# audiobook

## *Release 2.0.4*

**Deeapk Raj**

**Dec 11, 2022**

# CONTENTS:

*AudioBook* library can be used to read your favorite books, magazines, web articles, etc. out loud. It is a simple and easy to use library that can be used in any project. It is written in python and is compatible with python 3.7+. It is also compatible with Windows, Linux and Mac OS X.

AudioBook project developed under the MIT license by Py-Contributors. Py-Contributors is open source community of developers who are working on various python projects.

If you love open source contributions.

- Join the community on Discord.

- join the community on Github.

# OTHER PROJECTS BY PY-CONTRIBUTORS:

- Random-Profile-Generator
- Cybel - The Discord Bot
- Twitterify - Tweet-retweet bot

## 1.1 Introduction

*AudioBook* library can be used to read your favorite books, magazines, web articles, etc. out loud. It is a simple and easy to use library that can be used in any project. It is written in python and is compatible with python 3.7+. It is also compatible with Windows, Linux and Mac OS X.

Use cases

- Read your favorite books, magazines, web articles, etc. out loud.
- Save your favorite books, magazines, web articles, etc. as audio files.
- Convert your favorite books, magazines, web articles, etc. to json files.

### 1.1.1 Supported formats

- Pdf
- Epub
- Mobi
- Txt
- Rst
- Html
- Docs

### 1.1.2 Upcoming features

- Support for more formats
- Support for more languages

## 1.2 Installation

### 1.2.1 Install As Pip Package

Install the package using pip:

```
pip install audiobook
```

### 1.2.2 Install From Source

```
git clone https://github.com/Py-Contributors/AudioBook
cd AudioBook

# Install the package
python setup.py install
```

### 1.2.3 Test Installation

```
audiobook --help
```

### 1.2.4 Linux Installation Requirements

If you are using a Linux system and the voice output is not working, then Install espeak , ffmpeg and libespeak1 as shown below:

```
sudo apt-get install espeak ffmpeg libespeak1
```

## 1.3 Command line Usage

*AudioBook* can be used as a command line tool and as a library. only the command line tool is documented here.

Only basic functionality is implemented. The following features are not yet implemented:

- *–delete* option
- *–volume* option
- *–speed* option
- *–save_chapterwise* option

### 1.3.1 Support Format and extraction method

*AudioBook* supports the following formats and extraction methods:

| File Format | Supported | extraction_engine |
|-------------|-----------|-------------------|
| PDF | | pypdf2/pdfminor |
| TXT | | default set |
| EPUB | | default set |
| MOBI | | default set |
| HTML | | default set |
| DOCX | | default set |
| ODT | | default set |

### 1.3.2 As command line tool

```
$ audiobook --help
Usage: audiobook [OPTIONS] COMMAND [ARGS]...

optional arguments:
-h, --help            show this help message and exit
-p [PATH], --path [PATH]
                      book file path
-v, --version         show programs version number and exit
-l, --library         get all books in library
-c, --create-json     create json file from input file
-s, --save-audio      save audio files from input file
-r, --read-book       read the book from input file

commands: read, save, create
```

### 1.3.3 Get AudioBook version

```
$ audiobook --version
```

### 1.3.4 Read a book

```
$ audiobook -p <file_path> -r
```

### 1.3.5 Save audio book

```
$ audiobook -p <file_path> -s
```

Create a json file

```
$ audiobook -p <file_path> -c
```

## 1.4 import as module

```python
from audiobook import AudioBook

# argument: Speech-Speed="slow/normal/fast", volume = 0.0 to 1.0
ab = AudioBook(speed="normal", volume=1.0)
```

### 1.4.1 Read a book

```python
ab.read_book(book_path)
# additional parameter:
    # - password (if book is password protected)
    # - extraction_engine (default: "pypdf2/pdfminer") for pdf files

# example:

ab.read_book("book.pdf", password="1234", extraction_engine="pdfminer")
```

### 1.4.2 Save audio book

```python
ab.save_book(book_path, save_page_wise=False)
# additional parameter:
    # - password (if book is password protected)
    # - extraction_engine (default: "pypdf2/pdfminer") for pdf files
    # - save_page_wise (default: False)
    # - extraction_engine (default: "pypdf2/pdfminer") for pdf files

# example:

ab.save_book("book.pdf", password="1234", save_chapterwise=True, extraction_engine=
→"pdfminer")
```

### 1.4.3 save book as txt file

Under development

## 1.5 Changelog

All notable changes to this project will be documented in this file. The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

### 1.5.1 Unreleased

### 1.5.2 Proposed Changes/features

- Search book in the library
- Delete book from the library

### 1.5.3 V2.0.4

- Extended pdf_parser to extract table of contents
- Command line Support added
- Updated test cases
- Added support for CircleCi CI/CD

### 1.5.4 V2.0.3

- Removed from PyPi

### 1.5.5 V2.0.2 - 22-10-2022

- Docs files support added
- Pdfminer as engine added
- ODT file support added

### 1.5.6 V2.0.1 - 17-10-2022

- Mobi file support
- Epub file support
- User can now save the audiobook for future
- User can now listen to the book from the library

### 1.5.7 V2.0.0 - 14-10-2022

- Save Audio Book locally
- Listen to the book
- Speech-speed control
- Read password-protected PDF
- Create JSON file for the book

### 1.5.8 V1.0.5 - 24-10-2020

- Deprecated version
- Only can read PDF files

### 1.5.9 First Release - 22-10-2020

## 1.6 Roadmap and future plans

- *more test coverage*
- *more supported file formats*
- *save book as txt file*

## 1.7 Run test cases

For the test cases, we are using *pytest*. The test cases are located in the *tests* directory. To run the test cases, you can use the following command:

```
$ pytest tests
```

You can also run the test cases with coverage: