

9 Построение навигационной системы сайта

Можно с уверенностью сказать, что без ссылок не было бы Всемирной паутины. Возможность находиться на одной веб-странице, а затем, щелкнув кнопкой мыши, перейти на другую страницу, расположенную на компьютере на другом конце света, — вот что делает Всемирную паутину такой полезной и незаменимой. Ссылки предоставляют своеобразный способ навигации и управления содержимым сайта. Именно поэтому дизайнеры веб-страниц прилагают столько усилий для создания привлекательных ссылок, работающих должным образом.

В этой главе вы познакомитесь с форматированием ссылок, позволив им визуально выделяться из контента страниц. Вы узнаете, как обеспечить визуальные подсказки, чтобы посетители сайта могли видеть, какие ссылки они уже посещали, а какие — нет. Вы также научитесь использовать каскадные таблицы стилей для создания на веб-страницах кнопок и панелей навигации, как это делают профессиональные разработчики. И в заключение, в практической части, мы создадим полный набор средств — элементов навигации, одинаково хорошо работающих во всех разновидностях браузеров.

Выборка форматируемых ссылок

В языке CSS, как правило, сначала нужно выбрать тот объект, для которого вы хотите определить стиль. Для этого надо сообщить CSS не только, *что* форматировать, то есть конкретный объект, но и *когда* форматировать. Браузеры отслеживают процесс взаимодействия посетителя сайта со ссылками и затем отображают их по-разному в зависимости от статуса (*состояния*). Таким образом, применяя селекторы, вам предоставляется возможность адресовать стили к ссылкам в определенном состоянии.

Состояния ссылок

Большинство браузеров распознают четыре основных состояния ссылок:

- непосещенная ссылка;
- посещенная ссылка (это означает, что по ссылке уже выполнялся переход, то есть URL-адрес сохранен в журнале браузера);

- ссылка, над которой находится указатель мыши;
- ссылка в момент нажатия.

Как описано в главе 3, каскадные таблицы стилей предоставляют четыре псевдокласса, соответствующие этим состояниям: `:link`, `:visited`, `:hover` и `:active`. Используя их, вы можете применить различное форматирование для каждого состояния ссылки, обеспечивая соответствующие подсказки и однозначно давая понять посетителю сайта, какие ссылки он уже посещал, а какие — нет.

ПРИМЕЧАНИЕ

Браузеры распознают также псевдоклассы `:focus`. Ссылка получает состояние `:focus`, когда посетители используют клавиатуру, чтобы перейти на ссылку (нажимают клавишу Tab). Этот псевдокласс также полезно применять с текстовыми полями веб-форм, что будет продемонстрировано на практике в главе 11.

Предположим, вы хотите изменить цвет текста непосещенной ссылки с синего на ярко-оранжевый. Для этого добавьте следующий стиль:

```
a:link { color: #F60; }
```

Щелкнув кнопкой мыши на этой ссылке, вы измените ее состояние на посещенное, а цвет в большинстве браузеров станет фиолетовым. Чтобы изменить его на насыщенно-красный, примените такой стиль:

```
a:visited { color: #900; }
```

СОВЕТ

Если требуется одинаково отформатировать все состояния ссылок, используйте один и тот же шрифт и размер для всех состояний, а затем отформатируйте элемент `a`, создав общий селектор `a`. Далее вы сможете использовать конкретные состояния ссылок, например `a:visited`, для изменения цвета или настройки конкретного состояния.

ВАЖНОЕ ЗАМЕЧАНИЕ

Ограничения псевдокласса `:visited`

Для обеспечения конфиденциальности браузеры накладывают ряд ограничений на свойства каскадных таблиц стилей, применяемые к псевдоклассу `:visited`. Некоторое время назад злоумышленники научились с помощью JavaScript-сценариев считывать изменения в стиле `:visited` ссылок, чтобы определить, какие сайты посещались. Например, путем загрузки нового фонового изображения для посещенных ссылок можно опреде-

лить, был ли посетитель на сайтах PayPal.com, eBay.com, BankofAmerica.com и т. д. Из-за возможности возникновения проблем наложены ограничения на изменение свойств `color`, `background-color` и `border-color` посещенных ссылок, за исключением форматирования непосещенных ссылок. То есть теперь уже нельзя рассчитывать на внесение изменений с помощью псевдокласса `:visited`.

Псевдокласс `:hover` предоставляет творческую свободу (мы изучим его в этой главе несколько позже). Он позволяет полностью изменить вид гиперссылки при наведении указателя мыши и перемещении его поверх ссылки. Если раньше для визуального представления кнопок навигации при наведении на них указателя мыши вы применяли объемный код на языке JavaScript, то вам понравится способ достижения такого же эффекта с помощью простейшей строки CSS-кода (см. раз-

дел «Форматирование ссылок» текущей главы). Для начала рассмотрим легкий пример, в котором стиль изменяет цвет ссылки при наведении и перемещении указателя мыши:

```
a:hover { color: #F33; }
```

СОВЕТ

Будьте внимательны при добавлении свойств к псевдоклассу `:hover`. Свойства, изменяющие размер элемента, поверх которого находится указатель мыши, могут повлиять на другие элементы, находящиеся рядом. Например, если вы увеличиваете размер шрифта текстовой ссылки, поверх которой находится указатель мыши, то при наведении указателя текст будет увеличиваться, выталкивая другие элементы с их позиций. Результат вам может не понравиться.

