



empenoso

14 апр в 05:17

# Как создать множество документов на основе единого шаблона при помощи скрипта внутри гугл таблицы

Средний 6 мин 1.6K

Open source\*, JavaScript\*, API\*, Google API\*

Кейс

Иногда бывает необходимо создать множество повторяющихся документов, которые отличаются лишь номером, датой и ещё парой текстовых строк. Очень грустно тратить на их создание своё время - ведь требуется совершить множество одинаково повторяющихся действий. Ещё можно понять затраты времени на создание 5 документов, но если их надо создать, например 500 штук?

А ещё эта задача может усложниться за счёт необходимости вставки каких либо данных, которые надо сначала найти в интернете. Например, вставлять данные о погоде на дату создания документа.

К счастью, гугл таблицы могут упростить процесс создания однотипных копий с помощью шаблона. В этой статье покажу как можно создать неограниченное число копий на основе одного шаблона с помощью гугл скрипта внутри гугл таблицы.

## Создайте шаблон для вашего документа

Первым шагом является создание шаблона. Пусть образцом для этой статьи станет протокол сварки. Его можно создать заново или перенести в гугл таблицы. [Вот ссылка на этот пример.](#)



+1



33



6

|    | A                                 | B                 | C           |
|----|-----------------------------------|-------------------|-------------|
| 1  | Погодные условия                  | Ниже              |             |
| 2  | Окружающая температура, °C        | ТЕМПЕРАТУРА       |             |
| 3  | /=====/                           |                   |             |
| 4  | Оператор                          | СВАРЩИК           |             |
| 5  | Номер шва                         | НОМЕР             |             |
| 6  | Регистратор                       | Nowatech          |             |
| 7  | Серийный номер регистратора       | 15795             |             |
| 8  | Сварочный аппарат                 | Nowatech          |             |
| 9  | Серийный номер                    | 15795             |             |
| 10 | Версия                            | 2                 |             |
| 11 | Порог давления T1 T2              | 2.0 бар           |             |
| 12 | Порог давления T2 T3              | 0.5 бар           |             |
| 13 | /=====/                           |                   |             |
| 14 | Диаметр трубы, мм                 | 110               |             |
| 15 | SDR                               | 11                |             |
| 16 | Толщина стенки, мм                | 10                |             |
| 17 | Материал                          | 100               |             |
| 18 | /=====/                           |                   |             |
| 19 | Параметр                          | Заданный          | Фактический |
| 20 | /=====/                           |                   |             |
| 21 | Давление движения, бар            | --                |             |
| 22 | Высота грата, мм                  | 3.0               |             |
| 23 | Температура, гр                   | 220.0             |             |
| 24 | Давление сварки P1(P1), бар + ДД  | 38.6 (19.3)       |             |
| 25 | Время нагрева T1, сек             | --                |             |
| 26 | Время нагрева T2, сек             | 286               |             |
| 27 | Время паузы T3, сек               | 13                |             |
| 28 | Время увеличения давления T4, сек | 16                |             |
| 29 | Давление сварки P5(P5), бар + ДД  | 38.6 (19.3)       |             |
| 30 | Время охлаждения T5, мин(сек)     | 34(2040)          |             |
| 31 | Общее время, мин (сек)            | 40 (2400)         |             |
| 32 | /=====/                           |                   |             |
| 33 | Начало записи                     | 6 октября 2022 г. | 08:42       |
| 34 | Конец записи                      | 6 октября 2022 г. | 09:30       |
| 35 |                                   |                   |             |
| 36 |                                   |                   |             |

Шаблон протокола на одну страницу A4. Красным выделены поля, которые надо изменять

