



empenoso

24 фев в 05:25

## Как сканировать документы А3 формата, если под рукой только сканер А4 формата

Простой

5 мин

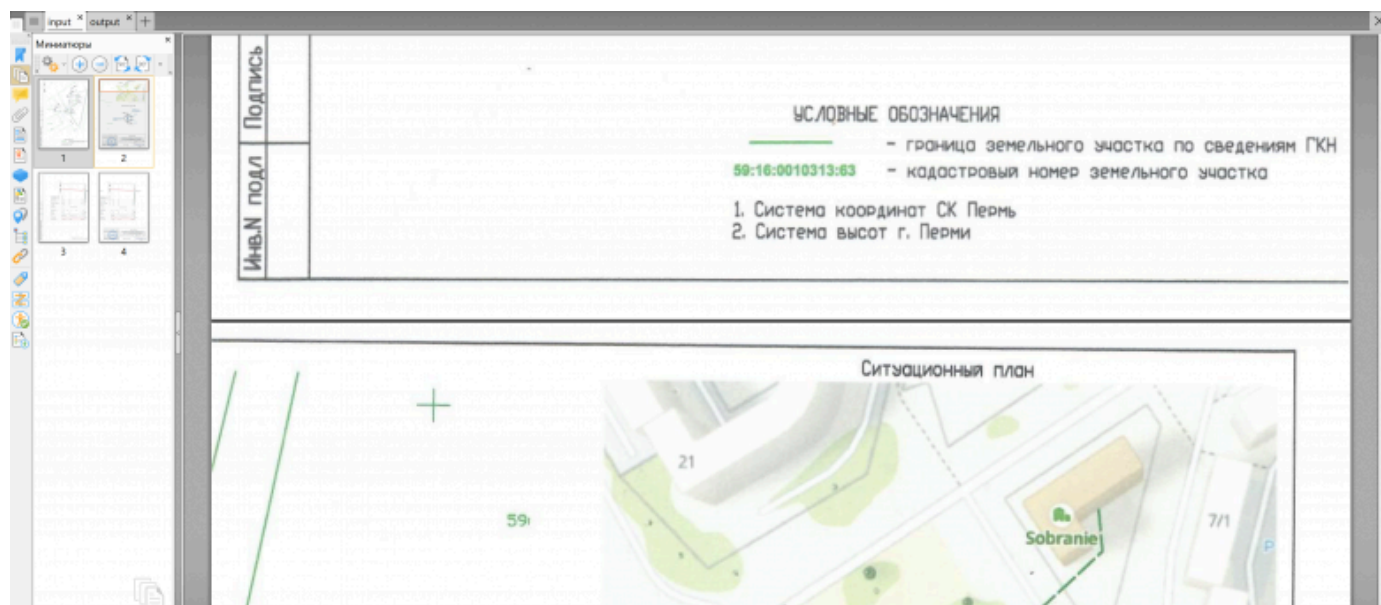
5.2K

Open source\*, Python\*, Лайфхаки для гиков, PDF

Кейс

Оцифровка чертежей и документов формата А3 не представляет сложности, если у вас есть сканер соответствующего размера. Однако что делать, если под рукой лишь стандартное МФУ формата А4?

Формат А3 в два раза больше А4: это два стандартных листа, соединенных по длинной стороне. В результате, чтобы оцифровать А3, приходится сканировать его по частям - левую и правую половинки, а затем как-то их объединять.



Ручная склейка занимает много времени, особенно если документов много. А если тратить время на ручную склейку не хочется? Автоматизация такого процесса избавляет от рутинной работы. Актуально для чертежей и схем.

РЕКЛАМА

**Получи грант за код**

Конкурс open source проектов

Для автоматизации склейки PDF-файлов потребуется всего несколько инструментов.

Первым делом понадобится **Python** — это простой в использовании язык программирования, идеально подходящий для подобных задач.

Для работы с PDF-файлами нам понадобится библиотека **PyPDF2**. Она позволяет извлекать страницы из PDF, объединять их, изменять порядок и выполнять другие операции.



Источник красивой картинки: dev.to

Чтобы склеить изображения из двух половинок, потребуется библиотека **Pillow**. С её помощью можно выполнять базовую обработку изображений: изменять размеры, соединять их и сохранять в нужном формате. Эта библиотека будет полезна, если ваши сканы нужно предварительно подогнать по размерам или качеству.

