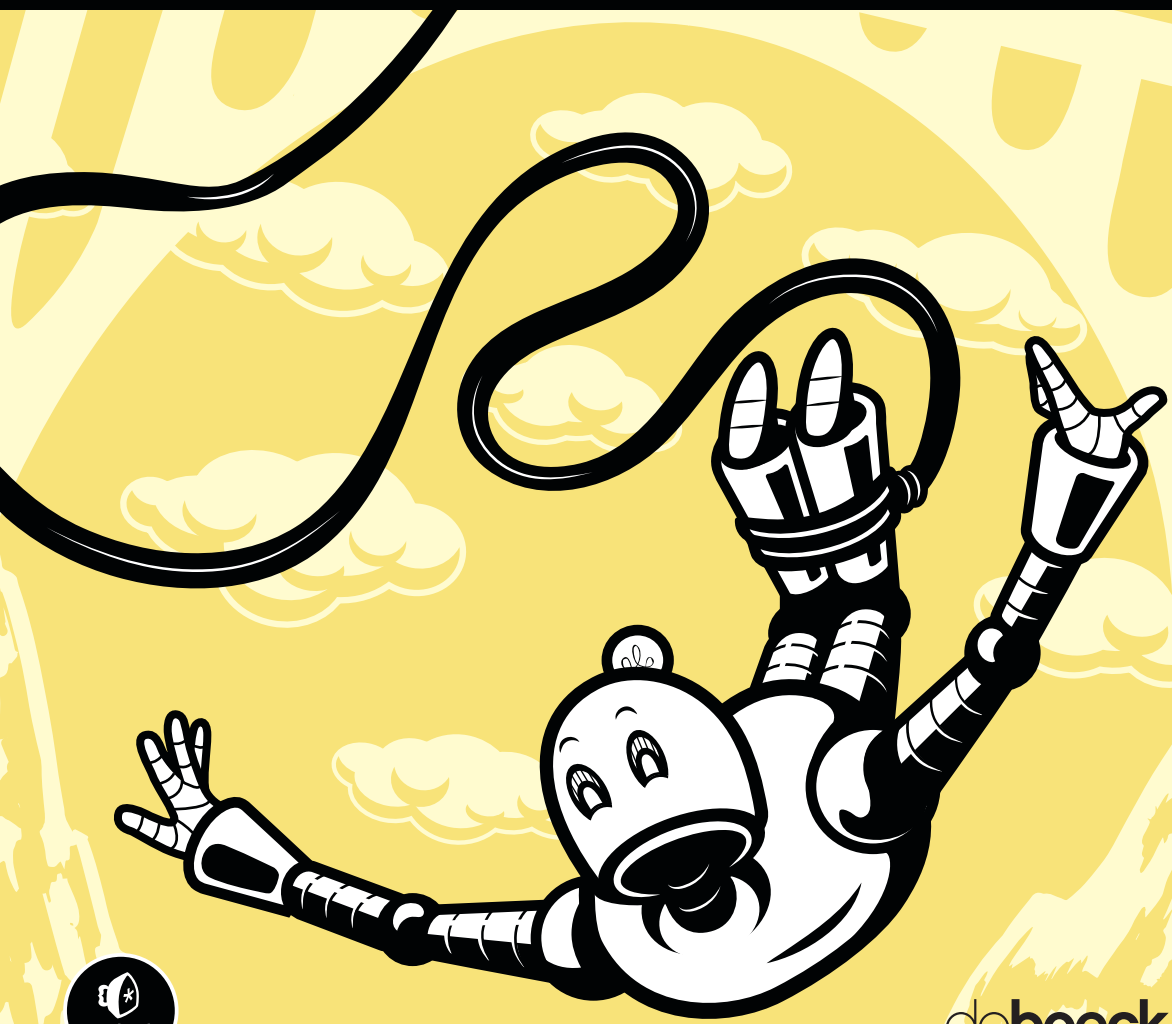
AU CŒUR DES ALGORITHMES

LES BASES DE LA PROGRAMMATION AVEC PYTHON

BRADFORD TUCKFIELD



no starch press

deboeck SUPÉRIEUR

AU CŒUR DES ALGORITHMES

Collection informatique

Brooks J.C., Grow C., Craig P., Short D., *Cybersécurité. Sécurisation des systèmes informatiques*

Desgraupes B., *LaTeX. Apprentissage, guide et référence*

Grimes R.A., *Hacking et contre-hacking. La sécurité informatique*

Lubanovic B., *Python. Comprendre les bases et maîtriser la programmation*

Monk S., *Programmation Arduino. Développez rapidement vos premiers programmes*

Taillet R., *Bien débuter en LaTeX*

Taillet R., *Python pour la physique. Calcul, graphisme, simulation*

Thiry T., *Les pratiques de l'équipe agile. Définissez votre propre méthode*, 2e éd.

