



ВВЕДЕНИЕ

Эта книга рассказывает о том, как в браузере может использоваться язык JavaScript. Код JavaScript помогает делать сайты более интерактивными, интересными и удобными для пользователя. Кроме того, вы познакомитесь с библиотекой jQuery, которая значительно упрощает программирование на JavaScript.

Чтобы изучить эту книгу максимально эффективно, читатель должен уметь создавать веб-страницы и верстать их с использованием языков HTML и CSS. Никакого другого дополнительного опыта программирования не требуется. Чтобы научиться программировать на JavaScript, вам понадобится следующее.

1

Понимать некоторые базовые концепции программирования, а также термины, используемые в языке JavaScript.

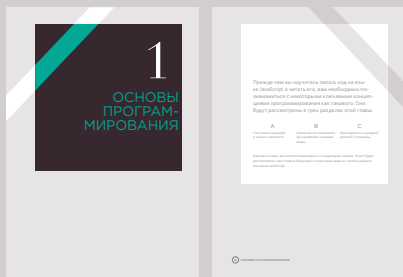
2

Изучить *сам язык*. Как и в случае с любым другим языком, вы должны будете освоить «словарь» JavaScript и научиться строить «фразы».

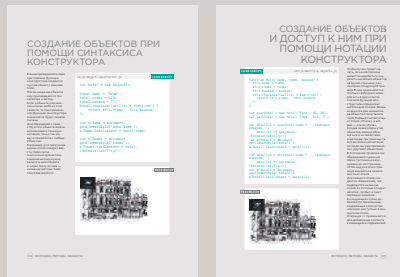
3

Освоить *применение* языка JavaScript. Для этого в книге приведено множество примеров, демонстрирующих, как сегодня используется JavaScript.

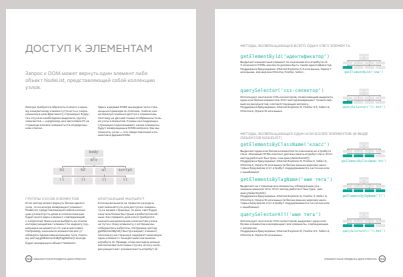
Для работы с книгой вам понадобится только компьютер, на котором установлен современный веб-браузер, а также ваш любимый текстовый редактор для написания кода — например, Блокнот (Notepad), TextEdit, Sublime Text или Coda.



Каждая глава начинается с **введения**, в котором перечисляются все ее основные темы



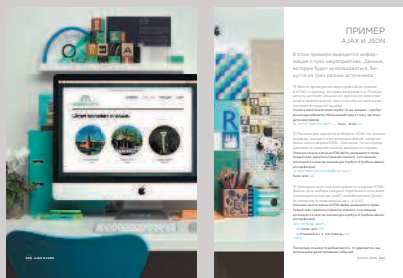
На **справочных** страницах представлены основные элементы JavaScript. HTML-код выделен синим цветом, CSS-код — розовым, JavaScript-код — зеленым



Страницы с **основной** информацией имеют белый фон и также содержат расширенные сведения по теме, обсуждаемой в главе



Диаграммы и инфографика представлены на страницах с темным фоном — это простой визуальный источник справочной информации по обсуждаемой теме



Разделы с **примерами** объединяют изученные темы и демонстрируют практическое применение изложенной выше информации



В конце каждой главы приведен **обзор**, где напоминает об основных моментах рассмотренных тем

КАК JAVASCRIPT ПОМОГАЕТ УЛУЧШИТЬ ИНТЕРАКТИВНОСТЬ ВЕБ-СТРАНИЦ

1

ДОСТУП К КОНТЕНТУ

Язык JavaScript позволяет выделить на HTML-странице любой элемент, атрибут или текст, например:

- выделить текст во всех элементах h1, расположенных на странице;
- выделить все элементы, имеющие атрибут class со значением note;
- узнать, какая информация была введена в текстовое поле, чей атрибут id имеет значение email.

Язык

JavaScript

позволяет придать веб-страницам дополнительную интерактивность, так как предоставляет доступ к их контенту и возможность изменять его, а также разметку на веб-странице, когда она открыта в браузере.

