

1 Définition : p-uplet

Un **p-uplet** est une **séquence correspondant à une collection de données** (d'un type de base par exemple), souvent noté entre parenthèses (comme en Python).

Exemple : `p_uplet = (12, 2.0, -56, 41.32)`

On accède à un élément d'un p-uplet par son **indice** qui donne sa position dans la séquence.

Exemple : `element_3 = p_uplet[2] # (nombre -56 avec le précédent exemple)`

À la différence des listes (tableaux), ces objets sont **non mutables** : on ne peut pas les modifier après leur création.

2 Exemple de fonction qui renvoie un p-uplet

La fonction suivante renvoie la liste 5 premières puissances d'un nombre n .

```
fonction puissances5(n) ;  
     $a \leftarrow n^2$  ;  
     $b \leftarrow n^3$  ;  
     $c \leftarrow n^4$  ;  
     $d \leftarrow n^5$  ;  
    retourner (n, a, b, c, d)
```

L'implémentation en Python est très simple :

```
def puissances5(n) :  
    a = n ** 2  
    b = n ** 3  
    c = n ** 4  
    d = n ** 5  
    return (n, a, b, c, d)
```

Remarque : en Python, un p-uplet s'appelle un **tuple**.

Un tuple est un objet **immuable** : il ne peut être modifié après sa création.

3 Définitions : p-uplet nommés

Un **p-uplet nommé** est un p-uplet pour lequel **chaque élément est identifié par un nom** plutôt que par son indice.

Remarque : en Python, un p-uplet nommé est un **dictionnaire**.

4 Dictionnaires en Python

Un dictionnaire est une **collection d'éléments identifiés par une clef à laquelle correspond une valeur** pour l'élément.

Comme on l'a déjà dit, en Python, c'est l'implémentation des p-uplets nommés.

Une différence avec les tuples, est la possibilité de les modifier après leur création : ce sont des objets **mutables**.

Création d'un dictionnaire :

```
dico = {'henri':(4,'roi') , 'louis':(16, 'roi')}
```

Ajout d'un élément :

```
dico['élisabeth'] = (1,'reine')
```

Modification d'un élément :

```
dico['élisabeth'] = (2,'reine')
```

5 Itérations sur les dictionnaires

Il est possible de parcourir les éléments complets d'un dictionnaire (ou seulement les clefs, ou seulement les valeurs).

Pour itérer sur les éléments complets :

```
for clef, valeur in dico.items():  
    print(f"La valeur {valeur} est associée à la clef {clef}")
```

Pour itérer sur les clefs du dictionnaire :

```
for clef in dico.keys():  
    print(clef)
```

Pour itérer sur les valeurs du dictionnaire :

```
for valeur in dico.values():  
    print(valeur)
```