## Установка Python

Если Python ещё не установлен на вашем компьютере, скачайте его с [официального сайта](#). Выберите последнюю стабильную версию для вашей операционной системы и установите её, следуя инструкциям. Во время установки не забудьте отметить пункт "Добавить Python



**Получи грант за код**

Конкурс open source проектов

## Установка библиотек

После установки Python, нужно добавить необходимые библиотеки. Сделать это просто: откройте терминал (или командную строку) и выполните следующие команды:

```
pip install PyPDF2 Pillow
```

Эти команды загрузят и установят всё необходимое. Теперь вы полностью готовы к написанию и запуску скрипта для автоматической склейки ваших PDF-документов.

## Подготовка исходного файла

Перед началом автоматической склейки важно правильно подготовить исходный PDF-файл. Он должен содержать сканы двух половинок листа формата A4 каждая, расположенные в виде отдельных страниц. Это нужно для того, чтобы программа смогла корректно объединить их в одну страницу.

Важно, чтобы страницы PDF-файла следовали в правильном порядке:

1. **Первая страница** — это левая половина листа A3.
2. **Вторая страница** — правая половина.



**Получи грант за код**

Конкурс open source проектов



### Получи грант за код

Конкурс open source проектов

Каждый лист A3 будет представлен двумя страницами. Если в вашем документе несколько листов, их половинки должны чередоваться так же: сначала левая часть первого листа, потом правая, затем левая часть второго листа, и так далее.

Эта структура поможет программе автоматически склеить половинки в правильном порядке.

## Как работает процесс склейки

Автоматизация склейки документов A3 формата включает извлечение страниц, обработку изображений и преобразование в PDF. Вот как это работает:

### 1. Извлечение страниц из PDF

Каждая страница PDF, представляющая половину листа A3, сначала извлекается. Этот процесс позволяет получить изображения страниц, которые затем можно обработать.

### 2. Преобразование страниц в изображения

С помощью библиотеки Pillow страницы преобразуются в изображения. Это необходимо для удобной манипуляции — например, склеивания или вращения.

### 3. Объединение изображений

Изображения, соответствующие левой и правой половинам листа, склеиваются по горизонтали. Этот этап объединяет две части в единый файл, точно воспроизводящий лист A3.

### 4. Конвертация обратно в PDF

Склеенные изображения снова преобразуются в PDF с помощью PyPDF2. Это позволяет сохранить формат документа и сделать его удобным для дальнейшего использования.



**Получи грант за код**

Конкурс open source проектов

## Сам скрипт

Вот и сам скрипт: он автоматически склеивает половинки страниц формата A4 в один файл формата A3. Может использоваться на Windows, macOS и Linux:

```
# pip install PyPDF2 pillow

# Подробное описание: https://habr.com/ru/articles/875846/

import sys
sys.stdout.reconfigure(encoding='utf-8')

import os
from PyPDF2 import PdfReader, PdfWriter
from PIL import Image

# Выводим общую информацию о начале обработки PDF
print("[INFO] Начало обработки PDF...\n")

# Имя входного PDF-файла
input_pdf_file = "input.pdf"
# Имя выходного PDF-файла
output_pdf_file = "output.pdf"

# Углы поворота для левой и правой частей (в градусах)
# Укажите поворот для каждой части, например, 0, 90, 180 или 270
left_rotation = 0
right_rotation = 180

# Создадим временную папку для сохранения изображений
output_images_dir = "temp_images"

# Считываем PDF и выгружаем каждую страницу как отдельное изображение
```



