

УДК 004.42  
ББК 32.972  
Ф25

**Фаррелл Б.**

Ф25 Веб-компоненты в действии / пер. с англ. Д. А. Беликова. – М.: ДМК Пресс, 2020. – 462 с.: ил.

**ISBN 978-5-97060-856-2**

Один из основных факторов, способствующих трансформации интернета в последние годы, – широкое внедрение разработки пользовательского интерфейса на основе компонентов. В этой книге подробно описываются рабочие процессы, которые дают вам полный контроль над стилями и поведением компонентов и существенно упрощают их создание, совместное и повторное использование в проектах.

В первой части рассмотрено получение простого компонента с нуля. Вторая часть посвящена улучшению организации проекта. В третьей части освещаются принципы совместной работы с несколькими компонентами, позволяющей решать более сложные задачи. Для всех примеров предоставляется исходный код.

Издание предназначено для веб-разработчиков, имеющих опыт работы с HTML, CSS и JavaScript.

УДК 004.42  
ББК 32.972

Original English language edition published by Manning Publications USA, USA. Copyright © 2019 by Manning Publications Co. Russian-language edition copyright © 2020 by DMK Press. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-1-617-29577-5 (англ.)  
ISBN 978-5-97060-856-2 (рус.)

Copyright © 2019 by Manning Publications Co.  
© Оформление, издание, перевод, ДМК Пресс, 2020

# Оглавление

---

<i>Часть I</i> ■ Первые шаги .....	26
1 ■ Фреймворк без фреймворка .....	27
2 ■ Ваш первый веб-компонент .....	45
3 ■ Делаем так, чтобы ваш компонент можно было использовать повторно .....	75
4 ■ Жизненный цикл компонента .....	106
5 ■ Реализация более качественного веб-приложения с помощью модулей .....	128
<i>Часть II</i> ■ Способы улучшить рабочий процесс вашего компонента .....	154
6 ■ Управление разметкой .....	155
7 ■ Шаблонирование контента с помощью HTML .....	182
8 ■ Теневая модель DOM .....	206
9 ■ Shadow CSS .....	222
10 ■ Проблемы Shadow CSS .....	251
<i>Часть III</i> ■ Объединяем компоненты воедино .....	274
11 ■ Реальный компонент пользовательского интерфейса .....	275
12 ■ Сборка и поддержка старых браузеров .....	309
13 ■ Тестирование компонентов .....	341
14 ■ События и поток данных приложения .....	363
15 ■ Соккрытие сложностей .....	396

# Содержание

---

Предисловие .....	15
От автора .....	17
Благодарности .....	19
Об этой книге .....	20
Об авторе .....	24
Об иллюстрации на обложке .....	25

## ЧАСТЬ I ПЕРВЫЕ ШАГИ ..... 26

<b>1</b>	<b>Фреймворк без фреймворка .....</b>	<b>27</b>
1.1	Что такое веб-компоненты? .....	30
1.1.1	Календарь с возможностью выбора даты .....	30
1.1.2	Теневая модель DOM .....	31
1.1.3	Что имеют в виду, когда говорят «веб-компоненты»? .....	33
1.1.4	Проблемная история импорта HTML .....	33
1.1.5	Библиотеки Polymer и X-Tag .....	35
1.1.6	Современные веб-компоненты .....	36
1.2	Будущее веб-компонентов .....	37
1.3	За пределами одного компонента .....	38
1.3.1	Веб-компоненты как и любой другой элемент DOM .....	39
1.3.2	От отдельного компонента к приложению .....	40
1.4	Ваш проект, ваш выбор .....	43
	Резюме .....	43

<b>2</b>	<b>Ваш первый веб-компонент .....</b>	<b>45</b>
2.1	Знакомство с HTMLElement .....	46
2.1.1	Ускоренный курс по наследованию .....	46
2.1.2	Наследование в ваших любимых элементах .....	47