Теперь рассмотрим пример специально для «одержимых» веб-дизайнеров, которым нравится делать свои страницы непохожими на остальные. Изменим вид гиперссылки на доли миллисекунд в тот момент, когда посетитель щелкнул на ссылке. Для этого напишем следующий стиль:

```
a:active {color: #B2F511; }
```

В большинстве случаев для обеспечения максимальной гибкости требуется включить в таблицу стилей псевдоклассы `:link`, `:visited` и `:hover`. Но для того, чтобы этот метод работал, вы должны расположить ссылки в определенной последовательности: `link`, `visited`, `hover` и `active`. Ниже представлен способ добавления всех четырех стилей ссылок:

```
a:link { color: #F60; }  
a:visited { color: #900; }  
a:hover { color: #F33; }  
a:active {color: #B2F511; }
```

Если вы нарушите последовательность, то стили `hover` и `active` перестанут функционировать. Например, если поместить `a:hover` перед `a:link` и `a:visited`, то при наведении на ссылку указателя мыши ее цвет не изменится.

ПРИМЕЧАНИЕ

Почему последовательность имеет такое значение? Здесь работает механизм каскадности (см. главу 5). Все эти стили имеют одинаковый приоритет, значит, порядок их следования в программном коде определяет стиль, который будет применен в конечном счете. Поскольку ссылка одновременно может быть непосещенной и наведенной, то `a:link`, будучи указанным последним, согласно правилам каскадности имеет более высокий приоритет, и `a:hover` с иным цветом никогда не будет применен.

Выборка отдельных ссылок

Простейшие стили, которые мы создавали в предыдущем разделе, представляют собой форматирование элементов `a`. Они производили выборку в определенном состоянии, но при этом форматировали абсолютно *все* ссылки, независимо от их расположения на веб-странице. Что же делать, если вы хотите отформатировать одни ссылки одним способом, а другие — другим? Существует простое решение — применить классы к нужным элементам.

Допустим, на веб-странице имеется набор ссылок и часть из них указывает на другие сайты, которые вы хотите выделить (например, сайты ваших друзей, партнеров, спонсоров). Возможно, вы захотите их отформатировать таким образом, чтобы посетители заранее знали, что это какие-то особенные ссылки, и захотели бы перейти по ним. В данном случае можете применить такой класс:

```
<a href="http://www.hydroponiconline.com" class="sponsor">Посетите этот замечательный сайт</a>
```

Для форматирования этой ссылки другим способом создадим следующий стиль:

```
a.sponsor { font-family: Arial, sans-serif; }
a.sponsor:link { color: #F60; }
a.sponsor:visited { color: #900; }
a.sponsor:hover { color: #F33; }
a.sponsor:active {color: #B2F511; }
```

Определения только имени класса, без указания элемента `a`, будет тоже достаточно:

```
.sponsor { font-family: Arial, sans-serif; }
.sponsor:link { color: #F60; }
.sponsor:visited { color: #900; }
.sponsor:hover { color: #F33; }
.sponsor:active {color: #B2F511; }
```

Теперь такое форматирование приобретут лишь ссылки с классом `sponsor`.

ПРИМЕЧАНИЕ

В приведенных примерах в демонстративных целях мы изменили только цвет ссылок. На практике для форматирования вы можете использовать любые свойства каскадных таблиц стилей (за исключением ссылок `:visited`, как объяснялось во врезке «Важное замечание» выше). В следующем примере вы увидите, что существует множество способов креативного подхода к форматированию.

Группирование ссылок с помощью селекторов потомков

Если набор ссылок располагается в одной области веб-страницы, то вы можете сэкономить время, применив *селекторы потомков*. Предположим, у вас есть пять ссылок, которые ведут на основные разделы вашего сайта. Они представляют собой главную панель навигации, и вы хотите придать им характерный вид. Заключите эти ссылки в HTML5-элемент `nav`. Теперь появилась возможность произвести выборку и отформатировать только эти ссылки:

```
nav a { font-family: Arial, sans-serif; }
nav a:link { color: #F60; }
nav a:visited { color: #900; }
nav a:hover { color: #F33; }
nav a:active {color: #B2F511; }
```

Вместо этого можно воспользоваться элементом `div` и добавить к нему класс:

```
<div class="mainNav">
```

```
.mainNav a { font-family: Arial, sans-serif; }  
.mainNav a:link { color: #F60; }  
.mainNav a:visited { color: #900; }  
.mainNav a:hover { color: #F33; }  
.mainNav a:active {color: #B2F511; }
```

Использование селекторов потомков облегчает процесс форматирования ссылок, которые должны выглядеть по-разному в каждой области веб-страницы (полное описание неисчерпаемых возможностей таких селекторов вы найдете в разделе «Использование селекторов потомков» главы 18).

