

МЭТТ
ФРИСБИ

Java Script

для
ПРОФЕССИОНАЛЬНЫХ
ВЕБ-РАЗРАБОТЧИКОВ

4-Е МЕЖДУНАРОДНОЕ
ИЗДАНИЕ



PROFESSIONAL

JavaScript® for Web Developers

Matt Frisbie



МЭТТ ФРИСБИ

JavaScript

ДЛЯ ПРОФЕССИОНАЛЬНЫХ
ВЕБ-РАЗРАБОТЧИКОВ

4-Е МЕЖДУНАРОДНОЕ
ИЗДАНИЕ



Санкт-Петербург • Москва • Минск

2022

ББК 32.988.02-018.1
УДК 004.738.5
Ф89

Фрисби М.

Ф89 JavaScript для профессиональных веб-разработчиков. 4-е международное изд. — СПб.: Питер, 2022. — 1168 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-1740-6

Самое полное руководство по современному JavaScript.

Как максимально прокачать свои навыки и стать топовым JS-программистом? Четвертое издание «JavaScript для профессиональных веб-разработчиков» идеально подойдет тем, кто уже имеет базовые знания и опыт разработки на JavaScript. Автор сразу переходит к техническим деталям, которые сделают ваш код чистым и переведут вас с уровня рядового кодера на высоту продвинутого разработчика.

Рост мобильного трафика увеличивает потребность в адаптивном динамическом веб-дизайне, а изменения в JS-движках происходят постоянно, так что каждый веб-разработчик должен постоянно обновлять свои навыки работы с JavaScript.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.988.02-018.1
УДК 004.738.5

Права на издание получены по соглашению с John Wiley & Sons, Inc. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги. Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1119366447 англ.
ISBN 978-5-4461-1740-6

© 2020 by John Wiley & Sons, Inc., Indianapolis, Indiana
© Перевод на русский язык ООО Издательство «Питер», 2022
© Издание на русском языке, оформление ООО Издательство «Питер», 2022
© Серия «Для профессионалов», 2022
© Павлов А., перевод с английского языка, 2020

Краткое содержание

Об авторе	26
О научных редакторах	27
Благодарности	28
Предисловие	29
Введение	32
Глава 1. Что такое JavaScript?	38
Глава 2. JavaScript в HTML	51
Глава 3. Основы языка	64
Глава 4. Переменные, область видимости и память	144
Глава 5. Ссылочные типы	170
Глава 6. Ссылочные типы коллекций	214
Глава 7. Итераторы и генераторы	274
Глава 8. Объекты, классы и объектно-ориентированное программирование	300
Глава 9. Прокси и Reflect	374
Глава 10. Функции	401
Глава 11. Промисы и асинхронные функции	447
Глава 12. Объектная модель браузера	497

Глава 13. Распознавание клиента.....	527
Глава 14. Объектная модель документа.....	555
Глава 15. Расширения DOM.....	611
Глава 16. DOM Level 2 и 3.....	631
Глава 17. События.....	670
Глава 18. Анимация и рисование на холсте.....	744
Глава 19. Работа с формами	784
Глава 20. API в JavaScript	821
Глава 21. Обработка ошибок и отладка	906
Глава 22. XML в JavaScript	933
Глава 23. JSON.....	945
Глава 24. Сетевые запросы и удаленные ресурсы	956
Глава 25. Клиентское хранилище.....	1010
Глава 26. Модули.....	1038
Глава 27. Рабочие потоки	1064
Глава 28. Лучшие практики.....	1133

Оглавление

Об авторе	26
О научных редакторах	27
Благодарности	28
Предисловие	29
Введение	32
Целевая аудитория	33
Темы, рассматриваемые в книге	33
Структура	34
Что нужно для эффективной работы с книгой	37
От издательства	37
 Глава 1. Что такое JavaScript?	38
Краткая история JavaScript	39
Реализации JavaScript	40
ECMAScript	40
Объектная модель документа	45
Объектная модель браузера	48
Версии JavaScript	49
Итоги	50
 Глава 2. JavaScript в HTML	51
Элемент <script>	51
Расположение тегов	54
Отложенные сценарии	55
Асинхронные сценарии	56
Динамическая загрузка сценариев	57
Изменения в XHTML	57
Устаревший синтаксис	59
Встроенный код или внешние файлы?	60
Режимы документа	61
Элемент <noscript>	62
Итоги	63

Глава 3. Основы языка	64
Синтаксис	64
Чувствительность к регистру	65
Идентификаторы	65
Комментарии	65
Строгий режим	66
Инструкции	66
Ключевые и зарезервированные слова	67
Переменные	68
Ключевое слово var	68
Объявления let	70
Объявления const	74
Типы данных	76
Оператор typeof	76
Тип Undefined	77
Тип Null	78
Тип Boolean	79
Тип Number	80
Тип String	87
Тип Symbol	94
Тип Object	108
Операторы	109
Унарные операторы	110
Поразрядные операторы	113
Логические операторы	119
Мультипликативные операторы	122
Оператор возведения в степень	124
Операторы сложения и вычитания	124
Операторы отношений	126
Операторы эквивалентности	128
Условный оператор	130
Операторы присваивания	130
Оператор «запятая»	131
Инструкции	131
Инструкция if	131
Инструкция do-while	132
Инструкция while	133
Инструкция for	133
Инструкция for-in	134
Инструкция for-of	135

Метки инструкций.....	135
Инструкции break и continue	135
Инструкция with	137
Инструкция switch.....	138
Функции.....	140
Итоги	142
Глава 4. Переменные, область видимости и память.....	144
Примитивные и ссылочные значения	144
Динамические свойства	145
Копирование значений.....	146
Передача аргументов.....	147
Проверка типа	149
Контекст выполнения и область видимости	150
Приращение цепочки областей видимости.....	152
Объявление переменной.....	153
Сборка мусора.....	158
Отслеживание и очистка	159
Подсчет ссылок	160
Производительность	161
Управление памятью.....	162
Итоги	168
Глава 5. Ссылочные типы	170
Тип Date	171
Унаследованные методы	173
Методы форматирования дат.....	174
Методы для работы с компонентами даты/времени	174
Тип RegExp.....	176
Свойства экземпляра RegExp	179
Методы экземпляра RegExp.....	180
Свойства конструктора RegExp	182
Ограничения шаблонов.....	184
Оболочки примитивных типов.....	184
Тип Boolean.....	186
Тип Number.....	187
Тип String.....	189
Встроенные одиночные объекты	204
Объект Global	204
Объект Math.....	208
Итоги	212

