

# PYTHON : FICHE RÉCAP

## LE SET

Le set fait partie des collections et utilise des 'tables de hachage'. Il permet de rendre uniques les éléments en supprimant les doublons.

Attention : Le set est énumérable (on peut récupérer tous les éléments) mais pas indexable (il n'y a pas de notion d'ordre dans le set)

### Initialiser un set

```
set_noms = { "Marie", "Paul", "Jean", "Marc", "Emilie", "Marie" }
```

# On utilise la syntaxe avec les accolades comme pour un dictionnaire. Mais ici il n'y a pas d'association avec une valeur. Les valeurs sont en réalité des clefs.

# Ici Marie apparait deux fois, mais le résultat du print affiche Marie qu'une seule fois:

```
print(set_noms)
```

```
{'Marc', 'Paul', 'Jean', 'Emilie', 'Marie'}
```

### Initialisation à partir d'une liste

# Il est également possible d'initialiser un set à partir d'une liste

```
noms = ["Marie", "Paul", "Jean", "Marc", "Emilie", "Marie"]
```

```
set_noms = set(noms)
```

### Enumération

# Il est possible de récupérer toutes les valeurs contenues dans un set. Cependant l'ordre n'est pas respecté

```
for s in set_noms:  
    print(s)
```

## Indexation

# Il n'est pas possible d'indexer in set. Par exemple 'print(set\_noms[0])' va créer une exception.

Pour contourner ce problème, il est possible de convertir le set en une liste :

```
noms = ["Marie", "Paul", "Jean", "Marc", "Emilie", "Marie"]  
noms_sans_doublons = list(set(noms))
```

```
print(noms_sans_doublons[0])
```