

1000+

вакансий с удалёнкой



Хабр Карьера



empenoso

24 фев в 03:25

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

Простой

5 мин

4.5K

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.

Что понадобится для выполнения задачи

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 в PATH", чтобы команды работали из терминала.

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 А4 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 А3.
2. **Вторая страница** — правая половина.

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.



Имя	Тип	Размер
..	<Папка>	
glued_page_1	pdf	308 471
glued_page_1	png	1 826 963
glued_page_2	pdf	278 951
glued_page_2	png	1 300 803
page_1	png	139 573
page_2	png	163 137
page_3	png	109 541
page_4	png	159 912

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"
os.makedirs(output_images_dir, exist_ok=True)

# Считываем PDF и выгружаем каждую страницу как отдельное изображение
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 А3...")
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 лева

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

    # Создаем новое изображение для А3 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}.png")
    new_image.save(glued_image_path)
    glued_images.append(glued_image_path)
    print(f"[INFO] Готово: склеенное изображение {glued_image_path}")

# 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)
```

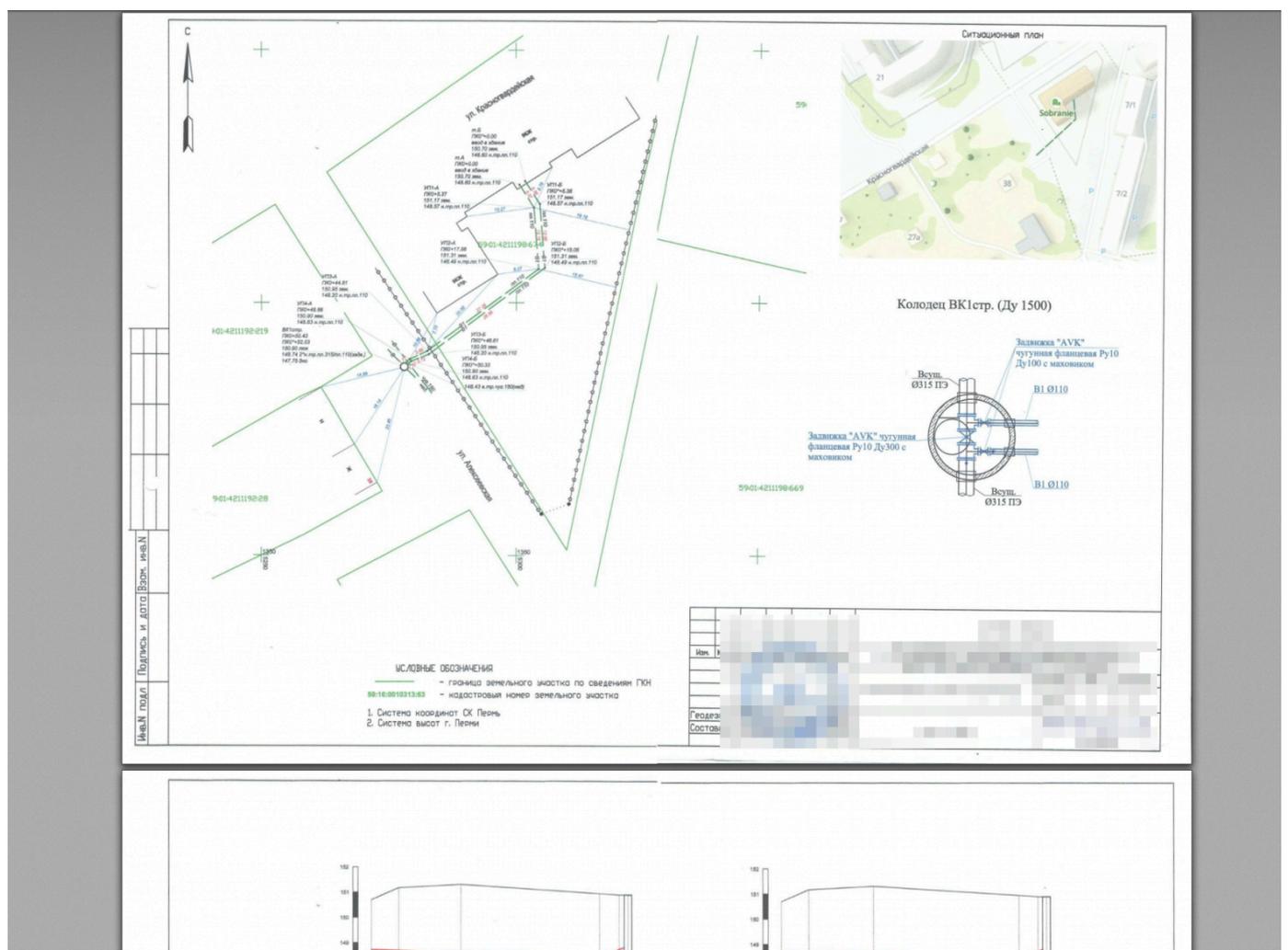
```
with open(pdf_page_path, "rb") as pdf_page_file:
    writer.add_page(PdfReader(pdf_page_file).pages[0])
```

```
# 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:



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.