Глава 6. Ссылочные типы коллекций	214
Тип Object	214
Тип Array	217
Создание массивов	217
Дыры в массивах	220
Индексирование в массивы	221
Идентификация массивов	223
Методы итераторов	223
Методы копирования и заполнения	224
Методы преобразования массивов	226
Методы для работы с массивом как со стеком	228
Методы для работы с массивом как с очередью	229
Методы изменения порядка следования элементов	230
Методы манипулирования элементами	232
Методы поиска элементов	234
Методы перебора элементов	236
Методы редукции массивов	238
Типизированные массивы	238
История	239
Использование типа ArrayBuffer	240
Тип DataView	241
Типизированные массивы	245
Тип Map	250
Базовый API	250
Порядок и перебор значений	252
Выбор между Object и Map	255
Тип WeakMap	256
Базовый API	256
Слабые ключи	257
Неитерируемые ключи	258
Полезные стратегии	259
Тип Set	261
Базовый API	261
Порядок и перебор значений	263
Определение формальных операций над Set	265
Тип WeakSet	267
Базовый API	267
Слабые ключи	268
Неитерируемые значения	269

Полезные стратегии.....	269
Итераторы и операторы распространения.....	270
Итоги	272
Глава 7. Итераторы и генераторы.....	274
Введение в итерацию	274
Паттерн Итератор	276
Протокол Iterable	276
Протокол Iterator	279
Определение пользовательского итератора.....	281
Преждевременное завершение итератора.....	283
Генераторы.....	285
Основы генераторов	285
Прерывание выполнения с помощью yield	287
Использование генератора в качестве итератора по умолчанию	296
Преждевременное завершение генераторов	297
Итоги	299
Глава 8. Объекты, классы и объектно-ориентированное программирование	300
Общие сведения об объектах	300
Типы свойств.....	301
Определение нескольких свойств	305
Чтение атрибутов свойств.....	305
Слияние объектов	307
Идентичность и равенство объектов	310
Расширенный синтаксис объектов.....	310
Деструктурирование объектов	314
Создание объектов.....	318
Обзор	319
Паттерн Фабрика	319
Паттерн Конструктор функции	320
Паттерн Прототип.....	324
Итерация по объекту.....	335
Наследование	340
Цепочки прототипов	340
Кража конструктора.....	346
Комбинированное наследование	347
Прототипное наследование	348
Паразитное наследование.....	350
Паразитное комбинированное наследование.....	351

Классы.....	353
Основы определения классов	354
Конструктор класса	355
Члены экземпляра, прототипа и класса	360
Наследование.....	364
Итоги	372

Глава 9. Прокси и Reflect374

Основы прокси.....	375
Создание сквозного прокси	375
Определение ловушек	376
Параметры ловушек и Reflect API	377
Инварианты ловушек.....	379
Отзывные прокси	380
Использование Reflect API.....	381
Замещение прокси	382
Нюансы и недостатки прокси	383
Прокси-ловушки и методы Reflect	385
get()	385
set().....	386
has().....	387
defineProperty().....	388
getOwnPropertyDescriptor().....	388
deleteProperty().....	390
ownKeys().....	390
getPrototypeOf()	391
setPrototypeOf().....	392
isExtensible()	393
preventExtensions().....	394
apply().....	394
construct()	395
Паттерны для прокси.....	396
Отслеживание доступа к свойствам	396
Скрытие свойств.....	396
Проверка свойств	397
Проверка параметров функции и конструктора	397
Привязка данных и наблюдатели	398
Итоги	399

Глава 10. Функции401

Стрелочные функции	402
Имена функций	404

Аргументы функций	405
Аргументы в стрелочных функциях	407
Отсутствие перегрузки	408
Значения параметров по умолчанию	409
Область параметров по умолчанию и временная мертвая зона	410
Аргументы распространения и остаточные параметры	412
Аргументы распространения	412
Остаточные параметры	413
Объявления функции и функции-выражения	414
Функции как значения	415
Внутреннее устройство функций	417
arguments	417
this	418
caller	420
new.target	420
Свойства и методы функций	421
Функции-выражения	424
Рекурсия	426
Оптимизация с помощью хвостовых вызовов	427
Требования к оптимизации с помощью хвостовых вызовов	428
Код для оптимизации с помощью хвостовых вызовов	430
Замыкания	431
Объект this	434
Утечки памяти	436
Немедленно вызываемые функции-выражения	437
Закрытые переменные	439
Статические закрытые переменные	441
Паттерн Модуль	443
Расширенный паттерн Модуль	444
Итоги	445
Глава 11. Промисы и асинхронные функции	447
Введение в асинхронное программирование	448
Синхронный и асинхронный JavaScript	448
Устаревшие паттерны асинхронного программирования	449
Промисы	452
Спецификация Promises/A+	452
Основы промисов	452
Методы экземпляра промиса	458
Композиция и цепочки промисов	469
Расширения промисов	477

Асинхронные функции	480
Основы асинхронных функций.....	481
Остановка и возобновление выполнения	487
Стратегии для асинхронных функций	490
Итоги	496

Глава 12. Объектная модель браузера497

Объект window	498
Глобальная область видимости	498
Отношения окон.....	499
Положение окна и соотношение пикселей.....	499
Размеры окна	500
Положение области просмотра окна	502
Открытие окон и навигация	503
Интервалы и тайм-ауты	507
Системные диалоговые окна	509
Объект location.....	512
Аргументы строки запроса.....	513
Работа с объектом location	515
Объект navigator.....	516
Обнаружение подключаемых модулей.....	519
Регистрация обработчиков	521
Объект screen	522
Объект history	523
Навигация	523
Управление состоянием истории.....	524
Итоги	525

