

Readteractive

Carleslc

Abstract

Tool for writing and generating interactive books, also known as gamebooks.

CHOOSE YOUR OWN ADVENTURE TIME • 14

YOU'RE THE STAR OF THE STORY!
CHOOSE FROM 30 POSSIBLE ENDINGS.

IRRITATED BY THE ICE KING

BY PENDLETON WARD



ILLUSTRATED BY STEPHEN ANDRADE

What is a gamebook?

A gamebook is a work of printed fiction that allows the reader to participate in the story by making choices. Gamebooks have been influenced by *Choose Your Own Adventure* Series and tabletop role-playing games.

Production of new gamebooks in the West decreased dramatically during the nineties as choice based stories have moved away from print based media, although the format may be getting a new lease of life on mobile and ebook platforms. Such digital gamebooks are considered interactive fiction.

This tutorial is built with Readteractive, so of course this is one interactive book.

Ready to join the adventure of using Readteractive?

Click the following option to continue!

(Getting Started)

Getting Started

Clone this repository with git or download the current version as zip.

Install dependencies

Note for **Windows**: It is easier to install this dependencies and run Readteractive with a linux-like shell like Cygwin.

- **Python 3**: Needed to process structure and build Markdown files used to generate your book.
- **Make**: Needed to bundle commands and generate your books in any format.
 - Linux: `sudo apt-get install build-essential`
 - Mac OS:
 - * (Xcode Utils) `xcode-select --install`
 - * Or, using Homebrew: `brew install make`
 - Windows: Install *make* from Cygwin installer.
- *(HTML, PDF, EPUB)* **Pandoc**: Needed to generate HTML, PDF and EPUB from Markdown files.
- *(MOBI)* **KindleGen**: Needed to generate MOBI for Kindle from EPUB file.
- *(Optional)* Kindle Previewer: Needed to preview how your MOBI files looks in tablet and Kindle devices.
- *(Optional)* **librsvg**: Convert SVG images for being used inside PDF
 - Linux: `apt-get install librsvg2-bin`
 - Mac OS, using Homebrew: `brew install librsvg`
 - Windows: Install *librsvg2* from Cygwin installer.

Choose an option:

- **(CLI)**
- **(Readteractive Structure)**
- **(How to build your book)**
- **(How to write your book)**
- **(How to style your book)**

Scaffolding

Install CLI and boost your productivity with Readteractive!



A yeoman generator for scaffolding Readteractive books.

We provide a *command line interface* to easily generate your project doing **scaffolding**, so you don't need to remember the syntax of each file and you can just **focus on writing**.

The (**Readteractive structure**) and files can be automatically generated using this CLI for your custom book.

It also provides a tool for **visualization** of your book with current chapters and the links between them, so you can have a general overview of the **narrative branches** of your book.

Features

- Book scaffolding
- Chapter scaffolding
- Book graph visualization

Install dependencies

- Yeoman: Needed to install readteractive-generator
 - Npm: Needed to install Yeoman
 - * Node: Provides Npm

How to use

- Install Readteractive Generator: `npm install -g readteractive-generator`
- Run: `yo readteractive`

Commands

- `yo readteractive` shows a wizard for generate a new book.
- `yo readteractive:chapter` shows a wizard for generate a new chapter.
- `yo readteractive:graph` shows a graph of your book with current chapters and the links between them.

Choose an option:

- (**Readteractive Structure**)
- (**Return back**)

Readteractive Structure

Let's have a look to how folders and files are organized.

Top folder

In the top folder there are the required files for building your books and configure dependencies. *Please, do not edit or move this files to another folder or build will fail.*

```
.  
  .git  
  .gitignore  
  book.py  
  chapter.py  
  get_property.py  
  process_book.py  
  pandoc-html.css  
  makefile  
  LICENSE  
  README.md
```

(Readteractive Structure: Books and chapters folders)

Books and chapters folders

Every book and chapter you generate needs to match the following structure:

```
.
├── book-example/
│   ├── _meta.yml
│   ├── first_chapter/
│   │   ├── first_chapter.md
│   │   └── first_chapter.yml
│   ├── second_chapter/
│   │   ├── image.png
│   │   ├── second_chapter.md
│   │   └── second_chapter.yml
│   ├── cover.png
│   └── stylesheet.css
```

This structure and files can be automatically generated using our **(CLI)** for your custom book.

Syntax of `.yml` files is based on **YAML**.

Syntax of `.md` files is based on **(Markdown)**.

