

2^E ÉDITION

PYTHON

POUR LES KIDS

La programmation accessible aux enfants

JASON R. BRIGGS



250 000 EX.
VENDUS
EN ANGLAIS

● Éditions
EYROLLES



PYTHON POUR LES KIDS

2^E ÉDITION

La programmation accessible aux enfants

Jason R. Briggs

● Éditions
EYROLLES


**no starch
press**

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

Copyright © 2023 year by Jason R. Briggs.

Title of English-language original: *Python for Kids: A Playful Introduction to Programming*, 2nd edition, ISBN 9781718503021, published by No Starch Press Inc. 245 8th Street, San Francisco, California United States 94103.

The French language 2nd edition Copyright © 2023 by SAS Editions Eyrolles under license by No Starch Press Inc. All rights reserved.

Depuis 1925, les Éditions Eyrolles s'engagent en proposant des livres pour comprendre le monde, transmettre les savoirs et cultiver ses passions ! Pour continuer à accompagner toutes les générations à venir, nous travaillons de manière responsable, dans le respect de l'environnement. Nos imprimeurs sont ainsi choisis avec la plus grande attention, afin que nos ouvrages soient imprimés sur du papier issu de forêts gérées durablement. Nous veillons également à limiter le transport en privilégiant des imprimeurs locaux. Ainsi, 89 % de nos impressions se font en Europe, dont plus de la moitié en France.

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.

© 2013 by Jason Briggs / No Starch Press pour l'édition en langue anglaise

© Groupe Eyrolles, 2015

© Éditions Eyrolles 2023 pour la présente édition, ISBN : 978-2-416-01347-8

© Traduction française : Hervé Soulard

Le programmeur informatique est un créateur d'univers dont elle ou il est le seul législateur.

Aucun dramaturge, aucun metteur en scène, aucun empereur, aussi puissant soit-il, n'a jamais exercé un pouvoir aussi absolu pour organiser une scène ou un champ de bataille, et n'a jamais dirigé des acteurs ou des troupes aussi indéfectiblement dévoués.

Joseph Weizenbaum (légèrement modifiée)

À propos de l'auteur

Jason R. Briggs programme depuis l'âge de huit ans, lorsqu'il a appris le BASIC sur un TRS-80 de Radio Shack. Sur le plan professionnel, il a ensuite créé des programmes en tant que développeur et architecte système. Il a également collaboré au magazine *Java developer's journal*. Ses articles ont été publiés dans *JavaWorld*, *ONJava* et *ONLamp*. *Python pour les kids* (Groupe Eyrolles, 2015) est son premier livre.

Pour contacter Jason, adressez-vous à l'éditeur par courriel à ahabian@eyrolles.com, qui transmettra.

À propos de l'illustrateur

Miran Lipovača est l'auteur de *Apprendre Haskell vous fera le plus grand bien !*. Il aime la boxe, jouer de la guitare basse et, bien entendu, dessiner. Il éprouve une fascination pour les Danses macabres et le nombre 71. Quand il passe des portes à ouverture automatique, il aime prétendre qu'il les ouvre en réalité à la force de son esprit.

À propos du relecteur technique

Daniel Zingaro, maître de conférence en informatique, est un enseignant reconnu de l'université de Toronto. Ses recherches se concentrent sur la compréhension et l'amélioration de l'apprentissage de l'informatique chez les étudiants. Il est l'auteur de deux ouvrages récents publiés par No Starch : *Algorithmic Thinking* (un guide pragmatique, non mathématique, sur les algorithmes et les structures de données) et *Learn to Code by Solving Problems* (un livre d'introduction à Python et à la pensée computationnelle).

À propos des relecteurs techniques de la première édition

Récemment diplômé de l'école Nueva à l'âge de 15 ans, Josh Pollock est étudiant à l'école secondaire supérieure de Lick-Wilmerding de San Francisco. Il a commencé à programmer avec Scratch à l'âge de 9 ans et en TI-BASIC à 11 ans, puis a débuté en Python et Java à 12 ans, et en UnityScript à 13 ans. Outre la programmation, Josh aime jouer de la trompette, développer des jeux informatiques et discuter avec les gens de sujets intéressants comme les disciplines académiques scientifiques, technologiques, mathématiques et d'ingénierie.