**Получи грант за код**

Конкурс open source проектов

```
reader = PdfReader(input_pdf_file)
page_images = []

print(f"[INFO] Извлечение изображений из {input_pdf_file}...")
for page_number, page in enumerate(reader.pages):
    # Получаем раскодированные данные изображений
    x_object = page.get("/Resources").get("/XObject").get_object()
    for obj in x_object:
        if x_object[obj].get("/Subtype") == "/Image":
            data = x_object[obj].get_data()
            image_path = os.path.join(output_images_dir, f"page_{page_number + 1}.png")
            with open(image_path, "wb") as image_file:
                image_file.write(data)
            page_images.append(image_path)
            print(f"[INFO] Изображение страницы {page_number + 1} сохранено: {image_path}")

# Обрабатываем и склеиваем по две страницы
print("[INFO] Склеивание изображений в формат A3...")
glued_images = []
for i in range(0, len(page_images), 2):
    left_image_path = page_images[i]
    right_image_path = page_images[i + 1] if i + 1 < len(page_images) else None

    # Загружаем изображения
    left_image = Image.open(left_image_path)
    left_image = left_image.rotate(left_rotation, expand=True) # Применяем поворот лев

    if right_image_path:
        right_image = Image.open(right_image_path)
        right_image = right_image.rotate(right_rotation, expand=True) # Применяем повс

    # Создаем новое изображение для A3 склеенного
    total_width = left_image.width + right_image.width
    max_height = max(left_image.height, right_image.height)
    new_image = Image.new("RGB", (total_width, max_height))

    # Вставляем левую и правую части
    new_image.paste(left_image, (0, 0))
    new_image.paste(right_image, (left_image.width, 0))
    glued_image_path = os.path.join(output_images_dir, f"glued_page_{i // 2 + 1}.pr
```

**Получи грант за код**

Конкурс open source проектов

```
# Конвертируем склеенные страницы обратно в PDF
print("[INFO] Конвертация склеенных изображений обратно в PDF...")
writer = PdfWriter()
for glued_image_path in glued_images:
    image = Image.open(glued_image_path)
    pdf_page_path = glued_image_path.replace(".png", ".pdf")
    image.save(pdf_page_path, "PDF", resolution=300.0)
    with open(pdf_page_path, "rb") as pdf_page_file:
        writer.add_page(PdfReader(pdf_page_file).pages[0])

# Сохраняем итоговый PDF
with open(output_pdf_file, "wb") as output_pdf:
    writer.write(output_pdf)

print(f"[INFO] Обработка завершена. Итоговый PDF сохранен как {output_pdf_file}")
```

Результат:



**Получи грант за код**

Конкурс open source проектов



## Заключение

После завершения обработки откройте итоговый PDF и убедитесь, что страницы формата A3 выглядят ровно и соединены.

Скрипт можно доработать для решения более сложных задач. Например:

- Склейка нестандартных форматов.
- Использование компьютерного зрения для того, чтобы исключить небольшие повороты каждого из листов.

Автоматизация процесса оцифровки больших форматов избавляет от рутины и экономит время, особенно при работе с чертежами, планами и схемами.



**Получи грант за код**

Конкурс open source проектов

**Автор:** Михаил Шардин

 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

24 февраля 2025 г.

**Теги:** [сканирование](#), [документы](#), [чертежи](#), [PyPDF2](#), [Pillow](#), [сезон open source](#)

**Хабы:** [Open source](#), [Python](#), [Лайфхаки для гиков](#), [PDF](#)

## Редакторский дайджест



Присылаем лучшие статьи раз в месяц



Оставляя свою почту, я принимаю [Политику конфиденциальности](#) и даю согласие на получение рассылок



**212**

Карма

**63.3**

Рейтинг

**Михаил Шардин** [@empenoso](#)

[Автоматизация](#) / [Данные](#) / [Финансы](#) / [Умные дома](#)

[Подписаться](#)



[Сайт](#) [Сайт](#) [GitHub](#)



