

# Technical Writer

90 days to write, self-  
publish and sell your  
first technical book



Sylvain Kerkour

# Technical Writer

90 days to write, self-publish and sell your first technical book

Sylvain Kerkour

v2023.1

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	What is a book, actually? . . . . .	2
1.2	Why write a book . . . . .	2
1.3	Why not a book . . . . .	3
1.4	Early Access . . . . .	4
<b>2</b>	<b>Writing</b>	<b>5</b>
2.1	Why your book . . . . .	5
2.2	The tools . . . . .	7
2.3	Transmitting knowledge . . . . .	14
2.4	Avoiding the writer's block . . . . .	17
2.5	Storytelling . . . . .	17
2.6	Editing your manuscript . . . . .	18
2.7	Summary . . . . .	19
<b>3</b>	<b>Publishing</b>	<b>20</b>
3.1	Why you may want to work with a publisher . . . . .	20
3.2	Why you actually want to self-publish your book . . . . .	20
3.3	Which price? . . . . .	21
3.4	From Markdown to ebooks . . . . .	22
3.5	Selling your ebooks . . . . .	34
3.6	Avoiding piracy . . . . .	35
3.7	Refunds . . . . .	37
3.8	Summary . . . . .	37
<b>4</b>	<b>Marketing</b>	<b>38</b>
4.1	Where to promote your book . . . . .	38
4.2	When to promote . . . . .	40
4.3	Momentums . . . . .	41
4.4	Tracking the origin of your sales . . . . .	41
4.5	Tips and tricks to boost your sales . . . . .	41
4.6	Summary . . . . .	42
4.7	Contact . . . . .	43

+++ title = "Technical Writer - Introduction" date = 2021-06-01T6:00:00Z type =  
"page" url = "/technical-writer/introduction" access = "paid\_members" +++

# Chapter 1

## Introduction

### 1.1 What is a book, actually?

I know that you know what a book is... A few sheets of paper strapped together that you can buy in a bookstore.

But that's not the answer I was waiting for.

A book is a **knowledge product** (also called information product, but I prefer to use the term *knowledge*)

It's a way to transfer knowledge from one Human to many others, and, most of the time, be paid to do so.

Books are not the only existing knowledge products: - video courses - podcasts - online courses

There are many ways to share and sell knowledge.

So, why would someone want to write a book?

### 1.2 Why write a book

**Everyone knows what a book is:** Books exist since thousands of years, and thus, and thus are here to stay, according to the [Lindy effect](#). Most people buy and read books since they are children, so this behavior is already anchored into the habits of customers.

**Making money:** As we will see in chapter 3, by self-publishing your book, you can make decent money. As I'm writing this, a little bit more than 1 year after the initial

announcement of my book, I made more than \$80,000 in sales.

**Being recognized by your peers:** Writing a book will give you certain visibility in your community which can lead to many opportunities such as interviews, job offers, or freelance contracts.

**Text and illustrations easy to update:** It's very hard to update a video or a podcast, that's why so few people bother and that video and audio creators treat their output as fire and forget. On the other hand, it's really easy to update text and illustrations so you can offer lifetime updates to your readers or sell a 2nd edition of your book a few years after the first edition.

**It requires no investment:** You can write, publish and sell a book with a simple smartphone. Of course, I do not recommend that, but I want to show you that you don't need to purchase a new camera, a powerful computer for video editing, huge hard disks or whatever to write a book.

**You can write a book from anywhere:** Related to the point above, as the stuff required to write a book is minimal, you can do it from anywhere: a big city, a cabin in the woods or while traveling.

But, writing a book, there also are many, many reasons why someone would **NOT** want to write a book.

## 1.3 Why not a book

**Undervalued:** Most people think that a book should cost between \$10 and \$30 so it's hard to sell your book at a higher price point. Thus, to make a profit, you'll need to sell at least one thousand books.

I will tell you an open secret: [Amazon Kindle Direct Publishing \(KDP\)](#) takes a 70% cut if the price of your book is less than \$2.99 or more than \$9.99, while, if your book costs between \$2.99 and \$9.99, they take "only" 30%. And this is before taxes. This is why you can find so many cheap books on Amazon, that take ~100 pages to explain and illustrate only 1 idea...

**It takes a looot of time:** If you are not committed full-time to your book, it can take many months to 1+ year to complete your book. That being said, by following the strategies of this book, you should be able to write your book in 90 days.

**A book needs to be updated:** Whether it be frameworks, tools, or best practices, technology evolves. For example, while writing my book about the Rust programming

language, between the start and the publication dates, some features of the language evolved and I needed to update the book before having completed it... Worse, is that if you want to update your book after its publication and provide lifetime updates to your readers (like I do), it's not necessarily easy to distribute the updated version to all your customers.

## 1.4 Early Access

As mentioned above, writing a book is really time-consuming, so you certainly don't want to write a book that nobody is going to read.

The best way to make sure that enough people are interested by your book is to launch it after having written only the first chapter.

Before pre-selling your product, you should find the answers of the 2 following questions:  
- What is your target number of sales? This means: below how many sales do you abandon the project? - What will you do if you don't hit this number? Refund your customers? Pivot to something else?

### 1.4.1 The plan to launch your book in Early-Access

**Write down the plan:** X chapters. Take your time to define the plan of your book, because then it's really hard to change it. For each chapter, note down the 3-6 most important ideas you want to share.

**Write the 1st chapter:** Take 1 week to write the first chapter. Do your best because early-access buyers will judge you on it. Refer to chapter 2 to learn how to write a technical book.

**Setup a platform to sell your book and distribute the updates:** Which one to choose? We will see in Chapter 3.

**Write a landing page:** This is the link that you are going to send everywhere. This is your product page. How to convert visitors into customers? This is the topic of chapter 4.

**Finally, launch your book:** Which platforms? When to launch it? We will see everything you need to know in Chapter 4.

```
+++ title = "Technical Writer - Writing" date = 2021-06-01T6:00:00Z type = "page"
url = "/technical-writer/writing" access = "paid_members"
```

```
+++
```

# Chapter 2

## Writing

Now that you know the bigger picture it's time to stop procrastinating and start writing your manuscript.

For now, we don't care about formatting, we don't care about publishing. We only care about writing great content.

### 2.1 Why your book

#### 2.1.1 Finding your niche

Great ideas often come by associating 2 ideas. For example: Rust is the best language for offensive security.

If I search “[Javascript](#)” on the Amazon Kindle store, the query returns more than 5000 results. For sure, someone interested in learning JavaScript is not going to buy all these books.

**For who are you writing your book, and why should they be interested?**

If the answer is “everybody” or “because it's interesting” then think again.

Today you know and achieved things that you didn't 1 year ago. So there are thousands to millions of people who would like to be where you are now. These are your potential readers.

But this is not enough, you still need to clarify which specific thing you want to share to your readers.

For that, use the following template: **Teaching X to Y so they can achieve Z**

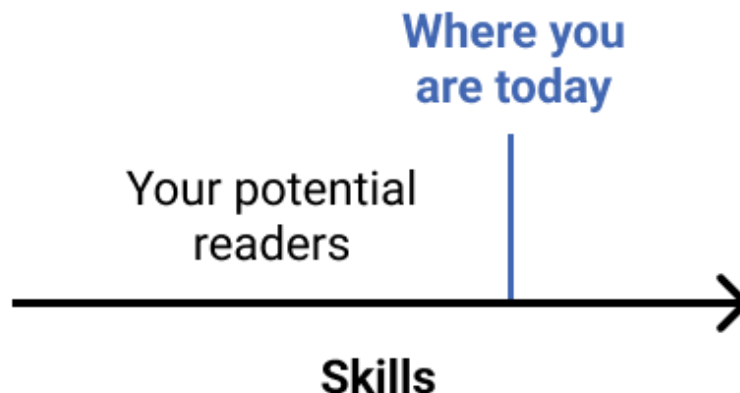


Figure 2.1: Skills

Where: - **X** is a specific system, framework, programming language... - **Y** is a specific community - **Z** is the final goal

For example: - Teaching offensive security to beginner Rust programmers so they can think like an attacker and create more secure programs. - Teaching Vue.js to intermediate JavaScript programmers so they can create Single Page Applications. - Teaching server programming to Go developers so they can build scalable backend applications.

This specific topic destined for a specific community is your **niche** and the community your **audience**.

### 2.1.2 Unique Value Proposition

Now that you have a niche, you need to find why people should choose YOUR book among many others that may already exist.

### 2.1.3 But, what is value

Defining value is hard, especially when we are talking about knowledge products, so let me show you first what value **is not**.

One day, I received an email telling me that my book was too expensive doing some comparison with another Rust book, to show that the price-per-page is too high and thus value not good.

I believe that the value of a knowledge product is the **exact opposite**.



The value of a knowledge product lies in how much time you can save by consuming it, rather than reading or watching tons of unstructured tutorials on the internet.

The maths are simple: if you are paid \$20 per hour, and a product costing \$100 saved you 100 hours, then you basically saved \$19,900 ( $\$20 * 100 \text{ hours} - \$100$ ).

In 2 words, it means that your book should provide a **simpler and faster** path to what your readers want.