Maria Fernandez possède une maîtrise en linguistique appliquée, s'intéresse à l'informatique et aux technologies depuis plus de vingt ans. Elle a enseigné l'anglais à de jeunes femmes réfugiées dans le cadre du *Global Village Project* de Géorgie. Elle réside actuellement dans le nord de la Californie et travaille dans un service d'expérimentation pédagogique.

Remerciements

Je remercie l'équipe de No Starch pour tout le travail qu'elle a fourni et Miran Lipovača pour ses illustrations particulièrement brillantes.

Merci à ma femme, à ma fille et à mon fils, qui ont dû vivre avec un époux et un père vivant le nez sur l'écran encore plus souvent que d'habitude.

À ma maman, pour les encouragements sans fin qu'elle m'a adressés au cours de toutes ces années.

Enfin, merci à mon papa d'avoir acheté un ordinateur dès les années 1970 et d'avoir su composer d'emblée avec moi, qui voulais l'utiliser autant que lui. Rien de tout cela n'eût été possible sans lui.

TABLES DES MATIÈRES

Avant-propos	1
Pourquoi Python ?	2
Comment apprendre à programmer ?	2
À qui est destiné ce livre ?	3
Que contient ce livre ?	4
Site d'accompagnement	5
Amuse-toi bien !	5

PARTIE 1

APPRENDRE À PROGRAMMER

1	
Les serpents rampent, mais pas tous	9
Quelques mots à propos du langage	10
Installer Python	11
Installer Python sous Windows	11
Installer Python sous macOS	12
Installer Python sous Ubuntu	13
Installer Python sous Raspberry Pi (Raspberry Pi OS ou Raspbian)	14
Lorsque Python est installé	16
Enregistrer des programmes Python	17
Ce que tu as appris	19

2	
Calculs et variables	21
Calculer avec Python	22
Les opérateurs de Python	23
L'ordre des opérateurs	23
Les variables sont comme des étiquettes	25
Utiliser les variables	27
Ce que tu as appris	29

3

Chaînes, listes, tuples et dictionnaires 31

Les chaînes de caractères.....	32
Créer des chaînes.....	32
Gérer les problèmes liés aux chaînes.....	33
Insérer des valeurs dans des chaînes.....	36
Multiplier des chaînes.....	37
Plus puissantes que les chaînes : les listes.....	38
Ajouter des éléments à une liste.....	41
Supprimer des éléments d'une liste.....	42
Arithmétique de liste.....	42
Tuples.....	44
Dictionnaires.....	45
Ce que tu as appris.....	47
Puzzles de programmation.....	48
1. Favoris.....	48
2. Compter les combattants.....	48
3. Salutations.....	48
4. Lettre multiligne.....	48

4

Dessiner avec une tortue 49

Utiliser le module turtle de Python.....	50
Créer un canevas.....	50
Déplacer la tortue.....	51
Ce que tu as appris.....	57
Puzzles de programmation.....	57
1. Un rectangle.....	57
2. Un triangle.....	57
3. Un carré sans coins.....	57
4. Un carré incliné sans coins.....	58

5

Poser des questions avec if et else 59

Instructions if.....	60
Un bloc est un groupe d'instructions.....	60
Des conditions pour comparer des choses.....	62
Instructions si-alors-sinon.....	64
Instructions if et elif.....	65
Combiner des conditions.....	66
Variables sans valeur : None.....	67
Différence entre chaînes et nombres.....	68

Ce que tu as appris	70
Puzzles de programmation.....	70
1. Es-tu riche ?	71
2. Barres chocolatées.....	71
3. Juste le bon nombre.....	71
4. Affronter des ninjas.....	71

6

Tourner en boucle 73

Utiliser les boucles for	74
Tant que nous parlons de boucles : while.....	81
Ce que tu as appris	83
Puzzles de programmation.....	83
1. La boucle Bonjour	83
2. Nombres pairs	84
3. Mes cinq ingrédients préférés.....	84
4. Ton poids sur la Lune.....	84

7

Recycler du code grâce aux fonctions et aux modules 85

Utiliser des fonctions	86
Qu'est-ce qu'une fonction ?.....	87
Variables et portée	88
Utiliser des modules	91
La fonction input	93
Ce que tu as appris	94
Puzzles de programmation.....	95
1. Fonction de base du poids sur la Lune	95
2. Fonction poids sur la Lune avec les années	95
3. Programme de poids sur la Lune	95
4. Programme de poids sur Mars	96