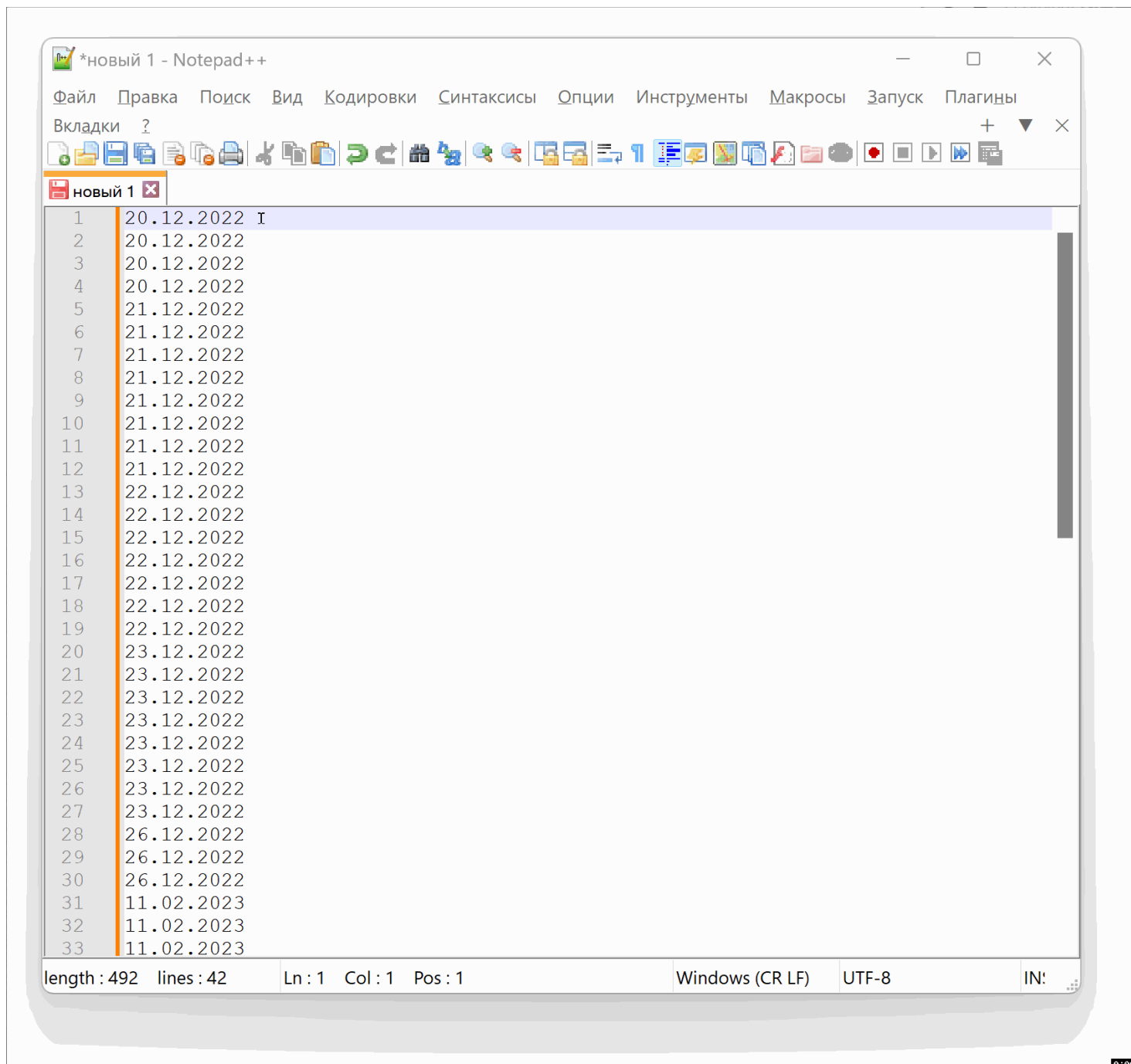
Красным выделены поля, которые надо изменять в процессе создания.