Chez le même éditeur

Agashe A., Detroja P., Mehta N., *Blockchain : bulle ou révolution ?*

Bhargava A., *Les algorithmes, c'est plus simple avec un dessin !*

Cypel A., *Au cœur de l'intelligence artificielle. Des algorithmes à l'IA forte*

AU CŒUR DES ALGORITHMES

Les bases de la programmation avec Python

Bradford Tuckfield
Traduit de l'anglais par Benoît Clenet

no starch press

deboeck SUPÉRIEUR

Ouvrage original :

Pour la traduction française
© De Boeck Supérieur s.a., 2023
Rue du Bosquet, 7 B-1348 Louvain-la-Neuve

Dépôt légal :
Bibliothèque nationale, Paris : septembre 2023 ISBN : 978-2-8073-4520-1
Bibliothèque royale de Belgique, Bruxelles : 2023/13647/081

Ce livre est dédié à mes parents, David et Becky Tuckfield,
pour la confiance qu'ils m'accordent et pour m'avoir appris
à jouer à *la pipopipette*.

À propos de l'auteur

Bradford Tuckfield est data scientist et écrivain. Il dirige une société de conseil en science des données baptisée Kmbara (*https://kmbara.com/*) et administre un site web d'œuvres de fiction nommé Dreamtigers (*http://thedreamtigers.com/*).

À propos du vérificateur technique

Alok Malik est data scientist à New Delhi en Inde. Il travaille sur le développement de modèles d'apprentissage profond à la fois en traitement du langage naturel et en vision par ordinateur, à l'aide de Python. Il a conçu et déployé des solutions telles que des modèles du langage, des outils de classification d'images et de textes, des traducteurs, des convertisseurs de voix en texte, des modèles de reconnaissance d'entités nommées et des détecteurs d'objets. Il a également co-écrit un livre sur le machine learning. À ses heures perdues, il lit des pages sur la finance, fait des MOOCs et joue aux jeux vidéo sur sa console.

SOMMAIRE

TABLE DES MATIÈRES

REMERCIEMENTS

« Un mot n'est pas le même dans un écrivain et dans un autre. L'un se l'arrache du ventre. L'autre le tire de la poche de son pardessus. » C'est ainsi que Charles Peguy décrivait l'écriture de chaque mot. La même chose est vraie pour les chapitres et les livres entiers. Tantôt, j'ai eu l'impression de tirer ce livre de la poche de mon manteau, et tantôt, j'ai eu le sentiment de me l'arracher du ventre. Il convient de remercier toutes les personnes ayant contribué à cette longue gestation, soit pour m'avoir prêté un manteau, soit pour m'avoir aidé à nettoyer mes viscères débordants.

Plusieurs personnes m'ont apporté leur aide sur le long chemin d'expériences et d'aptitudes nécessaires pour écrire ce livre. Mes parents, David et Becky Tuckfield, m'ont offert tant de cadeaux, à commencer par la vie et l'éducation que j'ai eues, et continuent de croire en moi, de m'encourager et de m'aider de tant de manières qu'il serait trop long de les lister ici. Scott Robertson m'a donné l'opportunité d'avoir ma première expérience en écriture de code, même si je n'étais pas qualifié ni très bon. Randy Jenson m'a accordé mon premier poste en sciences des données, en dépit encore une fois de mon manque d'expérience. Kumar Kashyap m'a donné la chance, pour la première fois, de diriger une équipe de développeurs en charge d'implémenter des algorithmes. David Zou fut le premier à me rémunérer pour un article (10 $ moins les frais PayPal pour 10 critiques de courts-métrages),

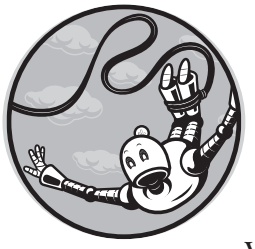
et c'était si bien qu'il m'encouragea à écrire davantage. Aditya Date fut la première personne à me suggérer d'écrire un livre et m'offrit la chance de pouvoir le faire.

