





01

DÉFINITION

Qu'est ce qu'un tuple?

EXEMPLES D'UTILISATION

Comment manipuler
un tuple?

02

03

OPÉRATEURS

Quels sont les différents
opérateurs de tuples?

FONCTIONS

Quelles sont les
fonctions de tuples?

04





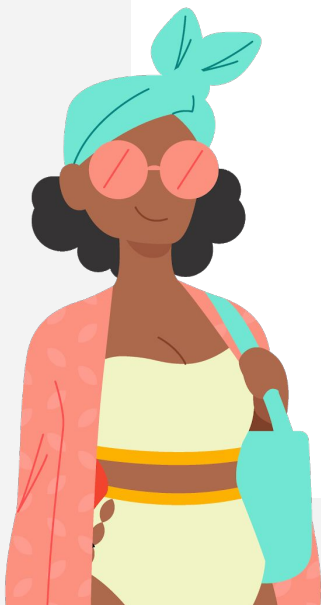
DÉFINITION

Les **tuples** sont des **séquences**, tout comme les listes. Les différences entre les tuples et les listes sont que les **tuples ne peuvent pas être modifiés** et ils utilisent des **parenthèses**.

Pour créer un tuple il suffit de donner **différentes valeurs séparées par des virgules**. Vous pouvez également placer ces valeurs **séparées par des virgules entre parenthèses**. Le tuple vide s'écrit avec deux parenthèses ne contenant rien.

Pour écrire un tuple contenant **une seule valeur**, vous devez **inclure une virgule**, même s'il n'y a qu'une seule valeur.

Comme les indices de chaînes de caractères, **les indexes commencent à 0**, et ils peuvent être découpés, concaténés, etc... tout comme les listes.





EXEMPLES D'UTILISATION

Voici quelques exemples classiques de manipulation de tuples:

- **Création** : Les éléments sont séparés par des virgules, entre ou sans parenthèses.
- **Concaténation** : Création d'un nouveau tuple à partir de plusieurs tuples.
- **Accession** : Le tuple suivi d'une paire de crochets, contenant un index ou une chaîne d'indexes existants.
- **Mise à jour** : Les valeurs des tuples ne sont pas modifiables.
- **Suppression** : Seule la suppression du tuple en entier est autorisée.

```
tuple1 = ('apple', 'banana')  
tuple2 = 1, 2, 3, 4, 5  
tuple3 = tuple1 + tuple2  
tuple4 = ()  
del tuple4  
tuple5 = (10,)  
tuple5[0] = 20 # Retourne une erreur de type
```

```
print("tuple3 :", tuple3)  
print("tuple3[1:5] :", tuple3[1:5])  
print("tuple5 :", tuple5)
```

Resultats

```
tuple3 : ('apple', 'banana', 1, 2, 3, 4, 5)  
tuple3[1:5] : ('banana', 1, 2, 3)  
tuple5 : (10,) # Les valeurs d'un tuple sont immuables
```



OPÉRATEURS

Les tuples répondent aux **opérateurs "+" et "*" comme des chaînes de caractères**. Ils signifient aussi **concaténation** et **répétition**, sauf que le résultat est un **nouveau tuple**.

En fait, les tuples répondent à toutes les **opérations de séquence générale** que nous avons utilisées sur les listes dans le chapitre précédent, excepté la mise à jour de valeur. Je rappelle, les valeurs d'un tuple sont **immuables**.

```
print("len((0, 1, 2)) : ", len((0, 1, 2)))  
print("(1, 2, 3) + (4, 5, 6) : ", (1, 2, 3) + (4, 5, 6))  
print("'apple' * 3 : ", ('apple') * 3)  
print("3 in (1, 2, 3) : ", 3 in (1, 2, 3))
```

```
for x in ("apple", "banana", "cherry"):  
    print(x)
```

Resultats :

```
len((0, 1, 2)) : 3 # Longueur  
(1, 2, 3) + (4, 5, 6) : (1, 2, 3, 4, 5, 6) # Concaténation  
'apple' * 3 : ('apple', 'apple', 'apple') # Répétition  
3 in (1, 2, 3) : True # Appartenance  
apple # Itération 1  
banana # Itération 2  
cherry # Itération 3
```



OPÉRATEURS

Comme les **tuples** sont des **séquences**,
l'**indexation** et le **découpage** fonctionnent de la
même manière pour les tuples que pour les listes.

