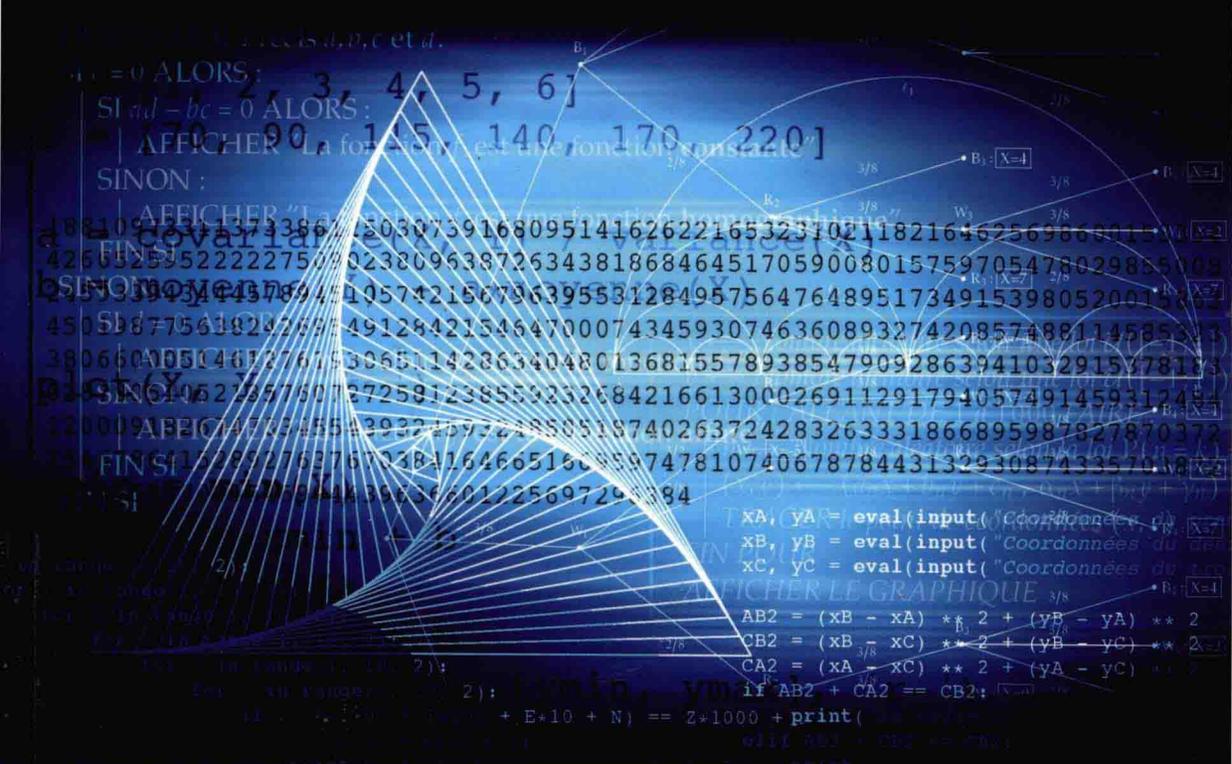


Python

Les bases de l'algorithmique et de la programmation

Vincent Maille



Python

Les bases de l'algorithmique
et de la programmation



Du même auteur, chez le même éditeur

Apprendre la programmation par le jeu - A la découverte du langage Python 3 - 2^e édition, 240 pages, 2015.

ISBN 9782340-004603

©Ellipses Édition Marketing S.A., 2015

32, rue Bargue 75740 Paris cedex 15



Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5.2° et 3°a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit constituerait une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

www.editions-ellipses.fr

Avant-propos

Pourquoi ce livre ?

Ayant enseigné pendant de nombreuses années en lycée en mathématiques et plus récemment en ISN, j'ai été amené à créer, modifier, expérimenter des exercices mathématiques et surtout algorithmiques variés. Ce livre est donc le fruit de mon travail mais surtout des échanges très riches entre collègues autour d'une machine à café, dans des stages de formation et bien sûr avec mes élèves autour de l'apport de l'algorithmique à l'enseignement au lycée.

