Pascal Lafourcade, Guenaëlle De Julis et Malika More

15 ÉNIGMES LUDIQUES POUR SE PERFECTIONNER EN PROGRAMMATION PYTHON

DUNOD

Découvrez aussi :

- P. Lafourcarde et M. More, *25 énigmes ludiques pour s'initier à la cryptographie*, Dunod, 2021.
- P. Lafourcarde et M. More, *15 énigmes ludiques pour s'initier à la programmation Python*, Dunod, 2022.
- J.-G. Dumas, P. Lafourcade, E. Roudeix, A. Tichit et S. Varrette, *Les NFT en 40 questions*, Dunod, 2022.
- J.-G. Dumas, P. Lafourcade, A. Tichit et S. Varrette, *Les blockchains en 50 questions*, 2e éd., Dunod, 2022.
- J.-G. Dumas, P. Lafourcade, P. Redon, *Architectures de sécurité pour Internet*, 2e éd., Dunod, 2020.
- J.-G. Dumas, J.-L. Roch, S. Varrette, E. Tannier, *Théorie des codes : Compression, cryptage, correction*, Dunod, 2018.
- D. Vergnaud, *Exercices et problèmes de cryptographie*, 4e éd. ? Dunod, 2023

Direction artistique : Nicolas Wiel

11 rue Paul Bert, 92240 Malakoff
www.dunod.com
ISBN 978-2-10-086332-7

Sommaire

Avant-propos

Ces 15 énigmes ludiques vont vous faire découvrir des concepts importants de l'informatique. Elles nécessitent plus ou moins de logique, de réflexion et d'astuce, mais il est surtout indispensable de réaliser des programmes informatique pour les résoudre. Ce livre s'adresse à des lycéens scientifiques et étudiants en cursus informatique. L'objectif est de proposer des énigmes dont la solution peut être obtenue à l'aide d'un programme. Il fallait donc choisir un langage pour présenter les solutions, mais il est bien entendu possible de programmer dans votre langage de programmation favori pour résoudre ces énigmes. En début d'ouvrage, vous trouverez quelques pages de description des principales commandes en Python 3 utiles pour la résolution des énigmes.

Pour chaque énigme, trois niveaux progressifs d'indices se trouvent dans un chapitre séparé au milieu du livre. Ainsi, si après avoir commencé à réfléchir, vous êtes bloqué, vous trouverez grâce à ces indices une aide graduée pour vous donner un coup de pouce et vous mettre sur la piste de la solution.

Les niveaux de difficulté des énigmes sont indiqués par des étoiles. Le niveau le plus facile, concernant les énigmes qui vous permettront de vous échauffer, est représenté par ☆. Pour les résoudre, il faudra écrire des programmes Python 3 assez simples.

Le niveau intermédiaire, dont les énigmes reposent sur une réflexion et des calculs plus complexes, est représenté par ☆☆. Il arrive que la solution repose sur une astuce un peu moins évidente que dans le premier niveau, les programmes à écrire pour ces énigmes étant un peu plus complexes.

Le niveau ☆☆☆ est le niveau le plus difficile. Il comporte des énigmes qui nécessitent beaucoup de réflexion ou qui demandent des connaissances en mathématiques et en informatique un peu plus avancées. Il faudra faire preuve de ténacité pour concevoir ces programmes afin de trouver la solution.

Les thèmes des énigmes sont l'occasion de découvrir de nombreux concepts importants en informatique. La plus grande partie de cet ouvrage est constituée des solutions détaillées de toutes les énigmes. Chaque solution contient non seulement le programme Python 3 amenant à la résolution de l'énigme, mais aussi des explications détaillées sur la conception de l'algorithme corres-

pondant. En guise de clin d'œil, chaque solution est accompagnée d'une citation scientifique ou littéraire en rapport avec l'énigme ou sa solution.

