Моей удивительной жене,
которая пишет гораздо более захватывающие книги,
по сравнению с книгами о веб-разработке,
посвященные драконам и катастрофам

Оглавление

Содержание

Предисловие

Интернет прошел долгий путь. То, что началось три десятилетия назад как относительно простое средство публикации, совместного использования, обнаружения и потребления контента, превратилось в мощную и гибкую платформу приложений, поддерживающую невероятное количество вариантов использования. Между тем сфера его присутствия расширилась, и теперь выход в интернет осуществляется не только с настольных компьютеров, но и с устройств всех типов.

В результате этого постепенного преобразования мы, веб-разработчики, преследуем постоянно меняющуюся цель. Сегодняшние веб-сайты на несколько порядков сложнее по сравнению с их ранними предшественниками, и ожидания, связанные с пользовательским интерфейсом, значительно выросли.

К счастью, наш инструментарий также не стоял на месте. Сама веб-платформа приобрела сотни новых возможностей, а последующие поколения библиотек, фреймворков и инструментов постоянно совершенствуют современный уровень развития технологий, помогая нам удовлетворять растущие требования.

Одним из основных факторов, способствующих трансформации интернета в последние годы, стало широкое внедрение разработки пользовательского интерфейса на основе компонентов. Разделение нашей работы на компоненты, каждый из которых отвечает за структуру, стиль и поведение части пользовательского интерфейса, помогло нам управлять сложностью и создавать более сложные сайты.

Компоненты можно повторно использовать в каком-либо проекте или совместно в разных проектах, что повышает нашу эффективность. Дизайн-системы можно выразить в виде наборов готовых к использованию компонентов, обеспечивающих согласованность, которые позволяют командам сосредоточиться на конкретных потребностях продукта.

Популярные фреймворки помогли осуществить революцию компонентов, и сегодня большинство компонентов специфичны для конкретного фреймворка или библиотеки. Но параллельно предпринимались многолетние усилия по созданию мощной, нативной модели компонентов для веб-платформы.

Веб-компоненты – это общий термин для нового семейства функций веб-платформ, предлагающих прямую поддержку разработки на основе компонентов. Пользовательские элементы позволяют расширять словарь HTML, определяя собственные теги, которые легко работают со встроенными в браузер тегами и могут использоваться в одних и тех же местах, независимо от фреймворка. Технология Shadow DOM позволяет вам применять инкапсуляцию в нативном стиле, гарантируя, что CSS-правила компонента не будут непреднамеренно нарушать – и не будут нарушаться – форматирование страницы.

Вам, наверное, интересно, какие преимущества дают веб-компоненты по сравнению с компонентными моделями. С одной стороны, веб-компоненты обещают повысить совместимость, упрощая обмен компонентами даже между комплектами технологий. Модель общих компонентов также снижает риск блокировки, позволяя вам выполнять больше работы по мере изменения набора инструментов с течением времени.

Книга, которую вы сейчас держите в руках, исключительно своевременна. Путь к стандартизации и поддержка веб-компонентов претерпевали взлеты и падения, но я рад сообщить, что цель уже видна: все, кроме одного из популярных браузеров, уже поддерживают веб-компоненты, а когда состоится официальный релиз следующей версии Microsoft Edge, головоломка будет завершена.

Пользовательские элементы, Shadow DOM и другие функции веб-компонентов по своей природе являются низкоуровневыми примитивами. Некоторые разработчики будут использовать эти функции только косвенно, поскольку поддержка веб-компонентов во фреймворках увеличилась с ростом поддержки браузеров. Многие из самых популярных фреймворков теперь облегчают разработку и совместное использование веб-компонентов, и стал появляться целый новый класс инструментов, ориентированных на веб-компоненты.

Но вы также можете использовать функции веб-компонентов напрямую, по отдельности или сочетая их. Читая эту книгу, вы подробно изучите каждую функцию и то, как они связаны друг с другом, что даст вам возможность сделать правильный выбор для себя и своей команды.

Бен Фаррелл использовал веб-компоненты с момента их возникновения в самых разных приложениях. В ходе своей работы он накопил огромное количество ценных знаний и обнаружил множество эффективных шаблонов, которыми он поделится с вами на этих страницах.

Бен приводит примеры, демонстрируя разные концепции с помощью убедительных проектов, которые освещают реалистичные варианты использования. Вы, конечно, многому научитесь, но также непременно найдете здесь идеи и код, которые можно применить непосредственно в своих собственных проектах.

Решив заняться веб-компонентами и взяв эту книгу, вы сделали хороший выбор. Наслаждайтесь этим путешествием!