Que vous soyez enseignant débutant, pédagogue à la recherche d'activités liant l'algorithmique aux mathématiques, lycéen, étudiant ou simplement intéressé par les sciences, vous aurez plaisir à découvrir un langage informatique puissant tout en résolvant des problèmes mathématiques.

C'est une des raisons pour laquelle la progression de ce livre se veut informatique et non mathématique. Ainsi dans chaque chapitre, vous retrouverez des exercices pour les différentes classes du lycée, de la seconde au BTS. Pour une entrée par thème mathématique et par classe, reportez-vous au tableau synthétique regroupant les exercices qui se trouvent à la fin de l'ouvrage.

Certains de ces exercices sont sûrement un peu ambitieux, mais vous disposez à chaque fin de chapitre d'un paragraphe "coup de pouce" qui vous donnera des pistes si vous êtes bloqués ainsi qu'un paragraphe "solutions" pour une correction complète. N'hésitez pas à lire ces solutions même si vous avez réussi l'exercice. Elles sont volontairement très détaillées et apportent souvent de nombreux compléments à la partie cours du chapitre volontairement assez synthétique.

Pourquoi Python ?

Avec l'arrivée des nouveaux programmes dans les classes préparatoires aux grandes écoles scientifiques, l'informatique a vu le jour comme une matière à part entière. Le langage Python a été retenu par la commission des programmes pour travailler l'algorithmique dans ces classes. Parmi les nombreux avantages de ce langage, on peut citer :

- Un langage libre et gratuit.
- Un langage multiplateforme, c'est-à-dire que les programmes que vous allez réaliser pourront tout aussi bien s'exécuter sous Windows, Mac ou Linux.
- Un langage simple et cependant puissant et rapide.
- Un langage qui possède de nombreuses bibliothèques mathématiques très évoluées dans de multiples domaines (traitement de l'image, analyse de Fourier, calcul numérique...) et qui permettent ainsi de réaliser facilement de nombreux programmes assez pointus.

Remerciements

Tout d'abord, merci à mes fidèles relecteurs qui ont pris le temps de m'aider à l'élaboration de ce livre par leurs conseils pédagogiques ou d'écriture. Merci donc à Christiane et Jean-Louis MAILLE, ainsi qu'à mes collègues Fatima ESTEVENS, Guillaume MIANNAY et Paul STIENNE pour leurs lectures attentives

Je tiens à remercier aussi le groupe de travail académique **AmiensPython**. Dans un premier temps, nous avons mis en place le projet AmiensPython ainsi qu'une documentation pour l'enseignant. Puis, nous avons fait évoluer la version qui est devenue EduPython, pour laquelle Agnès BARAQUIN a réalisé une mise à jour de la documentation.

De nombreux sites internet regorgent d'articles plus passionnants les uns que les autres. Le site <http://www.diophante.fr/> avec plus de 1500 problèmes mathématiques de tous niveaux, en est un très bon exemple. Merci à l'auteur Philippe FONDANAICHE pour l'autorisation de reprendre quelques énoncés ici traités à l'aide de l'outil informatique.

Utilisation du livre

L'objectif de ce livre est de proposer des problèmes plutôt ouverts qui peuvent être résolus à l'aide du cours de mathématique de lycée et de l'outil informatique. Les énoncés sont de nature assez variée, ainsi, selon les compétences que vous cherchez à travailler, vous pourrez choisir des exercices en vous appuyant sur les symboles qui précèdent les énoncés

- pour résoudre des problèmes mathématiques
- pour développer un raisonnement informatique
- pour apprendre à utiliser certaines fonctionnalités de Python

Chaque exercice est repéré par un nombre d'étoiles indiquant la difficulté mathématique :