8

Classes et objets 97

Organiser les choses en classes.....	98
Enfants et parents	99
Ajouter des objets aux classes	100
Définir des fonctions de classes.....	101
Ajouter des caractéristiques à une classe.....	101
Pourquoi utiliser des classes et des objets ?.....	103
Objets et classes en images.....	105

Autres fonctionnalités des objets et des classes	107
Fonctions héritées	108
Fonctions appelant d'autres fonctions	109
Initialiser un objet	110
Ce que tu as appris	112
Puzzles de programmation	112
1. Moulinet de girafe	112
2. Fourche de tortues	113
3. Deux petites spirales	113
4. Quatre petites spirales	114

9

Autres graphismes avec la tortue 115

Dessiner un carré, pour commencer	116
Dessiner des étoiles	117
Dessiner une voiture	121
Voir la vie en couleurs	122
Une fonction pour dessiner un cercle plein	123
Dessiner en noir et blanc	125
Fonction de dessin de carré	125
Dessiner des carrés pleins	127
Dessiner des étoiles pleines	128
Ce que tu as appris	130
Puzzles de programmation	130
1. Dessiner un octogone	131
2. Dessiner un octogone plein	131
3. Autre fonction de dessin d'étoile	132
4. Les quatres spirales revisitées	132

10

De meilleurs graphismes avec tkinter 133

Créer un bouton à cliquer	135
Utiliser des paramètres nommés	137
Créer le canevas de dessin	138
Dessiner des lignes	138
Dessiner des rectangles et des carrés	140
Dessiner de nombreux rectangles	142
Définir la couleur	145
Dessiner des arcs	148
Dessiner des polygones	151
Afficher du texte	152
Afficher des images	153
Créer une animation de base	156

Réagir à un événement	158
Autres façons d'utiliser l'identifiant	161
Ce que tu as appris	162
Puzzles de programmation	163
1. Remplir l'écran de triangles	163
2. Le triangle mobile	163
3. La photo mobile	163
4. Remplir l'écran de photos	163

PARTIE 2

REBONDIR !

11

Début de ton premier jeu : Rebondir ! 167

Frapper la balle magique	168
Créer le canevas du jeu	168
Créer la classe de la balle	169
Ajouter de l'action	172
Déplacer la balle	172
Faire rebondir la balle	174
Changer la direction de départ de la balle	176
Ce que tu as appris	178
Puzzles de programmation	178
1. Changer les couleurs	178
2. Couleurs changeantes	178
3. En position !	179
4. Ajouter la raquette... ?	179

12

Fin de ton premier jeu : Rebondir ! 181

Ajouter la raquette	182
Déplacer la raquette	183
Détecter quand la balle touche la raquette	185
Ajouter un facteur chance	188
Ce que tu as appris	192
Puzzles de programmation	192
1. Retarder le début du jeu	192
2. Un véritable « Partie terminée »	193
3. Accélérer la balle	193
4. Enregistrer le score du joueur	193

PARTIE 3

M. FILIFORME COURT VERS LA SORTIE

13

Créer les graphismes du jeu M. Filiforme 197

Plan du jeu de M. Filiforme	198
Obtenir Gimp	198
Créer les éléments de jeu	200
Préparer une image à fond transparent	201
Dessiner M. Filiforme	201
M. Filiforme court vers la droite	202
M. Filiforme court vers la gauche	203
Dessiner les plates-formes	204
Dessiner la porte	204
Dessiner l'arrière-plan	205
Gérer la transparence	206
Ce que tu as appris	207

14

Développer le jeu avec M. Filiforme 209

Créer la classe Jeu	210
Définir le titre de la fenêtre et créer le canevas	210
Terminer la fonction <code>__init__</code>	211
Créer la fonction boucle <code>_principale</code>	212
Créer la classe Coords	214
Vérifier les collisions	215
Les lutins entrent en collision horizontalement	215
Les lutins entrent en collision verticalement	218
Assembler le tout – Code final de détection de collision	218
La fonction <code>collision_gauche</code>	218
La fonction <code>collision_droite</code>	219
La fonction <code>collision_haut</code>	220
La fonction <code>collision_bas</code>	220
Créer la classe Lutin	221
Ajouter les plates-formes	222
Ajouter un objet plate-forme	223
Ajouter d'autres plates-formes	224
Ce que tu as appris	226