**Simpler** because when writing about technology and science, it's easy to overwhelm your readers with too many not-so-relevant details. You should always assume that your readers already have some knowledge and that they can ignore more advanced details.

**Faster** because we are all racing against time. Why wait many years for something that you can get tomorrow or in 1 month? All entrepreneurs will tell you that it's better to be fast, and maybe fail fast, than to be too late to the party and lose the interest of potential customers.

It's always better to [focus on the 20% that provides 80% of the results](#).

## 2.2 The tools

“Traditional” authors use word processors such as Google Docs or Microsoft Words to write the manuscript of their book. While these tools provide great collaborative and built-in spellcheck features, you cannot use them if you want to include code sections in your book, or if you want to customize the exported files, as we will see in the next chapter.

Instead, we are going to use **Markdown**.

To my knowledge, Markdown is the most portable format available for writers today: There are countless editors and as many tools to work with it: static site editors, PDF generators, [Knowledge bases](#)...

To start writing your book, you only need a markdown editor. That's all, really. Even a text editor like notepad would be enough actually, just inconvenient.

### 2.2.1 Choosing a Markdown Editor

There are many, many markdown editors, I recommend you to try a few and pick the one you are the more comfortable with. [Don't overthink](#).

I personally prefer minimalist editors because when everything is barebone, you have no other choice than to focus on your work.

### 2.2.1.1 Cross-platforms

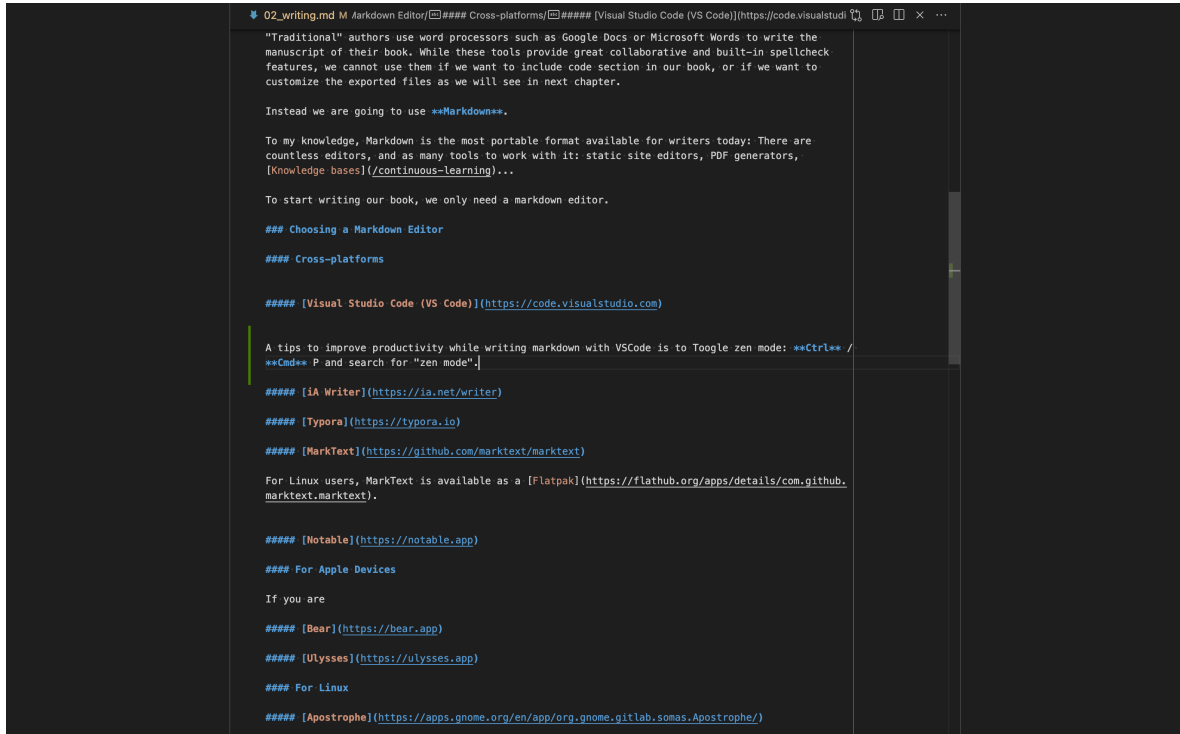


Figure 2.2: VS Code’s zen mode

**2.2.1.1.1 Visual Studio Code (VS Code)** A tip to improve productivity while writing markdown with VSCode is to toggle zen mode: **Ctrl** / **Cmd** P and search for “zen mode”.

**2.2.1.1.2 iA Writer** iA Writer is a minimalist and very clean markdown editor available for macOS, iOS & iPadOS, Android and Windows.

**2.2.1.1.3 MarkText** MarkText is an elegant and, more importantly, Open Source Markdown editor with a **WYSIWYG** (What You See Is What You Get) interface.

It’s unfortunately only available for computers.

For Linux users, MarkText is available as a **Flatpak**.

**2.2.1.1.4 Typora** Typora is another clean and minimalist markdown editor. Like MarkText, it’s unfortunately only available for computers.

