

**CURSOR** アプリ開発すみからすみまで完全ガイドブック

reodesuxz

# cursor アプリ開発すみからすみまで完全ガイドブック

reodesuxz

本書はこちらで販売中です [https://leanpub.com/cursor\\_app\\_guidebook](https://leanpub.com/cursor_app_guidebook)

この版は 2026-05-02 に発行されました。



本書は **Leanpub** の電子書籍です。Leanpub はリーンパブリッシングプロセスで著者や出版社を支援します。**リーンパブリッシング** は新しい出版スタイルです。軽量なツールを使って執筆中の電子書籍を出版し、読者のフィードバックをもらいながら魅力的な本に仕上がるまでピボットを繰り返すことができます。

© 2026 reodesuxz

# Contents

cursor アプリ開発の基礎 . . . . .	1
<b>Writing in Markua . . . . .</b>	<b>2</b>
Section One . . . . .	2
Including a Chapter in the Sample Book . . . . .	2
Links . . . . .	2
Images . . . . .	2
Lists . . . . .	7
Page Breaks . . . . .	8
Code Samples . . . . .	9
Tables . . . . .	10
Math . . . . .	11
Headings . . . . .	11
Block quotes, Asides and Blurbs . . . . .	13
Good luck, have fun! . . . . .	14

## **cursor アプリ開発の基礎**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます  
[https://leanpub.com/cursor\\_app\\_guidebook](https://leanpub.com/cursor_app_guidebook).

## **cursor アプリとは**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます  
[https://leanpub.com/cursor\\_app\\_guidebook](https://leanpub.com/cursor_app_guidebook).

## **cursor アプリ開発のメリット**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます  
[https://leanpub.com/cursor\\_app\\_guidebook](https://leanpub.com/cursor_app_guidebook).

## **cursor アプリ開発の基本概念**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます  
[https://leanpub.com/cursor\\_app\\_guidebook](https://leanpub.com/cursor_app_guidebook).

## **cursor アプリ開発の流れ**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます  
[https://leanpub.com/cursor\\_app\\_guidebook](https://leanpub.com/cursor_app_guidebook).

## **cursor アプリ開発のためのツール**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます  
[https://leanpub.com/cursor\\_app\\_guidebook](https://leanpub.com/cursor_app_guidebook).

## **cursor アプリ開発のためのライブラリ**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます  
[https://leanpub.com/cursor\\_app\\_guidebook](https://leanpub.com/cursor_app_guidebook).

## **cursor アプリ開発の Tips**

このコンテンツはサンプル本では読めません。この本は Leanpub で購入できます  
[https://leanpub.com/cursor\\_app\\_guidebook](https://leanpub.com/cursor_app_guidebook).

# Writing in Markua

Writing in Markua is easy! You can learn most of what you need to know with just a few examples.

To make *italic text* you surround it with single asterisks. To make **bold text** you surround it with double asterisks.

## Section One

You can start new sections by starting a line with two # signs and a space, and then typing your section title.

### Sub-Section One

You can start new sub-sections by starting a line with three # signs and a space, and then typing your sub-section title.

## Including a Chapter in the Sample Book

At the top of this file, you will also see a line at the top:

```
1 {sample: true}
```

Leanpub has the ability to make a sample book, which interested readers can download or read online. If you add this line above a chapter heading, then when you publish your book, this chapter will be included in a separate sample book for these interested readers.

## Links

You can add web links easily.

Here's a link to the [Leanpub homepage](#).

## Images

You can add an image to your book in a similar way.

First, add the image to the “Resources” folder for your book. You will find the “Resources” folder under the “Manuscript” menu to the left.

If you look in your book’s “Resources” folder right now, you will see that there is an example image there with the file name “palm-trees.jpg”. Here’s how you can add this image to your book:



If you want to add a figure title, you put it in quotes:



☒ 1. Palm Trees

If you want to add descriptive alt text, which is good for accessibility, you put it between the square brackets:



☒ 2. Palm Trees

You can also set the alt text and/or the figure title in an attribute list:



☒ 3. Palm Trees

Finally, if no title is provided, and the `alt-title` document setting is the default of `all`, the alt text will be used as the figure title instead of as alt text.



☒ 4. Palm Trees

You can set the important document settings at Settings > Generation Settings.

## Lists

### Numbered Lists

You make a numbered list like this:

1. kale
2. carrot
3. ginger

### Bulleted Lists

You make a bulleted list like this:

- kale
- carrot
- ginger

## Definition Lists

You can even have definition lists!

**term 1**

definition 1a

definition 1b

**term 2**

definition 2

## Page Breaks

We don't recommend that you manually break pages, since that is brittle and can lead to unexpected formatting if you edit text earlier in your chapter and forget about the manual page breaks. But if you really want to add a page break, you use the `{pagebreak}` directive on a line by itself, with blank lines above it and below it.

## Code Samples

You can add code samples really easily. Code can be in separate files (a “local” resource) or in the manuscript itself (an “inline” resource).