Не забывайте сохранять резервные копии исходных файлов — это защитит вас от потери данных.

Автор: Михаил Шардин

 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

24 февраля 2025 г.

Теги: [сканирование](#), [документы](#), [чертежи](#), [PyPDF2](#), [Pillow](#)

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

Редакторский дайджест



Присылаем лучшие статьи раз в месяц



183

87.1

Карма

Рейтинг

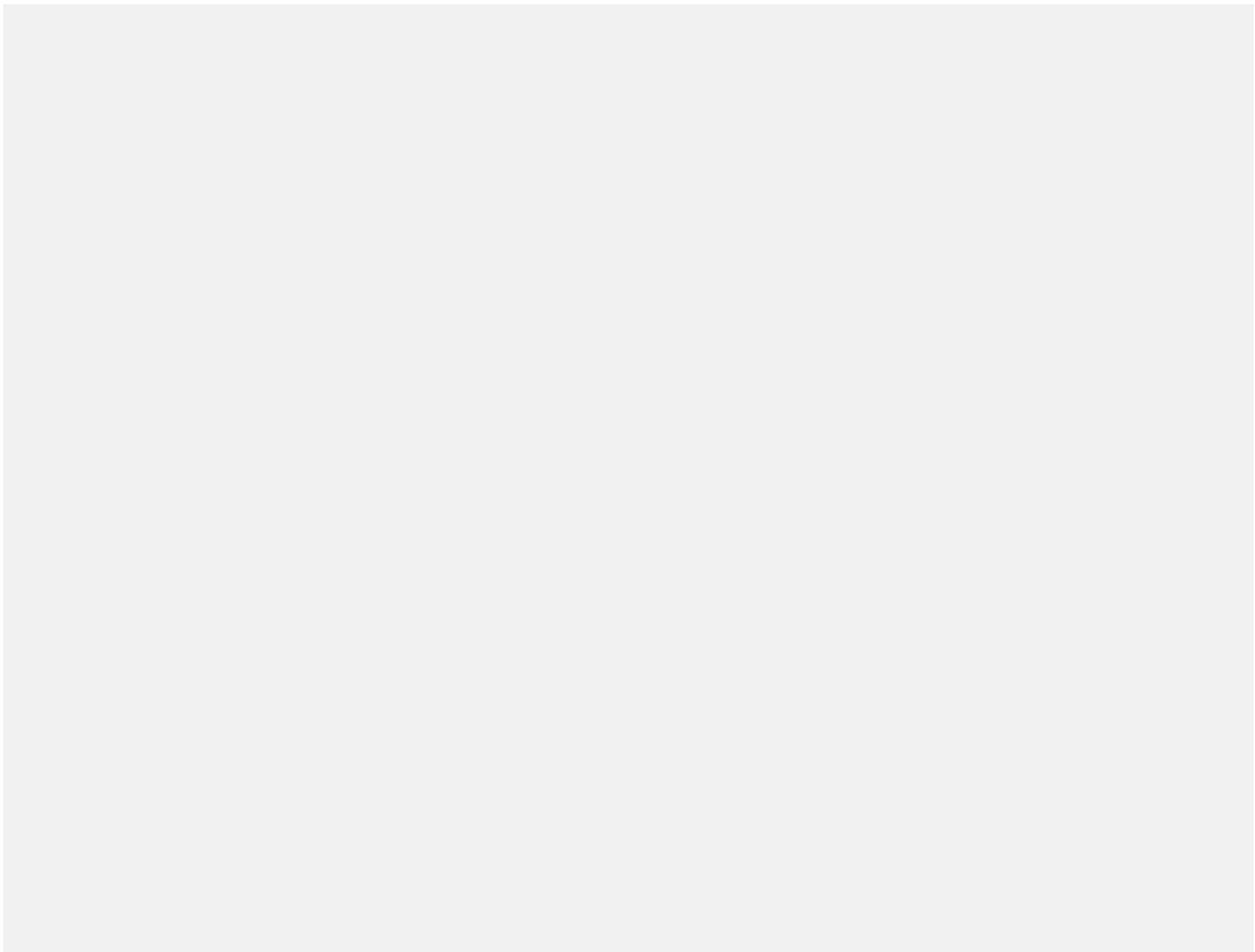
Михаил Шардин @empenoso

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

[Подписаться](#)