Даты были заданы заранее, как и их порядковые номера. Даты для работы скрипта надо записать в виде массива:

```
const values = ["2022-12-21", "2022-12-21", "2022-12-21", "2022-12-21", "2022-12-21", "2022-12-21", "2022-12-22", "2022-12-22"];
```

Для столбца из дат можно сделать это, например, через макросы в Notepad++:

(знаете способ лучше - напишите в комментариях к статье)



Запись макроса и его дальнейший запуск

На гифке показано как сделать запись макроса в Notepad++ и его дальнейший запуск до конца файла, для того чтобы сделать из обычного столбца дат массив в гугл скриптах.

## Гугл скрипты в помощь

Google Apps Script — это простой язык сценариев, который использует синтаксис JavaScript.

Для получения погоды (а это одно из полей, которые требует протокол) можно воспользоваться бесплатным [Free Weather API](#), которое даже не требует получения ключа для работы с ним.

Координаты местности задаются через параметры `latitude=58.08&longitude=55.76` :

```
function temperature(date) { //получаем погоду
  // date = "2022-10-04"
  // console.log(`temperature. date = ${date}`)
  const url = `https://archive-api.open-meteo.com/v1/archive?latitude=58.08&longitude=55
CacheServis(url);
  try {
    const response = UrlFetchApp.fetch(url)
    const json = JSON.parse(response.getContentText());
    const temperature_2m_mean = json.daily.temperature_2m_mean[0]
    // console.log(`Погода по координатам на ${date}: ${temperature_2m_mean}°C.`)
    const precipitation_sum = json.daily.precipitation_sum[0]
    // console.log(`Погода по координатам на ${date}: ${precipitation_sum} мм.`)
    const windspeed_10m_max = json.daily.windspeed_10m_max[0]
    // console.log(`Погода по координатам на ${date}: ${windspeed_10m_max} км/ч.`)
    // console.log(`Погода по координатам на ${date}: ${temperature_2m_mean}°C, ${prec
    return `${temperature_2m_mean}°C\n${precipitation_sum} мм осадков\nветер ${windspe
  } catch (error) {
    console.log(`temperature. Ошибка на ${date}: ${error}.`)
    return ""
  }
}
```

Поскольку даты повторяются пришлось воспользоваться сервисом кеширования, для того чтобы сократить время работы этой погодной функции:

```
function CacheServis(url) { //кэширование результата запроса по url
  var cache = CacheService.getScriptCache();
  var cached = cache.get(url);
  if (cached != null) {
    // Logger.log("CacheServis.\nУже было записано для " + url);
    return cached;
  }
  try {
    var result = UrlFetchApp.fetch(url);
  } catch (error) {
    Logger.log("CacheServis.\nОшибка обращения при кэшировании по url " + url + "\n\n")
  }
}
```

```

        return ""
    }
    var contents = result.getContentText();
    cache.put(url, contents, 21600); // cache for 6 hours
    Logger.log("CacheServis.\nЗаписали по новой для " + url);
    return contents;
}

```

Далее самая важная часть гугл скрипта - создание множества листов на основе одного шаблона:

```

function index() {
    var startTime = new Date();

    const values = ["2022-12-21", "2022-12-21", "2022-12-21", "2022-12-21", "2022-12-21",

    let n = 0
    let time = 7;

    values.forEach((date, index) => {
        if (index === 0 || date !== values[index - 1]) {
            time = 7;
        } else {
            time += 1;
        }
        n += 1
        console.log(`Выполняю копирование №${n} для ${date}.`)
        duplicate(date, n, time);

    var endTime = new Date();
    var duration = (endTime - startTime) / 60000;
    Logger.log("Время выполнения скрипта: " + duration.toFixed(2) + " минут");
    });
}

```

И сам код, который создаёт дубликаты вкладок и меняет их содержимое:

```

function duplicate(date, n, time) {

    var ss = SpreadsheetApp.getActiveSpreadsheet();
    var lastTabName = ss.getSheets().pop().getSheetName();
    var sheet = ss.getSheetByName('Шаблон').copyTo(ss);

