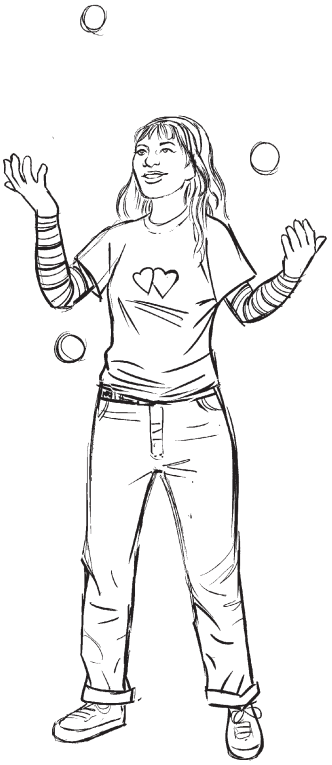
1 Первое знакомство с JavaScript

В незнакомых водах

JavaScript открывает фантастические возможности. JavaScript, **основной язык программирования** Всемирной паутины, позволяет **добавлять поведение** к веб-страницам. Забудьте о сухих, скучных, статичных страницах, которые просто занимают место на экране, — с JavaScript вы будете взаимодействовать с пользователями, реагировать на события, получать и использовать данные из интернета, выводить графику на страницах... и многое, многое другое. При хорошем знании JavaScript вы сможете дальше программировать **совершенно новое** поведение своих пользователей.

И вы будете в хорошей компании. JavaScript — не только один из **самых популярных** языков программирования, но он еще и **поддерживается** всеми современными браузерами; более того, появились встроенные реализации JavaScript, существующие отдельно от браузеров. А впрочем, хватит разговоров. Пора браться за дело!



2 Настоящий код

Следующий шаг

Вы уже знаете, что такое переменные, типы, выражения... и так далее. Вы уже кое-что знаете о JavaScript. Более того, этих знаний достаточно, чтобы начать писать настоящие программы, которые делают что-то интересное и которыми кто-то будет пользоваться. Правда, вам не хватает практического опыта написания кода, и мы прямо сейчас начнем решать эту проблему. Как? А просто возьмемся за написание несложной игры, полностью реализованной на JavaScript. Задача масштабная, но мы будем двигаться к цели постепенно, шаг за шагом. Итак, беремся за дело, а если вам вдруг захочется использовать нашу разработку в своих проектах — мы не против; распоряжайтесь кодом, как считаете нужным.

3 Знакомство с функциями

Функции для всех

В этой главе вы овладеете своей первой суперспособностью. Вы уже кое-что знаете о программировании; пришло время сделать следующий шаг и освоить работу с функциями. Функции позволяют писать код, который может повторно использоваться в разных ситуациях; код существенно более простой в сопровождении; код, который можно абстрагировать и присвоить ему простое имя, чтобы вы могли забыть о рутине и заняться действительно важными делами. Вы увидите, что функции не только открывают путь к мастерству программиста, но и играют ключевую роль в стиле программирования JavaScript. В этой главе мы начнем с основ: механики и всех тонкостей работы функций, а потом в оставшейся части книги будем совершенствовать ваши навыки работы с функциями. Итак, начнем с азов... прямо сейчас.

Наведение порядка в данных

4 Массивы

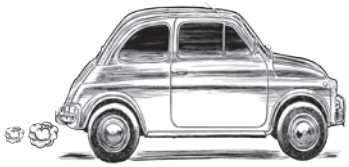
JavaScript может работать не только с числами, строками и логическими значениями. До настоящего момента мы работали исключительно с **примитивами** — простыми строками, числами и логическими значениями (например, «Fido», 23 и true). С примитивными типами можно сделать многое, но в какой-то момент возникнет необходимость в **расширенных данных** для представления всех позиций в корзине покупок, всех песен в плейлисте, группы звезд и их звездных величин или целого каталога продуктов. Подобные задачи требуют более серьезных средств. Типичным инструментом для представления таких однородных данных является **массив** JavaScript. В этой главе вы узнаете, как помещать данные в массив, передавать их и работать с ними. В последующих главах будут рассмотрены другие способы **структурирования данных**, но начнем мы с массивов.