Грей Нортон,
технический руководитель /
менеджер проекта Polymer, GOOGLE

От автора

Для меня знакомство с веб-компонентами началось в 2013 году. Я помню, что работал над забавным небольшим проектом с использованием Angular версии 1 и изучал некоторые аспекты управления CSS и классами, которые Angular в то время плохо обрабатывал. Я знал, что мог бы легко сделать то, что мне нужно, в простом HTML, CSS и JavaScript, но Angular затруднял это только потому, что то, что я делал, находилось за пределами проторенных троп.

Примерно в это же время я почувствовал, что действительно начинаю овладевать Angular, поэтому написал в блоге несколько постов о некоторых интересных, нетипичных подходах. Тогда же волнение по поводу Angular стало угасать, и только начиналось волнение по поводу React.

Честно говоря, я был разочарован. Я долго смотрел на цикл, в котором чувствовал себя пойманным в ловушку. В течение всего двух или трех лет я постоянно учился и получал хорошие знания по фреймворкам JS. Ни один из этих фреймворков не был совместим друг с другом. Я дошел до того, что почувствовал, что действительно могу сосредоточиться на своем проекте без фреймворка на заднем плане, но затем неожиданно появилось нечто новое, что заставило меня почувствовать, что мне нужно вернуться на круги своя.

В то же самое время Google была выпущена библиотека Polymer как очень ранняя и нестабильная версия. Создание отдельных компонентов, которые могли бы существовать где угодно, звучало как удивительное обещание. Первоначально мне нравилось то, чего она пыталась достичь, но API, предшествующий первой версии, который постоянно менялся, и тот факт, что я заменял свой рабочий процесс еще одним фреймворком, заставил меня все переосмыслить. Я начал изучать предлагаемые веб-стандарты, которые сделали возможным создание библиотеки Polymer, и увидел огромный потенциал. Я понял, что это была не библиотека Polymer, которой я восхищался, – в действительности это были веб-компоненты.

Я начал вести блог и дискуссии о веб-компонентах. Примерно в это же время присоединился к Adobe. Это было важно, потому что моя команда работала над небольшими прототипами с одним, может быть, двумя разработчиками проекта. Это означало, что я мог экспериментировать с технологией и инструментами по своему выбору. Почти для каждого проекта я продолжал продвигать веб-компоненты, экспериментируя и постоянно улучшая рабочий процесс для работы с ними.

Конечно, это было непросто. Иногда я полностью был лишен почвы под ногами! Поскольку веб-компоненты стали стандартом, которым они являются сегодня, мы увидели, что изменение API и функции стали устаревшими, но у меня не было выбора, потому что мне действительно нравится работать как можно ближе к браузеру, используя только

HTML, JS и CSS, и я рассматривал веб-компоненты как средство обеспечения структуры своих проектов, а не для того, чтобы они превратились в спагетти-код.

Я еще не был полностью убежден в жизнеспособности веб-компонентов. С одной стороны, я пока не использовал Shadow DOM. Я не хотел увлекаться чем-то, что поддерживала только Google, у которой была сомнительная поддержка полифилов. Но затем веб-компоненты появились в браузере Safari, и Mozilla также пообещал, что будет поддерживать их. Вишенкой на торте стал момент, когда браузеры начали поддерживать модули JS и импорт нативно, и я смог правильно разделить код и, что более важно, HTML и CSS. Когда все это произошло, я знал, что веб-компоненты начинают реализовывать свой потенциал.

Конечно, все происходило очень медленно в течение нескольких лет. Многие разработчики, которые изначально были в восторге от веб-компонентов, потеряли терпение, и я не виню их. Сначала я обратился к издательству Manning по поводу книги о веб-компонентах – до того, как произошли некоторые важные ключевые события, например когда крупные компании-разработчики популярных браузеров объединились, чтобы завершить версию спецификации номер 1. В то время Manning не было уверено, особенно из-за того, что книги в этой области не публиковались, поскольку было неизвестно, чем все это закончится.

