

1

Начало работы с JavaScript

Итак, вы решили начать изучать JavaScript — отличный выбор! JavaScript — язык программирования, который может применяться и в серверной, и в клиентской части приложения. Серверная часть приложения — это внутренняя логика, которая обычно выполняется на компьютерах в центрах обработки данных и взаимодействует с базой данных, в то время как клиентская часть запускается на устройстве пользователя, часто с использованием браузера для JavaScript.

Вполне вероятно, что вы уже пользовались функциями, написанными на JavaScript, — особенно если работали в таких браузерах, как Chrome, Firefox, Safari или Edge. JavaScript чрезвычайно распространен. После открытия страницы вам предлагается принять cookie-файлы. Как только вы нажимаете ОК, всплывающее окно исчезает — это результат работы JavaScript. Когда вы перемещаетесь по разделам сайта и открываете подменю, также работает JavaScript. Или когда вы фильтруете товары в каталоге интернет-магазина. А как насчет чатов, которые выскакивают в считанные секунды пребывания на сайте? Что ж, вы правы — это также JavaScript!

Практически любое взаимодействие, которое мы осуществляем с веб-страницами, происходит благодаря JavaScript. Кнопки, которые вы нажимаете, поздравительные открытки, которые создаете, вычисления, которые проводите. Все, чему мало статичной страницы, требует JavaScript.

В этой главе мы затронем следующие темы.

- Почему надо изучать JavaScript.
- Как настроить среду разработки.
- Как браузер понимает JavaScript.
- Как использовать консоль браузера.
- Как добавить JavaScript на веб-страницу.
- Как написать код JavaScript.



Решения упражнений и проектов, а также ответы на вопросы для самопроверки находятся в приложении.

Почему надо знать JavaScript

Существует множество аргументов в пользу изучения данного языка. JavaScript (берущий начало в 1995 году) считается одним из наиболее распространенных языков программирования. Причина этому — JavaScript поддерживается и распознается браузерами. Если на вашем компьютере установлены браузер и текстовый редактор — вы уже владеете всем необходимым для работы с JavaScript. Конечно, есть и более продвинутые инструменты — мы рассмотрим их позже в текущей главе.

JavaScript — отличный язык программирования для новичков. Тем не менее самые продвинутые разработчики программного обеспечения также должны его знать хотя бы немного, ведь однажды они неизбежно столкнутся с ним. JavaScript — ваш прекрасный выбор по многим причинам. Прежде всего, вы сможете создавать действительно классные приложения с его помощью быстрее, чем можете себе представить. К тому моменту, как вы доберетесь до главы 5, вам уже будут по зубам довольно сложные скрипты для взаимодействия с пользователями. А к концу книги вы спокойно будете писать динамические веб-страницы для всевозможных задач.

JavaScript используется для разработки приложений и скриптов множества типов. На нем можно писать код для браузеров, а также программировать логический, невидимый нам, слой кода в приложении (такой как связь с базой данных). JavaScript применим в играх, сценариях автоматизации и других продуктах. Он используется в различных стилях программирования — способах структурирования и написания программного кода. Как вы его примените, зависит от назначения разрабатываемого скрипта. JavaScript полезен в различных парадигмах программирования: объектно-ориентированного, функционального и процедурного (если вы никогда раньше не занимались разработкой, то можете не до конца понять перечисленные концепции, но на данном этапе в этом нет особой необходимости).

Обладая основами JavaScript, можно уверенно использовать огромное число библиотек и фреймворков. Они поднимут ваши навыки на новый уровень, облегчат работу и помогут делать больше за меньшее время. Примеры распространенных и эффективных библиотек: React, Vue.js, jQuery, Angular и Node.js (не переживайте, если для вас это пока просто названия: мы кратко рассмотрим их в самом конце книги).

Как мы уже говорили, JavaScript — крайне популярный язык программирования, поэтому в процессе его изучения новичок не столкнется с такими проблемами, для которых не найдется готового решения в интернете. Сообщество JavaScript огромно. Известный форум Stack Overflow хранит множество решений по всем видам проблем кода, а также огромный раздел о JavaScript. Наверняка вы часто будете заглядывать на эту страницу в поисках советов и рекомендаций.

