

# Vue.js

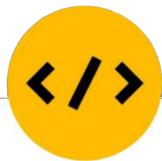
## В ДЕЙСТВИИ

Эрик Хэнчett,  
Бенджамин Листуон



MANNING





@CODELIBRARY\_IT

# *Vue.js in Action*

ERIK HANCHETT  
WITH BEN LISTWON



MANNING  
SHELTER ISLAND

Эрик Хэнчett, Бенджамин Листуон

# Vue.js в действии



Санкт-Петербург · Москва · Екатеринбург · Воронеж  
Нижний Новгород · Ростов-на-Дону · Самара · Минск

**2019**

ББК 32.988-02-018  
УДК 004.738.5  
Х88

## Хэнчэтт Эрик, Листуон Бенджамин

Х88 **Vue.js в действии.** — СПб.: Питер, 2019. — 304 с.: ил. — (Серия «Библиотека программиста»).

**ISBN 978-5-4461-1098-8**

Vue.js — это популярная библиотека для создания пользовательских интерфейсов. В ней значительно переосмыслены реактивные идеи, впервые появившиеся в React.js.

Книга «Vue.js в действии» рассказывает о создании быстрых и эластичных пользовательских интерфейсов для Интернета. Освоив ее, вы напишете полноценное приложение для интернет-магазина, где будут присутствовать списки товаров, админка, а также организован полноценный процесс онлайнового заказа.

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

ББК 32.988-02-018  
УДК 004.738.5

Права на издание получены по соглашению с Apress. 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 978-1617294624 англ.  
ISBN 978-5-4461-1098-8

© 2018 by Manning Publications Co. All rights reserved.  
© Перевод на русский язык ООО Издательство «Питер», 2019  
© Издание на русском языке, оформление ООО Издательство «Питер», 2019  
© Серия «Библиотека программиста», 2019

# Краткое содержание

Вступление .....	14
Предисловие .....	15
Благодарности .....	16
Об этой книге .....	18

## Часть I. Знакомимся с Vue.js

<b>Глава 1.</b> Введение в Vue.js .....	24
<b>Глава 2.</b> Экземпляр Vue .....	38

## Часть II. Представление и модель представления

<b>Глава 3.</b> Поддержка интерактивности .....	60
<b>Глава 4.</b> Формы и поля ввода .....	83
<b>Глава 5.</b> Условные выражения, циклы и списки .....	103
<b>Глава 6.</b> Работа с компонентами .....	123
<b>Глава 7.</b> Продвинутые компоненты и маршрутизация .....	144
<b>Глава 8.</b> Переходы и анимация .....	180
<b>Глава 9.</b> Расширение Vue .....	197

## Часть III. Моделирование данных, работа с API и тестирование

<b>Глава 10.</b> Vuex .....	218
<b>Глава 11.</b> Взаимодействие с сервером .....	237
<b>Глава 12.</b> Тестирование .....	270

## Приложения

<b>Приложение А.</b> Настройка среды разработки .....	290
<b>Приложение Б.</b> Ответы на вопросы .....	297
Шпаргалка .....	300

# Оглавление

Вступление .....	14
Предисловие .....	15
Благодарности .....	16
Об этой книге .....	18
Целевая аудитория .....	19
Структура книги .....	19
Исходный код .....	20
Требования к программному обеспечению .....	20
Интернет-ресурсы .....	21
Дополнительная информация .....	21
Об авторе .....	21
Об иллюстрации на обложке .....	22

## Часть I. Знакомимся с Vue.js

<b>Глава 1. Введение в Vue.js .....</b>	24
1.1. На плечах у гигантов .....	25
1.1.1. Шаблон проектирования MVC .....	25
1.1.2. Шаблон проектирования MVVM .....	27
1.1.3. Что такое реактивное приложение .....	28
1.1.4. Калькулятор на JavaScript .....	29
1.1.5. Калькулятор на Vue .....	32
1.1.6. Сравнение JavaScript и Vue .....	33
1.1.7. Как Vue использует MVVM и реактивность .....	33

1.2. Почему именно Vue.js .....	35
1.3. Мысли на будущее.....	36
Резюме.....	37
<b>Глава 2. Экземпляр Vue.....</b>	<b>38</b>
2.1. Наше первое приложение.....	38
2.1.1. Корневой экземпляр Vue.....	39
2.1.2. Убедимся в том, что приложение работает .....	41
2.1.3. Выведем что-нибудь внутри представления.....	44
2.1.4. Исследование свойств в Vue .....	45
2.2. Жизненный цикл Vue .....	47
2.2.1. Добавление хуков жизненного цикла.....	47
2.2.2. Исследование кода жизненного цикла .....	49
2.2.3. Стоит ли оставлять код жизненного цикла? .....	51
2.3. Вывод товара.....	51
2.3.1. Определение данных товара .....	51
2.3.2. Разметка представления товара .....	52
2.4. Применение фильтров вывода .....	55
2.4.1. Написание функции фильтра.....	56
2.4.2. Добавление фильтра в разметку и проверка разных значений .....	57
Упражнение.....	58
Резюме.....	58

## Часть II. Представление и модель представления

<b>Глава 3. Поддержка интерактивности .....</b>	<b>60</b>
3.1. Корзина покупок начинается с добавления массива.....	60
3.2. Привязка к событиям DOM .....	62
3.2.1. Основы привязки событий .....	62
3.2.2. Привязывание события к кнопке добавления в корзину.....	63
3.3. Кнопка для подсчета и вывода элементов в корзине .....	64
3.3.1. Когда следует использовать вычисляемые свойства .....	65
3.3.2. Проверка событий обновления с помощью вычисляемых свойств .....	66
3.3.3. Вывод количества элементов в корзине и тестирование .....	72

