

---

# txt2epub Documentation

*Release latest*

June 23, 2015



<b>1</b>	<b>Usage</b>	<b>3</b>
<b>2</b>	<b>Dependency</b>	<b>5</b>
<b>3</b>	<b>why another converter?</b>	<b>7</b>
<b>4</b>	<b>the way it works</b>	<b>9</b>
<b>5</b>	<b>some examples</b>	<b>11</b>
5.1	just text files . . . . .	11
5.2	rst files . . . . .	11



txt2epub creates an epub file from a bunch of text files.



---

### Usage

---

- Install: `python setup.py install --user`
- Execute: `txt2epub --keep-line-breaks output.epub input.txt`

For more options, please see `txt2epub -h`.





---

## Dependency

---

- `python`.
- `jinja2`: for rendering output from epub template.



---

### why another converter?

---

the reason for wanting a converter? my favourite newspaper comes in pdf format and it is very well readable on a fast and bright and large and heavy screen like for example an iPad, but it really is close to unreadable on a small thing like a e-ink based eReader.

so I started looking at what one would need to produce a epub file and it resulted that it's not all as obvious or simple as for one to want to do this by hand, but also not that difficult if you have a computer.

I found a dead and not-too-readable google-code program with this name, I thought I would give it a try here.

later on I found some more libraries, but I was far enough in the process that I thought I would finish it anyway!



---

### the way it works

---

- read all files from the input directory
- produce the `mimetype`
  - this is hard coded
- produce the `META-INF/container.xml`
  - it points always at `content/00_content.opf`
- produce the `content/content.opf`
  - metadata: if you pass `-options`, they go here
  - manifest: the files in the directory
  - spine: again the files in the directory
  - guide: empty (I would not know what to put here)
- convert the documents to valid html
- zip everything in the correct order



---

## some examples

---

the script works on all simple input and attempts to do something reasonable with more complex cases. the maintainers of this program will be glad to consider your remarks and include your patches!

and by the way, you very probably always want to use the two options `--creator` and `--title`, as they will make your document easier to identify in most ebook readers.

here we have a look at some examples.

### 5.1 just text files

- put all files in one directory,
- make sure that each chapter of your work is in its own file,
- you do not need any fancy formatting,
- paragraphs are separated by empty lines (you may specify `--keep-line-breaks`),
- you are satisfied with the lexicographic ordering of the chapters,
- execute: `txt2epub output.epub *.txt`
- result: your epub will have an index naming the input files

### 5.2 rst files

same as for text files, with some words of warning:

- not all rst is supported, if the script crashes on your input please open an issue [here](#)!
- files must have extension `.rst`, sorry,
- you can have a single rst file, the index is based on its headings.
- execute: `txt2epub output.epub *.rst`