Глава 13. Распознавание клиента527

Распознавание возможностей.....	528
Надежное распознавание возможностей	529
Использование распознавания возможностей для анализа браузера	530
Распознавание пользовательского агента	532
История композиции пользовательского агента	533
Использование пользовательского агента для анализа браузера	542
Распознавание программного и аппаратного обеспечения.....	544
Идентификация браузера и операционной системы.....	545
Метаданные браузера.....	547
Аппаратное обеспечение	553
Итоги	554

Глава 14. Объектная модель документа.....	555
Иерархия узлов.....	556
Тип Node.....	557
Тип Document.....	562
Тип Element.....	571
Тип Text.....	579
Тип Comment.....	582
Тип CDATASection.....	583
Тип DocumentType.....	584
Тип DocumentFragment.....	584
Тип Attr.....	586
Работа с DOM.....	587
Динамические сценарии.....	587
Динамические стили.....	589
Работа с таблицами.....	591
Использование объектов NodeList.....	593
Наблюдатели за изменениями.....	595
Основные примеры использования.....	595
Управление областью наблюдения с помощью MutationObserverInit.....	601
Асинхронные обратные вызовы и очередь записи.....	607
Производительность, память и сборка мусора.....	608
Итоги.....	609
 Глава 15. Расширения DOM.....	 611
Selectors API.....	611
Метод querySelector().....	612
Метод querySelectorAll().....	612
Метод matches().....	613
Element Traversal.....	614
HTML5.....	615
Новые средства работы с классами.....	615
Управление фокусом.....	618
Изменения типа HTMLDocument.....	618
Свойства кодировки.....	619
Пользовательские атрибуты данных.....	619
Вставка разметки.....	620
Метод scrollIntoView().....	625
Фирменные расширения.....	626
Свойство children.....	626
Метод contains().....	626
Вставка разметки.....	627

Прокрутка	629
Итоги	630
Глава 16. DOM Level 2 и 3.....	631
Изменения DOM.....	632
XML-пространства имен	632
Другие изменения	636
Стили.....	640
Доступ к стилям элементов	640
Работа с таблицами стилей	644
Размеры элементов.....	648
Обход.....	652
Тип NodeIterator	654
Тип TreeWalker.....	657
Диапазоны	659
Диапазоны в DOM.....	659
Простое выделение с помощью DOM-диапазонов	660
Сложное выделение с помощью DOM-диапазонов	661
Работа с контентом DOM-диапазона.....	663
Вставка контента DOM-диапазона	665
Свертывание DOM-диапазона	666
Сравнение DOM-диапазонов	667
Клонирование DOM-диапазонов	668
Очистка.....	668
Итоги	668
Глава 17. События	670
Распространение событий	671
Всплытие событий	671
Перехват событий	672
Распространение DOM-событий	672
Обработчики событий.....	673
HTML-обработчики событий	673
Обработчики событий DOM Level 0	676
Обработчики событий DOM Level 2.....	677
Обработчики событий в Internet Explorer.....	678
Кроссбраузерные обработчики событий.....	679
Объект event	681
Объект event в DOM.....	681
Объект event в Internet Explorer	685
Кроссбраузерный объект event.....	687

Типы событий.....	689
События пользовательского интерфейса	690
События изменения фокуса	696
События мыши и колесика мыши	697
События клавиатуры и редактирования текста	706
События композиции	712
События изменения DOM-структуры.....	713
События HTML5.....	713
События устройств.....	720
События касаний и жестов	724
Справка по событиям.....	727
Память и быстроедействие	732
Делегирование событий.....	733
Удаление обработчиков событий.....	734
Имитация событий.....	736
Имитация DOM-событий	736
Имитация событий в Internet Explorer.....	742
Итоги	743

Глава 18. Анимация и рисование на холсте744

Использование requestAnimationFrame.....	745
Ранние анимационные циклы.....	745
Проблемы с интервалами	746
requestAnimationFrame.....	746
cancelAnimationFrame.....	747
Управление производительностью с помощью requestAnimationFrame.....	748
Основы работы с элементом <canvas>.....	749
Двухмерный контекст	750
Заливка и рисование контура	751
Рисование прямоугольников	751
Рисование путей	753
Рисование текста	755
Преобразования	757
Рисование изображений	760
Тени	761
Градиенты.....	762
Узоры	764
Работа с данными изображений.....	765
Композиция изображений	766

WebGL.....	768
Контекст WebGL	768
Основы WebGL	769
Сравнение WebGL1 и WebGL2	782
Итоги	783

Глава 19. Работа с формами884

Общие сведения о формах	784
Отправка данных формы	785
Сброс формы.....	786
Поля форм	787
Работа с текстовыми полями	792
Выделение текста	793
Фильтрация ввода.....	796
Автоматический переход по нажатию клавиши табуляции.....	799
API проверки ограничений в HTML5	801
Работа со списками.....	805
Выбор элементов списка.....	807
Добавление элементов в список.....	808
Удаление элементов списка	809
Перемещение и переупорядочение элементов списка	810
Сериализация форм	810
Редактирование форматированного текста.....	813
Атрибут contenteditable	813
Работа с форматированным текстом	814
Выделение форматированного текста	817
Форматированный текст в формах	819
Итоги	820

Глава 20. API в JavaScript.....821

Atoms и SharedArrayBuffer	822
SharedArrayBuffer.....	822
Основы использования Atomics.....	823
Кросс-контекстный обмен сообщениями.....	830
Encoding API.....	831
Кодировка текста.....	832
Декодирование текста.....	834
Blob и File API.....	837
Тип File.....	837
Тип FileReader	837

