

JavaScript

Рецепты для разработчиков

3-е издание

Адам Д. Скотт, Мэтью Макдоналд,
Шелли Пауэрс



Санкт-Петербург • Москва • Минск

2026

Краткое содержание

Предисловие	15
-------------------	----

Часть I ЯЗЫК JAVASCRIPT

Глава 1. Настройка среды разработки	20
Глава 2. Строки и регулярные выражения	56
Глава 3. Числа	84
Глава 4. Даты	98
Глава 5. Массивы	110
Глава 6. Функции	141
Глава 7. Объекты	169
Глава 8. Классы	198
Глава 9. Асинхронное программирование	227
Глава 10. Ошибки и тестирование	254

Часть II JAVASCRIPT В БРАУЗЕРЕ

Глава 11. Инструментарий браузера	280
Глава 12. Работа с HTML	289
Глава 13. Получение удаленных данных	327
Глава 14. Сохранение данных	351
Глава 15. Работа с мультимедиа	371
Глава 16. Создание веб-приложений	387

Часть III
NODE.JS

Глава 17. Основы Node	420
Глава 18. Модули Node	442
Глава 19. Управление экосистемой Node	462
Глава 20. Удаленные данные	481
Глава 21. Построение веб-приложений с помощью Express	488
Об авторах	524
Иллюстрация на обложке	525

Оглавление

Предисловие	15
Аудитория этой книги	15
Структура издания	16
Условные обозначения	16
Примеры кода	17
Благодарности	18
От издательства	18

Часть I ЯЗЫК JAVASCRIPT

Глава 1. Настройка среды разработки	20
1.1. Выбираем редактор кода	21
1.2. Использование консоли разработки в браузере	23
1.3. Выполнение фрагментов кода в консоли разработчика	27
1.4. Использование строгого режима для выявления типичных ошибок	29
1.5. Вставка блоков HTML с помощью сокращенных команд Emmet	31
1.6. Установка менеджера пакетов npm (с Node.js)	33
1.7. Загрузка пакета с помощью npm	36
1.8. Обновление пакета с помощью npm	40
1.9. Настройка локального тестового сервера	42
1.10. Соблюдение стандартов кодирования с помощью статического анализатора	45
1.11. Согласованное оформление кода с помощью форматировщика	50
1.12. Эксперименты в интерактивной среде JavaScript	52
Глава 2. Строки и регулярные выражения	56
2.1. Проверка того, что строка существует и она не пуста	56
2.2. Преобразование числового значения в форматированную строку	59

2.3. Вставка специальных символов	61
2.4. Использование эмодзи	63
2.5. Использование шаблонных литералов для более наглядной конкатенации строк	65
2.6. Сравнение строк без учета регистра	67
2.7. Проверка того, содержит ли строка заданную подстроку	68
2.8. Замена всех вхождений строки	69
2.9. Замена тегов HTML на именованные сущности	70
2.10. Использование регулярных выражений для создания шаблонов при замене строк.....	71
2.11. Извлечение списка из строки	74
2.12. Поиск по шаблону.....	76
2.13. Удаление пробелов в начале и в конце строки	80
2.14. Замена первой буквы строки на прописную	81
2.15. Валидация адреса электронной почты.....	82

Глава 3. Числа84

3.1. Генерирование случайных чисел	84
3.2. Генерирование криптографически надежных случайных чисел.....	86
3.3. Округление до заданного десятичного разряда	88
3.4. Сохранение точности в дробных числах.....	89
3.5. Преобразование строки в число	91
3.6. Преобразование десятичных значений в шестнадцатеричные	93
3.7. Преобразование градусов в радианы	94
3.8. Вычисление длины дуги окружности.....	94
3.9. Манипуляции с очень большими целыми числами в формате BigInt	95

Глава 4. Даты.....98

4.1. Получение текущих даты и времени.....	98
4.2. Преобразование строки в дату.....	100
4.3. Добавляем дни к дате.....	102
4.4. Сравнение дат и проверка двух дат на равенство.....	103
4.5. Вычисление времени, прошедшего между двумя датами	105
4.6. Представление даты в виде форматированной строки.....	107

