

Актуальные
зарплаты в IT

110k

100k

150k

120k

180k



empenoso

18 фев в 03:25

Автоматизируем учёт облигаций: как избавиться от рутины

🟢 Простой

🕒 7 мин

👁️ 4.5K

Python*, API*, Open source*, Финансы в IT

Кейс

В преддверии одной большой конференции про облигации я, Михаил Шардин и Екатерина Кутняк решили написать статью о том, почему облигации важны и как можно автоматизировать рутинные действия с ними.



Передаю слово Екатерине:

Сейчас, наверно, уже самые начинающие инвесторы понимают насколько важны облигации в портфеле. Ведь они выполняют целый ряд функций, помимо зарабатывания денег.

Улучшают ликвидность, оптимизирую показатель риск/доходность, придают гибкость при принятии решений. А на сегодняшний день они ещё и самый доходный и предсказуемый инструмент.

Мало кто из частных инвесторов заморачивается ведением своего портфеля. Конечно, это неправильно, но вполне можно понять. Слишком много времени уходит на учет, а его всегда и всем не хватает. Поэтому и доверяют своему торговому приложению. Но не видя

картинки в целом, очень трудно принимать как стратегические, так и тактические решения по портфелю.

Сейчас есть приложения, которые помогают вести портфель. Они не бесплатные, но порядок в портфеле того стоит.

Это статья скорее для тех, кто понимает важность учета и пытается сам вести свой портфель в Excel. У меня было четыре статьи о том, как вести учет облигаций в Excel. Их можно посмотреть здесь:



Как вести учет облигаций в Excel

smart-lab.ru



Как вести учет облигаций в Excel. Расчет доходности

smart-lab.ru



Как вести учет облигаций в Excel. Часть 3. Дюрация

smart-lab.ru



Как рассчитать дюрацию «на коленке». Часть 4

smart-lab.ru

Но всегда есть способ что-то улучшить и оптимизировать. Например сервисы по учету бумаг, [такие как Snowball или другие](#), предлагают помимо прочих опций еще и график купонных выплат. Вещь, безусловно очень важная. Почему?

Чтобы максимально эффективно управлять облигационной частью портфеля, важно учитывать и анализировать все выплаты.

Почему учёт купонов так важен?

- Позволяет прогнозировать денежные потоки.
- Упрощает реинвестирование.

- ✔ Помогает оценить реальную доходность портфеля.

Например, брокерские приложения принципиально не учитывают выплаченные купоны, что вносит дикую путаницу в умах инвесторов. Они бесконечно жалуются на “красные” портфели, и сетуют на убытки. Забывая о том, что облигации - это еще и полученные купоны.

Собрать самому купонные выплаты можно, но очень трудоемко. Так как все портфели я веду в Excel, то прошла долгий путь от вбивания каждой купонной выплаты руками, до того, о чем мы расскажем в этой статье.

Итак, у нас есть проблема:

Как учитывать купонные выплаты, если в портфеле несколько облигаций с разными датами выплат? Вручную отслеживать их сложно, особенно если бумаги погашаются и покупаются новые.

Обычные способы учёта:

- Вести записи вручную в Excel.
- Использовать специализированные приложения (платно)

Ручной способ

В Excel трудоёмко, легко что-то упустить. От ручного вбивания каждой выплаты, а перешла к сводным таблицам. Это гораздо нагляднее. В результате, получается отлично отслеживать не только купоны, но и погашения, и оферты.

Можно смотреть выплаты по месяцам

Годы (Дата)	2025 
Названия строк 	Сумма по полю Купон
 фев	183 263.58 ₽
 мар	228 212.62 ₽
 апр	153 610.09 ₽
 май	197 245.46 ₽
 июн	218 980.19 ₽
 июл	215 468.50 ₽
 авг	225 193.98 ₽
 сен	221 011.24 ₽
 окт	146 784.05 ₽
 ноя	184 540.61 ₽
 дек	214 257.39 ₽
Общий итог	2 188 567.70 ₽

А можно раскрыть таблицу, и смотреть по дням в конкретном месяце:

Годы (Дата)	2025 
Названия строк 	Сумма по полю Купон
 фев	183 263.58 ₽
07.фев	1 841.60 ₽
12.фев	17 141.90 ₽
13.фев	20 606.40 ₽
14.фев	23 066.08 ₽
15.фев	5 499.00 ₽
17.фев	14 260.88 ₽
18.фев	2 388.47 ₽
19.фев	7 102.55 ₽
20.фев	18 568.29 ₽
21.фев	7 176.96 ₽
24.фев	6 537.60 ₽
25.фев	27 163.98 ₽
27.фев	9 969.81 ₽
28.фев	21 940.06 ₽
 мар	228 212.62 ₽

Можно добавить наименование выпусков, и тогда картинка получается максимально информативная:

Месяцы (Дата)	фев	м																
Годы (Дата)	2025																	
Сумма по полю Купон	Названия столбцов																	
Названия строк	07.02.2025	12.02.2025	13.02.2025	14.02.2025	15.02.2025	17.02.2025	18.02.2025	19.02.2025	20.02.2025	21.02.2025	24.02.2025	25.02.2025	27.02.2025	28.02.2025	Общий итог			
АБЗ-1-001P-05-606	1841.6														1841.6			
ВымпелКом-001P-06						7890									7890			
Газпром нефть-003P-08R									18568.29						18568.29			
ГПБ-005P-01P													7135.08		7135.08			
ЕвразХолдингФин-003P-01					5499										5499			
ИКС 5 Финанс-003P-02-606											6537.6				6537.6			
Инарктика-002P-01													8029.78		8029.78			
Металлоинвест ХК-001-06												19926			19926			
НОВАТЭК-001P-02								7102.54506							7102.54506			
Новотранс ХК-001P-04												7237.98			7237.98			
ОК РУСАЛ-50-001P-09													3489.81		3489.81			
ОФЗ-26240-ПД	10435.1														10435.1			
Полюс-ПБО-03				12653.84											12653.84			
РЖД-001P-28R	6706.8														6706.8			
Россети Центр-001P-03													6480		6480			
Ростелеком-001P-13R						4536									4536			
Ростелеком-002P-13R											7176.96				7176.96			
Росжснбанк-50-002P-04			20606.4												20606.4			
РСХБ-50-02-002P														6775.2	6775.2			
Селектел-001P-03R				10412.24											10412.24			
СЕЛЛ-Сервис-50-01							2388.47								2388.47			
Т-Финанс МФК-001P-01						1834.88									1834.88			
Общий итог	1841.6	17141.9	20606.4	23066.08	5499	14260.88	2388.47	7102.54506	18568.29	7176.96	6537.6	27163.98	9969.81	21940.06	183263.57511			

Безусловно, так гораздо нагляднее и удобнее. Но остается одна проблема, чтобы сделать сводные таблицы все еще нужна исходная таблица, содержащая всю необходимую информацию. Вот ее то по-прежнему приходилось делать руками.

