

Initiation à Python

Premiers pas

Tuple

LIST

Laboratoire d'Informatique, Système et Télécommunications

Cours du Pr. Abdellah AZMANI
Département Génie Informatique
Cours pour les étudiants des Masters
et des Cycles d'Ingénieurs



Université Abdelmalek Essaadi
Faculté des Sciences et Techniques
Tanger



Cours du Pr. Abdellah AZMANI

PYTHON

Premiers pas

Tuple

Qu'est ce qu'un tuple ?

Définition et Syntaxe

Un *tuple* (*uplet*) est une liste statique non modifiable.
Il sert si nous voulons définir des sortes de constantes.

un **tuple** est une collection d'éléments séparés par des virgules.

Pour initialiser tuple, on utilise « (« et «) »

```
>>> tuple = ()
```

Pour créer un tuple avec une seule valeur :

```
>>> tuple = ("voila", )
```

Attention : la virgule en fin de tuple dans cas est obligatoire

Pour créer un **tuple** avec plusieurs valeurs,

```
>>> tuplo = 'a', 'b', 'c', 'd', 'e'  
>>> print(tuplo)  
( 'a', 'b', 'c', 'd', 'e')
```

On remarque que la fonction print() a enfermé le tuple entre parenthèse. Aussi, pour une meilleure lisibilité, il est conseillé d'utiliser les parenthèse pour écrire ou représenter un tuple :

```
>>> tuplo = ('a', 'b', 'c', 'd', 'e')
```

PYTHON

Premiers pas

Tuple

Affichage

Afficher les valeurs d'un tuple

Pour manipuler les tuples, on procède comme pour les listes et les chaînes de caractères.

```
>>> tuplo = ('a', 'b', 'c', 'd', 'e')
```

Pour afficher une valeur :

```
>>> tuplo[0]
```

```
'a'
```

```
>>> tuplo[2]
```

```
'c'
```

Pour afficher toutes les valeurs du **tuplo**, nous écrirons :

```
>>> for T in tuplo:
```

```
...     print (T)
```

```
...
```

```
'a'
```

```
'b'
```

```
'c'
```

```
'd'
```

```
'e'
```

Comme pour les listes et les chaînes de caractères, on peut utiliser l'instruction « **in** » pour vérifier si un élément se trouve dans un tuple.

PYTHON

Premiers pas

Tuple Affichage

Afficher les valeurs d'un tuple

```
>>> for T in tuplo:  
...     print (T, end=" ->")  
...  
a->b->c->d->e->
```

En ajoutant « end » suivi d'un séparateur, Python comprend qu'il ne faut pas passer à la ligne et qu'il faut séparer les valeur écrite par print() par l'indicateur spécifié représenté ici par « -> »

PYTHON

Premiers pas

Tuple Opérations

Opérations sur les tuples

Python propose des opérations similaires à celles des listes sauf que les tuples ne sont pas modifiables :

```
>>> print(tup[2:4])  
( 'c', 'd' )
```

```
>>> tup[1:3] = ('x', 'y') # ==> ***** erreur ! *****
```

Erreur :

TypeError: 'tuple' object does not support item assignment

```
>>> tup = ('André',) + tup[1:]  
>>> print(tup)  
( 'André', 'b', 'c', 'd', 'e' )
```

PYTHON

Premiers pas

Tuple Opérations

Opérations sur les tuples

Le tuple permet une affectation multiple

```
>>> T1, T2 = ('a'), (100,)
```

```
>>> T1
```

```
'a'
```

```
>>> T2
```

```
100
```

Il permet également de renvoyer plusieurs valeurs lors d'un appel d'une fonction:

```
>>> def retourn_1_2 ():
```

```
...     return (1, 2)
```

```
#cette fonction retourne un tuple ayant comme valeur 1 et 2
```

```
...
```

```
>>> retourn_1_2 ()
```

```
(1, 2)
```

PYTHON

Premiers pas

Tuple Opérations

Opérations sur les tuples

Affectation multiple

```
>>> tup1, tup2 = ('a', 'b'), ('c', 'd', 'e')
```

Concaténation

```
>>> tup3=tup1 + tup2
```

```
>>> tup3
```

```
('a', 'b', 'c', 'd', 'e')
```

Duplication multiple

```
>>> tup4 = tup1*4
```

```
>>> tup4
```

```
('a', 'b', 'a', 'b', 'a', 'b', 'a', 'b')
```

Nombre d'élément

```
>>> len(tup4)
```

```
8
```

PYTHON

Premiers pas

Tuple Opérations

Opérations sur les tuples

Interdiction de supprimer un élément d'un tuple

On souhaite supprimer un élément d'un tuple par del()

```
>>> del tup4[2]
```

Python indique une Erreur

TypeError: 'tuple' object doesn't support item deletion

On aboutit au même résultat avec remove()

```
>>> remove tup4[2]
```

Python indique une Erreur

TypeError: 'tuple' object doesn't support item deletion

Quel est donc l'intérêt des tuples ?

Si il faut que les données transmises ne soient pas modifiées par erreur au sein d'un programme, faut utiliser les tuples.