J'ai également reçu des encouragements de plusieurs de mes professeurs et mentors. David Cardon m'a confié pour la première fois la chance de collaborer à la recherche académique, et m'a enseigné tant de choses durant cette période. Bryan Skelton et Leonard Woo m'ont montré des exemples de ce que je voulais devenir. Wes Hutchinson m'a initié à certains algorithmes fondamentaux, dont le partitionnement en *k*-moyennes, et m'a permis de mieux comprendre le fonctionnement des algorithmes. Chad Emmett m'a appris comment prendre en compte l'histoire et la culture, et le chapitre 2 lui est dédié. Uri Simonsohn m'a enseigné la manière d'appréhender les données.

Certaines personnes m'ont aidé de sorte que l'écriture de ce livre fut un vrai plaisir pour moi. Seshu Edala a ajusté mon agenda de façon à trouver le temps d'écrire, et m'a constamment encouragé. Ce fut un plaisir de collaborer avec Alex Freed durant le processus d'édition. Jennifer Eagar, par un transfert du service de paiement mobile Venmo réalisé des mois avant la date de publication, devint officieusement la première personne à acheter un exemplaire du livre, ce qui fut apprécié durant cette période difficile. Hlaing Hlaing Tun a été réconfortante, attentionnée, douce et stimulante durant tout ce temps.

Je ne pourrai jamais m'acquitter en retour de toutes ces marques de reconnaissance, mais je peux au moins remercier. Merci !

INTRODUCTION



Les algorithmes sont omniprésents, et vous en avez certainement fait tourner plusieurs aujourd'hui. Dans ce livre, vous décoderez quelques dizaines d'algorithmes : certains simples, d'autres complexes, tantôt célèbres, tantôt inconnus, mais tous intéressants et méritant d'être apprivoisés. Le premier algorithme que vous découvrirez est également le plus exquis : il permet de concocter un parfait aux fruits rouges, et il est intégralement reproduit à la Figure 1.

Certes, vous avez pour habitude d'appeler « recette » ce type d'algorithme, mais il s'accorde avec la définition qu'en donne Donald Knuth : un *algorithme* est un ensemble fini de règles qui engendre une succession d'opérations permettant de résoudre un type particulier de problème.

Parfait aux fruits rouges

Instructions :

1. Placez le sixième d'une tasse de myrtilles au fond d'une grande verrine.
2. Recouvrez les myrtilles d'une demi-tasse de véritable yaourt turc.
3. Superposez le tiers d'une tasse de céréales sur le yaourt.
4. Recouvrez les céréales d'une demi-tasse de véritable yaourt turc.
5. Placez quelques fraises au-dessus.
6. Surmontez le tout de votre crème fouettée préférée.

Figure 1 : Un algorithme est un ensemble fini de règles qui engendre une succession d'opérations permettant de résoudre un type particulier de problème.

La préparation de desserts n'est pas la seule activité régie par les algorithmes. Chaque année, le gouvernement américain impose à ses citoyens adultes d'exécuter un algorithme et menace d'emprisonner ceux qui ne le réaliseraient pas correctement. En 2017, des millions d'Américains ont rempli leur devoir en complétant l'algorithme indiqué à la Figure 2, qui est un extrait du formulaire 1040-EZ.

1	Wages, salaries, and tips. This should be shown in box 1 of your Form(s) W-2. Attach your Form(s) W-2.	1
2	Taxable interest. If the total is over $1,500, you cannot use Form 1040EZ.	2
3	Unemployment compensation and Alaska Permanent Fund dividends (see instructions).	3
4	Add lines 1, 2, and 3. This is your **adjusted gross income.**	4
5	If someone can claim you (or your spouse if a joint return) as a dependent, check the applicable box(es) below and enter the amount from the worksheet on back. ☐ **You** ☐ **Spouse** If no one can claim you (or your spouse if a joint return), enter $10,400 if **single;** $20,800 if **married filing jointly.** See back for explanation.	5
6	Subtract line 5 from line 4. If line 5 is larger than line 4, enter -0-. This is your **taxable income.** ▶	6
7	Federal income tax withheld from Form(s) W-2 and 1099.	7
8a	**Earned income credit (EIC)** (see instructions)	8a
b	Nontaxable combat pay election. 8b	
9	Add lines 7 and 8a. These are your **total payments and credits.** ▶	9
10	**Tax.** Use the amount on **line 6 above** to find your tax in the tax table in the instructions. Then, enter the tax from the table on this line.	10
11	Health care: individual responsibility (see instructions) Full-year coverage ☐	11
12	Add lines 10 and 11. This is your **total tax.**	12

