

Визуализация данных с помощью Python и JavaScript

Анализ и преобразование данных

Киран Дейл

УДК 004.4
ББК 32.973.26-018.2
Д27

Kyran Dale

Data Visualization with Python and JavaScript:
Scrape, Clean, Explore, and Transform Your Data 2nd Edition

© 2026 “Astana International Publishing” LLP Authorized Russian translation of the English edition of Data Visualization with Python and JavaScript, 2E ISBN 9781098111878
© 2023 Kyran Dale Limited This translation is published and sold by permission of O’Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Дейл, Киран.

Д27 Визуализация данных с помощью Python и JavaScript. Анализ и преобразование данных / Киран Дейл : [перевод с английского Ю. Смирновой]. — Алматы : Астана иностранная пресса, 2026. — 624 с. — (O’Reilly. Книги по программированию).

ISBN 978-601-12-4680-4

Хотите научиться эффективно представлять данные? Эта книга покажет полный путь преобразования сырых данных в яркие и информативные визуализации. Вы освоите инструменты Python и JavaScript, используя популярные и доступные библиотеки. Киран Дейл делится проверенными методами сбора, очистки и анализа данных, демонстрируя создание динамических веб-интерфейсов. Вы сможете уверенно создавать привлекательные и понятные представления данных как локально, так и прямо в браузере.

Будет полезно для всех, кто хочет прокачать навыки обработки и отображения данных в современных веб-приложениях.

УДК 004.4
ББК 32.973.26-018.2

ISBN 978-601-12-4680-4

© Смирнова Ю. Н., перевод на русский язык, 2026
© Издание на русском языке, оформление.
ТОО «Издательство «Астана иностранная пресса», 2026

Оглавление

Предисловие	10
Второе издание	14
Принятые в книге обозначения	15
Использование примеров кода	16
Благодарности	17
Введение	18
Для кого эта книга?	19
Почему именно Python и JavaScript?	22
Чему вы научитесь	25
Предварительные сведения	26
Тулчейн для визуализации данных	27
Как пользоваться этой книгой	30
Немного контекста	31
Резюме	33
Рекомендуемые книги	34

Раздел I. Базовый пакет инструментов

Глава 1. Подготовка окружения	36
Сопутствующий код	36
Python	36
Установка дополнительных библиотек	37
JavaScript	39
Базы данных	41
Интегрированная среда разработки	43
Резюме	44
Глава 2. Обучающий мостик между Python и JavaScript	45
Сходство и различия	45
Взаимодействие с кодом	46
Строим мост	49
Примеры различий	77
Шпаргалка	90
Резюме	92
Глава 3. Чтение и запись данных с помощью Python	94
Просто ли это?	94
Передача данных	95
Работа с системными файлами	96
CSV, TSV и табличные форматы данных	97
JSON	100
SQL	105

MongoDB	116
Работа с датами, временем и сложными типами данных	122
Резюме	123
Глава 4. Основы веб-разработки	125
Общая картина	125
Одностраничные приложения	126
Настройка инструментов	126
Создание веб-страницы	130
Chrome DevTools	140
Базовая страница с плейсхолдерами	142
Позиционирование и изменение размера контейнеров с помощью Flex	146
Масштабируемая векторная графика	155
Резюме	169

Раздел II. Получение данных

Глава 5. Получение данных из интернета с помощью Python	173
Получение данных из интернета с помощью библиотеки Requests	173
Получение файлов данных с помощью Requests	174
Использование Python для получения данных через web API	177
Доступ к web API с помощью библиотек	183
Скрейпинг данных	189
Получение объекта BeautifulSoup	191
Выбор тегов	192
Резюме	202
Глава 6. Эффективный скрейпинг с помощью Scrapy	203
Установка Scrapy	204
Постановка целей	205
Работа с XPath в Scrapy	207
Первый паук Scrapy	214
Скрейпинг биографических страниц лауреатов	221
Цепочка запросов и извлечение данных	224
Конвейеры Scrapy	229
Скрейпинг текста и изображений с помощью конвейера	231
Резюме	239