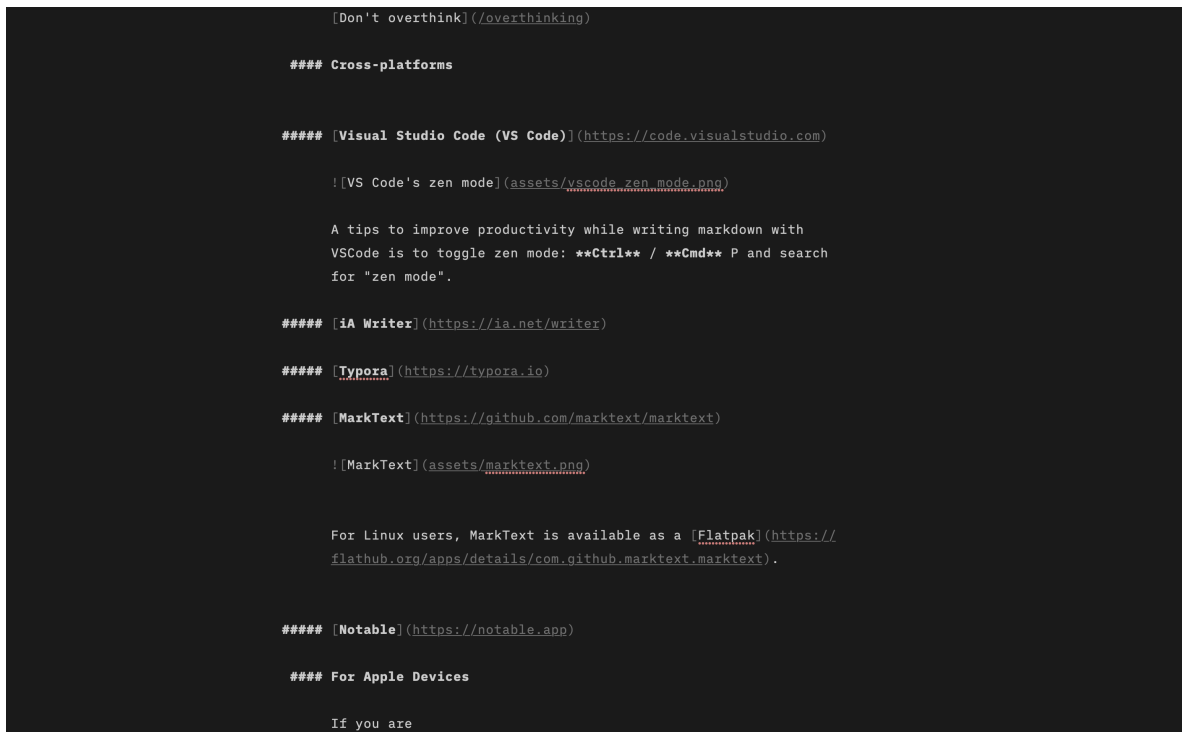


Figure 2.3: iA Writer

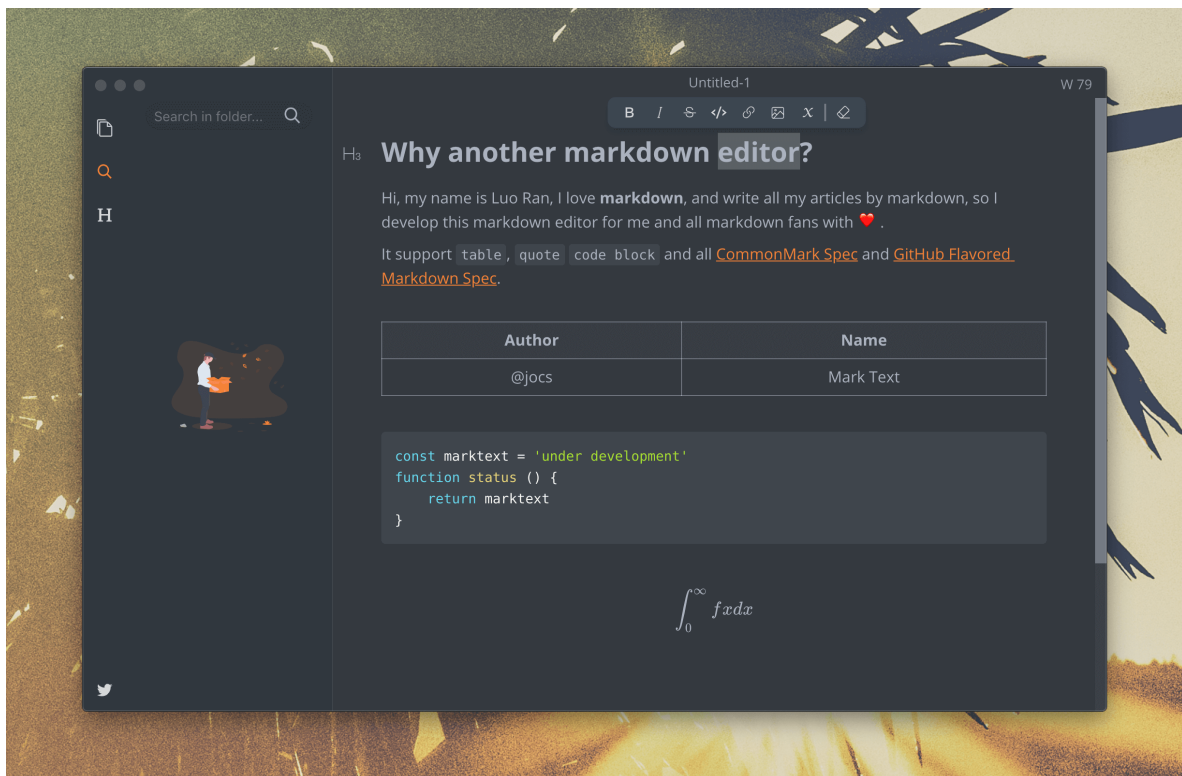


Figure 2.4: MarkText

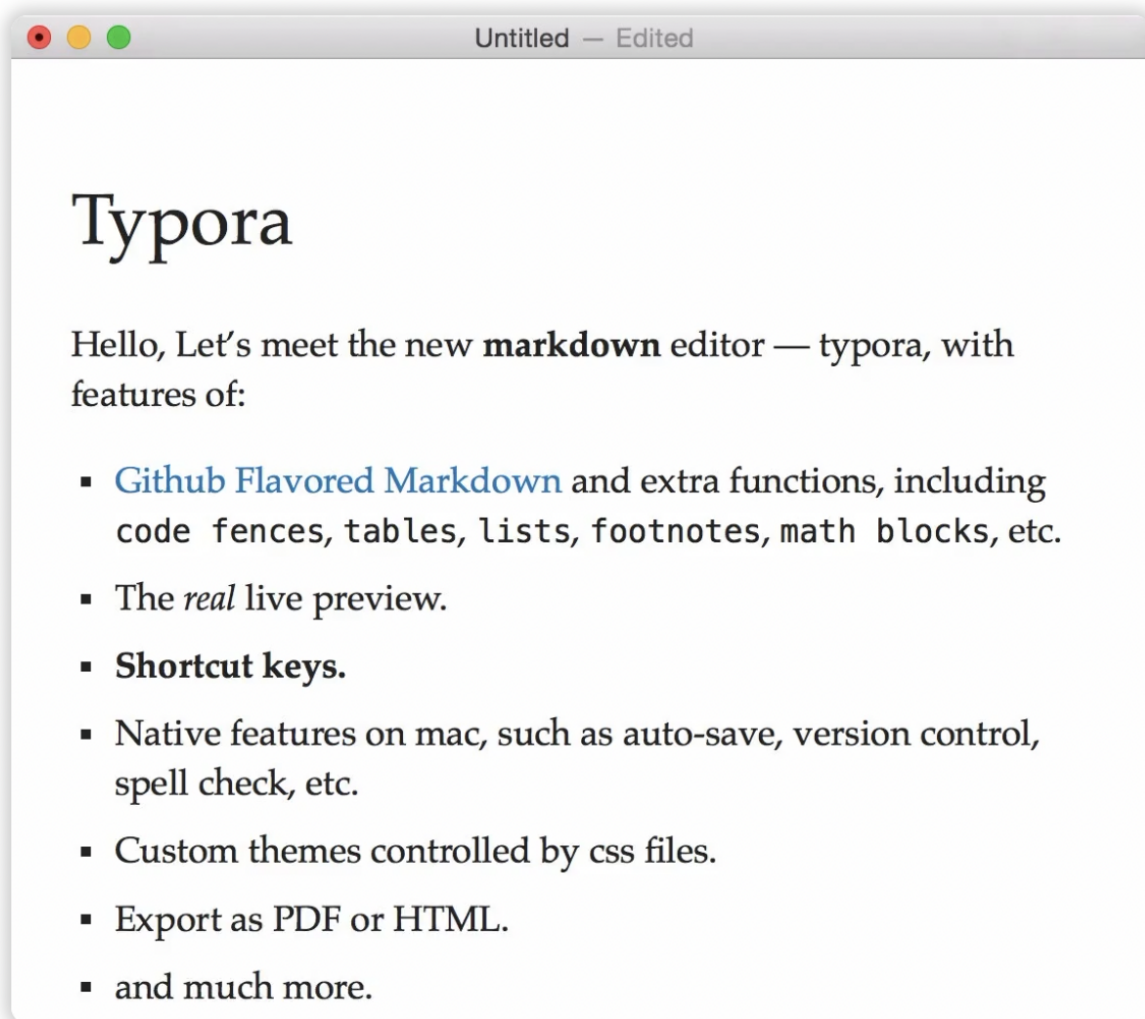


Figure 2.5: Typora

### 2.2.1.2 For Apple Devices

If you are hooked into the Apple universe, here are a few applications that leverage this ecosystem to provide a great experience.

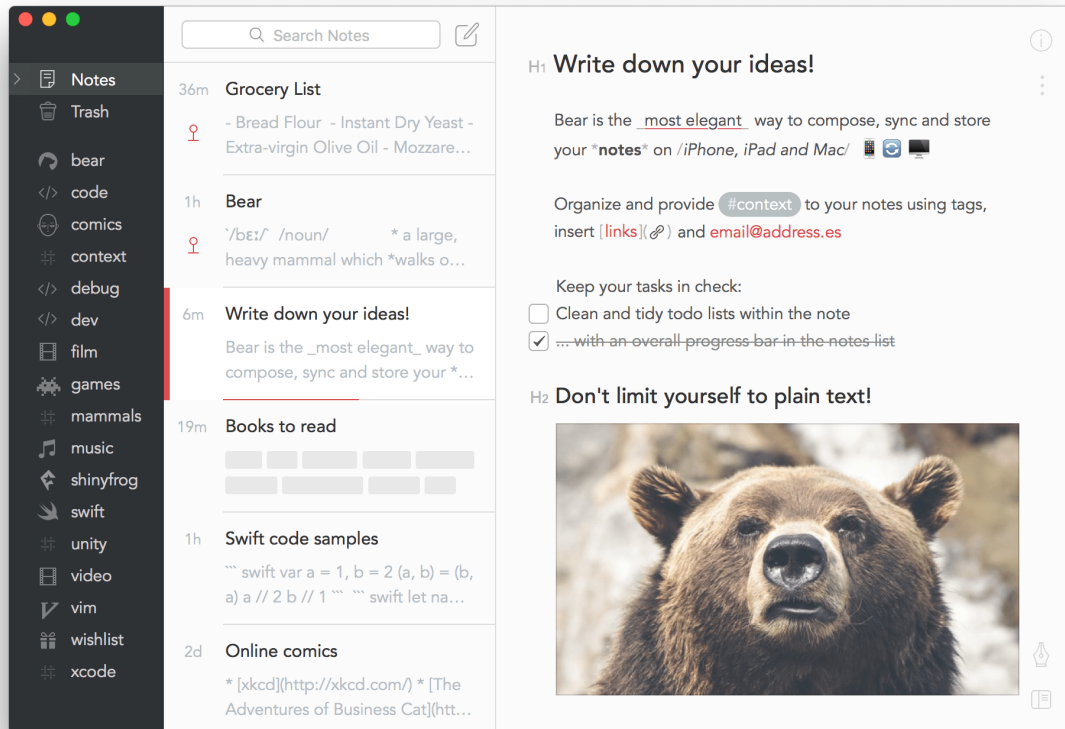


Figure 2.6: Bear

**2.2.1.2.1 Bear** Bear is more a Markdown notes application than a markdown editor. It provides advanced features for notes-taking, such as tags, exporting to Word docs, and encrypted notes.

The biggest problem with bear is that you can edit local files, only iCloud files, so you lose the ability to use your writing with other programs, as we will need to do to produce the ebooks files in the next chapter.

**2.2.1.2.2 Ulysses** Unlike Bear, you can edit local files with Ulysses which make it a great fit for our use case.

### 2.2.1.3 For Linux

On the other hand, if you are more an adept of the penguin, here are a few editors that you should consider.

