

---

# **pdfcropper**

***Release 1.0.0***

**Sep 27, 2017**



---

## Contents

---

<b>1</b>	<b>Česká verze</b>	<b>3</b>
<b>2</b>	<b>Script</b>	<b>5</b>
<b>3</b>	<b>API</b>	<b>7</b>
<b>4</b>	<b>Source code</b>	<b>11</b>
<b>5</b>	<b>Installation</b>	<b>13</b>
<b>6</b>	<b>Testing</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>



This project is aimed to provide simple and easy to use script for cropping PDF files as part of the E-deposit project.



# CHAPTER 1

---

Česká verze

---

- /index\_cz





From user's point of view, following script is his interface with pdfcropper:

### pdf\_cropper.py

Help:

```
usage: pdf_cropper.py [-h] [-c X X X X] [-e X X X X] [-r PAGE [PAGE ...]]
                    [-o OUTPUT]
                    filename

PDF cropper. This program can be used to crop and remove pages from PDF files.

positional arguments:
  filename              Input filename.

optional arguments:
  -h, --help            show this help message and exit
  -c X X X X, --crop X X X X
                        Crop vector for all or even pages. Vector is in format
                        LEFT RIGHT TOP BOTTOM. -c 50 50 10 10 for example.
  -e X X X X, --crop-odd X X X X
                        Crop vector for odd pages. Vector is in format LEFT
                        RIGHT TOP BOTTOM. -c 50 50 10 10 for example.
  -r PAGE [PAGE ...], --remove PAGE [PAGE ...]
                        Remove following pages. Page numbers starts from zero.
  -o OUTPUT, --output OUTPUT
                        Save modified file to given destination. If not set,
                        suffix '_cropped.pdf' is used.
```

Example of the use:

```
./pdf_cropper.py -c 10 10 5 5 -r 0 -- ukazka01b.pdf
```

Which will crop the file `ukazka01b.pdf` ten millimeters from left and right and five millimeters from up and down. `-r` or `--remove` parameter will also remove page at index 0. Result can be seen in `ukazka01b_cropped.pdf`.

## **Česká verze**

- `/scripts/pdf_cropper_cz`

From programmer's point of view, there are few collections of functions, which he (or she) may use:

### pdfcropper package

#### Submodules

##### cropper submodule

This module contains wrappers over `pyPdf`, which provides high level API for cropping PDF files.

---

**Note:** All sizes used in this module should be in millimeters.

---

#### API

`pdfcropper.cropper.POINT_MM = 0.35277777777777775`

1pt = inch/72, 1 inch = 25.4 mm

`pdfcropper.cropper.get_width_height (page)`

Return width and height of the *page*.

**Parameters** `page (obj)` – `PdfFileReader.pages` instance.

**Returns** (`width`, `height`) as float, in millimeters.

**Return type** `tuple`

`pdfcropper.cropper.crop_page (page, left, right, top, bottom)`

Crop *page* to size given by *left*, *right*, *top* and *bottom*.

**Parameters**

- **page** (*obj*) – pyPdf PdfFileReader’s page object.
- **left** (*int*) – Cut X millimeters from left.
- **right** (*int*) – Cut X millimeters from right.
- **top** (*int*) – Cut X millimeters from top.
- **bottom** (*int*) – Cut X millimeters from bottom.

**Warning:** This functions modifies the *page* reference!

**Returns** Modified page object.

**Return type** obj

`pdfcropper.cropper.crop_all(pdf, left, right, top, bottom, remove=[])`

Crop all pages in *pdf*. Remove pages specified by *remove*.

**Parameters**

- **pdf** (*obj*) – pyPdf PdfFileReader object.
- **left** (*int*) – Cut X millimeters from left.
- **right** (*int*) – Cut X millimeters from right.
- **top** (*int*) – Cut X millimeters from top.
- **bottom** (*int*) – Cut X millimeters from bottom.
- **remove** (*list/tuple, default []*) – List of integers. As the function iterates thru the pages in *pdf*, indexes of the pages which matches those in *remove* will be skipped.

**Returns** PdfFileWriter instance, with modified pages.

**Return type** obj

`pdfcropper.cropper.crop_differently(pdf, even_vector, odd_vector, remove=[])`

Crop *pdf* even pages by *even\_vector* and odd pages by *odd\_vector*. Remove pages specified by *remove*.

**Parameters**

- **pdf** (*obj*) – pyPdf PdfFileReader object.
- **even\_vector** (*list*) – List of coordinates to which all even pages will be cropped. [Left, Right, Top, Bottom].
- **odd\_vector** (*list*) – List of coordinates to which all odd pages will be cropped. [Left, Right, Top, Bottom].
- **remove** (*list/tuple, default []*) – List of integers. As the function iterates thru the pages in *pdf*, indexes of the pages which matches those in *remove* will be skipped.

**Returns** PdfFileWriter instance, with modified pages.

**Return type** obj

`pdfcropper.cropper.remove_pages(pdf, remove)`

Remove pages specified in vector *remove*.

**Parameters**

- **pdf** (*obj*) – pyPdf PdfFileReader object.

- **remove** (*list/tuple*, *default []*) – List of integers. As the function iterates thru the pages in *pdf*, indexes of the pages which matches those in *remove* will be skipped.

**Returns** PdfFileWriter instance, with modified pages.

**Return type** obj

pdfcropper.cropper.**read\_pdf** (*filename*)

Read pdf file specified by *filename*.

**Parameters** **filename** (*str*) – Path to the pdf file.

**Returns** PdfFileReader object.

**Return type** obj

pdfcropper.cropper.**save\_pdf** (*filename*, *content*)

Save *content* to *filename*.

**Parameters**

- **filename** (*str*) – Path which will be used for *content*.
- **content** (*obj*) – PdfFileWriter object which will be serialized.



## CHAPTER 4

---

### Source code

---

This project is released as opensource (GPL) and source codes can be found at GitHub:

- <https://github.com/edeposit/pdfcropper>





## CHAPTER 5

---

### Installation

---

Module is hosted at [PYPI](#), and can be easily installed using [PIP](#):

```
sudo pip install pdfcropper
```



Almost every feature of the project is tested in unit/integration tests. You can run this tests using provided `run_tests.sh` script, which can be found in the root of the project.

## Requirements

This script expects that `pytest` is installed. In case you don't have it yet, it can be easily installed using following command:

```
pip install --user pytest
```

or for all users:

```
sudo pip install pytest
```



### p

`pdfcropper`, [7](#)

`pdfcropper.cropper`, [7](#)



### C

`crop_all()` (in module `pdfcropper.cropper`), 8  
`crop_differently()` (in module `pdfcropper.cropper`), 8  
`crop_page()` (in module `pdfcropper.cropper`), 7

### G

`get_width_height()` (in module `pdfcropper.cropper`), 7

### P

`pdfcropper` (module), 7  
`pdfcropper.cropper` (module), 7  
`POINT_MM` (in module `pdfcropper.cropper`), 7

### R

`read_pdf()` (in module `pdfcropper.cropper`), 9  
`remove_pages()` (in module `pdfcropper.cropper`), 8

### S

`save_pdf()` (in module `pdfcropper.cropper`), 9