L'aspect ludique de ce livre motivera certains lecteurs pour coder les solutions et les poussera à faire preuve de créativité pour écrire des programmes résolvant ces énigmes.

Cet ouvrage s'adresse à un public qui aime programmer pour découvrir et apprendre des concepts importants de l'informatique. Il s'adresse aussi, indirectement, à tous les enseignants d'informatique, car ces énigmes sont une banque d'exercices corrigés au même titre qu'un manuel de cours.

Les énigmes, les indices et les solutions contiennent de nombreux encarts : biographiques, historiques, techniques, mathématiques, Python 3, culturels, de solution de codes Python 3, d'objectifs pédagogiques pour les enseignants en rapport avec les concepts abordés ou encore des pistes pour aller plus loin. Ils sont représentés respectivement par ces icônes :

Les fichiers des énoncés des énigmes, des programmes des solutions et des compléments en Python 3 sont disponibles à l'adresse suivante :

http://dunod.link/kl9s5z5

Remerciements : Nous remercions Cédric Lauradoux de nous avoir montré la voie pour la création de ces énigmes. Nous adressons aussi nos remerciements à Flavien Binet, Emmanuel Delay, Diane Leblanc-Albarel, Christine Solnon, Thibault Ralet pour leurs contributions à l'élaboration du contenu de ce livre. Nous exprimons également notre gratitude à Matthieu Daniel et Anne Le Duc pour leurs commentaires et suggestions constructives, à la suite de leurs relectures assidues.

Clermont-Ferrand, le 12 septembre 2023.
Malika More, Guenaëlle De Julis, et Pascal Lafourcade *.

* Nous serons heureux de répondre à vos questions par email.

Chaque problème a sa solution, il y a donc des codes pour tous les goûts.

Une brève introduction à Python 3

Dans cet ouvrage, les solutions des énigmes sont données en Python 3 à titre d'illustration, un logiciel libre [#]. Il est bien entendu possible de rédiger les solutions dans le langage de programmation de votre choix, puisqu'un langage de programmation comprenant des instructions pour effectuer des calculs arithmétiques, des affectations de variables, des instructions conditionnelles et des boucles (`while`) permet de programmer n'importe quel algorithme.

1 JAN Origine de Python

Le langage Python a été créé par Guido van Rossum en février 1991. À cette époque, il lisait un recueil des sketchs de la série télévisée comique britannique *Monty Python's Flying Circus*, diffusée sur la BBC dans les années 1970. Il cherchait pour son langage de programmation un nom court, unique et légèrement mystérieux, et il choisit « Python », en l'honneur de cette émission dont il était fan.
Concernant les versions successives du langage Python, il est important de savoir que la version 2 n'est pas compatible avec la version 3, car de nombreux changements de syntaxe ont été faits. De plus, depuis 2020, la version Python 2 n'est plus mise à jour.

Figure 1 – Guido van Rossum.

Les commandes les plus utilisées dans les solutions des différentes énigmes sont présentées et, dans certains cas, des précisions supplémentaires sont données dans les énoncés ou dans les indices. Pour plus de détails sur Python 3, et pour apprendre à programmer en Python 3 de façon organisée, il existe de nombreux ouvrages et des sites de cours dédiés. Le site de référence (mais qui n'est pas le plus simple pour débuter), est le manuel officiel accessible en ligne `https://docs.python.org/fr/3/`. Si vous êtes déjà à l'aise avec Py-

[#] `https://www.python.org`

thon 3, vous n'aurez probablement pas besoin de vous référer à ce chapitre. Sinon, n'hésitez pas à le parcourir pour chercher des idées lorsque vous manquez d'outils pour programmer la solution d'une énigme.

Commenter tout ou partie d'une ligne

Tout d'abord, notez que le symbole # permet, en Python 3, de commenter le reste de la ligne, ce qui signifie que tout ce qui est écrit après ce symbole sur la même ligne n'est pas pris en compte lors de l'exécution du programme.

Syntaxe et bonnes pratiques

Des règles sont définies dans PEP [†] afin d'homogénéiser le code et d'appliquer des bonnes pratiques qui facilitent le travail sur des projets à plusieurs. Ces règles ne changent pas l'exécution d'un programme. Ces conventions de style, dont les principales sont présentées ci-dessous, sont définies dans PEP8.

Tout d'abord, il est essentiel de savoir que les indentations en Python 3 jouent un rôle crucial et que leur présence ou leur absence aux bons endroits peut modifier le comportement d'un programme, voire générer des erreurs. La proposition PEP8 recommande d'utiliser une indentation de quatre espaces.

Ensuite, les longues lignes doivent être évitées. La proposition suggère des lignes de maximum 80 caractères. Pour cela, il existe l'opérateur \ connu sous le nom de saut de ligne explicite : il permet de diviser une seule longue ligne continue en plusieurs lignes de code plus petites et plus faciles à lire.

```
1 string = "Cette" + "chaîne" + "est" + "bien" + "trop" + "longue" \
2  + "pour" + "tenir" + "sur" + "une" + "ligne."
```

Concernant le nommage des variables, des fonctions ou des constantes, là encore la proposition PEP8 indique que :

- les noms de variable ou de fonction doivent être en *snake case*, c'est-à-dire en minuscules, avec une séparation des mots par des «_».
  Par exemple : `une_variable`, `une_fonction(param1, param2)`.
- les constantes sont aussi en *snake case* mais en majuscules.
  Par exemple : `UNE_CONSTANTE`.

Par ailleurs, lorsqu'un programme grossit, il est habituel de le découper en plusieurs fonctions. Pour faciliter la lecture, il est recommandé (bien que non explicité dans PEP) d'attribuer des noms explicites aux fonctions avec, de préférence, un verbe d'action qui la décrit. Par exemple, une fonction qui lit une

[†] `https://peps.python.org/pep-0000/`

image peut s'appeler `lire_image()`, ou encore une fonction qui compte les voisins d'un nœud dans un graphe peut s'appeler `compter_voisins()`.

Enfin, pour une question de lisibilité, la proposition recommande de mettre une espace avant et après les opérateurs :

```
# incorrect
x=1
x = x+1
x = (a+b) * (a-b)
une_liste = [1,2,3]

# correct
x = 1
x = x + 1
x = (a + b) * (a - b)
une_liste = [1, 2, 3]
```

Affectation et opérations mathématiques simples

Le symbole `=` permet d'affecter la valeur d'une expression à une variable. Par exemple, les deux instructions `x=3` puis `x=x+1` affectent successivement la valeur `3` à la variable `x` puis la valeur `3+1`, soit `4`, à cette même variable.

Les opérations numériques (addition, soustraction, multiplication, division, puissance), ainsi que les opérations booléennes (négation, et, ou), se notent respectivement `a+b`, `a-b`, `a*b`, `a/b`, `a**b`, `not a`, `a and b`, `a or b`. Concernant la division euclidienne, si $a = b.q + r$ avec $0 \leq r < b$, alors q = `a//b`, et r = `a%b` ou q, r =`divmod(a, b)`.

Lorsque ces opérations sont possibles, pour convertir la variable `y` en entier, il faut écrire `x=int(y)`, et pour convertir la variable `x` en chaîne de caractères, il faut écrire `y=str(x)`.

Instruction conditionnelle : `if`

Pour exprimer une condition, les mots-clés `if`, `else` et `elif` sont utilisés. Le code suivant affecte la valeur `3` à la variable `res` si la valeur de la variable `x` est strictement plus grande que `3`. Sinon, si `x` vaut zéro, alors `res` reçoit la valeur `0`, et sinon `res` prend la valeur `1`.

```
if x > 3:
    res = 3
elif x == 0:
    res = 0
else:
    res = 1
```