

# Cheat Sheet – Tuples Python

## Principes

- Structure ordonnée et immuable
  - Structure ordonnée : les éléments ont un ordre précis (une position). Le premier, le deuxième, etc. comptent, et cet ordre est conservé.
  - Structure immuable : une fois créée, la structure ne peut pas être modifiée. On ne peut ni changer, ni ajouter, ni supprimer un élément ; il faut en créer une nouvelle si on veut autre chose.
- Plus sûre pour données constantes

## Syntaxe

- `t = (1, 2)`
- `t = 1, 2`
- `(1,)` pour un singleton

## Usages

- Retour multiple de fonction
- Clés de dictionnaire
- Coordonnées

## Exercice guidé – Comprendre les tuples en Python

### Objectif

Comprendre ce qu'est un tuple, quand l'utiliser, et pourquoi il est immuable, en le comparant à une liste dans un contexte concret.

### Mise en situation (réseau / IT)

Tu récupères des informations fixes sur un équipement réseau :

- nom de l'équipement
- adresse IP
- type d'équipement

Ces informations ne doivent pas être modifiées par erreur dans le script.

### Étape 1 – Créer un tuple

Copie et exécute ce code :

```
equipement = ("SW-CORE-01", "192.168.10.1", "switch")  
print (equipement)
```

#### Observation attendue

Les données sont regroupées

La syntaxe utilise des parenthèses

### Étape 2 – Accéder aux éléments (structure ordonnée)

Copie et exécute ce code :

```
print ("Nom :", equipement[0])  
print ("IP :", equipement[1])  
print ("Type :", equipement[2])
```

#### À comprendre

- Le tuple est ordonné
- Chaque élément a une position (index)
- On accède comme une liste

### Étape 3 – Tester l'immutabilité (volontairement provoquer une erreur)

Copie et exécute ce code :

```
equipement[1] = "192.168.10.2"
```

#### Résultat attendu

Une erreur apparaît (TypeError)

## Cheat Sheet – Tuples Python

### À comprendre

Un tuple protège les données contre les modifications accidentnelles.

### Étape 4 – Comparaison avec une liste

Copie et exécute ce code :

```
equipement_liste = ["SW-CORE-01", "192.168.10.1", "switch"]
equipement_liste[1] = "192.168.10.2"
print (equipement_liste)
```

### Constat

La liste est modifiable

Le tuple ne l'est pas

### Étape 5 – Cas d'usage concret

Lis ce code :

```
for info in equipement:
    print (info)
```

### Question

Est-ce que l'ordre d'affichage est respecté ?

Pourquoi est-ce important pour des données techniques ?

### Synthèse

Complète les phrases :

Un tuple est une structure \_\_\_\_\_ et \_\_\_\_\_.

On utilise un tuple quand les données \_\_\_\_\_.

On préfère une liste quand les données \_\_\_\_\_.

## Correction attendue (formateur)

1. ordonnée et immuable
2. ne doivent pas changer
3. doivent être modifiées

## Message clé à retenir

Tuple = données fixes et protégées

Liste = données modifiables