

# ***La Programmation Pour Les Enfants***

***écrit par Peter Armstrong  
traduit par Valérie Taesch***

***Édition  
Mac***

# La Programmation Pour Les Enfants

Peter Armstrong and Valérie Taesch

This book is for sale at

<http://leanpub.com/laprogrammationpourlesenfants>

This version was published on 2014-06-22



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2014 Peter Armstrong and Valérie Taesch

# **Tweet This Book !**

Please help Peter Armstrong and Valérie Taesch by spreading the word about this book on [Twitter](#) !

The suggested hashtag for this book is [#programmingforkids](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter :

<https://twitter.com/search?q=#programmingforkids>

*Pour Evan, littéralement.*  
*Pour Maxime et Élise, plus tard.*

# Table des matières

Introduction . . . . .	i
Chapitre 1 : Bonjour, ligne de commande! . . . . .	1
Exercices . . . . .	12
Chapitre 2 : Bonjour le monde! . . . . .	13
Exercices . . . . .	15
Ceci est un extrait . . . . .	16
Pour les parents . . . . .	17
A propos de l’auteur . . . . .	20
A propos de la traduction . . . . .	21
A propos de Leanpub . . . . .	22

# Introduction

Ce livre va t'apprendre comment écrire un programme informatique !

Tu vas avoir besoin d'un ordinateur Mac (Apple) pour suivre les chapitres. Les programmes sont courts, tu pourras les écrire tous toi-même, même si tu n'es pas à l'aise avec le clavier.

Ce livre est un ensemble de petits chapitres. Chaque chapitre traite un sujet.

A la fin de chaque chapitre, il y aura des exercices à faire. C'est vraiment important que tu les fasses tous ! Faire les exercices permet de s'assurer que tu as bien compris le chapitre. Les réponses de tous les exercices sont à la fin du livre, et il y a des liens vers les réponses dans tous les exercices.

Si tes parents veulent en savoir plus, ils peuvent tout à fait lire le livre avec toi, ou alors juste lire la section [Pour Les Parents](#) qui se trouve à la fin du livre.

Si tu utilises l'ordinateur de tes parents, ils voudront peut-être rester avec toi. Cela ne pose pas de problème, ils vont aussi apprendre des choses dans ce cas !

Pour suivre le livre sur ton Mac, tu vas avoir besoin de lire le fichier PDF que tes parents ont téléchargé. Si tu double-cliques sur le fichier PDF, il va s'ouvrir dans un programme appelé Aperçu. Tu vas devoir passer du programme Aperçu, pour lire le livre, au Terminal, pour écrire les exercices. Si tu cliques dans Aperçu pour changer de page, n'oublies pas de re-cliquer dans la fenêtre Terminal **avant** de taper les commandes au clavier.

Si le fait de passer du programme Aperçu au Terminal te gêne, tu peux procéder autrement : si tes parents ont une imprimante, tu

peux leur demander d'imprimer le livre. Une autre solution, si ils ont un iPad, est de télécharger la version EPUB du livre, de l'ajouter à iTunes et de synchroniser l'iPad.

Commençons !

# Chapitre 1 : Bonjour, ligne de commande !

Ce chapitre parle de la ligne de commande !

Ce livre est fait pour des enfants qui, comme toi, utilisent un Mac, soit à la maison, soit à l'école.

Normalement tu utilises un Mac en cliquant avec une souris. Dans ce chapitre tu vas apprendre une nouvelle façon d'utiliser un Mac : en ligne de commande, uniquement avec le clavier.

Dans le passé, il y a plus de 30 ans, le seul programme disponible pour l'utilisateur sur tous les ordinateurs était la ligne de commande.

Même si la ligne de commande est simple, elle est très puissante. Si tu sais comment l'utiliser, tu sera comme un magicien qui, en écrivant des textes étranges, fera faire des choses incroyables à ton ordinateur.

Relax, ce n'est pas effrayant. Si tu commets une erreur, tu ne casseras pas tout !

Pour commencer, apprenons les bases.

Tu sais déjà ce qu'est un clavier. Mais il y a certaines touches que tu n'as peut-être encore jamais utilisées. Il s'agit des touches Commande (appelée aussi Cmd ou Pomme), Option (appelée aussi Alt), Contrôle (Ctrl) et Shift (appelée aussi Maj).