<b>3.4. Сделаем кнопку дружественной для пользователей.....</b>	<b>73</b>
3.4.1. Учет товара .....	74
3.4.2. Учет и работа с вычисляемыми свойствами .....	75
3.4.3. Знакомство с директивой v-show .....	76
3.4.4. Отображение и скрытие кнопки с помощью v-if и v-else.....	77
3.4.5. Добавление кнопки корзины в виде переключателя .....	79
3.4.6. Отображение страницы оформления заказа с помощью директивы v-if.....	80
3.4.7. Сравнение v-show с v-if/v-else .....	81
<b>Упражнение.....</b>	<b>82</b>
<b>Резюме.....</b>	<b>82</b>
<b>Глава 4. Формы и поля ввода .....</b>	<b>83</b>
4.1. Связывание с помощью v-model .....	84
4.2. Повторный взгляд на связывание значений .....	92
4.2.1. Привязка значений к флагажку .....	92
4.2.2. Привязка значений и переключатели .....	93
4.2.3. Знакомство с директивой v-for .....	95
4.2.4. Директива v-for без обязательного аргумента key .....	97
4.3. Знакомство с модификаторами .....	98
4.3.1. Использование модификатора .number .....	99
4.3.2. Обрезка вводимых значений.....	100
4.3.3. Модификатор .lazy .....	102
<b>Упражнение.....</b>	<b>102</b>
<b>Резюме.....</b>	<b>102</b>
<b>Глава 5. Условные выражения, циклы и списки.....</b>	<b>103</b>
5.1. Сообщения о наличии товара .....	103
5.1.1. Сообщение об оставшемся товаре с помощью v-if.....	104
5.1.2. Дополнительные сообщения с использованием v-else и v-else-if .....	106
5.2. Циклический перебор товара .....	108
5.2.1. Добавление звездного рейтинга на основе диапазона v-for .....	108
5.2.2. Привязка HTML-класса к звездному рейтингу.....	110
5.2.3. Информация о товарах .....	112

5.2.4. Импорт товаров из файла product.json.....	114
5.2.5. Рефакторинг приложения с использованием директивы v-for .....	115
5.3. Сортировка записей.....	120
Упражнение.....	122
Резюме.....	122
<b>Глава 6. Работа с компонентами .....</b>	<b>123</b>
6.1. Что такое компоненты .....	123
6.1.1. Создание компонентов .....	124
6.1.2. Локальная регистрация .....	125
6.2. Иерархия компонентов .....	127
6.3. Использование входных параметров для передачи данных.....	128
6.3.1. Передача литералов через входные параметры.....	128
6.3.2. Динамические входные параметры.....	129
6.3.3. Проверка входных параметров .....	133
6.4. Определение шаблона для компонента.....	135
6.4.1. Вложенные шаблоны .....	136
6.4.2. Атрибут text/x-template .....	137
6.4.3. Использование однофайловых компонентов .....	138
6.5. Работа с пользовательскими событиями .....	139
6.5.1. Отслеживание событий.....	139
6.5.2. Изменение входных параметров потомка с помощью .sync .....	141
Упражнение.....	142
Резюме.....	142
<b>Глава 7. Продвинутые компоненты и маршрутизация.....</b>	<b>144</b>
7.1. Работа со слотами.....	145
7.2. Именованные слоты .....	148
7.3. Слоты с ограниченной областью видимости.....	150
7.4. Создание приложения с динамическими компонентами .....	152
7.5. Создание асинхронных компонентов .....	154
7.6. Рефакторинг зоомагазина с помощью Vue-CLI.....	156
7.6.1. Создание нового приложения с помощью Vue-CLI.....	157
7.6.2. Настройка маршрутов .....	159

7.6.3. Добавление CSS, Bootstrap и Axios .....	160
7.6.4. Подготовка компонентов.....	162
7.6.5. Создание компонента Form .....	164
7.6.6. Добавление компонента Main .....	165
7.7. Маршрутизация .....	168
7.7.1. Добавление параметризированного маршрута product .....	168
7.7.2. Настройка router-link с помощью тегов .....	171
7.7.3. Оформление router-link с помощью стилей .....	173
7.7.4. Добавление дочернего маршрута edit .....	174
7.7.5. Перенаправление и подстановочные маршруты .....	177
Упражнение.....	179
Резюме .....	179
<b>Глава 8. Переходы и анимация.....</b>	<b>180</b>
8.1. Что такое переходы .....	180
8.2. Основы анимации .....	185
8.3. JavaScript-хуки .....	186
8.4. Переход между компонентами .....	189
8.5. Обновление зоомагазина .....	192
8.5.1. Добавление перехода в проект зоомагазина.....	192
8.5.2. Добавление анимации в проект зоомагазина .....	193
Упражнение.....	196
Резюме .....	196
<b>Глава 9. Расширение Vue .....</b>	<b>197</b>
9.1. Повторное использование кода с помощью примесей .....	197
9.1.1. Глобальные примеси .....	202
9.2. Пользовательские директивы с примерами .....	203
9.2.1. Глобальные пользовательские директивы с модификаторами, значениями и аргументами .....	206
9.3. Функция отрисовки и JSX .....	208
9.3.1. Пример функции отрисовки .....	209
9.3.2. Пример использования JSX .....	212
Упражнение.....	216
Резюме .....	216

### Часть III. Моделирование данных, работа с API и тестирование

<b>Глава 10.</b> Vuex .....	218
10.1. Для чего нам Vuex .....	218
10.2. Состояние и мутации в Vuex.....	222
10.3. Геттеры и действия.....	225
10.4. Использование Vuex в проекте зоомагазина с помощью Vue-CLI .....	228
10.4.1. Установка Vuex с помощью Vue-CLI.....	228
10.5. Вспомогательные методы Vuex.....	231
10.6. Краткое введение в модули .....	234
Упражнение.....	236
Резюме.....	236
<b>Глава 11.</b> Взаимодействие с сервером.....	237
11.1. Отрисовка на стороне сервера .....	238
11.2. Введение в Nuxt.js .....	240
11.2.1. Создание приложения для поиска музыки.....	241
11.2.2. Создание проекта и установка зависимостей .....	243
11.2.3. Создание элементов интерфейса и компонентов.....	246
11.2.4. Обновление стандартной разметки .....	249
11.2.5. Добавление хранилища на основе Vuex .....	249
11.2.6. Использование промежуточных обработчиков .....	250
11.2.7. Создание маршрутов в Nuxt.js .....	251
11.3. Взаимодействие с сервером с помощью Firebase и VuexFire.....	256
11.3.1. Подготовка к работе с Firebase .....	257
11.3.2. Подключение Firebase к проекту зоомагазина.....	261
11.3.3. Хранение состояния аутентификации в Vuex.....	262
11.3.4. Поддержка аутентификации в компоненте Header .....	263
11.3.5. Работа с базой данных Realtime Database в файле Main.vue.....	267
Упражнение.....	268
Резюме.....	269
<b>Глава 12.</b> Тестирование .....	270
12.1. Создание тестовых сценариев.....	271
12.2. Непрерывная интеграция, доставка и развертывание .....	272
12.2.1. Непрерывная интеграция.....	272

