

Algorithmique et Programmation

Tuples

Tuples (immutable list)

Un tuple est une séquence non modifiable de valeurs. C'est une séquence comme les listes, la seule différence, c'est qu'il est non modifiable (immutable).

Création d'un tuple

On peut créer un tuple facilement par insertion de valeurs en deux parenthèses séparées par virgules.

| |
|------------------------------------------------------------|
| Tupl1=("physics", "chemistry", 2015) |
| Tupl2 = (1,2,3,4,6,50) |
| Tupl3 = (12,) # n'oubliez pas la virgule après la valeur. |
| Tupl4 = () # tuple vide à partir de la version 3 de Python |

Accès aux valeurs d'un tuple

Comme toutes les autres séquences pour accéder aux valeurs d'un tuple, vous devez utiliser les crochets :

| | | | |
|------------------------------------------|-------------|---------|---------|
| T = ("physics", "chemistry", 1999, 2015) | | | |
| T [0] | T [1] | T [2] | T [3] |
| 'physics' | 'chemistry' | 1999 | 2015 |

Accès aux sous-tuples d'un tuple (tranches)

Comme toutes les autres séquences pour accéder aux sous-tuples d'un tuple, vous devez utiliser deux indices séparés par « : »

| | | | |
|------------------------------------------|-------------|-----------|-------------|
| T = ("physics", "chemistry", 1999, 2015) | | | |
| T [0 :1] | T [2 :] | T [-1:] | T [-1:-1] |
| ('physics',) | (1999,2015) | (2015,) | () |

Tuples sont non-modifiables « immutables »

Comme les chaînes de caractères « str », les tuples sont non-modifiables donc, vous êtes interdits de modifier les valeurs et sous-tuples d'un tuple, mais vous pouvez utiliser ces valeurs et sous-tuples pour créer des variations de votre tuple initiale.

| | | |
|----------------------------------------------------|---|------------------------|
| (12, 'Hello', 13.1452) | + | (14, -1j+15, 'Python') |
| (12, 'Hello' , 13.1452 , 14 , -1j+15 , 'Python') | | |

Suppression des éléments d'un tuple

Les tuples sont non-modifiables, donc la suppression des valeurs et sous-tuples d'un tuple n'est encore pas possible. Ce qu'on peut faire c'est supprimer tout le tuple.

```
tup = ("physics", "chemistry", 1999, 2015)
```

```
del tup
```

« **del** » supprimera tout le tuple, après cette operation la variable « tup » n'est plus definie.

Operations de bases sur les tuples

Les tuples utilisent les operateurs « + » et « * » juste comme les chaines de caractères « str ».

| Expression | Résultat | Description |
|----------------------------------------------|---------------------------------------|---------------------------------|
| <code>len((1,2,3))</code> | 3 | Nombre de valeurs dans un tuple |
| <code>(1,2,3)+(4,5,6)</code> | <code>(1,2,3,4,5,6)</code> | Concaténation |
| <code>('Hi',)*4</code> | <code>('Hi', 'Hi', 'Hi', 'Hi')</code> | Répétition |
| <code>3 in (1,2,3)</code> | True | Teste d'appartenance |
| <code>for x in (1,2,3) : print(x)</code> | 1 2 3 | Iteration par boucle for |

Délimiteurs ne sont pas nécessaires

Toute séquence de valeurs séparées par virgule qui n'est pas délimiter par des crochets, des parenthèses ou bien des accolades est par défaut un tuple.

```
T = 'abc', 'hello', 12.35, -1, False, 18+0.5j
```

Manipulation des tuples (fonctions)

Python inclut des fonctions de manipulation des tuples :

| | |
|-------------------------|----------------------------------------------|
| <code>len(tuple)</code> | Retourne le Nombre des valeurs dans un tuple |
| <code>max(tuple)</code> | Retourne la plus grande valeur dans un tuple |
| <code>min(tuple)</code> | Retourne la plus petite valeur dans un tuple |