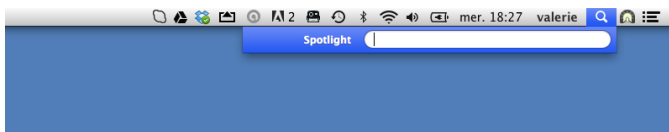
Les touches **Commande**, **Option**, **Contrôle** et **Shift**

Nous allons lancer un programme qui s'appelle **Terminal**. **Terminal** permet d'envoyer des commandes à ton Mac.

Pour démarrer **Terminal**, appuie sur la touche **Contrôle** et, tout en la maintenant enfoncée, appuie sur la barre espace en même temps. (Une façon plus simple d'expliquer cette manipulation est de dire “appuie **Contrôle** + **Espace**. C'est de cette façon que je vais t'expliquer comment faire par la suite.)

Taper **Contrôle** + **Espace** ouvre un programme appelé **Spotlight** dans le coin en haut à droite de ton écran.

Si ce n'est pas le cas, tu peux aussi ouvrir **Spotlight** en cliquant avec ta souris sur la loupe que est dans la barre en haut à droite de ton écran.



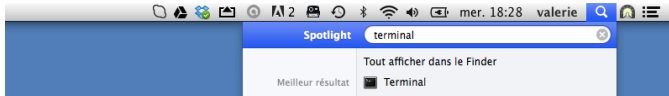
**Spotlight**

**Spotlight** te permet d'écrire le nom du programme que tu veux lancer.

Nous allons lancer **Terminal**. Donc, tape **terminal** dans **Spotlight** et appuie sur la touche **Entrée**.

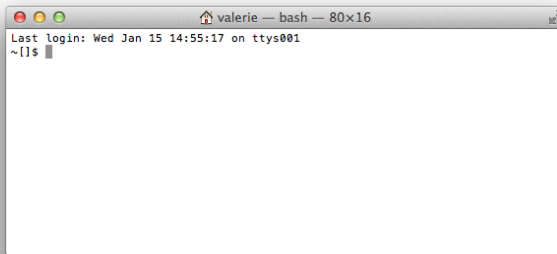
Quand tu auras à peine commencé à taper, tu devrais voir une liste déroulante apparaître. Si c'est le cas et que tu vois **Terminal** dans

la liste des résultats, tu peux directement cliquer dessus sans avoir besoin de finir de taper **terminal**.



### Lancer Terminal

Tu devrais voir Terminal, qui ressemble à quelque chose comme ça :



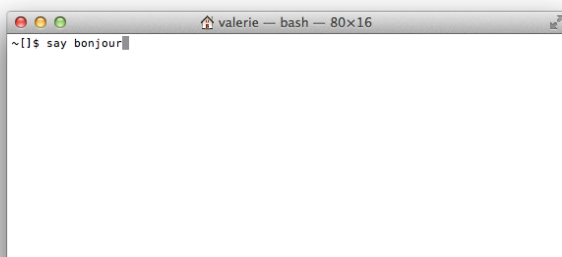
### Terminal

Terminal est la ligne de commande du Mac.

(Ne t'inquiètes pas des caractères `~[]$` au début de la ligne. Mon “invite de commande” est comme ça sur mon ordinateur, il est possible de modifier cela. Ton invite de commande devrait ressembler à `>` ou `~.`)

Avant de commencer, essayons de faire dire bonjour à la ligne de commande. Pour celà, nous allons utiliser la commande **say**. (“say” est un mot anglais qui signifie “dire”)

Ecris **say bonjour** dans Terminal et appuie sur Entrée.

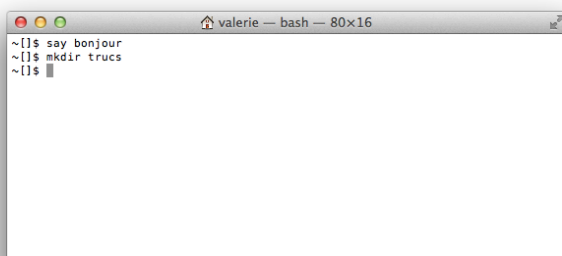


Normalement ton Mac devrait te dire bonjour ! Si ce n'est pas le cas, vérifie que le son de ton ordinateur est allumé et que le volume est assez fort.