2

ИЗМЕНЕНИЕ КОНТЕНТА

Язык JavaScript можно использовать для добавления на страницу (или для удаления с нее) элементов, атрибутов и текста, например:

- добавить абзац текста после первого элемента h1;
- изменить значение атрибутов class, чтобы вступили в силу новые правила CSS, которые коснутся элементов с этими атрибутами;
- изменить размер или положение элемента img.

3

ПРОГРАММИРОВАНИЕ ПРАВИЛ

Язык JavaScript позволяет указать последовательность операций, которые должен выполнить браузер (это похоже на кулинарный рецепт). Данная последовательность обеспечивает доступ к контенту страницы либо дает возможность изменять его, например:

- сценарий галереи позволяет узнать, по какому изображению пользователь щелкнул мышью, а затем показать увеличенный вариант этого изображения;
- ипотечный калькулятор позволяет собрать значения из формы, выполнить расчеты и отобразить величины платежей;
- анимация позволяет проверить размеры окна браузера и переместить изображение в нижнюю часть области просмотра (также употребляется термин «окно просмотра»).

В JavaScript

действуют многие

общеизвестные правила

программирования.

Этот язык помогает сделать

страницу интерактивной,

реагирующей на действия

пользователя.

4

РЕАГИРОВАНИЕ НА СОБЫТИЯ

Язык JavaScript дает возможность создать сценарий, запускающийся после конкретного события. Вот несколько примеров таких событий:

- пользователь нажал кнопку;
- пользователь щелкнул мышью по ссылке (или коснулся ее на сенсорном экране);
- указатель мыши был наведен на какой-либо элемент;
- пользователь ввел данные в форму;
- прошел заранее установленный временной интервал;
- завершилась загрузка веб-страницы.

ПРИМЕРЫ JAVASCRIPT-КОДА В БРАУЗЕРЕ

Возможность изменять контент уже загруженной в браузере HTML-страницы дает массу преимуществ. Приведенные ниже примеры основываются на том, что разработчик через программный код может:

- **получать** доступ к контенту страницы;
- **изменять** контент страницы;
- **задавать (программировать)** правила или инструкции, которым должен следовать браузер;
- **реагировать** на события, инициируемые пользователем или браузером.



СЛАЙД-ШОУ

(см. главу 11)

Слайд-шоу позволяют последовательно отображать некоторое количество изображений (или другого HTML-контента) в одной области конкретной веб-страницы. Слайд-шоу либо проигрывается автоматически, либо предоставляет пользователю возможность самостоятельно выбирать и просматривать интересующие его изображения. Слайд-шоу позволяет увеличивать объем контента, отображаемого в ограниченном пространстве.

Реагирование: на сценарии, срабатывающие при загрузке страницы.

Доступ разработчика: к каждому изображению слайд-шоу.

Изменение: можно показывать лишь первый слайд, а остальные скрывать.

Программирование: установка таймера, определяющего, когда следует показывать очередной слайд.

Реагирование: на нажатие пользователем кнопки, соответствующей другому слайду.

Программирование: определение, какой слайд показывать.

Изменение: демонстрация запрошенного слайда.



ФОРМЫ

(см. главу 13)

Валидация форм (проверка того, правильно ли они заполнены) важна в случаях, когда на сайте принимается информация, вводимая пользователями. JavaScript-код позволяет предупредить пользователя об ошибках в заполнении формы, если они возникнут. Кроме того, JavaScript позволяет выполнять довольно сложные вычисления с использованием введенных данных, и показывать полученные результаты пользователю.

Реагирование: на нажатие пользователем кнопки отправки формы по завершении ввода.

Доступ разработчика: к введенному в поле формы значению.

Программирование: проверка того, достаточно ли длинным является значение.

Изменение: показ предупреждающего сообщения, в случае если введенное значение недостаточно длинное.

Вышеприведенные примеры помогают составить впечатление о том, что можно сделать на веб-странице при помощи JavaScript-кода, а также о приемах, приведенных на страницах этой книги.