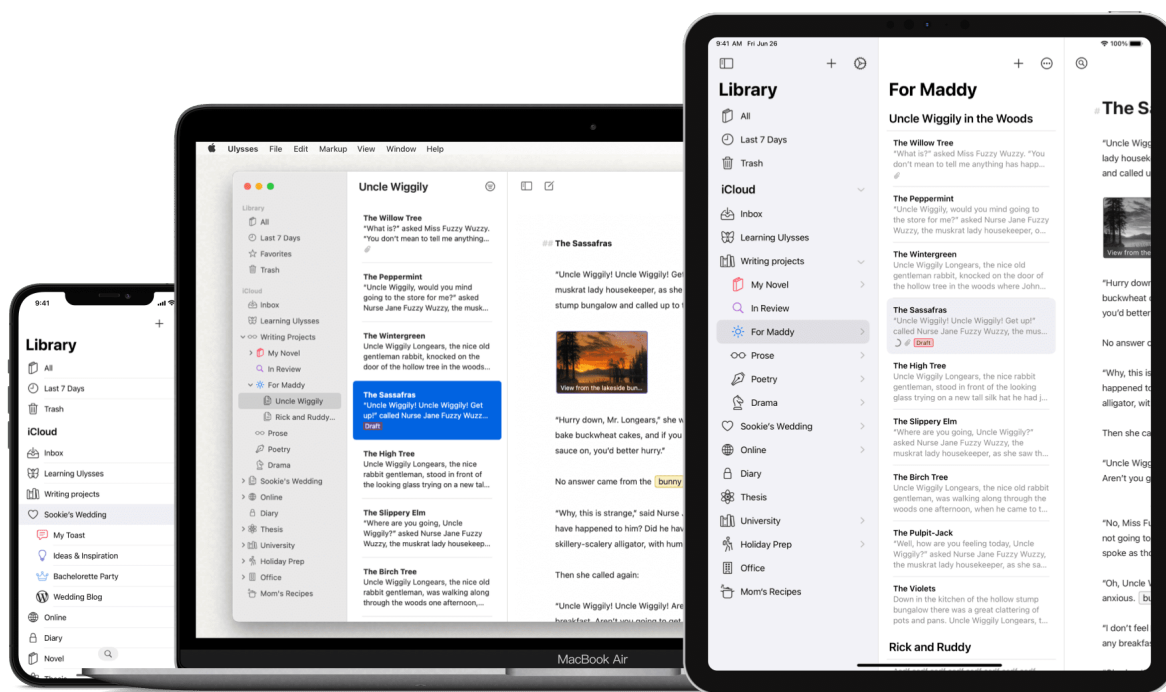


Figure 2.7: Ulysses

**2.2.1.3.1 Apostrophe** Apostrophe is a minimalist Markdown editor based on [GTK+](#) and licensed under the GNU General Public License v3.0 (GPLv3).

It uses pandoc under the hood to process Markdown (more on pandoc in chapter 3).

**2.2.1.3.2 ghostwriter** ghostwriter is a Windows and Linux text editor for Markdown, developed in C++ with the [Qt 5](#) framework.

#### 2.2.1.4 My favorite Markdown editor

I've personally settled on iA Writer: I do not only love its minimalistic aesthetic, but I also love that it has the same interface on computers, tablets and phones, which allows me to have a consistent workflow. The cherry on the cake is that contrary to some other markdown editors, it works directly with files on the filesystem, so I can view and edit my markdown files in other applications. A nice bonus is the built-in spell-checker, which saves a lot of time during the editing phase, where I can focus on wording and grammar.

I then use VSCode's Git integration to commit and push my files in a private Git repository.

On iOS devices, you can use the [Working Copy](#) application to synchronize your work with any Git repository.

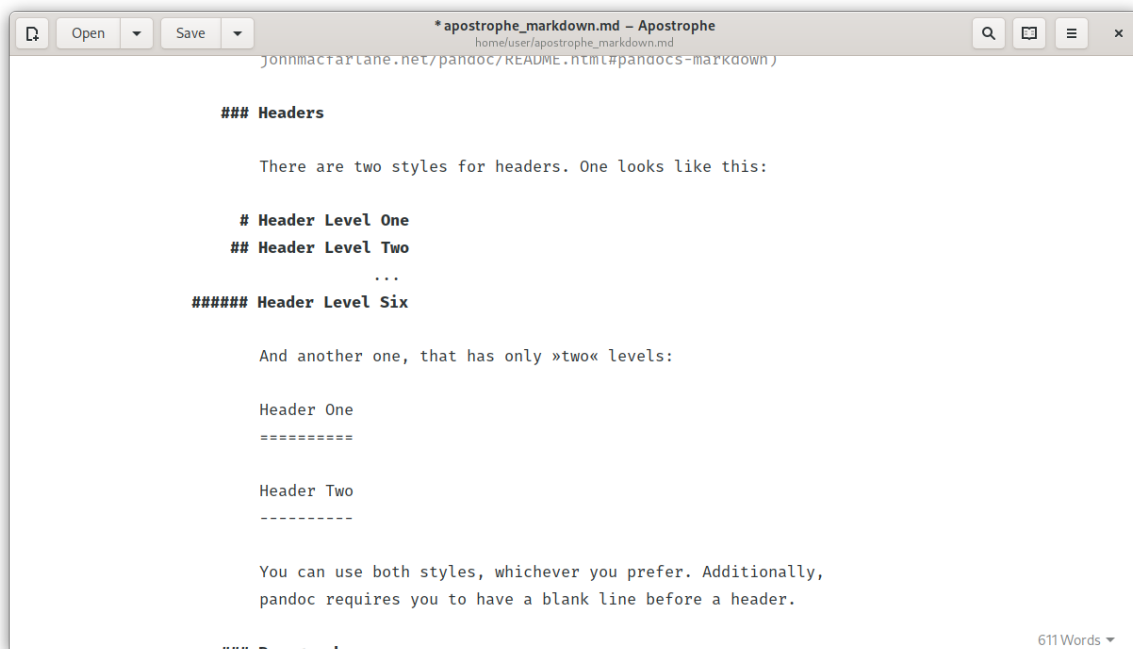


Figure 2.8: apostrophe

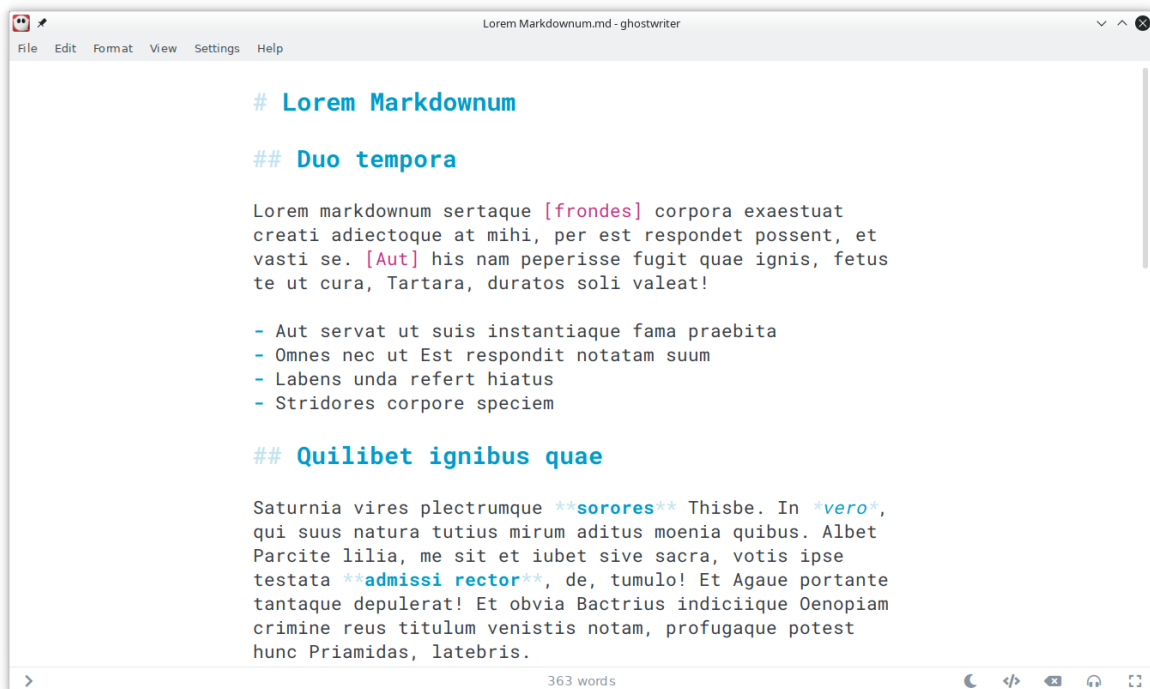


Figure 2.9: ghostwriter



## 2.2.2 Illustrations

As the saying goes: “A picture is worth a thousand words”.

Including illustrations in your book may allow you to explain complex things, simply (more on that below).

You want a tool that exports your illustration to the `.png` or `.jpg` formats? Why not `.svg` ? Because the [SVG image format](#) is a vector format, and not all tools handle it because it needs to be interpreted, while PNG and JPEG are raster formats which make them easy to display.

### 2.2.2.1 Figma

Figma

Figma is a relatively new-comer in the world of graphic design. Its unique value proposition is a web-based (and thus available on all platforms), live-editable by multi-users vector editor. While its principal use-cases are application mock-ups and logos, I personally use it for my illustrations because of how easy it is to use.