- | | |
|---|--|
|  | Exercice sollicitant des connaissances du niveau collège |
|  | Exercice accessible dès la classe de seconde du lycée |
|  | Exercice nécessitant des connaissances de première ou de terminale |
|  | Exercice faisant appel à de nombreuses connaissances du lycée |

Dans le codage précédent le nombre de branches des étoiles permet de repérer la difficulté informatique :

- | | |
|---|--|
|  | Facile |
|  | De difficulté moyenne |
|  | Difficile |
|  | Pour ceux qui aiment relever les défis ! |

Les éléments de cours sont repérés par la tache d'encre  , le symbole  indique des remarques sur les différentes versions de Python, comme nous le verrons dans le prochain chapitre.

Une des principales particularités de ce livre est la possibilité d'utiliser le complément en ligne sur le site :

<http://prototheque.free.fr/ellipses>



En saisissant les **CODES** signalés sur fond gris, vous pourrez télécharger les différents codes sources des programmes proposés. Tous les exercices y sont corrigés.

Bonne lecture et bonne programmation !

Table des matières

A Bien démarrer	11
1 - Installation de Python sur votre machine	11
a) Téléchargement	11
b) EduPython, AmiensPython, Python 2.x, Python 3.x???	11
2 - A la découverte de la console Python	12
a) Interface EduPython	12
b) Variables et affectations	15
c) Affectations simultanées	17
3 - Les types d'objets	18
a) Les entiers	19
b) Les flottants	19
c) Les chaînes de caractères	19
d) Les booléens	20
4 - Aide pour les exercices	21
5 - Solutions des exercices	21
6 - Compléments : flottants, décimaux et réels	24
B Utiliser des scripts	27
1 - Python, un langage interprété	27
a) Créer	28
b) Enregistrer	28
c) Exécuter	28
d) Commenter	28
2 - Entrée & Sortie	29
a) Afficher	29
b) Interagir avec l'utilisateur	30
3 - Aide pour les exercices	33
4 - Solutions des exercices	34
C Tester	37
1 - Si ... alors ...	37
2 - Opérateurs de comparaison	39
3 - Si ... alors ... sinon ...	40
4 - Disjonction de cas : SINON, SI...	42

5 - Opérateurs logiques	44
6 - Exercices	44
7 - Aide pour les exercices	46
8 - Solutions des exercices	46
9 - Complément : les booléens, un nouveau type	54
D Boucles	57
1 - Cours	57
a) Boucle conditionnelle TANT QUE	57
b) Boucle inconditionnelle POUR	60
2 - Exercices	62
3 - Aide pour les exercices	65
4 - Solutions des exercices	66
E Utilisation de modules	77
1 - Cours	77
a) Différentes manières d'importer un module	77
b) Le module math	78
c) Le module random	84
d) Le module lycee d'EduPython	85
2 - Exercices	86
3 - Aide pour les exercices	89
4 - Solutions des exercices	90
F Les fonctions	103
1 - Cours	103
2 - Exercices	106
3 - Aide pour les exercices	110
4 - Solutions des exercices	111
G Les chaînes	125
1 - Cours	125
a) Présentation	125
b) Les fonctions	126
c) Code ASCII	127
2 - Exercices	128
3 - Aide pour les exercices	131
4 - Solutions des exercices	134
H La tortue	145
1 - Cours	145
2 - Exercices	148
3 - Aide pour les exercices	154
4 - Solutions des exercices	155

I	Listes	171
1 -	Cours	171
2 -	Exercices	173
3 -	Aide pour les exercices	179
4 -	Solutions des exercices	181
5 -	Complément : les listes sont des alias	195
J	Graphismes	199
1 -	Fonctions de base	199
2 -	Exercices du chapitre	201
3 -	Aide pour les exercices	209
4 -	Solutions des exercices	211

Liste des thèmes d'étude :

Approcher le nombre $\sqrt{2}$	99
Nombres parfaits & nombres amicaux	124
Cryptographie	141
Le flocon de Van Koch	167
Les nombres et leurs empans	197

Chapitre A

Bien démarrer

1 - Installation de Python sur votre machine

a) Téléchargement

Vous trouverez sur le site du livre : <http://prototheque.free.fr/ellipses/> en tapant le code `INSTALLATION`, un tutoriel vidéo présentant l'installation de la distribution EduPython qui sera utilisée dans ce livre. Pour installer EduPython, il suffit d'aller dans la rubrique téléchargement du site officiel : <http://edupython.tuxfamily.org/> puis de suivre les instructions.