12.2.2. Непрерывная доставка.....	273
12.2.3. Непрерывное развертывание.....	273
12.3. Виды тестов .....	274
12.4. Подготовка среды для тестирования.....	274
12.5. Создание первого тестового сценария с помощью vue-test-utils .....	277
12.6. Тестирование компонентов .....	280
12.6.1. Тестирование входных параметров.....	281
12.6.2. Тестирование текста.....	282
12.6.3. Тестирование CSS-классов.....	282
12.6.4. Симуляция Vuex .....	283
12.7. Настройка отладчика Chrome .....	285
Упражнение.....	288
Резюме.....	288

## Приложения

<b>Приложение А.</b> Настройка среды разработки.....	290
A.1. Chrome Developer Tools .....	290
A.2. vue-devtools для Chrome .....	291
A.3. Загрузка сопутствующего кода для отдельных глав .....	293
A.4. Установка Node.js и npm.....	293
A.4.1. Установка Node.js с помощью автоматических установщиков .....	294
A.4.2. Установка Node.js с помощью NVM .....	294
A.4.3. Установка Node.js с помощью системы управления пакетами в Linux .....	295
A.4.4. Установка Node.js с помощью MacPorts или Homebrew .....	295
A.4.5. Проверяем, установлен ли пакет Node .....	295
A.5. Установка Vue-CLI .....	296
<b>Приложение Б.</b> Ответы на вопросы .....	297
Глава 2.....	297
Глава 3.....	297
Глава 4.....	297
Глава 5.....	298

---

Глава 6.....	298
Глава 7.....	298
Глава 8.....	298
Глава 9.....	298
Глава 10.....	299
Глава 11.....	299
Глава 12.....	299
Шпаргалка.....	300

# Вступление

Клиентская веб-разработка стала на удивление сложной. Если вы никогда не имели дела с современными фреймворками для JavaScript, то можете потратить целую неделю на создание своей первой программы, которая всего лишь выводит приветствие! Это звучит нелепо, и здесь я с вами соглашусь. Проблема в том, что большинство фреймворков требуют навыков работы с терминалом, углубленного знания JavaScript, владения такими инструментами, как NPM (Node Package Manager – диспетчер пакетов Node), Babel, Webpack, – и часто это еще не полный список.

Однако к Vue все это не относится. Мы называем эту библиотеку прогрессивной, поскольку она масштабируется в *обе* стороны. Если приложение простое, используйте Vue по аналогии с JQuerу, вставляя элемент `script`. Но с ростом требований и приобретением вами новых навыков работа с Vue станет все более разносторонней и продуктивной.

У Vue есть еще одна отличительная черта. Этот проект разрабатывается не только программистами, но и дизайнерами, преподавателями и специалистами других гуманитарных профессий. Благодаря этому наши документация, руководства и инструменты для разработки одни из лучших в мире. Ощущения от *использования* Vue имеют для нас такое же значение, как производительность, надежность и гибкость.

Эрику удалось сделать эту книгу ориентированной на обычных людей. Прежде всего она необыкновенно наглядная. Множество подробных иллюстраций и снимков экрана с примечаниями неразрывно связывают приведенные здесь примеры с реальным процессом разработки. В итоге, чтобы закрепить свои знания, вам придется научиться работать с браузером и инструментами для разработки Vue и, что важнее, овладеть методами отладки при возникновении проблем.

Для читателей, у которых нет существенного опыта клиентской разработки, написания программ на JavaScript или даже просто программирования, Эрик тщательно излагает фундаментальные концепции и объясняет принцип работы и назначение Vue. Добавьте к этому методологию описания новых возможностей в контексте реальных проектов – и вы получите идеальную книгу для относительно неопытных разработчиков, которые начинают знакомство с современными клиентскими фреймворками с Vue и хотят расширить свои знания в этой области.

*Крис Фриц, член основной группы разработки Vue и куратор документации*

# Предисловие

Эту книгу мне предложили написать в начале 2017 года, после того, как по личным причинам от этой возможности вынужден был отказаться Бенджамин Листуон (Benjamin Listwon). Я как раз недавно получил диплом магистра делового администрирования университета Невады, а с момента публикации моей последней книги *Ember.js Cookbook* (Pact Publishing, 2016) прошел целый год. Я завел на YouTube канал *Program with Erik* («Программируем с Эриком»), и большая часть моего времени уходила на попытки записать как можно более качественные видеоуроки для небольшой, но растущей аудитории. Одновременно с этим я начал серию скринкастов, посвященных Vue.js, и получил положительные отзывы от зрителей. Это подтолкнуло меня к дальнейшему изучению Vue.

Я начал с прослушивания докладов Эвана Ю (Evan You), создателя Vue.js, и знакомства с планами развития этого фреймворка. Затем перешел к размещенным на YouTube бесчисленным руководствам и видеоурокам от других разработчиков. Начал заглядывать на форумы и в группы Facebook, чтобы узнать, о чем сейчас говорят. Куда бы я ни зашел, люди везде выражали свой восторг по поводу Vue.js и его будущего. Поэтому я и начал размышлять о возможности написания данной книги.

