Introduction à Python, variables

Spé NSI - Lycée du parc

Année 2020-2021

Introduction

Dans ce chapitre, on prends contact avec Python qui est le langage choisi pour enseigner la programmation en spécialité NSI. On introduit aussi la première notion fondamentale celle de variable.

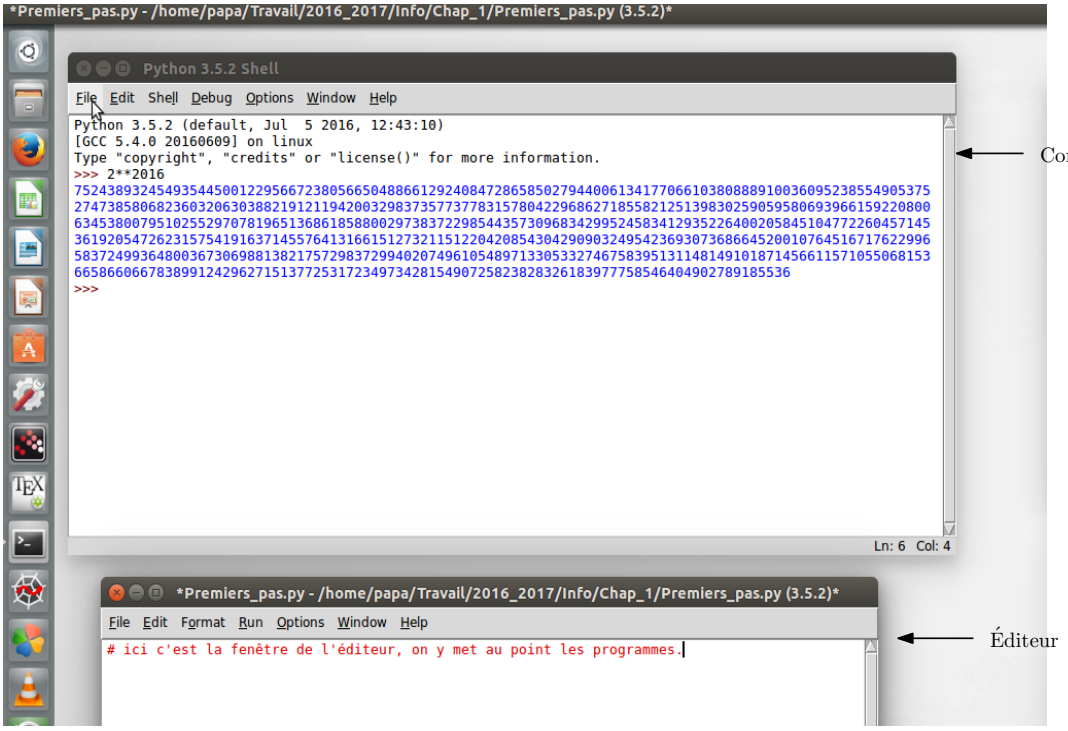
I Le langage Python

Définition (Langage de programmation)

Un **programme informatique** est un texte qui décrit les opérations que l'on souhaite faire exécuter par un ordinateur. Ce texte est écrit en suivant les règles d'un **langage de programmation**. Dans ce cours, nous utiliserons le langage **Python** (version 3).

Un langage de programmation fait interface entre l'homme et la machine, il est compréhensible par le premier et exécutable par la seconde.

Pour rédiger et mettre au point un programme on utilise généralement un Environnement de Développement Intégré : EDI (ou IDE en anglais). Pour Python il y en existe de nombreux, on va commencer avec le plus simple : **IDLE** qui est fourni avec l'installation standard du langage Python [1].



1. Voir à la fin du chapitre pour installer Python sur votre machine personelle.

Lorsque l'on lance IDLE, une fenêtre de console (Python Shell) apparait. Elle sert à exécuter des commandes simples et aide à la mise au point des programmes. C'est aussi dans la console que l'on voit le résultat de l'exécution des programmes. La console permet de mettre en oeuvre le *mode interactif de Python*.

À l'aide du menu *File* de la console, on peut ouvrir une nouvelle fenêtre pour écrire un programme : c'est l'éditeur, il permet de mettre en oeuvre le *mode programme*.

