

Хабр



КАК СТАТЬ АВТОРОМ



Войти

Все важнейшие
IT-события здесь

5

6

7

8

9

10

11



12

13

14

15

16

17

18



empenoso

27 янв в 03:25

Как автоматизировать свою переписку с девушкой в Telegram при помощи Python



Простой



6 мин



33К

Мессенджеры*, API*, Python*, Node.JS*

Кейс

Бывает так: девушка ждет внимания, а вы увлечены работой, хобби или ещё чем-то, и забываете о ней. Установка отложенных сообщений — идеальный компромисс. Эта статья не о создании чат-бота, который будет "любить" за вас. Речь о ненавязчивом напоминании для самого себя и создании точки входа для настоящего диалога.

Зачем это всё?

Автоматизация позволяет отправлять заранее подготовленные, теплые и заботливые сообщения без излишнего пафоса в случайное время в заданном диапазоне. Представьте, "утро начинается не с кофе", а с вашего сообщения. Вот пример списка фраз для утреннего приветствия:

```
greetings = [  
    "Доброе утро! Как ты себя чувствуешь сегодня?",  
    "Привет, как настроение? Готовность к новому дню?",  
    "Доброе утро! Что нового на сегодня?",  
    "Утро доброе! Как у тебя всё происходит с утра?",
```

```
"Доброе утро, как ты себя сегодня чувствуешь?",  
"Привет! Как ты, как сны?",  
"Как твои дела, готовность к этому дню?",  
"Доброе утро! Как настроение, что планируешь на день?",  
"Здравствуй! Как ты себя чувствуешь после сна?"
```

```
]
```

Даже пара теплых слов не оставит девушку равнодушной, и это перейдёт уже в настоящий разговор.

Автоматизация не заменяет заботу, а помогает ее проявить, напоминая о важном человеке, даже если вы очень заняты.

Puppeteer: почему это не лучший вариант

Изначально я пытался автоматизировать переписку через веб-версию Telegram, используя [Puppeteer](#). Идея казалась простой: скрипт заходит в Telegram, отправляет сообщения.

Был создан даже модуль авторизации через QR-код с двухфакторной аутентификацией. На практике же возникли сложности: изменения разметки веб-версии Telegram, сложности с реализацией отложенной отправки сообщений. Но для WhatsApp, например, есть подобная уже готовая [JavaScript-библиотека для автоматизации задач whatsapp-web.js](#).

```

JS authjs
old > JS authjs > ...
1 // auth.js
2
3 function delay(ms) {
4   return new Promise(resolve => setTimeout(resolve, ms));
5 }
6
7 // Авторизация в Telegram
8 async function authorize(page) {
9   console.log('Переход на страницу авторизации Telegram...');
10  await page.goto('https://web.telegram.org');
11
12  // Ожидание появления QR-кода (с таймаутом)
13  console.log('Ожидание QR-кода... (до 20 секунд)');
14  try {
15    await page.waitForSelector('#auth-qr-form > div > div > div > div.qr-container > svg', {
16      timeout: 20000
17    });
18    console.log('QR появился');
19  } catch (error) {
20    console.warn('Контейнер QR-кода не найден в течение 20 секунд. Возможно, загрузка займет больше времени. Продолжение...');
21  }
22
23  await delay(10000);
24
25  // Продолжить на русском кнопка
26  // await page.click('#auth-pages > div > div.tabs-container.auth-pages__container > div.tabs-tab.page-signQR.active > div > div.input-wrapper >
27  await page.click('#auth-qr-form > div > button:nth-child(5)');
28
29  // Ожидание ручной авторизации (1 минута)
30  console.log('Пожалуйста, отсканируйте QR-код и введите пароль (ожидание 1 минуты)...');
31  await delay(60000);
32
33  // Для тестов в консоли Хрома:
34  // document.querySelector('#StoryToggler > div:nth-child(3) > div > img');
35
36  // Проверка успешной авторизации
37  try {
38    // await page.waitForSelector('#column-left > div > div > div.sidebar-header.can-have-forum > div.sidebar-header__btn-container > div.btn-ico
39    await page.waitForSelector('#StoryToggler > div:nth-child(3) > div > img', {
40      timeout: 20000
41    });
42  } catch (error) {
43    console.warn('Не удалось найти элемент, подтверждающий авторизацию. Возможно, процесс еще не завершен. Продолжение...');
44  }
45
46  // Завершение
47  console.log('Авторизация завершена');
48  return true;
49 }
50
51 // Запуск
52 (async () => {
53   const page = await puppeteer.launch({
54     headless: false,
55     args: ['--no-sandbox', '--disable-setuid-sandbox', '--disable-dev-shm-usage']
56   });
57   const auth = await authorize(page);
58   if (auth) {
59     console.log('Успешная авторизация!');
60   } else {
61     console.log('Авторизация не удалась');
62   }
63   await page.close();
64 })();