Tu remarqueras que ton ordinateur ne parle pas bien français, il a un fort accent anglais :)

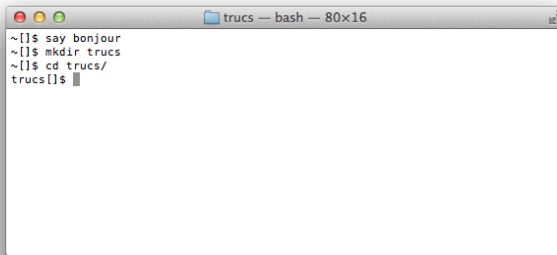
Je tiens à préciser que quelque soit la commande que tu entres dans Terminal, tu as toujours besoin d'appuyer sur la touche Entrée du clavier pour lancer la commande.

Maintenant commençons par créer un dossier qui contiendra tous les trucs que nous allons faire dans ce livre. Ecris **mkdir trucs** dans Terminal et appuie sur la touche Entrée.



La commande **mkdir** est une version courte de “make dir”, c’est à dire “make directory”. En français cela signifie “créer un dossier”. Tu viens donc de créer un dossier qui s’appelle **trucs**.

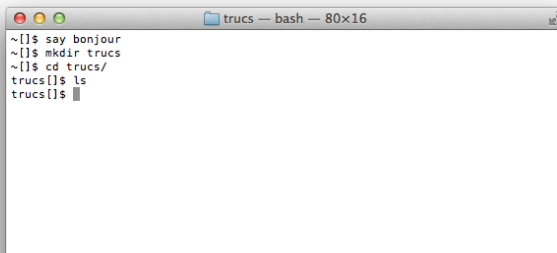
Ensuite, nous allons nous placer dans ce dossier. Ecris **cd trucs** dans Terminal et appuie sur la touche Entrée.

A screenshot of a macOS Terminal window. The title bar shows a folder icon, the text 'trucs — bash — 80x16', and a close button. The terminal content shows a sequence of commands and their outputs: '~[]\$ say bonjour' followed by a system sound icon; '~[]\$ mkdir trucs'; '~[]\$ cd trucs/'; and finally 'trucs[]\$' with a cursor. The window has a light gray border and standard macOS window controls (red, yellow, green buttons) in the top-left corner.

```
~[]$ say bonjour
~[]$ mkdir trucs
~[]$ cd trucs/
trucs[]$
```

La commande **cd** signifie “change directory”, qu’on traduit par “change de dossier”. Tu as donc changé de dossier pour aller dans celui appelé **trucs** que tu viens juste de créer.

Regardons ce qu’il y a à l’intérieur de ce dossier. Dans Terminal, écris **ls** (L minuscule suivi de S minuscule) et appuie sur la touche Entrée.

A screenshot of a macOS Terminal window, similar to the one above. The title bar shows 'trucs — bash — 80x16'. The terminal content shows the same sequence of commands as before, but with an additional command: 'trucs[]\$ ls', which results in an empty output line, indicating the directory is currently empty. The window has a light gray border and standard macOS window controls in the top-left corner.

```
~[]$ say bonjour
~[]$ mkdir trucs
~[]$ cd trucs/
trucs[]$ ls
trucs[]$
```

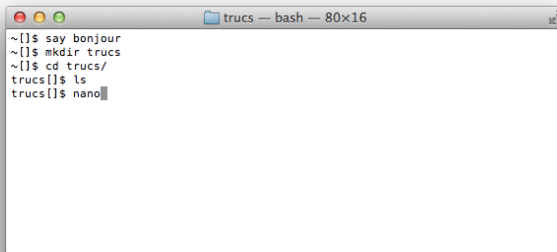
Taper **ls** ne montre rien du tout !

La commande **ls** signifie “list” (en français “liste”), son rôle est de montrer les fichiers et dossiers qui sont contenus dans ton dossier. Il n’y a rien dans le dossier **trucs** puisque nous venons juste de le créer. C’est pour cela que taper **ls** ne montre rien.

Changeons cela en créant un fichier.