### 2.2.2.2 Diagrams.net

diagrams.net

Its principal advantage is its huge library of pre-made elements, such as AWS services, Android and iOS icons...

### 2.2.2.3 Excalidraw

excalidraw

Finally, there is [excalidraw](#) which provides a free and simple diagram editor. It’s unique value proposition is that the diagrams look as if they were hand-drawn.

### 2.2.2.4 A few things to remember

Because all these services are cloud services, they may automatically terminate your account, or cease operations overnight. **This is why you should always keep a local backup of the source files of your illustrations.** For Figma it means a `.svg` file, for diagrams.net a `.drawio` file.

## 2.3 Transmitting knowledge

As we saw in the last chapter, the value of a book lies in the knowledge it contains and how it is shared with the reader.



## 2.3.1 Writing the detailed plan

The first thing to do, is to write the detailed plan of your book with 3 levels of sub-headers:

# Chapter Title

## Section Title

### Important Point Title

Each **chapter** should discuss an important idea.

Each **section** should explain a method, a system or examples to approach the important idea of the chapter that you will detail in the **important points**.

You should then take a few days to review that this plan makes sense and is clear enough for your audience.

A nice tool to help you write the plan of your book is a MindMap.

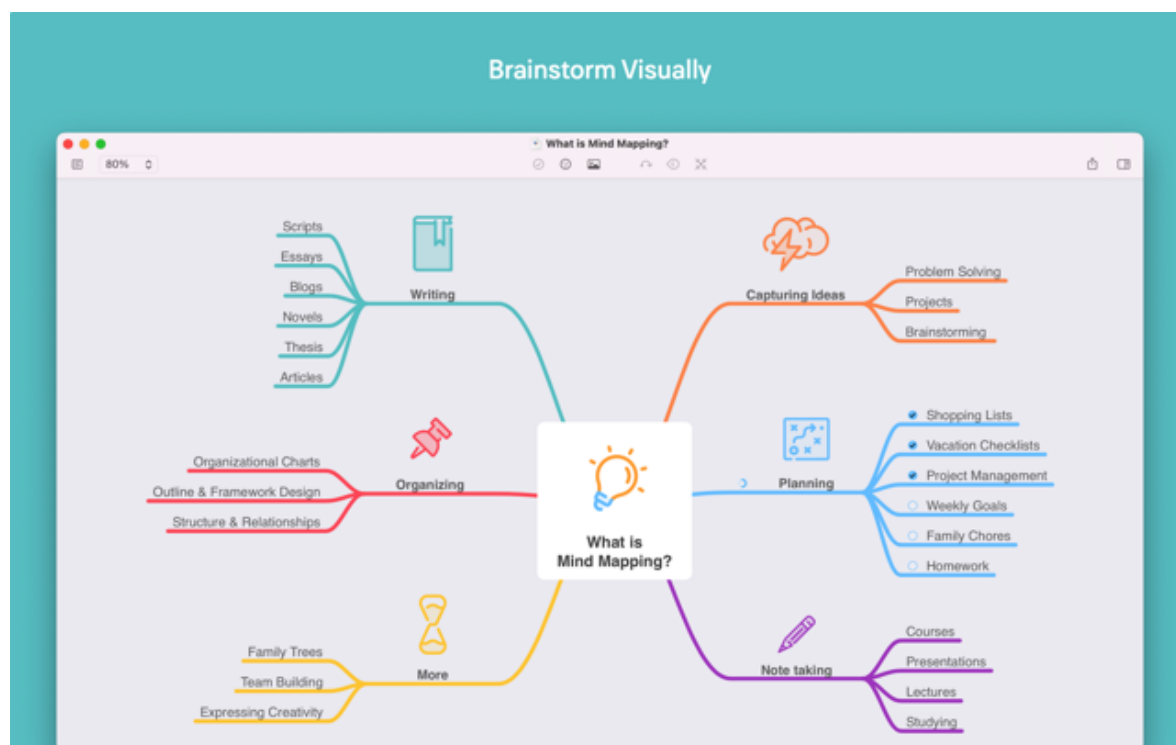


Figure 2.10: MindNode

A mindmap is a way to hierarchically represent linked ideas. It enables you to fit way more ideas on a screen than in a linear, text-based, fashion.

A nice one that I recommend for the Apple ecosystem is [MindNode](#) which allows your to export your mindmap as Markdown so you just have to paste it in your Markdown editor.

An Open Source and cross-platform alternative that directly works on Markdown is [Markmap](#).

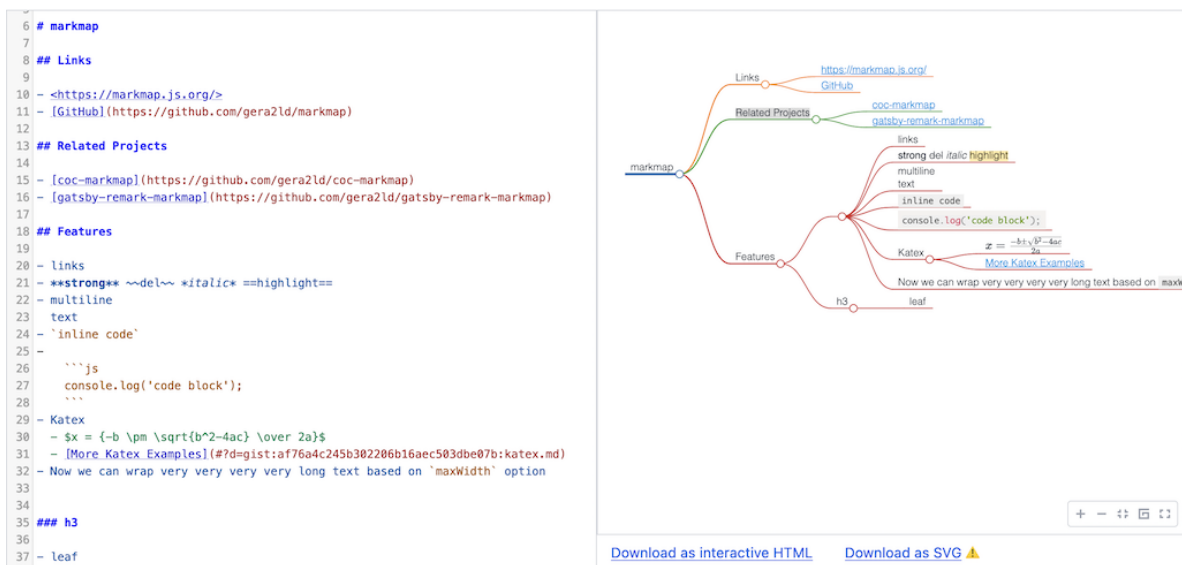


Figure 2.11: Markmap

You write Markdown, and it generates a mindmap.

### 2.3.2 Why? How? What?

Sharing knowledge effectively is not intuitive but can be achieved easily by using the following plan.

Start with **Why**. Here you need to describe the problem that you are going to solve. Your reader need to understand why this thing that looks innocuous at a first glance is actually a problem.

Then, provide the **theoretical background**, the **what**.

Then comes the **how**, the **concrete implementation** where you detail your solution, and explain why you prefer this solution over the others.

Finally, provide a **summary**. The best way to do that is by using a short list of bullet points with the important ideas at the end of the chapter.

For example, you may want to write a book about a specific software framework. There is value in explaining the theory behind the choices made by the creators.

Always remember that a technical book is fundamentally different than a free blog post. Most of the time, people writing blog posts do it because they have something to sell whether it be an idea or a product, on the other hand, with a book, the reader has already bought the product, so you should stay focused on **providing value**.

## 2.4 Avoiding the writer's block

Writing long-form content can be paralyzing, especially when you are not used to it!

Nobody can avoid this phenomenon called the **writer's block**.

I know only one method that works to avoid it: I write as if I was talking with a friend. I can write stupid or wrong things, no problems. It's text. I can edit it, format it, rework it or delete it later!

Sometimes, I even talk to myself as if I were talking with my friend in front of me, and write down what I'm saying.

It's as simple as that. You build a castle by placing one stone after the other, not by trying to build everything at the same time.

One thing that you should **NOT** do is to use substances to find inspiration. Writing while being drunk or having smoked marijuana can be easier, but it's a very bad habit that can **destroy your life. DON'T DO THAT**, because you will always need more and enter an infernal spiral. Do you need to be intoxicated to speak with your heart to your friends. No! So you don't need to be intoxicated to write.

### 2.4.1 Start with the big ideas

Another trick to avoid the writer's block is to start with the big ideas, write them down even if they are not correct sentences, and then you'll be able to fill the holes more easily.

## 2.5 Storytelling

Almost every writing resource will keep trumpeting about how telling a story is a good way to hook your readers and to make your book not boring.

While I particularly enjoy good stories for fictions and humanities, but I think that for technical reads, it does more harm than good.

Technical topics are already hard enough to understand and apprehend. Storytelling adds indirection and obfuscates what really matters.

This is why I think that storytelling is a good fit for the introduction of the book and the conclusion, but not to transmit deep technical knowledge.

A better way is to simply illustrate your writings with anecdotes, real or fictional.