Но и у этой проблемы есть техническое решение.



Передаём слово Михаилу:

Автоматизированное получение данных о купонах и выплатах номинала облигаций

Разработал Python скрипт, который автоматически скачивает данные о купонах и выплатах номинала через API Московской биржи для списка облигаций из первой вкладки Excel-файла и записывает результат обратно в этот же файл на вкладку "Ден.поток". Третья вкладка содержит формулы анализа полученных данных.

Автосохранение bonds.xlsx Поиск

Файл Главная Вставка Рисование Разметка страницы Формулы Данные Рецензирование Вид

Вставить Буфер обмена Шрифт Выравнивание Число

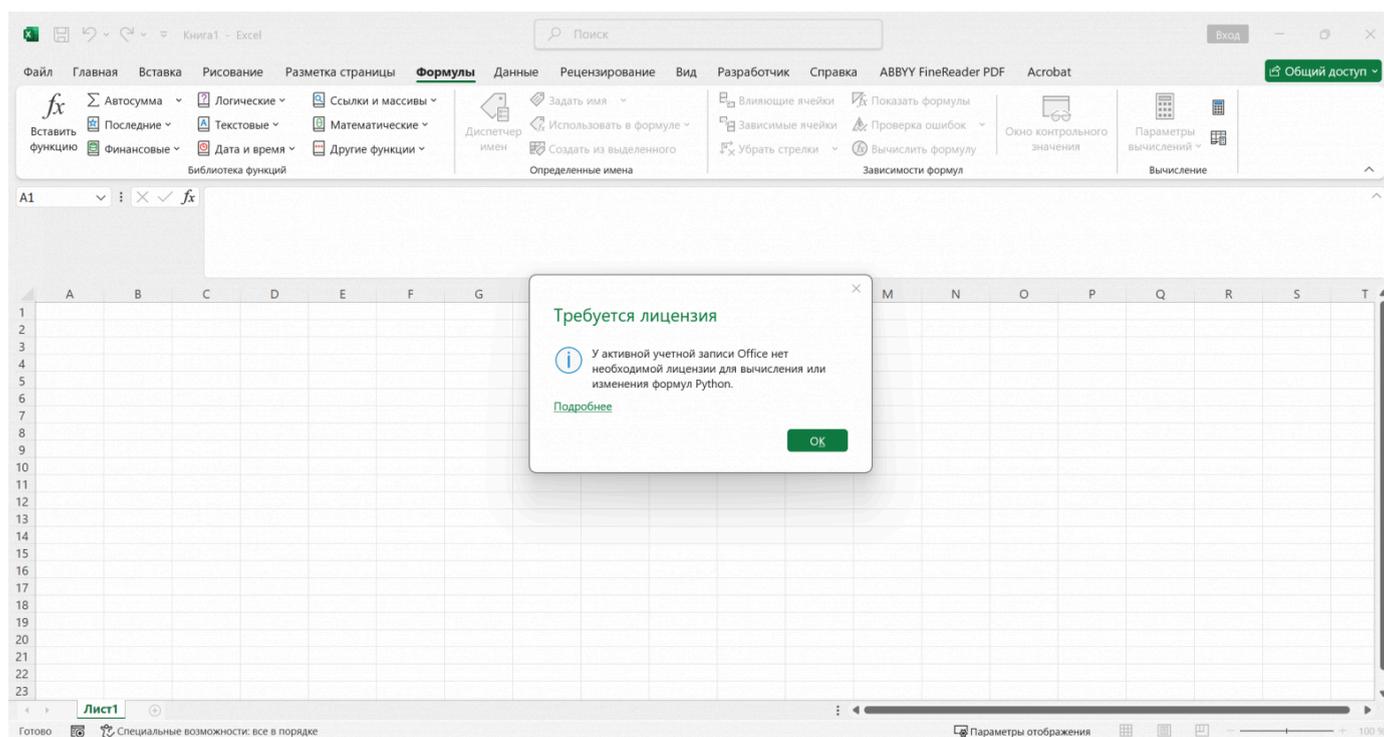
В3 fx =СУММЕСЛИМН(Ден.поток!D:D; Ден.поток!C:C; ">=" & A3; Ден.поток!C:C; "<" & ДАТАМЕС(A3;1); Ден.поток!A:A; "*" (купон) *)

	А	В	С	Д	Е
1	🔍 Бюджет поступлений в рублях от 15.02.2025 на год вперёд (итого 2 362 517 Р):				
2	Месяцы	Только купоны, Р	Совокупные выплаты: купоны + погашения, Р		
3	Февраль 2025	19 257,40 Р	19 257,40 Р		
4	Март 2025	48 437,96 Р	58 597,96 Р		
5	Апрель 2025	94 792,80 Р	104 952,80 Р		
6	Май 2025	67 593,41 Р	305 208,81 Р		
7	Июнь 2025	30 372,39 Р	65 737,79 Р		
8	Июль 2025	85 494,26 Р	779 819,66 Р		
9	Август 2025	56 320,41 Р	91 685,81 Р		
10	Сентябрь 2025	33 100,73 Р	68 466,13 Р		
11	Октябрь 2025	84 727,45 Р	540 052,85 Р		
12	Ноябрь 2025	29 359,01 Р	86 286,61 Р		
13	Декабрь 2025	21 290,93 Р	242 450,93 Р		
14	Январь 2026	57 613,52 Р	507 733,52 Р		
15					
16	🔍 Все ближайшие выплаты с 15.02.2025 по 17.03.2025:				
17	Интерскол КЛС БО-03 (купон 🗝)	RU000A10ATB6	7 марта 2025 г.	4 801,50 Р	
18	АБЗ-1 001P-05 (купон 🗝)	RU000A1070X5	9 марта 2025 г.	2 578,24 Р	
19	МОНОПОЛИЯ 001P-03 (купон 🗝)	RU000A10ARS4	1 марта 2025 г.	4 268,88 Р	
20	Интерлизинг 001P-07 (купон 🗝)	RU000A1077X0	10 марта 2025 г.	2 806,70 Р	
21	Интерлизинг 001P-07 (номинал 100)	RU000A1077X0	10 марта 2025 г.	10 160,00 Р	
22	Каршеринг Россия 001P-02 (купон 🗝)	RU000A106A86	21 февраля 2025 г.	6 933,54 Р	
23	Банк ВТБ (ПАО) Б-1-343 (купон 🗝)	RU000A106TM6	1 марта 2025 г.	3 238,11 Р	
24	Новые технологии 001P-02 (купон 🗝)	RU000A10ANE5	24 февраля 2025 г.	4 113,20 Р	
25	Ритейл Бел Финанс 001P-01 (купон 🗝)	RU000A101QF7	26 февраля 2025 г.	4 765,01 Р	
26	Селектел 001P-04R (купон 🗝)	RU000A1089J4	8 марта 2025 г.	2 601,63 Р	
27	Магнит ПАО БО-005P-01 (купон 🗝)	RU000A10ANZ8	23 февраля 2025 г.	3 445,65 Р	

