

Организованное
местоПсевдофизНаук

Библиотека незгораемая

Super
Hot

Прикладное Ахтоведение

справочник
персонажа-лучника

Издательство «светлое кольцо»

Нуарсити
2017

7. Сохраняем

```
args = init_args(sys.argv)
pdf_in = os.path.abspath(args.pdf_in[0])
pdf_out = os.path.abspath(args.pdf_out[0])
rotate180_even = args.rotate180_even
out_one_by_one = args.out_one_by_one
save_odd_even = args.save_odd_even
add_blank_before, add_blank_after = args.add_blank_before, args.add_blank_after
dpi = args.dpi

pout(f"Getting from pdf \"{pdf_in}\" pages with dpi={dpi}... ", endl=False)
imgs = pdf_convert_from_path(pdf_in, dpi)
pout("OK")
if(len(imgs) <= 0):
    print(f"\"pdf_in\" has {len(imgs)} pages. Exiting...")
    exit()

imgs = solve_transform(imgs, args)

imgs = adding_blank(imgs, add_blank_before, add_blank_after)

imgs = fill_4(imgs)

imgs_odd, imgs_even = split(imgs)

imgs_odd = A5_A4(imgs_odd, False)
imgs_even = A5_A4(imgs_even, True)

if(notate180_even == True):
    imgs_even = rotate_even_180(imgs_even)

if(out_one_by_one == False):
    pout("Connecting the pages together: first the numbers with odd, then with even...", endl=False)
    imgs = imgs_odd + imgs_even # insert
    pout("OK")
else:
    pout(f"Connecting pages together: odd ({len(imgs_odd)}) and even ({len(imgs_even)}) pages alternate
one by one...", endl=False)
    imgs = [imgs_odd[i//2] if i%2==0 else imgs_even[i//2] for i in range(len(imgs_odd)+len(imgs_even))]
    pout("OK")

pout(f"Saving output pdf \"{pdf_out}\"...", endl=False)
imgs[0].save(pdf_out, save_all=True, append_images=imgs[1:])
pout("OK")

if(save_odd_even == True):
    fn, fe = os.path.splitext(pdf_out)
    odds_path, evens_path = fn + "_odds" + fe, fn + "_evens" + fe
    pout(f"Saving odds \"{odds_path}\"...", endl=False)
    imgs_odd[0].save(odds_path, save_all=True, append_images=imgs_odd[1:])
    pout("OK")
    pout(f"Saving odds \"{evens_path}\"...", endl=False)
    imgs_even[0].save(evens_path, save_all=True, append_images=imgs_even[1:])
    pout("OK")

pout(f"{'='*20}DONE!{'='*20}")
```




```
# -*- coding: utf-8 -*-
```

```
import argparse
import sys
import os
```

```
from pdf2image import convert_from_path as pdf_convert_from_path
```

```
from img_hundler import ImageHundler
```

```
def init_args(args: list) -> "ArgumentParser":
    parser = argparse.ArgumentParser(prog = "mkbook",
                                     description="Make book from pdf to print.")
    parser.add_argument("pdf_in", type=str, nargs=1,
                        help="Path to source pdf.")
    parser.add_argument("pdf_out", type=str, nargs=1,
                        help="Path to output pdf")
    parser.add_argument("--dpi", type=int, default=200, required=False,
                        help="More value means better quality, but larger file size.")
    parser.add_argument("--rotate_left", default=True, action='store_false',
                        help="When rotating landscape (horizontally) sheets, rotate them to the left
                        (counterclockwise).")
    parser.add_argument("--rotate180_even", default=False, action='store_true',
                        help="Rotates output pages with even numbers by 180 degrees.")
    parser.add_argument("--out_by_one", default=False, action='store_true',
                        help="The output pages go one after the other, not the even ones first, and then the
                        odd ones.")
    parser.add_argument("--save_odd_even", default=False, action='store_true',
                        help="Additionally save pages with odd and even numbers separately.")
    parser.add_argument("--add_blank_before", type=int, default=0,
                        help="Add blank sheets to the beginning.")
    parser.add_argument("--add_blank_after", type=int, default=0,
                        help="Add blank sheets to the end.")
    args = parser.parse_args(args[1:])
```

```
    return args
```

```
def solve_transform(imgs: list, args: list) -> list:
    left_right = not args.rotate_left
    pout("Resizing for A4 and landscape to portrait... ", endl=False)
    for i, img_i in enumerate(imgs):
        new_img = ImageHundler(img_i)
        new_img.resize(444_fill())
        img_size = new_img.get().size
        if(img_size[0] > img_size[1]): # Landscape
            new_img.rotate90(left_right)
        imgs[i] = new_img.get()
    pout("OK")
```

```
pout("Bringing all pages to the same size...", endl=False)
max_w = max(imgs, key=lambda x:x.size[0]).size[0]
max_h = int(max_w* (2**(0.5)) + 0.5)
res = []
for img_i in imgs:
    buff = ImageHundler(img_i)
    buff.resize((max_w, max_h))
    res.append(buff.get())
pout("OK")
```

```
    return res
```

```
def adding_blank(imgs: list, before: int, after: int) -> list:
    blank = ImageHundler(imgs[0])
    blank = blank.get_as_empty()

    res_before = []
    for i in range(before):
        res_before.append(blank.copy())

    res_after = []
    for i in range(after):
        res_after.append(blank.copy())
```

Barry! Breakfast is ready!