СОВЕТ

Применение маркированных списков для представления ссылок — очень распространенный метод (скоро вы увидите пример). В таком случае вы можете добавить идентификатор или класс к элементу `ul` списка, например `<ul class="mainNav">`, а затем создать селекторы потомков наподобие `.mainNav a:link` для их форматирования.

Форматирование ссылок

Теперь, когда вы знаете, как создавать селекторы, производящие выборку конкретных ссылок, нужно определить, как форматировать ссылки. Да как угодно! Вам предоставляется полный арсенал свойств языка CSS, а творческие способности ограничены только воображением. Единственное, что нужно учесть, — ваши ссылки должны выглядеть *как ссылки*. Они не обязательно будут синего цвета с подчеркиванием, но пусть они отличаются от остального контента веб-страниц, чтобы посетители сайта сразу могли понять, что они *кликабельны*.

Если сделать ссылку похожей на кнопку, у которой появляются границы и изменяется цвет фона при наведении на нее указателя мыши, большинство посетителей поймет, что они могут щелкнуть на ней. Можно также воспользоваться линейным градиентом для добавления к ссылке текстуры и глубины. Ссылки, расположенные в объемных фрагментах текста, должны четко выделяться. Этого можно добиться использованием полужирного начертания шрифта, сохранив традиционное подчеркивание, а также добавив стиль, изменяющий внешний вид ссылки при установке на нее указателя мыши. В качестве подсказки можно добавить рисунок (например, в виде стрелки), который визуально подскажет посетителю о том, что щелчок кнопкой мыши перенесет его в какой-либо другой раздел текущего сайта или вообще на другой сайт.

Подчеркивание ссылок

Еще с самого появления Всемирной паутины ярко-синий подчеркнутый текст сигнализировал: «Нажмите здесь, чтобы перейти туда-то». Однако стандартные атрибуты подчеркивания и цветового оформления ссылок — это то, что в первую очередь изменяет любой веб-дизайнер. Подчеркивание — слишком распространенный способ выделения, который порядком поднадоел (рис. 9.1, 1). У вас есть возможность изменить ситуацию, одновременно обеспечив хорошее оформление ссылок.

- **Полное удаление подчеркивания.** Чтобы убрать стандартное подчеркивание, используйте свойство `text-decoration` с присвоенным ему значением `none`:

```
a {text-decoration: none;}
```

Естественно, полное удаление подчеркивания может смутить посетителей сайта. Если вы не предусмотрите каких-то других визуальных подсказок, то ваши ссылки будут выглядеть точно так же, как и весь остальной текст веб-страницы (см. рис. 9.1, 2). Если вы пойдете этим путем, то обеспечьте выделение текста ссылок каким-то другим способом, например полужирным начертанием (см. рис. 9.1, 3), цветом фона, подсказкой в виде рисунка или преобразованием ссылки в имитированную кнопку.

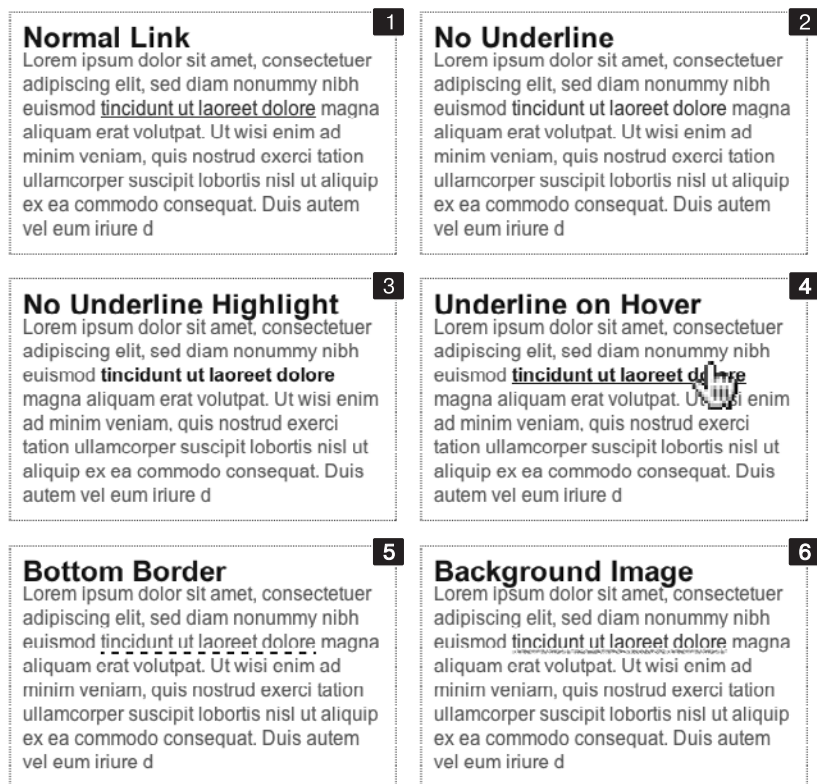


Рис. 9.1. Существует множество способов сделать стандартное подчеркивание более привлекательным