```

Этот опыт показал: не всегда нужно изобретать велосипед. Puppeteer — отличный инструмент, но в этом случае он не оправдал себя. К счастью, существует более эффективное решение — библиотека [Telethon на Python](#).

Telethon: находка для автоматизации

Telethon упрощает взаимодействие с Telegram. Забудьте о сложных настройках Puppeteer. Telethon — ваш личный программный ассистент.

Как это работает?

1. Регистрация приложения на my.telegram.org для получения `api_id` и `api_hash`.
2. Авторизация своего скрипта с помощью номера телефона и кода подтверждения (если установлен).



Преимущества Telethon:

- Простая установка (`pip install telethon`).
- Дополнительные возможности: чтение диалогов, создание чатов и каналов, скачивание медиафайлов.

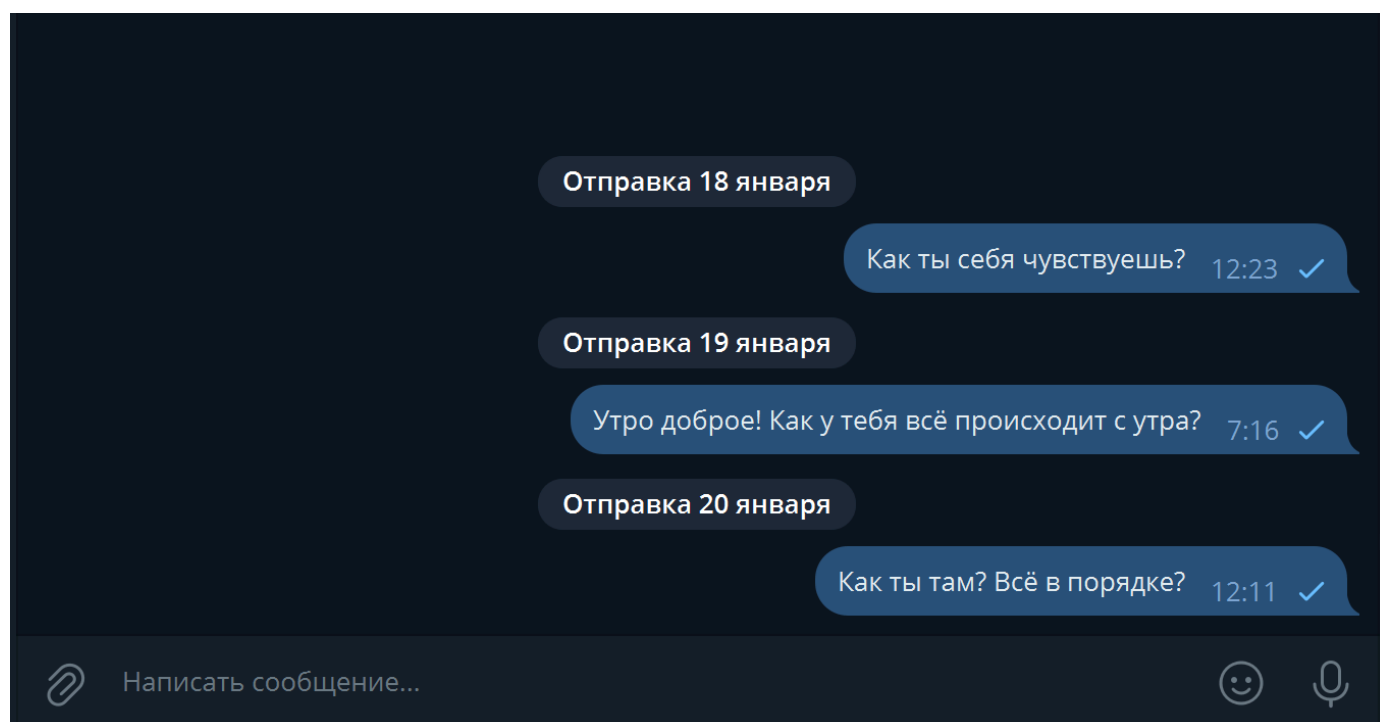
С Telethon можно создавать сценарии общения. Главное - не переусердствовать, чтобы не попасть под бан аккаунта в Telegram.