Глава 5. Массивы	110
5.1. Проверка того, является ли объект массивом.....	110
5.2. Перебор всех элементов массива	111
5.3. Проверка равенства двух массивов	113
5.4. Разбиение массива на отдельные переменные	116
5.5. Передача массива в функцию, которая принимает список значений	117
5.6. Клонирование массива.....	118
5.7. Слияние двух массивов	120
5.8. Копирование части массива, выбранной по положению элемента	122
5.9. Извлечение из массива элементов, удовлетворяющих заданному условию	123
5.10. Очистка массива.....	124
5.11. Удаление дубликатов	125
5.12. Сведение двумерного массива.....	126
5.13. Точный поиск элементов массива	128
5.14. Поиск элементов массива, удовлетворяющих заданному критерию.....	129
5.15. Удаление и замена элементов массива.....	131
5.16. Сортировка массива объектов по заданному свойству	132
5.17. Преобразование элементов массива	134
5.18. Использование всех элементов массива в одном вычислении	134
5.19. Проверка содержимого массива.....	136
5.20. Построение коллекции недублирующихся значений	137
5.21. Создание коллекции элементов, индексированной по ключу.....	139
 Глава 6. Функции.....	 141
6.1. Передача одной функции в другую в качестве аргумента.....	141
6.2. Использование стрелочных функций.....	145
6.3. Предоставление значения параметра по умолчанию	148
6.4. Создание функции, принимающей неограниченное число аргументов ...	149
6.5. Использование именованных параметров функции	150
6.6. Создание функции с сохранением состояния посредством замыкания....	153
6.7. Создание функции-генератора, которая возвращает несколько значений.....	155
6.8. Уменьшение избыточности за счет частичного применения.....	160

6.9. Фиксация this посредством привязки функций.....	163
6.10. Реализация рекурсивного алгоритма	166

Глава 7. Объекты.....169

7.1. Проверка того, относится ли объект к заданному типу	169
7.2. Объединение данных с помощью объектных литералов	171
7.3. Проверка существования свойства у объекта	174
7.4. Перебор всех свойств объекта	176
7.5. Проверка того, является ли объект пустым	178
7.6. Объединение свойств двух объектов.....	180
7.7. Выбор способа определения свойств	181
7.8. Запрет любых изменений объекта	184
7.9. Перехват и изменение объектов с помощью прокси-объектов.....	185
7.10. Клонирование объектов	189
7.11. Создание глубокой копии объекта.....	191
7.12. Создание абсолютно уникальных ключей для свойств объекта	193
7.13. Создание перечислений с помощью Symbol.....	195

Глава 8. Классы198

8.1. Создание класса для многократного использования	198
8.2. Добавление в класс новых свойств	202
8.3. Улучшенное строковое представление класса.....	207
8.4. Создание произвольного класса посредством шаблона «Конструктор»	208
8.5. Создание возможности для объединения методов класса в цепочку	211
8.6. Создание статических методов класса.....	213
8.7. Создание объектов посредством статических методов	216
8.8. Наследование функционала другого класса	218
8.9. Объединение классов JavaScript в модули.....	223

Глава 9. Асинхронное программирование.....227

9.1. Обновление страницы в цикле.....	228
9.2. Использование функции, которая возвращает промис.....	230
9.3. Замена асинхронной функции с обратным вызовом на промис	234
9.4. Конкурентное выполнение нескольких промисов	237
9.5. Ожидание выполнения промиса с помощью await и async.....	240

9.6. Создание асинхронной функции-генератора.....	244
9.7. Выполнение фоновых задач с помощью Web Worker.....	247
9.8. Поддержка сообщений о ходе выполнения задач в Web Worker.....	251

Глава 10. Ошибки и тестирование254

10.1. Обнаружение и обезвреживание ошибок.....	254
10.2. Перехват различных типов ошибок.....	257
10.3. Перехват асинхронных ошибок.....	259
10.4. Обнаружение необработанных ошибок.....	261
10.5. Выдача обычной ошибки.....	264
10.6. Выдача нестандартных ошибок.....	266
10.7. Написание модульных тестов для кода.....	268
10.8. Отслеживание покрытия кода тестами.....	276

Часть II JAVASCRIPT В БРАУЗЕРЕ

Глава 11. Инструментарий браузера280

11.1. Отладка кода JavaScript.....	280
11.2. Анализ производительности во время выполнения кода.....	282
11.3. Обнаружение неиспользуемого кода JavaScript.....	284
11.4. Выдача наилучших рекомендаций посредством Lighthouse.....	286

Глава 12. Работа с HTML289

12.1. Доступ к определенному элементу, поиск его родительского и дочерних элементов.....	289
12.2. Перебор результатов, полученных от querySelectorAll(), с помощью forEach().....	292
12.3. Привязка к элементу действия в ответ на щелчок.....	293
12.4. Поиск всех элементов с данным атрибутом.....	295
12.5. Выбор всех элементов определенного типа.....	296
12.6. Исследование дочерних элементов с помощью Selectors API.....	298
12.7. Изменение класса элемента.....	300
12.8. Присвоение элементу атрибута style.....	300
12.9. Создание абзаца и вставка в него текста.....	303
12.10. Вставка нового элемента в определенной точке DOM.....	304