Раздел III. Очистка и исследование данных с помощью pandas

Глава 7. Введение в NumPy	243
Массив NumPy	244
Создание функций для работы с массивами	251
Резюме	253
Глава 8. Знакомство с библиотекой pandas	254
Почему pandas оптимальна для визуализации данных	254
Зачем разработали pandas	254
Классификация данных и измерения	255
DataFrame	256

Создание и сохранение структур DataFrame	261
Создание DataFrame из Series	272
Резюме	275
Глава 9. Очистка данных с помощью pandas	276
Чистая правда о грязных данных	276
Проверка качества данных	278
Индексы и отбор данных с помощью pandas	282
Очистка данных	286
Полная функция для очистки данных	306
Добавление столбца born_in	307
Сохранение очищенных наборов данных	312
Резюме	313
Глава 10. Визуализация данных с помощью Matplotlib	314
Pyplot и объектно-ориентированная библиотека Matplotlib	314
Запуск интерактивной сессии	315
Создание интерактивных графиков с помощью глобального состояния pyplot	316
Фигуры и объектно-ориентированная Matplotlib	322
Типы графиков	327
Seaborn	336
Резюме	345
Глава 11. Анализ данных с помощью pandas	347
Начало исследования	348
Построение графиков с помощью pandas	350
Гендерные диспропорции	352
Национальные тренды	360
Возраст и ожидаемая продолжительность жизни лауреатов	373
Нобелевская «диаспора»	380
Резюме	382

Раздел IV. Передача данных

Глава 12. Передача данных	385
Передача данных	386
Доставка файлов данных	391
Динамическое обновление данных с помощью Flask API	396
Использование динамической или статической доставки	398
Резюме	399
Глава 13. RESTful Data с помощью Flask	400
Инструменты для работы с RESTful	400
Создание базы данных	401
Flask RESTful для работы с данными	402
Добавление маршрутов RESTful API	405
Расширение API с помощью MethodView	411
Пагинация возвращаемых данных	414
Удаленное развертывание API на Heroku	418
Резюме	422

Раздел V. Визуализация данных с помощью D3 и Plotly

Глава 14. Перенос диаграмм в интернет с помощью Matplotlib и Plotly	425
Создание статических диаграмм с помощью Matplotlib	425
Построение диаграмм с помощью Plotly	430
Из Notebook в веб-формат с помощью Plotly	444
Создание нативных JavaScript-диаграмм с помощью Plotly	448
Интерактивная визуализация Plotly с помощью JavaScript и HTML	454
Резюме	459
Глава 15. Разработка концепции визуализации Нобелевской премии	460
Для кого эта визуализация?	460
Выбор визуальных элементов	461
Строка меню	462
Распределение премии по годам	463
Карта, показывающая выборку стран нобелевских лауреатов	464
Столбчатая диаграмма, показывающая количество лауреатов по странам	465
Список выбранных лауреатов	466
Визуализация целиком	468
Резюме	469
Глава 16. Создание визуализации	470
Предварительные сведения	471
HTML-каркас	473
Стили CSS	477
Движок JavaScript	481
Запуск приложения для визуализации данных о нобелевских лауреатах	496
Резюме	497
Глава 17. Введение в D3 на примере столбчатой диаграммы	498
Формулирование задачи	499
Работа с выборкой	499
Добавление элементов DOM	503
Использование D3	510
Шкалы в D3: от данных к их визуальному представлению	510
Привязка данных к элементам DOM — главное преимущество D3	516
Обновление DOM при изменении данных	516
Сборка столбчатой диаграммы	520
Оси и метки	523
Переходы	529
Резюме	534
Глава 18. Визуализация отдельных премий	535
Создание структуры	535
Шкалы	536
Оси	537
Метки номинаций	538
Вложенные данные	540
Добавление лауреатов с помощью вложенных объединений данных	543
Добавим немного блеска!	547
Резюме	549

