

Les tuples

Plan

1. Création d'un t-uple
2. Afficher les valeurs d'un t-uple
3. La taille d'un t-uple
4. Les tranches dans un t-uple
5. Ajouter un élément à un tuple
6. Vérifier l'existence d'une valeur dans
7. Parcourir un t-uple
8. Convertir un tuple en liste
9. Convertir une liste en tuple

Les tuples: définition

- Un **tuple**, permet de créer une **collection ordonnée de plusieurs éléments**. En mathématiques, on parle de **p-uplet**.
- Par exemple, un couple est constitué de 2 éléments, un triplet est constitué de 3 éléments,....
- Exemple :
$$a = (3, 4, 7)$$
- **Rq**: Les parenthèses ne sont pas obligatoires.

Les tuples: création

- Autre notation :

```
b, c = 5, 6  
print("b=", b)  
print("c =", c)
```

```
>>  
b=5  
c=6
```

- Cette écriture est équivalente à :

```
(b, c) = (5, 6)
```

Les tuples: création

- On peut également avoir une seule variable à gauche qui reçoit un tuple à droite du signe "=".

```
a = (3, 4)
u, v = a
print("u =", u)
print("v =", v)
```

```
>>
u=3
v=4
```

Afficher une valeur d'un tuple

- Le tuple est une sorte de liste, on peut donc utiliser les crochets `[]` pour lire les données du tuple.

```
a = (3, 4, 7)  
print(a[1])
```

```
>>  
4
```

La longueur d'un tuple

- Comme une liste, on peut récupérer la longueur d'un tuple avec la fonction `len()`

```
a = (3, 4, 7)
L=len(a)
print("la longueur de tuple est :", L)
```

- Le programme va afficher la valeur 3

Les tranches dans les tuples

- On peut utiliser les tranches dans les tuples de même que pour les listes:

```
a=(1,6,9,8,90,11)  
print(a[1:4])
```

```
>>  
(6,9,8)
```


Ajouter un élément à une liste

- Contrairement à une liste, un tuple est une séquence d'élément(s) qui ne peut pas être modifiée directement. Par contre il est possible de concaténer plusieurs tuples (c.a.d créer un nouveau tuple)

```
t1=("mouhamed",20,"oujda")  
t2=t1 + ("salim",)  
print(t2)  
  
#resultat  
( 'mouhamed', 20, 'oujda', 'salim')
```

- **Note:** attention à ne pas oublier la virgule ("salim",) sinon c'est un string et non pas un tuple !

Vérifier l'existence d'une valeur dans un tuple

- On peut faire cette vérification comme suit:

```
T=(14,5,8,3)

a=int(input("Saisir une valeur à chercher"))
if a in T:
    print("Le nombre existe")
else :
    print("Le nombre n'existe pas ")
```

Parcourir un tuple

- On peut parcourir un tuple comme une liste:

```
a = (3, 4, 7)
for c in a:
    print(c)
```

```
a = (3, 4, 7)
for i in range(len(a)):
    print(a[i])
```

Convertir un tuple en liste

- Pour convertir un tuple en python, on utilise la fonction **list()** déjà vu.
- Exemple:

```
t1=("mouhamed",20,"oujda")  
l=list(t1)  
print(l)
```

```
#résultat  
['mouhamed', 20, 'oujda']
```

Convertir une liste en tuple

- On utilise la fonction **tuple()** pour convertir une liste en tuple.
- Exemple:

```
L=[14,5,87,30]  
T=tuple(L)  
print(T)
```

```
#résultat  
(14, 5, 87, 30)
```

Fonction qui retourne une valeur composée

- Une fonction ne peut retourner qu'une donnée à la fois. Cependant, Python met à notre disposition des types de données composites comme les **listes** ou les **tuples** par exemple.
- On va donc pouvoir utiliser `return` pour faire retourner "plusieurs valeurs" à la fois

Fonction qui retourne une valeur composée

- Exemple de fonction qui retourne un tuple:

- Définition**

```
def ordre(a,b):  
    if a<b :  
        return (a,b)  
    else:  
        return (b,a)
```

- Appel:**

```
ordonne=ordre(12,5)  
print(ordonne)
```

- Résultat:**

(2,5)