Syntax of `stylesheet.css` file is based on **(CSS)**, but this is optional as explained later.

(Readteractive Structure: Book metadata)

Book metadata

`_meta.yml`

```
title: Example _gamebook_  
author:  
- First Author  
- Second Author  
start: first_chapter  
language: es  
cover-image: cover.png  
stylesheet: stylesheet.css  
description: |
```

```
  This is the description of your book.  
  Hope you enjoy writting with Readteractive.
```

- (Required) **Title**: The title of your book. You can use (**Markdown**) here using *****bold***** and **_italic_**.
- (Optional) **Author**: The author of your book, or a list of authors. You can use (**Markdown**) here using *****bold***** and **_italic_**.
- (Optional) **Starting chapter**: The chapter id (folder) of the first chapter in the book. Defaults to the first alphanumerically chapter.
- (Optional) **Language**: ISO 639-1 Language Code. Defaults to **en** (English).
- (Optional) **Cover Image**: File of the main image of your book.
- (Optional) **Stylesheet**: The (**CSS**) stylesheet of your book.
- (Optional) **Description**: The description of your book.

(Readteractive Structure: Chapter)

Chapter Structure

`chapter-id.yml`

`title: Example _chapter_`

You can use (Markdown) here using **bold** and *italic*.

`chapter-id.md`

The text of your chapter. You can use (Markdown) here.

- (What about chapters order?)
- (How to build your book)

Chapters Order

It is true that a gamebook does not have a strict order for chapters because users can jump from one to another according to the options they decide to follow, but sometimes is more elegant to have chapters sorted by narrative branch or other order you decide.

Once book is built, the first chapter shown will be the chapter specified in `_meta.yml` (if provided). The following chapters are shown in alphabetical order, but you can define your own order giving a numerical prefix to chapter identifiers like in this example:

```
.
  book-example/
    _meta.yml
    0_before/
      0_before.md
      0_before.yml
    1_after/
      image.png
      1_after.md
      1_after.yml
```

In this example, assuming that no `start` is set inside `_meta.yml`, the first chapter to show up will be `0_before`, followed by `1_after`.

- **(How to build your book)**
- **(Return back)**

How to build your book

If you have all dependencies installed is as easy as executing one command:

```
make BOOK=book-example
```

This will ensure your book and chapters structure is right, check for broken links between chapters and then pack and build your book in different formats:

- HTML
- PDF
- EPUB
- MOBI

The PDF version use LaTeX to get a high-quality typography.

Each chapter will generate a header in the table of contents of the PDF, EPUB and MOBI metadata, and each option will have a clickable link that jumps to the next chapter page.

The HTML version is more dynamic. It only shows the current chapter you have followed with a click on a link so story proceeds more interactively over the book. It also works offline (is self-contained).

If you want to keep available all visited chapters doing scroll you can set the variable `SCROLL`:

```
make html BOOK=book-example SCROLL=yes
```

The MOBI version is useful for Kindle devices.

Furthermore, you can build only the desired format:

```
make html BOOK=book-example
make pdf BOOK=book-example
make epub BOOK=book-example
make mobi BOOK=book-example
```

If you are going to print your book then you should be asking about chapter and page references, because links does not work in paper. You are right. If you do so you will need to have chapter order defined as explained in (**chapters order**). Then, you can build your book using the variable `PRINTED`:

```
make BOOK=book-example PRINTED=yes
```

This will add the chapter order on each section and all link references will have that number after the option text. Then you can go to the last page where the Table of Contents is located and follow the page of the referenced chapter.

Built files are saved in your book folder:

```
.
├── book-example/
│   ├── book-example.html
│   ├── book-example.pdf
│   ├── book-example.epub
│   └── book-example.mobi
```

- (**How to write your book**)
- (**Return back**)

How to write your book

A gamebook has many chapters with links between them. We refer to a chapter as a step with description in one of your narrative branches.

Do you have previous experience writing *gamebooks*?

- (Yes)
- (No)

Some tips for new interactive writers

If you have no previous experience writing gamebooks these are some *tips*:

1. Plan your story. You can make a mind map with each of your narrative branches and which options follow which branch.
2. Define each chapter and write them with Readteractive.
3. Visualize your chapter graph with our (**CLI**) whenever you need to have an overview of your book and ensure each chapter have the links you want.
4. (**Build**) your book from time to time to have a look of how it is looking.

(**How to write options**)

How to write options

Each chapter can link to a many different chapters using the following syntax:

A diagram showing the option syntax: (Text -> [next]). The text is enclosed in parentheses, followed by an arrow pointing to a bracketed section labeled 'next'.