Глава 19. Картографирование с помощью D3	550
Доступные карты	550
Форматы данных для картографирования в D3	551
Библиотека D3-geo, проекции и пути	556
Соединение элементов воедино	562
Обновление карты	565
Добавление индикаторов показателей	569
Готовая карта	572
Создание простой всплывающей подсказки	573
Резюме	578
Глава 20. Визуализация данных отдельных лауреатов	579
Создание списка лауреатов	580
Создание биографического блока	584
Резюме	587
Глава 21. Строка меню	589
Создание HTML-элементов с помощью D3	590
Создание строки меню	590
Резюме	601
Глава 22. Заключение	602
Подведение итогов	602
Дальнейшее развитие	605
Заключительные замечания	607
Приложение А. Паттерн enter/exit библиотеки D3	608
Метод enter	609
Доступ к привязанным данным	613
Об авторе	616
Послесловие	617
Алфавитный указатель	618

Предисловие

Главная цель этой книги — представить тулчейн (англ. toolchain, «цепочка инструментов») для визуализации данных (далее также — визуализация), который дает большие преимущества в эпоху интернета. Цель создания этого тулчейна — извлекать из полученных данных каждую крупную ценную информацию и передавать в браузер. После передачи вы можете делиться своими визуализациями со всем миром или с ограниченным кругом лиц (например: внутри локальной сети или с использованием аутентификации). Интернет открывает огромные возможности для визуализации, и будущее этой области тесно связано с JavaScript, лучшим языком для веб-разработки. Однако JavaScript не располагает средствами для предварительной обработки сырых данных, поэтому к процессу визуализации требуется привлекать другие языки программирования. Я надеюсь, что прочитав эту книгу, вы согласитесь со мной: Python — наиболее подходящий язык для совместной работы с JavaScript для визуализации данных в браузере.

Хотя книга получилась довольно объемной (что автор чувствует особенно остро), в ней не удалось охватить все замечательные инструменты Python и JavaScript для визуализации. Пришлось сосредоточиться на тех инструментах, которые формируют основу наиболее эффективных решений. Большое число полезных библиотек, оставшихся за рамками книги, подчеркивает жизнеспособность экосистемы Data Science на базе Python и JavaScript. Пока писалась книга, появились новые отличные библиотеки на обоих языках, и этот процесс продолжается.

Любая визуализация данных подразумевает их трансформацию. Чтобы продемонстрировать основные инструменты визуализации, рассмотрим превращение одного способа отображения набора данных (с помощью списков и HTML-таблиц) в другой: более современный, интерактивный и наглядный, основанный на браузере. Наша задача — преобразовать базовый список лауреатов Нобелевской премии, взятый из Википедии, в современную интерактивную визуализацию в браузере. Таким образом, тот же самый набор данных будет представлен в более доступной и привлекательной форме.

Чтобы создать на основе сырых данных интерактивную визуализацию с широкими возможностями, нам понадобятся лучшие в своем классе инструменты. Для начала необходимо получить набор данных. Иногда мы получаем его

от коллег или друзей, но, чтобы немного усложнить задачу и отработать важные навыки, научимся использовать *скрейпинг* наборов данных из интернета, в данном случае со страниц Википедии о Нобелевской премии. Для этого воспользуемся мощной Python-библиотекой Scrapy. Затем полученный набор сырых данных потребуется очистить и проанализировать, и для этого нет равных библиотеке pandas из экосистемы Python. Pandas в связке с Matplotlib и Jupyter Notebook — золотой стандарт для такого рода аналитики. Из очищенных и проанализированных данных, сохраненных в SQL-формате с помощью SQLAlchemy или SQLite, выберем интересные для визуализации аспекты. Я расскажу, как использовать Matplotlib и Plotly для встраивания статических и динамических диаграмм из Python в веб-страницы. Однако лучшей библиотекой для масштабной веб-визуализации остается D3 на основе JavaScript. Мы познакомимся с основами D3, создавая визуализацию данных о лауреатах Нобелевской премии.

В книге представлен набор инструментов, формирующий цепочку, а связующей нитью выступает визуализация данных о лауреатах Нобелевской премии. Структура книги позволяет легко находить главы по интересующему вас вопросу. Каждый раздел является самостоятельным, что помогает быстро отыскать и вспомнить пройденный материал.

Книга содержит пять разделов. Первый является введением в базовый набор инструментов Python и JavaScript для визуализации. В остальных четырех показано, как собирать и очищать сырые данные, анализировать их и превращать в современную веб-визуализацию. Давайте кратко сформулируем, какие основные уроки можно извлечь из каждого раздела.