[Сайт](#) [Сайт](#) [Github](#)



 Комментарии 9

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



rssdev10

22 часа назад

Почему въехав по «визе талантов» в США я с радостью вернулся в Россию

 Средний

 32 мин

 34К

Мнение

 +170

 108

 491

**melnik909**

18 часов назад

Вы не знаете CSS. Мои вопросы о CSS с ответами. Часть 2

Средний 7 мин 2K

Обзор

+38

30

1

**DAN_SEA**

16 часов назад

Генерация случайных чисел

Средний 10 мин 2.3K

Обзор

+31

23

33

**OrkBiotechnologist**

22 часа назад

VPS за 139 рублей — дом для вашего резюме на основе Hugo

Простой 7 мин 8.2K

Тutorial

+28

45

12

**nebykoff_anton**

23 часа назад

До и после: оптимизация изображений для Lighthouse и не только

Простой 8 мин 941

Tutorial

+28

24

2

**Корченіу**

23 часа назад

Имитатор касаний. Ч2: Железная часть

 Средний  9 мин  1.6K

Кейс

 +24  20  12

 tertiumnon
17 часов назад

Минимум книг, которые нужно прочитать начинающему или продолжающему свою кривую обучения программисту

 Простой  3 мин  8.2K

Обзор

 +16  180  16

 PatientZero
1 час назад

Пишем стек TCP/IP с нуля: Ethernet, ARP, IPv4 и ICMPv4

 Простой  13 мин  738

Тutorial Перевод

 +15  21  1

 lbkanter
46 минут назад

Бэкдор Auto-color: разбор угрозы, технический анализ и способы защиты

 Средний  4 мин  210

Обзор

 +13  4  2

 alexander-shustanov
18 часов назад

В поисках идеального Database-клиента для IDE: Amplicode выбирает DBeaver

Простой 6 мин 2.5K

+13

13

7

Бегущий по лезвию ИИ — 2025: сезон футурологии на Хабре с крутыми призами

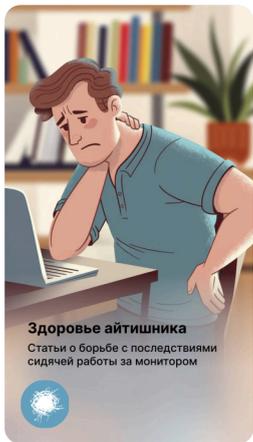
Турбо

Показать еще

ИСТОРИИ



Как расти на работе?



Здоровье айтишника



Угадайте будущее в новом сезоне



С праздником весны!

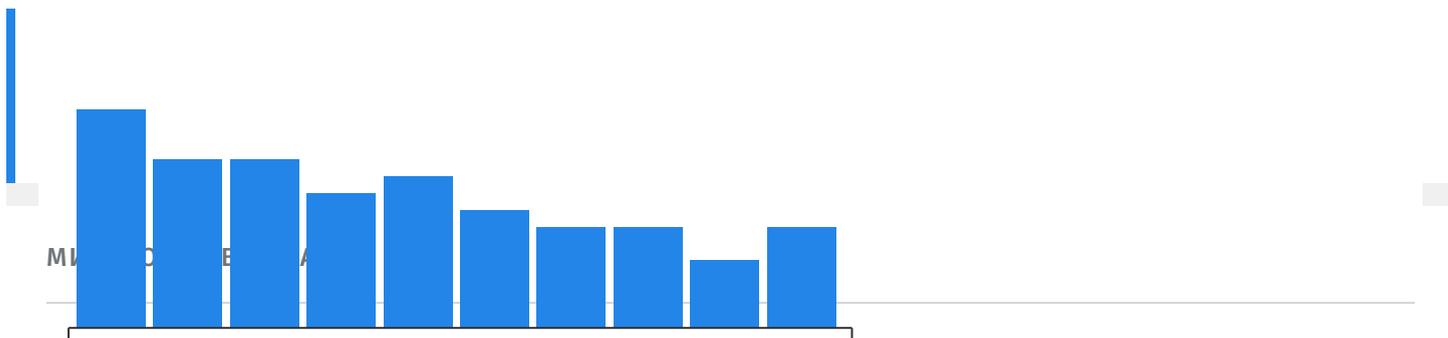


Всегда котов!

СРЕДНЯЯ ЗАРПЛАТА В IT

212 662 ₺/мес.

— средняя зарплата во всех IT-специализациях по данным из 33 637 анкет, за 1-ое пол. 2025 года. Проверьте «в рынке» ли ваша зарплата или нет!