Исходные данные Ден.поток **Аналитика** Описание

Расчёт для портфеля на 5 млн рублей

Я выбрал использовать Python скрипт отдельно от встроенной функции Python в Excel, потому что встроенная функция накладывает множество ограничений. Например, у меня был Microsoft Office 2021 - в нём функции Python нет. Пришлось установить Microsoft Office 2024 - только в нём есть встроенный Python в Excel. Но и этого оказалось недостаточно чтобы запускать встроенный Python в Excel. Необходимо иметь Office 365 (Mondo 2016) подписку, чтобы функция активировалась. И будет работать только под Windows. Да и сам Python в Excel обрезанный, можно использовать только библиотеки рекомендованные Microsoft и Anaconda.



Исходя из этого я выбрал написать отдельный Python скрипт - ведь тогда уже не важно какая именно версия Эксель у вас установлена. И даже не важно какая операционная система - Windows или macOS. Да даже если и Linux - Python поддерживает множество операционных систем, а свободный аналог Экселя под Linux называется LibreOffice Calc.

Как это помогает?

- Видно, когда и сколько денег поступит.
- Можно заранее планировать реинвестирование.
- Повышается контроль над кэшем в портфеле.

Все это касается уже существующих портфелей.

Но есть еще один неочевидный плюс для тех, кто прежде чем купить облигационный портфель, планирует его структуру. Можно не прилагая больших усилий найти оптимальное соотношение бумаг, чтобы получить планируемый поток кэша. Это очень удобно именно на этапе выбора структуры.

Как это выглядит на практике?

Скрипт является [open-source проектом, размещенным на GitHub](#). Это означает что любой человек может не только свободно посмотреть его код, но и предложить свои улучшения через отправку pull request (PR) на GitHub.

The screenshot shows a GitHub repository page for 'moex-bond-search-and-analysis' by user 'empenoso'. The repository is public and has 1 branch and 0 tags. It contains 16 commits and 1 star. The file list shows:

- .gitignore (Initial commit, 2 days ago)
- 2_bonds_search_coupons.py** (Скачивание и обработка данных о денежном потоке обл..., now)
- LICENSE (Initial commit, 2 days ago)
- README.md (Update README.md, 1 hour ago)
- bonds.xlsx** (Скачивание и обработка данных о денежном потоке обл..., now)

The README file is selected, showing the title 'Поиск ликвидных облигаций на Московской Бирже и их анализ' and a description: 'Этот репозиторий содержит три Python-скрипта, которые помогают частным инвесторам находить ликвидные облигации, анализировать денежные потоки и отслеживать новости эмитентов.' The repository also has a license of Apache-2.0 and 0 forks.

Для того чтобы Вы могли воспользоваться скриптом - сначала скачайте его на свой компьютер с Гитхаба по кнопке < > Code / Download ZIP.