Как избежать бана

Telegram не любит автоматизацию и тщательно следит за подозрительной активностью. Поэтому, если вы не хотите потерять доступ к своему аккаунту, нужно быть предельно аккуратным.

Чего точно не стоит делать? Массовая рассылка одинаковых сообщений. Но вам это не понадобится, ведь цель скрипта — общение с одним человеком, а не спам.

Как обезопасить себя? Используйте скрипт только для личного общения. Это уменьшает риск привлечь внимание системы.



Результат работы скрипта

Соблюдайте [лимиты API](#), описанные в статье на Хабре. Например, не отправляйте больше 10 медиасообщений за 5 минут - даже с лёгкой анимацией это вызовет ошибку.

Скрипт: первая версия

```
# pip install telethon python-dotenv pytz

import asyncio
from telethon import TelegramClient
from datetime import datetime, timedelta, time
import random
import pytz
from dotenv import load_dotenv
import os

# Загрузка конфигурации
load_dotenv()
api_id = os.getenv("api_id")
api_hash = os.getenv("api_hash")
session_name = 'session_name'

# Таймзона и сообщения
ekb_timezone = pytz.timezone('Asia/Yekaterinburg')
messages = ["Доброе утро! Как твои дела?", "Привет! Надеюсь, у тебя всё отлично!"]

async def schedule_messages(client, user, messages, days):
    for day in range(days):
        send_time = ekb_timezone.localize(
            datetime.combine(datetime.now().date() + timedelta(days=day + 1),
                             time(random.randint(8, 10), random.randint(0, 59)))
        )
        message = random.choice(messages)
        await client.send_message(user, message, schedule=send_time.astimezone(pytz.UTC))
        print(f"Сообщение запланировано: {message} на {send_time}")

async def main():
    client = TelegramClient(session_name, api_id, api_hash)
    await client.start()
    user = await client.get_entity("USERNAME")
    await schedule_messages(client, user, messages, 7)
    await client.disconnect()
```

```
asyncio.run(main())
```

Подготовка:

1. Установите необходимые библиотеки: `pip install telethon python-dotenv pytz` .
2. Создайте файл `.env` и добавьте туда `api_id` и `api_hash` , полученные на my.telegram.org.
3. Замените `USERNAME` на имя пользователя или номер телефона.