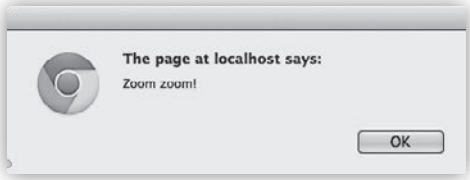
5

Знакомьтесь: объекты

Поездка в Объективиль

До настоящего момента мы использовали примитивы и массивы. И при этом применялась методология **процедурного программирования** с простыми командами, условиями, циклами for/while и функциями. Такой подход был далек от принципов **объектно-ориентированного программирования**. Собственно, он вообще не имел *ничего общего* с объектно-ориентированным программированием. Мы использовали объекты время от времени (причем вы об этом даже не знали), но еще не написали ни одного собственного объекта. Пришло время покинуть скучный процедурный город и заняться созданием собственных объектов. В этой главе вы узнаете, почему объекты сильно улучшают нашу жизнь — во всяком случае **в области программирования**. Так и знайте: привыкнув к объектам, вы уже не захотите возвращаться обратно. Да, и не забудьте прислать открытку, когда обживетесь.

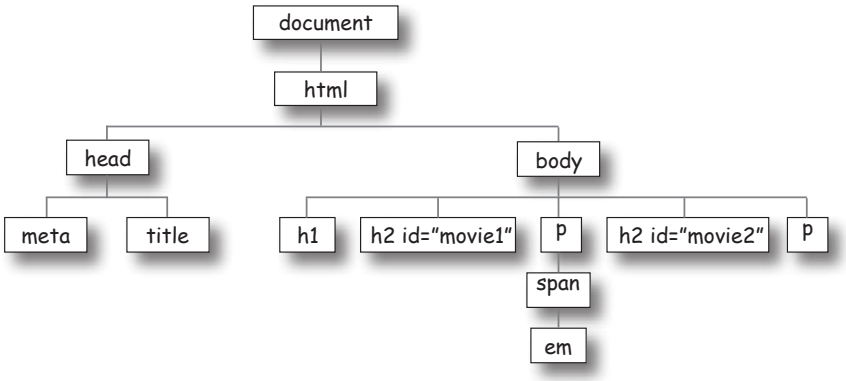




Взаимодействие с веб-страницей

6 Модель DOM

Вы значительно продвинулись в изучении JavaScript. Фактически вы из новичка в области сценарного программирования превратились в... **программиста**. Впрочем, кое-чего не хватает: для полноценного использования ваших навыков JavaScript необходимо уметь взаимодействовать с веб-страницей, в которой располагается ваш код. Только в этом случае вы сможете писать **динамические** страницы, которые реагируют на действия пользователя и обновляются после загрузки. Как взаимодействовать со страницей? Через объектную модель документа **DOM** (Document Object Model). В этой главе мы рассмотрим DOM и общие принципы работы с этой моделью из JavaScript для расширения возможностей страницы.



7 Типы, равенство, преобразования и все такое

Серьезные типы

Настало время серьезно поговорить о типах. Одна из замечательных особенностей JavaScript заключается в том, что начинающий может достаточно далеко продвинуться, не углубляясь в подробности языка. Но чтобы в полной мере **овладеть языком**, получить повышение по работе и заняться тем, чем действительно стоит заниматься, нужно хорошо разбираться в **типах**. Помните, что мы говорили о JavaScript в главе 1, — что у этого языка не было такой роскоши, как академическое определение, прошедшее экспертную оценку? Да, это правда, но отсутствие академической основы не помешало Стиву Джобсу и Биллу Гейтсу; не помешало оно и JavaScript. Это означает, что система типов JavaScript... ну, скажем так... не самая продуманная, и в ней найдется немало **странностей**. Но не беспокойтесь, в этой главе мы все разберем, и вскоре вы научитесь благополучно обходить все эти неприятные моменты с типами.

8 Всё вместе

Построение приложения

Подготовьте свой инструментарий к работе. Да, ваш инструментарий — ваши новые навыки программирования, ваше знание DOM и даже некоторое знание HTML и CSS. В этой главе мы объединим всё это для создания своего первого полноценного **веб-приложения**. Довольно примитивных игр с одним кораблем, который размещается в одной строке! В этой главе мы построим **полную версию**: большое игровое поле, несколько кораблей, ввод данных пользователем прямо на веб-странице. Мы создадим структуру страницы игры в разметке HTML, применим визуальное оформление средствами CSS и напишем код JavaScript, определяющий поведение игры. Приготовьтесь: в этой главе мы займемся полноценным, мощным программированием и напишем вполне серьезный код.

