

## 1 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.

Les métadonnées EXIF d'une photographie numérique pourraient par exemple être représentées par un p-uplet nommé (présentées ici au format JSON) :

```
exif_data = {  
    "Version Exif" : "Exif version 2.1"  
    "Date" : "2003:08:11"  
    "Compression" : "JPEG"  
    "Espace colorimétrique" : "sRGB"  
    "Dimension X" : 2240  
    "Dimension Y" : 1680  
}
```

On accède alors à une donnée directement en utilisant une clef plutôt qu'un indice.

La 'dimension X' de la photo précédente s'obtient ainsi : `exif_data["Dimension X"]`

On aurait pu représenté les données EXIF par un simple p-uplet mais son utilisation aurait été moins agréable.

Exemple en Python :

```
exif_data = ("Exif version 2.1", "2003:08:11", "JPEG", "sRGB", 2240, 1680)
```

Avec cette représentation, la 'dimension X' de la photo précédente s'obtient ainsi : `exif_data[4]`

Ça fonctionne, mais c'est moins lisible.

## 2 Dictionnaires en Python

En Python, un p-uplet nommé est implémenté par un **dictionnaire**.

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

Une différence importante 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')`

*Remarque* : les **clefs** des dictionnaires doivent être des objets **non mutables** (ex : tuples, chaînes de caractères, entiers). Contre-exemple : une liste ne peut pas être une clef.

### 3 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 (couples clef/valeur) :

```
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)
```