```

```
sheet.setName(n);
sheet.getRange("B2").setValue(temperature(date));
sheet.getRange("B5").setValue(n);
sheet.getRange("B33").setValue(date);
sheet.getRange("B34").setValue(date);
sheet.getRange("C33").setValue(`${time}:${Math.floor(Math.random() * 3) + 1}:00`);
sheet.getRange("C34").setValue(`${time+1}:${Math.floor(Math.random() * 6) + 4}:00`);

// ss.setActiveSheet(sheet);
}
```

Я знаю что этот код написан не самым оптимальным образом и вряд ли стоит рассчитывать на скорость работы, но этот код может за 6 бесплатных минут работы гугл скрипта (а именно такое ограничение любого гугл аккаунта), создать около 300 дубликатов вкладки. А ещё обращение к внешнему сервису за погодой сильно замедляет его работу.

## Как пользоваться?

Для того чтобы запустить этот скрипт из гугл таблицы и предоставить все необходимые разрешения для его работы надо выполнить несколько шагов:

### Шаг 1. Откройте таблицу Google `Протоколы_шаблон` и скрипт :

- Войдите в свою учетную запись Google и откройте электронную таблицу, содержащую этот скрипт.

### Шаг 2. Создайте собственную копию этой гугл таблицы примера:

- Нажмите на меню «Файл», затем выберите «Создать копию...».
- В окне «Создать копию» введите имя для своей копии электронной таблицы и выберите место для нее.
- Нажмите «ОК», чтобы создать копию.
- Ваша новая копия электронной таблицы откроется в новой вкладке, теперь вы можете начать работать с ней.
- Все данные и форматирование исходной электронной таблицы будут перенесены в новую копию.
- Если вы сделали копию общей электронной таблицы, у вас будут свои собственные отдельные данные, и вы не испортите данные исходной электронной таблицы, которой с вами поделились.

### Шаг 3. Откройте редактор в вашей таблице скриптов:

- Щелкните меню «Расширения», затем выберите «Apps Script».
- Это откроет редактор скриптов в новом окне.

#### Шаг 4. Запустите скрипт:

- В редакторе сценариев выберите вкладку `replication`, а в ней функцию `index` и щелкните кнопку «Выполнить».
- Вам потребуется авторизовать сценарий, нажав «Просмотреть разрешения», а затем нажав «Разрешить», чтобы предоставить сценарию разрешение на доступ к информации вашей учетной записи для того, чтобы начать использовать этот скрипт. Вы в любое время можете посмотреть список выданных вами [разрешений на специальной странице](#) и в один клик их отозвать.

#### Шаг 5. Проверьте результат:

- Если сценарий выполняется успешно, выходные данные функции будут отображаться на вкладке «Журнал выполнения» в нижней части окна редактора сценариев.

| Журнал выполнения |            | ✕                                       |
|-------------------|------------|---|
| 15:03:55          | Примечание | Выполнение начато                       |
| 15:03:56          | Информация | Выполняю копирование №1 для 2022-12-21. |
| 15:03:57          | Информация | Время выполнения скрипта: 0.02 минут    |
| 15:03:57          | Информация | Выполняю копирование №2 для 2022-12-21. |

Выполнение работы скрипта

- Также во время работы скрипта вы можете открыть таблицу и наблюдать как появляются новые вкладки.