Весь скрипт находится в файле « 2_bonds_search_coupons.py ». Вы можете познакомиться с его содержимым перед запуском - для этого откройте его, например, блокнотом.

```
# 💰 Скачивание и обработка данных о денежном потоке облигаций 💰
#
# Этот Python скрипт автоматически скачивает данные о купонах и выплатах номинала
# через API Московской биржи для списка облигаций из Excel-файла bonds.xlsx и
# записывает результат обратно в этот же файл.
#
# Установка зависимостей перед использованием: pip install requests openpyxl
#
# Автор: Михаил Шардин https://shardin.name/
# Дата создания: 29.01.2025
# Версия: 1.1
#
# Актуальная версия скрипта всегда здесь: https://github.com/empenoso/moex-bond-search-
#

import requests
import openpyxl
```

```
from datetime import datetime

# Настройка кодировки для корректного вывода русского текста
import sys
sys.stdout.reconfigure(encoding='utf-8')

# Загружаем Excel-файл
file_path = "bonds.xlsx"
wb = openpyxl.load_workbook(file_path)
sheet_data = wb["Исходные данные"]
sheet_result = wb["Ден.поток"]

# Очищаем лист с результатами
sheet_result.delete_rows(1, sheet_result.max_row)
sheet_result.append(["Название", "Идентификатор", "Дата выплаты", "Денежный поток, Р (к

# Считываем данные из листа "Исходные данные"
ArraySymbolQuantity = []
for row in sheet_data.iter_rows(min_row=2, max_row=sheet_data.max_row, values_only=True):
    if row[0] and row[1]: # Проверяем, что данные не пустые
        ArraySymbolQuantity.append(row)

print(f"Считано {len(ArraySymbolQuantity)} облигаций для обработки.")

CashFlow = []

# Обрабатываем каждую облигацию
for ID, number in ArraySymbolQuantity:
    print(f"\nОбрабатываем {ID}, количество: {number} шт.")
    url = f"https://iss.moex.com/iss/statistics/engines/stock/markets/bonds/bondization
    print(f"Запрос к {url}")

    response = requests.get(url)
    json_data = response.json()

# Обработка купонов
for coupon in json_data.get("coupons", {}).get("data", []):
    name = coupon[1].replace("'", '').replace('"', '').replace("\\", '')
    isin = coupon[0]
    coupon_date = coupon[3]

# Преобразуем дату в объект datetime
coupon_datetime = datetime.strptime(coupon_date, "%Y-%m-%d")
```

```
if coupon_datetime > datetime.now():
    value_rub = (coupon[9] or 0) * number
    CashFlow.append([f"{name} (купон 💎)", isin, coupon_datetime, value_rub])
    print(f"Добавлен купон: {CashFlow[-1]}")

# Обработка выплат номинала
for amort in json_data.get("amortizations", {}).get("data", []):
    name = amort[1].replace("'", '').replace('"', '').replace("\\", '')
    isin = amort[0]
    amort_date = amort[3]

    # Преобразуем дату в объект datetime
    amort_datetime = datetime.strptime(amort_date, "%Y-%m-%d")

    if amort_datetime > datetime.now():
        value_rub = (amort[9] or 0) * number
        CashFlow.append([f"{name} (номинал 💰)", isin, amort_datetime, value_rub])
        print(f"Добавлена выплата номинала: {CashFlow[-1]}")

# Записываем данные в Excel
for row in CashFlow:
    sheet_result.append(row)

# Устанавливаем формат ячеек
for cell in sheet_result["C"][1:]: # Пропускаем заголовок
    cell.number_format = "DD.MM.YYYY"

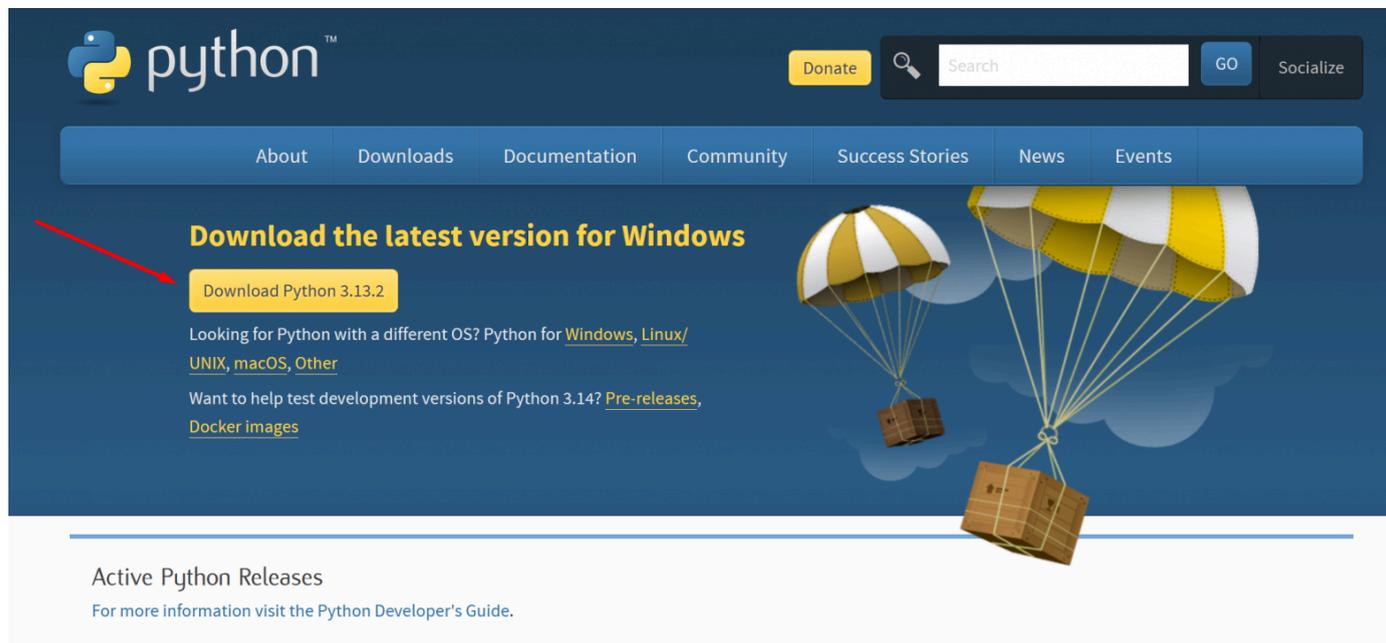
for cell in sheet_result["D"][1:]:
    cell.number_format = '# ##0,00 P'

# Добавляем запись об обновлении
update_message = f"\nДанные автоматически обновлены {datetime.now().strftime('%d.%m.%Y')}
sheet_result.append(["", update_message])
print(update_message)

# Сохраняем изменения в файле
wb.save(file_path)
print(f"Файл {file_path} успешно обновлён.")

# В конце скрипта
input("Нажмите любую клавишу для выхода...")
```

Хотя нумерация скрипта начинается с двойки, это полностью самостоятельный скрипт, который ищет и скачивает данные о купонах и выплатах номинала через API Московской биржи.



The screenshot shows the Python website's navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. A search bar and a 'Donate' button are also visible. The main content area features a large heading 'Download the latest version for Windows' with a red arrow pointing to a yellow button labeled 'Download Python 3.13.2'. Below this button, there are links for other operating systems: 'Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)'. There are also links for 'Pre-releases' and 'Docker images'. The background of the main content area features an illustration of two parachutes with cargo boxes.

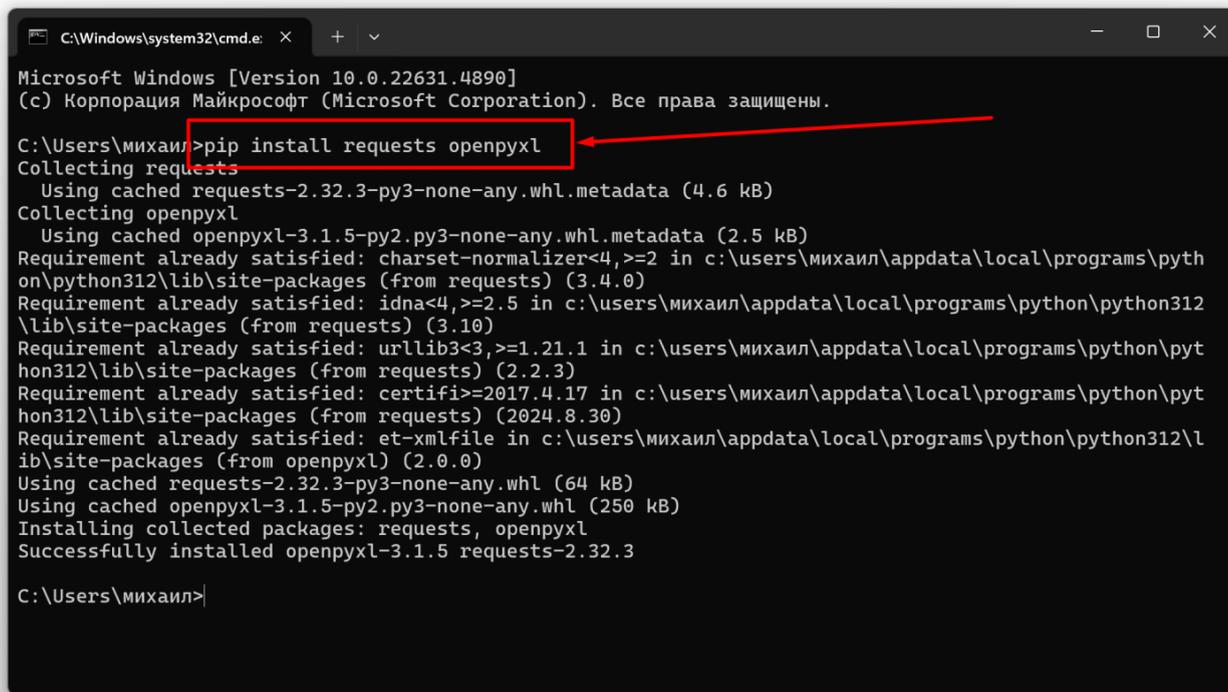
Теперь чтобы запустить скрипт сначала нужно установить Python. Скачайте и установите последнюю версию с [официального сайта Python](#) для Вашей операционной системы.