## 2.6 Editing your manuscript

Once you are pretty satisfied with what you have written, it's time to review and edit it.

The golden rule of editing is to never review your writings the same day you wrote them. You need a fresh mind to

### 2.6.1 Ask Why? Why? Why? Why?

Why this sentence is here? Why this word and not another one? Why can't I remove this section?

### 2.6.2 Trim the fat

When reviewing your book, you should read it with the intent of removing 20% of it.

Why? Because most of your readers certainly already know a lot of things, so respect your readers' time and go straight to the point.

Your readers want to get the job done, and your role is to help them to do it fast.

Start again from your 3-level plan, and verify that when writing your book you did not obfuscate the plan with too many words. It's often the problem of self-improvement books: they want to fill as many pages as possible and then it can become very hard to understand what are the really important ideas.

### 2.6.3 Tools

As a non-native English writer, the only tool I use for editing is a Grammar and style checker.

The most famous one is [Grammarly](#).

If you prefer to support Open Source tools, there is [LanguageTool](#) which is very similar and is even cheaper but, in my experience, a little bit less effective.

Please note that I never use their invasive browser extension which may be a huge problem for your privacy. Instead, I only use the web interface.

### 2.6.4 Getting feedback

The easiest, and in my opinion, most effective way to get feedback about your book is to publish parts of it as blog posts.

Then you share these posts on social websites such as [Reddit](#), [HackerNews](#) or [Twitter](#).

As we will see in the Marketing chapter, it's also a great way to make people want to buy it.

## 2.7 Summary

- The value of a technical book lies in how many hours you can save to your readers
- During the editing phase, trim the fat
- Use [Grammarly](#) or [LanguageTool](#) to fix your spelling, grammar and wording.
- Get feedback by publishing parts of your book as blog posts.

```
+++ title = "Technical Writer - Publishing" date = 2021-06-01T6:00:00Z type = "page" url  
= "/technical-writer/publishing" access = "paid_members"
```

```
+++
```

# Chapter 3

## Publishing

There are quite a few publishers specializing into technical topics, the most famous being:

- <https://www.oreilly.com>
- <https://packtpub.com>
- <https://nostarch.com>
- <https://www.manning.com>

### 3.1 Why you may want to work with a publisher

Publishers already have huge audiences (their customers), and a working supply chain.

They will save you a lot of marketing efforts.

You can see the terms of each publisher here: \* <https://www.oreilly.com/work-with-us.html> \* <https://authors.packtpub.com/> \* <https://nostarch.com/writeforus> \* <https://www.manning.com/write-for-us>

### 3.2 Why you actually want to self-publish your book

#### 3.2.1 Opacity

~6 months after I publicly announced the completion of my book, a famous tech publishing company contacted me to see if I was interested in working with them.

We exchanged a few emails, but the experience left me a very bad taste in my mouth: every time I asked specific questions, they were evasive and couldn't answer them.

The only thing that was clear from the exchange was that very few of their authors made enough money to live from their writings and thus had

So, you want me to work with you, but at the end I will be the only one who is not able to put food on the table thanks to the book, all while providing the expertise?

Hahah, no, thank you!

### **3.2.2 It takes way more time**

By working with an editor, you'll have to work with their editors and managers, which requires a lot of effort and time.

### **3.2.3 They steal your customers**

When sharing your work, you are building trust with your readers.

When you self-publish a book, you are building an audience but, more importantly, a customer base.

Everybody in sales knows that it's easier to take care of your customer than to acquire new ones. Thus, if people have already bought your product, there are more chances that they will buy another one again.

But, when you sell your work through a publisher, customers are not yours. The publisher has the email addresses of your readers, not you. So, if you decide to write another book, you will need to start your marketing efforts from zero, again.

### **3.2.4 Drawbacks**

On the other hand, self-publishing your book has a few drawbacks:

You need to do no everything by yourself (you can still hire contractors if you don't want to).

Marketing and building an audience takes a lot of time when starting from zero.

It's harder to review your own work, so there will certainly be a few mistakes in the first versions of your book.

## **3.3 Which price?**

If you are a nobody (like I was when I published my first book), then the price should be between \$30 and \$50.

The unintuitive pricing strategy for Early-Access:

You certainly think that your early supporters should get a better price. But this is false.

Your early access customers are the one the most interested buy your book. The ones that are ready to pay the most.

Once you have saturated your niche, it remains only the curious. The one for who you are not solving a painful problem, but reading your book can be nice, not more.

That's why it makes sense to actually sell your book at a higher price during early access and offer some bonuses in return. When I say a higher price, it's like \$10 more.

For example, my book was sold \$48 during the early-access period with the bonuses and is now \$38 without the bonuses.

### 3.3.1 The coupon strategy

Another strategy that works well is to put your book at a very high price, \$60-\$80, for example, and to regularly announce limited-time offers.

Here, you will make most of your sales during the discount period, playing with the Fear Of Missing Out of your customers.

But, this strategy only works if you have an audience that you can regularly contact to push your discounts.

## 3.4 From Markdown to ebooks

We are going to see 2 ways to turn your manuscript into ebooks for your readers.

The first one is [honkit](#), which is the “quick and dirty” way to get the job done, and the second one is [pandoc](#) which requires a little bit more configuration, but is way more powerful and will allow you to produce good-looking ebooks.

But first, you need to create a cover for your book.

### 3.4.1 Creating a cover

The easiest way to create a cover is to use [Figma](#) using a 700 (width) x 930 (height) frame.

You can then export it to PNG and PDF: `cover.png` and `cover.pdf`, two files that we will use later to generate the ebooks.

### 3.4.2 Using HonKit

`honkit` is a fork of [GitBook](#) when it was Open Source.

```
$ npm i -g honkit
$ mkdir mybook
```



```
$ honkit init mybook
$ honkit build
```

You can read more about it on [their website](#).

I personally don't use it because I find the end-result ugly. But if you need a simple promotional ebook or a quick PDF, it can get the job done quickly.

### 3.4.3 Using Pandoc

Instead, I use [Pandoc](#). Pandoc wraps a complex toolchain to turns many text formats (Mark-down, AsciiDoc...) to documents (PDF, EPUB, DOCX...). It uses [LaTeX](#) under the hood, but this is of no interest to us.

#### 3.4.3.1 The pandoc configuration I use

Pandoc is not easy to configure, this is why, to save you countless hours of your precious time, I give you the configuration files and templates that I use to generate my ebooks.

Given the following files tree:

```
.
├── Dockerfile
├── ebooks
├── Makefile
├── README.md
├── src
│   ├── 01_first_chapter.md
│   ├── 02_second_chapter.md
│   ├── assets
│   │   ├── cover.pdf
│   │   └── cover.png
│   ├── config
│   │   ├── epub.css
│   │   ├── inline_code.tex
│   │   ├── settings.txt
│   │   └── tango_mod.theme
```

Created with the following commands:

```
$ mkdir my_book
$ cd my_book
$ mkdir -p ebooks src/config src/assets
```

```
$ cat << EOF > src/01_first_chapter.md

# Hello
This is the first chapter.
EOF

$ cat << EOF > src/02_second_chapter.md

# World

This is the second chapter.
EOF
```

And the following configuration files:

## Dockerfile

```
FROM ubuntu:latest

RUN apt update
RUN apt upgrade -y

# Create ebook user
ENV USER=ebook
ENV UID=10001

RUN adduser \
    --disabled-password \
    --gecos "" \
    --home "/nonexistent" \
    --shell "/sbin/nologin" \
    --no-create-home \
    --uid "${UID}" \
    "${USER}"

# Install dependencies
ENV DEBIAN_FRONTEND noninteractive
RUN apt install -y vim calibre pdftk epubcheck binutils make wget imagemagick

RUN apt install -y pandoc libpar-packer-perl perl-doc zlib1g zlib1g-dev expat
```

```

    texlive-latex-base texlive-latex-extra texlive-xetex texlive librsvg2-bin
↪ \
    texlive-fonts-recommended texlive-fonts-extra texlive-xetex
↪ texlive-latex-recommended

USER ebook:ebook

WORKDIR /ebook

CMD ["make", "all"]

```

## Makefile

```

BOOK_FILE = output_book_file
BOOK_TITLE = TITLE OF YOUR BOOK
CONFIG_DIR = src/config
CHAPTERS_FILES = $(CONFIG_DIR)/settings.txt src/*.md

COVER = src/assets/cover.png
DATE := $(shell date "+%B %e, %Y")
VERSION = v2021.41
DIST_DIR = ebooks
DOCKER_IMAGE = localhost/writing/ebook

.PHONY: all
all: pdf epub mobi epubcheck

.PHONY: pdf
pdf:
    pandoc $(CHAPTERS_FILES) \
        --resource-path=src \
        --output=$(DIST_DIR)/$(BOOK_FILE)_content.pdf \
        --pdf-engine=xelatex \
        --table-of-contents --toc-depth=2 \

```

```

--number-sections \
--top-level-division=chapter \
--include-in-header config/inline_code.tex \
-V fontsize=12pt \
-V documentclass=report \
-V linkcolor:blue \
--highlight-style $(CONFIG_DIR)/tango_mod.theme \
-M date="$(VERSION)"
pdftk src/assets/cover.pdf $(DIST_DIR)/$(BOOK_FILE)_content.pdf cat output
↪ $(DIST_DIR)/$(BOOK_FILE).pdf
rm $(DIST_DIR)/$(BOOK_FILE)_content.pdf

