

Les boucles FOR et WHILE

Savoir-faire associés

Ceci sont les savoirs-faire à acquérir grâce au TP

Compétence(s)

Ceci sont les compétences acquises au cours du TP

Auteurs : GL, SV, VB – à télécharger sur: <http://www.ptsi-eiffel-bordeaux.fr>



Figure 1. image nommée "intro-TPI.png" éventuelle

Objectif(s) du TP

Les objectifs de ce TP sont de :

- Apprendre à manipuler les listes grâce à différentes méthodes intégrées ;
- Apprendre à tracer une courbe ;
- Mettre en œuvre une boucle FOR ;
- Mettre en œuvre une boucle WHILE ;
- Appliquer les différentes notions vues au tracé de la réponse temporelle d'un modèle d'ordre 1.

Table des matières

1	Méthodes principales de manipulation des listes	3
1.1	list.append()	3
1.2	list.extend()	3
1.3	list.remove()	3
1.4	list.reverse()	4
1.5	list.sort()	4
2	Tracé de courbe	5
3	Mise en œuvre d'une boucle FOR	5
4	Mise en œuvre d'une boucle WHILE	6

5 Application au tracé de la réponse temporelle d'un modèle d'ordre 1

6

1. Méthodes principales de manipulation des listes

La liste est la structure de donnée la plus utilisée en Python. Pour programmer correctement dans ce langage, il est donc crucial de savoir l'utiliser efficacement dans tous les cas. Le but de ce partie est de vous apprendre certaines fonctions et techniques à utiliser avec les listes pour les manier de manière optimale dans vos programmes.

1.1 list.append()

Activité 1 Tester l'exemple ci-dessous dans un script, l'exécuter et commenter le résultat obtenu :



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

Syntaxe générale :



```
list.append(element)
```

Utilité : comme son nom l'indique en anglais, cette fonction sert à ajouter à la liste "list" un élément "element" à la fin de cette dernière.

1.2 list.extend()

Activité 2 Tester l'exemple ci-dessous dans un script, l'exécuter et commenter le résultat obtenu :



```
li = [1, 2, 3, 4, 5]
print li
li.extend([6, 7, 8])
print li
```

Syntaxe générale :



```
list.extend(autre_liste)
```

Utilité : comme son nom l'indique en anglais, cette fonction sert à ajouter à la liste "list" une autre liste "autre_liste" à la fin de cette dernière.

1.3 list.remove()

Activité 3 Tester l'exemple ci-dessous dans un script, l'exécuter et commenter le résultat obtenu :



```
li = [11,12, 33, 47, 58]
print li
li.remove(33)
print li
```

Syntaxe générale :



```
list .remove(element)
```

Utilité : comme son nom l'indique en anglais, cette fonction sert à retirer de la liste "list" le premier élément rencontré égal à "element". Cela ne correspond donc pas à la position de cet élément dans la liste.

1.4 list.reverse()

Activité 4 Tester l'exemple ci-dessous dans un script, l'exécuter et commenter le résultat obtenu :



```
li = [1,2,5,3,4]
print li
li .reverse()
print li
```

Syntaxe générale :



```
list .reverse()
```

Utilité : comme son nom l'indique en anglais, cette fonction sert à inverser l'ordre des éléments de la liste "list". Il ne s'agit en aucun cas d'un classement par ordre croissant ou décroissant des éléments de la liste.

1.5 list.sort()

Activité 5 Tester l'exemple ci-dessous dans un script, l'exécuter et commenter le résultat obtenu :



```
li = [1,5,4,2,3]
print li
li .sort()
print li
```

Syntaxe générale :



```
list .sort()
```

Utilité : comme son nom l'indique en anglais, cette fonction sert à classer par ordre croissant ou alphabétique les éléments de la liste "list".

2. Tracé de courbe

Pour pouvoir tracer une courbe avec Python, il faut faire appel à une librairie spécifique qui est déjà pré-installée dans votre environnement de développement. Il s'agit de matplotlib.

Activité 6 Tester l'exemple ci-dessous dans un script, l'exécuter et indiquer sous la forme de commentaires le rôle de chaque ligne de ce script :



```
import matplotlib.pyplot as plt
temps=[0,1,2,3,4,5]
s=[2,4,7,5,12,8]
plt . plot (temps, s)
plt . xlabel ('temps (s)')
plt . ylabel ('s(t)')
plt . title ('Evolution temporelle de s(t)')
plt . grid (True)
plt . savefig ("graphe.png")
plt . show()
```

Remarque : pour insérer un commentaire dans un script, il faut insérer le symbole # et écrire le commentaire du côté droit de ce symbole.