Затем, найдите в меню "Пуск" "Командную строку" или "Терминал" и откройте её. В командной строке введите:

```
pip install requests openpyxl
```

Это установит необходимые для работы скрипта инструменты - без их установки скрипт работать не будет:

- Requests: библиотека для отправки HTTP-запросов и работы с ответами от серверов.
- OpenPyXL: библиотека для чтения и записи данных в файлы Excel (.xlsx).



```
C:\Windows\system32\cmd.e. x + v
Microsoft Windows [Version 10.0.22631.4890]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\михаил>pip install requests openpyxl
Collecting requests
  Using cached requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Collecting openpyxl
  Using cached openpyxl-3.1.5-py2.py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\михаил\appdata\local\programs\python\python312\lib\site-packages (from requests) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in c:\users\михаил\appdata\local\programs\python\python312\lib\site-packages (from requests) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\михаил\appdata\local\programs\python\python312\lib\site-packages (from requests) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\михаил\appdata\local\programs\python\python312\lib\site-packages (from requests) (2024.8.30)
Requirement already satisfied: et-xmlfile in c:\users\михаил\appdata\local\programs\python\python312\lib\site-packages (from openpyxl) (2.0.0)
Using cached requests-2.32.3-py3-none-any.whl (64 kB)
Using cached openpyxl-3.1.5-py2.py3-none-any.whl (250 kB)
Installing collected packages: requests, openpyxl
Successfully installed openpyxl-3.1.5 requests-2.32.3

C:\Users\михаил>
```

Заполните файл-шаблон « `bonds.xlsx` » Вашими собственными данными.

Переименовывать файл и менять название вкладок нельзя - в скрипте жестко заданы названия файла и вкладки. Убедитесь, что Excel-файл « `bonds.xlsx` » закрыт перед запуском.

Автосохранение bonds.xlsx Поиск

Файл **Главная** Вставка Рисование Разметка страницы Формулы Данные

Вставить Буфер обмена Шрифт Выравнивание

E11

	A	B	C	D	E	F	G
	Код ценной бумаги с Московской биржи	Количество					
1							
2	RU000A10ATB6	198					
3	RU000A106HF5	242					
4	RU000A104ZK2	210					
5	RU000A1070X5	224					
6	RU000A10ARS4	196					
7	RU000A1077X0	254					
8	RU000A106JV8	207					
9	RU000A104YT6	205					
10	RU000A106A86	219					
11	RU000A103G00	242					
12	RU000A1066A1	229					
13	RU000A103QK3	240					
14	RU000A106TM6	201					
15	RU000A10AHE5	182					
16	RU000A101QF7	809					
17	RU000A105PP9	211					
18	RU000A106R95	223					
19	RU000A1089J4	211					
20	RU000A1066J2	214					
21	RU000A109ZK1	197					
22	RU000A10ANZ8	195					
23	RU000A0JVVW71	186					
24	RU000A10ASS2	194					
25	RU000A10ASB8	193					
26	RU000A107MM9	207					
27							