**Вот и всё!** Следуя этим простым шагам, вы можете запустить этот гугл скрипт из своей электронной таблицы и убедиться, что всё работает. После этого можно скачать готовую книгу

<https://habr.com/ru/articles/728840/> | Михаил Шардин, <https://shardin.name/>

Страница 7 из 13

электронной таблицы и объединить, но все работает. После этого можно скачать готовую книгу, как файл Экселя или pdf и передать её в дальнейшую работу.

|    |                                  |             |  |
|----|----------------------------------|-------------|--|
| 25 | Время нагрева T1,сек             | --          |  |
| 26 | Время нагрева T2,сек             | 286         |  |
| 27 | Время паузы T3,сек               | 13          |  |
| 28 | Время увеличения давления T4,сек | 16          |  |
| 29 | Давление сварки P5(P5) бар ± ПП  | 38.6 (10.3) |  |

+ ≡ Шаблон ▾

1 ▾ 2 ▾ 3 ▾ 4 ▾ 5 ▾ 6 ▾ 7 ▾ 8 ▾

Вновь созданные вкладки

**Дополнительный шаг.** При помощи функции `deleteRedundantSheets` можно удалить все автоматически созданные листы, для того чтобы подготовить этот шаблон уже с другими параметрами.

## Что в итоге

Использование связки гугл таблица + гугл скрипт для создания однотипных документов на основе единого шаблона является действенным способом автоматизации создания документов.

Этот процесс поможет сэкономить время и повысить производительность. С помощью пошагового руководства, представленного в этой статье, вы сможете легко настроить этот процесс самостоятельно, используя связку гугл таблица + гугл скрипт.

Настраивая свои шаблоны и сценарии в соответствии со своими потребностями, вы сможете создавать свои собственные профессионально выглядящие документы.

После того, как все будет работать хорошо, вы сможете воспользоваться преимуществами оптимизированного процесса создания документов и сосредоточиться на самом важном — легком и эффективном достижении ваших целей.

Автор: [Михаил Шардин](#),

14 апреля 2023 г.

Только зарегистрированные пользователи могут участвовать в опросе. [Войдите](#), пожалуйста.

## Используете гугл скрипты в работе?

25% Да



50% Нет

8

25% Не знаю что это

4

Проголосовали 16 пользователей. Воздержавшихся нет.

**Теги:** [шаблон](#), [документ](#), [гугл скрипт](#), [таблицы](#)

**Хабы:** [Open source](#), [JavaScript](#), [API](#), [Google API](#)

## Редакторский дайджест



Присылаем лучшие статьи раз в месяц

**128 9.2**

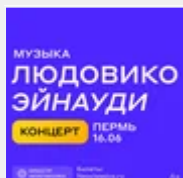
КармаРейтинг

**Михаил Шардин** [@empenoso](#)

Разработчик

[Сайт](#)

### Реклама



neoclassica.ru РЕКЛАМА • 6+

## «Музыка Ludovico Einaudi» - Концерт Оркестра Неоклассика

Лучшие произведения итальянского композитора в исполнении Оркестра Неоклассика

[Комментарии 6](#)

## Публикации

[ЛУЧШИЕ ЗА СУТКИ](#)[ПОХОЖИЕ](#)