Ooming!

Hang on a second.

Hello?

- Barry?
- Adam?

- Oan you believe this is happening?
- I can't. I'll pick you up.

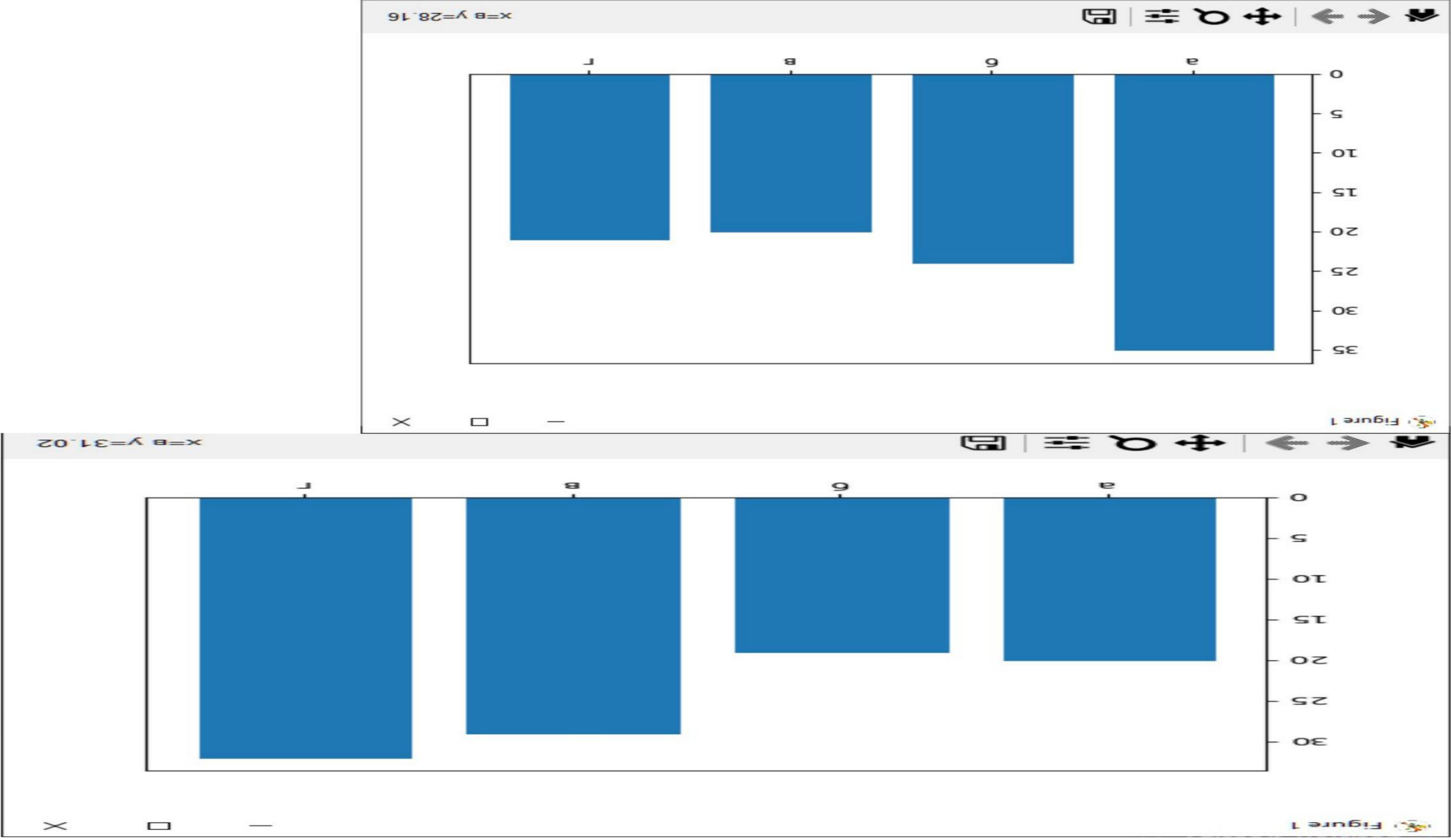
Looking sharp.

