

ГЛАВА 2

НАВИГАЦИЯ



Система навигации веб-сайта должна подстраиваться под контент любого размера

Довольно часто система навигации является основой для дизайна веб-сайта. Навигация — такой же важный компонент страницы, как и любой другой. Как правило, разработка системы навигации заключается в том, что веб-дизайнер любовно вырисовывает кнопки, вкладки или текст в графическом редакторе, а затем размещает их во множество вложенных `<table>`. Иногда для создания эффекта при наведении курсора на вкладки используется JavaScript, который заменяет один набор изображений на другой. Конечный результат может выглядеть просто великолепно. Однако то, что скрыто за внешней красотой, как правило, не так привлекательно. Более подробно мы изучим этот вопрос после того, как рассмотрим принятую практику создания систем навигации.

Давайте рассмотрим пример, который хорошо проиллюстрирует типичный подход к разработке графических вкладок. После того как мы разберемся в недостатках данного метода, мы сможем разработать аналогичный (на этот раз пуленепробиваемый!) дизайн, используя экономичную разметку, три небольших изображения и, конечно же, CSS.

ОБЩЕПРИНЯТЫЙ ПОДХОД

В качестве примера я решил использовать основные навигационные вкладки, которые нашел на сайте LanceArmstrong.com (рис. 2.1). Мой выбор обусловлен двумя причинами. Прежде всего, я фанат Лэнса Армстронга. Но, что более важно, дизайн сайта просто кричит, что ему нужна переделка с использованием CSS, которая обеспечила бы необходимую гибкость. Наша цель — использовать CSS с наименьшим количеством кода и создать простой для обслуживания и удобный для масштабирования сайт.

СОВЕТ

Дизайн сайта давно изменен. Поэтому если вы хотите увидеть, как выглядели оригинальные вкладки, просто перейдите по следующей ссылке: <http://web.archive.org/web/20041204124045/http://www.lancearmstrong.com/about.htm>.



Рис. 2.1. Навигационные вкладки со старой версии сайта LanceArmstrong.com в активном и неактивном состояниях

ПОДСВЕТКА

Дизайн вкладок обладает одной интересной особенностью помимо изменения фона и цвета рамки. В каждом из состояний вкладки (активном и неактивном) используется градиентная заливка по вертикали с переходом в сплошную: белую — в активном состоянии и светло-желтую — в неактивном. Верхняя рамка подведена линией более светлого тона, чем остальной абрис. Толщина линии — 1 пиксел. Она добавляет объем и создает эффект, как будто источник света расположен сверху страницы (рис. 2.2).

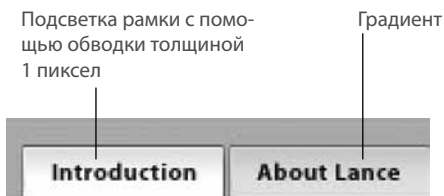


Рис. 2.2. Увеличенное изображение элементов навигации. Обратите внимание на небольшую подсветку рамки в верхней части каждой вкладки. Градиентная заливка используется для имитации объема и создания светотени

Прежде чем перейти к следующему пункту, хочу напомнить, что в целом вкладки реализованы удачно. Даже учитывая, что «хороший дизайн» довольно субъективное понятие (ведь у вас может быть совсем другое мнение на этот счет), все должны согласиться с тем, что дизайнер потратил много времени на разработку системы навигации LanceArmstrong.com, сделав ее привлекательной и функциональной. Цель была достигнута. Именно поэтому я выбрал этот сайт в качестве примера.

На LanceArmstrong.com каждая вкладка представляет собой графический объект. Картинки изображают активное или неактивное состояния, в зависимости от того, какую страницу вы просматриваете. Кроме того, используется графический логотип сайта. На рис. 2.3 показано одно из четырех возможных изображений навигационной системы.