12.11. Проверка того, установлен ли флажок	305
12.12. Вставка значений в таблицу HTML	306
12.13. Удаление строк из таблицы HTML	309
12.14. Скрытие частей страницы	312
12.15. Создание окон, всплывающих по наведению указателя мыши	313
12.16. Валидация данных формы	315
12.17. Выделение ошибочно заполненных полей форм и реализация специальных возможностей	319
12.18. Создание автоматически обновляемой области, доступной на специализированных устройствах	325
Глава 13. Получение удаленных данных	327
13.1. Запрос удаленных данных с помощью Fetch API	327
13.2. Использование метода XMLHttpRequest	331
13.3. Отправка данных формы	332
13.4. Заполнение списка выбора данными, полученными с сервера	336
13.5. Синтаксический анализ данных, полученных в формате JSON	340
13.6. Получение и синтаксический анализ данных в формате XML	342
13.7. Передача двоичных данных и загрузка изображения	344
13.8. Обмен HTTP cookies между несколькими доменами	345
13.9. Двухнаправленный обмен данными между клиентом и сервером посредством WebSockets	346
13.10. Длинный опрос удаленного источника данных	348
Глава 14. Сохранение данных	351
14.1. Сохранение информации в cookies	351
14.2. Хранение данных на стороне клиента с помощью sessionStorage	354
14.3. Создание хранилища данных на стороне клиента на основе localStorage	360
14.4. Сохранение больших объемов данных на стороне клиента с помощью IndexedDB	364
14.5. Упрощение IndexedDB с помощью библиотеки	367
Глава 15. Работа с мультимедиа	371
15.1. JavaScript для SVG	371
15.2. Доступ к SVG из скрипта веб-страницы	374
15.3. Построение столбчатой диаграммы в формате SVG с помощью библиотеки D3	376

15.4. Интеграция элементов SVG и Canvas в HTML.....	379
15.5. Выполнение процедуры в начале воспроизведения аудиофайла	382
15.6. Управление отображением видео с помощью элемента video и JavaScript	383
Глава 16. Создание веб-приложений.....	387
16.1. Создание пакетов JavaScript	387
16.2. JavaScript для мобильного интернета	389
16.3. Создание прогрессивного веб-приложения.....	392
16.4. Тестирование и профилирование прогрессивных веб-приложений	399
16.5. Получение текущего URL.....	403
16.6. Перенаправление на другой URL	404
16.7. Копирование текста в буфер обмена.....	405
16.8. Вывод на стационарном компьютере таких же уведомлений, как на мобильном устройстве.....	407
16.9. Открытие в браузере файла с локального устройства.....	410
16.10. Расширение возможностей с помощью Web Components	413
16.11. Выбор фреймворка для разработки на стороне клиента	416
 Часть III NODE.JS	
Глава 17. Основы Node.....	420
17.1. Управление версиями Node с помощью Node Version Manager	420
17.2. Ответ на простой запрос браузера.....	423
17.3. Интерактивная проверка кода Node с помощью REPL.....	425
17.4. Чтение данных из файла и запись данных в файл	428
17.5. Получение данных из терминала	433
17.6. Получение пути к выполняемому скрипту	435
17.7. Работа с таймерами и циклом событий Node	436
Глава 18. Модули Node	442
18.1. Поиск нужного модуля Node через npm	442
18.2. Преобразование библиотеки в модуль Node.....	444
18.3. Перенос кода в модульную среду.....	445
18.4. Создание устанавливаемого модуля Node	448

18.5. Создание мультиплатформенных библиотек	454
18.6. Тестирование модулей	458
Глава 19. Управление экосистемой Node	462
19.1. Использование переменных среды	462
19.2. Что делать с адом обратных вызовов	464
19.3. Доступ к функциям командной строки из приложений Node	467
19.4. Передача аргументов в командную строку	470
19.5. Создание утилиты командной строки с подсказкой с помощью Commander	471
19.6. Обеспечение работоспособности экземпляра Node	474
19.7. Отслеживание изменений и перезапуск приложения в процессе разработки на локальном компьютере	475
19.8. Многократное выполнение задач по расписанию	477
19.9. Тестирование производительности и возможностей приложения WebSockets	478
Глава 20. Удаленные данные	481
20.1. Получение удаленных данных	481
20.2. Анализ экранных данных	483
20.3. Доступ к данным в формате JSON посредством RESTful API	485
Глава 21. Построение веб-приложений с помощью Express	488
21.1. Использование Express для ответов на запросы	488
21.2. Использование Express-Generator	492
21.3. Задача маршрутизации	497
21.4. Работа с OAuth	499
21.5. Аутентификация пользователей в OAuth 2 с помощью Passport.js	510
21.6. Обработка форматированных данных	514
21.7. Построение RESTful API	516
21.8. Построение API GraphQL	519
Об авторах	524
Иллюстрация на обложке	525