Был ли я настроен слишком оптимистично или просто провел с ними достаточно времени, чтобы узнать потенциал веб-компонентов, но издательство связалось со мной через год, чтобы сделать еще одно предложение. Даже тогда, в начале 2018 года, дело все равно могло бы принять дурной оборот, если бы другие компании-разработчики браузеров решили пойти на попятную. Кроме того, в то время я не подходил к разработке веб-компонентов так, как это делало большинство разработчиков, используя импорт HTML в качестве отправной точки. Тем не менее на протяжении этой книги класс LitElement от команды Polymer начал действовать в очень похожей со мной манере, используя шаблонные литералы для хранения разметки и стиля. Это, в сочетании с поддержкой веб-компонентов, когда над ними работала и компания Microsoft, осенью 2018 года позволило мне вздохнуть с облегчением, зная, что подходы, описанные в моей книге, идут в ногу с настоящим и будущим веб-компонентов. Я определенно продолжу совершенствовать свой рабочий процесс, по мере того как новые функции появляются в браузере и придумываются сообществом, но я очень рад нынешнему положению веб-компонентов, поскольку моя книга скоро будет опубликована. И конечно же, мне не терпится поделиться всем этим с читателями!

Благодарности

Данная книга была бы невозможна без всех тех удивительных людей, которые помогали мне на протяжении этого пути. Я хочу поблагодарить своих друзей из Северной Каролины и замечательных людей, которые проводят и посещают конференцию NCDevCon, за то, что они почти постоянно слушали мои доклады о веб-компонентах в Yammer. В частности, я хотел бы поблагодарить Эдриана Помилио за то, что он поразил меня своим выступлением в 2011 году, в котором были показаны пользовательские элементы, прежде чем они стали чем-то особенным.

Я также хотел бы поблагодарить членов команды GE Design System за то, что они были моими «сообщниками» в этом деле, в то время когда веб-компоненты были абсолютным новшеством, и мы были уверены, что все остальные считают нас безумцами. В частности, я хотел бы поблагодарить Мартина Рэгга, Джеффа Райхенберга и Джона Роджерсона за то, что они копались со мной в деталях при написании этой книги о новом способе создания сайтов. Еще хотел бы поблагодарить команду Google Polymer за помощь и руководство в течение этого времени, а также их технического руководителя Грея Нортона за написание предисловия к книге.

В Adobe я хотел бы поблагодарить всю команду Adobe Design (и за ее пределами) за поддержку и искреннюю радость по поводу публикации моей первой книги.

Конечно, моя жена Ребекка Гомес Фаррелл не только поддерживала меня, но и сама оказалась замечательным писателем и редактором. Помимо того что она принесла мне крепкий напиток, когда он мне понадобился, она помогла новому писателю стать лучше, давая стоящие, профессиональные советы.

Я хотел бы поблагодарить редакционную команду издательства Manning, в которую входят редакторы-консультанты по аудитории Кристен Уоттерсон, Кевин Харрелд и Ребекка Райнхарт, а также редактор-консультант по технической аудитории Дуглас Дункан, технический корректор Мэтью Уэлк, редактор по производству Энтони Калькара, редактор Ребекка Деуэль-Гальегос и корректор текста Тиффани Тейлор. Наконец, я хотел бы поблагодарить рецензентов, чьи отзывы и понимание сыграли важную роль в формировании этой книги, в том числе Альберто Чарланти, Алисию Бейкер, Бирну Себарте, Клайва Харбера, Дэниела Купера, Эрнана Гарсия, Джеймса Карелла, Джона Ларсена, Хуана Асенсио, Джастина Каллеха, Оливера Ковача, Пьетро Маффи, Рональда Бормана, Рассела Доуна Кахолес, Райана Барроуз, Серхио Арбео, Стефана Троста, Томаса Оверби Хансена, Тимоти Р. Кейна и Кумара С. Унникришнана (TR Technology & Ops).

Об этой книге

Книга «*Веб-компоненты в действии*» не диктует, какие подходы должны использовать разработчики. Вместо того чтобы рассказывать читателям, что делать, я использую более исследовательский подход, чтобы охватить основы веб-компонентов. Вы должны признать, что хотя эксперты могут сказать вам, что такое хороший рабочий процесс на сегодняшний день, захватывающий момент касательно стандартов состоит в том, что их можно создавать таким образом, которого никто не ожидает.

В этой книге я стремлюсь дать вам отличные идеи и рабочие процессы для начала работы. Я также надеюсь дать вам знания для дальнейшего использования веб-компонентов, способами, которые я еще не рассматривал, и для проектов, с которыми я не сталкивался.

Кому следует прочитать эту книгу

Данная книга предназначена для веб-разработчиков, которые интересуются веб-компонентами и хотят узнать больше о стоящих за ними стандартах и о том, как они объединяются с другими веб-технологиями для создания автономных компонентов или приложений.

Она также подходит для разработчиков, которым нужны идеи о том, как освободиться от сложных фреймворков или библиотек и вернуться к написанию простого HTML, JS и CSS без каких-либо шагов сборки.