- **Добавление подчеркивания только при наведении на ссылку указателя мыши.** Некоторые веб-дизайнеры убирают подчеркивание для всех ссылок, выделяют их каким-то другим способом, а затем включают атрибут подчеркивания только при наведении указателя мыши, как показано на рис. 9.1, 4. Чтобы обеспечить такой эффект, удалите подчеркивание ссылок, а затем повторно введите его с помощью псевдокласса `:hover`:

```
a {
  text-decoration: none;
  background-color: #F00;
}
a:hover {
  background-color: transparent;
  text-decoration: underline;
}
```

- **Использование свойства border-bottom.** Так вы можете управлять цветом, толщиной и стилем стандартной линии подчеркивания ссылок, которая по умолчанию имеет толщину 1 пиксел и тот же цвет, что и текст. Большого разнообразия можно добиться путем использования вместо подчеркивания свойства border-bottom, как показано на рис. 9.1, 5. Скрыть обычное подчеркивание и добавить черту в виде пунктирной линии-границы можно следующим образом:

```
a {
  text-decoration: none;
  border-bottom: dashed 2px #9F3;
}
```

Вы можете изменить стиль, толщину линии и цвет границы. Чтобы увеличить промежуток между текстом и границей, добавьте свойство padding.

- **Использование фонового изображения.** Вы можете преобразить вид ссылок, добавив фоновый рисунок. Например, на рис. 9.1, 6 показана граница в виде рукописной линии. Подобная методика подчеркивания заголовков описана в практикуме главы 8. Напомню суть метода. Сначала создается рисунок линии подчеркивания с помощью графического редактора типа Fireworks или Photoshop, в котором имеется инструмент Кисть (Brush), имитирующий рисование мелом, фломастером и т. д. Затем создается стиль для ссылки, убирающий стандартное подчеркивание и добавляющий фоновое изображение. Оно должно быть расположено по нижнему краю ссылки и повторяться в горизонтальном направлении. Можно также добавить небольшой отступ снизу для изображения линии:

```
a {
  text-decoration: none;
  background: url(images/underline.gif) repeat-x left bottom;
  padding-bottom: 5px;
}
```

Создание кнопок

Вы можете придать ссылкам вид кнопок, присутствующих в окнах и на панелях инструментов компьютерных программ. В этом вам помогут свойства border, background-color и padding. С их помощью можно легко создавать прямоугольные кнопки разного вида (рис. 9.2).

Допустим, вы назначили класс ссылке, которую хотите отформатировать в виде кнопки: `Скидки!`. Чтобы добавить вокруг

этой ссылки простейший контур черного цвета (см. рис. 9.2, *вверху слева*), создайте следующий стиль:

```
a.button {  
    border: solid 1px rgb(0,0,0);  
}
```

Можете также окрасить кнопку, установив цвет фона:

```
a.button {  
    border: solid 1px rgb(0,0,0);  
    background-color: rgb(51,51,51);  
}
```

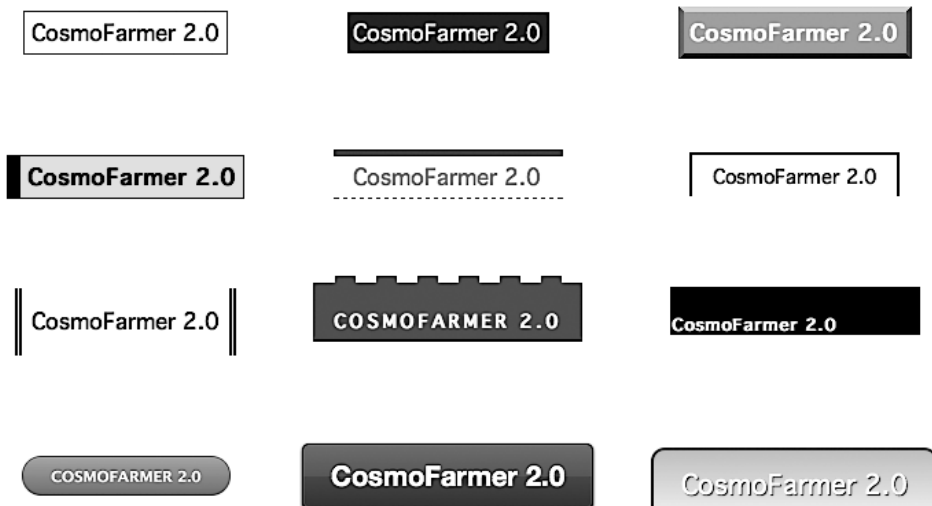


Рис. 9.2. Чтобы превратить ссылку в кнопку, на которой хочется щелкнуть, следует использовать свойства `border`, `background` и `padding`