После долгих раздумий и разговора с женой я решил согласиться. К счастью, Бенджамин уже заложил отличный фундамент, поэтому я смог сразу приступить к работе. Следующие десять месяцев состояли из бесчисленных ночей и выходных, потраченных на исследования, тестирование и написание текста.

Не стану приукрашивать: книга далась мне нелегко. Скажем так, не все пошло по плану. Вдобавок к личным проблемам, пропущенным срокам и творческому кризису пришлось вносить существенные изменения после выхода новой версии Vue.js.

Но, несмотря на все это, я очень горжусь книгой. Каждая неудача заставляла меня работать с удвоенной энергией. Я решил закончить книгу и сделать ее максимально качественной. Надеюсь, это будет заметно при чтении.

Огромное спасибо читателям, которые купили эту книгу. Я очень надеюсь на то, что она поможет вам с изучением Vue.js. Пожалуйста, напишите мне о своих впечатлениях. Свяжитесь со мной через Twitter (@ErikCH), электронную почту ([erik@programwitherik.com](mailto:erik@programwitherik.com)) или список рассылки по адресу [goo.gl/UmemSS](http://goo.gl/UmemSS). Еще раз спасибо!

# Благодарности

Прежде всего я хотел бы поблагодарить свою жену Сьюзен, потому что без ее помощи эта книга никогда не была бы завершена. Спасибо сыну и дочке — Уайатту и Вивиан. Это ради них я так тяжело работаю. Спасибо всем рецензентам, членам форума Vue.js in Action и тем, кто прислал отзывы, — ваша помощь сделала эту книгу намного лучше. Благодарю также Криса Фрица (Chris Fritz) за прекрасное предисловие. Наконец, я хотел бы выразить безмерную и искреннюю признательность сообществу Vue.js, Эвану Ю и всем, кто внес свой вклад в этот замечательный фреймворк.

*Эрик Ханчетт*

В первую очередь я хотел бы выразить искреннюю благодарность своей жене Киффен за поддержку и поощрение — не только в то время, когда я участвовал в этом проекте, но и во всех аспектах нашей совместной жизни. Звездой и сердцем нашей семьи является сын Лео: спасибо тебе за неиссякаемые улыбки, объятия и радость. За одобрение, понимание и поддержку от всего сердца хотел бы поблагодарить редакцию издательства Manning. Я по-настоящему признателен Эрику, без которого эта книга никогда не увидела бы свет: желаю тебе всего наилучшего. Наконец, спасибо Эвану Ю и всем тем, кто помогает развитию Vue.js, — вы собрали воедино прекрасный программный продукт и еще более прекрасное сообщество, частью которого я имею честь быть.

*Бенджамин Листуон*

Выражаем совместную признательность техническому редактору Джою Келкару (Jay Kelkar), а также всем рецензентам, давшим на разных этапах свои отзывы. Среди них Алекс Миллер (Alex Miller), Алексей Галиуллин (Alexey Galiullin), Крис Коппенбаргер (Chris Coppenbarger), Клайв Харбер (Clive Harber), Дарко Божи-

новски (Darko Bozhinovski), Ферит Топцу (Ferit Topsu), Гарро Лиссенберг (Harro Lissenberg), Ян Петер Хервейер (Jan Pieter Herweijer), Джеспер Петерсен (Jesper Petersen), Лаура Стедмэн (Laura Steadman), Марко Летич (Marko Letic), Пауло Нуин (Paulo Nuin), Филиппе Шаррье (Philippe Charriere), Рохит Шарма (Rohit Sharma), Рональд Борман (Ronald Borman), Райан Гарви (Ryan Harvey), Райан Габер (Ryan Huber), Сандер Зегвельд (Sander Zegveld), Убалдо Пескаторе (Ubaldo Pescatore) и Витторио Марино (Vittorio Marino).

# Об этой книге

Позвольте мне, прежде чем приступить к программированию на Vue.js, остановиться на нескольких моментах, о которых вы должны знать.

Здесь вы найдете все, что может понадобиться для овладения Vue.js. Цель этой книги — дать вам знания, с помощью которых вы без колебаний присоединитесь к любому проекту, использующему эту библиотеку.

Проводя исследования для этой книги, я неоднократно слышал, что лучший источник информации для изучения Vue.js — официальная документация. Действительно, на официальном сайте есть отличные руководства, и я настоятельно рекомендую вам с ними ознакомиться. Однако они неидеальны и не охватывают все темы, поэтому советую использовать их как дополнительные справочные материалы. Во время написания книги я задался целью не останавливаться на темах, рассмотренных в официальной документации. Я сделал примеры более понятными и наглядными, чтобы вам проще было применять описанные здесь концепции к собственным проектам. Темы, которые показались мне недостаточно важными или просто не вошли в книгу, даны в виде ссылок на официальные руководства.

Этой книгой можно пользоваться по-разному. Прочитав ее с начала до конца, вы получите полноценное представление о Vue.js. Или работайте с ней как со справочником, просматривая информацию о тех функциях, о которых вам хочется знать больше. Оба подхода допустимы.

Позже мы перейдем к созданию приложений на Vue.js с помощью системы сборки. Не волнуйтесь, инструкции по использованию соответствующего инструмента под названием Vue-CLI приведены в приложении А. Vue-CLI помогает создавать более сложные приложения, беря на себя сборку и преобразование исходного кода.

Лейтмотивом этой книги будет разработка на основе Vue.js приложения зоомагазина. В некоторых главах этот пример рассматривается чаще, в других — реже. Это было сделано специально, чтобы вы могли легко усвоить абстрактные концепции без связи с конкретным проектом. Вместе с тем мы не забываем и о читателях, которые предпочитают учиться на реальных примерах.

## Целевая аудитория

Эта книга предназначена для всех, кто заинтересован в изучении Vue.js и имеет опыт работы с JavaScript, HTML и CSS. От вас не требуются глубокие знания этой области, но понимание основ, таких как массивы, переменные, циклы и HTML-элементы, не помешает. Что касается CSS, то мы будем задействовать библиотеку Bootstrap 3. При этом, чтобы выполнять упражнения, вам не обязательно о ней что-либо знать — она применяется лишь для визуального оформления.