Если JavaScript — ваш первый язык программирования, то вас ждет много приятного. Разработчики программного обеспечения в целом любят помогать. В интернете есть форумы и учебные пособия с ответами практически на любые вопросы. И хоть вам, как начинающему, может быть непросто их понять, продолжайте учиться — и вскоре все станет ясно.

Настройка среды разработки

Есть множество способов настройки среды программирования JavaScript. Скорее всего, ваш компьютер уже оснащен минимальным набором необходимого для работы с JavaScript. Рекомендуем немного упростить свою жизнь и использовать встроенную среду разработки (IDE).

Встроенная среда разработки

Встроенная среда разработки (IDE) — это специальное приложение для написания, запуска и отладки кода. Вы можете открыть его, как любую другую программу. Например, чтобы создать текстовый документ, необходимо запустить редактор, выбрать файл и начать писать. Так же и в программировании: вы открываете IDE и пишете код. Если нужно запустить код, просто нажмите соответствующую кнопку в IDE. Для JavaScript может потребоваться открыть браузер вручную.

Однако IDE способна на большее. Обычно в ней есть функция подсветки синтаксиса: определенные элементы кода выделяются определенным цветом, что позволяет быстрее находить ошибки. Другая полезная особенность — это автодополнение кода: редактор сам предлагает варианты заполнения, доступные в текущем месте кода. Большинство IDE содержат специальные плагины, которые помогают сделать работу интуитивно понятной и добавить дополнительные функции, например горячую перезагрузку (hot reload) в браузере.

Существует множество IDE, и все они отличаются наборами предлагаемых функций. В книге мы используем Visual Studio Code, но это личное предпочтение. Для своей работы вы можете взять и другие IDE: Atom, Sublime Text и WebStorm. Среды разработки продолжают появляться, поэтому, скорее всего, самой популярной на момент прочтения данной книги в этом списке нет. С актуальными предложениями вы можете познакомиться, запустив быстрый поиск в интернете по запросу `JavaScript IDE`. При выборе IDE прежде всего уделите внимание следующим моментам: убедитесь, что среда поддерживает подсветку синтаксиса, отладку и автодополнение кода JavaScript.

Браузер

Вам также понадобится браузер. Большинство браузеров отлично подходят для задач JavaScript, но лучше не использовать Internet Explorer, который не поддерживает его обновленные функции. Два хороших варианта: Chrome и Firefox. Они прекрасно работают с актуальным функционалом JavaScript, к тому же полагают полезными плагинами.

Дополнительные инструменты

В своей работе не проходите мимо дополнительных возможностей, используемых при программировании, — в том числе плагинов браузеров, которые помогут вам с отладкой или упростят просмотр страницы. На начальном этапе вам все это не понадобится; тем не менее уже сейчас берите на заметку инструменты, которые ценят другие разработчики.

Онлайн-редактор

Это хорошее решение в случае, если у вас нет компьютера (возможно, только планшет) или вы не можете ничего на него устанавливать. В частности, для таких ситуаций есть отличные онлайн-редакторы. Мы не даем конкретных названий: онлайн-редакторы быстро развиваются, и предложенный нами список, вероятно, устареет к моменту выхода книги. Сделайте в интернете запрос **online JavaScript IDE** — система выдаст вам множество результатов, где вы сможете начать программировать на JavaScript простым нажатием кнопки.

Как браузер понимает JavaScript

JavaScript является интерпретируемым языком программирования: это значит, компьютер распознает его в процессе работы кода. Некоторые языки перед запуском кода требуют обработки (данной процесс называется компиляцией) — но для JavaScript в этом нет необходимости. Компьютер интерпретирует JavaScript на лету. При этом «движок», понимающий JavaScript, называется интерпретатором.

Веб-страница — это не только JavaScript. Она создается с помощью трех языков: HTML, CSS и JavaScript.