La distribution EduPython utilise l'excellent éditeur PyScripter qui ne fonctionne malheureusement que sous windows. Si vous utilisez un autre système d'exploitation, vous n'aurez qu'à utiliser n'importe quelle distribution Python sous Python 3.x. Par exemple, la distribution PYZO disponible sur le site : <http://www.pyzo.org/> est un produit très complet et multiplateforme. Dans tous les cas, les programmes présentés dans ce livre ne dépendent pas du système d'exploitation que vous utilisez, c'est là tout l'intérêt de Python.

b) EduPython, AmiensPython, Python 2.x, Python 3.x ???

Expliquons ce qui se cache derrière ces mots....

Il existe sur le marché actuel deux versions majeures de Python : Python 2 (la dernière étant la 2.7) et Python 3 (la version actuelle est la 3.4). Le passage de Python 2 à Python 3 a induit quelques changements majeurs (par exemple la définition de la division n'est plus la même) rendant malheureusement incompatibles les programmes de la version 2 à la version

3. A l'heure actuelle Python 2 est encore assez utilisé, mais n'est plus mis à jour. Ce livre traite donc de Python 3, le langage qui est développé à ce jour.

En ce qui concerne AmiensPython, EduPython et Pyzo dont nous avons déjà parlé, il s'agit de distributions Python (un package comportant Python, un éditeur de programmes, ...) permettant d'utiliser Python.

Avec l'arrivée de l'algorithmique dans l'enseignement des mathématiques au lycée, un groupe de travail de l'académie d'Amiens constitué de 4 enseignants, Agnès BARAQUIN, François PREDINAS, Julien POLLET et moi-même sous la direction de Ludovic LEGRY, IA-IPR, a produit une distribution clé en main, portable et francisée pour l'usage de Python en classe. Cette distribution porte le nom d' AMIENSPYTHON , elle fonctionne sous Python 2.7. Une brochure à usage pédagogique a été réalisée comportant de nombreux exemples d'application.

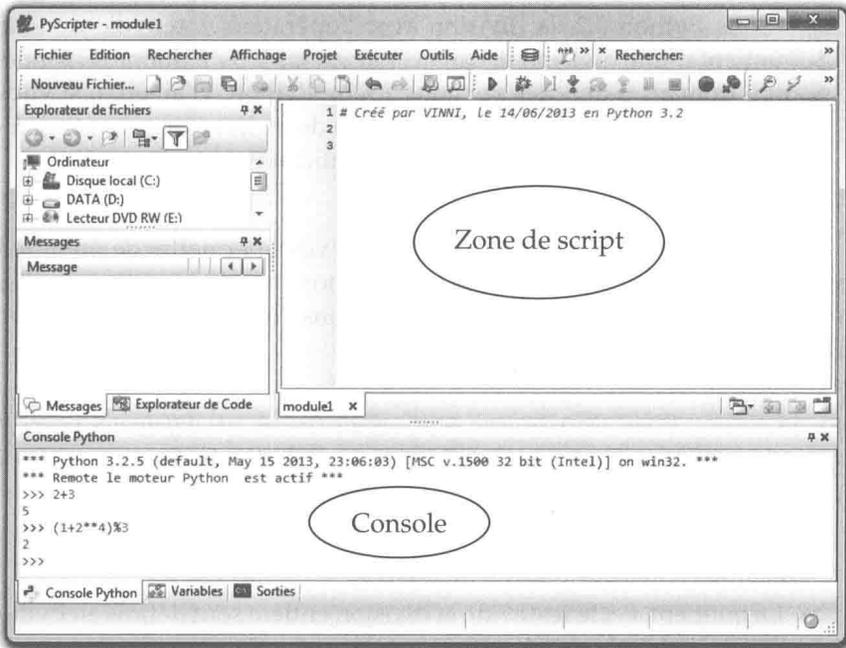
Avec l'arrivée de Python dans les programmes du supérieur, il nous a semblé pertinent de proposer une version d'AmiensPython fonctionnant sous Python 3. Edupython a donc été créé comme un outil permettant de programmer sous Python 3.2.