ПЕРЕЗАГРУЗКА ФРАГМЕНТА СТРАНИЦЫ

(см. главу 8)

Возможно, вы не хотите, чтобы пользователи перезагружали всю веб-страницу целиком, особенно когда требуется обновить лишь небольшую ее часть. Перезагрузить один конкретный фрагмент страницы куда быстрее, а сайт тем самым функционально станет похож на веб-приложение.

Реагирование: на сценарий, срабатывающий, когда пользователь щелкает по ссылке.

Доступ разработчика: к ссылке, по которой щелкнул пользователь.

Программирование: загрузка нового контента, запрошенного по ссылке.

Доступ разработчика: нахождение на странице элемента, который необходимо заменить.

Изменение: замена имеющегося контента на новый.

В следующих главах вы узнаете, когда и как можно получать доступ к контенту и изменять его, как программными средствами задавать правила и реагировать на события.



ФИЛЬТРАЦИЯ ДАННЫХ

(см. главу 12)

Если на странице содержится сравнительно много информации, то можно помочь пользователю найти именно те данные, которые его интересуют. Для этого применяются фильтры. В нашем случае кнопки генерируются при помощи данных, содержащихся в атрибутах HTML-элементов `img`. Когда пользователь щелкает мышью по кнопкам, он получает лишь те изображения, которые снабжены нужным ключевым словом.

Реагирование: на сценарий, срабатывающий при загрузке страницы.

Программирование: собирание ключевых слов из атрибутов изображений.

Программирование: преобразование ключевых слов в кнопки, на которые может нажимать пользователь.

Реагирование: на нажатие пользователем одной из кнопок

Программирование: нахождение подмножества изображений, которые следует показать.

Изменение: демонстрация подборки изображений, использующих определенный тег.

СТРУКТУРА КНИГИ

Чтобы помочь вам изучить JavaScript, мы разделили книгу на две части.

КОНЦЕПЦИИ

В первых девяти главах вы познакомитесь с основами программирования на JavaScript. Вам предстоит узнать, как этот язык позволяет создавать более увлекательные, интерактивные и удобные сайты.

Глава 1 посвящена некоторым ключевым концепциям компьютерного программирования. Вы узнаете, как компьютер оперирует данными и создает на их основе модель мира, как язык JavaScript используется для изменения контента HTML-страниц.

Главы 2–4 рассказывают об основах языка JavaScript.

В **главе 5** объясняется, как объектная модель документа (DOM) позволяет получать доступ к контенту документа и изменять его, когда страница уже загружена в браузере.

В **главе 6** мы поговорим о том, как можно инициировать выполнение того или иного кода при помощи событий.

Глава 7 рассказывает о библиотеке jQuery, которая помогает упростить и ускорить написание сценариев.

В **главе 8** мы познакомимся с Ajax — набором практических приемов, позволяющих изменять часть веб-страницы, не перезагружая ее целиком

В **главе 9** рассматриваются интерфейсы программирования приложений (API). Здесь мы затронем и сравнительно новые API, предназначенные для работы с HTML5 и такими сайтами, как Google Карты.

ПРАКТИКА

К началу второй части вы уже успеете изучить множество примеров использования JavaScript-сценариев на популярных сайтах. Здесь мы обобщим все приемы, освоенные вами ранее, и продемонстрируем, как профессионалы применяют язык JavaScript на практике. Мы познакомим вас не только с рядом продвинутых примеров, но и научим разрабатывать сценарии и писать их с нуля.

Глава 10 посвящена ошибкам и отладке. Здесь мы подробно обсудим, как компьютер обрабатывает JavaScript-код.

В **главе 11** рассмотрены способы создания контент-панелей, в частности, слайдеров, модальных окон, панелей с вкладками и аккордеонов.

В **главе 12** мы обсудим ряд способов фильтрации и сортировки данных. В частности, речь пойдет о фильтрации галереи изображений и о перегруппировке строк в таблице, производимой, когда пользователь щелкает по заголовкам столбцов.

Глава 13 посвящена различным улучшениям форм и валидации записей в формах.