Exercice 1

Dans le mode interactif tester les commandes suivantes :

```
>>> 2 + 3

>>> 2+*5

>>> 2-1 * 5

>>> (2 - 1)*5

>>> 13 / 4

>>> 13 // 4

>>> 13 % 4

>>> 5 / (2 - 2)

>>> 2**10

>>> 1,2 + 1,3

```

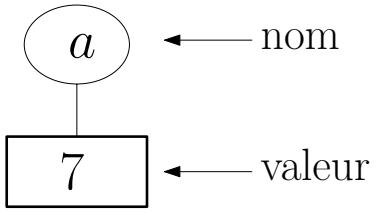
1

II Affectation des variables

Pour stocker et manipuler les informations élémentaires on utilise des **variables** qui associent à un nom une valeur : une variable est une boîte qui porte un nom.

Pour donner une valeur à une variable on utilise une instruction fondamentale : **l'affectation**. Sa syntaxe utilise le symbole = sous la forme :

nom_variable = expression




```
>>> a = 7
>>> a
7
```

2

Remarques :

Ici, l'utilisation du symbole d'égalité n'a rien à voir à celle des mathématiques où il énonce une relation [2]. De plus, dans le contexte des fonctions, « variable » signifie « qui peut prendre une valeur quelconque » mais en informatique « variable » indique que la valeur associée est susceptible de *varier* lors de l'exécution du programme.

En Python, le symbole = sert à décrire **une action** [3] : le membre de droite est évalué et la valeur obtenue est affectée à la variable dont le nom constitue le membre de gauche. Si elle n'existe pas déjà, celle-ci est alors créée. Sinon son ancienne valeur est remplacée par la nouvelle.

En conséquence, il ne se comporte pas de manière symétrique :

```
>>> 7 = a
SyntaxError: can't assign to literal
```
3

Le **nom de la variable** peut comporter plus d'une lettre mais il faut respecter quelques règles :

- On peut utiliser les lettres minuscules ou majuscules qui sont différentiées ainsi que les 10 chiffres ($0 \to 9$). Les autres caractères sont interdits sauf le tiret bas _ (le tiret usuel - est interdit car c'est le symbole de la soustraction).
- Le premier caractère est obligatoirement une lettre.
- Les mots réservés du language sont interdits :

```
  >>> else = 5
SyntaxError: invalid syntax
```
4

- Utiliser un nom de fonction ne provoquera pas d'erreur directe mais c'est absolument à éviter :

```
>>> print('Hello Word')
Hello Word
>>> print = 1789
>>> print('Hello word')
Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    print('Hello word')
TypeError: 'int' object is not callable
```
5

Bonne pratique : En fonction des circonstances dans le code, il est recommandé d'utiliser des noms de variables indiquant leurs rôles.

Définition (Expression)

Une **expression** est une combinaison de valeurs explicites (appelées littéraux), de variables, d'opérateurs et de fonctions qui est évaluée pour donner lieu à une valeur.

2. Par exemple : $a(b+c) = ab + ac$.
3. En pseudo-code on écrira par exemple : a $\leftarrow$ 5.

Lors de l'exécution d'une instruction d'affectation, il y a d'abord évaluation de l'expression puis la valeur obtenue est attribuée à la variable.

```
>>> a = -2
>>> a
-2
>>> b = a*(a + 1)/2
>>> b
1.0
>>> a = min(-1, b, 5)
>>> a
-1
```

6

Exercice 2

Déterminer si les suites de symboles sont des expressions ou des instructions.

* x + 1/(x*x +1)
* x = y + 1
* x < y + 1
* Flag = y >= x + 1

Entraînement 1

Déterminer si les suites de symboles sont des expressions ou des instructions.

* x == y + 1
* x = y + 1
* x <= y + 1
* estPremier = nbDiviseur(n) == 2

Exercice 3

Tester, expliquer :

a)

```
>>> a - 1 = 5
```

7

---

b)

```
>>> u = v
```

8

Exercice 4

Quelle instruction provoquera l'incrémentation de la variable a ? Sa décrémentation ? Son doublement ?

Connaissez-vous une écriture courte qui donne le même résultat ?

Exercice 5 Pour permuter les valeurs de deux variables, on essaie le code suivant :

```
>>> x = 1
>>> y = 100
>>> y = x
>>> x = y
>>> x
1
>>> y
1
```

9

Expliquer ce qu'il s'est passé :

Comment faire pour remédier à cette erreur ?