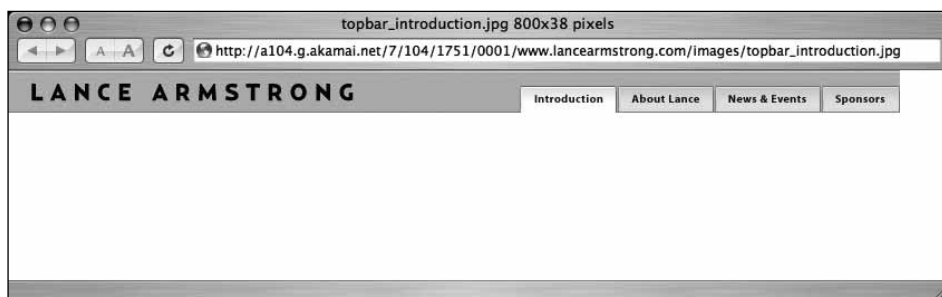


Рис. 2.3. Вся система вкладок и логотип образуют единое большое изображение. На сайте есть еще как минимум три аналогичных изображения с подсветкой других разделов

СТАНДАРТНЫЕ СПОСОБЫ СОЗДАНИЯ ЭФФЕКТОВ

В настоящий момент навигация на LanceArmstrong.com не предусматривает эффектов при наведении курсора. Их можно добавить с помощью JavaScript и второго комплекта изображений, но это потребует еще большего количества строк кода. Мы займемся эффектами для вкладок позднее, когда будем изменять дизайн в разделе «Пуленепробиваемый подход». Но в нашем случае понадобится лишь несколько строк CSS вместо объемного кода, дополнительных изображений и JavaScript.

Итак, мы видим навигационные вкладки, разработанные в соответствии с дизайном сайта. Нажатие на каждую из вкладок перемещает нас на один из главных разделов сайта. Каждой вкладке соответствует текстовая надпись. Теперь давайте рассмотрим систему более скрупулезно и научимся создавать аналогичный дизайн, но другим способом.

Уязвимые места

Что здесь не так? Я не собираюсь критиковать дизайнеров LanceArmstrong.com. На этом сайте используется такой же способ навигации, как и на миллионах других. Я просто рассматриваю его как пример, потому что он интересен с визуальной точки зрения и его довольно легко переделать с использованием CSS и простой разметки. Давайте перечислим особенности сайта, которые являются причиной его уязвимости.

Слишком много кода

Объемный код на JavaScript, использованный для размещения изображений и добавления динамики системе навигации (переключение изображений при наведении курсора), затрудняет работу с самим сайтом. Каждая вкладка представлена изображениями, которые размещаются в серии вложенных таблиц, GIF-разделителями (прозрачными изображениями, используемыми только для заполнения пустого места между элементами) и другими элементами разметки, необходимыми для создания точного макета. В результате генерируется слишком объемный код, который приводит к тому, что сайт становится тяжеловесным и, следовательно, медленно загружается.

Обратимся к LanceArmstrong.com: все четыре вкладки расположены на одном изображении. Для него определена карта ссылок, которая делает каждую вкладку активной. Если бы каждая вкладка была отдельным изображением, размещенным в таблице с GIF-разделителями, потребовалось бы больше кода. Но у каждого из этих способов имеются свои недостатки.

Недоступность сайта для некоторых категорий пользователей

Другим побочным эффектом большого количества кода является то, что он усложняет работу в текстовых браузерах и вспомогательных приложениях, которыми пользуются люди с ограниченными возможностями. Вся система навигации является одним изображением (см. рис. 2.3), но большинство дизайнеров забывают про атрибут `alt`, который необходимо прописывать для каждой зоны ссылки при использовании карты ссылок. Пользователи программ для чтения с экрана или пользователи, отключающие изображения в браузерах для более быстрой загрузки (при медленных соединениях), едва ли смогут комфортно работать с таким сайтом.

Проблема масштабирования

Это очень серьезная проблема. Так как вкладки представлены изображениями, слабовидящие пользователи не смогут изменять размер текста через браузер, чтобы облегчить чтение. Систему навигации, построенную на изображениях, невозможно масштабировать.

ПРИМЕЧАНИЕ

Исключением является функция Масштаб страницы, которую поддерживает большинство современных браузеров. Она изменяет размеры всего документа.

ОТСУТСТВИЕ ГИБКОСТИ

Если в будущем редакторы сайта LanceArmstrong.com захотят изменить текст на вкладке или, например, написать вместо «О Лэнсе» слова «Почему мы любим Лэнса», у них возникнут проблемы. Редактирование, удаление или замена текста на изображениях потребуют создания новой подборки картинок. Это, в свою очередь, приведет к необходимости изменения размерной сетки в карте ссылок. А ведь все это — немалый объем дополнительной работы.

Кроме того, как мы помним, система вкладок представлена одним изображением. Это означает, что для внесения одного изменения потребуется обновление четырех изображений (причем для каждого состояния вкладки) — это потенциальная головная боль для дизайнера.