Тип FileReaderSync.....	839
Blobs и частичное чтение.....	840
URL объекта и Blob-объекты.....	841
Перетаскивание файла чтения.....	842
Медиа-элементы.....	843
Свойства.....	844
События.....	846
Пользовательские медиапроигрыватели.....	847
Обнаружение поддержки кодека.....	848
Тип аудио.....	848
Встроенное перетаскивание.....	849
События перетаскивания.....	849
Пользовательские цели перетаскивания.....	850
Объект dataTransfer.....	851
dropEffect и effectAllowed.....	852
Возможность перетаскивания.....	854
Дополнительные члены.....	854
Notifications API.....	854
Разрешения для уведомлений.....	855
Отображение и скрытие уведомлений.....	855
Обратные вызовы жизненного цикла уведомлений.....	856
Page Visibility API.....	856
Streams API.....	857
Введение в потоки.....	858
Читаемые потоки.....	859
Записываемые потоки.....	861
Потоки преобразования.....	863
Соединение потоков.....	864
API производительности.....	866
High Resolution Time API.....	867
Performance Timeline API.....	868
Веб-компоненты.....	872
Шаблоны HTML.....	872
Теневая DOM.....	876
Пользовательские элементы.....	884
Web Cryptography API.....	890
Генерация случайных чисел.....	890
Использование объекта SubtleCrypto.....	892
Итоги.....	904

Глава 21. Обработка ошибок и отладка.....906

Уведомления об ошибках.....	907
Консоли браузеров для ПК.....	907
Консоли мобильных браузеров.....	907
Обработка ошибок.....	908
Инструкция try-catch.....	908
Генерирование ошибок.....	912
Событие error.....	915
Стратегии обработки ошибок.....	916
Идентификация потенциальных источников ошибок.....	917
Различение критичных и некритичных ошибок.....	922
Протоколирование ошибок на сервере.....	923
Приемы отладки.....	924
Вывод сообщений на консоль.....	925
Выполнение в консоли.....	926
Использование средства отладки JavaScript.....	926
Вывод сообщений на страницу.....	927
Заменяющие методы консоли.....	927
Генерирование ошибок.....	928
Частые устаревшие ошибки Internet Explorer.....	929
Недопустимый символ.....	929
Член группы не найден.....	929
Неизвестная ошибка выполнения.....	930
Синтаксическая ошибка.....	930
Не удастся найти указанный ресурс.....	931
Итоги.....	932

Глава 22. XML в JavaScript.....933

Поддержка XML DOM в браузерах.....	933
DOM Level 2 Core.....	934
Тип DOMParser.....	934
Тип XMLSerializer.....	936
Поддержка XPath в браузерах.....	936
DOM Level 3 XPath.....	936
Результат из одного узла.....	939
Результаты простых типов.....	939
Тип результата по умолчанию.....	940
Поддержка пространств имен.....	940
Поддержка XSLT в браузерах.....	942
Тип XSLTProcessor.....	942

Использование параметров	943
Сброс процессора	944
Итоги	944
Глава 23. JSON	945
Синтаксис	946
Простые значения	946
Объекты	946
Массивы.....	947
Синтаксический анализ и сериализация.....	949
Объект JSON.....	949
Параметры сериализации.....	950
Параметры синтаксического анализа.....	954
Итоги	955
Глава 24. Сетевые запросы и удаленные ресурсы.....	956
Объект XMLHttpRequest.....	957
Использование объекта XHR	957
Заголовки HTTP	960
Запросы GET	961
Запросы POST	962
XMLHttpRequest Level 2.....	963
Тип FormData	964
Тайм-ауты.....	964
Метод overrideMimeType()	965
События хода обмена данными.....	966
Событие load	966
Событие progress.....	967
Обмен ресурсами с запросом происхождения	968
Предварительные запросы.....	969
Запросы с учетными данными.....	970
Альтернативные методики кроссдоменного взаимодействия.....	970
Проверка связи с помощью изображения	971
JSONP.....	971
FETCH API.....	973
Основы использования API	973
Общие паттерны Fetch	981
Объекты Headers	983
Объект Request.....	986
Объект Response	989
Запросы, ответы и описание тела	995

BEACON API.....	1003
Веб-сокеты.....	1005
API.....	1005
Отправка и получение данных	1006
Другие события.....	1006
Безопасность.....	1007
Итоги	1008

Глава 25. Клиентское хранилище1010

Cookie-файлы	1010
Ограничения	1011
Части cookie-файла.....	1012
Cookie-файлы в JavaScript	1013
Вложенные cookie-файлы	1016
Замечания по поводу cookie-файлов.....	1021
Веб-хранилище	1021
Тип Storage.....	1022
Объект sessionStorage	1022
Объект localStorage.....	1024
Событие storage.....	1024
Пределы и ограничения	1025
IndexedDB	1025
Базы данных.....	1026
Хранилища объектов.....	1026
Транзакции	1027
Вставка данных	1028
Запросы с курсорами	1029
Диапазоны ключей.....	1032
Указание направления перемещения курсора	1033
Индексы.....	1034
Проблемы параллельного доступа.....	1036
Пределы и ограничения	1037
Итоги	1037

Глава 26. Модули1038

Паттерн Модуль	1039
Идентификаторы модулей.....	1039
Зависимости модуля	1039
Загрузка модулей	1040
Точки входа.....	1040
Асинхронные зависимости	1041

Программные зависимости.....	1042
Статический анализ.....	1042
Циклические зависимости.....	1043
Импровизированные модульные системы.....	1044
Загрузчики модулей до ES6.....	1047
CommonJS.....	1047
Асинхронное определение модулей.....	1051
Универсальное определение модулей.....	1052
Устаревший модуль загрузчика.....	1052
Модули в ES6.....	1053
Маркировка и определение модулей.....	1053
Загрузка модулей.....	1054
Модульное поведение.....	1055
Экспортирование модулей.....	1055
Импорт модулей.....	1058
Сквозной экспорт модулей.....	1061
Модули рабочих потоков.....	1062
Обратная совместимость.....	1062
Итоги.....	1063
Глава 27. Рабочие потоки	1064
Введение в рабочие потоки.....	1065
Сравнение рабочих потоков и потоков выполнения.....	1065
Типы рабочих потоков.....	1066
WorkerGlobalScope.....	1067
Выделенные рабочие потоки.....	1068
Основные сведения о выделенных рабочих потоках.....	1068
Выделенные рабочие потоки и явные MessagePort.....	1072
Жизненный цикл выделенного рабочего потока.....	1072
Настройка параметров рабочих потоков.....	1074
Создание рабочего потока из встроенного JavaScript.....	1075
Динамическое выполнение сценария внутри рабочего потока.....	1076
Передача задач вложенным рабочим потокам.....	1078
Обработка ошибок рабочих потоков.....	1078
Общение с выделенным рабочим потоком.....	1079
Передача данных рабочего потока.....	1083
Пулы рабочих потоков.....	1089
Общие рабочие потоки.....	1092
Основы использования общих рабочих потоков.....	1093
Жизненный цикл общих рабочих потоков.....	1096
Подключение к общему рабочему потоку.....	1098