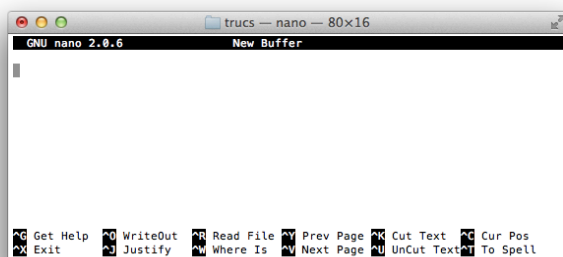
Nous allons utiliser un programme qu’on appelle un éditeur de texte pour créer un fichier. Un éditeur de texte te permet d’écrire des mots dans un fichier, et de sauvegarder ce fichier tout comme tu le fais lorsque que tu sauvegardes une partie dans ton jeu vidéo. Nous allons utiliser un petit éditeur de texte appelé **nano**. Nano veut dire “vraiment petit”, ce nom est parfait parce que **nano** est vraiment un éditeur de texte très simple.

Dans Terminal, écris **nano** et appuie sur la touche Entrée.

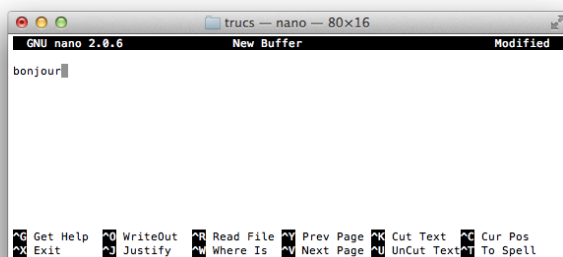


```
trucs — bash — 80x16
~[]$ say bonjour
~[]$ mkdir trucs
~[]$ cd trucs/
trucs[]$ ls
trucs[]$ nano
```

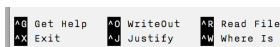
Cette commande lance nano. Ton écran devrait ressembler à ça :



Dans nano, écris **bonjour**. Tu n'as pas besoin de finir en appuyant sur la touche Entrée.

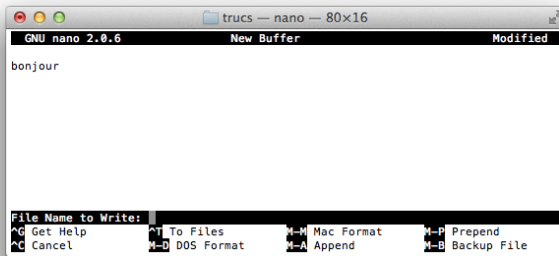


Tu peux voir que nano t'aide en te donnant la liste des commandes que tu peux utiliser en gardant la touche **Contrôle** enfoncée et en tapant sur une lettre en même temps. (Nano représente la touche **Contrôle** lorsqu'il affiche le caractère ^, il ne s'agit pas de taper sur la touche ^ du clavier mais bien d'utiliser la touche **Contrôle**)

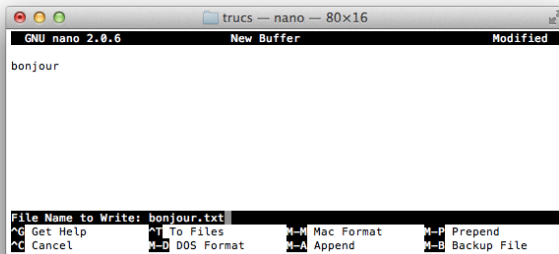


Pour sauvegarder le fichier, garde la touche **Contrôle** enfoncée tout en appuyant sur la lettre **o** de ton clavier.

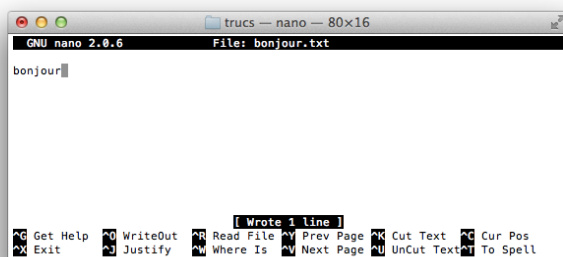
Nano te demande de lui indiquer le nom que tu veux donner à ton fichier, en affichant le texte “File Name to Write :” (“Nom Du Fichier à Ecrire :”).



Ecris **bonjour.txt** et appuie sur la touche Entrée.