Как видно, в системе еще многое можно улучшить. Каждый из минусов является достаточной причиной для того, чтобы найти альтернативный способ решения задачи.

ПУЛЕНЕПРОБИВАЕМЫЙ ПОДХОД

Давайте переделаем систему навигации с помощью экономичных методов CSS, учитывая недостатки общепринятой практики создания таких систем, но сохранив понравившийся дизайн. Нашей целью является устранение недостатков с точки зрения пуленепробиваемого подхода.

Как и с любым другим проектом, нужно сначала решить, как сделать разметку более гибкой и удобной. Имеет смысл структурировать список ссылок как обычный список. Используем новый семантический элемент `<nav>` из HTML5 для оформления списка:

```
<nav role="navigation">
  <ul>
    <li id="t-intro"><a href="/">Introduction</a></li>
    <li id="t-about"><a href="about.html">About Lance </a></li>
    <li id="t-news"><a href="news.html">News & Events</a></li>
    <li id="t-sponsors"><a href="sponsors.html">Sponsors</a></li>
  </ul>
</nav>
```

СОВЕТ

Для получения дополнительной информации о значении меток см. стандарт www.w3.org/WAI/PF/aria-practices/#kbd_layout (на английском языке).

Обратите внимание на то, что мы также используем атрибут `role` элемента `<nav>` для введения в систему навигации WAI-ARIA-метки. Роли метки обеспечивают навигацию для вспомогательных приложений. Иными словами, роли значительно повышают доступность сайта путем добавления дополнительной семантики.

ПРИМЕЧАНИЕ

WAI-ARIA (Web Accessibility Initiative—Accessible Rich Internet Applications) — стандарт доступности активных интернет-приложений, определяет подходы к тому, чтобы сделать содержимое сайтов и интернет-приложения более доступными для людей с ограниченными возможностями. Стандарт представлен проектом технической спецификации, опубликованным консорциумом W3C.

Простой маркированный список, который вкладывается в `<nav>`, — единственное, что нам понадобится для создания пуленепробиваемой таблицы в HTML5. Элементарная разметка гарантирует работу навигационной системы во всех браузерах, устройствах и даже вспомогательных приложениях. И это уже не говоря о том, что страница станет существенно «легче». Сравните количество кода, использованного в нашем случае, с общепринятым подходом на базе вложенных таблиц. Разница очевидна.

Обратите внимание на то, что каждому пункту списка присвоен идентификатор. Идентификаторы понадобятся нам позднее, когда нужно будет указать, на какой странице сайта мы находимся, чтобы сделать вкладку активной.

УПРОЩЕНИЕ

Как правило, маркированный список без стилового оформления выглядит как простой список с маркерами в левой части строки (рис. 2.4). Любое устройство или текстовый браузер, не поддерживающие CSS, смогут воспроизвести его.

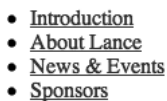
- 
- Introduction
 - About Lance
 - News & Events
 - Sponsors

Рис. 2.4. Маркированный список без стилового оформления

ДВА НЕБОЛЬШИХ ИЗОБРАЖЕНИЯ

Вы еще помните, что в рамках общепринятого подхода для каждого комплекта вкладок используются отдельные изображения? Фактически нам понадобятся четыре изображения: по одному для каждой зоны сайта, в которой активируется конкретная вкладка. Для того чтобы упростить процесс, предлагаю использовать только пару небольших изображений, которые мы разложим и скопируем по горизонтали на каждой из вкладок. Так как они будут находиться рядом, нам не нужно учитывать ширину или высоту каждого из элементов. К этому вопросу мы вернемся позднее в разделе «Пуленепробиваемый подход».

Если снова посмотреть на исходные вкладки, видно, что вертикальный градиент начинается в верхней части и постепенно переходит в сплошную заливку. Используем Adobe® Photoshop® для создания двух изображений (одно для активных состояний, второе — для неактивных), которые будут воспроизводить градиент, переходящий в прозрачный фон (рис. 2.5). Мы решили заменить цвет в нижней части каждого изображения на прозрачный, а позднее залить прозрачную часть с помощью CSS.

СОВЕТ

Еще одно преимущество списка заключается в следующем: большинство онлайн-приложений используют информацию о количестве элементов в списке. Это позволяет подсказать пользователю, что находится дальше. Например, пользователи популярной программы JAWS от Freedom Scientific могут пропустить список или перейти к следующему элементу на странице (более подробную информацию о навигации в JAWS см. по ссылке www.freedomscientific.com/fs_products/Surfs_Up/Navigating.htm).