Puzzles de programmation	226
1. Damier	226
2. Damier à deux images alternées	227
3. Étagère et lampe.	227
4. Arrière-plan aléatoire.	227

15

Créer M. Filiforme

229

Initialiser le personnage en fil de fer.	230
Charger les images du personnage	230
Définir les variables.	231
Lier les touches aux actions.	232
Faire tourner le personnage vers la gauche ou la droite	233
Faire sauter le personnage	234
Ce que nous avons jusqu'ici	234
Ce que tu as appris	236

16

Achever le jeu avec M. Filiforme

237

Animer le personnage	238
Créer la fonction animer.	238
Vérifier la présence d'un mouvement	238
Changer d'image.	240
Connaître l'emplacement du personnage.	241
Déplacer le personnage	242
Début de la fonction déplacer	242
Le personnage a-t-il touché le bas ou le haut du canevas ?	243
Le personnage a-t-il touché un côté du canevas ?	244
Collision avec d'autres lutins	245
Collision en bas	245
Vérifier la gauche et la droite	247
Tester le lutin du personnage.	249
La sortie !	249
Créer la classe LutinPorte	249
Détecter la porte	250
Ajouter l'objet porte	251
Le jeu final	252
Ce que tu as appris	258
Puzzles de programmation.	259
1. Tu as gagné !	259
2. Animer la porte.	259
3. Plates-formes mobiles	259
4. Une lampe comme lutin	259

Conclusion : et à partir de là ?	261
Installer pip sous Windows	262
Installer pip sous Ubuntu	262
Installer pip sous Raspberry Pi	263
Installer pip sous macOS	264
Essayer PyGame	264
Programmation graphique et de jeux	266
Langages de programmation	266
JavaScript	266
Java	267
C#	267
C/C++	268
Ruby	269
Go	269
Rust	269
Swift	269
Et pour finir	270
 Annexe A	
Mots-clés de Python	271
AND	272
AS	272
ASSERT	272
ASYNC	273
AWAIT	273
BREAK	273
CLASS	274
CONTINUE	274
DEF	275
DEL	275
ELIF	276
ELSE	276
EXCEPT	276
FINALLY	276
FOR	276
FROM	277
GLOBAL	278
IF	279
IMPORT	279
IN	279
IS	280
LAMBDA	280
NONLOCAL	280

NOT	280
OR.	281
PASS.	281
RAISE.	283
RETURN	283
TRY.....	283
WHILE	284
WITH	284
YIELD.	284

Annexe B

Fonctions natives de Python

285

Utiliser des fonctions natives.	286
La fonction abs	286
La fonction all	287
La fonction any	288
La fonction bin	288
La fonction bool	288
La fonction callable	290
La fonction chr	291
La fonction dir	291
La fonction divmod	293
La fonction eval	294
La fonction exec	295
La fonction float	296
La fonction input	297
La fonction int.	297
La fonction len	298
La fonction list	299
Les fonctions max et min	300
La fonction ord	301
La fonction pow	301
La fonction range	302
La fonction sum	303
Ouvrir un fichier en Python	303
Ouvrir un fichier sous Windows.	304
Ouvrir un fichier sous macOS.	304
Ouvrir un fichier sous Linux (Ubuntu)	304
Écrire dans des fichiers	305

Annexe C

Résolution des problèmes **307**

Erreurs « TK » lors de l'importation de turtle sous Ubuntu 308

Erreur d'attribut lors de l'utilisation de turtle 308

Problèmes lors de l'exécution de turtle 308

Une classe ne prend pas d'arguments 309

Index **311**



AVANT-PROPOS

Pourquoi apprendre la programmation informatique ? La programmation encourage la créativité, le raisonnement et la résolution de problèmes. Le programmeur a l'opportunité de créer quelque chose à partir de rien, d'utiliser la logique pour transformer les constructions de programmation en une forme que l'ordinateur peut exécuter et, quand les choses ne se passent pas tout à fait comme prévu, d'exploiter ses capacités de résolution pour trouver ce qui ne va pas. La programmation est une activité amusante, parfois pleine de défis, voire quelquefois frustrante, et les compétences qu'elle permet d'acquérir sont utiles à la fois à l'école et au travail... Même si ta future carrière n'a rien à voir avec l'informatique ! Et, quoi qu'il en soit, la programmation, c'est franchement chouette pour passer un agréable après-midi, alors qu'il ne fait pas beau dehors.

