

## Les Uplets (Tuple) :

Collection d'objets qui peuvent être distincts non mutable

### Déclaration :

---

```
>>> u = 18, "salut", '4', ['a', 3]
>>> u[1]                                # renvoi "salut"
```

On peut itérer sur un uplet

```
>>> type(u[3])                          # renvoi "list"
>>> u[3][0]                             # renvoi 'a'
```

## Les ensembles (set):

Collection non ordonnée d'objet non mutable (mais on peut modifier l'ensemble) de types distincts. Tous les Objets de l'ensemble sont distincts.

### Déclaration :

---

```
>>> s = {1, 2, 5, 3, 7}
>>> s = set()
>>> s                                    # renvoi set()

>>> s2 = set(['a', 4, 18, 'bonjour'])
>>> s2                                  # {'a', 4, 18, 'bonjour'}

>>> s3 = set([(1, 2, 3)])
>>> s3                                  # {(1,2,3)}
```

On peut modifier un ensemble :

```
>>> s4.add(4)
>>> s4.remove(4)

>>> 2 in s4                             # True
>>> 'bonjour' not in s4                 # True
```

On peut réaliser

## **l'union de deux ensemble : union, |**

```
>>> s = s1.union(s2)
>>> s = s1 | s2
```

## **l'intersection : intersection, &**

```
>>> s = s1.intersection(s2)
>>> s = s1 & s2
```

## **la difference: difference, -**

```
>>> s = s1.difference(s2)
>>> s = s1 - s2
```

## **la difference symetrique (ou éclusif): difference, -**

```
>>> s = s1.symmetric_difference(s2)
>>> s = s1 ^ s2
```

# **Les dictionnaires (dictionary) :**

Un dico est un ensemble de couple clé, valeur où :

- les clés sont des types non mutable
- les valeur sont de n'importe quel type

## **Déclaration:**

---

```
>>> dico = {'ensemble': {'set', 'together'}, 'uplet': 'tuple'}
>>> dico = dict()
>>> dico = {}

>>> dico = dict(zip(['ensemble', 'uplet'], [{'set', 'together'}, 'tuple']))
>>> dico           # {'uplet': 'tuple', 'ensemble': {'set', 'together'}}

>>> dico = {c:v for (c,v) in zip(range(4), ['a','b','c','d'])}
>>> dico           # {0: 'a', 1: 'b', 2: 'c', 3: 'd'}
```

## **Attention à la taille des listes**

```
>>> dico = {c:v for (c,v) in zip(range(5), ['a','b','c','d'])}
>>> dico          # {0: 'a', 1: 'b', 2: 'c', 3: 'd'}

>>> dico = {c:v for (c,v) in zip(range(4), ['a','b','c','d','e'])}
>>> dico          # {0: 'a', 1: 'b', 2: 'c', 3: 'd'}
```

## Acces et modification :

---

```
>>> dico ['uplet']          # tuple
>>> dico ['liste'] = 'list'  # ajout le couple liste:list
>>> dico ['ensemble'] = 'ens'      #modifie l'entrée ensemble
```

## Les vues (views) :

---

la vue dict\_keys est un objet faisant référence aux clés d'un dictionnaire

```
>>> dico.keys()          # objet dict_keys('ensemble', 'uplet','liste')
```

la vue dict\_values est un objet faisant référence aux valeurs d'un dictionnaire

```
>>> dico.values()        # objet dict_values('ens','tuple','list')
```

la vue dict\_items est un objet faisant référence aux couples clés,valeurs d'un dictionnaire

```
>>> dico.items()         # objet dict_items(('ensemble','ens'),('uplet','tuple'),('liste','list'))
```

Les opérateurs ensemblistes `|,&,-,^` s'applique aussi aux vues

***les vues sont des objets itérables. De plus si on modifie le dictionnaire correspondant, les vues sont modifiées et inversement.***