**Получи грант за код**

Конкурс open source проектов

 Комментарии 9

## Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ

**Tirarex**

14 часов назад

### Как я делал сеть на 2,5 гигабита с минимальным бюджетом — апгрейд, доступный каждому



Простой



9 мин



11K

Тutorial



+41



83



34

**Erwinmal**

18 часов назад

### Сэндвич, сэр? История британских бутербродов от аристократических салонов до вокзальных буфетов



Простой



13 мин



3K

**Получи грант за код**

Конкурс open source проектов

 +41 20 2**oneastok**

19 часов назад

## Умное зеркало на Raspberry Pi: пошаговое руководство



Простой



4 мин



4.5K

Обзор

Перевод

 +22 56 14**iLushkersky**

14 часов назад

## Жизнь на Марсе? (снова)



Простой



3 мин



2.3K

 +17 5 10**TrexSelectel**

16 часов назад

## Nintendo Virtual Boy: неожиданное возрождение виртуальной реальности из 90-х



5 мин



1.1K

 +14 3 3**mio\_anni**

19 часов назад

## От мини-ЭВМ и перфокарт к IDE и фреймворкам. Как поменялось программирование за 50 лет — взгляд изнутри



12 мин



1.9K

 +12 15 35**Получи грант за код**

Конкурс open source проектов

## Станислав Петров: «Ключевые отличия РЕД ОС М от Android – вовсе не в интерфейсе»

 Средний  8 мин  6.5K

Интервью

 +10

 10

 43



Albert\_Wesker

19 часов назад

## Миф о быстром и медленном пути выполнения программы

 Средний  11 мин  1.5K

Обзор

Перевод

 +9

 16

 0



kilokanat

5 часов назад

## Механическая клавиатура LARKeyboard

 Простой  5 мин  511

Тutorial

 +8

 4

 2



beeline\_cloud

10 часов назад

## Научный «дипфейк»? Как галлюцинации нейросетей — и другие проблемы — просачиваются в академические статьи

 Простой  8 мин  885

Аналитика

 +8

 11

 2



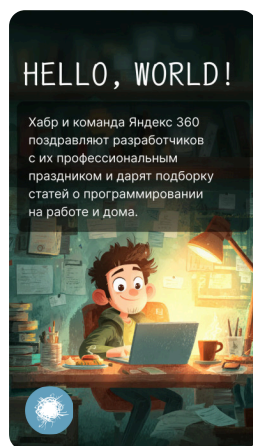
Получи грант за код

Конкурс open source проектов

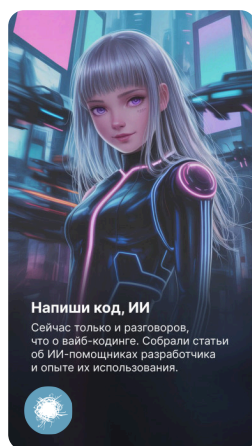
Турбо

Показать еще

## ИСТОРИИ



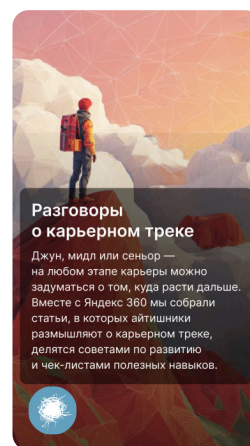
**Чай, торт и код: с Днём программиста!**