Figure 2 : Les instructions pour la déclaration des impôts correspondent à la définition d'un algorithme.

Comment se fait-il que les impôts et les parfaits ont un point commun ? Les impôts sont inévitables, bourrés de chiffres, difficiles et universellement détestés. Les parfaits sont occasionnels, raffinés, faciles à préparer et appréciés sans exception. La seule caractéristique qu'ils partagent est que les gens les réalisent en suivant des algorithmes.

En plus de le définir, le célèbre informaticien Donald Knuth nota qu'un *algorithme* est presque synonyme de *recette*, *procédure* et *charabia*. Dans le cas de la déclaration d'impôts représentée par l'extrait du formulaire 1040-EZ, nous avons 12 étapes (une liste finie) qui précisent les

opérations (telles que l'addition à l'étape 4 et la soustraction à l'étape 6) permettant de résoudre un type particulier de problème : chercher à éviter la prison pour fraude fiscale. Dans le cas du parfait, nous avons six étapes qui précisent les opérations (telles que « placer » à l'étape 1 et « recouvrir » à l'étape 2) pour résoudre une catégorie spécifique de problème : chercher à déguster un parfait.

À mesure que vous approfondirez les algorithmes, vous commencerez à en voir partout et vous en viendrez à apprécier toute la puissance qu'ils renferment. Au premier chapitre, nous discuterons de la remarquable aptitude humaine à saisir une balle au vol, et découvrirons les détails de l'algorithme dissimulé dans le subconscient qui nous permet de le faire. Ensuite, nous aborderons les algorithmes permettant de déboguer du code, de décider quelle quantité de nourriture manger lors d'un buffet, d'optimiser ses revenus, de trier des listes, de planifier des tâches, de corriger du texte, de distribuer des messages et de gagner à des jeux tels que les échecs et le sudoku. Ce faisant, nous apprendrons à classer les algorithmes selon plusieurs critères jugés importants par les informaticiens professionnels. Nous parviendrons alors à un certain niveau de dextérité voire même, osons-le dire, à l'*art* des algorithmes, laissant le champ libre à la créativité et à la personnalité dans cette activité par ailleurs rigoureuse et quantitative.

À qui s'adresse ce livre ?

Ce livre donne une introduction ludique aux algorithmes, agrémentée de code Python. Pour en tirer tout le bénéfice, il est néanmoins conseillé d'avoir pratiqué, au moins de façon rudimentaire, ce qui suit :

Programmation ou codage. Chaque exemple important est illustré par du code Python. Ce livre s'efforce d'apporter toutes les explications détaillées des extraits de code afin qu'ils soient compréhensibles par ceux n'ayant aucune expérience en programmation et avec Python. Cependant, celui qui aurait une compréhension élémentaire des fondamentaux de la programmation – tels que la déclaration de variables, les boucles `for`, les instructions `if/then` et les appels de fonction – sera mieux préparé pour en tirer parti.

Mathématiques de l'enseignement secondaire. Les algorithmes sont souvent utilisés pour traiter des problèmes mathématiques, comme la résolution d'équations, l'optimisation et le calcul numérique. Ils appliquent également de nombreux principes qui sont associés au raisonnement mathématique, tels que la logique et la nécessité d'avoir des définitions précises. Certaines de nos discussions s'aventureront sur le territoire mathématique, à savoir l'algèbre, le théorème de Pythagore, la constante π et un tout petit peu de calcul intégral très élémentaire. Nous nous évertuerons à ne rien laisser d'abscons et nous ne nous hasarderons pas au-delà des mathématiques enseignées au lycée.

