


Cours	Cours	
BTS SNIR	Python	
2 <sup>ème</sup> année	Les tuples	

Dans cette leçon, nous allons nous intéresser à un nouveau type de données composites Python qui peut faire penser aux listes à première vue : les tuples. Nous allons souligner les différences entre les listes et les tuples, voir les cas d'usage des tuples et apprendre à créer des tuples.

## Présentation des tuples et cas d'utilisation

Les chaînes de caractères et les listes sont deux types séquentiels de données : ce sont des données qui sont organisées sous la forme de séquence de caractères ou de valeurs. Les tuples sont un autre type séquentiel de données.

Les tuples ressemblent aux listes : un tuple consiste en différentes valeurs entourées par des virgules. Notez qu'on encadre généralement les valeurs d'un tuple avec un couple de parenthèses même si cela n'est pas obligatoire.

Les tuples peuvent contenir différents types de valeurs comme des nombres, des chaînes, des listes etc. et même d'autres tuples imbriqués. Illustrons immédiatement cela :

```
>>> t1 = 1, 3, 5
>>> t2 = "Pierre", 29
>>> t3 = (12, 34, "Trail")
>>> t4 = ("Pierre", 29, ["Trail", "Triathlon"])
>>> t5 = ("Pierre", 29, ("trail", "triathlon"))
>>>
>>> t2[0]
'Pierre'
>>> t2[1]
29
>>> t4[2]
['Trail', 'Triathlon']
>>> t5[2]
('trail', 'triathlon')
```

La grande différence entre un tuple et une liste est qu'un tuple est une donnée immuable à la différence d'une liste qui est altérable. Cela signifie qu'on ne va pas pouvoir modifier les valeurs d'un tuple après sa création.

Il va donc être intéressant d'utiliser des tuples plutôt que des listes dans les cas où on veut s'assurer que les données ne soient pas modifiées dans un programme.

## Création d'un tuple vide ou à une valeur

Notez que dans le cas où on souhaite créer un tuple vide, on utilisera une paire de parenthèses vides. Si on souhaite créer un tuple avec une seule valeur, alors il faudra faire suivre cette valeur d'une virgule.

```
[>>> vide = ()
[>>> unique = ("Pierre",)
[>>> unique[0]
'Pierre'
```

## Le déballage de séquence

Une fonctionnalité intéressante des tuples est le “déballage de séquence”. Un déballage de séquence correspond à une façon rapide d'affecter les différentes valeurs d'un tuple dans des variables séparées. Pour cela, il va nous falloir autant de variables qu'il y a de valeurs dans notre tuple. Ensuite, il va falloir respecter la syntaxe `variable1, variable2 ... = tuple` comme ceci :

```
[>>> t = ("Pierre", 29, ["Trail", "Triathlon"])
[>>> nom, age, sport = t
[>>>
[>>> nom
'Pierre'
[>>> age
29
[>>> sport
['Trail', 'Triathlon']
[>>> t
('Pierre', 29, ['Trail', 'Triathlon'])
```

Attention ici : il faut bien faire attention à écrire les variables qui vont recevoir les valeurs du tuple avant le tuple car dans le cas contraire cela ne fonctionnerait pas.