Полная версия скрипта

Полная версия скрипта автоматизации своей переписки с девушкой в Telegram представлена ниже. Скрипт подключается к аккаунту через API, загружает данные аутентификации из `.env` , и планирует отправку случайных сообщений (утренних и дневных) для заданных пользователей в определённое время:

```
# pip install telethon python-dotenv pytz

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

import asyncio
from telethon import TelegramClient, utils
from datetime import datetime, timedelta, time
import random
from dotenv import load_dotenv
import os
import pytz

# Загрузка переменных окружения из .env
load_dotenv()
api_id = os.getenv("api_id") # Ваш API ID
api_hash = os.getenv("api_hash") # Ваш API Hash
session_name = 'nodejs_session' # Название сессии

# Сообщения
```

```

greetings = [
    "Доброе утро! Как ты себя чувствуешь сегодня?",
    "Привет, как настроение? Готовность к новому дню?",
    "Доброе утро! Что нового на сегодня?",
    "Утро доброе! Как у тебя всё происходит с утра?",
    "Доброе утро, как ты себя сегодня чувствуешь?",
    "Привет! Как ты, как sny?",
    "Как твои дела, готовность к этому дню?",
    "Доброе утро! Как настроение, что планируешь на день?",
    "Здравствуй! Как ты себя чувствуешь после сна?"
]

supportive_messages = [
    "Как ты там? Всё в порядке?",
    "Как проходит твой день?",
    "Надеюсь, день не слишком утомительный для тебя!",
    "Как настроение? Всё ли хорошо?",
    "Не забывай делать перерывы, если устала!",
    "Как ты? Чем занимаешься сейчас?",
    "Если что-то нужно, не стесняйся мне писать.",
    "Надеюсь, день приносит тебе только положительные моменты!",
    "Как ты себя чувствуешь? Хочешь поговорить?"
]

# Таймзона Екатеринбурга
ekb_timezone = pytz.timezone('Asia/Yekaterinburg')

async def schedule_messages(client, user, messages, start_hour, end_hour, days, description):
    """Планирует сообщения для заданного пользователя на несколько дней."""
    for day in range(days):
        # Вычисляем дату и время в Екатеринбурге
        now_ekb = datetime.now(ekb_timezone)
        schedule_date_ekb = now_ekb.date() + timedelta(days=day + 1)
        schedule_time_ekb = ekb_timezone.localize(datetime.combine(schedule_date_ekb, time(
            start_hour, 0, 0)))

        # Конвертируем время в UTC для Telethon
        schedule_time_utc = schedule_time_ekb.astimezone(pytz.UTC)

        message_text = random.choice(messages)

        try:
            await client.send_message(user, message_text, schedule=schedule_time_utc)
            print(f"✅ Отложенное сообщение отправлено для {description} на {schedule_time_utc}")
        except Exception as e:
            print(f"❌ Ошибка при отправке сообщения: {e}")

```



```

        print(f"❌ Ошибка при отправке сообщения для {description}: {e}")

    async def main():
        client = TelegramClient(session_name, api_id, api_hash)
        await client.start()

        print("🚀 Запуск программы...")
        try:
            tasks = []

            user_Name1 = await client.get_entity("empenoso") # имя пользователя
            tasks.append(schedule_messages(client, user_Name1, greetings, 7, 7, 3, "пользователь"))
            tasks.append(schedule_messages(client, user_Name1, supportive_messages, 12, 12, 3, "пользователь"))

            user_Name2 = await client.get_input_entity("+7912XXXXX") # Непосредственно номер
            tasks.append(schedule_messages(client, user_Name2, greetings, 7, 7, 3, "пользователь"))

            await asyncio.gather(*tasks)
            print("🎉 Все сообщения успешно настроены.")
        except Exception as e:
            print(f"❌ Ошибка: {e}")
        finally:
            await client.disconnect()
            print("🔌 Соединение с Telegram завершено.")

if __name__ == '__main__':
    asyncio.run(main())

```

Часовой пояс Екатеринбурга используется потому что я живу в Перми.

Заключение

Данная автоматизация — инструмент для выражения эмоций. Скрипт не заменит живое общение, но поможет не забыть проявить внимание. Отложенные сообщения — это лишь начало. Не переставайте общаться лично. Автоматизируйте с душой, искренностью и чувством меры!

Автор: Михаил Шардин

 [Моя онлайн-визитка](#)

Telegram «Умный Дом Инвестора»

27 января 2025 г.

Теги: telethon, puppeteer, девушки

Хабы: Мессенджеры, API, Python, Node.JS

Редакторский дайджест

Присылаем лучшие статьи раз в месяц



183

Карма

87.1

Рейтинг

Михаил Шардин @empenoso

Автоматизация / Данные / Финансы / Умные дома

Подписаться



[Сайт](#) [Сайт](#) [Github](#)