2 - A la découverte de la console Python

a) Interface EduPython

Le logiciel PyScripter fait office d'éditeur pour EduPython. Si vous n'avez pas pu installer Edupython, les autres éditeurs ont un fonctionnement analogue. L'interface est découpée en plusieurs zones, en particulier :

- La zone de script qui permet de saisir des programmes plus longs et de les exécuter ensuite.
- La console qui permet d'exécuter en direct des commandes Python ; c'est aussi dans cette console que s'afficheront les résultats des programmes.



Dans la zone console de Python, on peut donc saisir des instructions qui seront exécutées immédiatement. Cette saisie se fait après les `>>>` que l'on appelle des **chevrons**.

Ainsi, on peut déjà, dans un premier temps, utiliser cette console comme une calculatrice. Voici quelques opérations possibles :

Opérateur	Effet
$+$, $-$, $*$ a / b	Respectivement la somme, la différence et le produit Quotient décimal de a par b
$a // b$ $a \% b$	Quotient entier de a par b Reste entier dans la division de a par b
$a ** b$	Résultat de a^b . Attention, le symbole $^$ en Python a une autre signification.

 Sous Python 3.2, la division avec l'opérateur / renvoie donc le quotient d'une division décimale, par exemple 7/2 donne 3.5 alors qu'il donnait 3 avec les versions 2.x de Python. D'ailleurs, si vous utilisiez AmiensPython, cette définition de la division qui nous semblait plus pédagogique avait déjà été anticipée.

Les quelques exemples qui suivent vont vous permettre de prendre la console Python en main en jouant avec les nombres. Vous pouvez utiliser les touches  et  pour vous déplacer dans l'historique de ce que vous avez déjà tapé au fur et à mesure.

Ex A1  *Ecrire deux séquences de calculs (vraiment) différentes pour afficher 35 en utilisant exactement 5 fois la touche  et autant que nécessaire les touches , , , , ,  et .*

Quelques remarques supplémentaires :

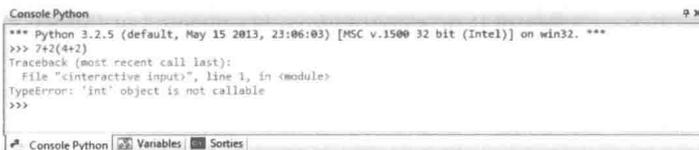
- Le quotient q et le reste r de la division entière sont définis en Python grâce au théorème suivant :

SI a et b sont deux nombres entiers relatifs avec b non nul
ALORS il existe un unique couple d'entiers (q, r) vérifiant :

$$\begin{cases} a = b \times q + r \\ |r| < |b| \\ r \text{ et } b \text{ sont de même signe} \end{cases}$$

Nous verrons un peu plus loin qu'il faudra être vigilant lorsque l'on utilise des nombres négatifs.

