

Cours : Comprendre les *tuples* en Python

Introduction

Un **tuple** (type tuple en Python) est une **séquence ordonnée d'éléments**, comme une liste.

Mais contrairement à une liste, un tuple est *immutable* (non modifiable).

Exemples :

```
coord = (3, 5)
infos = ("Alice", 25, "Paris")
mixte = (1, "deux", 3.0, True)
```

Accès aux éléments

Comme pour les listes (cf. notes[0] ou notes[-1]) :

```
coord = (3, 5, 7)
print(coord[0]) # 3
print(coord[-1]) # 7
```

On peut lire les éléments... mais pas les modifier !

Différence clé avec les listes

Type	Ordonné	Modifiable (mutable)	Syntaxe
tuple	Oui	✗ Non	(a, b, c)
list	Oui	✓ Oui	[a, b, c]

Exemple :

```
t = (1, 2, 3)
l = [1, 2, 3]

l[0] = 10 # OK 🎉
t[0] = 10 # ✗ Erreur : TypeError
```

⚠ Mutabilité en détail

- Un tuple ne peut pas être modifié après sa création :

```
t = (1, 2, 3)
t.append(4) # ❌ Impossible
```

- Mais si un élément à l'intérieur est mutable, il peut changer :

```
t = (1, [2, 3], 4)
t[1].append(5) # ✅ Autorisé !
print(t) # (1, [2, 3, 5], 4)
```

❖ Quand utiliser un tuple ?

- ✓ Données fixes (coordonnées, date...) ✓ Pour éviter les modifications accidentnelles

Conversion tuple ↔ liste

```
t = (1, 2, 3)
l = list(t) # tuple → liste
l[0] = 10
t2 = tuple(l) # liste → tuple
```

Exemple utile

```
personne = ("Alice", 25, "Paris")
age = personne[1]
```

Impossible de faire :

```
personne[1] = 30 # ❌ TypeError
```

Il faut créer un nouveau tuple :

```
personne = ("Alice", 30, "Paris")
```