Служебные рабочие потоки	1099
Основы использования служебных рабочих потоков	1100
Кеш служебного рабочего потока	1109
Клиенты служебных потоков	1115
Служебные рабочие потоки и согласованность	1115
Жизненный цикл служебных рабочих потоков	1117
Инверсия контроля и постоянство служебных потоков	1123
Управление кешированием файлов служебных потоков с помощью updateViaCache	1123
Принудительный запуск операций в служебном потоке	1124
Обмен сообщениями в служебном потоке	1125
Перехват события извлечения	1127
Всплывающие уведомления	1129
Итоги	1132

Глава 28. Лучшие практики1133

Удобство сопровождения кода	1133
Какой код удобно сопровождать?	1134
Соглашения по формату кода	1134
Слабая связанность	1138
Принципы программирования	1142
Быстродействие	1147
Область видимости	1147
Выбор оптимального подхода	1149
Сокращение количества инструкций	1155
Оптимизация взаимодействия с DOM	1157
Развертывание	1160
Процесс сборки	1161
Проверка кода	1163
Сжатие	1164
Итоги	1166

Приложения www.piter.com

*Джордан за ее непоколебимую поддержку,
несмотря на все мои «уже почти готово».*

Об авторе

МЭТТ ФРИСБИ (MATT FRISBIE) занимается разработкой веб-приложений более десяти лет. За это время он побывал соучредителем стартапа, инженером в технологической компании «Большой четверки» и первым инженером стартапа Y Combinator, который в итоге превратился в бизнес на миллиард долларов. Как инженер-программист Google Мэтт работал на платформах AdSense и Accelerated Mobile Pages (AMP); его код работает на большинстве веб-браузеров планеты. До этого он был первым инженером в DoorDash, где заложил основу для планирования обновлений их драйверов, управления меню и инфраструктуры распределения заказов. Мэтт написал две книги и выпустил две серии видео для O'Reilly и Packt. Он выступает на встречах и вебкастах фронтендщиков, и к тому же является перво-классным сомелье. В Twitter Мэтта можно найти под ником [@mattfriz](#).

О научных редакторах

ХАИМ КРАУЗЕ (CHAIM KRAUSE) — любитель компьютеров, электроники, животных и электронной музыки. Больше всего он радуется, когда удастся объединить два или более этих интересов в одном проекте. Почти всему Хаим обучился самостоятельно. Он в шутку говорит всем, что единственная разница между его домашними занятиями и работой — это используемый вход в систему. Как вечного студента, его часто раздражают технические ошибки в документации, которые отнимают драгоценное время и вызывают пустую досаду. Одна из причин, по которой он работает научным редактором технических книг — помочь другим избежать тех же ошибок.

МАРСИЯ УИЛБУР (MARCIA WILBUR) — технический писатель. Она дает консультации в области полупроводников и специализируется на индустриальном интернете вещей (IIoT) и ИИ. Марсия имеет ученые степени в области информатики, технических коммуникаций и информационных технологий. В качестве президента Copper Linux User Group, она активно участвует в сообществе разработчиков: возглавляет West Side Linux + Pi и East Valley и ведет регулярные проекты Raspberry Pi, Beaglebone, Banana Pi/Pro и ESP8266. Эти проекты включают в себя домашнюю автоматизацию, игровые приставки, видеонаблюдение, сеть, мультимедиа и другие «Pi-развлечения».

Помимо работы, она является волонтером в различных организациях, использующих Pi-модули и Linux, чтобы обеспечить доступ к образовательному контенту для школ в сельских, недостаточно обслуживаемых и пострадавших от стихийных бедствий районах. Ради интереса она служит сообществу в роли ведущего разработчика Debian для Linux Respin, инструмента для резервного копирования и настройки дистрибутива.

Благодарности

Спасибо Wiley, что позволили мне заняться этим проектом. Написание четвертого издания «JavaScript для профессиональных веб-разработчиков» было одним из наиболее сложных, но в то же время интересных проектов, над которыми я когда-либо работал. Эта книга не вышла бы в свет без терпения и поддержки от Wiley. Спасибо сотрудникам Wiley, в частности Джиму Минателю (Jim Minatel), который передал этот проект в мои руки и довел его до конца.

Хотел бы поблагодарить Николаса Закаса (Nicholas Zakas), автора первых трех изданий, за всю работу, которую он проделал до моего участия. Книга не получилась бы настолько удачной без железного фундамента, заложенного им. Я желаю ему скорейшего выздоровления.

Отдельное спасибо Адаоби Оби Тултон (Adaobi Obi Tulton) за ее наставничество. Ее участие было бесценным на протяжении всего процесса, и я не справился бы без ее терпения и опыта.

Я также хотел бы поблагодарить всех, кто вычитывал рукопись: Сэмюэля Каллнера (Samuel Kallner), Хаима Краузе (Chaim Krause), Марсию Уилбур (Marcia Wilbur), Нэнси Рапопорт (Nancy Rapoport), Атияппан Лалит Кумар (Athiyappan Lalith Kumar) и Эвелин Веллборн (Evelyn Wellborn). Книга вроде этой просто развалилась бы без вашего участия.

Наконец, хотел бы поблагодарить Зака Тратара (Zach Tratar) за предисловие. Мне посчастливилось встретиться с Заком в тот же день, когда я переехал в Сан-Франциско. За прошедшие годы он проявил себя как безумно эрудированный и чрезвычайно приятный человек, не говоря уже о том, насколько он хороший инженер программного обеспечения. Я считаю большой честью то, что он согласился внести свой вклад в эту книгу.