Можно создать более впечатляющие кнопки, сочетая различные рамки или варьируя цвета, стили и ширину со свойством `background-color`. А для тех браузеров, которые поддерживают свойства `linear-gradient`, `text-shadow` и `box-shadow`, а также скругленные углы со свойством `border-radius`, можно добиться для ссылок весьма привлекательного вида (нижний ряд на рис. 9.2).

ПРИМЕЧАНИЕ

В этих примерах и `a.button`, и `.button` подойдут в качестве имени стиля. В случае `a.button` стиль применяется только к элементам `a` с классом `button`, в то время как `.button` относится к любому элементу с этим именем класса. Если вы хотите быть уверенными, что стиль применяется только к конкретному элементу, то добавьте имя элемента в начале. Добавление имени элемента также полезно при просмотре CSS-кода — оно дает понять, для форматирования какого контента предназначен стиль. Когда вы видите `a.button`, становится ясно, что стиль нацелен на определенные ссылки.

Имейте в виду: совсем не обязательно, чтобы линии границ со всех четырех сторон ссылки-кнопки были одинакового стиля. Возможно, что у ссылки даже не будет каких-либо границ. Широко применяемая дизайнерская методика превращения плоской кнопки в объемную предполагает использование четырех границ разного цвета, как показано в верхнем правом углу на рис. 9.2. Придание кнопке объемности не представляет сложности, но вы должны помнить, что освещение заставляет предмет выглядеть таким. Представьте, что свет падает на одну из четырех сторон; эта обращенная к источнику света сторона — самая светлая, а противоположная — самая темная (поскольку приподнятая кнопка в ненажатом состоянии препятствует прохождению света и вызывает появление «тени»). Остальные две стороны должны иметь цвета промежуточных оттенков между «светлыми» и «темными». Рассмотрим CSS-код для придания кнопке-ссылке, отображенной в верхнем правом углу на рис. 9.2, объемного вида:

```
a.button {
  background: #B1B1B1;
  color: #FFF;
  font-weight: bold;
  border-width: 4px;
  border-style: solid;
  border-top-color: #DFDFDF;
  border-right-color: #666;
  border-bottom-color: #333;
  border-left-color: #858585;
}
```

Вы также можете (и это рекомендуется) создать стиль для кнопки в состоянии :hover. Кнопки будут реагировать на наведение посетителем на ссылку указателя мыши, обеспечивая интерактивность и обратную связь. В случае с объемной кнопкой при ее нажатии очень эффективен метод инверсной смены цветов: темный фон кнопки должен стать светлым, светлая граница — темной и т. д.

По-настоящему добавить глубину кнопке можно, используя линейные градиенты и скругленные углы, блочные и текстовые тени. Например, нижняя средняя кнопка на рис. 9.2 создана с помощью следующего кода:

```
background-color: #ee432e;
background-image: -webkit-linear-gradient(top, #ee432e 0%, #c63929 50%, #b51700 50%, #891100 100%);
background-image: -moz-linear-gradient(top, #ee432e 0%, #c63929 50%, #b51700 50%, #891100 100%);
background-image: -o-linear-gradient(top, #ee432e 0%, #c63929 50%, #b51700 50%, #891100 100%);
background-image: linear-gradient(top, #ee432e 0%, #c63929 50%, #b51700 50%, #891100 100%);
border: 1px solid #951100;
border-radius: 5px;
box-shadow: inset 0px 0px 0px 1px rgba(255, 115, 100, 0.4), 0 1px 3px #333333;
padding: 12px 20px 14px 20px;
text-decoration: none;
color: #fff;
```

```
font: bold 20px/1 "helvetica neue", helvetica, arial, sans-serif;  
text-align: center;  
text-shadow: 0px -1px 1px rgba(0, 0, 0, 0.8);
```

В строке 1 устанавливается темно-красный цвет фона; этот цвет предоставляется для браузеров, не поддерживающих линейные градиенты (Internet Explorer 9 и более ранние версии, а также другие устаревшие браузеры). В строках 2–5 задается линейный градиент для различных браузеров. В строке 6 добавляется простая граница, а в строке 7 создаются скругленные углы, после чего в строке 8 добавляется тонкая внутренняя тень и вторая, внешняя тень ниже кнопки, то есть применяются две тени, что вполне допустимо. В строке 9 добавляется отступ, обеспечивающий появление пустого пространства вокруг текста кнопки, а в строке 10 удаляется подчеркивание, которое обычно появляется под ссылками. В последних строках устанавливаются свойства для текста в кнопке, включая цвет, шрифт, выравнивание и текстовую тень.