2.2	Правила именования вашего элемента .....	48
2.3	Определение вашего пользовательского элемента (и обработка столкновений) .....	50
2.4	Расширение HTMLElement для создания логики пользовательского компонента .....	51
2.5	Использование вашего пользовательского элемента на практике .....	56
2.6	Создание (полезного) первого компонента .....	59
2.6.1	Настраиваем свой веб-сервер .....	59
2.6.2	Пишем свой HTML-тег .....	61
2.6.3	Создаем свой класс .....	62
2.6.4	Добавляем содержимое в наш компонент .....	63
2.6.5	Добавляем стили .....	64
2.6.6	Логика компонента .....	65
2.6.7	Добавляем интерактивности .....	67
2.6.8	Последние штрихи .....	69
2.6.9	Улучшение компонента .....	73
2.7	Примечания относительно поддержки в браузерах .....	73
	Резюме .....	74

<b>3</b>	<b>Делаем так, чтобы ваш компонент можно было использовать повторно .....</b>	<b>75</b>
3.1	Реальный компонент .....	76
3.1.1	Пример использования поиска в 3D .....	76
3.1.2	Начнем с HTTP-запроса .....	77
3.1.3	Обертываем свою работу в пользовательский компонент ...	77
3.1.4	Отображение результатов поиска .....	80
3.1.5	Стилизация нашего компонента .....	81
3.2	Делаем наш компонент настраиваемым .....	83
3.2.1	Создание API компонента с помощью устанавливающих методов .....	84
3.2.2	Используя наш API извне .....	84
3.3	Использование атрибутов для конфигурирования .....	86
3.3.1	Аргумент против API компонента .....	86
3.3.2	Реализация атрибутов .....	87
3.3.3	Чувствительность к регистру символов .....	88
3.4	Прослушивание изменений в атрибутах .....	89
3.4.1	Добавление поля ввода текста .....	89
3.4.2	Метод attributeChangedCallback .....	90
3.4.3	Атрибуты, за которыми ведется наблюдение .....	91
3.5	Делаем другие вещи еще более настраиваемыми .....	94
3.5.1	Использование метода hasAttribute для проверки существования атрибута .....	94
3.5.2	Полная настройка URL-адреса HTTP-запроса для разработки .....	95

3.5.3	Руководство по передовым методикам.....	96
3.5.4	Избегайте использования атрибутов для расширенных данных .....	96
3.5.5	Отражение свойств и атрибутов .....	97
3.6	Обновление компонента-ползунка .....	99
	Резюме .....	105

<b>4</b>	<b>Жизненный цикл компонента.....</b>	<b>106</b>
4.1	API веб-компонентов .....	106
4.2	Обработчик <code>connectedCallback</code> .....	107
4.2.1	Конструктор в сравнении с методом <code>connectedCallback</code> .....	111
4.3	Остальные методы жизненного цикла веб-компонента ...	114
4.3.1	Метод <code>disconnectedCallback</code> .....	114
4.3.2	Метод <code>adoptedCallback</code> .....	117
4.4	Сравнение с жизненным циклом React .....	118
4.5	Сравнение с жизненным циклом игрового движка.....	120
4.6	Жизненный цикл компонента v0.....	126
	Резюме .....	127

<b>5</b>	<b>Реализация более качественного веб-приложения с помощью модулей.....</b>	<b>128</b>
5.1	Использование тега <code>&lt;script&gt;</code> для загрузки ваших веб-компонентов .....	129
5.1.2	Крошечные сценарии более организованы, но усугубляют проблему со ссылками .....	131
5.1.3	Включение стилей CSS для самостоятельных компонентов .....	132
5.1.4	Ад зависимостей .....	134
5.2	Использование модулей для решения проблем зависимости .....	134
5.2.1	Создание музыкального инструмента с использованием веб-компонентов и модулей JS .....	135
5.2.2	Начинаем с самого маленького компонента .....	138
5.2.3	Импорт и вложение веб-компонента в веб-компонент.....	139
5.2.4	Использование веб-компонента для обертки всего веб-приложения.....	141
5.3	Добавляем интерактивности в наш компонент.....	143
5.3.1	Прослушивание событий движения мыши .....	144
5.3.2	Передача данных в дочерние компоненты.....	144
5.3.3	Заставляем наши компоненты вибрировать с помощью CSS.....	146
5.4	Обертывание сторонних библиотек в виде модулей .....	148
5.4.1	Инструменты пользовательского интерфейса для обертывания модуля с помощью Node.js .....	148