Если у вас нет большого опыта в программировании, вам было бы полезно в первый раз прочитать книгу от начала и до конца. Однако мы надеемся, что, когда вы освоите основы программирования и станете писать собственные сценарии, эта книга останется для вас полезным справочником.

HTML И CSS: КРАТКИЙ ЭКСКУРС

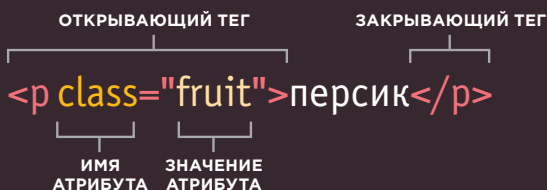
Прежде чем переходить к изучению языка JavaScript, давайте вспомним некоторые аспекты HTML и CSS и, в частности, поговорим о том, как атрибуты HTML и свойства CSS образуют пары «имя/значение».

HTML-ЭЛЕМЕНТЫ

Элементы языка HTML содержатся на веб-странице для разметки ее структуры. Элемент состоит из открывающего тега, закрывающего тега и содержимого (контента).

Как правило, теги используются попарно (открывающий плюс закрывающий). Некоторые элементы не имеют контента и поэтому называются «пустыми» (например, элемент `img`). У них есть только один самозакрывающийся тег.

Открывающие теги могут сопровождаться атрибутами. Атрибут подробнее характеризует данный элемент. Каждый атрибут имеет имя и значение, которое обычно заключается в кавычки.

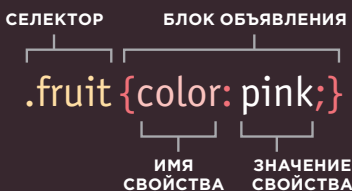


ПРАВИЛА CSS

В таблицах CSS используются правила, указывающие, как в браузере должен отображаться контент одного или нескольких элементов. В каждом правиле CSS есть селектор и блок объявлений.

Селектор CSS указывает, какой элемент (или элементы) подчиняется данному правилу. Блок объявлений содержит информацию о том, как должны выглядеть эти элементы.

У каждого объявления в блоке есть свойство (аспект, которым вы хотите управлять) и присваиваемое ему значение.



ПОДДЕРЖКА БРАУЗЕРАМИ

Некоторые примеры, приведенные в этой книге, не работают в браузерах Internet Explorer 8 и более ранних. В следующих главах мы коснемся того, как следует работать со старыми версиями браузеров.

В каждой версии веб-браузера появляются новые возможности. Зачастую они упрощают решение тех или иных задач либо считаются более желательными для употребления, чем старые приемы.

Однако далеко не все пользователи Интернета стремятся обновлять свои браузеры до самых последних версий, поэтому веб-разработчик не может опираться лишь на новейшие технологии.

Как будет показано ниже, поддержка тех или иных функций браузерами остается во многом несогласованной, и такие противоречия отражаются на JavaScript-разработке. Библиотека jQuery помогает нивелировать такие кроссбраузерные различия (в этом и заключается одна из основных причин, почему jQuery так быстро приобрела популярность среди веб-разработчиков). Впрочем, прежде чем знакомиться с jQuery, давайте разберемся, чего именно она помогает достичь.

Чтобы упростить изучение языка JavaScript, мы посвятим первые несколько глав таким возможностям этого языка, которые не поддерживаются в браузере Internet Explorer 8, и тем не менее:

- далее на страницах этой книги вы *научитесь* работать с Internet Explorer 8 и более старыми браузерами (поскольку нам известно, насколько распространены клиентские программы, требующие функционирования сайтов в Internet Explorer 8) — для этого вам просто нужно будет немного лучше разбираться в коде и учитывать ряд дополнительных проблем;
- среди файлов примеров вы найдете альтернативные варианты решения каждого примера, не работающего в Internet Explorer 8 в том виде, в котором оно приведено у нас в книге; пожалуйста, сообщайте нам о таких случаях — нам важно знать обо всех проблемах, возникших у читателей при изучении материала.

Глава 1

ОСНОВЫ
ПРОГРАММИРОВАНИЯ