

Cours

1 Variables informatiques

Définition d'une variable

Il y a quatre types de variables : les entiers, les booléens (nombres ne s'écrivant qu'avec des 0 et des 1 tels que les comprennent les machines, les robots, les ordinateurs) les nombres à virgules (appelés flottants ou float) et les chaînes de caractères.

Vocabulaire à connaître, notations

Entiers, booléen, flottant (nombres à virgules ou float), chaîne de caractères.

2 Notion de liste

Définition d'une liste

Une liste est comme son nom l'indique une liste de nombres ou de caractères qui sont indexés (c'est-à-dire que chaque élément de la liste est repérable par un indice). Voici comment créer la liste [14,23,15,6] avec Python :

```
Liste=[14, 23, 15, 6]
```

Remarque importante

Dans Python, le premier terme d'une liste a pour indice 0.

Par exemple, si Liste=[14,23,15,6], alors Liste[0]=14.

Ensuite, on a Liste[1]=23, Liste[2]=15 et Liste[3]=6.

Vocabulaire à connaître, notations

Liste, Liste[i] désigne l'élément de la liste d'indice i.

Manipulations de base sur les listes

La fonction Push (ou ajouter)

Cette fonction ajoute un élément en fin de liste.

```
Liste=[14,23,15,6]

def Push(Liste,élément):
    Liste.append(élément)
    return Liste
```

Par exemple, on veut ajouter le terme 7 à la liste [14,23,15,6].

```
>>> Push(Liste,7)
[14, 23, 15, 6, 7]
```

La fonction Pop (ou retirer)

Cette fonction retire un élément en fin de liste.

```
Liste=[14,23,15,6]

def Pop(Liste):
    del Liste[-1]
    return Liste
```

Par exemple, on veut retirer le dernier élément de la liste [14,23,15,6].

```
>>> Pop(Liste)
[14, 23, 15]
```

La fonction Échanger

Cette fonction échange les termes Liste[i] et Liste[j].

```
Liste=[14,23,15,6]

def Echanger(Liste,i,j):
    a=Liste[j]
    Liste[j]=Liste[i]
    Liste[i]=a
```

Par exemple, on veut échanger les termes Liste[1] et Liste[3] de la liste.

```
>>> Echanger(Liste,1,3)
>>> print(Liste)
[14, 6, 15, 23]
```

La fonction Étendre

Cette fonction étend la liste d'une autre liste.

```
Liste=[1,2,3]

def Etendre(Liste,M):
    Liste.extend(M)
```

Par exemple, si on veut étendre la liste [1,2,3] de la liste [4,5,6].

```
>>> Etendre(Liste,[4,5,6])
>>> print(Liste)
[1, 2, 3, 4, 5, 6]
```

Vocabulaire à connaître, notations

def (pour définir une fonction), Push (ajouter), Pop (retirer), Échanger, Étendre.

3 Instruction conditionnelle

Définition

Il y a deux types d'instruction conditionnelle, le « si » (if) et le « si-sinon » (if-else). Ces instructions permettent d'aiguiller l'algorithme sur ce qu'il doit faire avant de continuer ou pas.

Vocabulaire à connaître

Instruction conditionnelle « si » (ou « if »), et « si-sinon » (ou « if-else »).

4 Boucle bornée (for), boucle non bornée (while)

Définition

Il y a deux types de boucle, la boucle « pour » (for) et la boucle « tant que » (while). La boucle « pour » possède un itérateur incorporé qui augmente de 1 à chaque tour et qui est programmé pour effectuer un certain nombre de tours.

La boucle « tant que » (while) effectue, à chaque nouveau tour, une vérification lui permettant de savoir s'il faut continuer ou bien s'arrêter.

Vocabulaire à connaître, notations

Boucle « pour » (ou boucle « for »), boucle « tant que » (ou boucle « while »).

Remarque

Sous Python, si on veut 100 tours de boucle, on écrit « for i in range(0,100) » ou « for i in range(1,101) ».

5 Fonction à un ou plusieurs arguments**Exemple d'une fonction à un argument**

Voici comment créer la fonction carrée avec Python :

```
def f(x) :  
    return x**2
```

Ensuite, quand on appuie sur F5 (pour accéder à la console), on peut calculer n'importe quelle image :

```
>>> f(2)  
4  
>>> f(3)  
9  
>>> f(4)  
16  
>>> f(-1)  
1
```

Vocabulaire à connaître, notations

def (pour définir une fonction).

