

Apprendre à programmer avec Python

L5FL024 / LYOU008 - Groupe L

Séance 10 : for et listes

Marine DELABORDE

Sorbonne
Nouvelle 
université des cultures



Informations générales

Rappels

- **Horaires** : le mardi de 17h à 19h (discord)
- **Modalités d'enseignement** : distanciel synchrone
- **Contact** : marine.delaborde@gmail.com
- **Ressources** :
 - icampus + <http://www.linguist.univ-paris-diderot.fr/~amsili/Ens/L5FL024.php>
 - discord <https://discord.gg/37vxR6>
- **Fiche d'informations** : <https://framaforms.org/fiche-dinformations-1600359707>
- **Fiche de présence**: <https://lite.framacalc.org/9jbo-presence-python-gl>
- **Modalités de contrôle des connaissances** :
Quiz (4 meilleurs sur 5) 50% + Examen (dernière séance) 50%

Quizz 3 : correction

- **Quiz 4/5 : 10 minutes**
- **Les notes des 4 meilleurs quiz seront prises en compte pour représenter 50% de la note finale du cours**

exercices sur repl.it : <https://repl.it/languages/python>

Corrections sur :

<http://www.linguist.univ-paris-diderot.fr/~amsili/Ens/L5FL024.php>

Rappel : for

- Instruction qui travaille sur des séquences (une chaîne de caractères est une séquence de caractères) :

```
mot = "bonjour"
```

```
for lettre in mot:
```

```
    print(lettre)
```

=> b, o, n, j, o, u, r

rappel : range()

- **Fonction qui génère une suite arithmétique**

- `range(5) = range(0,5) = 0, 1, 2, 3, 5`

- Pour vérifier :

```
for n in range(2,6):  
    print(n)
```

=> 2, 3, 4, 5

rappel : les listes

- **Structure de données**

```
liste = [] #initialisation
```

- entre **crochets** et séparés par une **virgule**

Ex :

```
fruits = ['pomme', 'poire', 'cerise']  
print(fruits)
```

- **les types peuvent être différents** (str, int, float, list, etc.)

Ex :

```
mixte = ['pomme', 3, 'cerise']  
print(mixte)
```

- **données accessibles via des indices** (positifs ou négatifs)

Ex :

```
fruits = ['pomme', 'poire', 'cerise']  
print(fruits[0])
```

rappel : listes + for

- Parcours de la liste

```
fruits = ['pomme', 'poire', 'cerise', 'kiwi', 'fraise']
```

```
for fruit in fruits:
```

```
    print(fruit)
```

Correction : exercices (feuille n°9)

1. mois = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Remplacer 12 par décembre dans la liste **mois**.
2. lst = [3,'girafe',True,2.9,'maison',3987,'janvier',2.0, 39, 268]
 - a. Modifier le 6ème élément de la liste lst et lui donner la valeur "o"
 - b. Afficher la tranche de la liste de True à 2.0
3. Créer une liste des jours de la semaine et afficher chaque jour à l'aide d'une boucle for. À la fin, afficher le 3ème élément de la liste (mercredi).

Longueur d'une liste

- Fonction `len()` :

- `len(liste)`

- Exemple :

```
ma_liste = [1, 2, 3, 4]
print(len(ma_liste))
```

```
>>> 4
```

- Fonctionne aussi pour les strings (nombre de caractères)

Méthodes des objets de type “liste”

- **liste.méthode(paramètres)**
- Ajouter : **append(x)**
 - Exemple :

```
ma_liste = [1, 2, 3, 4]
ma_liste.append(5)
```
- Supprimer une valeur : **remove(x)**
 - Exemple :

```
ma_liste.remove(4) #ma_liste.remove(ma_liste[3])
```
- Supprimer via un indice : **pop(x)**
 - Exemple :

```
ma_liste.pop(2) #ma_liste.remove(ma_liste[2])
```

Supprimer un élément d'une liste avec la fonction **del** : `del ma_liste[3]`

Supprimer le contenu d'une liste avec la fonction **del** : `del ma_liste[:]` équivalent à `ma_liste.clear()`

Méthodes des objets de type “liste”

- Compter le nombre d’occurrences d’une valeur : **count(x)**
- Trouver l’index d’une valeur : **index(x)**
- Insérer un élément à une position précise : **insert(i, x)**
- Trier une liste : **sort()**
- Copier une liste : **copy()**

- Exemple :

```
ma_liste = [1, 2, 3, 4]
ma_nouvelle_liste = ma_liste.copy()
print(ma_nouvelle_liste)
```

append() et extend()

```
ma_liste = [1, 2, 3, 4]
ma_liste.append([5, 6])
print(ma_liste)
```

```
>>> [1, 2, 3, 4, [5, 6]] #liste de liste
```

```
ma_liste = [1, 2, 3, 4]
ma_liste.extend([5, 6])
print(ma_liste)
```

```
>>> [1, 2, 3, 4, 5, 6]
```

Liste // String

- **str > list : `split()`**

- Exemples :

```
phrase = "Je sais faire des programmes en python"  
liste = phrase.split() #séparateur = espace  
liste_v = phrase.split(',') #séparateur = virgule
```

- **list > str : `join()`**

- Exemple avec l'espace comme séparateur :

```
liste = ['Je', 'sais', 'faire', 'des', 'programmes', 'en', 'python']  
phrase = ' '.join(liste)
```

Exercices (feuille n°10)

1. `nombres = [78, 2, 2332, 12, 823, 3, 399, 23090, 93920, 99, 8383, 3904, 34, 449, 34]`
 - 1.1. Créer une fonction "**ex1**" (prenant une liste comme paramètre) qui :
 - 1.1.1. ajoute 198 à la fin de la liste,
 - 1.1.2. supprime 78,
 - 1.1.3. trie la liste,
 - 1.1.4. retourne la longueur de la liste
 - 1.2. Afficher la liste et sa longueur dans le terminal

2. `liste_nombres = [34, 67, 98, 109, 284, 387]`

Créer une nouvelle liste qui contient le carré de chaque élément de la liste **liste_nombres**.

3. `message = "Bonjour à tous,\nBonjour à toutes !\nBonne journée !"`
 - 3.1. Afficher la longueur de la variable **message**.
 - 3.2. Créer une liste "**bonjour**" basée sur la séparation de la chaîne de caractères de la variable **message**, séparée suivant le saut de ligne (**\n**).
 - 3.3. Afficher la longueur de la liste **bonjour**.