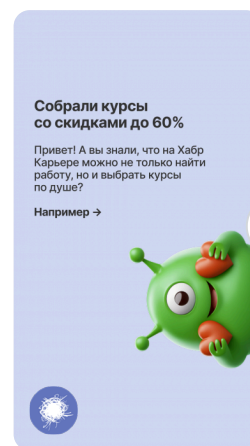
**Made in AI**



**Чего хотят лиды в бигтехе?**



**Как расти в IT: советы, гайды и опыт сеньоров**



**Курсы со скидками до 60%**

## ВАКАНСИИ

## Python разработчик

от 75 000 Р · ITK academy · Казань · Можно удаленно

## Ведущий Python разработчик

от 200 000 до 250 000 Р · Сбер · Москва

## Fullstack разработчик веб-сервиса | Python, Next.js (React)

от 100 000 Р · MatchQ · Можно удаленно

## Python разработчик

от 120 000 Р · DimaTech Ltd · Краснодар · Можно удаленно

## Инженер по автоматизированному тестированию (python)

от 200 000 Р · ИНГО БАНК · Можно удаленно



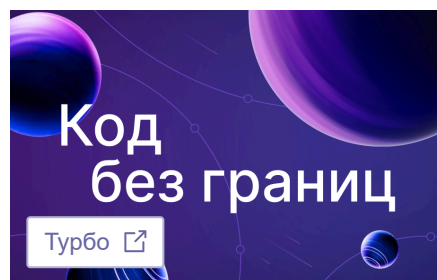
**Получи грант за код**

Конкурс open source проектов

## МИНУТОЧКУ ВНИМАНИЯ



2 пилота и 50 сценариев: ИБ-команда тестирует NGFW

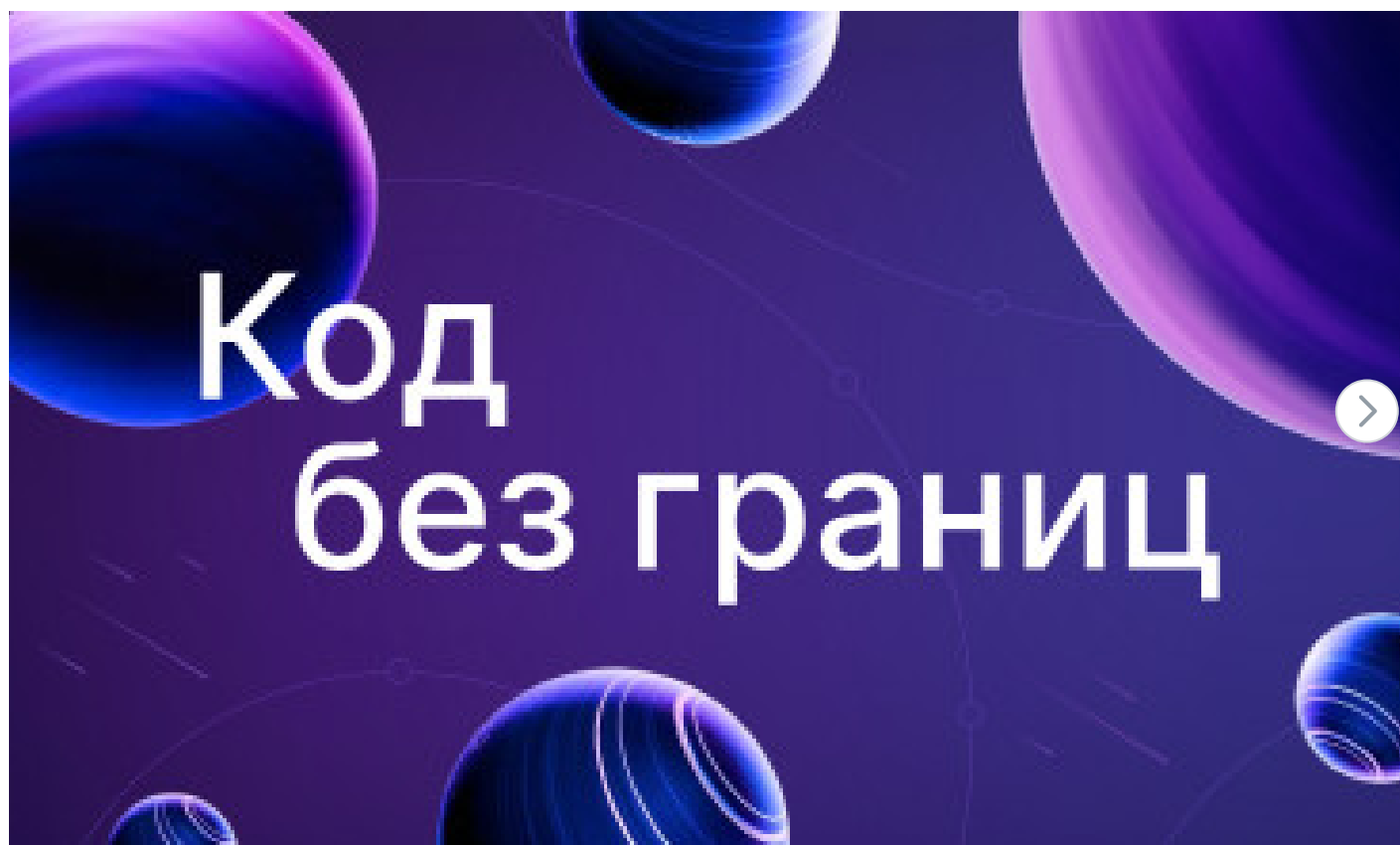


Отправь свой open source проект на конкурс и выиграй грант



Посмотри в Календарь, вдруг сегодня есть мероприятие?

## БЛИЖАЙШИЕ СОБЫТИЯ



3 сентября – 31 октября

**Программа грантов для развития open source проектов «Код без границ»**



**Получи грант за код**

Конкурс open source проектов

[Больше событий в календаре](#)

# Хабр



 [Настройка языка](#)

[Техническая поддержка](#)

© 2006–2025, Habr



**Получи грант за код**

Конкурс open source проектов