



Découvrir et approfondir python



Types construits séquences, listes :

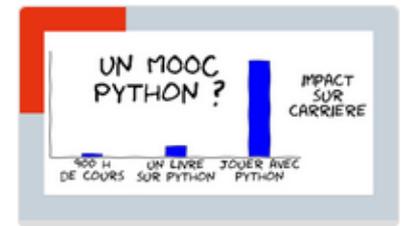
 V W2-03 Les séquences.mp4

 V W2-04 Les listes.mp4

Vidéos proposées par l'INRIA sur



Python 3 : des fondamentaux aux concepts avancés du langage



Thierry PARMENTELAT et Arnaud Legoud

Pour bien comprendre et approfondir :

Cocher les affirmations exactes sur les séquences :

- Une séquence de n éléments contient n éléments indicés de 1 à n
- Une séquence peut contenir aucun élément
- Les séquences sont ordonnées
- Une séquence de n éléments contient n éléments indicés de 0 à n-1
- Les nombres entiers sont des séquences
- Les séquences contiennent une relation d'ordre

Accès aux éléments individuels d'une séquence, slicing. Cocher les affirmations correctes :

Nous avons défini une chaîne de caractère s qui contient la chaîne ci-dessous :

L e s t y p e s c o n s t r u i t s

- ❖ Que me sélectionne s[10] :
 - ' '
 - 'c'
 - 'o'
- ❖ La longueur de la chaîne est donnée par la fonction built-in :
 - s.len
 - s.len()
 - len(s)
- ❖ Pour sélectionner uniquement 'types' il faut faire :
 - s[4 :9]
 - s[4 :8]
 - s[-14 :-9]
 - s[-16 :-11]
- ❖ Que donne l'instruction s.count('s') :
 - 2
 - 4
 - 6

Cocher les affirmations exactes sur les listes :

- Une liste ne peut contenir que des éléments d'un même type
- Une liste est un objet mutable (modifiable)
- Une liste est une séquence
- Je peux utiliser des opérations de slicing sur les listes
- Deux listes sont définies `s1 = [1,2,3,4,5]` et `s2 = [6,7,8,9,10]` je peux les regrouper avec l'opérateur +
- J'ai une liste dénommée `ma_liste`, Je peux appliquer la méthode de tri `ma_liste.sort()` sur cette liste quelque soit les types d'éléments contenus sans provoquer d'erreurs
- Une liste peut être triée sans que l'interpréteur python en face une copie temporaire.

Opérations sur les listes :

Voilà une liste : `ma_liste = [1, 'Bonjour', 45.38, 'le', True, 'monde']`

- ❖ Que me sélectionne `>>> ma_liste[10]`
 - rien None
 - une exception index out of range
 - 'monde'
- ❖ Que me sélectionne `>>> s = ma_liste[1::2]`
 - [1, 'Bonjour', 45.38]
 - rien
 - ['Bonjour', 'le', 'monde']
- ❖ Que contient chaîne après l'instruction `>>> chaine = " ".join(s)`
 - Bonjourlemonde
 - Bonjour le monde
 - 'Bonjour le monde'

Utilisons les listes :

Voilà une séquence d'opérations sur les listes :

```
>>> ma_liste = [1, 'Bonjour', 45.38, 'le', True, 'monde']  
  
>>> ma_liste[2:5]=['le', 'nouveau']  
  
>>> del ma_liste[0]
```

- ❖ Que contient `ma_liste` après l'exécution de la séquence :
 - [1, 'Bonjour', 45.38, 'le', True]
 - ['Bonjour', 'le', 'nouveau', 'monde']
 - [1, 'Bonjour', 45.38, 'le', True, 'monde', 'nouveau', 'monde']