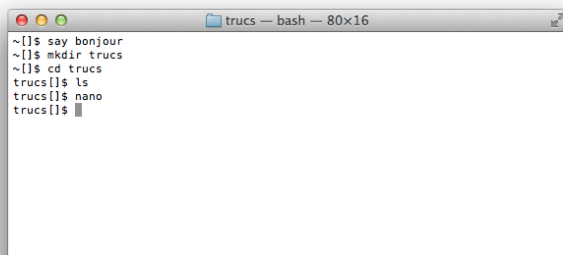
Nano sauvegarde ton fichier et t’informe que ce fichier contient 1 ligne, en affichant “Wrote 1 line” (“1 ligne écrite”).



(Si jamais tu as tapé sur la touche Entrée après avoir écrit “bonjour”, nano affichera “Wrote 2 lines” (“2 lignes écrites”). Ne t’inquiètes pas, ce n’est pas grave.)

Maintenant que le fichier est sauvegardé, quittons nano en gardant la touche **Contrôle** enfoncée tout en appuyant sur la lettre **x** de ton clavier. (Une façon plus simple de t’expliquer la manipulation à faire est de te dire “appuie **Contrôle** + **x**”. C’est ce que nous allons dire à partir de maintenant dans la suite du livre.)

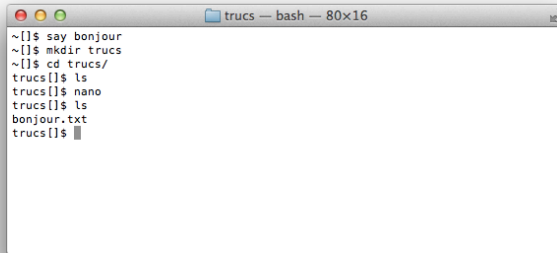
Tu es de retour en ligne de commande dans Terminal.



Regardons maintenant ce qu’il y a à l’intérieur de notre dossier **trucs**. Souviens-toi, la dernière fois que nous avons tapé **ls**, il n’y avait rien du tout. Ecris à nouveau **ls** dans Terminal. Tu verra le



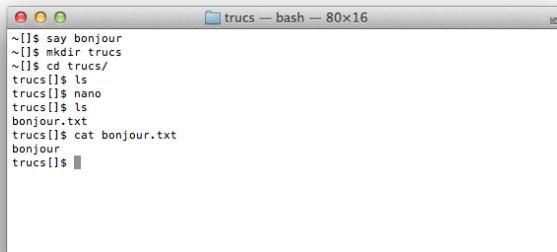
fichier `bonjour.txt` que tu viens de créer.



```
trucs — bash — 80x16
~[]$ say bonjour
~[]$ mkdir trucs
~[]$ cd trucs/
trucs[]$ ls
trucs[]$ nano
trucs[]$ ls
bonjour.txt
trucs[]$
```

Regardons ce qu'il y a dans ce fichier. Ecris `cat bonjour.txt` dans Terminal et appuie sur la touche Entrée.

Ici `cat` n'est pas le chat, l'animal. Il s'agit d'une commande qui affiche ce que contient un fichier. Comme nous avons écrit `bonjour` avant, c'est ce que tu devrais aussi voir maintenant :



```
trucs — bash — 80x16
~[]$ say bonjour
~[]$ mkdir trucs
~[]$ cd trucs/
trucs[]$ ls
trucs[]$ nano
trucs[]$ ls
bonjour.txt
trucs[]$ cat bonjour.txt
bonjour
trucs[]$
```

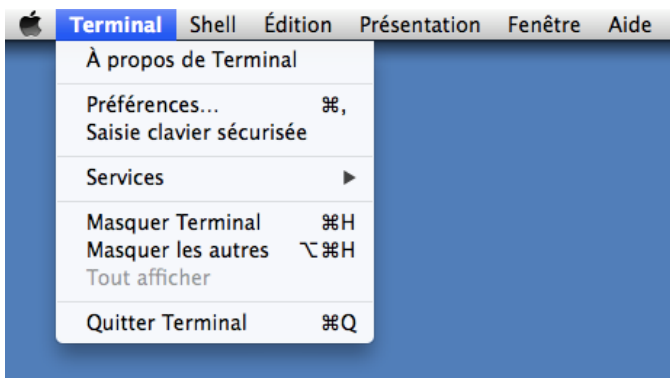
Le fichier `bonjour.txt` contient le mot "bonjour", donc en écrivant `cat bonjour.txt` tu as vu apparaître le mot `bonjour`.

