

1 TD 1 : Premiers pas en Python

(correction page ??)

Un programme informatique est la réalisation concrète d'une idée. Cette idée consiste en une succession de manipulations ou calculs appliqués à des données. Par exemple, admettons que vous vouliez compter le nombre d'emails que vous avez envoyés à plus d'une personne. Les données sont les emails, les manipulations consisteront à :

1. parcourir la liste des emails,
2. pour chacun d'entre eux, extraire les destinataires,
3. compter ceux pour lesquelles il y a plus d'un destinataire.

Un programme est une succession d'instructions qui permettent de décrire les manipulations souhaitées.

Troisième demi-heure : 5 pas

Cinq pas à faire durant cette demi-heure pour lesquels, il faudra recopier¹ ce qui suit :

1. La calculatrice

```
x = 5
y = 10
z = x + y
print (z)    # affiche z
```

2. L'incrémentation

```
x = 2
x = x + 1
print (x)    # affiche 3
x += 5
print (x)    # affiche 8
```

3. La répétition ou les boucles

```
a = 0
for i in range (0, 10) :
    a = a + i    # répète dix fois cette ligne
print (a)
```

4. L'aiguillage ou les tests

```
a = 10
if a > 0 :
    print(a)    # un seul des deux blocs est pris en considération
else :
    a -= 1
    print (a)
```

5. Les chaînes de caractères

1. ou copier/coller depuis http://www.xavierdupre.fr/site2013/enseignements/code_exemples/seance1_premiers_pas.html

```
a = 10
print (a)    # quelle est la différence
print ("a")  # entre les deux lignes
s = "texte"
s += "c"
print (s)
```

Quatrième demi-heure : 5 pas à reculons

Dans cette partie, il s'agit d'interpréter pourquoi un programme ne font pas ce qu'ils sont censés faire ou pourquoi ils provoquent une erreur.

1. Un oubli

```
a = 5
a + 4
print (a)    # ou voudrait voir 9 mais c'est 5 qui apparaît
```

Il ne faut pas hésiter à ajouter des `print` pour vérifier le contenu des variables à un endroit précis du programme.

2. Une erreur de syntaxe

```
a = 0
for i in range (0, 10)
    a = a + i
print (a)
```

Qui provoque :

```
File "i.py", line 2
    for i in range (0, 10)
                    ~
SyntaxError: invalid syntax
```

3. Une autre erreur de syntaxe

```
a = 0
for i in range (0, 10)
    a = a + i
print (a)
```

Qui provoque :

```
File "i.py", line 4
    print (a)
          ~
IndentationError: unindent does not match any outer indentation level
```

4. Une opération interdite

```
a = 0
s = "e"
print (a + s)
```

Qui provoque :

```
Traceback (most recent call last):
  File "i.py", line 3, in <module>
    print (a + s)
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

5. Un nombre impair de...

```
a = 0
for i in range (0, 10) :
    a = (a + (i+2)*3)
print (a)
```

Qui provoque :

```
File "i.py", line 4
    print (a)
          ^
SyntaxError: invalid syntax'
```

Pour aller plus loin ou pour ceux qui ont fini plus tôt

- Ecrire un programme qui calcule la somme des 10 premiers entiers au carré.
- Ecrire un programme qui calcule la somme des 5 premiers entiers impairs au carré.
- Ecrire un programme qui calcule la somme des 10 premières factorielles : $\sum_{i=1}^{10} i!$.