```

**.PHONY:** epub

epub:

```

pandoc $(CHAPTERS_FILES) \
--resource-path=src \
--output=$(DIST_DIR)/$(BOOK_FILE).epub \
--table-of-contents --toc-depth=2 \
--top-level-division=chapter \
--number-sections \
--listings \
--standalone \
--epub-cover-image=$(COVER) \
--metadata title="$(BOOK_TITLE)" \
--highlight-style $(CONFIG_DIR)/tango_mod.theme \
--css $(CONFIG_DIR)/epub.css \
-M date="$(DATE)"

```

**.PHONY:** epubcheck

epubcheck:

```

java -jar /usr/bin/epubcheck ebooks/$(BOOK_FILE).epub

```

**.PHONY:** mobi

mobi:

```

ebook-convert $(DIST_DIR)/$(BOOK_FILE).epub $(DIST_DIR)/$(BOOK_FILE).mobi
↪ --cover $(COVER)

```

```
.PHONY: docker
docker:
    docker build -t $(DOCKER_IMAGE):latest .
```

src/config/settings.txt

```
---
title: TITLE OF YOUR BOOK
subtitle: SUBTITLE OF YOUR BOOK
author: YOUR NAME
subject: "markdown"
keywords: [books,programming,technology]
language: en-US
cover-image: src/assets/cover.png
lof: false
lof-own-page: true
toc-own-page: true
titlepage: false
colorlinks: true
geometry: "left=3cm, top=2cm, right=3cm, bottom=2cm"
linestretch: 1.25
---
```

src/config/tango\_mod.theme

```
{
    "text-color": null,
    "background-color": "#f8f8f8",
    "line-number-color": "#aaaaaa",
    "line-number-background-color": null,
    "text-styles": {
        "Other": {
            "text-color": "#8f5902",
            "background-color": null,
            "bold": false,
            "italic": false,
            "underline": false
        },
    },
}
```

```

"Attribute": {
  "text-color": "#c4a000",
  "background-color": null,
  "bold": false,
  "italic": false,
  "underline": false
},
"SpecialString": {
  "text-color": "#4e9a06",
  "background-color": null,
  "bold": false,
  "italic": false,
  "underline": false
},
"Annotation": {
  "text-color": "#8f5902",
  "background-color": null,
  "bold": false,
  "italic": false,
  "underline": false
},
"Function": {
  "text-color": "#000000",
  "background-color": null,
  "bold": false,
  "italic": false,
  "underline": false
},
"String": {
  "text-color": "#4e9a06",
  "background-color": null,
  "bold": false,
  "italic": false,
  "underline": false
},
"ControlFlow": {
  "text-color": "#204a87",
  "background-color": null,
  "bold": false,

```

```

        "italic": false,
        "underline": false
    },
    "Operator": {
        "text-color": "#ce5c00",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Error": {
        "text-color": "#a40000",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "BaseN": {
        "text-color": "#0000cf",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Alert": {
        "text-color": "#ef2929",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Variable": {
        "text-color": "#000000",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Extension": {

```

```

        "text-color": null,
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Preprocessor": {
        "text-color": "#8f5902",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Information": {
        "text-color": "#8f5902",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "VerbatimString": {
        "text-color": "#4e9a06",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Warning": {
        "text-color": "#8f5902",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Documentation": {
        "text-color": "#8f5902",
        "background-color": null,
        "bold": false,
        "italic": false,

```



```

        "underline": false
    },
    "Import": {
        "text-color": null,
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Char": {
        "text-color": "#4e9a06",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "DataType": {
        "text-color": "#204a87",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Float": {
        "text-color": "#0000cf",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Comment": {
        "text-color": "#8f5902",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "CommentVar": {
        "text-color": "#8f5902",

```

```

        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Constant": {
        "text-color": "#000000",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "SpecialChar": {
        "text-color": "#000000",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "DecVal": {
        "text-color": "#0000cf",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    },
    "Keyword": {
        "text-color": "#204a87",
        "background-color": null,
        "bold": false,
        "italic": false,
        "underline": false
    }
}
}
}

```

src/config/epub.css

```

/* This defines styles and classes used in the book */
body { margin: 5%; text-align: justify; font-size: medium; }

```



```

\renewcommand{\texttt}[1]{
  \colorbox{Light}{\oldtexttt{#1}}
  \small
}

% \renewenvironment{Shaded} {\begin{snugshade}\footnotesize}
%   {\end{snugshade}}

\usepackage{fvextra}
\DefineVerbatimEnvironment
  ↪ {Highlighting}{Verbatim}{breaklines,breakanywhere,commandchars=\\\{\}}

```

Edit `Makefile` and `src/config/settings.txt` with the name of your book and the correct metadata

```
$ make docker
```

Then put the cover files of your book ( `cover.png` and `cover.pdf` ) in the `src/assets` folder.

then:

```
$ docker run -ti --rm -v `pwd`: /ebook localhost/writing/ebook
```

And TDAAAAAAA! You can find your ebook files in the `ebooks` folder.

## 3.5 Selling your ebooks

Now you have the actual digital product, it's time to sell it.

### 3.5.1 Gumroad

If you have a business that don't need to handle VAT, [Gumroad](#) is certainly the easiest platform to get started to sell digital products.

### 3.5.2 Podia

[Podia](#) links with your [Stripe](#) account to handle payments. They are a good and cheaper alternative to gumroad as you pay a fixed monthly subscription. One nice thing for European businesses is that they can handle VAT.

I really like the experience as a customer as it gives access to a portal where I can re-download

my past purchases.

One downside of Podia is that you can't configure the VAT rate, and thus are limited to the digital service rate, while most countries have a specific VAT rate for (e)books.

### 3.5.3 Paddle

If you are an European business subject to collectin VAT and don't want to have to worry about this sh\*t, [Paddle](#) may be your best option

One downside of Paddle is that they only send a one-time link by email to download the digital product, so if you make updates to your book, you will ned to find a way to distribute your updates to your readers.

### 3.5.4 Stripe links

[Stripe Links](#) are not a good fit to sell your book because you can't easily include a link that will limit the download to customers.

### 3.5.5 Amazon KDP (Kindle Direct Publishing)

if you want to make money from your book, [Amazon KDP](#) is not a good solution. They'll take a 70% cut if your book costs more than \$9.99 or less than \$2.99, and this is before taxes.

Furthermore, like with traditional publishers, customers are not yours. They are Amazon's which means that

The last thing that your should consider is that you need to fill and sign a few taxes forms to sell your book here, which may be hard to understand if you don't have an American accountant.

For all these reasons I've personally decided that I will never publish on Amazon KDP.

### 3.5.6 Summary

Use Gumroad or Podia if you don't need to handle European VAT. Use Paddle or Podia if you need to handle European VAT.

## 3.6 Avoiding piracy

If your book is successful, it will be shared illegally, which means various financial damages for you. You just can't avoid that.

Instead, what you can do is be prepared for that and pro-actively look for pirated versions of your book online.

There are mainly 4 sources of pirated ebooks: - ZLibrary - <https://1filedownload.com> - Torrents - Forums and private chats

While it's really hard to monitor private forums and chats, it's, on the other hand, easy to check once a month The Pirate Bay and ZLibrary to see if someone posted a pirate version of your book.

### 3.6.1 The DMCA notice

One day, I found a pirated version of my book on the famous ebook pirate website: ZLibrary. I was extremely disappointed by my readers, but that's life, and at least, it had a good review on the website .

I sent the following email to their support, and within a few days or weeks, I don't remember exactly. The pirated book was no longer available on their website.

To whom it may concern,

My name is [YOUR NAME] and I am writing to notify you of the copyright infringement and unlawful use of my copyrighted material that appear on the service for which you are the designated agent.

The infringing material, which I contend belongs to me, includes the following:  
My book: [THE NAME OF YOUR BOOK].

The infringing material appears at the following location:

- [URLS]
- [OF THE PIRATED]
- [BOOK]
- And all the other mirrors

The original material is available at: [LANDING PAGE OF YOUR BOOK]

This letter is the official notification under Section 512(c) of the Digital Millennium Copyright Act, and I request that you remove the infringing material from your servers.

I also request that you immediately notify the infringer of this notice and inform them to cease and desist from further infringement.

I am providing this notice in good faith and with the firm belief that the use of the described material is unauthorized.

If you have any questions, please contact me directly: [YOUR EMAIL ADDRESS]

Sincerely,  
[YOUR NAME]

## 3.7 Refunds

Don't be cheap on refunds.

An unsatisfied customer who gets a refund won't make much noise.

On the other hand, an unsatisfied customer who feels that you scammed them may make you lose more money than if you just refunded them by spreading their hate and misinformation on social media.