Раздел I. Базовый пакет инструментов

О чем этот раздел:

- Обучающий мостик между Python и JavaScript, создан, чтобы сгладить переход между языками, подчеркнуть их сходные элементы и подготовить окружение для создания современной визуализации с помощью обоих языков. В последней версии JavaScript¹ появилось еще больше общего с Python, поэтому переключаться с одного на другой стало проще.
- Одна из сильных сторон Python — чтение/запись основных форматов обмена данными (например, JSON и CSV), а также поддержка баз данных (как SQL, так и NoSQL). Python легко передает данные, преобразуя их

¹ Есть много версий JavaScript на основе спецификации ECMAScript, но больше всего новых функциональных возможностей у ES6.

из одного формата в другой и меняя базы данных по мере необходимости. Такая гибкость в управлении данными — ключевой элемент, обеспечивающий плавную работу любого тулчейна визуализации.

- Мы также рассмотрим базовые навыки веб-разработки, которые необходимы для создания современной интерактивной визуализации в браузере. Чтобы минимизировать рутинное веб-программирование и сосредоточиться на разработке ваших визуальных проектов, мы не станем делать сложный сайт, ограничимся одностраничным веб-приложением (Single-Page Application, SPA) на JavaScript. Введение в SVG (Scalable Vector Graphics, «язык разметки векторной графики»), на котором в основном строятся D3-визуализации, — подготовка к созданию визуализации данных по Нобелевской премии в разделе IV.

Раздел II. Подготовка данных

В этой части книги мы рассмотрим, как самому получить данные из интернета с помощью Python, если вам не предоставили готовый файл с чистыми данными:

- Если вам повезло, и в открытом доступе есть такой файл в подходящем формате, например JSON или CSV, то достаточно отправить простой HTTP-запрос. Кроме того, для вашего набора данных может отыскаться специальный web API, хорошо, если это будет RESTful API. В качестве примера мы рассмотрим применение Python-библиотеки Tweepy для доступа к Twitter API. Мы также увидим, как использовать Google Таблицы (Google Spreadsheets), популярный инструмент для обмена данными в визуализации.
- Если же данные представлены в интернете в формате, ориентированном на людей, например: в виде HTML-таблицы, списков или структурированного контента, то задача усложняется. Тогда для извлечения сырого HTML-контента придется использовать *скрейпинг*, а затем с помощью парсера извлечь из полученных данных нужную информацию. Мы рассмотрим, как использовать для скрейпинга легковесную библиотеку Beautiful Soup и куда более многофункциональную и тяжеловесную Scrapy, самую крупную звезду веб-скрейпинга на небосклоне Python.

Раздел III. Очистка и исследование данных с помощью pandas

В этом разделе для очистки и исследования наборов данных мы задействуем «тяжелую артиллерию» — Python-библиотеку pandas. Сначала мы рассмотрим

pandas как часть экосистемы NumPy, которая предоставляет доступ к мощным низкоуровневым библиотекам для быстрой обработки массивов данных. Особое внимание уделим использованию pandas для очистки и анализа набора данных о лауреатах Нобелевской премии:

- Даже те данные, которые получены через официальные web API, в основном грязные. Чтобы очистить их и подготовить для визуализации, потребуется гораздо больше времени, чем вы, вероятно, ожидаете. Мы возьмем наш тренировочный набор данных о лауреатах Нобелевской премии и постепенно очистим его. Найдем и удалим неточные даты, аномальные типы данных, пропуски и прочую «грязь», прежде чем приступить к исследованию данных и их последующей визуализации.
- Очистив (насколько сумеем) набор данных о Нобелевской премии, мы увидим, как просто с помощью pandas и Matplotlib интерактивно исследовать данные, создавать диаграммы со всевозможными срезами данных, а также получить общее представление о них и отыскать ценную информацию, которую вы хотите донести до пользователя с помощью визуализации.

Раздел IV. Доставка данных

Здесь мы разберемся, как с помощью Flask создать минимальный API, чтобы передавать в браузер как статический, так и динамический контент.