Как организована эта книга: дорожная карта

Эта книга состоит из трех частей, охватывающих 15 глав и приложение.

В первой части приводятся первые шаги, описывающие получение простого компонента с нуля:

- в первой главе описывается, что имеется в виду, когда речь идет о веб-компонентах и различных стандартах, которые объединяются для их создания;
- вторая глава рассказывает о создании самого первого веб-компонента, а также знакомит вас с концепциями минимума, необходимыми для создания чего-то полезного;
- третья глава выводит минимальный компонент на следующий уровень, делая его многоразовым;
- в четвертой главе подробно описывается API веб-компонентов и жизненный цикл в сравнении их с другими API и жизненными циклами, с которыми вы, возможно, сталкивались;
- в пятой главе вы познакомитесь с модулями для более подходящего повторного использования кода и организации проекта.

Вторая часть основана на минимальном компоненте и охватывает концепции улучшения рабочего процесса разработчика и организации проекта:

- шестая глава подробно описывает использование модулей для разделения и импорта логики представления, такой как HTML и CSS, для лучшей организации вашего компонента;
- седьмая глава посвящена альтернативному, но не предпочтительному способу организации вашего компонента с помощью HTML-импорта, разбивая его на части, которые также относятся к другим аспектам веб-компонентов;
- восьмая глава знакомит вас с технологией Shadow DOM и рассказывает о ее пользе для защиты и инкапсуляции вашего компонента;
- в девятой главе мы продолжаем изучение Shadow DOM, чтобы охватить его CSS-аспекты;
- в десятой главе исследуются проблемы, которые могут возникнуть у разработчиков веб-компонентов с CSS в Shadow DOM, и способы избежать их или преодолеть.

Третья и последняя часть посвящена совместной работе с несколькими компонентами, чтобы создать нечто большее:

- в одиннадцатой главе рассматриваются освещенные ранее концепции, которые используются для создания нового, более отточенного компонента, основанного на уже созданных дочерних компонентах;
- в двенадцатой главе мы продвигаем этот абсолютно новый компонент, чтобы быть более готовым к промышленной эксплуатации благодаря применению инструментов сборки, которые позволяют использовать его в старых браузерах, не поддерживающих веб-компоненты;
- в тринадцатой главе мы дополняем этот компонент, написав для него тесты, которые выполняются в трех разных контекстах, чтобы изучить различные варианты, доступные для разработчиков веб-компонентов;
- в четырнадцатой главе обсуждается передача сообщений между вашими компонентами и подробно рассматривается распространенный шаблон проектирования;
- в пятнадцатой главе рассказывается о будущем веб-компонентов, а также о возможностях, которые они могут предоставить сегодня, скрывая сложность и делая все, от видеоэффектов в реальном времени до смешанной реальности, более простым в использовании.

Наконец, в приложении рассказывается о новых функциях Java Script (ES6/ES2015) и о том, как они помогают веб-компонентам.

О коде

Исходный код предоставляется для всех примеров в этой книге и доступен для скачивания с веб-сайта издательства Manning по адресу www.manning.com/books/web-components-in-action и из репозитория GitHub,

который можно найти по адресу https://github.com/bengfarrell/webcomponentsinaction. Репозиторий организован в папки для каждой главы, и в них обычно находятся вложенные папки для каждого раздела. Исключения составляют случаи работы над большим примером, охватывающим всю главу.

Код можно выполнить только с помощью браузера, и его не нужно компилировать до тех пор, пока вы не дойдете до следующих глав, посвященных инструментам сборки. Как правило, для запуска соответствующего HTML-файла, который служит примером, понадобится простой HTTP-сервер, но только для того, чтобы решить проблемы, связанные с разными источниками.

В этой книге содержится множество примеров исходного кода, как в виде пронумерованных листингов, так и встроенных в обычный текст. В обоих случаях исходный код форматируется `шрифтом фиксированной ширины`, `подобным этому`, чтобы отделить его от обычного текста. Иногда код также выделяется **`жирным шрифтом`**, чтобы выделить код, который изменился по сравнению с предыдущими шагами в этой главе, например когда к существующей строке кода добавляется новая функция.

Во многих случаях оригинальный исходный код переформатируется; мы добавили разрывы строк и переработали отступы, чтобы обеспечить доступное пространство страницы в книге. В редких случаях даже этого было недостаточно, и списки содержат маркеры продолжения строки (➥). Кроме того, комментарии в исходном коде часто удаляются из листингов, когда описание кода есть в тексте. Аннотации к коду сопровождают многие листинги, выделяя важные понятия.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.