Tu connais maintenant quelques commandes de base de la ligne de commande ! On vient de voir un ensemble de commandes étranges comme `mkdir`, `cd`, `ls` et `cat`. Ce qui est intéressant c'est que toutes

ces commandes sont en fait des programmes écrits par d'autres personnes !

Dans le prochain chapitre **tu** vas créer **ton** premier programme, que **tu** pourras lancer depuis la ligne de commande !

Enfin, nous allons quitter Terminal. Appuie sur **Command + Q** dans Terminal ou tu peux aussi choisir "Quitter Terminal" depuis le menu Terminal.



Si jamais il y a quelque chose que tu ne comprends pas en ligne de commande et que tu ne sais plus comment t'en sortir, tu peux toujours quitter Terminal et l'ouvrir à nouveau pour réessayer.

A toi de jouer maintenant, fais les exercices de la page suivante avant de continuer !

## Exercices

1. Lance Terminal et place toi (`cd`) dans le dossier `trucs` .
2. Utilise `nano` pour créer un fichier appelé `hourra.txt` qui contient le texte “hourra”. Quitte `nano` lorsque tu as sauvegardé.
3. Utilise la commande `cat` pour afficher le contenu du fichier `hourra.txt`.
4. Quitte Terminal.

Tu peux consulter les [réponses des exercices](#), ou simplement aller au chapitre suivant.

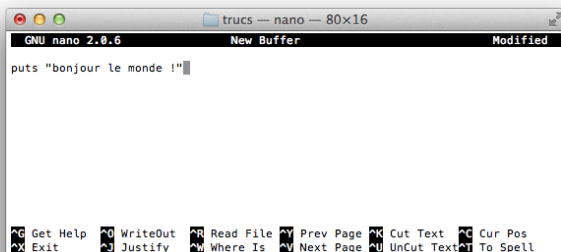
# Chapitre 2 : Bonjour le monde !

Allons-y, écrivons notre premier programme !

Démarre Terminal, place toi (`cd`) dans le dossier `trucs` et lance `nano`. (Si cette phrase n’a pas de sens pour toi, regarde [Chapitre 1 Exercices](#).)

Nous allons créer un programme vraiment très simple. Ecris `puts "bonjour le monde !"` dans `nano`.

Tu n’as pas besoin d’appuyer sur Entrée. Je t’expliquerai plus tard ce que signifie `puts`.



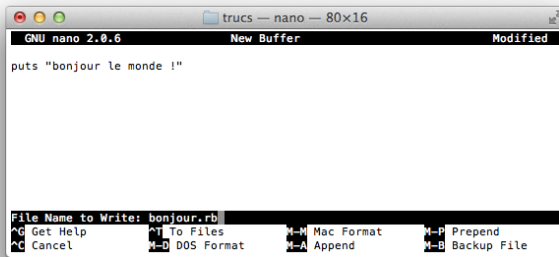
Et voilà !

Appuie **Contrôle + o** pour sauvegarder le fichier.

Nano te demande de lui indiquer le nom du fichier en affichant “File Name to Write :” (“Nom Du Fichier A Ecrire”).

Ecris `bonjour.rb` et appuie sur la touche Entrée. Tu devrais voir ce que tu tapes dans la zone à côté de “File Name to Write” en bas de

la fenêtre.

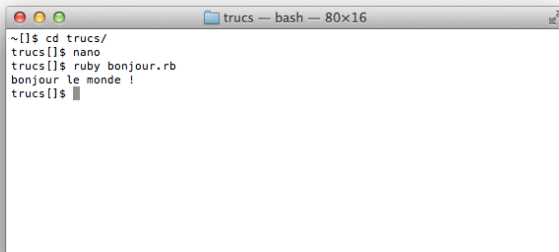


Nano va sauvegarder ton fichier et te dire qu'il contient une ligne ("Wrote 1 line"). (Si tu as appuyé sur la touche Entrée après le texte, nano dira que le fichier contient 2 lignes. C'est bon quand même.)

Maintenant que tu as sauvegardé ton fichier, appuie **Contrôle + x** pour quitter nano.

Executons ton programme !

Dans Terminal, écris **ruby bonjour.rb** et appuie sur la touche Entrée. Il faut un espace entre **ruby** et **bonjour.rb**.



**Ton premier programme !**

Félicitations, tu viens d'écrire ton premier programme ! Comme tu as pu le constater, **puts** écrit quelque chose à l'écran.