9

Обработка событий

Асинхронность

В этой главе вам предстоит подняться на принципиально новый уровень. До настоящего момента мы писали код, который обычно выполняется сверху вниз. Конечно, в нем использовались функции, объекты и методы, но выполнение шло по заранее намеченной колее. Жаль, что нам приходится сообщать такие новости во второй половине книги, но **такая структура кода нехарактерна для JavaScript**. Большая часть кода JavaScript пишется для **обработки событий**. Каких событий? Да любых. Пользователь щелкает на странице, данные поступают из сети, в браузере срабатывает таймер, в DOM происходят изменения... Это далеко не полный список. Более того, в браузере **постоянно** происходят события, которые в основном остаются незамеченными. В этой главе мы пересмотрим свой подход к программированию и узнаем, для чего же нужно писать код, реагирующий на события.

10

Анонимные функции и функции высшего порядка

Функции без ограничений

Изучайте функции и блистайте. В каждом ремесле, искусстве и дисциплине есть ключевой принцип, который отличает игроков «среднего звена» от настоящего профессионала, и когда речь заходит о JavaScript, признаком профессионализма является хорошее понимание **функций**. Функции играют фундаментальную роль в JavaScript, и многие приемы, применяемые при **проектировании и организации** кода, основаны на хорошем знании функций и умении использовать их. Путь изучения функций на этом уровне интересен и непрост, так что приготовьтесь... Эта глава немного напоминает экскурсию по шоколадной фабрике Вилли Вонка — во время изучения функций JavaScript вы увидите немало странного, безумного и замечательного.

11 Современный синтаксис, лексическая область видимости и замыкания

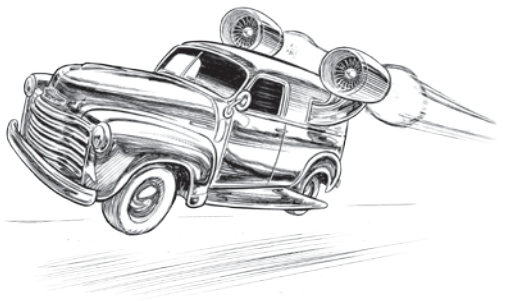
Серьезные функции

Мы узнали много нового о функциях, но это далеко не всё. В этой главе мы пойдем дальше и разберемся в темах, которыми обычно занимаются профессионалы. Вы расширите свое владение синтаксисом и научитесь действительно эффективно пользоваться аргументами, параметрами и присваиванием. Мы вернемся к областям видимости и некоторым особенностям их в JavaScript. Изучение нюансов областей видимости приведет нас к замыканиям — концепции, которая нередко считается чем-то мистическим, но является чрезвычайно важной для освоения JavaScript. В результате ваш код JavaScript станет более выразительным, чем вы могли себе представить.

12 Нетривиальное создание объектов

Создание объектов

До настоящего момента мы создавали объекты вручную. Для каждого объекта использовался **объектный литерал**, который задавал все без исключения свойства. Для небольших программ это допустимо, но для серьезного кода потребуется что-то получше. На помощь приходят **классы**. Классы упрощают создание объектов, причем вы можете создавать объекты по единому **шаблону,** — иначе говоря, классы гарантируют, что все объекты обладают одинаковыми свойствами и содержат одинаковые методы. Код, написанный с использованием классов, получается гораздо более **компактным** и снижает риск ошибок при создании большого количества объектов. Итак, за дело...



Приложение: остатки

Главные десять тем... (не рассмотренных в книге)

Мы прошли долгий путь, и книга почти завершена. Мы будем скучать, но прежде чем расстаться, хотим сказать еще несколько напутственных слов. В эту относительно маленькую главу не удастся включить все, что вам нужно знать. На самом деле мы *попытались* уместить в книгу все, что необходимо знать о программировании на JavaScript (что еще не было описано в других главах), сократив размер шрифта до .00004. Все поместилось, но никто не мог прочитать текст. В общем, мы отказались от этих попыток и оставили самое интересное для этого приложения.

И это *действительно* последний раздел книги.