В начале книги представлен код на языке ECMAScript 2015, известном также как ES6. Желательно ознакомиться с его спецификацией, прежде чем приступать к чтению. По большей части я использую всего несколько возможностей ES6, таких как стрелочные функции и импорт. Я дам знать, когда мы будем переходить на этот язык.

## Структура книги

Книга разбита на три части. Часть I знакомит с Vue.js. В главах 1 и 2 вы создадите свое первое приложение, а также узнаете, что такое экземпляр Vue и как он относится к нашему коду.

В части II, а именно в главах 3–9, мы более подробно рассмотрим концепции представления (View) и модели-представления (ViewModel), а также самые «сочные» аспекты Vue.js. Можно сказать, что часть I — всего лишь затравка, а основной курс начинается с части II. Вы узнаете о сложностях, связанных с созданием программ на Vue.js. Мы начнем с изучения реактивной модели, а затем создадим приложение зоомагазина, которое будет использоваться в дальнейшем.

Мы также рассмотрим формы, поля ввода и привязку информации с помощью мощных директив, встроенных в Vue.js, после чего перейдем к условным выражениям и циклам.

Чрезвычайно важны главы 6 и 7. В них вы научитесь разбивать приложение Vue.js на несколько логических частей, используя компоненты, а также познакомитесь с инструментами сборки, которые понадобятся для построения программ.

Помимо прочего, в главе 7 затрагивается тема маршрутизации. До этого для навигации по приложению будем применять простые условные выражения. Благодаря маршрутизации мы сможем правильно переходить между компонентами и передавать информацию между маршрутами.

Глава 8 познакомит вас с гибкой анимацией и переходами, которые выполняются в Vue.js. Эти замечательные возможности, встроенные в язык, заслуживают вашего внимания.

В главе 9 вы научитесь расширять Vue с помощью примесей и пользовательских директив без дублирования кода.

Часть III посвящена моделированию данных, работе с API и тестированию. В главах 10 и 11 мы подробно рассмотрим систему управления состоянием в Vue

под названием Vuex, а затем попробуем пообщаться с серверной стороной. Вы также познакомитесь с Nuxt.js — фреймворком для серверной отрисовки.

Тестирование рассматривается в главе 12. Мы обсудим базовые навыки, необходимые в любой профессиональной среде.

## Исходный код

Эта книга содержит множество примеров исходного кода в виде пронумерованных листингов и элементов строк. Чтобы отличить их от обычного текста, используется моноширинный шрифт. Иногда изменения, вносимые на следующем этапе, выделяются жирным шрифтом, например, когда к уже существующей строке кода добавляется что-то новое.

Во многих случаях оригинальный исходный код был переформатирован — я добавил переносы строк и убрал отступы, чтобы сэкономить место. Иногда даже этого оказывалось недостаточно и приходилось вставлять символы продолжения строки (→). Кроме того, при описании исходного кода часто убирал из него комментарии. Во многих листингах есть примечания, которые выделяют важные моменты.

Исходный код для книги доступен для загрузки на сайте издательства ([www.manning.com/books/vue-js-in-action](http://www.manning.com/books/vue-js-in-action)) и в моем репозитории на GitHub ([github.com/ErikCH/VuejsInActionCode](https://github.com/ErikCH/VuejsInActionCode)). Дополнительные инструкции по загрузке исходного кода и настройке среды программирования можно найти в приложении А.

При чтении книги вы заметите, что я часто разбиваю листинги на отдельные файлы. В архиве для каждой главы код имеется как в виде единого файла, так и в виде нескольких модулей, чтобы было легче ориентироваться. Если вы найдете ошибку в коде, не стесняйтесь отправить мне письмо. Я буду поддерживать свой репозиторий на GitHub и комментировать любые обновления в файле README.

## Требования к программному обеспечению

Чтобы вам было легче, весь код в этой книге адаптирован к работе в любом современном браузере. Я лично проверил его в Firefox 58, Chrome 65 и Microsoft Edge 15. Не рекомендую запускать представленные здесь примеры в более старых браузерах, так как вы неминуемо столкнетесь с какими-нибудь проблемами. Сама библиотека Vue.js не поддерживает IE8 и более старые версии. Совместимость с ECMAScript 5 обязательна.

В нескольких начальных главах я использую возможности языка ES6. Для выполнения этих примеров вам понадобится современный браузер.

Приложение зоомагазина, которое мы будем разрабатывать в этой книге, может работать в мобильном браузере. Но, поскольку оно для этого не оптимизировалось, советую запускать его на настольном компьютере.

Вам не нужно беспокоиться о выборе операционной системы — главное, чтобы работал браузер. Это фактически единственное требование.

## Интернет-ресурсы

Как уже упоминалось, в ходе работы с примерами, представленными в книге, отличным подспорьем послужит официальное руководство Vue.js. Оно находится по адресу [vuejs.org/v2/guide/](https://vuejs.org/v2/guide/) и постоянно обновляется.

Ведется также список избранных проектов, относящихся к Vue.js, который вы найдете на странице GitHub [github.com/vuejs/awesome-vue](https://github.com/vuejs/awesome-vue). Там есть ссылки на подкасты, дополнительные ресурсы, сторонние библиотеки и даже компании, которые работают с Vue.js. Настоятельно рекомендую ознакомиться с ним.

У Vue.js огромное и активно растущее сообщество пользователей. Одно из лучших мест в Сети для общения с другими разработчиками на Vue.js — официальный форум проекта [forum.vuejs.org/](https://forum.vuejs.org/). Там вы сможете получить помощь и обсудить все, что касается этой библиотеки.

Если вы ищете видеоуроки, загляните на страницу моего канала на YouTube [erik.video](https://www.youtube.com/user/erikvideo), который содержит множество материала по Vue.js и JavaScript в целом.

## Дополнительная информация

В этой книге я раскрыл большое количество тем. Если вы застряли в каком-то месте или нуждаетесь в помощи, то всегда можете связаться непосредственно со мной. Если я не смогу вам помочь, то по крайней мере направлю в нужное русло. Не стесняйтесь. Члены сообщества Vue.js дружелюбны к новичкам.