- Le **premier index** est **0**.
- Un **index négatif**, indique qu'il faut **compter à partir de la droite**.
- "**x:y**" permet d'indiquer une **section**. Si **x** est **absent**, la section commence avec le **premier élément** et si **y** est **absent**, la fin de la section est le **dernier élément** du tuple.

```
fruits = ('Mango', 'Pineapple', 'Papaya')
```

```
print('fruits[2] : ', fruits[2])  
print('fruits[-2] : ', fruits[-2])  
print('fruits[1:] : ', fruits[1:])  
print('fruits[:2] : ', fruits[:2])  
print('fruits[:] : ', fruits[:])
```

Resultats :

```
fruits[2] : Papaya  
fruits[-2] : Pineapple  
fruits[1:] : ['Pineapple', 'Papaya']  
fruits[:2] : ['Mango', 'Pineapple']  
fruits[:] : ['Mango', 'Pineapple', 'Papaya']
```



FONCTIONS

len(list)

Donne le nombre d'élément contenu dans le tuple.

max(list)

Retourne l'élément du tuple ayant une valeur maximale.

min(list)

Retourne l'élément du tuple ayant une valeur minimale.

tuple(seq)

Convertit une liste en tuple.

```
tup1, tup2 = ['mango', 'apple', 'cherry'], [123, 'banana']  
list1 = (123, 'apple', 'banana', 'cherry')
```

```
print("First tuple length : ", len(tup1))  
print("Second tuple length : ", len(tup2))  
print("Max value element : ", max(tup1))  
print("Min value element : ", min(tup1))  
tupA = tuple(list1)  
print("Tuple elements : ", tupA)
```

Resultats :

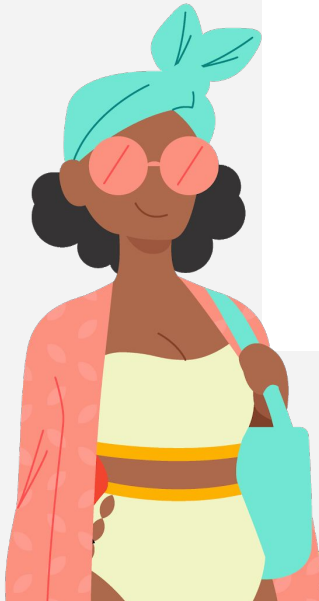
```
First tuple length : 3  
Second tuple length : 2  
Max value element : mango  
Min value element : apple  
Tuple elements : (123, 'apple', 'banana', 'cherry')
```


The background of the slide is a vibrant, colorful pattern of various tropical leaves and foliage. The leaves are in shades of green, yellow, orange, red, and purple, creating a dense and lively border around the central text. Some leaves are large and broad, while others are smaller and more delicate. The overall style is modern and artistic.

EXERCICES

EXERCICES

1. Quelles sont les différences entre les tuples et les listes?
2. Créez un tuple vide.
3. Créez un tuple avec différents types de données.
4. Extraire les valeurs du tuple que vous avez créé précédemment.
5. Ajoutez un élément à votre tuple.



EXERCICES

6. Convertissez le tuple suivant en chaîne de caractères : ('p', 'y', 't', 'h', 'o', 'n')

7. Combien de fois retrouvons nous l'élément "G" dans le tuple suivant :

```
gen = ("A", "A", "A", "U", "U", "U", "A", "A", "G", "U", "U", "C", "A", "A", "T", "U", "U", "A", "A", "A",  
"C", "U", "U", "G", "G", "A", "A", "C", "U", "U", "G", "A", "G", "C", "U", "C", "G", "A", "T", "C", "U", "A",  
"G", "A", "C", "C", "U", "G", "A", "G", "A", "U", "C", "U", "A", "G", "G", "U", "C", "C", "A", "G", "T",  
"U", "C", "A", "A", "G", "C", "U", "C", "G", "T", "C", "A", "A", "G", "U", "T", "C", "G", "A", "G", "C", "G",  
"G", "A", "C", "C", "U", "G", "G", "G")
```

8. Combien d'éléments possède le tuple précédent?

9. Supprimez le second élément du tuple suivant : ("w", 0, "o", "r", "k", " ", "h", "a", "r", "d")

10. Affichez les 4 premiers éléments du dernier tuple.



The background of the entire slide is a dense, colorful pattern of various tropical leaves and foliage. The leaves are rendered in a flat, stylized manner with bold outlines. Colors include bright red, teal, yellow, orange, and purple. Some leaves have detailed vein patterns, while others are solid colors. The pattern is symmetrical, with foliage framing the central white area.