Pourquoi Python ?

Python est un langage de programmation facile à apprendre ; il possède également des caractéristiques réellement utiles pour le programmeur débutant. Le code est assez facile à lire, comparé à d'autres langages, et il dispose d'une console de commandes interactive pour y entrer tes programmes afin de les voir fonctionner.

En plus de sa structure de langage simple et de la console interactive qui facilitent les expérimentations, Python propose certaines fonctionnalités qui améliorent fortement l'apprentissage et permettent de rassembler des animations simples pour créer tes propres jeux. Parmi celles-ci, on trouve le module `turtle`, inspiré des célèbres graphismes avec la tortue – il est utilisé par le langage Logo depuis les années 1960 – et conçu pour une utilisation éducative. Il y a aussi le module `tkinter` qui permet de créer des interfaces graphiques et des animations évoluées grâce à la bibliothèque Tk.

Comment apprendre à programmer ?

Comme pour tout ce que tu essaies pour la première fois, il vaut mieux toujours débuter par les bases. Donc, commence par les premiers chapitres et résiste à l'envie de sauter directement aux derniers. Personne ne peut jouer une symphonie du premier coup, simplement en prenant un instrument. Les élèves pilotes ne commencent pas à piloter un avion tant qu'ils n'ont pas compris et assimilé les commandes essentielles. Les gymnastes ne sont (généralement) pas capables de réaliser des saltos arrière dès leur premier essai. Si tu passes trop vite à la suite, non seulement tu n'auras pas en tête les idées de base mais, en plus, le contenu des chapitres suivants te paraîtra plus compliqué qu'il ne l'est réellement.

À mesure que tu avances dans ta lecture, essaie chacun des exemples proposés pour voir comment ils fonctionnent. À la fin de la plupart des chapitres, tu trouveras des puzzles de programmation. Essaie de les résoudre, car ils t'aideront à améliorer tes connaissances en programmation. Retiens que, plus tu assimiles les bases, plus il te sera facile de comprendre les idées plus complexes qui suivront. Si tu rencontres des choses frustrantes ou trop complexes à résoudre, voici quelques pistes qui te seront très utiles.

1. Découpe un problème en parties plus petites. Essaie de comprendre ce que fait chaque petit extrait de code, ou réfléchis à une seule partie d'une idée difficile (concentre-toi sur une portion de code au lieu d'essayer de comprendre tout en même temps).

2. Si cela ne suffit pas à t'aider, le mieux est souvent de laisser le problème de côté pendant un moment. Endors-toi dessus et reviens-y un autre jour. Souvent, cette méthode aide à résoudre de nombreux problèmes et elle est particulièrement utile pour les programmeurs.

À qui est destiné ce livre ?

Ce livre a été rédigé pour toute personne qui s'intéresse à la programmation, qu'il s'agisse d'un enfant ou d'un adulte approchant la programmation pour la première fois. Si tu veux apprendre à écrire tes propres logiciels, au lieu de simplement utiliser ceux réalisés par d'autres, *Python pour les kids* constitue un excellent point de départ.

Dans les chapitres suivants, tu trouveras des informations pour installer Python, démarrer la console et réaliser des calculs simples, afficher du texte à l'écran, créer des listes, réaliser des opérations élémentaires de contrôle de flux à l'aide des instructions `if` et des boucles `for`. Au passage, tu comprendras ce que signifient ces instructions ! Tu apprendras à réutiliser du code dans des fonctions, les éléments fondamentaux des classes et des objets, ainsi que les descriptions de quelques-uns des nombreux modules et fonctions intégrés à Python.