nnееео



21 час назад

## Убийство разработки: опыт Selectel

🕒 8 мин 👁 13K

Обзор

💎 +63

🔖 31

💬 18



ru\_vds

22 часа назад

## Предполётные испытания космического сервера. Спутник «ушёл на золото»

👉 Простой 🕒 5 мин 👁 3.3K

💎 +39

🔖 22

💬 18



eran

21 час назад

## Как мы создаём новые языки в Yandex SpeechKit. Рассказываем на примере узбекского

👉 Простой 🕒 8 мин 👁 2.1K

💎 +22

🔖 8

💬 7



CyberPaul

20 часов назад

## В изоляции. История появления и развития контейнеров

👉 Простой 🕒 9 мин 👁 2.2K

Ретроспектива

💎 +21

🔖 43

💬 2



Flamyatina

19 часов назад

## Доступность сервиса: экспресс-тестирование

👉 Простой 🕒 4 мин 👁 714

◆ +20

📖 12

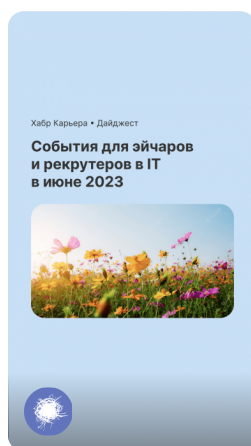
💬 0

## Атлант исправил плечи: вторая линия поддержки IT-проектов ЕВРАЗа

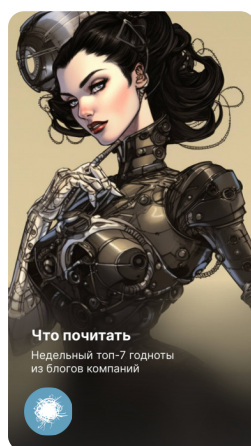
Турбо

Показать еще

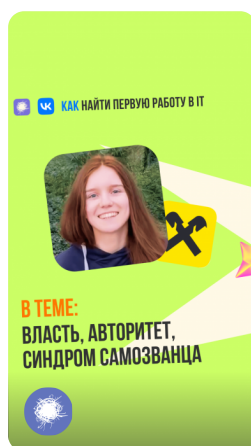
### ИСТОРИИ



События для HR и рекрутеров в IT в июне



Топ-7 годноты из блогов компаний



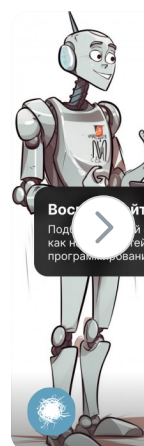
Власть, авторитет, синдром самозванца



Чем кулинария похожа на код?



Как взлететь к звёздам с «Промо»



Воспитай айтишника

### КУРСЫ

#### 🔴 Офлайн-курс JavaScript-разработчик

22 августа 2023 · 29 900 ₽ · Бруноям

#### 🟠 FullStack JavaScript программист в Москве

19 июня 2023 · 330 000 ₽ · Elbrus Coding Bootcamp

#### 🟠 FullStack JavaScript программист в Санкт-Петербурге

19 июня 2023 · 290 000 ₽ · Elbrus Coding Bootcamp

#### 🟠 FullStack JavaScript программист Онлайн

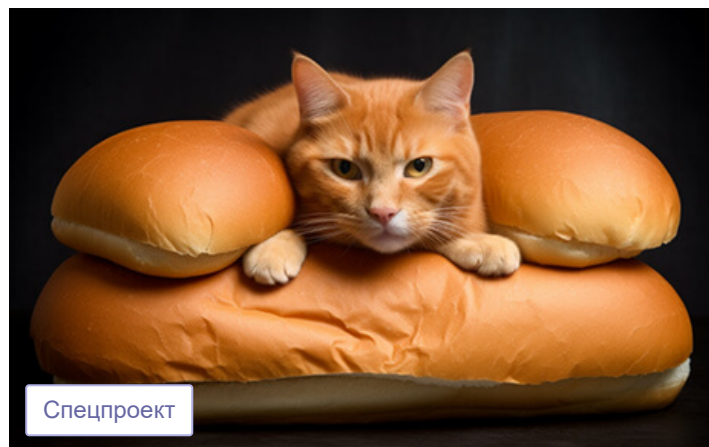
3 июля 2023 · 260 000 ₽ · Elbrus Coding Bootcamp

#### 🟢 Product Manager

19 июня 2023 · 123 900 ₽ · Нетология

[Больше курсов на Хабр Карьере](#)

## МИНУТОЧКУ ВНИМАНИЯ

[Спецпроект](#)[Зачем дата-инженеру программировать](#)[Спецпроект](#)[Все плюшки открытых платформ для разработчика](#)

## РАБОТА

[React разработчик](#)

111 вакансий

[JavaScript разработчик](#)

282 вакансии

[Все вакансии](#)

# Хабр

[🌐 Настройка языка](#)[Техническая поддержка](#)[Вернуться на старую версию](#)

© 2006–2023, Habr