СОВЕТ

Создание привлекательных кнопок часто требует целого набора свойств каскадных таблиц стилей (что можно заключить из приведенного выше кода). Если в создании всего этого кода требуется помощь, обратитесь к генератору кнопок CSS по адресу bestcssbuttongenerator.com. Чтобы взглянуть на другие впечатляющие CSS-кнопки, посетите сайты tinyurl.com/nmfmrxc и tinyurl.com/7jsqkb5.

Применение изображений

Добавление к ссылкам изображений — один из самых легких и эффективных способов улучшить внешний вид элементов навигации сайта. Существует множество методик и вариантов дизайна, но при этом следует заметить, что ни в одном из грамотных методов не применяется HTML-элемент `img`. Вместо этого используется свойство `background-image`, с помощью которого можно повысить привлекательность любой ссылки. Несколько примеров приведено на рис. 9.3 (более совершенные методы использования изображений для создания графических кнопок и эффектов ролловеров¹ описаны в практикуме этой главы).

Чтобы вспомнить свойство `background-image` и все, что с ним связано, вернитесь к разделу «Добавление фоновых изображений» главы 8. Пока же напомним вам несколько моментов, которые нужно иметь в виду при использовании изображений со ссылками.

- **Не забывайте про значение `no-repeat`.** По умолчанию фоновые изображения выводятся на заднем плане формируемого элемента. Со многими рисунками, применяемыми для ссылок, это приводит к плохому результату (см. рис. 9.3, *вверху справа*). Если вы не используете едва заметный узор, похожий на градиентную заливку, не забудьте отключить повторное отображение фонового рисунка: `background-repeat: no-repeat;`
- **Позиционируйте с помощью свойства `background-position`.** Чтобы точно поместить фоновое изображение, используйте свойство `background-position` (см. раздел

¹ Ролловер — это элемент веб-страницы, изменяющий свой вид в зависимости от внешнего воздействия.

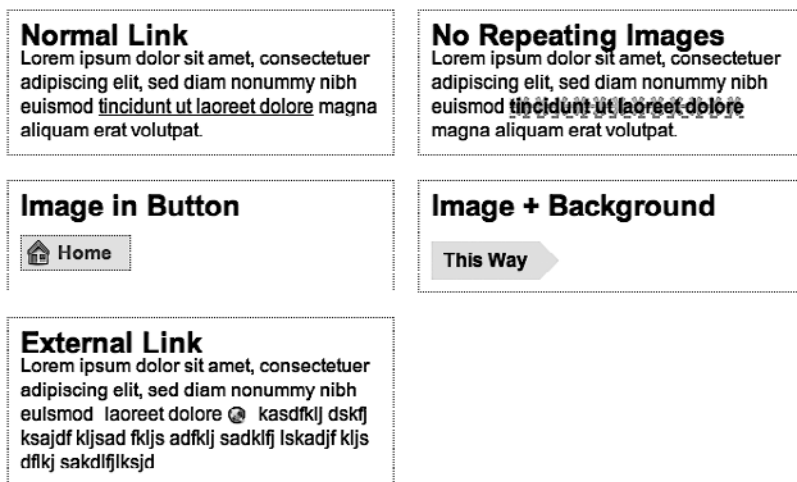


Рис. 9.3. Даже простой рисунок может оживить ссылку и сделать ее более понятной для посетителя: значок земного шара (внизу слева) — один из способов обозначения ссылок во Всемирной паутине

«Позиционирование фоновых изображений» главы 8). Если необходимо позиционировать изображение по правому краю ссылки, но при этом центрировать его вертикально на строке, то добавляйте следующий CSS-код: `background-position: right center;`.

Для еще более точного размещения изображений используйте значения параметров в пикселах или единицах `em`. Эти единицы измерения упрощают смещение рисунка на нужное расстояние от левого края ссылки. Комбинируя их с процентными значениями, вы сможете не только легко центрировать изображение вертикально по отношению к тексту ссылки, но и обеспечить его небольшое смещение на точное расстояние от левого края: `background-position: 10px 50%;`.

СОВЕТ

При позиционировании фоновых изображений первое значение свойства `background-position` определяет положение по горизонтали (по направлению слева направо); а второе — по вертикали (по направлению сверху вниз).

Увы, но не существует способа точно позиционировать изображение по отношению к правому или нижнему краю ссылки. Поэтому, если вы хотите поместить рисунок на небольшом расстоянии от правого края, доступно два варианта. Первый заключается в том, чтобы в графическом редакторе добавить с правой стороны рисунка пустое пространство. Размер этого отступа должен быть равен тому расстоянию, на которое вы хотите сместить изображение от правого края ссылки. После этого для его выравнивания по правому краю формируемого элемента используйте свойство `background-position`; например `background-position: right top;`. Кроме того, можете использовать процентные значения: `background-position: 90% 75%;` — такие параметры обеспечат позиционирование

точки изображения, находящейся на расстоянии 90 % от его левого края, на расстоянии 90 % от левого края формируемого элемента. Однако такой метод не обеспечивает точности позиционирования, так что придется немного поэкспериментировать (о том, как работает позиционирование с процентными значениями, написано в разделе «Позиционирование фоновых изображений» главы 8).