 Комментарии 184

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



34K

Мнение

**+169**

108



492

**melnik909**

18 часов назад

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



Средний



7 мин



2K

Обзор

**+38**

31



1

**DAN_SEA**

16 часов назад

Генерация случайных чисел

 Средний  10 мин  2.3K

Обзор

 +31

 23

 33



OrkBiotecnologist

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

Тutorial

 +28

 24

 2



Корcheniy

23 часа назад

Имитатор касаний. Ч2: Железная часть

 Средний  9 мин  1.6K

Кейс

 +24

 20

 12



PatientZero

1 час назад

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

 Простой  13 мин  751

Тutorial

Перевод

 +16 21 1

tertiumnon

17 часов назад

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



Простой



3 мин



8.2K

[Обзор](#) +16 182 17

lbkanter

48 минут назад

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



Средний



4 мин



221

[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

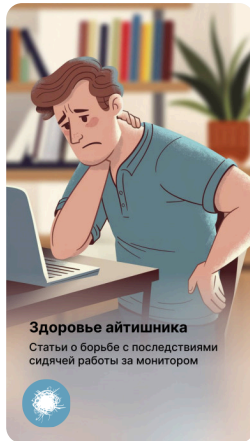
Бегущий по лезвию ИИ — 2025: сезон футурологии на Хабре с крутыми призами

[Турбо](#)[Показать еще](#)

ИСТОРИИ



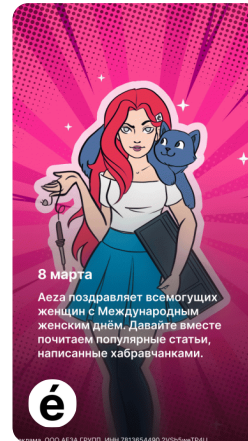
Как расти на работе?



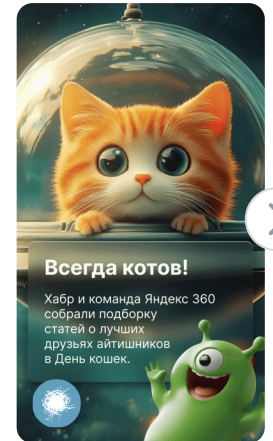
Здоровье айтишника



Угадайте будущее в новом сезоне



С праздником весны!

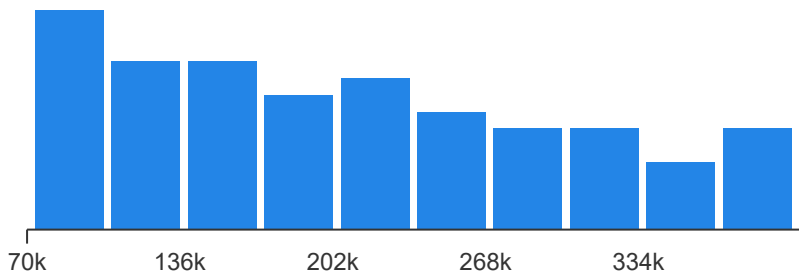


Всегда котов!

СРЕДНЯЯ ЗАРПЛАТА В IT

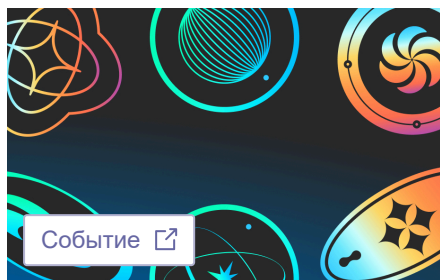
212 662 ₽/мес.

— 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!

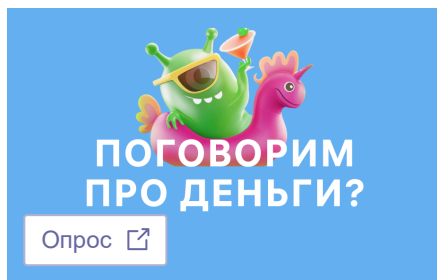


[Проверить свою зарплату](#)

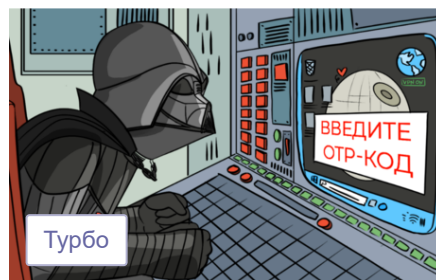
МИНУТОЧКУ ВНИМАНИЯ



Как писать про здоровье на
Хабре? Расскажем на вебинаре



Как айтишники подходят к
финансовому планированию?