HTML определяет то, что отображается на странице: ее содержимое хранится в нем. Если на странице есть абзац, в HTML-коде это будет прописано. Видим

заголовок — значит, в HTML он тоже есть. И так далее. HTML состоит из элементов, которые называются тегами. Они описывают конкретный компонент страницы. Вот небольшой пример кода страницы с текстом `Hello world!`:

```
<html>
  <body>
    Hello world!
  </body>
</html>
```

Не переживайте, если не работали с HTML ранее: в главе 9 есть ускоренный курс этого языка.

CSS — это шаблон веб-страницы. Размер шрифта, семейство шрифтов и расположение текста на странице задаются с его помощью. Так, например, синий цвет текста — результат работы CSS.

JavaScript — последний кусочек пазла: он определяет, что может делать страница и как она будет взаимодействовать с пользователем или серверной частью.

Работая с JavaScript, вы рано или поздно столкнетесь с термином *ECMAScript*. Это спецификация, или стандартизация, для языка JavaScript. Текущий стандарт — *ECMAScript 6* (также называемый *ES6*). Браузеры используют его для поддержки JavaScript (в дополнение к таким возможностям, как *объектная модель документа (DOM)*, — ее мы рассмотрим позже). Существует множество незначительно различающихся реализаций JavaScript, и ECMAScript можно считать базовой спецификацией, про которую с уверенностью можно сказать, что в конкретную реализацию она будет включена.

Использование консоли браузера

У браузера есть встроенные возможности для просмотра кода на текущей странице — возможно, вы уже имели с ними дело. Если, находясь в браузере, нажать клавишу F12 на компьютере с Windows или щелкнуть правой кнопкой мыши на пункте *Inspect* (Проверка) в системе macOS, экран приобретет вид, представленный на одном из следующих снимков экрана.

На вашем компьютере итог может несколько отличаться, но, как правило, на macOS результат нажатия *Inspect* выглядит так, как показано на рис. 1.1.

На снимке экрана сверху видно несколько вкладок. Рассмотрим элементы, которые содержат в себе HTML и CSS (помните такие?). Если вы щелкнете на вкладке

консоли, в нижней части панели найдется место, где можно непосредственно ввести код. Там могут появляться различные предупреждения и сообщения об ошибках — если страница в целом работает, не беспокойтесь о них: подобные уведомления не редкость.