- Le calcul de $a * * b$ peut être effectué même lorsque b n'est pas entier. Dans ce cas, si $a > 0$ et $b \in \mathbb{R}$, Python utilise la formule vue en terminale : $a^b = e^{b \ln a}$ (si $a < 0$, c'est abordable dans l'ensemble des nombres complexes mais ne figure pas dans le programme du lycée).
- Le calcul suivant : $7+2(4+2)$ n'a pas de sens : Python n'interprète pas les multiplications implicites, il faut donc taper $7+2*(4+2)$.



```

Console Python
*** Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC v.1500 32 bit (Intel)] on win32. ***
>>> 7+2(4+2)
Traceback (most recent call last):
  File "c:\interactive input", line 1, in <module>
TypeError: 'int' object is not callable
>>>

```

Si vous connaissez les puissances d'exposant réel, vous pouvez faire l'exercice qui suit :

Ex A2  Qu'affichent les calculs suivants ?

A. $16 ** 0.5$

C. $27 ** 1 / 3$

B. $16 ** 0.25$

D. $27 ** (1 / 3)$

 Comme en mathématiques, Python respecte les priorités des opérations. En informatique, ce concept porte le nom de **précédence** des opérateurs. Dans le cas du langage Python, les précédences sont définies ainsi (dans l'ordre croissant) :

$+, -$	Addition et soustraction
$*, /, //, \%$	Multiplification, division, quotient et reste
$+x, -x$	Positif, négatif
$**$	Exponentiation

Ex A3  Qu'affichent les calculs suivants ? Après avoir cherché les réponses, ne pas hésiter à tester avec la console pour vérifier.

A. $7 // 2$

E. $-14 // (-3)$

I. $1 - 5 * 2$

B. $4 \% 2$

F. $5 ** 0 + 7$

J. $5 - 3 / - 2$

C. $3 ** 3$

G. $2 ** - 2$

K. $- 13 \% 3$

D. $-5 // 2$

H. $-2 ** 10$

L. $13 \% - 3$

b) Variables et affectations

L'avantage d'utiliser un langage de programmation pour effectuer des calculs réside dans le fait de pouvoir stocker les informations en mémoire. On peut, lorsque l'on réalise un programme ou un algorithme, utiliser des variables pour stocker les valeurs. Le fait de donner un nom à un calcul, un texte ou un autre objet s'appelle **l'affectation**.



En Algorithmique

$$a \leftarrow 3$$

On lit ...

a reçoit la valeur 3

En Python

$$a = 3$$

Dans la console Python, lorsque l'on affecte une valeur à une variable, le contenu de celle-ci n'est pas affiché. Il faut taper le nom de la variable pour afficher sa valeur.

Au moment d'une affectation, la quantité à droite du signe = est évaluée et stockée dans la variable de gauche. L'écriture $a = a + 1$ a donc un sens en informatique. Cela signifie simplement que la variable a a augmenté de 1.

Le symbole = joue donc un rôle d'affectation et non d'égalité. D'ailleurs, $b = a * 2$ n'est plus vérifiée à la fin de notre exemple.

```
>>> a = 3
>>> b = a * 2
>>> b
6
>>> a + 1
4
>>> a
3
>>> a = a + 1
>>> a
4
>>> b
6
```

Voici donc un petit exercice permettant de travailler le vocabulaire mathématique et les opérations avec Python...

Ex A4  Si n est un entier positif, relier les expressions Python aux expressions françaises correspondantes :

- | | |
|--------------------|---|
| • $n + 1$ | • Le chiffre des unités de n |
| • $n // 10 \% 10$ | • Le premier nombre impair qui suit n |
| • $n // 10$ | • Le chiffre des dizaines de n |
| • $n // 2 * 2 + 1$ | • Le nombre de dizaines de n |
| • $n ** 2$ | • Le carré de n |
| • $n \% 10$ | • Le double de n |
| • $n * 2$ | • L'entier qui suit n |

En vous appuyant sur l'exercice précédent, vous pouvez alors tenter celui-ci.

Ex A5 

1. Si n est une variable contenant un nombre à 2 chiffres non nuls, écrire une suite d'instructions permettant de stocker dans une variable p le nombre écrit "à l'envers" : si $n = 12$ alors $p = 21$.
2. Même question avec un nombre à (exactement) 3 chiffres.