

# Individualized Creation of Educational Material for Generation Z

Gregor von Laszewski, Josiah Clemons, Kang Jie Gan,  
Naimesh Chaudhari, Geoffrey C. Fox  
Indiana University, Bloomington, IN 47408  
laszewski@gmail.com

**Abstract**—TBD. change me in metadata.yml

## I. INTRODUCTION

:o2: add you names in metadata.yml

:o2: add abstract in metadata.yml

:o2: improve the writup

This service provides a cloud-based online bookmanager for individualized learning material for big data and cloud computing [1].

## II. SUMMARY

We have a number of distributed documentation written by different groups and organizations and users. Although we can put an html link up with the collection students, researchers, tutorial participants can benefit from a tool that integrates all of them into a single Document and not just a collection of links. This allows for a “book” like distribution of the content written by various people and hosted on various and GitHub repositories.

In our case it also serves as a mechanism for generating class proceedings of reports developed by students so they can take them “home” after they did the class.

### A. The solution

We have a simple tool called bokmanager that creates an epub of documents specified in a yaml table of contents. The documents will be fetched from the url, the images downloaded and an epub generated. This epub (if not too big) can then be browsed not only on your computer, but also on your iPads and cell phones, so you can add them to your book collection

### B. Sample yaml file

- <https://github.com/cyberaide/bookmanager/blob/master/tests/python.yml>

### C. Summary of Benefit

- Simple to use
- leverages pandoc, so more formats in future will be supported
- pulls together information from several sources
- auto generated title page
- epub

## III. IDEA

We have a tool called cyberaide bookmanager [2][3] that can create books based on markdown files that are distributed online in github (not only for data science and cloud computing, but that is where we have most material).

## IV. DESIGN ASPECTS

1. We like to host the creation of such a book in a container and place it online on a cloud or web service. (containerization is easy and mostly done for the command line, but not for the html/gui portion)
2. We like to allow people that visit a web page, select class modules that are automatically integrated in customized books for that student
3. The integration could be as simple as clicking on some chapters in a table of content that we distribute
4. The interface could be enabled with javascript where you drag and drop the chapters into a hierarchy with chapter titles that could be added or are derived from the included markdown
5. We would like to facilitate a markdown test that prior to the inclusion tests if the markdown is ok while using markdown lint (mdl).
6. (optional) as extension we like to include different formats, such as PDF, word, rst, txt, .org mode (this is in part easy as we use in the container pandoc that supports these formats).

## V. REQUIREMENTS

Obviously, you need to have some Javascript/HTML/Webpage experience. These days almost all students have them so I think that is doable. I

am open to suggestions for javascript framework, but i heard electron may be a good choice so we can ost the service in the cloud, but also run locally.

Other than a manual and programing the project has a 2 page report requirement in markdown not LaTeX. The reason this is so low is that you spend more time on programming than writing the report. You should think of the report also like a mini manual, so you let the users know why its good and how they can use it.

## VI. BOOKMANAGER

Bookmanager is a tool to create a publication from a number of sources on the internet. It is especially useful to create customized books, lecture notes, or handouts. Content is best integrated in markdown format as it is very fast to produce the output. At present we only produce epubs, but it will be easy to also create pdf, html, work, odt and others. As we use pandoc we can support the formats supported by it.

Implemented Features:

- Table of contents with indentation levels can be specified via yaml
- Special variable substitution of elements defined in the yaml file
- Documents are fetched from github
- The documents will be inspected and the images found in them are fetched (we assume the images are relative to the document, http links will not be modified)
- Automatic generation of a cover page
- Output is generated in a dest directory

Planned enhancements:

- integration of References via pandoc citeref
- integration of Section, Table, Image references via pandoc crossref

If you like to help get in contact with Gregor von Laszewski [laszewski@gmail.com](mailto:laszewski@gmail.com)

```
$ pip install cyberaide-bookmanager
```

### A. Usage

The manual page is listed in Section VII

### B. Cover Page

Book manager can create a simple cover page for you. and example is given at

- <https://github.com/cyberaide/bookmanager/blob/master/tests/example/cover.png>

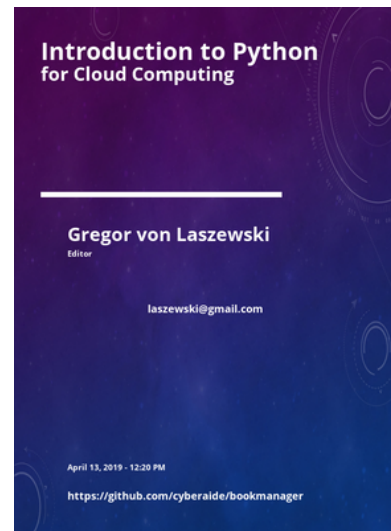


Figure 1: Cover Page

### C. Example creation

```
$ git clone https://github.com/cyberaide/bookmanager.git
$ cd bookmanager
$ pip install -e .
$ bookmanager tests/python.yaml get
$ open dest/book.epub
```

### D. References

- Example Yaml file: <https://github.com/cyberaide/bookmanager/blob/master/tests/python.yml>

### E. Requirements

Book manager requires the existence of some cloudmesh yaml files, In future releases we intend to remove them. Simply do

```
$ mkdir -p ~/.cloudmesh
$ wget -P ~/.cloudmesh https://raw.githubusercontent.com/
```

In addition we require an up to date version of pandoc. Please consult with the pandoc documentation on how to do this. Unfortunately the versions distributed with ubuntu are outdated. On ubuntu you can say:

```
wget -q https://github.com/jgm/pandoc/releases/download/2
sudo dpkg -i pandoc-2.7.2-1-amd64.deb
pandoc --version
```

We recommend pandoc version 2.7.2.