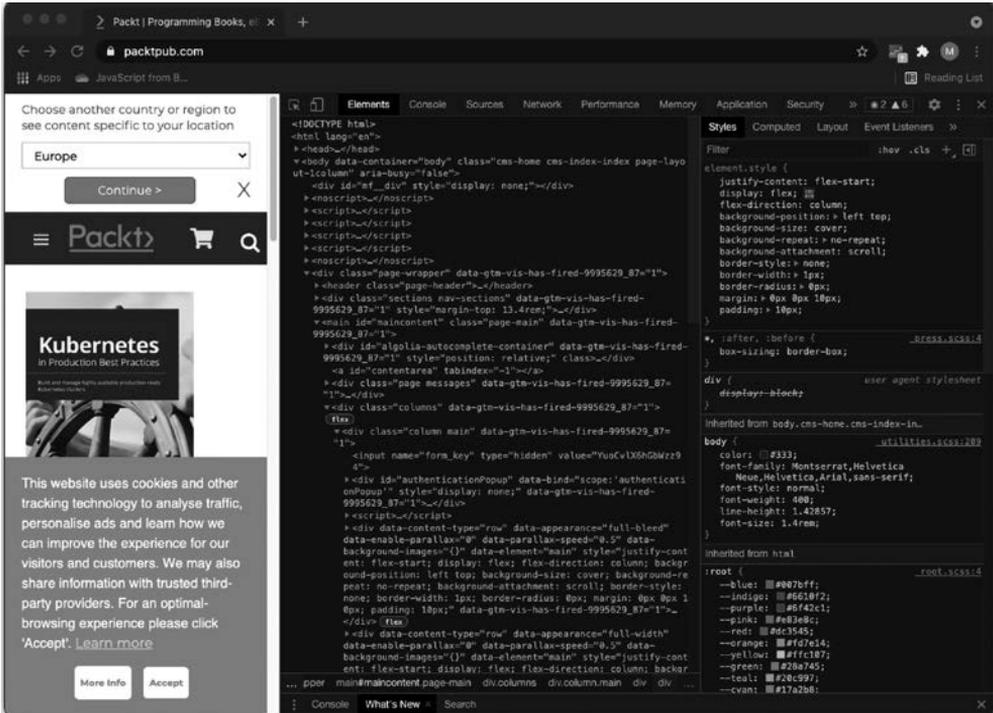


Рис. 1.1. Консоль браузера на сайте Packt

Разработчики используют консоль для логирования происходящего и отладки. Отладка — это поиск проблем в случаях, когда приложение не отображает желаемый результат. Консоль дает некоторое представление о том, что происходит, когда вы вводите осмысленные сообщения. Итак, первая команда, которую мы выучим:

```
console.log("Hello world!");
```

Щелкните на вкладке консоли и введите этот код. После нажатия Enter на экране отобразится вывод вашего кода. Он должен выглядеть как на рис. 1.2.

В книге мы будем часто использовать `console.log()`, чтобы проверить фрагменты кода и просматривать результаты. Существуют и другие консольные

методы. Например, `console.table()` представляет введенные данные в виде таблицы. Еще один консольный метод, `console.error()`, выводит сообщение об ошибке.

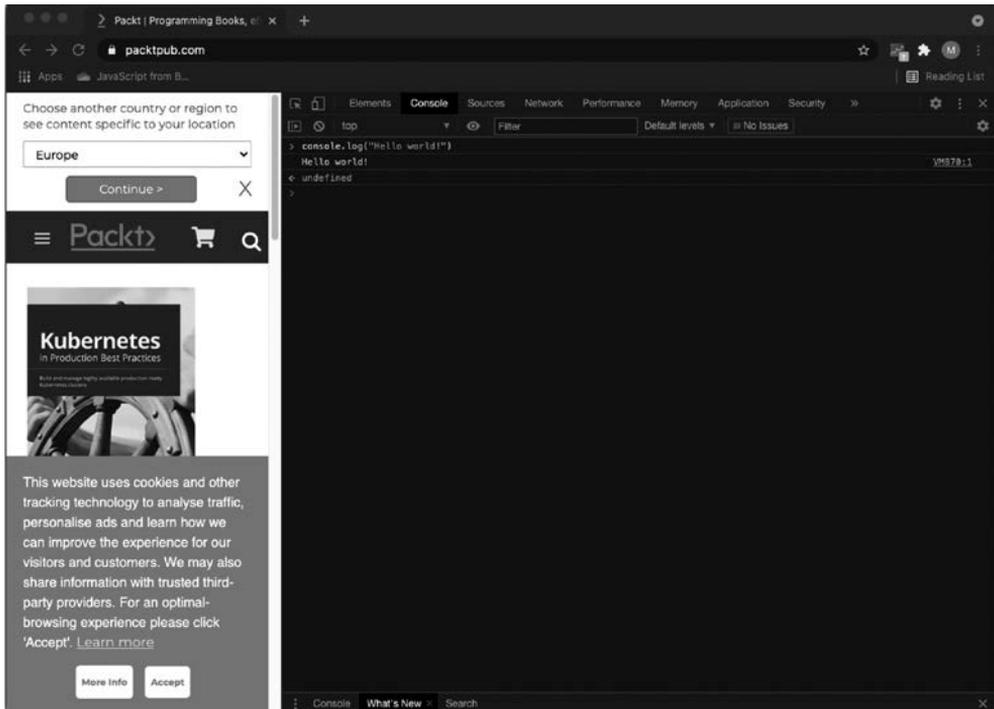


Рис. 1.2. JavaScript в консоли браузера

Практическое занятие 1.1

Работа с консолью

1. Откройте консоль браузера, напишите `4 + 10` и нажмите `Enter`. Что было представлено в качестве ответа?
2. Введите команду `console.log()`, поместив в круглые скобки какое-нибудь значение. Попробуйте вписать свое имя в кавычках (кавычки нужны для указания факта, что перед нами текстовая строка; мы рассмотрим это подробнее в следующей главе).

