Cours Introduction à la Programmation Python III

(IP1 Python)

Arnaud Sangnier
sangnier@irif.fr

Mercredi 4 Octobre 2017
MIASHS et MATHS

Dans les épisodes précédents

- Présentation de certains aspects de Python 3
- Ce que nous avons vu :
  - Les données et leur type : int et str
  - Les variables : affectation, lecture, modification
  - Les fonctions : def fonction (…) :
  - Les tests : if () : … else …
  - Les boucles : for i in range (0,10,1)

Construction de programmes - I

- Comment construire vos programmes
  - Rappel : un programme est une suite d'instructions
  - Quelles instructions avons nous vu :
    - Affectation de variables x = 5 (la première fois)
    - Modification de variables x = x + 1
    - Appel de fonctions y = f(x) ou g(x) (si g ne renvoie pas de valeur)
    - Tests :

```
if (x > 5) :
    [suite d'instructions]
else :
    [suite d'instructions]
```

    - Boucles :

```
for i in range (0,10,1) :
    [suite d'instructions]
```

Exemple

- Écrire un programme qui affiche 15 lignes de la forme **********...**** avec 60 étoiles par ligne
  - Il faut faire une boucle par ligne (qui va donc faire 15 tours
  - À la fin de cette boucle il faut aller à la ligne
  - Avant il faut afficher la ligne, c'est-à-dire les 60 étoiles
    - On fait une boucle qui fait 60 tours

```
for i in range (0,15,1) :
    for j in range (0,60,1) :
        print(''*'',end='''')
    print('''')
```

Cette instruction appartient à la boucle interne

Cette instruction appartient à la boucle externe

Construction de programmes - II

- On peut ainsi utiliser un if dans un if, un for dans un for (boucles imbriquées), un if dans un for etc
- Il faut aussi définir les fonctions que l'on utilise dans le programme. Typiquement on les met avant dans le fichier.

```
def function (x, y) :
    [suite d'instructions]
```

- Comme un appel de fonctions est une instructions, **on peut aussi appeler une fonction dans une fonction!**

Exemple

- Écrire une fonction qui renvoie la somme des carrés de 1 à n où n est donné en paramètres
  - Combien de paramètres prend la fonction : un de type int
  - La fonction retourne-t-elle une valeur : oui de type int
  - Quelle est le nom de la fonction : somCarres

```
def somCarres (n) :
  x = 0
  for i in range(1,n+1,1) :
    x = x + (i * i)
  return x
```

Exemple

- Écrire une fonction qui pour un n donné calcule le produit S1 * S2 * … * SN où Si est la somme des carrés des entiers de 1 à i
  - Combien de paramètres prend la fonction : un de type int
  - La fonction retourne-t-elle une valeur : oui de type int
  - Quelle est le nom de la fonction : prodSomCarres
  - On va utiliser la fonction précédente

Exemple

```
def somCarres (n) :
  x = 0
  for i in range(1,n+1,1) :
    x = x + (i * i)
  return x

def prodSomCarres (n) :
   r = 1
   for i in range(1,n+1,1) :
     r = r * somCarres(i)
   return r

z = prodSomCarres (3)
print(z)
```

Ces deux n sont différents

Bien tester sa fonction

Exemple

```
def somCarres (n) :
  x = 0
  for i in range(1,n+1,1) :
    x = x + (i * i)
  return x

def prodSomCarres (m) :
   r = 1
   for i in range(1,m+1,1) :
     r = r * somCarres(i)
   return r

z = prodSomCarres (3)
print(z)
```

Exemple

- Écrire une fonction qui prend deux paramètres n et l et qui affiche n lignes de l étoiles
  - Combien de paramètres prend la fonction : deux de type int
  - La fonction retourne-t-elle une valeur : non
  - Quelle est le nom de la fonction : ligneEtoiles
  - On a ce code qui affiche 15 lignes de 60 étoiles

```
for i in range (0,15,1) :
    for j in range (0,60,1) :
      print(‘’*’’,end=’’’’)
     print(’’’’)
```

Exemple

```
def ligneEtoiles (n, l) :
  for i in range (0,n,1) :
    for j in range (0,l,1) :
      print(‘’*’’,end=’’’’)
    print(’’’’)
```

Appel de fonctions dans les fonctions

- Lorque l'on appelle une fonction dans une autre fonction, il faut faire attention à ne pas créer de comportements infinis
- Par exemple :

```
def f (x) :
  y = 2 * x
  z = g(y)
  return z

def g(u) :
   t = u *u
   w = f (t)
   return w

a = f(5)
print(a)
```

Qu'affiche ce programme ?
RIEN
Il ne termine jamais et finit par provoquer une erreur

Un nouveau type

- Le type int représente les valeurs entières : -2,1,-100,5,0,…
- Le type str représente les chaînes de caractères : ‘’aaa”, ‘’Hello”, ‘’Bob”,…
- Le type bool représente les booléens.
- Les booléens n’ont que deux valeurs possibles : True et False
- Les opérateurs booléens
  - opérateur unaire : not (non)
  - opérateurs binaires : or (ou) et and (non)
  - pour savoir comment interpréter ces opérateurs il faut connaître leur tables de vérité

Les tables de vérité - I

- a and b

a	b	a and b
True	True	True
False	True	False
True	False	False
False	False	False

- a or b

a	b	a or b
True	True	True
False	True	True
True	False	True
False	False	False

Les tables de vérité - II

- Les tables de vérité peuvent exister pour des formules plus complexes
- Par exemple : a and (b or (not c))

a	b	c	not c	b or (not c)	a and (b or (not c))
True	True	True	False	True	True
False	True	True	False	True	False
True	False	True	False	False	False
False	False	True	False	False	False
True	True	False	True	True	True
False	True	False	True	True	True
True	False	False	True	True	True
False	False	False	True	True	False

Où trouve-t-on les booléens ?

- En fait, vous les utilisez déjà
- Quand on fait un test, par exemple 3 < 2, il correspond à une valeur booléenne, ici False
- Ainsi si on fait type(3<2) on obtient le type bool
- Donc dans un if (…) ce que l'on met entre parenthèses est une valeur de type bool
- Ainsi toutes les expressions suivantes sont des booléens
  - 2==3, 2<=3, "aaaa"=="bbbb" etc

Booléens dans des fonctions - I

- On peut ainsi faire des fonctions qui renvoient un booléen
- Par exemple, une fonction qui dit si un nombre n est premier ou non
- Pour cela on teste tous les entiers entre 2 et n-1, pour voir si il existe un diviseur de n

```
def estPremier (n) :
        r=True
        for i in range(2,n,1) :
            If (r % i == 0) :
                r= False
        return r
```

Booléens dans des fonctions - II

- On peut aussi mettre un booléen en paramètre
- Écrire une fonction prenant en paramètre un booléen b et un entier n et qui affiche n lignes avec n étoiles si b est vrai et n lignes avec n dièses si n est faux.

```
def sharpOrStar(b,n) :
  S = ""
  if (b) :
     c="*"
  else :
    c= "#"
  for i in range(0,n,1) :
    s =s + c
  for i in range(0,n,1) :
    print(s)
```