Mais, que signifie le mot "ruby" ?

Ruby est un langage de programmation. Tu parles français, mais il existe beaucoup d'autres langues parlées dans le monde. Dans la même logique, il y a beaucoup de langages de programmation différents pour dire à l'ordinateur ce qu'il doit faire. Ruby est l'un des plus simple à utiliser, et il est installé de base sur ton Mac.

Le programme que tu viens d'écrire est un programme en Ruby ! L'extension de ton fichier (ce que tu as écrit après le `.` dans le nom de ton fichier) pour Ruby est `rb`, c'est pour cela que nous avons nommé le fichier `bonjour.rb`.

Dans le prochain chapitre, nous allons jouer avec Ruby dans sa propre ligne de commande !

## Exercices

1. Ecris et lance un programme Ruby appelé `salut.rb` qui affiche "salut !" à l'écran.

Tu peux consulter les [réponses des exercices](#), ou simplement aller au chapitre suivant.

# Ceci est un extrait

Ceci est un extrait d'un livre disponible à la vente sur Leanpub :  
[<https://leanpub.com/laprogrammationpourlesenfants>] (<https://leanpub.com/laprogrammationpourlesenfants>).

Il s'agit de la traduction française de <https://leanpub.com/programmingforkids><sup>1</sup>

Le livre complet compte actuellement plus de 10 chapitres. Pour soutenir la création de ce livre, n'hésitez pas à l'acheter même si il est en cours de rédaction.

Sachez que 30% des droits d'auteurs seront reversés à Watsi [<https://watsi.org/>].

---

<sup>1</sup><https://leanpub.com/programmingforkids>

# Pour les parents

Ce livre est fait pour vous et votre enfant, mais votre enfant peut aussi le lire seul.

Je l'ai écrit pour apprendre les bases de la programmation à mon fils de 9 ans. Il joue aux jeux vidéos depuis plusieurs années, il a voulu se mettre à la programmation quand l'envie de développer son propre jeu vidéo a germé dans son esprit. Ce livre est une première étape pour atteindre ce but. (Non ce livre ne vous apprend pas à développer le jeu vidéo dont rêve un enfant de 9 ans ; il faut beaucoup plus de connaissances pour cela !)

Ce livre a pour objectif de devenir le meilleur livre au monde pour les enfants qui souhaitent apprendre la programmation. L'apprentissage devrait se faire en commençant par lire ce livre en premier. La programmation informatique est une compétence utile, peu importe ce que fera votre enfant une fois adulte. (Je suis persuadé que, par exemple, la programmation est bien plus importante qu'un bon nombre de cours de mathématique enseignés à l'école). Mais ce qui est bien plus important, c'est qu'apprendre à réaliser un programme informatique permet de développer rigueur et discipline dans la façon de penser. Ce livre existe pour montrer aux enfants qu'ils peuvent programmer un ordinateur, et pour leur donner envie d'aller peut-être plus loin par la suite.

Mis à part cette annexe, ce livre est écrit pour les enfants d'école primaire et du collège (en France). Mon but est de le rendre accessible aux enfants de 9 à 12 ans. Mon fils a étudié le livre pendant que je le rédigeais. (Je ne pense pas que ce livre soit adapté aux enfants de moins de 8 ans : j'avais commencé à écrire le livre quand mon fils avait 7 ans, mais il n'était pas encore prêt. J'ai donc attendu 2 ans avant de me remettre à la rédaction. Si votre enfant a moins de 8 ans, je vous conseille de plutôt vous orienter vers un logiciel comme



Scratch qui sera plus adapté à son âge.)

Vous n'avez pas besoin d'avoir de connaissances en programmation. Les exemples sont aussi courts que possible, j'ai pris en compte le fait que les lecteurs ne sont pas forcément à l'aise avec un clavier. (Mon fils ne sait pas taper au clavier, donc si les exemples sont trop longs, il me le fait bien savoir !)

Mon but pour ce livre est qu'il devienne le meilleur des premiers livres à lire pour apprendre à programmer pour les enfants. Après ce livre, à eux de décider vers quoi ils veulent s'orienter.