В процессе чтения пытайтесь самостоятельно реализовывать изученные концепции. Практическое применение — один из лучших способов усвоения материала. Вместо зоомагазина попытайтесь создать собственную торговую площадку, используя эту книгу как руководство.

И последнее: подходите к работе творчески и создайте что-нибудь интересное. Если у вас получится, обязательно напишите мне в Twitter: @ErikCH!

## Об авторе



Эрик Ханчетт занимается веб-разработкой на протяжении последних десяти лет. Он автор книги *Ember.js Cookbook* (Packt Publishing, 2016), канала YouTube [erik.video](https://www.youtube.com/user/erikvideo) и блога по адресу [programwitherik.com](http://programwitherik.com). Кроме того, Эрик ведет список рассылки [goo.gl/UmemSS](http://goo.gl/UmemSS), в котором делится секретами разработки на JavaScript. Свободное от работы или написания кода время он проводит со своими детьми Уайаттом и Вивиан и женой Сьюзен.

## Об иллюстрации на обложке

Иллюстрация на обложке называется «Костюм молодой русской торговки с Охты в 1765 году»<sup>1</sup> и позаимствована из четырехтомника Томаса Джейффериса (Thomas Jefferys) «Коллекция платья разных народов, старины и современного»<sup>2</sup>, изданного в Лондоне в 1757–1772 годах. На титульном листе утверждается, что это гравюра ручной раскраски с применением гуммиарабика.

Томаса Джейффериса (1719–1771) называли географом при короле Георге III. Это английский картограф и ведущий поставщик карт своего времени. Он чертил и печатал карты для правительства и других государственных органов. На его счету целый ряд коммерческих карт и атласов, в основном Северной Америки. Занимаясь картографией в разных уголках мира, он проявлял интерес к местным традиционным нарядам, которые впоследствии были блестящие переданы в данной коллекции. В конце XVIII века заморские путешествия для собственного удовольствия были относительно новым явлением, поэтому подобные коллекции пользовались популярностью, позволяя получить представление о жителях других стран как настоящим туристам, так и «диванным» путешественникам.

Разнообразие иллюстраций в книгах Джейффериса — яркое свидетельство уникальности и оригинальности народов мира в то время. С тех пор тенденции в одежде сильно изменились, а региональные и национальные различия, которые были такими значимыми 200 лет назад, постепенно сошли на нет. В наши дни часто сложно отличить друг от друга жителей разных континентов. Оптимисты могут сказать, что взамен культурного и визуального многообразия мы получили более насыщенную и интересную личную жизнь (или, по крайней мере, ее интеллектуальную и техническую стороны).

В то время как большинство компьютерных изданий мало чем отличаются друг от друга, компания Manning выбирает для своих книг обложки, показывающие богатое региональное разнообразие, которое Джейфферис воплотил в своих иллюстрациях два столетия назад. Это ода находчивости и инициативности современной компьютерной индустрии.

<sup>1</sup> Habit of a Young Market Woman of Octha in Russia in 1765.

<sup>2</sup> A Collection of the Dresses of Different Nations, Ancient and Modern.

# Часть I

## Знакомимся с Vue.js

Прежде чем переходить к изучению всех тех крутых возможностей, которые предлагает Vue, нужно поближе познакомиться с этой библиотекой. В первых двух главах мы рассмотрим философию, стоящую за Vue.js и шаблоном проектирования MVVM, а также сравним ее с аналогичными проектами.

Усвоив основы, на которые опирается Vue.js, мы подробно изучим корневой экземпляр Vue (сердце любого приложения) и его структуру. Затем вы узнаете, как привязать к нему программные данные.

Эти главы станут хорошим началом. Прочитав их, вы сможете создавать простые приложения и будете знать, как работает Vue.js.

# 1

## Введение в Vue.js

### В этой главе

- Шаблоны проектирования MVC и MVVM.
- Определение реактивного приложения.
- Описание жизненного цикла Vue.
- Оценка архитектуры Vue.js.

Мы уже давно привыкли к интерактивным сайтам. В середине 2000-х годов, на заре Web 2.0, интерактивность и вовлечение пользователей в процесс были в центре внимания. Именно в этот период появились компании Twitter, Facebook и YouTube. Благодаря бурному росту социальных сетей и пользовательского контента Интернет менялся в лучшую сторону.

Чтобы поспевать за этими изменениями и предоставлять большую интерактивность, разработчики начали создавать библиотеки и фреймворки, которые упрощали построение интерактивных сайтов. В 2006 году Джон Резиг (John Resig) выпустил jQeуту, тем самым значительно облегчив написание клиентских скриптов внутри HTML. Со временем появились и другие подобные проекты. Сначала они были большими, монолитными и навязывали разработчикам свое видение. Теперь же произошел сдвиг в сторону компактных облегченных библиотек, которые очень просто добавить в любое приложение. Вот мы и подошли к Vue.js.

Vue.js – это библиотека, позволяющая внедрять интерактивное поведение и дополнительные возможности в любой контекст, в котором выполняется JavaScript. Vue можно использовать как на отдельных страницах, решая простые задачи, так и в качестве фундамента для полноценных промышленных приложений.

### ПРИМЕЧАНИЕ

В Интернете названия Vue и Vue.js практически взаимозаменяемы. В этой книге я чаще использую более простой вариант, Vue, оставляя Vue.js для тех случаев, когда речь идет непосредственно о коде или библиотеке.

Вы увидите, как Vue и вспомогательные библиотеки позволяют создавать полноценные сложные веб-приложения. Мы исследуем все, от интерфейса, с которым взаимодействуют посетители, до базы данных, снабжающей наш код информацией.

Увидим также, каким образом каждый из приводимых здесь примеров вписывается в общую картину, рассмотрим лучшие методики, которые предлагает индустрия веб-разработки, и научимся интегрировать представленный далее код в собственные проекты, как существующие, так и новые.