«Люк, я твой фактор!» —
защищаем подключения с MFA

РАБОТА

Python разработчик

65 вакансий

Data Scientist

55 вакансий

Node.js разработчик

56 вакансий

Django разработчик

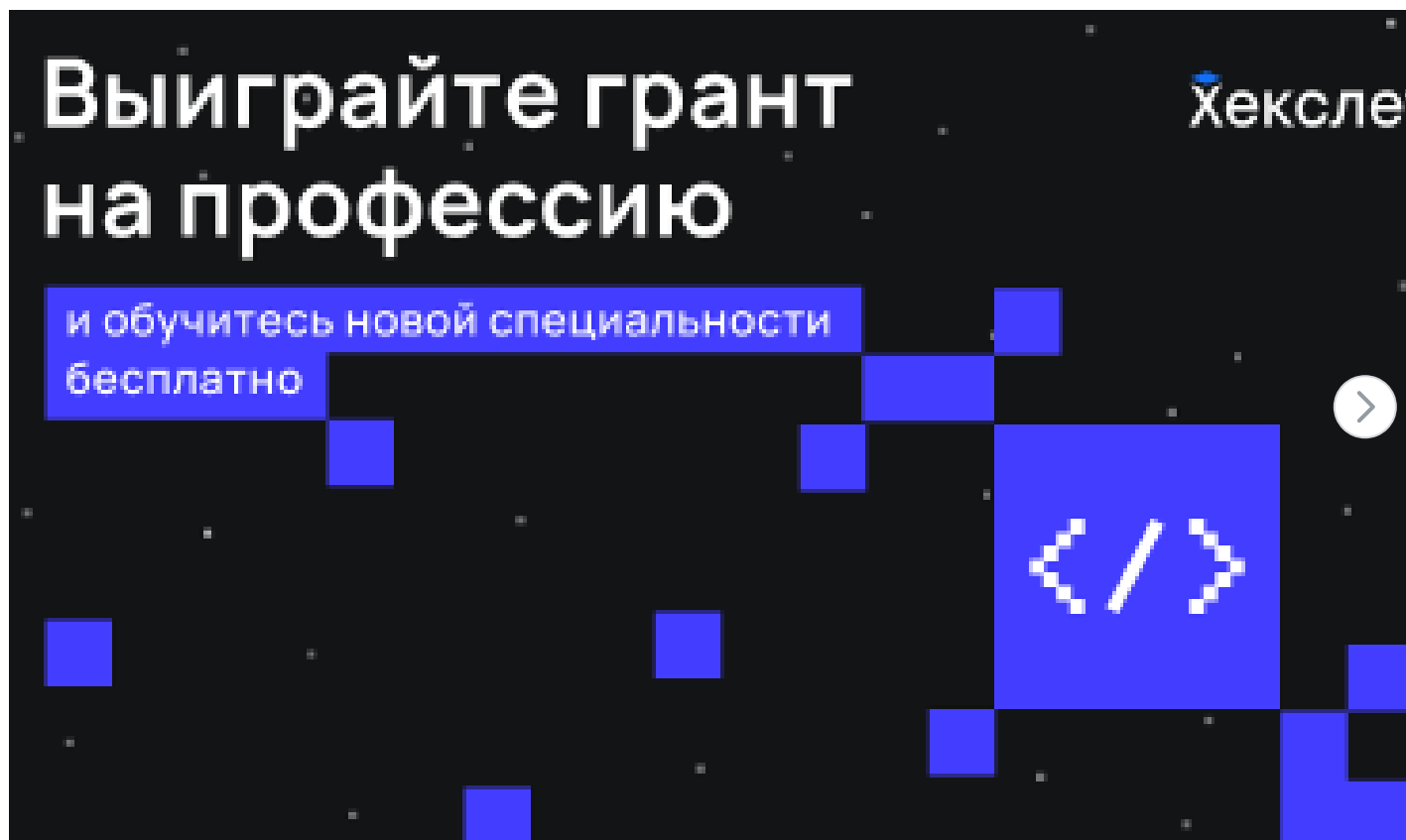
20 вакансий

JavaScript разработчик

134 вакансии

[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