Сначала посмотрим, как использовать Flask для работы со статическими файлами, а затем, как запустить собственный базовый API для данных из локальной БД. Минимализм Flask позволяет создать очень тонкий сервисный слой между результатами обработки данных с помощью Python и их конечной визуализацией в браузере.

Прелесть открытого ПО в том, что всегда можно найти надежную и простую в использовании библиотеку, которая решит вашу задачу лучше, чем если бы вы делали все вручную. Во второй главе раздела рассмотрим, насколько просто использовать лучшие в своем классе Python-библиотеки (на примере Flask) при создании надежного и гибкого RESTful API для обслуживания данных онлайн. Мы также рассмотрим простое развертывание сервера данных на облачной платформе Heroku, популярной среди Python-разработчиков.

Раздел V. Визуализация данных с помощью D3 и Plotly

В первой главе мы рассмотрим, как из данных, отобранных после анализа с помощью `pandas`, создать диаграммы или карты и опубликовать их в интернете. Статические диаграммы полиграфического качества мы сделаем с помощью `Matplotlib`, а интерактивные элементы и динамические диаграммы — с помощью `Plotly`. Мы рассмотрим, как вызвать формирование диаграммы `Plotly` из `Jupyter Notebook` и отобразить полученную диаграмму на веб-странице.

Часть, посвященная D3 — одна из самых сложных в книге, но D3 незаменима, если нужно создавать многоэлементные визуализации. Одним из плюсов библиотеки D3 является возможность найти в интернете множество примеров ее применения, хотя большинство из них демонстрируют только какую-то одну технику. Примеров, которые показывают, как организовать взаимодействие нескольких визуальных элементов, очень мало. В главах о D3 мы разберем, как синхронизировать обновление временной диаграммы (отображающей все вручения Нобелевской премии), карты, столбчатой диаграммы и списка лауреатов, когда пользователь применяет фильтры или меняет показатель присуждения премии (абсолютный или на душу населения).

Эти главы позволят вам дать волю воображению и учиться на практике. Я бы порекомендовал выбрать интересные для вас данные и на их основе разработать что-нибудь с помощью D3.

Второе издание

Я с некоторым сомнением воспринял предложение издательства O'Reilly поработать над вторым изданием этой книги. Первое издание получилось объемнее, чем ожидалось, и на его доработку могло потребоваться немало труда. Однако, когда я проверил актуальность описанных в книге библиотек и изменений, касающихся визуализации, в экосистемах Python и JavaScript, выяснилось, что большинство библиотек (например: `Scrapy`, `NumPy`, `pandas`) остаются отличными вариантами, и существенных изменений текста не требуется.

Больше всего изменилась библиотека D3, но при этом ее стало проще использовать и легче изучать. Модульность стала стандартом разработки на JavaScript, что сделало JS-код чище и привычнее для питонистов.

Выбор нескольких Python-библиотек теперь выглядит менее удачным, а пара из них попросту устарела. В первом издании довольно подробно рассматривалась `MongoDB` — база данных NoSQL, но теперь я считаю, что старый добрый SQL лучше подходит для работы с визуализацией, а легкая

однофайловая бессерверная SQLite — идеальное решение, если для визуализации требуется БД.

Вместо замены устаревшего RESTful-сервера на другую Python-библиотеку, я решил показать, как создать простой сервер с нуля, используя такие замечательные библиотеки на Python, как `marshmallow`, которые полезны во многих сценариях визуализации.

С учетом времени, отведенного на обновление книги, я решил показать исследования и анализ с помощью `Matplotlib` и `pandas` на примере набора данных из первого издания, сосредоточившись на обновлении всех библиотек до версий, актуальных на середину 2022 года. Это позволило мне выделить время на изложение нового материала и, самое главное, — написать главу о `Plotly`. Эта библиотека на Python упрощает перенос наработок из `Jupyter Notebook` в интерактивное веб-представление. Особое преимущество этого подхода — доступ к картам богатой картографической экосистемы `Mapbox`.

Основной упор во втором издании я делал на следующем:

- обновить все библиотеки;
- удалить и/или заменить библиотеки, которые не выдержали испытания временем;
- добавить новый материал, связанный с изменениями в быстроразвивающемся мире визуализации с помощью Python и JavaScript.