Добавление JavaScript на веб-страницу

Существует два способа добавления JavaScript-кода на веб-страницу. Один из них — ввести код прямо в HTML между парными тегами `<script>`. Первый `<script>` объявляет, что в рамках скрипта нужно выполнить следующий код. Далее располагается содержимое скрипта. Затем мы закрываем скрипт тем же тегом `<script>`, только в начале текста между скобками добавляем слеш — `</script>`. Второй способ — подключить файл JavaScript к HTML-файлу, используя соответствующий тег в заголовке страницы HTML.

Непосредственно в HTML

Перед вами пример простой веб-страницы, на которой всплывает окно с надписью `Hi there!`:

```
<html>
  <script type="text/javascript">
    alert("Hi there!");
  </script>
</html>
```

Если сохранить этот код как файл с расширением `.html` и открыть его в браузере, результат будет похожим на представленный на рис. 1.3. Сохраним файл как `Hi.html`.

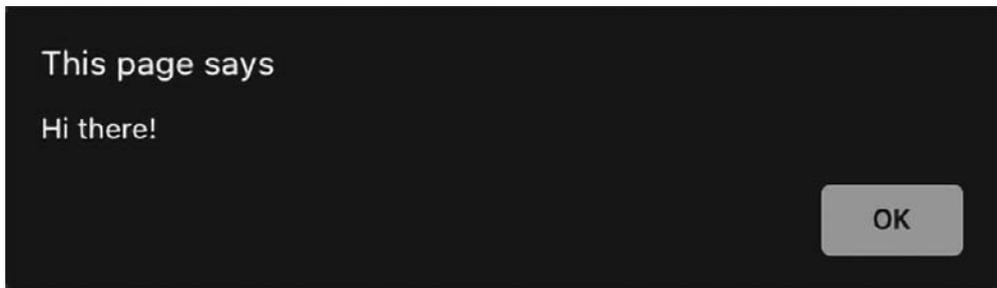


Рис. 1.3. Код JavaScript реализовал всплывающее окно с текстом `Hi there!`

Команда `alert` создает всплывающее окно с сообщением. Текст сообщения оформляется в скобках.

Сейчас наш контент располагается непосредственно между тегами `<html>`. Это не лучшее решение. Нужно создать еще два элемента — `<head>` и `<body>`. В `head` мы укажем метаданные; также в данном теге внешние файлы подключаются к HTML-файлу. В `body` поместится содержимое веб-страницы.

Кроме этого, необходимо сообщить браузеру, с каким типом документа мы работаем, объявив `<!DOCTYPE>`. Поскольку мы начали писать код JavaScript в HTML, наполнение тега будет следующим: `<!DOCTYPE html>`. Пример:

```
<!DOCTYPE html>
<html>

<head>
  <title>This goes in the tab of your browser</title>
</head>

<body>
The content of the webpage
  <script>
    console.log("Hi there!");
  </script>
</body>
</html>
```

В результате веб-страница должна отобразить следующее: `The content of the webpage`. Если вы посмотрите в консоль браузера, то обнаружите там сюрприз: исполненный код JavaScript и строку `Hi there!`.

Практическое занятие 1.2

JavaScript на HTML-странице

1. Откройте редактор кода и создайте HTML-файл.
2. Пропишите HTML-теги `doctype`, `html`, `head` и `body` и добавьте скрипт.
3. Между тегами скрипта напишите какой-нибудь JavaScript-код. Можете использовать `console.log("hello world!")`.

Присоединение стороннего файла к веб-странице

Еще один способ добавить JavaScript-код на страницу — подключить к файлу HTML сторонний файл. Данный вариант считается лучшим, поскольку код страницы в целом будет лучше организован и не будет слишком длинным. В дополнение к этим преимуществам вы сможете использовать JavaScript при создании других страниц на сайте, не копируя исходный код. Скажем, все десять страниц вашего сайта содержат один и тот же код JavaScript, в который вам нужно внести изменения. Если вы настроите страницу так, как показано в примере ниже, вам придется изменить только один файл.