5.4.2	Не идеально, но работает .....	149
5.4.3	Использование обернутого модуля для воспроизведения нот .....	149
5.4.4	Больше никакого автовоспроизведения аудио .....	151
5.4.5	Игра на веб-арфе .....	153
Резюме .....		153

## ЧАСТЬ II СПОСОБЫ УЛУЧШИТЬ РАБОЧИЙ ПРОЦЕСС ВАШЕГО КОМПОНЕНТА ..... 154

<b>6</b>	<b>Управление разметкой</b> .....	155
6.1	Строки. Теория .....	156
6.1.1	Когда <code>innerHTML</code> становится уродливым .....	156
6.2	Использование шаблонных литералов .....	157
6.2.1	Приложение для создания визиток .....	158
6.3	Импорт шаблонов .....	161
6.3.1	Хранение разметки вне логики основного компонента .....	162
6.3.2	Модуль для <code>HTML</code> и <code>CSS</code> .....	162
6.4	Логика шаблона .....	165
6.4.1	Создание меню из данных .....	166
6.4.2	Больше логики генерации и более жесткая автоматизация .....	167
6.5	Кеширование элементов .....	168
6.5.1	Не заставляйте меня использовать метод <code>querySelector</code> в моем компоненте .....	169
6.6	Умные шаблоны .....	171
6.6.1	Использование <code>lit-html</code> .....	172
6.6.2	Модуль <code>repeat</code> .....	172
6.6.3	Нужно ли вам использовать это? .....	174
6.6.4	Внедрение слушателей событий в разметку .....	175
6.7	Обновление ползунка .....	177
Резюме .....		181

<b>7</b>	<b>Шаблонирование контента с помощью <code>HTML</code></b> .....	182
7.1	Покойся с миром, <code>HTML</code> -импорт .....	183
7.1.1	Полифилинг .....	184
7.1.2	Что внутри .....	185
7.2	Тег <code>&lt;template&gt;</code> .....	187
7.2.1	Фрагменты документа .....	188
7.2.2	Использование содержимого шаблона .....	190
7.3	Выберите свой вариант шаблона .....	193
7.4	Динамически загружаемые шаблоны .....	196
7.5	Вход в теньевую модель <code>DOM</code> с помощью тег <code>&lt;slot&gt;</code> .....	200

7.5.1	Тег <code>&lt;slot&gt;</code> без имени .....	203
	Резюме .....	205

8	<b>Теневая модель DOM</b> .....	206
8.1	Инкапсуляция .....	207
8.1.1	Защита API вашего компонента .....	208
8.1.2	Защита DOM вашего компонента .....	209
8.2	Использование теневой модели DOM .....	211
8.2.1	Корень теневого дерева .....	213
8.2.2	Закрытый режим .....	215
8.2.3	Конструктор вашего компонента и метод <code>connectedCallback</code> : сравнение .....	218
	Резюме .....	221

9	<b>Shadow CSS</b> .....	222
9.1	Утечка стилей .....	222
9.1.1	Утечка стилей в нижестоящие компоненты .....	224
9.1.2	Утечка стилей в ваш компонент .....	225
9.2	Проблема утечки стилей решается с помощью теневой модели DOM .....	228
9.2.1	Когда происходит утечка стилей .....	231
9.3	План тренировок .....	233
9.3.1	Оболочка приложения .....	234
9.3.2	Селекторы <code>host</code> и <code>ID</code> .....	236
9.3.3	Сетка упражнений и список планов .....	238
9.4	Адаптируемые компоненты .....	242
9.4.1	Создание компонента упражнения .....	243
9.4.2	Стили компонента упражнений .....	245
9.5	Обновляем ползунок .....	248
	Резюме .....	250

10	<b>Проблемы Shadow CSS</b> .....	251
10.1	Контекстные селекторы .....	251
10.1.1	Немного интерактивности .....	252
10.1.2	Контекстные стили .....	256
10.1.3	Обходной путь .....	260
10.2	Темы компонента .....	262
10.2.1	Селекторы <code>::shadow</code> и <code>/deep/</code> .....	263
10.2.2	CSS-переменные .....	265
10.2.3	Применяем CSS-переменные в нашем примере .....	267
10.3	Использование теневой модели DOM на практике (сегодня) .....	269
10.3.1	Поддержка со стороны браузеров .....	269
10.3.2	Полизаполнение .....	270