Предисловие

Промышленная революция родилась из производства стали, а интернет-революция — из JavaScript. Выкованное и закаленное за счет постоянных итераций в течение последних 25 лет, доминирование JavaScript в разработке приложений в настоящее время трудно поставить под сомнение, но так было не всегда.

Брендану Эйху (Brendan Eich) потребовалось всего десять дней, чтобы создать первую версию JavaScript. Она казалась хрупкой, но, как показывает история, первое впечатление обманчиво. Сегодня каждый аспект JavaScript — каждая деталь, о которой вы узнаете в этой книге, — является результатом долгих дискуссий. Не каждое решение идеально — в конце концов идеального языка программирования не существует. Но если судить только по одному его повсеместному распространению, JavaScript может приблизиться к идеалу. Это единственный язык, который можно применить везде: серверы, настольные браузеры, мобильные веб-браузеры и даже собственные мобильные приложения.

JavaScript теперь используется разработчиками всех направлений и уровней опыта: от тех, кто заботится о хорошо спроектированном, элегантном ПО, до тех, кому просто нужно по-быстрому собрать рабочий вариант для достижения бизнес-целей.

Как вы будете его использовать, зависит только от вас. Эта сила ваша.

За последние 15 лет разработки ПО инструменты JavaScript и его лучшие методики сильно изменились. Мой опыт работы с языком начался в 2004 г., когда доминировали Geocities, Yahoo Groups и Macromedia Flash player. JavaScript был похож на игрушку, и я играл в некоторых из популярных в то время песочниц: RSS и MySpace Profile Pages. Помощь другим в изменении и настройке личных сайтов показалась мне настоящим Диким Западом и в итоге зацепила меня.

Когда я запускал свою первую компанию, настройка хоста для базы данных занимала несколько дней, и JavaScript был встроен в HTML. Не было никаких «приложений» внешнего интерфейса — все это в основном представляло из себя бессистемные функции. По мере того как Ajax, возглавляемый jQuery, становился все более популярным, развивался новый мир с более надежными приложениями. Это движение набрало головокружительную скорость, а затем неожиданно были выпущены мощные фреймворки. Модели внешнего интерфейса! Привязка данных! Управление маршрутами! Реактивное отображение! Именно во время этой фронтенд-революции я переехал в Кремниевую долину, чтобы помочь запустить

компанию, основанную Леди Гагой, и вскоре миллионы пользователей начали использовать мой код. Находясь в Кремниевой долине достаточно долго, я внес свой вклад в проекты с открытым исходным кодом, обучил больше разработчиков, чем могу сосчитать, и поймал удачу за хвост. Моя последняя компания была приобретена Stripe в 2018 г., где я сейчас работаю над созданием экономической инфраструктуры для интернета.

Я имел удовольствие встретиться с Мэттом в тот день, когда он впервые вылетел в Пало-Альто, чтобы возглавить инжиниринг при небольшом стартапе Сласо: в то время я только присоединился к нему в качестве советника. Энергия и страсть Мэтта к отличному ПО были очевидны, и молодая компания быстро выпустила прекрасный продукт. Как было принято в Кремниевой долине со времен HP, этот стартап зародился у кого-то дома. Но это не был обычный дом. Это был «хакерский дом», где постоянно жили десять или около того блестящих разработчиков ПО. Хотя это была не жизнь высшего класса — двухъярусные кровати и стулья, выброшенные кем-то и затем подобранные на улице, были обычным явлением, — количество и качество ежедневно пирующего там кода поражало воображение. В нерабочее время большинство разработчиков просто переключали внимание и занимались своими сторонними проектами еще пару часов. Новички, не умевшие писать код, часто вдохновлялись: у них возникало желание учиться и за несколько недель они осваивали программирование.

Мэтт был движущей силой этого сосредоточия продуктивности. Он был самым опытным инженером-программистом в доме, а также оказался самым неиспорченным и профессиональным. Диплом в сфере компьютерной инженерии тогда не был чем-то привычным, поэтому при виде алгоритмов, расчетов производительности и кода, написанных на окнах или на доске, можно было догадаться, что в этот момент Мэтт работает над следующим большим проектом. Со временем мы стали близкими друзьями. Его ум, любовь к наставничеству и способность переводить большинство вещей в шутку — качества, которыми я восхищался.

Мэтт невероятно талантливый инженер-программист и лидер; именно его уникальный набор опыта и знаний делает его одним из самых квалифицированных людей в мире для написания этой книги.

Он не просто проводил время, обучая других, — он сам работал.

В Сласо он разработал множество полноценных продуктов, чтобы помочь учителям улучшить учебный процесс в своих классах. В DoorDash, будучи первым инженером в компании, он создал надежную сеть логистики и доставки, которая достигла небывалого роста и теперь стоит свыше двенадцати миллиардов долларов. Наконец, программное обеспечение Мэтта, применяемое в Google, используют миллиарды людей по всему миру.

Огромное владение, огромный рост и огромный масштаб. Большинству разработчиков за всю карьеру удастся достичь только чего-то одного, если им повезет. Мэтт не только добился всего этого, но и стал автором бестселлера, в «свободное

время» написав две другие книги по JavaScript и Angular¹. Честно говоря, надеюсь, что в его следующей книге будут даны чертежи машины времени, которые он явно скрывает от нас.

Эта книга — надежный инструмент, наполненный знаниями по JavaScript и реальными перспективами. Я рад, что вы продолжаете учиться и воплощать свои идеи. Делите книгу на части, делайте заметки и не забывайте открывать редактор кода — в конце концов, интернет-революция еще только начинается!

*Зак Тратар,
инженер-программист в Stripe,
бывший соучредитель и генеральный директор Jobstart*

¹ Мэтт Фрисби. Angular. Сборник рецептов. Вильямс, 2018 г.