Турбо

Как избежать ошибок при бизнес-миграции и сохранить данные?



ИТ-КОМИКС

Девушка с розовыми волосами и Слизень на планете Рбах



Событие

Как писать про здоровье на Хабре? Расскажем на вебинаре

РАБОТА

Data Scientist

55 вакансий

Django разработчик

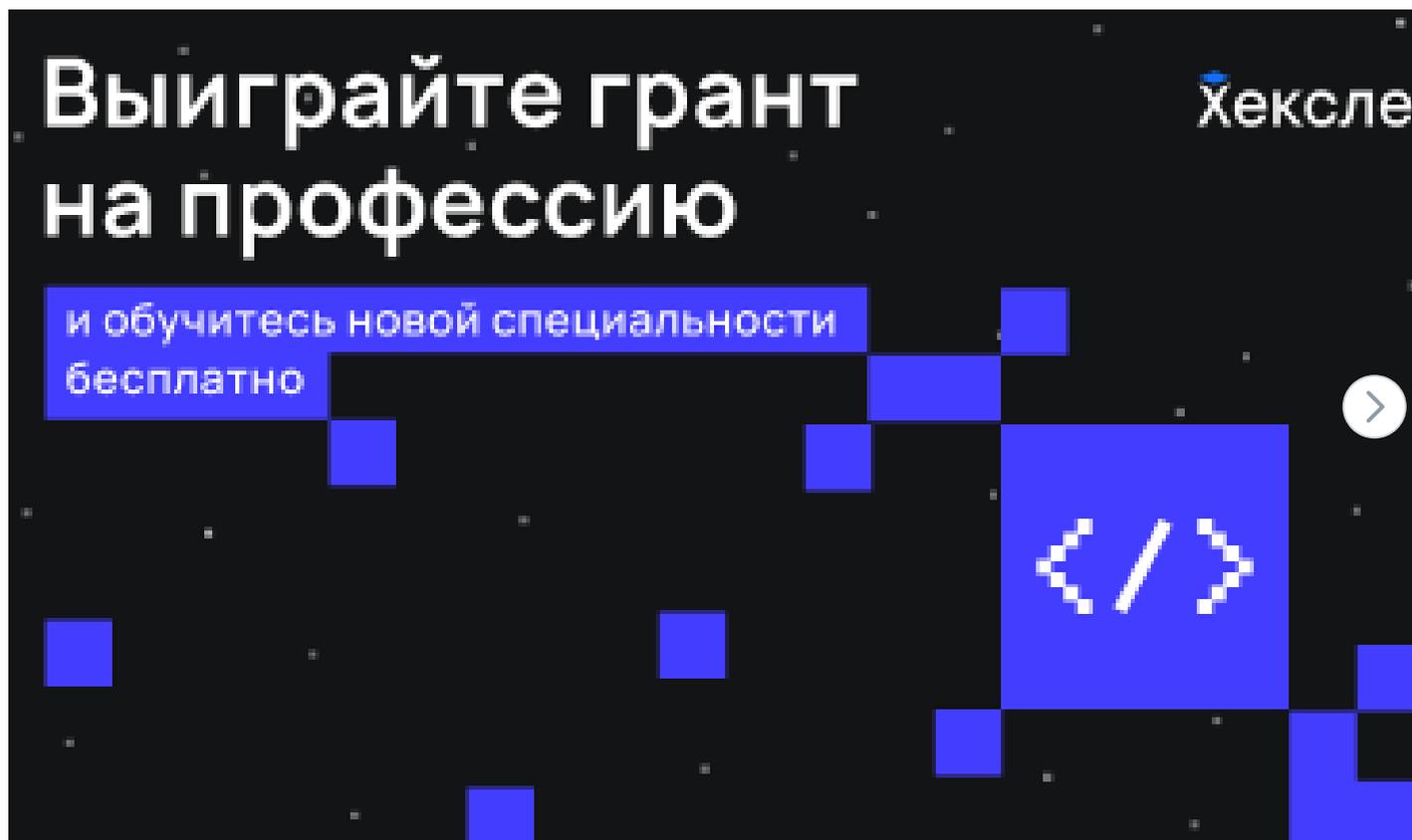
20 вакансий

Python разработчик

65 вакансий

[Все вакансии](#)

БЛИЖАЙШИЕ СОБЫТИЯ



17 февраля – 24 марта

Конкурс «Снежный код» от Хекслета. Три гранта на бесплатное 10-месячное обучение

Онлайн

Разработка

[Больше событий в календаре](#)

Хабр



🌐 [Настройка языка](#)

[Техническая поддержка](#)

© 2006–2025, Habr