10.3.3 Дизайн-системы .....	271
Резюме .....	273

## ЧАСТЬ III ОБЪЕДИНЯЕМ КОМПОНЕНТЫ ВОЕДИНО ..... 274

### 11 Реальный компонент пользовательского интерфейса ..... 275

11.1 Создаем палитру цветов .....	276
11.1.1 Компоненты нашего компонента .....	278
11.2 Компонент выбора координат .....	280
11.2.1 Класс инструмента выбора координат .....	280
11.2.2 HTML-код и стили инструмента для выбора координат .....	284
11.2.3 Демостраницы для компонентов .....	285
11.3 Палитра цветов .....	287
11.3.1 Наблюдение за изменениями атрибутов для взаимодействия .....	292
11.3.2 Реакция на изменения в полях ввода .....	294
11.3.3 Реакция на изменения атрибутов .....	296
11.4 Работаем над внешним видом палитры .....	298
11.4.1 Загрузка CSS-переменных для улучшения дизайна .....	299
11.4.2 Использование импорта для более сложных стилей .....	302
Резюме .....	307

### 12 Сборка и поддержка старых браузеров ..... 309

12.1 Обратная совместимость .....	310
12.1.1 Включение теневой модели DOM .....	311
12.1.2 Сравнение с полифилами .....	315
12.1.3 Shadow CSS и дочерние элементы .....	316
12.2 Наименьший общий знаменатель .....	319
12.3 Процессы сборки .....	321
12.3.1 Использование сценариев NPM .....	322
12.4 Сборка компонентов .....	323
12.4.1 Почему мы выполняем сборку .....	324
12.4.2 Компоновка модулей с помощью Rollup .....	326
12.4.3 Запуск сборки с помощью прт .....	330
12.5 Транспилиция для IE .....	332
12.5.1 Babel .....	333
12.5.2 CSS-vars-ponyfill .....	337
Резюме .....	339



<b>13</b>	<b>Тестирование компонентов</b>	341
13.1	Модульное тестирование и разработка через тестирование	342
13.2	Web Component Tester	343
13.2.1	Пишем тесты	347
13.3	Сравнение со стандартной тестовой конфигурацией при использовании Karma	352
13.3.1	Плагин karma-web-components	359
13.3.2	Несколько тестов в одном проекте	361
13.3.3	Замечание относительно Safari	362
	Резюме	362
<b>14</b>	<b>События и поток данных приложения</b>	363
14.1	Использование фреймворков	364
14.2	События	365
14.2.1	Нативные события и WebComponentsReady	365
14.2.2	Когда определяются пользовательские элементы	367
14.2.3	Пользовательские события	368
14.2.4	Всплытие пользовательского события	370
14.3	Передача событий через веб-компоненты	372
14.3.1	Распространение нативных событий с помощью теневой модели DOM	373
14.3.2	Распространение пользовательских событий с помощью теневой модели DOM	374
14.4	Разделение данных	376
14.4.1	Модель–представление–контроллер	377
14.4.2	Локальное хранилище	380
14.4.3	Подключение пользовательского интерфейса к модели данных	383
14.5	Воспроизведение упражнений	386
14.6	Передача событий с помощью шины	390
14.6.1	Статические методы чтения и типы событий	393
14.6.2	Шаблоны проектирования как рекомендация	394
	Резюме	395
<b>15</b>	<b>Соккрытие сложностей</b>	396
15.1	Взгляд в будущее веб-компонентов	397
15.2	3D и смешанная реальность	399
15.2.1	A-Frame	402
15.2.2	Компонент model-viewer	406
15.2.3	model-viewer и поиск с помощью Poly	408
15.2.4	Дополненная реальность и model-viewer	410
15.2.5	Ваш собственный 3D-компонент	413
15.3	Видеоэффекты	422

15.3.1	Обработка пикселей с помощью JavaScript.....	423
15.3.2	Шейдеры WebGL .....	426
15.4	Отслеживание движений рук и машинное обучение .....	429
	Резюме .....	435
Приложение	ES2015 для веб-компонентов .....	436
Указатель	.....	460