Эта книга рассчитана в основном на веб-разработчиков, у которых уже есть некоторый опыт работы с JavaScript и общее представление об HTML и CSS. Тем не менее благодаря универсальности своего *программного интерфейса* (application programming interface, API) Vue подходит для разработчиков всех мастей и проектов любой степени сложности. Она станет надежным подспорьем, даже если вам нужно создать лишь небольшой прототип или простую любительскую программу для личного пользования.

## 1.1. На плечах у гигантов

Прежде чем начинать писать код для нашего первого приложения или углубляться в недра Vue, следует обратиться к истории программного обеспечения. Лишь зная о проблемах и вызовах, с которыми сталкивался мир веб-разработки в прошлом, можно по достоинству оценить преимущества, которыми обладает эта библиотека.

### 1.1.1. Шаблон проектирования MVC

Клиентский шаблон проектирования «*модель – представление – контроллер*» (Model – View – Controller, MVC) лежит в основе множества современных фреймворков для разработки веб-приложений (если вы уже знакомы с MVC, можете пролистывать дальше).

Здесь следует упомянуть, что со временем концепция MVC претерпела некоторые изменения. В так называемой классической версии MVC был предусмотрен отдельный набор правил для того, каким образом должны взаимодействовать между собой представление, контроллер и модель. Мы же остановимся на упрощенной интерпретации этого клиентского шаблона проектирования, которая лучше подходит для современных реалий.

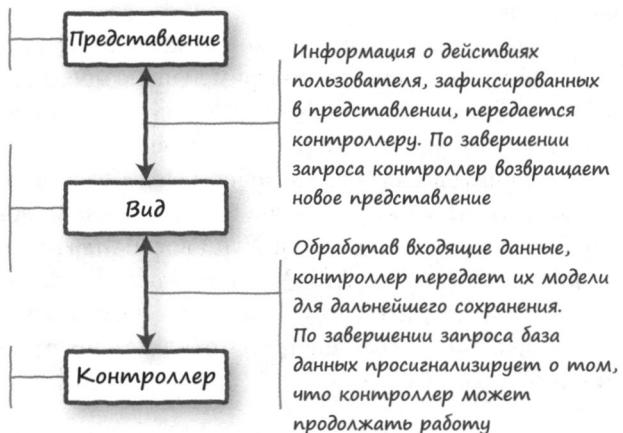
MVC применяется для разделения ответственности в приложении (рис. 1.1). Представление отвечает за вывод информации пользователю – это *графический пользовательский интерфейс* (graphical user interface, GUI). В центре находится контроллер. Он помогает передавать события из представления в модель, а данные – из

модели в представление. Внизу мы видим модель, которая содержит бизнес-логику и может предоставлять некое хранилище данных.

Представление отвечает за вывод информации пользователю

Контроллер служит посредником. Он помогает собирать и преобразовывать данные из модели, чтобы передать их представлению, а также преобразовывает и направляет соответствующие события из представления в модель

В типичном MVC-приложении модель представлена бизнес-логикой (или логикой предметной области). Она может содержать базу данных



**Рис. 1.1.** Взаимоотношения модели, представления и контроллера согласно MVC

## ДОПОЛНИТЕЛЬНО

Если хотите узнать больше о шаблоне проектирования MVC, можете начать со статьи Мартина Фаулера (Martin Fowler), посвященной эволюции MVC: [martinfowler.com/eaaDev/uiArchs.html](http://martinfowler.com/eaaDev/uiArchs.html).

Авторы веб-фреймворков любят этот шаблон проектирования за его надежную, проверенную временем архитектуру. Если вас интересует устройство современных веб-фреймворков, советую почитать книгу Эммита А. Скотта-младшего (Emmitt A. Scott Jr.) *SPA Design and Architecture* (Manning, 2015).

В современной разработке MVC часто задействуется в рамках одного приложения, предоставляя отличный механизм для разделения программного кода на разные роли. В сайтах, основанных на MVC, каждый запрос инициирует передачу потока информации от клиента к серверу, от сервера к базе данных, а затем обратно. Этот процесс отнимает много времени и ресурсов, что сказывается на отзывчивости пользовательского интерфейса.

Со временем приложения, основанные на веб-технологиях, стали более интерактивными. Теперь они применяют асинхронные запросы и клиентскую архитектуру MVC, чтобы передача данных на сервер не блокировала выполнение и не требовала ожидания ответа. Веб-страницы все больше напоминают настольные программы, поэтому задержки, связанные с клиент-серверным взаимодействием, могут замедлить или даже нарушить работу приложения. И вот тут на выручку приходит следующий шаблон.

### Несколько слов о бизнес-логике

Клиентский шаблон проектирования MVC отличается большой гибкостью при выборе места реализации бизнес-логики. На рис. 1.1 логика сконцентрирована в модели. Но это было сделано лишь для простоты, можно было выбрать для этого другие уровни приложения, включая контроллер. С тех пор как в 1979 году шаблон MVC был представлен Трюгве Ринсгаутом (Trygve Reenskaug) для языка Smalltalk-76, он претерпел некоторые изменения.

Представьте процесс проверки почтового индекса, введенного пользователем.

- Представление может содержать код на JavaScript, который проверяет почтовый индекс по мере ввода или перед отправкой.
- Модель способна проверить почтовый индекс при создании объекта для хранения входящих данных.
- База данных может накладывать ограничения на поле почтового индекса, то есть модель тоже применяет бизнес-логику. Некоторые разработчики считают это плохим тоном.

Иногда бывает сложно определить, что именно представляет собой бизнес-логика. Мы отдельно остановимся на том, как и где она должна реализовываться. Вы также узнаете, каким образом Vue и сопутствующие библиотеки помогают изолировать функциональность в строго заданных рамках.

## 1.1.2. Шаблон проектирования MVVM

Когда JavaScript-фреймворки начали поддерживать асинхронные методы программирования, исчезла необходимость в обновлении всей веб-страницы при каждом запросе. Частичное обновление представления позволяет сайтам и приложениям быстрее реагировать, но в определенной степени приводит к дублированию кода. Логика представления часто повторяет бизнес-логику.