### Local Code Samples

Here’s a local code resource:

#### ☒ 5. Hello World in Ruby

---

```
1 require 'time'
2
3 # This is just some pointless code so you can see the syntax highlighting...
4 def display_info
5   pi = Math::PI.round(10)
6   time_last_year = (Time.now - 365 * 24 * 60 * 60).getlocal("-08:00")
7   formatted_time = time_last_year.strftime("%Y-%m-%d %H:%M:%S")
8   puts "Pi to 10 decimal places: #{pi}"
9   puts "The time 1 year ago in Pacific Time: #{formatted_time}"
10 end
```

---

### Inline Code Samples

Inline code samples can either be spans or figures.

A span looks like `puts "hello world"` this.

A figure looks like this:

```

1 require 'time'
2
3 # This is just some pointless code so you can see the syntax highlighting...
4 def display_info
5   pi = Math::PI.round(10)
6   time_last_year = (Time.now - 365 * 24 * 60 * 60).getlocal("-08:00")
7   formatted_time = time_last_year.strftime("%Y-%m-%d %H:%M:%S")
8   puts "Pi to 10 decimal places: #{pi}"
9   puts "The time 1 year ago in Pacific Time: #{formatted_time}"
10 end

```

You can also add a figure title using the title attribute:

#### ☒ 6. Hello World in Ruby

---

```

1 require 'time'
2
3 # This is just some pointless code so you can see the syntax highlighting...
4 def display_info
5   pi = Math::PI.round(10)
6   time_last_year = (Time.now - 365 * 24 * 60 * 60).getlocal("-08:00")
7   formatted_time = time_last_year.strftime("%Y-%m-%d %H:%M:%S")
8   puts "Pi to 10 decimal places: #{pi}"
9   puts "The time 1 year ago in Pacific Time: #{formatted_time}"
10 end

```

---

## Tables

You can insert tables easily inline, using the GitHub Flavored Markdown (GFM) table syntax:

Header 1	Header 2
Content 1	Content 2
Content 3	Content 4 Can be Different Length

Tables work best for numeric tabular data involving a small number of columns containing small numbers:

Central Bank	Rate
JPY	-0.10%
EUR	0.00%
USD	0.00%
CAD	0.25%

Definition lists are preferred to tables for most use cases, since reading a large table with many columns is terrible on phones and since typing text in a table quickly gets annoying.

## Math

You can easily insert math equations inline using either spans or figures.

Here's one of the kinematic equations  $d = v_i t + \frac{1}{2} a t^2$  inserted as a span inside a sentence.

Here's some math inserted as a figure.

$$\left| \sum_{i=1}^n a_i b_i \right| \leq \left( \sum_{i=1}^n a_i^2 \right)^{1/2} \left( \sum_{i=1}^n b_i^2 \right)^{1/2}$$

☒ 7. Something Involving Sums

## Headings

Markua supports both of Markdown's heading styles.

The preferred style, called atx headers, has the following meaning in Markua:

```

1 {class: part}
2 # Part
3
4 This is a paragraph.
5
6 # Chapter
7
8 This is a paragraph.
9
```

```
10 ## Section
11
12 This is a paragraph.
13
14 ### Sub-section
15
16 This is a paragraph.
17
18 #### Sub-sub-section
19
20 This is a paragraph.
21
22 ##### Sub-sub-sub-section
23
24 This is a paragraph.
25
26 ##### Sub-sub-sub-sub-section
27
28 This is a paragraph.
```

Note the use of three backticks in the above example, to treat the Markua like inline code (instead of actually like headers).

The other style of headers, called Setext headers, has the following headings:

```
1 {class: part}
2 Part
3 =====
4
5 This is a paragraph.
6
7 Chapter
8 =====
9
10 This is a paragraph.
11
12 Section
13 -----
```

14

15 This is a paragraph.

Setext headers look nice, but only if you're only using chapters and sections. If you want to add sub-sections (or lower), you'll be using atx headers for at least some of your headers. My advice is to just use atx headers all the time. (The `{class: part}` attribute list on a chapter header to make a part header does actually work with Setext headers, but it's really ugly.)

Note that while it is confusing and ugly to mix and match using atx and Setext headers for chapters and sections in the same document, you can do it. However, please don't.

## Block quotes, Asides and Blurbs

Block quotes are really easy too.

—Peter Armstrong, *Markua Spec*

Asides are useful for longer text.

But typing them like this isn't fun.

Asides can be written this way, since adding a bunch of `A>` stuff at the beginning of each line can get annoying with longer asides.

Blurbs are useful

Blurbs are useful

There are many types of blurbs, which will be familiar to you if you've ever read a computer programming book.



This is a discussion.

You can also specify them this way:



This is a discussion



This is an error.



This is information.



This is a question. (Not a question in a Markua course; those are done differently!)



This is a tip.



This is a warning.



This is an exercise. (Not an exercise in a Markua course; those are done differently!)

## **Good luck, have fun!**

If you've read this far, you're definitely the right type of person to be here!

Our last piece of advice is simple: once you have a couple chapters completed, publish your book in-progress!

This approach is called Lean Publishing. It's why Leanpub is called Leanpub.

If you want to learn more about Lean Publishing, read [this](#) or watch [this](#).