## 3.8 Summary

- Self-publishing is not only easier and faster, but you will also make more money.
- Use [HonKit](#) (fast and dirty) or [Pandoc](#) (harder but cleaner) to turn your Markdown files into ebooks.
- If you spot a pirate version of your book, send a DMCA notice to the website administrators.

```
+++ title = "Technical Writer - Marketing" date = 2021-06-01T6:00:00Z type = "page" url  
= "/technical-writer/marketing" access = "paid_members"
```

```
+++
```

# Chapter 4

## Marketing

Marketing your book is certainly the hardest part of the journey. On one hand, because everybody is already fighting hard for the attention of “consumers”, and on the other hand because it’s a marathon, not a sprint.

### 4.1 Where to promote your book

The holy rule of indie marketing is that it’s better to reach few people with a deep interest of your topic than many people that are not interested.

#### 4.1.1 Your own blog

There are 2 ways to make “try” your book.

Either you provide a free chapter, or you can publish excerpts on your own blog.

I believe that the latter is the best solution for many reasons.

First, by publishing excerpts, you can get feedback on specific parts of your book.

Second, because it’s easier to promote a link to a blog post than to a PDF.

And finally, because by publishing great content on your blog, you will increase its notoriety and thus its SEO (Search Engine Optimization).

This is why your blog should be the first place where you promote your book. All other channels should funnel people to your blog.

#### 4.1.2 Reddit

[Reddit](#) is certainly the place where you can reach the most people when you have no audience.



People organize themselves into communities around topics, so you just need to target 1 or 2 communities that could be interested in what you have to say.

I have noticed that the key to success on Reddit is to simply talk with your heart. If you try some marketing or sales trick, people will notice, and drama comes very fast on Reddit.

Be aware that most communities don't appreciate spam, thus, you will need to be a good [netizen](#) and answer questions and comments.

### 4.1.3 Twitter

On the other hand, reaching people on Twitter is hard due to their algorithm optimized for viral (dumb) content, and that you will need to pay to “promote” your tweets.

Basically, Twitter's algorithm works as follow: the more you spend time or money on it, the more you are rewarded, which makes me believe that the investment is not worth it when you are beginning. Your time is your most precious asset.

That being said, it's still possible to reach a large audience by finding the good [#hashtags](#) for your topic, which are often followed by bots that will happily retweet your posts.

### 4.1.4 Specialized newsletters

Better than Twitter, are specialized newsletters about your topic. Often run by fans that are always looking for content to include, you can send them your content to publish, and if they find it valuable for their audience, they will happily publish it.

For example, for my book *Black Hat Rust*, I try to publish at least 1 post per week in the [This Week In Rust newsletter](#).

### 4.1.5 GitHub

If your book contains code samples, I **strongly** recommend you to create a GitHub repository.

As I'm writing this, the GitHub repository of my book [Black Hat Rust](#) has ~1700 stars which led to hundred of direct sales (tracked with coupons, so the number of indirect sales is higher) which means thousands of \$\$.

It's also a good idea to inform prospects why they would NOT want to buy your book. There is nothing worse than a disappointed or angry customer, so it's better not to make a sale than having someone angrily tarnishing your work on online forums.

### 4.1.6 Be careful of drama

It is indisputable, **drama sells**.

It generates a lot of visits, which generally means increased revenues, but you don't necessarily want to do that because it can destroy your reputation in the long term.

That being said, here is what I identified that tends to generate the most drama and reactions: Strong opinion on a controversial topic, but with few linked sources.

The idea is to destroy something that many people think it's true, but you think is wrong, and not to include too much sources? Why? Because then many people on the forums will quickly comment that you don't have any proof. But as many people may already be on your side, they will reply with sources and arguments. And it's a controversial topic, it will never ends.

This is how to generate "engagement", the graal of Silicon-Valley's style funded Startups which means that your post will be ranked higher by the algorithms, and also why I hate Twitter so much.

## 4.2 When to promote

Another thing to know is that the time of the day and the day of the week have a huge impact on the effectiveness of your marketing efforts.

There are many articles on Internet about when to publish: \* <https://optinmonster.com/the-best-time-to-send-emails-heres-what-studies-show/> \* <https://www.wordstream.com/blog/ws/2014/09/04/best-time-to-send-email-campaign> \* <https://www.mailjet.com/blog/news/best-time-to-send-email-newsletters/> \* <https://coschedule.com/blog/best-time-to-send-email>

I will save you tons of SEO fluff: the best moment in the day to post something is at 9 a.m. in the timezone where the majority of your audience lives.

The best days to post are Tuesday or Wednesday.

Why?

Because people mostly check the news and social networks at work and when commuting. They have better things to do than reading your stuff during the weekends and after work, like hanging out with friends or playing video games, or taking care of their family.

Also, depending on the channel that people use to follow your content, there might be some lag between the day of publication and the moment when your content is read. Some people use social media, some use RSS readers, others will discover your through weekly newsletters.

So by publishing on Tuesday or Wednesday, you increase your chances of being read by your audience.

## 4.3 Momentums

Momentums are those moments when the wind is in your back, and you can fully open the sail to make a lot of sales.

It's when you get some attention from people on the internet, and you need to capitalize on it to keep it as long as possible and make as many sales as possible.

One example is [GitHub's trending page](#).

When your repository get a certain amount of stars in 1 day, it becomes promoted on the trending page, which brings tons of curious people. Furthermore, when appearing on the trending page of a specific language, [Rust](#) for example, some Twitter bots will tweet your repository, giving it an even bigger reach.

Another example is Twitter bots: If you find the good `#hashtags`, your tweets will be boosted by bots with thousands of people interested on a specific topic.

So when one of these things happen, it's time to roll up your sleeves and surf the wave

## 4.4 Tracking the origin of your sales

There are basically 2 ways to understand where your sales come from.

The first one is to set up an analytics service, such as [Plausible](#) or [Fathom](#) and look at the Referrers. The problem is that a lot of websites strip the Referrer header making the data unreliable.

The second one is to use **coupon codes**: discount codes with the name of the source, `GITHUB`, `TWITTER` for example. The problem with this method is that people can copy an URL with a coupon code and share it on private forums or chats obfuscating where your customers really found about your book.

Because both of these methods have caveats and are not reliable alone, the best way is to look at both and try to identify patterns.

## 4.5 Tips and tricks to boost your sales

Sales of a book are never linear: there are some peaks related to the (temporary) success of your marketing efforts.

Thus, you need to regularly use tricks to boost your sales.

Here are the most effective.

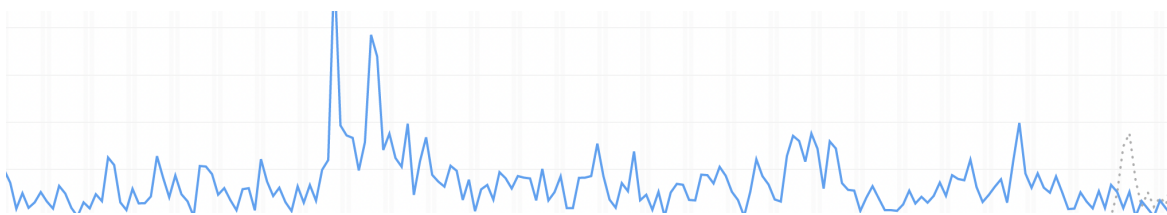


Figure 4.1: Sales over time

### 4.5.1 Early-access bonuses

It may be hard to win people's trust when you are a nobody on internet, and you are asking for money in exchange, that, maybe, some day in the future, you will give them the book that you promised.

A way to break the ice is to offer some early-access bonuses related to the content of the book.

The trick is to offer the bonuses only to the early-access supporters, so they buy the book **now** and don't wait for the book to be completed.

It can be anything: some artwork, a call with you,

For example, for my book, I offered all early-access supporters a curated list of advanced malware analyses. The analyses were not from me, but it would still require a few hours of work and some advanced knowledge to make a list with such a high ratio of information.

### 4.5.2 Flash sales

Flash sales create two effects in the mind of your prospects.

First, they feel having a good deal, and everyone is happy when have good value for our money.

The second, and most important effect is the sense of urgency. People always feel they can buy your book later, until they never buy it. With flash sales, you give them a good reason to do it **now**.

A way to make your flash sales more effective is to justify them.

For example, with the invasion of Ukraine, I expected cyberattacks to rise significantly, so I made a Cyberwar sale to help people prepare.

## 4.6 Summary

- Have your own blog and funnel people to it to promote your book.
- Use momentums to your advantage.

- It's better to reach few people with a deep interest of your topic than many people not interested.

+++ title = "Technical Writer - Conclusion" date = 2021-06-01T6:00:00Z type = "page" url = "/technical-writer/conclusion" access = "paid\_members"

+++

By now, I hope that you feel confident enough to write your first technical book, so it's **YOUR** turn.

You may be tempted to read more resources on how to write a book to be sure to do it right.

I recommend you **NOT TO DO THAT**. Instead, start writing your book **now**, make mistakes, notice where you have difficulties, and only then start reading specific resources to improve your weaknesses.

Ultimately, remember that done is better than perfect, and even if your first book is not impeccable, you will be able to learn from your mistakes and either edit it, or write a new one.

Goooooooo!

## 4.7 Contact

Are you annoyed by something? Please tell me, and I will do my best to improve it!

Or, did you enjoy the read and want to say thank you?

You can contact me by email: Javascript required

I'm not active on social networks because they are too noisy and time-sucking, by design.