SOLUTIONS

CORRECTION

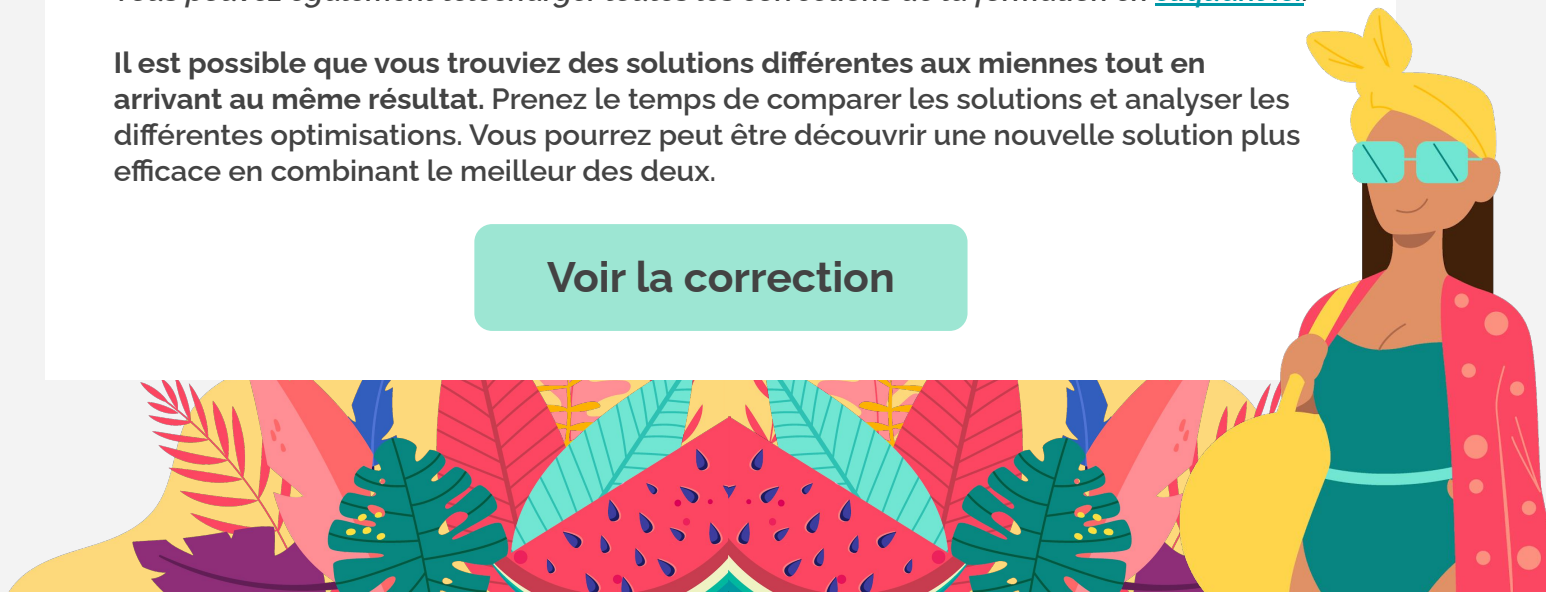
Pour visualiser la correction du chapitre cliquer sur le bouton ci-dessous.

*Le fichier de la correction s'ouvrira dans un nouvel onglet de votre navigateur préféré.
Pour cela vous devez avoir accès à une connexion Internet.*

Vous pouvez également télécharger toutes les corrections de la formation en [cliquant ici](#).

Il est possible que vous trouviez des solutions différentes aux miennes tout en arrivant au même résultat. Prenez le temps de comparer les solutions et analyser les différentes optimisations. Vous pourrez peut être découvrir une nouvelle solution plus efficace en combinant le meilleur des deux.

Voir la correction





**Félicitation vous avez terminé
le chapitre sur les tuples avec
Python!**

A decorative border of various tropical leaves in vibrant colors like red, orange, yellow, green, and purple surrounds the central white area. The leaves include Monstera, palm, and other exotic foliage.

CRÉDITS

- Modèle de la présentation par [Slidesgo](#)
- Icônes par [Flaticon](#)
- Images et infographies par [Freepik](#)