Figure 1: Option syntax

Replace **Text** with your custom option text. Replace **next** with the chapter id which this link is pointing to.

Readteractive will generate a link to the chapter on your book for every reference following this syntax.

You can use (**Markdown**) in **Text**.

You can skip the chapter prefix defined for (**custom order**) at the moment of writing an option in **next**:

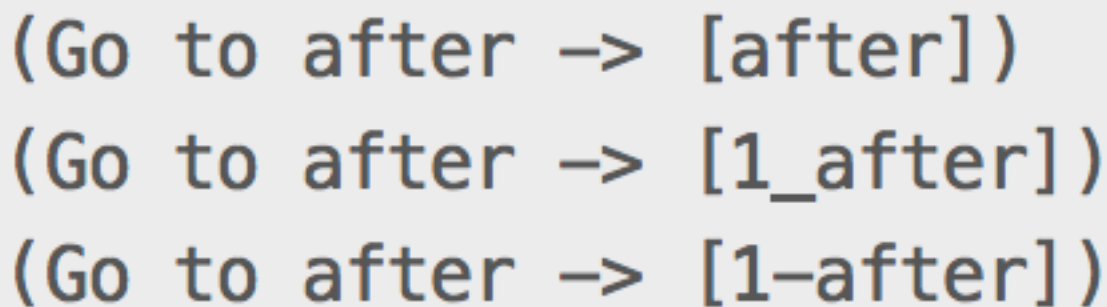
A diagram showing three examples of prefix skipping syntax: (Go to after -> [after]), (Go to after -> [1_after]), and (Go to after -> [1-after]). Each example is enclosed in parentheses, followed by an arrow pointing to a bracketed section.

Figure 2: Prefix skipping

These options are equivalent.

This prefix skipping only works for digits followed by - or _.

- (**How to style your book**)
- (**Return back**)

How to style your book

Choose an option:

- (Boost your writing with Markdown)
- (The art of a stylesheet)
- (What about PDF style?)
- (I like numbers and formulas)

(I am no longer interested in styling)

Markdown and Pandoc

You can use Markdown syntax for styling your chapters. This README file is written entirely in Markdown.

Markdown is simpler than HTML, but if you need to use it, Markdown supports HTML tags.

The example chapter `start.md` provides very useful examples using Markdown.

In order to improve your writing even more you can use a Markdown highlight editor. There are many Markdown editors, some are online like [this](#) or [this](#), and other are desktop applications like Typeora. Code editors like Atom or Sublime Text have markdown plugins too (Atom package, Sublime Text 3 package).

When you build your book, Readteractive will make HTML, PDF and EPUB versions using Pandoc. Pandoc allows you to do a few more things besides. You can read more about that in the Pandoc Manual.

(I want to know more about styles)

CSS

If you know CSS you can customize the style of the book even more than with markdown and HTML.

For custom CSS of the HTML version edit the file `pandoc-html.css` at the top folder.

With EPUB and MOBI (e-book versions) you can optionally specify a `stylesheet.css` for your e-book in the `_meta.yml` file to have custom styles such font family or size.

If no stylesheet is provided then this one is used by default.

This option is not available for PDF because LaTeX is used

(I want to know more about styles)

PDF

Although CSS is not available for PDF version because its style is set by LaTeX you can change margins and page breaks.

You can edit margin size in centimeters (`cm`) or inches (`in`) for PDF files setting the variable `PDF_MARGIN`:

```
make pdf BOOK=book-example PDF_MARGIN=1cm
```

You can define an explicit page break using `\newpage` and explicit new line using `\newline` inside your chapter `.md` file. Unfortunately at the moment this only works for PDF version.

(I want to know more about styles)

MathML

$$E = mc^2$$

Math equations are rendered using MathML, supported for HTML and PDF but only for some EPUB3 readers and currently gives unrecognised tags on *KindleGen* converting to MOBI.

(I want to know more about styles)

Contents

What is a gamebook?	3
Getting Started	4
Install dependencies	4
Scaffolding	5
Features	5
Install dependencies	5
How to use	5
Readeractive Structure	6
Top folder	6
Books and chapters folders	7
Book metadata	8
Chapter Structure	9
Chapters Order	10
How to build your book	11
How to write your book	12
Some tips for new interactive writers	13
How to write options	14
How to style your book	15
Markdown and Pandoc	16
CSS	17
PDF	18
MathML	19