Use the stairs. Your father
paid good money for those.

Sorry. I'm excited.

Here's the graduate.

Such a hothead.
I guess he could have
just gotten out of the way.
I love this incorporating
an amusement park into our day.
That's why we don't need vacations.



Содержание

Введение	5
1 Аналитический раздел	6
1.1 Анализ того и сего	6
1.2 Существующие подходы к созданию всячины	7
2 Конструкторский раздел	10
2.1 Архитектура всячины	10
2.2 Подсистема всякой ерунды	10
2.2.1 Блок-схема всякой ерунды	10
3 Технологический раздел	12
4 Экспериментальный раздел	14
Заключение	15
Список использованных источников	16
А Картинки	17
Б Еще картинки	18

Bee Movie Script

According to all known laws
of aviation,

there is no way a bee
should be able to fly.

Its wings are too small to get
its fat little body off the ground.

The bee, of course, flies anyway

because bees don't care

what humans think is impossible.

Yellow, black. Yellow, black.
Yellow, black. Yellow, black.

Ooh, black and yellow!
Let's shake it up a little.



В необъятных просторах лунной пустыни, где магический свет покрывает землю, это
существо, которое отличалось своей невероятной эксцентричностью. Это было нечто
среднего между космической курьер и пурпурным единорогом. У него были перья
разных цветов, меняющихся в зависимости от его настроения, а его рога пронизывали



We're very proud of you, son.

A perfect report card, all B's.
Very proud.

Ma! I got a thing going here.

- You got lint on your fuzz.

- Ow! That's me!

- Wave to us! We'll be in row 118,000.

- Bye!

Barry, I told you,

stop flying in the house!

- Hey, Adam.

- Hey, Barry.

- Is that fuzz gel?

- A little. Special day, graduation.

Never thought I'd make it.

Three days grade school,

three days high school.

Those were awkward.

Three days college. I'm glad I took

a day and hitchhiked around the hive.

You did come back different.

- Hi, Barry.

- Artie, growing a mustache? Looks good.

- Hear about Frankie?

- Yeah.

- You going to the funeral?

- No, I'm not going.

Everybody knows,

sting someone, you die.

Don't waste it on a squirrel.

```

    return res_before + imgs + res_after

def fill_4(imgs: list) -> list:
    c = len(imgs)
    if(c % 4 == 0):
        return imgs
    else:
        pout("Adding blank pages so that their number is a multiple of 4...", endl=False)
        buff = ImageHundler(imgs[0])
        diff = 4 - (c % 4)
        res = imgs + [buff.get_as_empty() for _ in range(diff)]
        pout("OK")
        return res

def split(imgs: list) -> "[odd], [even]":
    pout("Dividing pages into 2 sets: odd and even...", endl=False)
    odd, even = [], []
    for i, img_i in enumerate(imgs):
        if((i+1)%2 == 1): # odd
            odd.append(img_i)
        else: # even
            even.append(img_i)
    pout("OK")
    return (odd, even)

def rotate_even_180(imgs: list) -> list:
    pout(f"Rotating pages with even numbers 180 degrees...", endl=False)
    for i, img_i in enumerate(imgs):
        imgs[i] = imgs[i].rotate(180, expand=True)
    pout("OK")
    return imgs

def A5_A4(imgs: list, odd_even: bool = False) -> list:
    """
    odd_even == False -> odd
    odd_even == True -> even
    """
    odd_even_str = "odd" if odd_even == False else "even"
    pout(f"Fastening the pages in a set of {odd_even_str} pages...", endl=False)
    c = len(imgs)
    if(c % 2 != 0):
        raise ValueError(f"Len ({c}) of imgs must be even! ")

    res = []
    for i in range(c//2):
        img_l, img_r = imgs[i], imgs[c-1-i]
        if(odd_even == False): # !!!
            img_l, img_r = img_r, img_l
        img_l = ImageHundler(img_l)
        img_l.append_right(img_r)
        res.append(img_l.get())

    pout("OK")
    return res

def pout(s: str, endl: bool = True):
    if(endl==False):
        print(s, end="")
    else:
        print(s)
    sys.stdout.flush()

if __name__ == '__main__':
    """
    1. Получить все листы из pdf, как png.
    2. Сделать их все формата A4 (1:1.41).
    3. Сделать все листы портретными.
    4. Добавить пустых листов, чтобы их кол-во было кратно 4.
    5. Делаем 2 множества: чётных и нечётных.
    6. Соединяем (append_right) элементы этих множеств.
    """

```




ПЕРИОДИЧЕСКАЯ СИСТЕМА ХИМИЧЕСКИХ ЭЛЕМЕНТОВ Д. И. МЕНДЕЛЕЕВА

[illegible]