Исходные данные | Ден.поток | Аналитика | Описание

Готово Специальные возможности: все в порядке

Теперь, найдите « 2_bonds_search coupons.py » в проводнике, дважды кликните по нему. Это запустит скрипт, и он обновит файл Excel. После окончания работы можно посмотреть логи работы - скрипт ждёт нажатия кнопки.

```

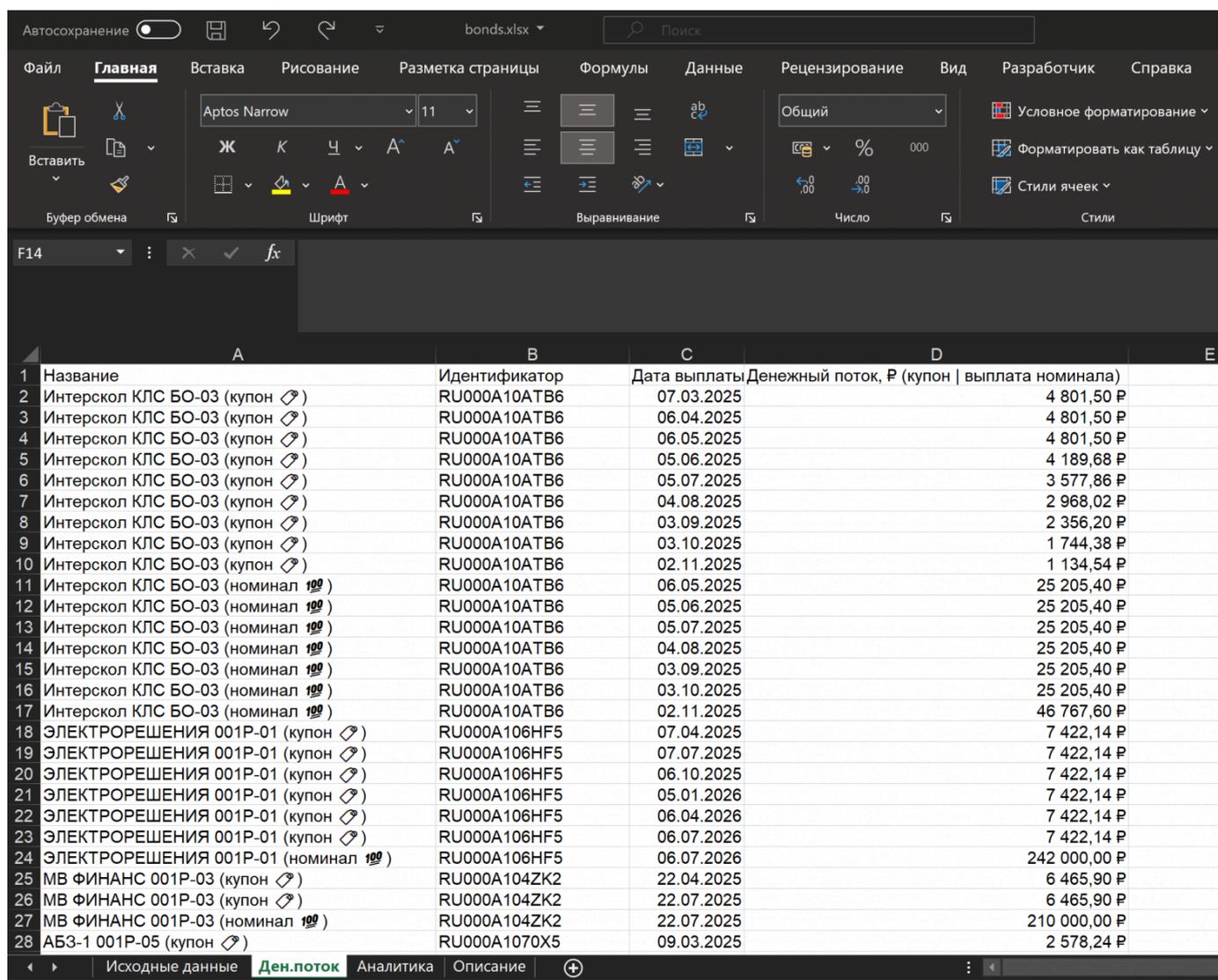
C:\Users\михаил\AppData\Lo x + v -
Добавлен купон: ['Магнит ПАО БО-005P-01 (купон 🟡)', 'RU000A10ANZ8', datetime.datetime(2025, 10, 21, 0, 0), 3445.6500000
000005]
Добавлен купон: ['Магнит ПАО БО-005P-01 (купон 🟡)', 'RU000A10ANZ8', datetime.datetime(2025, 11, 20, 0, 0), 3445.6500000
000005]
Добавлен купон: ['Магнит ПАО БО-005P-01 (купон 🟡)', 'RU000A10ANZ8', datetime.datetime(2025, 12, 20, 0, 0), 3445.6500000
000005]
Добавлен купон: ['Магнит ПАО БО-005P-01 (купон 🟡)', 'RU000A10ANZ8', datetime.datetime(2026, 1, 19, 0, 0), 3445.65000000
00005]
Добавлен купон: ['Магнит ПАО БО-005P-01 (купон 🟡)', 'RU000A10ANZ8', datetime.datetime(2026, 2, 18, 0, 0), 3445.65000000
00005]
Добавлен купон: ['Магнит ПАО БО-005P-01 (купон 🟡)', 'RU000A10ANZ8', datetime.datetime(2026, 3, 20, 0, 0), 3445.65000000
00005]
Добавлен купон: ['Магнит ПАО БО-005P-01 (купон 🟡)', 'RU000A10ANZ8', datetime.datetime(2026, 4, 19, 0, 0), 3445.65000000
00005]
Добавлена выплата номинала: ['Магнит ПАО БО-005P-01 (номинал 🟡)', 'RU000A10ANZ8', datetime.datetime(2026, 4, 19, 0, 0),
195000]

Обрабатываем RU000A0JW71, количество: 186 шт.
Запрос к https://iss.moex.com/iss/statistics/engines/stock/markets/bonds/bondization/RU000A0JW71.json?iss.meta=off
Добавлен купон: ['Российские ЖД ОАО БО-02 (купон 🟡)', 'RU000A0JW71', datetime.datetime(2025, 4, 17, 0, 0), 19476.05999
9999998]
Добавлен купон: ['Российские ЖД ОАО БО-02 (купон 🟡)', 'RU000A0JW71', datetime.datetime(2025, 10, 16, 0, 0), 19476.0599
99999998]
Добавлена выплата номинала: ['Российские ЖД ОАО БО-02 (номинал 🟡)', 'RU000A0JW71', datetime.datetime(2025, 10, 16, 0,
0), 186000]

Обрабатываем RU000A10ASS2, количество: 194 шт.
Запрос к https://iss.moex.com/iss/statistics/engines/stock/markets/bonds/bondization/RU000A10ASS2.json?iss.meta=off
Добавлен купон: ['Ростелеком ПАО 001P-14R (купон 🟡)', 'RU000A10ASS2', datetime.datetime(2025, 3, 5, 0, 0), 3451.2599999
9999998]

```

Все эти данные, полученные через API Мосбиржи доступны только вам и для ознакомления.