Entraînement 2

En utilisant une seule variable auxiliaire (`aux`) et sans utiliser de tuples, effectuer la permutation circulaire des variables des `x`, `y`, `z`, `t` selon le schéma : `x` ⟶ `y` ⟶ `z` ⟶ `t` ⟶ `x`

Exercice 6

Prévoir le résultat des instructions :

```
>>> a = 5
>>> b = a + 2
>>> c = a + b
>>> a = 3
>>> c = a - b
>>> (a, b, c)
```

10

```
>>> a = 1
>>> b = 4 - a
>>> b = 2*b + 1
>>> c = (a + 2)*b
>>> a = -b
>>> (a, b, c)
```

11

Définition (État courant d'un programme)

Lors de l'exécution d'un programme, l'ensemble des variables à un instant donné s'appelle l'état d'exécution ou l'état courant du programme.

Suivre l'évolution de l'état courant d'un programme permet souvent de le comprendre en profondeur et de le mettre au point. On peut utiliser un tableau comme ci-contre.

ligne	état courant après		
	a	b	c
1			

Le site pythontutor vous permet de visualiser l'état courant de vos programmes.

`http://www.pythontutor.com/`

Entraînement 3

On suppose que les variables `x`, `y` et `z` contiennent respectivement des valeurs entières a, b et c. À l'aide d'un tableau d'état, déterminer ce que fait l'ensemble des instructions suivantes :

```
>>> y = y - z
>>> z = y + z
>>> y = y - z
>>> x = x - y
>>> y = x + y
>>> x = x - y
```

12

Exercice 7

On suppose que, dans l'état courant, les variables x, y, et z ont respectivement pour valeur 2, -1 et 5. Évaluer les expressions suivantes :

(a) x + 1
(b) y ** 2 == x - 1
(c) x <= z
(d) x * y + 1
(e) x ** y
(f) x+2 * y

III Type

Définition (Type)

Chaque variable possède un type qui est déterminé au moment de l'affectation mais il peut changer (on parle de typage dynamique). Le type indique la nature des données qu'une variable contient.

Les types Python que nous utiliserons le plus souvent sont :

type Python	"traduction"	exemple
`int`	entier	1871
`float`	flottant (décimal)	3.1415
`bool`	booléen (`True` ou `False`)	`True`
`tuple`	n-uplet	`(2, 0, 1, 7)`
`str`	chaîne de caractères (string)	`'Python rocks'`
`list`	liste	`['toto', 5, 2.71, False]`
`function`	fonction	`max`

On obtient le type d'une variable par la fonction **type** :

```
>>> x = 101
>>> type(x)
<class 'int'>
>>> x = 101.
>>> type(x)
<class 'float'>
>>> y = 101
>>> type(x == y)
<class 'bool'>
>>> x == y
True
>>> type(False)
<class 'bool'>
```

13

```
>>> a = (1, 3.14159)
>>> type(a)
<class 'tuple'>
>>> ch = 'Hello word'
>>> type(ch)
<class 'str'>
>>> L = [1, 'abcdefgh', 2.71]
>>> type(L)
<class 'list'>
>>> def carre(x):
        return x*x

>>> type(carre)
<class 'function'>
```

14

Remarque : Python est un langage orienté objet. Tout y est objet. Les objets même les plus simples ont des classes ce qui explique l'apparition du terme **class** là où attendrait plutôt **type**. Le terme de type est plus général.

Noter l'utilisation du caractère ' pour délimiter les chaînes de caractères.

Exercice 8 Prévoir le type des expressions suivantes :

(a) (x, y) == (1, b)
(b) 'u == 3'
(c) 1+2.
(d) 3.14
(e) 3,14
(f) [3,14]

Quelques opérateurs usuels

On notera que certains opérateurs peuvent s'appliquer à des objets de types différents ; ils sont polymorphes.

Opérateur	Type	Action
+	int, float	Addition
-	int, float	Soustraction
*	int, float	Multiplication
**	int, float	Puissance
/	float	Division décimale
//	int	Quotient de la division euclidienne
%	int	Reste de la division euclidienne
+	str, list	Concaténation de deux chaines de caractères ou listes
not	bool	Négation (False ↔ True)
and	bool	Et logique
or	bool	Ou logique

Les règles de priorités usuelles s'appliquent. Si on a le moindre doute, le parenthèsage est de rigueur.

