



$(1, 2) \neq (2, 1)$

Tuples

Un tuple permet de **regrouper des valeurs** pour en former **une** nouvelle.

Syntaxe : Les éléments d'un tuple sont encadrés entre 2 parenthèses ().

Particularités

 **Immuable**
 **Ordonné**
 **Indexé**
 **Doublon**

Ex : Individu = (« pierre », « sarah »)
 métier = (« ingénieur »)

01



Listes

Une liste permet de stocker en séquence ordonnée plusieurs valeurs.

Syntaxe : Les éléments d'une liste sont encadrés entre 2 crochets [].

Particularités

 **Modifiable**
 **Ordonné**
 **Indexé**
 **Doublon**

Ex : Individu = [« pierre », « sarah »]
 métier = [« ingénieur »]

02



Ensembles

Les ensembles permettent de stocker plusieurs valeurs distinctes.

Syntaxe : Les éléments d'un ensemble sont encadrés entre 2 accolades { }.

Particularités

 **Modifiable**
 **Non Ordonné**
 **Non indexé**
 **Unique**

Ex : A = {« pierre », « sarah »}
 B = {« pierre », « paul »}

03




Dictionnaires

Le dictionnaire est un tableau associatif, il permet de stocker des valeurs et de les retrouver à partir de clefs.

Syntaxe : Les éléments d'un dictionnaire sont encadrés entre deux accolades { }.

Particularités

 **Modifiable**
 **Non Ordonné**
 **indexé**
 **Doublon**

Une clé est unique est immuable

Ex : Personne = {^{Cle}'Prénom' : ^{Valeur}'Pierre',
 'Métier' : 'Ingénieur'}

04

RAPPEL

Opérations uniquement possibles sur des ensembles !

Intersection (ET)

>>> A & B
 {« pierre »}



Union (OU)

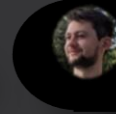
>>> A | B
 {« pierre », « sarah », « paul »}



Différence symétrique ('OU' exclusif)

>>> A ^ B
 {« sarah », « paul »}





INDEXATION

Listes et les tuples

ville = ['Grenoble', 'Paris', 'Lyon']

Élément	Grenoble	Paris	Lyon
Index	0	1	2

Récupération d'un élément avec la **variable contenant la liste** ou le **tuple** suivi son **index** entre crochets [].

```
>>> ville[0]
Grenoble
```

Dictionnaires

fruits = {'poire' : 3, 'pomme' : 5, 'fraise' : 20}

Récupération de valeurs :

```
>>> fruits['poire']
3
```

MODIFICATION

fruits = {'poire' : 3, 'pomme' : 5, 'fraise' : 20}

Ajout



```
>>> fruits['raisin'] = 4
{'poire' : 3, 'pomme' : 5,
'fraise' : 20, 'raisin' : 4}
```

Modification de valeurs

```
>>> fruits['pomme'] = 31
{'poire' : 3, 'pomme' : 31,
'fraise' : 20, 'raisin' : 4}
```

Suppression d'éléments



```
>>> del fruits['fraise']
{'poire' : 3, 'pomme' : 31,
'raisin' : 4}
```

DEGROUPEMENT ET PERMUTATION

grenoble = ('Grenoble', (45, 5))

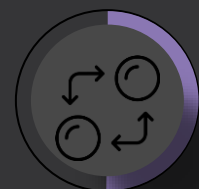


Dégroupage

```
>>> ville, (x, y) = grenoble
>>> x
45
```

Récupération des éléments du tuple dans plusieurs variables

Permutation



```
>>> x, y = y, x
>>> x, y
(5, 45)
```

Création en arrière plan d'une variable temporaire pour permettre la permutation