	A	B	C	D	E
1	Название	Идентификатор	Дата выплаты	Денежный поток, Р (купон выплата номинала)	
2	Интерскол КЛС БО-03 (купон)	RU000A10ATB6	07.03.2025		4 801,50 Р
3	Интерскол КЛС БО-03 (купон)	RU000A10ATB6	06.04.2025		4 801,50 Р
4	Интерскол КЛС БО-03 (купон)	RU000A10ATB6	06.05.2025		4 801,50 Р
5	Интерскол КЛС БО-03 (купон)	RU000A10ATB6	05.06.2025		4 189,68 Р
6	Интерскол КЛС БО-03 (купон)	RU000A10ATB6	05.07.2025		3 577,86 Р
7	Интерскол КЛС БО-03 (купон)	RU000A10ATB6	04.08.2025		2 968,02 Р
8	Интерскол КЛС БО-03 (купон)	RU000A10ATB6	03.09.2025		2 356,20 Р
9	Интерскол КЛС БО-03 (купон)	RU000A10ATB6	03.10.2025		1 744,38 Р
10	Интерскол КЛС БО-03 (купон)	RU000A10ATB6	02.11.2025		1 134,54 Р
11	Интерскол КЛС БО-03 (номинал)	RU000A10ATB6	06.05.2025		25 205,40 Р
12	Интерскол КЛС БО-03 (номинал)	RU000A10ATB6	05.06.2025		25 205,40 Р
13	Интерскол КЛС БО-03 (номинал)	RU000A10ATB6	05.07.2025		25 205,40 Р
14	Интерскол КЛС БО-03 (номинал)	RU000A10ATB6	04.08.2025		25 205,40 Р
15	Интерскол КЛС БО-03 (номинал)	RU000A10ATB6	03.09.2025		25 205,40 Р
16	Интерскол КЛС БО-03 (номинал)	RU000A10ATB6	03.10.2025		25 205,40 Р
17	Интерскол КЛС БО-03 (номинал)	RU000A10ATB6	02.11.2025		46 767,60 Р
18	ЭЛЕКТРОРЕШЕНИЯ 001Р-01 (купон)	RU000A106HF5	07.04.2025		7 422,14 Р
19	ЭЛЕКТРОРЕШЕНИЯ 001Р-01 (купон)	RU000A106HF5	07.07.2025		7 422,14 Р
20	ЭЛЕКТРОРЕШЕНИЯ 001Р-01 (купон)	RU000A106HF5	06.10.2025		7 422,14 Р
21	ЭЛЕКТРОРЕШЕНИЯ 001Р-01 (купон)	RU000A106HF5	05.01.2026		7 422,14 Р
22	ЭЛЕКТРОРЕШЕНИЯ 001Р-01 (купон)	RU000A106HF5	06.04.2026		7 422,14 Р
23	ЭЛЕКТРОРЕШЕНИЯ 001Р-01 (купон)	RU000A106HF5	06.07.2026		7 422,14 Р
24	ЭЛЕКТРОРЕШЕНИЯ 001Р-01 (номинал)	RU000A106HF5	06.07.2026		242 000,00 Р
25	МВ ФИНАНС 001Р-03 (купон)	RU000A104ZK2	22.04.2025		6 465,90 Р
26	МВ ФИНАНС 001Р-03 (купон)	RU000A104ZK2	22.07.2025		6 465,90 Р
27	МВ ФИНАНС 001Р-03 (номинал)	RU000A104ZK2	22.07.2025		210 000,00 Р
28	АБЗ-1 001Р-05 (купон)	RU000A1070X5	09.03.2025		2 578,24 Р

Скрипт автоматически заполняет вкладку с денежным потоком, а все сводные данные в человеческом понятном виде находится на вкладке аналитика. Надеюсь этот скрипт облегчит Вашу работу!

Итоги: почему стоит учитывать облигации и как это сделать эффективно?

- ◆ Без учёта облигаций инвестор упускает важную часть доходности.
- ◆ Системный учёт купонных выплат даёт чёткое понимание будущих денежных потоков.
- ◆ Автоматизация позволяет не тратить время на рутину.
- ◆ Использование скрипта упрощает управление облигационной частью портфеля.

Авторы статьи: [Михаил Шардин](#) и [Екатерина Кутняк](#)

Код и техническая реализация: [Михаил Шардин](#)

P.S. Я планирую написать и опубликовать ещё три продолжения.

P.P.S. Небольшой анонс для публикации следующей недели: для меня, как для частного инвестора многие варианты облигаций с большими процентами окажутся мусорными. И у меня, как частного инвестора, много других проблем и забот - и мне не всегда есть время разбираться чем занят тот или иной эмитент. Нет времени знакомиться с эмитентом или проводить какую-то аналитику.

Публикация следующей недели будет посвящена скрипту, который по коду ценной бумаги с Московской биржи внутри файла `bonds.xlsx` автоматически собирает все последние новости об этой компании.

Теги: [моех](#), [облигации](#)

Хэбы: [Python](#), [API](#), [Open source](#), [Финансы в IT](#)