Ce livre est écrit en supposant que vous utilisez un Mac. Je pense que le Mac est le meilleur ordinateur pour apprendre à un enfant à programmer. Comme il s'agit d'un livre à destination d'enfants débutants, je n'ai pas voulu l'écrire pour qu'il s'adapte au Mac, à Window et à Linux. J'ai choisi un système d'exploitation pour que l'enfant puisse suivre les étapes sans être dérangé par des informations qui ne le concernent pas.

En plus d'apprendre la programmation, ce livre enseigne également les bases de l'utilisation de la ligne de commande sous Mac, en utilisant le programme Terminal. J'ai fait ce choix parce que pour moi la meilleure façon d'apprendre est de suivre les exemples pas à pas, et la façon la plus efficace de suivre est de tout écrire soi-même. Les vrais programmeurs utilisent la ligne de commande tous les jours. Si vous voulez apprendre à développer, vous devriez utiliser Terminal et le système de fichiers. Oui, il est possible d'apprendre des choses en passant par votre navigateur web avec des sites comme Codecademy, même si cela est sympa et instructif, cela est fondamentalement différent de ce que fait un programmeur. Et bien que ce soit plus facile avec ces plateformes, l'apprentissage est moins gratifiant.

Si vous laissez votre enfant utiliser **votre** Mac, je vous recommande **fortement** de rester à côté de lui et de suivre les chapitres avec lui ! Par exemple, je ne prévois pas d'apprendre la commande pour

supprimer les fichiers, mais votre enfant pourrait la trouver et mal l'utiliser.

Les exemples sont en Ruby. Ruby est vraiment un langage de programmation simple. Si vous avez déjà entendu parler d'un site internet créé avec "Ruby on Rails", c'est que vous avez entendu parlé de Ruby : c'est le langage de programmation utilisé pour créer Rails. Mais ce n'est pas un livre pour apprendre à développer avec Ruby. Les exemples auraient pu facilement être écrits pour fonctionner avec JavaScript, CoffeeScript ou Python.

Enfin, je suis toujours intéressé par vos retours ! Est-ce que votre enfant a rencontré des difficultés à suivre le livre ?

Si vous avez des remarques ou des questions à propos du livre, faites-le moi savoir ! N'hésitez pas à m'envoyer un email à [peter@leanpub.com](mailto:peter@leanpub.com).

# A propos de l'auteur

Je suis le fondateur de Ruboss, une société de consulting informatique basée à Vancouver au Canada. Nous sommes les créateurs de Leanpub, un site internet accessible à tous ceux qui veulent publier leurs e-book en même temps qu'ils l'écrivent, ce qui a été fait pour ce livre. Je suis aussi un programmeur, un auteur et un père. J'ai écrit deux livres pour les développeurs (*Flexible Rails* et *Hello ! Flex 4*), je sais donc comment expliquer les choses aux programmeurs. Mon fils est un enfant de 9 ans très intelligent, j'ai donc pu expérimenter le fait de donner des explications à des enfants. Ce livre est un vrai challenge, et je voudrais sincèrement qu'il devienne le meilleur livre pour apprendre les bases de la programmation à tous les enfants.

# A propos de la traduction

La traduction a été faite par Valérie Taesch, ingénieur en développement logiciel. Impliquée dans des manifestations qui ont pour but de sensibiliser les enfants à la programmation (Devovx4Kids [[http ://www.devovx4kids.org/](http://www.devovx4kids.org/)]), ce livre représente pour elle un nouveau moyen de leur faire découvrir le monde de l'informatique.

Il est parfois délicat de traduire ou d'expliquer des commandes qui sont de base en anglais, alors si vous notez des maladresses ou des erreurs, n'hésitez pas à la contacter par email [valerie.taesch@gmail.com](mailto:valerie.taesch@gmail.com).

# A propos de Leanpub

Ce livre est un livre de Leanpub. Je suis un des co-fondateurs de Leanpub. Leanpub est un site internet qui permet à n'importe qui de publier un e-book alors même qu'il est en cours de rédaction, j'appelle cela le Lean Publishing. Pour plus d'information, je vous recommande ce [livre gratuit] (<https://leanpub.com/lean>) (en anglais) que j'ai écrit l'an dernier.

En plus d'être totalement gratuit, être un auteur sur Leanpub est aussi très avantageux. Nous payons les auteurs 90% moins \$0,50 de droits d'auteur par copie du livre vendue. Un livre acheté \$10 permettra à l'auteur de toucher \$8.50 par copie du livre !