Введение

Техлид в Google однажды поделился со мной убедительным взглядом на JavaScript: это *не совсем* связный язык программирования — по крайней мере, в формальном смысле. Спецификация ECMA-262 *определяет* JavaScript, но единственной истинной *реализации* его не существует. Более того, язык далеко не герметичен. Он плавает в настоящем океане смежных спецификаций, которые управляют API всего, что касается JavaScript: DOM, сетевых запросов, системного оборудования, хранилища, событий, файлов, криптографии и *comen* других. Веб-браузеры и их различные JS-движки реализуют эти спецификации по своему усмотрению. Chrome использует Blink/V8, Firefox — Gecko/SpiderMoney, а Safari — WebKit/JavaScriptCore. Браузеры будут запускать *почти* весь JavaScript таким образом, который соответствует спецификациям, но в интернете полно примеров идиосинкразий каждого браузера. Поэтому JavaScript более точно характеризуется как совокупность реализаций браузера.

Хотя веб-пуристы могут настаивать на том, что JavaScript не должен быть неотъемлемым компонентом веб-страниц, они должны признать, что современный веб сильно бы без него сократился. Нелишне говорить, что JavaScript практически неизбежен: в телефоны, компьютеры, планшеты, телевизоры, игровые приставки, умные часы, холодильники и даже в автомобили теперь встроены веб-браузеры, поддерживающие JavaScript. Почти три миллиарда человек сейчас используют смартфон с веб-браузером. Живое сообщество языка создает целый поток высококачественных проектов с открытым исходным кодом. Браузеры теперь имеют первоклассную поддержку API, эмулирующих нативные мобильные приложения. В опросе разработчиков Stack Overflow за 2019 г. JavaScript был признан самым популярным языком программирования седьмой год подряд.

Возрождение JavaScript близко.

Эта книга полностью описывает эволюцию JavaScript, начиная с его реализаций в ранних браузерах Netscape и заканчивая современными возможностями, включающими поддержку головокругительного спектра браузерных технологий. Книга охватывает большое количество продвинутых тем в мельчайших деталях, но при этом гарантирует, что читатель будет понимать, как использовать эти темы и где они уместны. Используя полученные значения, вы сможете решать бизнес-задачи, с которыми каждый день сталкиваются многие веб-разработчики по всему миру.

ЦЕЛЕВАЯ АУДИТОРИЯ

Эта книга ориентирована на три группы читателей:

- опытных разработчиков, разбирающихся в объектно-ориентированном программировании и желающих изучить JavaScript в контексте традиционных объектно-ориентированных языков, таких как Java и C++;
- разработчиков веб-приложений, которым нужно сделать свои веб-сайты и веб-приложения более удобными в использовании;
- начинающих разработчиков на JavaScript, желающих лучше понять этот язык.

Кроме того, книга может заинтересовать вас, если вы используете что-то из этого:

- Java;
- PHP;
- Python;
- Ruby;
- Golang;
- HTML;
- CSS.

Эта книга не подойдет вам, если вы не владеете базовыми навыками программирования или всего лишь хотите добавить на веб-сайт простые средства взаимодействия с пользователями. Если вы узнали себя в этом описании, вам лучше обратиться к пятому изданию книги *Beginning JavaScript* (Wiley, 2015).

ТЕМЫ, РАССМАТРИВАЕМЫЕ В КНИГЕ

Книга, которую вы держите в руках, объединяет введение в JavaScript для разработчиков и описание более сложных и полезных возможностей этого языка.

Сначала мы рассмотрим историю и эволюцию JavaScript, после чего подробно обсудим компоненты языка, уделив особое внимание стандартам, таким как ECMAScript и Document Object Model (DOM).

Взяв эту информацию за основу, мы рассмотрим базовые концепции JavaScript, в том числе классы, промисы, итераторы и прокси. Затем перейдем к углубленному изучению обнаружения клиентов, событий, анимации, форм, ошибок и JSON.

Последняя часть книги посвящена новейшим и наиболее важным спецификациям, появившимся за последние несколько лет. Они включают в себя Fetch, модули, веб-работники, рабочие потоки служб и набор новых API.

СТРУКТУРА

В книге 28 глав:

1. **Что такое JavaScript?** Эта глава содержит общие сведения о JavaScript: вы узнаете, как появился этот язык, как он развивался и что представляет собой сегодня. Мы обсудим, как JavaScript соотносится с ECMAScript, объектной моделью документа (DOM) и объектной моделью браузера (BOM). Кроме того, вы ознакомитесь с соответствующими стандартами от Европейской ассоциации производителей вычислительной техники (ECMA) и консорциума World Wide Web (W3C).
2. **JavaScript в HTML.** Описано применение JavaScript в сочетании с HTML для создания динамических веб-страниц. Также рассмотрены различные способы внедрения JS-кода в страницу, типы JavaScript-контента и их использование с элементом `<script>`.
3. **Основы языка.** Рассмотрены базовые концепции языка, в том числе его синтаксис и управляющие инструкции. Указаны сходства и различия JavaScript и других C-подобных языков, а также описано приведение типов в связи со встроенными операторами. Охватывает все языковые примитивы, включая тип `Symbol`.
4. **Переменные, область видимости и память.** Здесь рассказано о переменных, которые в JavaScript являются слабо типизированными. Глава содержит сведения о различиях между примитивными и ссылочными значениями и контексте выполнения в связи с переменными. Вы также узнаете о том, как работает сборщик мусора и как память возвращается среде, когда переменные покидают область видимости.
5. **Ссылочные типы.** Эта глава посвящена встроенным в JavaScript ссылочным типам, таким как `Date`, `RegExp`, примитивы и оболочки примитивов. Для каждого ссылочного типа, определенного в ECMA-262, приведены как теоретические сведения, так и подробности его реализации в браузерах.
6. **Ссылочные типы коллекций.** Продолжает рассмотрение встроенных ссылочных типов с `Object`, `Array`, `Map`, `WeakMap`, `Set` и `WeakSet`.
7. **Итераторы и генераторы.** Представляет две новые фундаментальные концепции из недавней версии ECMAScript: итераторы и генераторы. Каждая из них обсуждается как относительно ее фундаментального поведения, так и относительно ее использования в существующих языковых конструкциях.
8. **Объекты, классы и объектно-ориентированное программирование.** В этой главе рассмотрены приемы использования классов и объектно-ориентированного программирования на JavaScript. Она начинается с углубленного изучения типа `Object` JavaScript и продолжается рассмотрением прототипного наследования. Далее следует полное объяснение классов ES6 и того, почему они являются близкими родственниками прототипного наследования.