Des chapitres inspectent les graphismes à l'aide de la tortue, sous forme simple ou plus évoluée, ainsi que l'utilisation du module `tkinter` pour dessiner à l'écran de l'ordinateur. À la fin de nombreux chapitres, les puzzles de programmation de complexité variée sont là pour graver dans le marbre les nouvelles connaissances acquises, en te poussant à rédiger par toi-même de petits programmes. Ensuite, lorsque tu auras construit tes connaissances fondamentales de la programmation, tu apprendras à écrire tes propres jeux. Tu développeras deux jeux graphiques pour maîtriser la détection de collision, les événements et les différentes techniques d'animation.

Les exemples de ce livre utilisent pour la plupart la console d'environnement d'exécution IDLE (qui signifie « environnement de développement intégré »), intégrée à Python. IDLE fournit ce qu'on appelle la « coloration syntaxique », le copier-coller (comme quand tu copies et colles dans les autres applications) et une fenêtre d'édition où tu peux enregistrer du code pour plus tard. C'est donc à la fois un environnement interactif pour l'expérimentation et une sorte d'éditeur de texte. Les exemples proposés fonctionnent aussi parfaitement dans une console standard et un éditeur de texte traditionnel, mais la

coloration syntaxique d’IDLE et son environnement légèrement plus convivial facilitent la compréhension ; c’est pourquoi le tout premier chapitre te montre comment le mettre en place.

Que contient ce livre ?

Voici un bref aperçu de ce que tu trouveras dans cet ouvrage.

Le chapitre 1 forme une introduction à la programmation, avec des instructions pour installer Python.

Le chapitre 2 présente les calculs de base et les variables, tandis que le chapitre 3 décrit certains types fondamentaux de Python, comme les chaînes, les listes et les *tuples*.

Le chapitre 4 propose un avant-goût du module `turtle`. On saute de la programmation élémentaire au déplacement d’une tortue, sous la forme d’une flèche à l’écran.

Le chapitre 5 examine les variantes des conditions et de l’instruction `if`, puis le chapitre 6 décrit les boucles `for` et `while`.

Au chapitre 7, tu commences à utiliser et à créer des fonctions puis, au chapitre 8, tu examines les classes et les objets. Tu abordes suffisamment d’idées de base pour assimiler certaines techniques de programmation dont tu auras besoin dans les chapitres de développement de jeux, qui viennent plus tard dans le livre. À ce stade, le contenu commence à devenir plus complexe.

Le chapitre 9 revient sur le module `turtle` et propose d’essayer quelques formes plus complexes. Le chapitre 10 aborde l’utilisation du module `tkinter` pour créer des graphismes encore plus élaborés.

Aux chapitres 11 et 12, tu réalises ton premier jeu, « Rebondir ! », qui se fonde sur les connaissances acquises dans les chapitres précédents. Aux chapitres 13 à 16, tu crées un autre jeu, « M. Filiforme court vers la sortie ». Ces chapitres devraient te poser quelques petits soucis ; alors, si rien ne va, télécharge le code à partir du site d’accompagnement et compare ton code à ces exemples qui fonctionnent.

Dans la postface, tu découvriras comment employer l’outil d’installation d’un package Python (`pip`) afin d’installer le module `PyGame`, ainsi qu’un petit exemple de son utilisation. Tu feras ensuite connaissance avec d’autres langages de programmation.

Enfin, l’annexe A reprend en détail les mots-clés de Python, tandis que l’annexe B établit la liste des fonctions intégrées utiles (la signification de « mots-clés » et de « fonctions » sera donnée plus loin dans ce livre). L’annexe C propose des éléments de résolution des problèmes les plus fréquents.

Site d'accompagnement

Si tu as besoin d'aide au cours de ta lecture, essaie d'aller voir du côté du site web d'accompagnement, à l'adresse <http://www.editions-eyrolles.com/go/pykids>. Tu y trouveras tous les exemples du livre disponibles en téléchargement, ainsi que des liens vers des informations complémentaires, notamment là où tu pourras récupérer les codes sources employés dans cet ouvrage.

Amuse-toi bien !

À mesure que tu avances dans ce livre, rappelle-toi que la programmation peut être divertissante et qu'il ne faut pas la considérer comme du travail. Pense à la programmation comme à une manière de créer des jeux et applications amusants, que tu pourras partager avec tes amis par exemple.

Apprendre à programmer constitue un formidable exercice mental et les résultats peuvent être très gratifiants. Par-dessus tout, quoi que tu fasses, amuse-toi bien !