Celui qui se sent à l'aise avec ces prérequis pourra maîtriser tout le contenu du livre. Il a été rédigé à destination des lecteurs suivants :

Les étudiants. Ce livre convient parfaitement comme cours d'introduction aux algorithmes, à l'informatique et à la programmation pour le lycée ou le premier cycle universitaire.

Les informaticiens. Certains informaticiens professionnels pourraient acquérir des connaissances utiles de ce livre, notamment les développeurs ou les ingénieurs cherchant à se familiariser avec Python, ou ceux qui souhaitent approfondir les fondements de l'informatique et la manière d'améliorer le code par le raisonnement algorithmique.

Les amateurs curieux. Le véritable public de ce livre sont les amateurs curieux. Les algorithmes sont présents quasiment à chaque instant de notre quotidien, de sorte que chacun devrait trouver dans ce livre au moins une bonne raison d'enrichir sa perception du monde qui l'entoure.

À propos de ce livre

Cet ouvrage ne couvre pas tous les aspects de chaque variété d'algorithme, il s'agit uniquement d'une introduction. Après l'avoir lu, vous aurez une compréhension solide de ce qu'est un algorithme, vous saurez comment écrire du code pour implémenter les principaux d'entre eux et comment estimer et optimiser leur performance. Vous vous familiariserez aussi avec les algorithmes les plus répandus chez les professionnels. Les chapitres sont organisés de la manière suivante :

Chapitre 1 : Résolution de problèmes, dans lequel nous analyserons la manière d'attraper une balle, découvrirons des preuves de l'existence d'un algorithme subconscient guidant le comportement humain et discuterons des enseignements à en tirer concernant l'utilité des algorithmes et la manière de les concevoir.

Chapitre 2 : Histoires d'algorithmes, où nous voyagerons dans l'espace et le temps pour découvrir comment les égyptiens de l'Antiquité et les paysans russes multipliaient les nombres, comment les Grecs de l'Antiquité trouvèrent les plus grands communs diviseurs et comment les savants japonais de l'époque médiévale dévoilèrent les carrés magiques.

Chapitre 3 : Minimisation et maximisation, dans lequel nous introduirons la montée et la descente de gradient. Ces méthodes élémentaires pour trouver le maximum et le minimum d'une fonction sont utilisées pour l'optimisation, un objectif important de nombreux algorithmes.

Chapitre 4 : Tri et recherche, où nous présenterons les algorithmes fondamentaux pour trier des listes et y rechercher des éléments. Nous introduirons également la mesure de leur efficacité et de leur rapidité.

Chapitre 5 : Maths pures, où nous nous focaliserons sur des algorithmes exclusivement mathématiques, dont ceux engendrant des fractions continues, calculant les racines carrées et produisant des nombres pseudo-aléatoires.

Chapitre 6 : Optimisation avancée, dans lequel nous lèverons le voile sur une méthode puissante permettant de trouver des solutions optimales : le recuit simulé. Nous évoquerons également le problème du voyageur de commerce, un grand classique en informatique théorique.

Chapitre 7 : Géométrie, où nous examinerons comment produire des diagrammes de Voronoï, qui peuvent se révéler utiles dans une large palette d'applications géométriques.

Chapitre 8 : Langage, dans lequel nous développerons la manière d'ajouter intelligemment des espaces à un texte qui en manque et comment suggérer efficacement les prochains mots d'une phrase.

Chapitre 9 : *Machine learning*, où nous discuterons des arbres de décision, une méthode fondamentale d'apprentissage automatique.

Chapitre 10 : Intelligence artificielle, dans lequel nous nous tournerons vers un projet ambitieux : implémenter un algorithme capable de jouer contre nous – et peut-être même de gagner. Nous débuterons avec un jeu simple, la Pipopipette, et examinerons comment rendre l'algorithme plus performant.

Chapitre 11 : Aller encore plus loin, où nous évoquerons les pistes pour progresser vers les travaux algorithmiques les plus avancés. Nous expliquerons comment concevoir un *chatbot* et comment gagner un million de dollars en créant un algorithme de sudoku.

Préparer l'environnement

Nous implémenterons les algorithmes décrits dans ce livre grâce au langage Python. C'est un logiciel libre et gratuit, pouvant fonctionner sur les principales plateformes. Voici les instructions permettant d'installer Python sur Windows, macOS ou Linux.