9. **Прокси и Reflect.** Данная глава представляет две тесно связанные между собой концепции: Proxy и Reflect API. Они могут использоваться для перехвата и добавления дополнительного поведения в основные операции в языке.
10. **Функции.** Функции-выражения относятся к наиболее мощным аспектам применения JavaScript. В этой главе описываются замыкания, подробности функционирования объекта this, паттерн Модуль, создание закрытых членов объектов, стрелочные функции, параметры по умолчанию и операторы расширения.
11. **Промисы и асинхронное программирование.** В этой главе рассмотрены две новые тесно связанные конструкции асинхронного программирования: тип Promise и async/await. Глава начинается с обсуждения парадигмы асинхронного JavaScript и продолжается обзором использования промисов и их отношений к асинхронным функциям.
12. **Объектная модель браузера.** В этой главе описана объектная модель браузера (BOM), которая предоставляет объекты для взаимодействия с браузером. Вы ознакомитесь со всеми BOM-объектами, включая window, document, location, navigator и screen.
13. **Распознавание клиента.** В этой главе рассмотрены способы распознавания клиентского браузера и поддерживаемых им функциональных возможностей. Вы узнаете о распознавании возможностей, анализе строки пользовательского агента, о достоинствах и недостатках каждого подхода и о том, какой подход оптимален в той или иной ситуации.
14. **Объектная модель документа.** В этой главе описаны объекты, определенные в спецификации DOM Level 1. После ознакомления с XML в контексте DOM вы сможете подробно изучить модель DOM и предоставляемые ею возможности по манипулированию содержимым страницы.
15. **Расширения DOM.** Глава содержит сведения о том, как API и сами браузеры расширяют функционал DOM. В число рассматриваемых тем входят Selectors, Element Traversal API и расширения HTML5.
16. **DOM Level 2 и 3.** В этой главе, основанной на двух предыдущих главах, рассказано о том, как спецификации DOM Level 2 и 3 расширяют DOM дополнительными свойствами, методами и объектами. Включает охват дополнений DOM4, таких как наблюдатели за изменениями.
17. **События.** Из этой главы вы узнаете о природе JavaScript-событий, их генерировании и о том, как события переопределены в DOM.
18. **Анимация и рисование на холсте.** Эта глава посвящена тегу <canvas> и его использованию для динамичного создания графики. Рассмотрены двумерный контекст и контекст WebGL (трехмерный), что поможет приступить к созданию анимаций и игр. Включает обзор WebGL1 и WebGL2.
19. **Работа с формами.** В этой главе рассказывается, как с помощью JavaScript улучшить взаимодействие с формами и обойти ограничения браузера. Особое

внимание уделено работе с элементами форм, такими как текстовые поля и списки, а также проверке и обработке данных.

20. **API в JavaScript.** Рассматривает широкий ассортимент JavaScript API, включая Atomics, Encoding, File, Blob, Notifications, Streams, Timing, Web Components и Web Cryptography.
21. **Обработка ошибок и отладка.** В этой главе рассмотрены способы обработки ошибок в JS-коде. Также описаны инструменты и приемы отладки для каждого браузера и приведены рекомендации по упрощению процесса отладки.
22. **XML в JavaScript.** В этой главе рассмотрены возможности JavaScript, используемые для чтения XML-данных и манипулирования ими. Описаны различия возможностей и объектов в разных веб-браузерах и приведены советы по написанию кроссбраузерного кода. Также в главе приведены сведения об использовании XSLT-преобразований для трансформации XML-данных на клиентских системах.
23. **JSON.** В этой главе представлен формат JSON — альтернатива XML. Описаны возможности синтаксического анализа и сериализации JSON и приведены сведения о том, как обеспечить безопасность при использовании JSON.
24. **Сетевые запросы и удаленные ресурсы.** Данная глава исследует все наиболее распространенные способы запроса данных и ресурсов браузером. Включает в себя рассмотрение унаследованного объекта XMLHttpRequest, а также современного Fetch API.
25. **Клиентское хранилище.** В этой главе рассказано о том, как определить, что приложение работает в автономном режиме, и описаны различные методики сохранения данных на клиентском компьютере. В главе рассмотрены как традиционные файлы cookie, так и более новые возможности, такие как веб-хранилище и база данных IndexedDB.
26. **Модули.** Здесь обсуждается шаблон Модуль и его влияние на кодовые базы. Затем рассматриваются загрузчики модулей до ES6, такие как CommonJS, AMD и UMD, заканчивается глава подробным описанием нового шаблона ES6 — Модуль — и его правильного использования.
27. **Рабочие потоки.** Эта глава в целом охватывает выделенные рабочие потоки, общие рабочие потоки и рабочие потоки служб. Включает обсуждение того, как рабочие потоки ведут себя на уровне операционной системы и на уровне браузера, а также стратегии оптимального использования различных типов потоков.
28. **Лучшие практики.** Эта глава посвящена использованию JavaScript в корпоративной среде. В ней описаны приемы обслуживания кода, в том числе методики написания и форматирования кода и общие приемы программирования. Также приведены советы по оптимизации и повышению быстродействия кода. Наконец, рассмотрены вопросы развертывания приложений, включая реализацию процесса сборки.

ЧТО НУЖНО ДЛЯ ЭФФЕКТИВНОЙ РАБОТЫ С КНИГОЙ

Для выполнения примеров из книги вам потребуется следующее:

- любая современная операционная система вроде Windows, Linux, MacOS, Android или iOS;
- любой современный браузер вроде IE11+, Edge 12+, Firefox 26+, Chrome 39+, Safari 10+, Opera 26+, or iOS Safari 10+.

Полный исходный код можно загрузить с сайта <https://www.wiley.com/en-us/Professional+JavaScript+for+Web+Developers%2C+4th+Edition-p-9781119366447>.

ОТ ИЗДАТЕЛЬСТВА

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете приложения из «JavaScript для профессиональных веб-разработчиков», а также подробную информацию о наших книгах.