- **Добавляйте отступы с помощью свойства `padding`.** Если вы используете для выделения ссылки изображение или значок, нужно добавить отступ с той стороны, где рисунок примыкает к тексту. Так, в примере 3 на рис. 9.3 ссылка имеет отступ слева шириной 30 пикселей для предотвращения наложения текста (слово Home) ссылки на изображение дома, в то же время, как показано на нижнем изображении слева, небольшой отступ справа обеспечивает пространство для отделения текста ссылки от значков земного шара.

ПРИМЕЧАНИЕ

Поскольку элемент `a` — строчный, добавление верхнего или нижнего отступов (или полей) не окажет никакого эффекта. Причина такого поведения описана в подразделе «Отображение строчных и блочных элементов» раздела «Управление размерами полей и отступов» главы 7. Однако можно преобразовать ссылку в блочный элемент (`display: block`), чтобы к нему можно было применить верхние и нижние отступы и поля, или воспользоваться свойством `inline-block` (`display: inline-block`). Оба этих метода будут рассмотрены позже.

- **Пользуйтесь псевдоклассами.** Не забывайте о псевдоклассах `:hover` и `:visited`. Они помогут создать для ссылок замечательные динамические эффекты и обеспечить полезную обратную связь. Можно использовать *разные* фоновые рисунки для каждого из этих псевдоклассов, чтобы, например, изображение непосещенной ссылки в виде выключенной лампочки при наведении на нее указателя мыши сменялось на изображение включенной.

Если вы решите использовать изображение для ссылки в состоянии `:hover`, помните, что браузеры не загрузят рисунок, пока посетитель сайта на самом деле не наведет на него указатель мыши. Может возникнуть значительная задержка, прежде чем появится изображение. Однако уже после первой загрузки задержка исчезнет. О том, как решить данную проблему, читайте в подразделе «Использование ролловеров» далее в этой главе.

Создание панелей навигации

Каждому сайту требуется удобная панель управления с элементами навигации: во-первых, чтобы привести посетителей к нужной им информации, а во-вторых, чтобы им понравилось управление сайтом и они вернулись сюда вновь. Контент большинства сайтов организован в виде разделов. Например, продукция, контактная информация, журнал отзывов и т. д. Такая организация позволяет посетителям успешно ориентироваться в содержимом сайта, и они точно знают, что и где смогут найти. В большинстве случаев ссылки на основные разделы сайта можно найти на *панели навигации*. Каскадные таблицы стилей облегчают создание красивых и функциональных панелей, ролловеров и т. д.

Использование маркированных списков

По сути, панель навигации — не что иное, как группа ссылок, точнее, *список* разделов сайта. Как разъясняется в главе 1, задача языка HTML состоит в том, чтобы обеспечить структурирование. Следовательно, вы всегда должны использовать HTML-элементы в соответствии с их функциональным назначением, чтобы они соответствовали заключенному в них содержанию. К примеру, для списка элементов это должен быть `ul`, элемент неупорядоченного списка. Не имеет значения, будет список вообще *без* маркеров или будет располагаться горизонтально вдоль верхнего края веб-страницы: все форматирование элемента `ul` обеспечено средствами языка CSS. Пример показан на рис. 9.4.

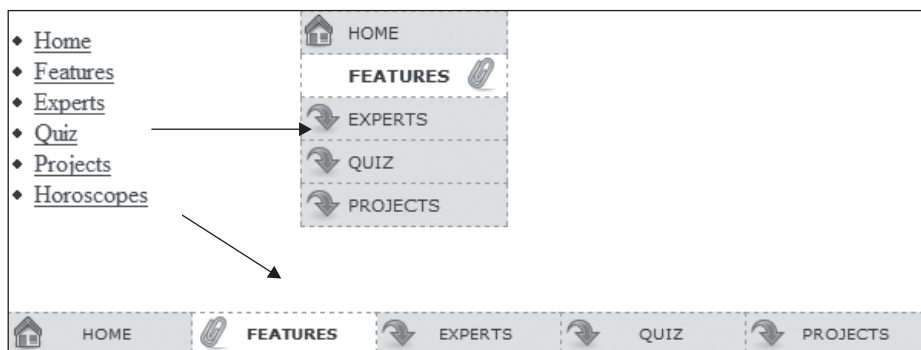


Рис. 9.4. Используя каскадные таблицы стилей, вы можете превратить обычные элементы `ul` в, например, вертикальное или горизонтальное управляющее меню (панель навигации)

Применение HTML для создания панели навигации — это и есть использование языка разметки по прямому назначению. В каждом элементе списка имеется по одной ссылке. Кроме того, нам всего лишь необходимо форматировать этот маркированный список (*не нужно*, чтобы его пункты составляли панель навигации). Правильным подходом будет применение класса или идентификатора к элементу `ul`:

```
<ul class="nav">
<li><a href="index.html">Главная</a></li>
<li><a href="news.html">Новости</a></li>
<li><a href="reviews.html">Обзоры</a></li>
</ul>
```

CSS-код немного отличается в зависимости от того, какая нужна панель навигации — горизонтальная или вертикальная. В любом случае вы должны сделать два шага.