Exemple d'une fonction à deux arguments

Voici comment créer la fonction somme de deux nombres a et b avec Python :

```
def somme(a,b) :  
    return a+b
```

Ensuite, quand on appuie sur F5 (pour accéder à la console), on peut calculer n'importe quelle somme de deux nombres :

```
>>> somme(12,79)  
91
```

Remarque

Python peut créer des fonctions à trois, quatre arguments ou même plus.

Voici par exemple, la fonction produit de trois nombres a , b et c .

```
def produit(a,b,c):  
    return a*b*c
```

6 Fonction renvoyant un nombre aléatoire**Fonction renvoyant un nombre aléatoire compris entre a et b**

Voici comment procéder :

```
from random import *  
  
def nombre_aléatoire(a,b):  
    return uniform(a,b)
```

Ensuite, quand on appuie sur F5 (pour accéder à la console), on peut obtenir un nombre aléatoire par exemple compris entre 2 et 16.

```
>>> nombre_aléatoire(2,16)  
4.1545898286135134
```

Exercices

Compétences attendues

- Utiliser un générateur de nombres aléatoires entre 0 et 1 pour simuler une loi de Bernoulli de paramètre p .

Exercice 1.1

Mettre en œuvre des algorithmes

Considérons une variable aléatoire X qui suit une loi de Bernoulli de paramètre p (c'est-à-dire qu'elle prend deux valeurs : la valeur 1 avec la probabilité p , et la valeur 0 avec la probabilité $q=1-p$).

On a donc $p(X=1)=p$ et $p(X=0)=1-p$.

1. Écrire une fonction renvoyant un nombre aléatoire compris entre deux nombres a et b .
2. En déduire une fonction permettant de simuler une loi de Bernoulli de paramètre p .

Compétences attendues

- Utiliser la notion de compteur

Exercice 1.2

Mettre en œuvre des algorithmes

- Écrire un algorithme, sous Python, qui simule 100 piles ou faces et affiche le nombre de fois où on tombe sur pile au cours de ces 100 lancers.

Exercice 1.3

Mettre en œuvre des algorithmes, raisonner

Voici un programme qui compte la fréquence d'apparition de la lettre « e » dans le texte de Montesquieu (1689-1755) : « Moins on pense, plus on parle ».

```

texte="Moins on pense, plus on parle"
nb=0
for i in range(0,len(texte)):
    if texte[i]=='e':
        nb=nb+1
frequence=nb/len(texte)
print(frequence)

```

1. Écrire un algorithme, sous Python, qui compte la fréquence d'apparition de la lettre « j » dans le texte de Montesquieu :
« Le commerce guérit des préjugés destructeurs et c'est presque une règle générale que partout où il y a des mœurs douces, il y a du commerce et que partout où il y a du commerce, il y a des mœurs douces. »
2. En déduire alors la fréquence de la lettre « j » dans ce texte.

Compétences
attendues

- Utiliser le principe d'accumulateur pour calculer une somme, un produit

Exercice 1.4

Mettre en œuvre des algorithmes

- Écrire un algorithme, sous Python, qui calcule la somme $1+2+3+\dots+n$.

Exercice 1.5

Mettre en œuvre des algorithmes

- Écrire un algorithme, sous Python, qui calcule la somme $1\times 2\times 3\times \dots\times n$.

Compétences
attendues

- Identifier les entrées et les sorties d'une fonction

Exercice 1.6

Mettre en œuvre des algorithmes, changer de registre

Voici une fonction écrite sous Python :

```
def f(x):
    if x<1:
        return x+1
    else:
        return -2*x+4
```

1. Identifiez les entrées et sorties de cette fonction.
2. Que se passe-t-il si on modifie la ligne « *if x < 1* » en « *if x ≥ 3* » ?

Compétences attendues

- Structurer un programme en ayant recours aux fonctions

Exercice 1.7

Mettre en œuvre des algorithmes

1. Écrire une fonction permettant de déterminer la moyenne de deux nombres a et b défini par la formule $moy(a,b) = \frac{a+b}{2}$.
2. En déduire une fonction permettant de calculer l'écart type de nombres a et b défini par la formule : $\sqrt{[a - moy(a,b)]^2 + [b - moy(a,b)]^2}$.
3. a. Écrire un programme qui demande la valeur de deux nombres a et b et qui affiche l'écart type de a et b .
b. Qu'affiche ce programme lorsqu'on le fait fonctionner pour $a = 8$ et $b = 12$?

Compétences attendues

- Générer une liste (en extension, par ajouts successifs ou en compréhension)

Exercice 1.8

Mettre en œuvre des algorithmes

1. Créer sous Python la liste $L = [0, 1, 2, 3, 4, 5, 6]$.
2. Générer en extension la liste L en la liste $M = [0, 1, 2, 3, \dots, 18]$ par ajouts successifs.