Installer Python sur Windows

Pour installer Python sur Windows, procédez comme suit :

1. Ouvrez la page relative à la dernière version de Python pour Windows (prenez garde à bien inclure le slash final) : *https://www.python.org/downloads/windows/*.
2. Cliquez sur le lien concernant la version de Python que vous souhaitez télécharger. Pour obtenir la version la plus récente, cliquez sur le lien **Latest Python 3 Release – Python 3.X.Y**, où *3.X.Y* représente le numéro de la dernière version, comme 3.11.0. Le code de ce livre a été testé sur Python 3.6 et 3.8[1]. Si vous préférez télécharger une version plus ancienne, déroulez la section « Stable Releases » jusqu'à la version désirée.

1. [NdT] Le code de la version française du livre a été testé sur Python 3.11.2.

3. Le lien sur lequel vous avez cliqué à l'étape 2 vous envoie vers une page dédiée à la version de Python choisie. Dans la section « Files », cliquez sur le lien **Windows installer (64-bit)**.
4. Le lien de l'étape 3 lance le téléchargement d'un fichier *.exe* sur votre ordinateur. C'est un fichier exécutable : double-cliquez dessus pour le lancer. Il réalisera automatiquement le processus d'installation. Cochez la case **Add Python 3.X to PATH** où *X* représente le numéro de version de l'exécutable que vous avez téléchargé, par exemple 11. Après cela, cliquez sur **Install Now** et choisissez les options par défaut.
5. Lorsque vous voyez le message « Setup was successful », cliquez sur **Close** pour achever le processus d'installation.

Une nouvelle application apparaît alors sur votre ordinateur. Elle s'appelle Python 3.*X*, où *X* est la version de Python 3 que vous avez installée. Dans la barre de recherche Windows, tapez `Python`. Lorsque l'application apparaît, cliquez dessus pour ouvrir la console Python. Vous pouvez saisir des commandes Python dans celle-ci, et elles s'exécuteront ici.

Installer Python sur macOS

Pour installer Python sur macOS, procédez comme suit :

1. Ouvrez la page relative à la dernière version de Python pour macOS (prenez garde à bien inclure le slash final) : *https://www.python.org/downloads/mac-osx/*.
2. Cliquez sur le lien concernant la version de Python que vous souhaitez télécharger. Pour obtenir la version la plus récente, cliquez sur le lien **Latest Python 3 Release – Python 3.X.Y**, où *3.X.Y* représente le numéro de la dernière version, comme 3.11.0. Le code de ce livre a été testé sur Python 3.6 et 3.8. Si vous préférez télécharger une version plus ancienne, déroulez la section « Stable Releases » jusqu'à la version désirée.
3. Le lien sur lequel vous avez cliqué à l'étape 2 vous envoie vers une page dédiée à la version de Python choisie. Dans la section « Files », cliquez sur le lien **macOS 64-bit universal2 installer**.
4. Le lien de l'étape 3 lance le téléchargement d'un fichier *.pkg* sur votre ordinateur. C'est un fichier exécutable : double-cliquez dessus pour le lancer. Il réalisera automatiquement le processus d'installation. Choisissez les options par défaut.
5. L'exécutable créera un dossier sur votre ordinateur nommé *Python 3.*X, où *X* représente le numéro de la version de Python que vous avez installée. Dans ce dossier, double-cliquez sur l'icône intitulée IDLE. Ceci ouvrira le Shell Python 3.*X.Y*, où 3.*X.Y* est la version choisie. C'est une console vous permettant d'exécuter toutes les commandes Python.

Installer Python sur Linux

Pour installer Python sur Linux, procédez comme suit :

1. Déterminez quel gestionnaire de paquets est utilisé par votre distribution Linux. Les plus connus sont `yum` et `apt-get`.
2. Ouvrez la console Linux (également appelée terminal) et exécutez les deux commandes suivantes :

```
> sudo apt-get update
> sudo apt-get install python3.11
```