Рис. 2.5. Два увеличенных изображения, которые мы будем использовать для создания градиента. Цвет в нижней части каждого изображения заменен на прозрачный (участок с шахматной заливкой), который мы с помощью CSS позднее заполним цветом

Обратите внимание на то, что каждое изображение имеет ширину 10 пикселей и включает 1-пиксельную линию по верху рамки для подсветки вкладки. После нее начинается градиент, переходящий в прозрачный фон.

ПРИМЕНЕНИЕ СТИЛЯ

Теперь с помощью CSS нам нужно свести вместе компоненты системы навигации. Первый этап — определить правила CSS, которые позволят разместить элементы навигации по горизонтали.

Так как роли меток WAI-ARIA используются в документе лишь один раз, воспользуемся ими для создания стиля вкладок с помощью селектора атрибутов, не присваивая идентификаторы или классы (еще одна причина полюбить метки!). Только для элемента `<nav>` можно определить роль маркера `navigation`, для которого зададим следующий стиль:

```
nav[role="navigation"] {  
    margin: 0;  
    padding: 10px 0 0 46px;  
    list-style: none;  
    background: #FFCB2D;  
}  
nav[role="navigation"] li {  
    float: left;  
    margin: 0 1px 0 0;  
    padding: 0;  
    font-family: "Lucida Grande", sans-serif;  
    font-size: 80%;  
}
```

СОВЕТ

Более подробная информация по заданию стилей с помощью атрибутов роли метки приведена в статье Джереми Кита по адресу <http://addactio.com/journal/4267/>.

Не забывайте, что для данных примеров понадобится HTML5 JS и переустановка таблицы стилей. Более подробную информацию можно найти во введении книги.

Используем свойство `float` для горизонтального размещения элементов. Параллельно объявим желтый цвет фона, поверх которого будут располагаться сами вкладки. Желтый фон очень важен, так как он является частью масштабируемой системы навигации (желтый контейнер и вкладки). Кроме того, мы сбросили свойства отступов и заполнения. При этом мы задали 1-пиксельный отступ справа (для разделения вкладок), указали шрифт и установили `font-size` равным 80% от значения по умолчанию. Используя знания, полученные в главе 1 «Гибкое управление текстом», для задания базового размера на странице мы воспользовались ключевым словом `small` в элементе `<body>`. Теперь можно спать спокойно: пользователи IE/Win смогут масштабировать текст на вкладках.

На рис. 2.6. показано, как выглядит наша система навигации в данный момент.



Рис. 2.6. А выглядит система навигации довольно неаккуратно

Понимаю, что она совсем не похожа на то, что вам хотелось бы увидеть. Но до результата осталось лишь несколько строчек кода.

ИСПОЛЬЗУЕМ ПЛАВАЮЩИЕ ЭЛЕМЕНТЫ ДЛЯ ФИКСАЦИИ

Первая проблема, которую необходимо решить: так как элементы `<li>` стали плавающими, они не имеют постоянного положения и не заполняют внешний тег `<ul>`, задающий фоновый цвет. Иными словами, так как внутренние элементы становятся подвижными, внешний тег `<ul>` не знает свою высоту.

Для решения этой проблемы сделаем `<ul>` плавающим, как остальные элементы. Это позволит `<li>` заполнить пространство, растягивая желтый фон. Так как элементы становятся плавающими, их содержимое, условно говоря, «упаковывается» (становится той ширины и высоты, которая необходима в конкретный момент времени). Определим ширину для `<ul>`, предполагая (в данном примере), что тег находится внутри макета с фиксированной шириной такого же размера (рис. 2.7).



Рис. 2.7. Делая элемент `<ul>` плавающим, мы растягиваем фон позади него

ПРИМЕЧАНИЕ

В теге `<ul>` мы указали ширину 720px, а слева добавили отступ 46px для разделения вкладок. Так как отступ является составной частью элемента, общая ширина навигационной панели составит 766px.

СОВЕТ

Так как `ul` превращается в плавающий элемент, необходимо убедиться в том, что свойства последующих элементов сброшены. Каждый последующий элемент, например горизонтальный элемент или блок контента, должен иметь свойство `clear: left`. Это необходимо для того, чтобы элемент оказался под навигационной панелью.