On rappelle que les fonctions mathématiques usuelles sont utilisables à condition de les importer :

```
>>> from math import cos, pi
>>> cos(pi/3)
0.5000000000000001
```
15

Pour **convertir** d'un type vers un autre on utilise une fonction dont le nom est le type vers lequel on veut convertir :

```
>>> str(1789)
'1789'
>>> int('1789')
1789
>>> float('3.14159')
3.14159
```

```
>>> bool(0)
False
>>> bool(5)
True
>>> list('2017')
['2', '0', '1', '7']
```
16

III.1 Les booléens

Ils ne peuvent avoir que deux valeurs : True ou False.

Les booléens sont souvent obtenus à l'aide des opérateurs de comparaison.

Opérateur	test effectué
==	égalité
!=	différent
<=	inférieur ou égal
>=	supérieur ou égal
<	inférieur strict
>	supérieur strict

Il est à noter que ces opérateurs sont extrêmement polymorphes ; Python évalue en booléen parfois là où l'on attendrait une erreur, ce qui peut rendre cette erreur difficile à détecter. Attention à bien faire la différence entre = et == :

x = y est une instruction qui est exécutée et x == y est une expression qui est évaluée.

On peut chainer les comparaisons :

```
>>> 4>2<8
True
>>> 1 < 5 < 2
False
```
17

Les opérateurs logiques permettent de construire des expressions complexes :

```
>>> a = 5
>>> not(1 < a < 4) or ( a < 0)
True
```
18

Exercice 9 Quelle est la valeur des expressions booléennes suivantes lorsqu'elles sont correctes ?

(a) 3*1.5 > 4

(b) 2.* 8 == 16

(c) 3 - 1 => 3

(d) 0 < 10**5 = 100000

(e) **not**(True **or** False)

(f) 'tric' + 'trac' == 'TricTrac'

III.2 Les n-uplets

Les n-uplets (*tuple* en anglais) généralisent la notion de couple ou de triplet. Il correspondent au produit cartésien d'ensemble en math. Pour définir un tuple en Python, on utilise les parenthèses et la virgule.

```
>>> t = (2 , 3 , 5 , 7)
>>> t + (11, 13)  # Concaténation
(2, 3, 5, 7, 11, 13)
>>> t[0]   # Accès aux composantes
2
>>> t[6]
Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    t[6]
IndexError: tuple index out of range
>>> 3 in t   # Test d'appartenance
True
>>> len(t)    #Longueur
4
>>> u = ('MPSI', 2.4, 1 == 0.5 * 2, (1, 2))
>>> u[2]
True
>>> u[3]    #  un tuple peut être dans un tuple
(1, 2)
>>> type(())   # Il existe un tuple vide
<class 'tuple'>
```
19

Les tuples ne sont pas modifiables, on dit aussi qu'ils sont immuables.

```
Traceback (most recent call last):
  File "<pyshell#34>", line 1, in <module>
    u[0] = 'PCSI'
TypeError: 'tuple' object does not support item assignment
```

20

Pour déconstruire un tuple, on utilise une affectation :

```
>>> (x, y) = A
>>> x
1
>>> y
5.5
>>> (_, v) = A  # Si on ne veut qu'une composante
>>> v
5.5
```

21

IV Un exemple de programme

En mode programme, pour faire afficher un nombre ou une chaîne de caractères (dans la console), on utilise l'instruction `print` et pour permettre à l'utilisateur de saisir une chaîne de caractères, on utilise `input` :

```
print('Calcul de la somme de deux nombres')
premier = input('Entrez le premier nombre : ')
second = input('Entrez le second : ')
a = int(premier)
b = int(second)
s = a + b
print('La somme est : ', s)
```

22

Exercice 10

Adaptez le programme précédent pour calculer l'aire d'un triangle.

V Ressources en ligne et installation de Python

Quelques liens vers des ressources :

- Site officiel : http ://www.python.org/ http ://www.python.org/download/ http ://www.python.org/doc/
- Cours sur Python : http ://inforef.be/swi/python.htm ou http ://python.developpez.com/cours/
- Tutoriel sur Python : http ://www.siteduzero.com/tutoriel-3-223267-apprenez-a-programmer-en-python.html

— Le site de l'Olympiade française d'informatique (France-ioi) :
http ://www.france-ioi.org/algo/chapters.php.
— Le site compagnon du livre de spécialité NSI chez Ellipses : `https://www.nsi-première.fr`

Python est installé par défaut sur la distribution Linux Ubuntu, sous Windows ou MAC OSX on peut télécharger les installateurs à partir de http ://www.python.org/download/ et on disposera d'un environnement graphique Idle avec console et éditeur de texte.