Шаблон MVVM (Model – View – ViewModel – «модель – представление – модель представления») – это следующий этап в развитии MVC. Его отличительными чертами являются *модель представления* (ViewModel, VM) и связывание данных. MVVM обеспечивает основу для построения клиентских приложений, которые делают взаимодействие с пользователем и обратную связь более отзывчивыми, избегая при этом затратного дублирования кода в рамках всей архитектуры. Этот шаблон упрощает также модульное тестирование. Но имейте в виду, что для простых пользовательских интерфейсов он может оказаться избыточным.

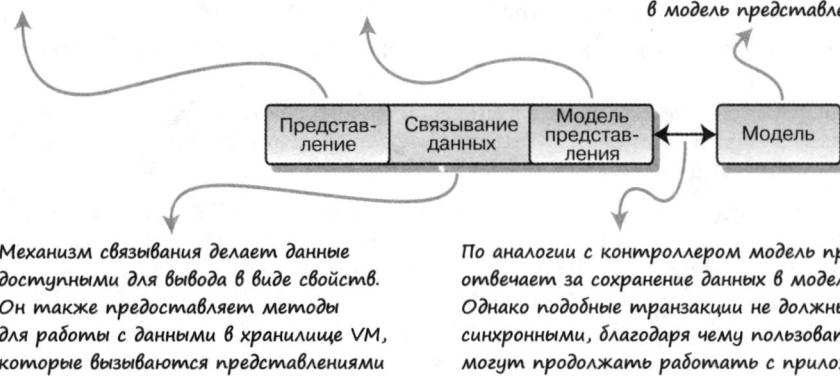
MVVM позволяет создавать веб-приложения, которые мгновенно реагируют на действия пользователя и позволяют свободно переходить от одной задачи к другой. Модель представления тоже способна играть несколько ролей (рис. 1.2). Такая консолидация ответственности имеет одно фундаментальное последствие для представлений приложения: при изменении данных внутри VM автоматически обновляются все представления, связанные с этой моделью. Механизм связывания

делает данные доступными и гарантирует, что любые их изменения будут отражены в представлении.

Представление по-прежнему отвечает за то, что происходит на экране, но вся логика принятия решений переместилась в модель представления. Само представление лишь выводит содержимое в зависимости от наличия и количества данных в текущем состоянии приложения

Модель представления хранит данные приложения в объекте, известном как хранилище. В хранилище находятся все данные, которые могут потребоваться приложению в любой момент времени (то есть состояние приложения)

Модель по-прежнему играет роль постоянного репозитория для данных приложения. Некоторые архитектуры в мире JavaScript используют модель исключительно в качестве хранилища, не налагая никаких ограничений на входящие данные; при этом вся логика принятия решений выносится в модель представления



**Рис. 1.2.** Элементы шаблона проектирования MVVM

## ДОПОЛНИТЕЛЬНО

Больше информации об MVVM можно найти в статье Мартина Фаулера (Martin Fowler), посвященной шаблону проектирования Presentation Model: [martinfowler.com/eaaDev/PresentationModel.html](http://martinfowler.com/eaaDev/PresentationModel.html).

### 1.1.3. Что такое реактивное приложение

Сама по себе парадигма реактивного программирования не нова. Однако в веб-приложениях ее начали использовать относительно недавно, в основном благодаря наличию фреймворков Vue, React и Angular.

В Интернете много информации о теории реактивности, но нас интересует более узкая тема. Чтобы веб-приложение можно было назвать реактивным, оно должно выполнять следующие функции:

- отслеживать изменения в своем состоянии;
- распространять уведомления об изменениях для всех своих компонентов;
- автоматически отрисовывать представления в ответ на изменения состояния;
- своевременно реагировать на действия пользователя.

Для достижения этих целей реактивные веб-приложения по возможности применяют принципы проектирования MVVM, асинхронные методики, позволяющие избежать блокирований взаимодействия с пользователем, и элементы функционального программирования.

Шаблон проектирования MVVM и реактивное программирование не имеют прямого отношения друг к другу, но у них общая направленность: обе концепции стремятся сделать приложение более отзывчивым и надежным с точки зрения пользователя.

## ДОПОЛНИТЕЛЬНО

Если хотите узнать больше о реактивной парадигме программирования Vue, почитайте руководство «Подробно о реактивности» по адресу [ru.vuejs.org/v2/guide/reactivity.html](http://ru.vuejs.org/v2/guide/reactivity.html).

### 1.1.4. Калькулятор на JavaScript

Чтобы разобраться в понятиях реактивности и связывания данных, напишем калькулятор на чистом JavaScript (листинг 1.1).

**Листинг 1.1.** Калькулятор на JavaScript: chapter-01/calculator.html

```
<!DOCTYPE>
<html>
  <head>
    <title>A JavaScript Calculator</title>
    <style>
      p, input { font-family: monospace; }
      p, { white-space: pre; }
    </style>
  </head>
  <body>
    <div id="myCalc">
      <p>x <input class="calc-x-input" value="0"/></p>
      <p>y <input class="calc-y-input" value="0"/></p>
      <p>-----</p>
      <p>= <span class="calc-result"></span></p>
    </div>
    <script type="text/javascript">
      (function(){
        function Calc(xInput, yInput, output) {
          this.xInput = xInput;
          this.yInput = yInput;
          this.output = output;
        }
        Calc.xName = 'xInput';
        Calc.yName = 'yInput';
        Calc.resultName = 'calc-result';
        Calc.prototype.sum = function() {
          this.output.value = +this.xInput.value + +this.yInput.value;
        };
        Calc.prototype.subtract = function() {
          this.output.value = +this.xInput.value - +this.yInput.value;
        };
        Calc.prototype.multiply = function() {
          this.output.value = +this.xInput.value * +this.yInput.value;
        };
        Calc.prototype.divide = function() {
          this.output.value = +this.xInput.value / +this.yInput.value;
        };
      })();
    </script>
  </body>
</html>
```

Поля для ввода значений x и y, привязанных к функции runCalc

Вывод введенных значений x и y

Конструктор для создания экземпляра Calc