3. Mise en œuvre d'une boucle FOR

Activité 7 Tester les deux exemples ci-dessous dans deux scripts distincts, les exécuter et indiquer sous la forme de commentaires le rôle de chaque ligne de ces scripts :

Script 1 :



```
for i in range(9):
    print(i)
```

Script 2 :



```
for i in range(3,9):
    print(i)
```

Syntaxe générale :



```
for variable in range(valeur_debut,valeur_fin-1):
    instruction 1
    instruction 2
    ...
```

ou



```
for variable in range(valeur_fin-1):
    instruction 1
    instruction 2
    ...
```

Utilité : Une boucle FOR est utilisée lorsque une suite d'instructions doit être exécutée un nombre de fois connu à l'avance. Prendre garde à la valeur de l'indice i suivant la syntaxe retenue.

4. Mise en œuvre d'une boucle WHILE

Activité 8 Tester l'exemple ci-dessous dans un script, l'exécuter et indiquer sous la forme de commentaires le rôle de chaque ligne de ce script :



```
var=0
while var<10:
    print(var)
    var=var+1
print "End of loop"
```

Syntaxe générale :



```
var=...
while var condition val:
    instruction 1
    instruction 2
    ...
    var=var+1
```

Utilité : Une boucle WHILE est utilisée lorsque une suite d'instructions doit être exécutée un nombre de fois non obligatoirement connu à l'avance. La suite d'instructions sera exécutée tant que la condition ne sera pas vérifiée.

Les conditions sont de la forme :

- == pour tester l'égalité entre la variable "var" et la valeur "val"
- != pour tester la non égalité entre la variable "var" et la valeur "val"
- < pour tester l'infériorité stricte de la variable "var" par rapport à la valeur "val"
- > pour tester l'a supériorité stricte de la variable "var" par rapport à la valeur "val"
- <= pour tester l'infériorité de la variable "var" par rapport à la valeur "val"
- >= pour tester la supériorité de la variable "var" par rapport à la valeur "val"

5. Application au tracé de la réponse temporelle d'un modèle d'ordre 1

L'objectif de cette partie est de mettre à profit l'ensemble des notions vues précédemment afin de tracer l'évolution temporelle de la réponse d'un modèle d'ordre 1. L'étude se limitera ici à la réponse du modèle pour une entrée en forme d'échelon $e(t) = e_0 \cdot \Gamma(t)$.

Le modèle d'ordre 1 correspond à une équation différentielle de degré 1 de la forme : $\tau \cdot \frac{ds(t)}{dt} + s(t) = K \cdot e(t)$, avec $s(t)$ la réponse du modèle et $e(t)$ la sollicitation à laquelle il est soumis.

Il est également souhaité que l'intervalle de calcul de la réponse de ce modèle $[t_i, t_f]$ soit paramétrable, ainsi que la valeur de la condition initiale notée $s(0) = s_0$.

Principe général Il s'appuie sur la définition de la dérivée d'une fonction continue : $\frac{ds(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{s(t+\Delta t) - s(t)}{(t+\Delta t) - (t)} = \lim_{\Delta t \rightarrow 0} \frac{s(t+\Delta t) - s(t)}{\Delta t}$
 Δt est appelé le pas de temps. La dérivée de la fonction de sortie de l'équation différentielle étudiée peut alors être approchée par l'expression discrète précédente si le pas de temps est suffisamment petit.

Activité 9 Ouvrir un nouveau script et proposer une suite d'instructions qui permettent de récupérer tous les paramètres nécessaires à la résolution numérique du modèle du premier ordre étudié ($\tau, K, e0, ti, tf, s0, pas$).

Activité 10 Créer deux listes vides nommées temps et s. Ajouter à ces listes la valeur initiale du temps ti, ainsi que la valeur initiale s0.

Activité 11 A partir de la version discrète du calcul de la dérivée première de la fonction s(t), écrire une boucle permettant de calculer la valeur de s(t+ Δt) à partir de la valeur précédente s(t) et des autres paramètres. A chaque passage dans la boucle, il faudra ajouter aux listes temps et s les valeurs calculées pour le temps suivant et la valeur suivante de s(t).

Activité 12 Terminer le script par le tracé de la réponse s(t) en fonction du temps.

Activité 13 Tester le script créé pour un pas de 1s et de 0.1s. Noter vos observations et proposer des solutions pour régler le problème observé.

Activité 14 Tester différents jeux de paramètres et surtout faire varier la valeur du pas de temps. Noter vos observations. Comment choisir alors la valeur du pas optimale ?