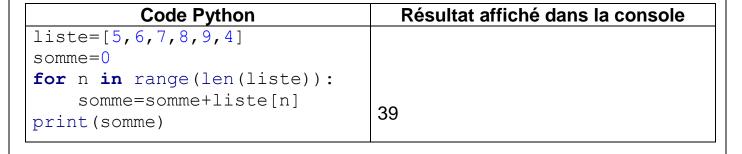
```
T=(4,8,10)
x=T[0] # x vaut 4
T[0]=5 # erreur, T non modifiable
```

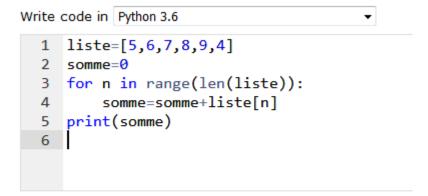
```
Tab=[[4,2,9],[0,7,8]]
x=Tab[0] # x vaut [4,2,1] : index 0 de Tab
y=Tab[0][2] # y vaut 9 : index 2 de Tab[0]
```

#### **Annexe**

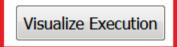
#### Visualiser un code pas à pas graphiquement

#### http://pythontutor.com/visualize.html#mode=edit



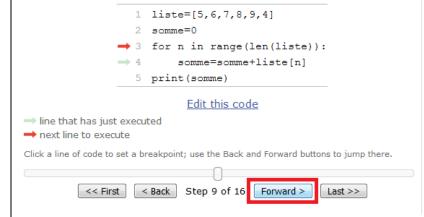


Help improve this tool by completing a short user survey



Live Programming Mode

## Déroulement de l'exécution pas à pas



Python 3.6