Редакторский дайджест

Присылаем лучшие статьи раз в месяц

**183**

Карма

87.1

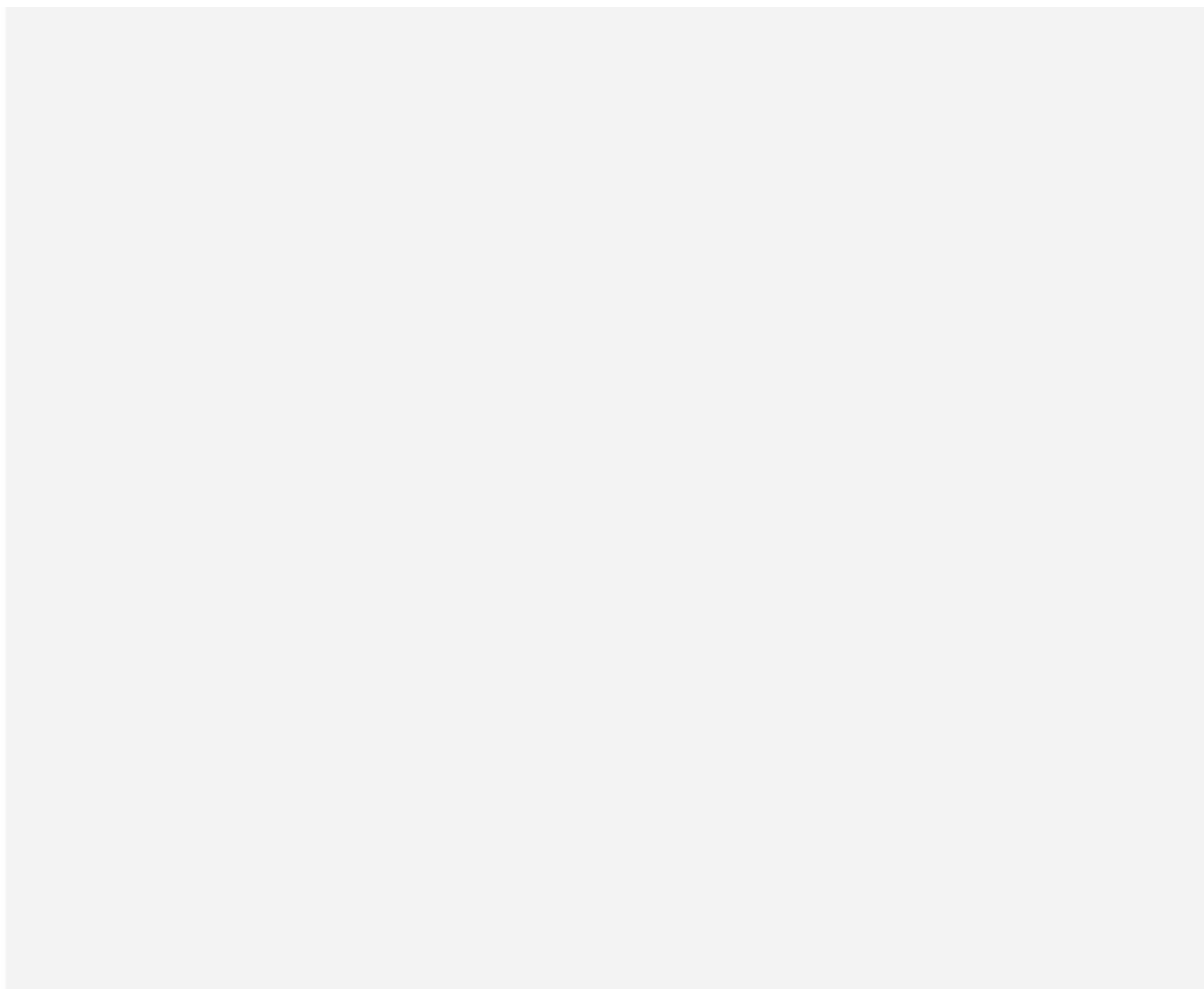
Рейтинг

Михаил Шардин [@empenoso](#)

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

[Подписаться](#)

[Сайт](#) [Сайт](#) [Github](#)



 [Комментарии 3](#)

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](#)

 **+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**Корcheniy**

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

 +16 181 16**PatientZero**

1 час назад

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

 Простой  13 мин  747

Тutorial

Перевод

 +15 21 1**lbkanter**

46 минут назад

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

 Средний  4 мин  216

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

ТурбоПоказать еще

ИСТОРИИ



Как расти на работе?



Здоровье айтишника



Угадайте будущее в новом сезоне



С праздником весны!



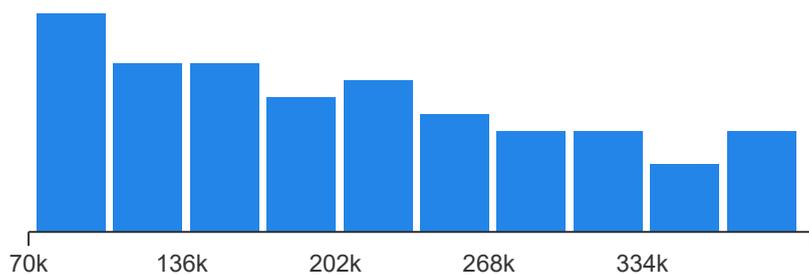
Всегда котов!

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



Девушка с розовыми волосами и Слизень на планете Рбах



Казино и аутсорс: как работают айтишники в РБ

РАБОТА

[Python разработчик](#)

65 вакансий

[Django разработчик](#)

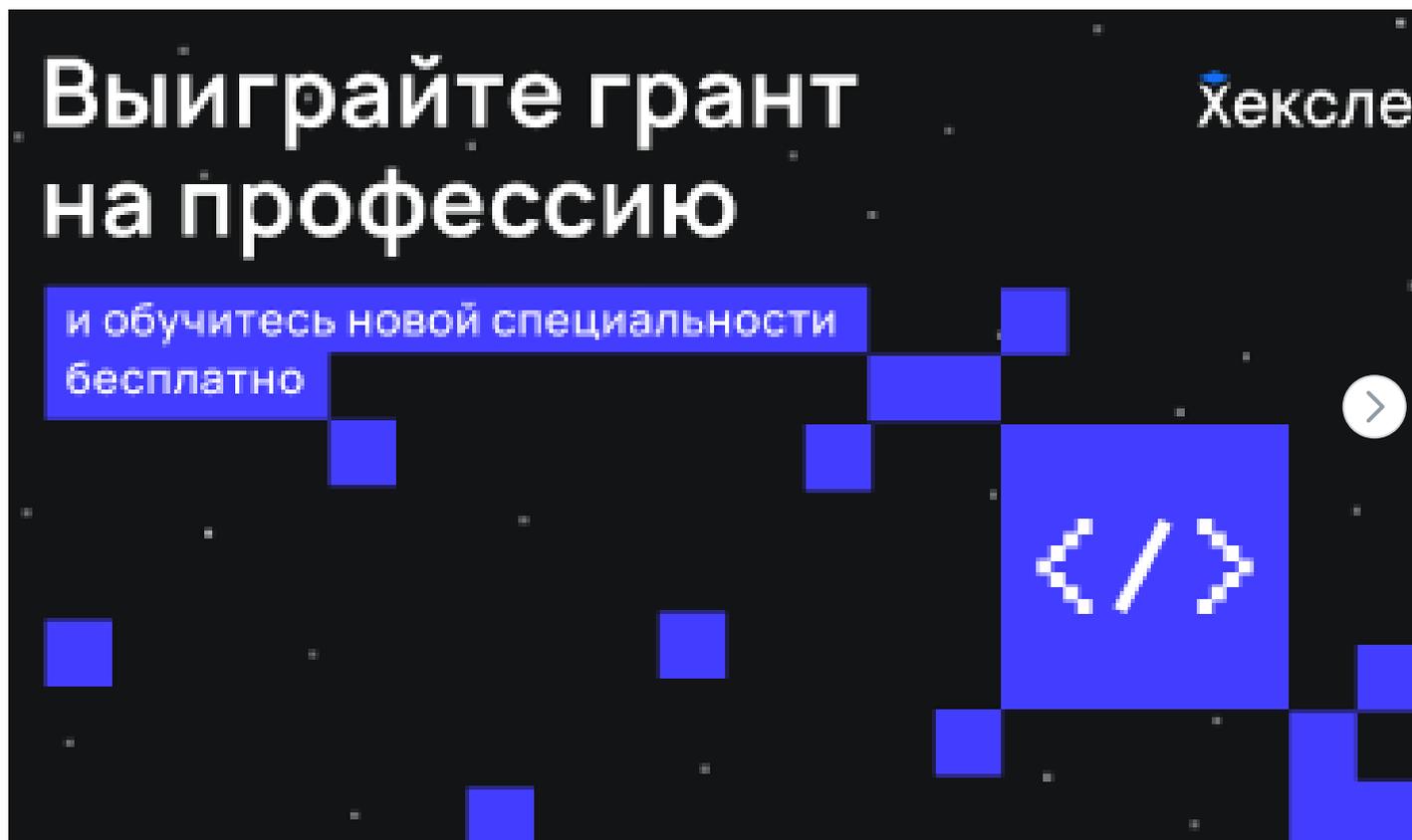
20 вакансий

[Data Scientist](#)

55 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