   Si vous utilisez `yum` ou un autre gestionnaire de paquets, remplacez chaque instance d'`apt-get` apparaissant sur les deux lignes par `yum` ou le nom de votre gestionnaire de paquets. De même, si vous voulez installer une version plus ancienne de Python, remplacez **`3.11`** (la version la plus récente au moment où ce livre est écrit) par tout autre numéro de version, comme **`3.8`**, l'une des versions utilisées pour tester le code dans ce livre. Pour connaître la dernière version de Python, allez sur *https://www.python.org/downloads/source/*. Vous y verrez alors un lien **Latest Python 3 Release – Python 3.*X.Y***, où 3.*X.Y* représente un numéro de version. Utilisez seulement les deux premiers chiffres dans la commande d'installation indiquée ci-dessus.
3. Lancez Python en exécutant la commande suivante dans la console Linux :

```
python3
```

La console Python s'ouvre alors dans la fenêtre du terminal. Vous pourrez y taper les commandes Python.

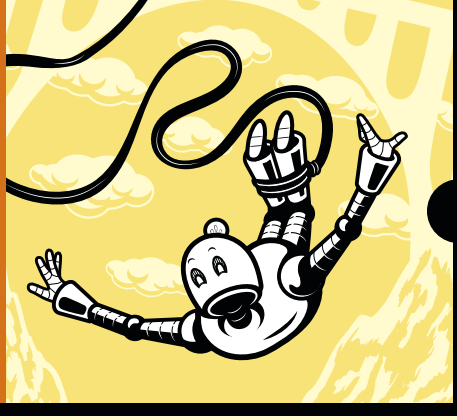
Installation de modules tiers

Certaines parties de code introduites dans ce livre reposent sur des modules Python ne faisant pas partie du logiciel standard téléchargé à partir du site web officiel. Pour installer des modules tiers sur votre ordinateur, suivez les instructions données sur la page *http://automatetheboringstuff.com/2e/appendixa/*.

Résumé

Notre étude des algorithmes nous fera voyager tout autour du monde et remonter l'histoire de plusieurs siècles. Nous explorerons les innovations provenant de l'Égypte et de Babylone dans l'Antiquité, d'Athènes au règne péricléen, de Bagdad, de l'Europe médiévale, du Japon à l'époque d'Edo, du Raj britannique, jusqu'à notre époque moderne et sa technologie époustouflante. Nous serons amenés à découvrir de nouvelles voies pour aborder les problèmes, avec des contraintes qui semblaient au départ impossibles à surmonter. Ce faisant, nous nous

LES BASES DE LA PROGRAMMATION AVEC PYTHON



Une introduction aux algorithmes, appliquée au langage Python. Parcourez les algorithmes les plus intéressants pour rechercher, trier et optimiser, et d'autres plus avancés, utilisés en machine learning et en intelligence artificielle. Vous y découvrirez aussi les algorithmes qu'utilisaient nos ancêtres de l'Antiquité pour effectuer une multiplication, trouver le plus grand commun diviseur et concocter des carrés magiques.

Vous apprendrez à :

- générer des diagrammes de Voronoï ;
- utiliser des algorithmes pour construire un chatbot rudimentaire, gagner à un jeu de stratégie et solutionner une grille de sudoku ;
- coder les algorithmes de montée et de descente de gradient afin d'identifier le minimum et le maximum d'une fonction ;
- exploiter le recuit simulé pour réaliser une optimisation globale ;
- construire un arbre de décision pour prédire le niveau de bonheur d'une personne ;
- corriger du code, maximiser ses recettes et produire des nombres aléatoires ;
- mesurer l'efficacité et la rapidité des algorithmes.

Vous découvrirez également certains algorithmes utiles en maths pures et comment des idées mathématiques peuvent les améliorer.

Retrouvez en détails les algorithmes très puissants utilisés aujourd'hui, et comment les implémenter et les coder en Python 3, tout en mesurant et en optimisant leur performance.

Bradford Tuckfield, titulaire d'un doctorat, est data scientist, consultant et co-auteur de *Applied Unsupervised Learning* with R. Ses travaux de recherche ont été publiés dans de prestigieuses revues de maths, d'économie et de médecine. Il a également rédigé des articles culturels pour des magazines et des journaux politiques.

Traduction de l'anglais de Benoît Clenet, ingénieur informaticien. Passionné de physique, d'astronomie et de mathématiques. Il a traduit plusieurs ouvrages dans ces domaines.

no starch press

deboeck SUPÉRIEUR

www.deboecksuperieur.com

24,90 €

ISBN : 978-2-8073-4520-1

9 782807 345201