Я считаю, что концепция тулчейна для визуализации осталась в силе, и конвейер преобразований — от сырых, необработанных веб-данных через исследовательский анализ до безупречной веб-визуализации — остается прекрасным способом изучения ключевых инструментов.

Принятые в книге обозначения

Типографские соглашения:

Курсивом

выделяются новые термины, URL-адреса, адреса электронной почты, имена и расширения файлов.

Моноширинным шрифтом

выделен код программы, а также встречающиеся в тексте программные элементы: переменные, имена функций и баз данных, типы данных, ключевые слова, операторы и переменные окружения.

Полужирным моноширинным шрифтом

выделены команды и другой текст, который пользователь должен ввести без изменений.

Моноширинным курсивом

выделен текст, который надо заменить на заданные пользователем значения, или на значения, определяемые контекстом.



Этот элемент означает подсказку или предложение.



Этот элемент означает общее примечание.



Этот элемент означает предупреждение или предостережение.

Использование примеров кода

Дополнительные материалы (примеры кода, упражнения и т. д.) доступны для скачивания по адресу <https://github.com/Kyrand/dataviz-with-python-and-js-ed-2>.

Предназначение этой книги — помочь вам решить свои задачи. Как правило, вы можете использовать примеры кода из этой книги в своих программах и документации. Вам не нужно обращаться к нам за разрешением, кроме тех случаев, когда вы собираетесь воспроизвести значительную часть кода. Например, вам не требуется разрешение, если вы пишете программу, в которой используются несколько фрагментов кода из данной книги. Однако для продажи или распространения компакт-диска с примерами из книг O'Reilly разрешение требуется. Цитировать эту книгу с примерами из кода вы можете свободно, но если вы собираетесь включить значительное число примеров кода из книги в документацию по своему продукту, вам следует обратиться к нам за разрешением.

Благодарности

Прежде всего хочу поблагодарить Меган Бланшет (Meghan Blanchette), которая положила начало этой книге и помогла мне с самыми трудными главами. Затем Дон Шанафелт (Dawn Schanafelt) взяла бразды правления в свои руки и проделала большую часть необходимой редакторской работы. Кристен Браун (Kristen Brown) блестяще подготовила книгу к печати, в чем ей помогла стальная хватка литературного редактора Джилиана МакГарви (Gillian McGarvey). Работа с этими талантливыми, преданными делу профессионалами была для меня не только честью и привилегией, но и обучением: если бы я с самого начала знал все, чему научился от них, мне было бы гораздо легче писать книгу. Но ведь так всегда и бывает?

Огромное спасибо Эми Зелински (Amy Zielinski), благодаря которой автор выглядел лучше, чем он того заслуживает.

Книга существенно улучшилась благодаря ряду ценных замечаний Кристофа Вио (Christophe Viau), Тома Парслоу (Tom Parslow), Питера Кука (Peter Cook), Иэна Макиннеса (Ian Macinnes) и Иэна Освальда (Ian Ozsvald).

Я также хотел бы поблагодарить отважных охотников за ошибками, допущенными в черновике книги. Это Дуглас Келли (Douglas Kelley), Павел Сук (Pavel Suk), Брайам Хаусман (Brigham Hausman), Марко Хемкен (Marco Hemken), Нобль Кеннамер (Noble Kennamer), Манфреди Бьясутти (Manfredi Biasutti), Мэтью Мальдонадо (Matthew Maldonado) и Хирт Боувенс (Geert Bauwens).

Второе издание

Прежде всего, я должен поблагодарить Ширу Эванс (Shira Evans) за сопровождение книги от замысла до реализации, и Грегори Хаймана (Gregory Human), который держал меня в курсе дел по черновикам и предоставлял обратную связь. Мне вновь посчастливилось работать с Кристен Браун (Kristen Brown), именно она подготовила к печати второе издание книги.

Также благодарю технических рецензентов Джордана Голдмайера (Jordan Goldmeier), Дрю Уинстела (Drew Winstel) и Джесс Мэйлс (Jess Males) за отличные советы.