### F. Example Yaml file

The example in the appendix Section VIII shows a yaml file including a table of contents that is used to assemble the document from github locations.

### *G. Automated github links*

It is possible to replace the local link that will be added to the files with a link to a github repository. At this time this is only supported for documents that are in the same repository.

Simply add the following in case your local files are in `../chapter`. While specifying it in the `base`. This variable specifies the link to the source. The variable `github`, will be used to replace the base with a link to the github repository.

file:

```
"github": "https://github.com/cloudmesh-community/book/blob/master/chapters"
"base": "../chapters"
```

### *H. Links*

- Bookmanager home page
- [2] Bookmanager on Pypi
- [3] Bookmanager on GitHub
- [1] Bookmanager Service on Github

## VII. MANUAL PAGE

**bookmanager** -- a helper to create books  
from markdown files in a yaml  
TOC.

### Usage:

```
bookmanager version
bookmanager YAML cover
bookmanager YAML get [--format=FORMAT]
                        [--force]
bookmanager YAML download
bookmanager YAML level
bookmanager YAML epub [--force]
bookmanager YAML pdf
bookmanager YAML html
bookmanager YAML docx
bookmanager YAML check [--format=FORMAT]
bookmanager YAML urls [--format=FORMAT]
bookmanager YAML list [--format=FORMAT]
                        [--details]
```

### Arguments:

YAML the yaml file

### Options:

```
-h --help
-f, --format=FORMAT      [default: markdown]
-d, --details             [default: False]
```

## Description

In principal you only need one command at this time. All other commands are available for test purposes.

You can create an epub with

- **bookmanager** YAML get [--format=FORMAT]

The command searches for all images within the markdown document and fetches them so the document can be created locally with the images. We assume all images in the md document are for now not specified via http locations but via relative locations.

To view the document use your favourite ePub Reader

Other commands include:

- **bookmanager** YAML download [--format=FORMAT]  
downloads the urls into the ./dest directory for local processing
- **bookmanager** YAML check [--format=FORMAT]  
checks if the urls in the yaml file exist
- **bookmanager** YAML urls [--format=FORMAT]  
lists all urls of the yaml file
- **bookmanager** YAML list [--format=FORMAT]  
lists the yaml file

Not implemented are the following features:

- pdf: **bookmanager** pdf book.yml

## YAML Table of Contents format:

The table of contents for the book can be controlled with a simple yaml file that has some specific contextual enhancements. This includes the creation of a BOOK section that has the sections outlined in hierarchical form, and contains chapter and section headers without links that are automatically generated.

Examples of yaml files are provided at

- python.yml

Bugs and enhancement suggestions can be submitted via GitHub

## VIII. YAML FILE EXAMPLE

---

## metadata:

```

image: "cover.png"
title: "Introduction to Python"
subtitle: "for Cloud Computing"
author: 'Gregor von Laszewski'
subauthor: "Editor"
email: "laszewski@gmail.com"
url: "https://github.com/cyberaide/bookmanager"
description: "Book creator"
abstract: "my abstract"
keywords: "pandoc"
stylesheet: "epub.css"
dest: "./dest/book"
filename: "vonLaszewski-python.epub"

```

## git:

```

"book": "https://raw.githubusercontent.com/cloudmesh-community/book/master/chapters"
"credit": "https://raw.githubusercontent.com/cyberaide/bookmanager/master/bookmanager/template"

```

## BOOK:

- PREFACE:
  - "{git.credit}/disclaimer.md"
- INTRODUCTION:
  - "{git.book}/prg/SECTION-PYTHON.md"
  - "{git.book}/prg/python/python-intro.md"
  - "{git.book}/prg/python/python-install.md"
  - "{git.book}/prg/python/python-interactive.md"
  - "{git.book}/prg/python/python-editor.md"
  - "{git.book}/prg/python/python.md"
- LIBRARIES:
  - "{git.book}/prg/python/python-libraries.md"
  - "{git.book}/prg/python/python-data.md"
  - "{git.book}/prg/python/python-matplotlib.md"
  - "{git.book}/prg/python/python-docopts.md"
  - "{git.book}/prg/python/python-cmd5.md"
  - "{git.book}/prg/python/python-cmd.md"
  - "{git.book}/prg/python/opencv/opencv.md"
  - "{git.book}/prg/python/opencv/secchi.md"
- DATA:
  - "{git.book}/SECTION/SECTION-DATA.md"
  - "{git.book}/data/formats.md"
- MONGO:
  - "{git.book}/data/mongodb.md"
  - "{git.book}/data/mongoengine.md"
- APPLICATIONS:
  - "{git.book}/prg/python/fingerprint/fingerprint.md"
  - "{git.book}/prg/python/facedetection/facedetection.md"

## *Refernces*

- [1] Names misisng and G. von Laszewski, “Bookmanager service source code.” GitHub [Online]. Available: <https://github.com/cyberaide/bookmanager-service>
- [2] G. von Laszewski, “Bookmanager.” PyPi [Online]. Available: <https://pypi.org/project/cyberaide-bookmanager/>
- [3] G. von Laszewski, “Bookmanager source code.” GitHub [Online]. Available: <https://github.com/cyberaide/bookmanager>