- 1. Удалить маркеры.** Чтобы панель навигации не была похожа на маркированный список, удалите маркеры, присвоив свойству `list-style-type` значение `none`:

```
ul.nav {
  list-style-type: none;
}
```

2. **Убрать отступы и поля.** Поскольку браузеры сами создают отступы для элементов списка, нам придется избавиться от них. Однако одни браузеры используют для этих целей свойство `padding`, а другие — `margin`, поэтому нам нужно присвоить обоим свойствам значение 0:

```
ul.nav {  
  list-style-type: none;  
  padding-left: 0;  
  margin-left: 0;  
}
```

По сути, эти два шага обеспечивают всем элементам списка простой вид, как у обычного блочного элемента — абзаца или заголовка (с одним исключением: браузер не добавляет полей между элементами списка). С этого момента можно начинать форматирование. Если вам нужна вертикальная панель навигации, продолжайте читать дальше; если же хотите создать горизонтальную панель управления, то пропустите следующий раздел.

Вертикальные панели навигации

Вертикальная панель навигации — это всего лишь набор ссылок, расположенных одна за другой в виде списка или стека. Удалив маркеры, а также поля и отступы слева (как описано в предыдущем разделе), вы выполните большую часть работы, но вы должны знать, что существует еще несколько дополнительных хитростей, обеспечивающих правильное отображение панели навигации.

1. Отобразите ссылку в виде блочного элемента.

Поскольку элемент `a` — строчный, ссылка будет занимать по ширине пространство, равное ширине содержимого элемента. Кнопки со ссылками в виде текста различной ширины (например, **Главная** и **Продукция**) будут иметь различную ширину. Ступенчатое вертикальное отображение кнопок друг над другом выглядит не очень хорошо (рис. 9.5, 1). Кроме того, верхние и нижние отступы и поля не оказывают никакого эффекта на строчные элементы. Чтобы обойти эти ограничения, преобразуем ссылки в блочные элементы:

```
ul.nav a {  
  display: block;  
}
```

Значение `block` не только выравнивает все кнопки по ширине, но и делает всю область ссылок активной при щелчке по ней кнопкой мыши, подобно настоящим кнопкам. Это тот случай, когда посетители щелкают на области кнопки, где нет текста (например, отступы вокруг), но ссылка по-прежнему работает.

2. Ограничьте ширину кнопок.

Преобразование пунктов списка в блочные элементы обеспечивает ссылкам ширину элементов, в которые они вложены. Поскольку ссылки просто помещены на веб-страницу, их ширина окажется равной ширине окна браузера (см. рис. 9.5, 2). Есть несколько способов ограничить ширину ссылок. Сначала можно установить

ширину элемента `a`. Например, если вы хотите, чтобы все кнопки имели ширину 8 em, добавьте в стиль свойство `width` следующим образом:

```
ul.nav a {  
  display: block;  
  width: 8em;  
}
```



Рис. 9.5. Всего четырьмя простыми действиями можно превратить маркированный список ссылок в привлекательную панель навигации

Установка ширины любого элемента, в который заключены ссылки, например `li` или `ul`, приведет к тому же результату.

Если текст кнопки занимает всего одну строку, то можно его центрировать вертикально, чтобы над и под ним были одинаковые промежутки. Укажите высоту ссылки `a` и назначьте такое же значение свойства межстрочного интервала:

```
a {  
  height: 1.25em;  
  line-height: 1.25em;  
}
```

ПРИМЕЧАНИЕ

Не обязательно явно назначать свойство `width`, если панель навигации вложена в элемент разметки веб-страницы, для которого уже установлена ширина. В части III вы узнаете, каким образом элементарно создать боковую панель, примыкающую к левому или правому краю веб-страницы.

Боковая панель имеет свою ширину, и помещенный в нее маркированный список со ссылками-кнопками автоматически приобретает эту ширину.

Теперь, когда мы закончили всю эту малопродуктивную работу, займемся основной задачей — форматированием кнопок. Можно добавить отступы, поля, изображения, изменить фоновый цвет. Если хотите расположить кнопки так, чтобы они не соприкасались между собой, можете добавить нижнее (или верхнее) поле к каждой из ссылок.

Горизонтальные панели навигации

Каскадные таблицы стилей позволяют преобразовать список ссылок, расположенных вертикально одна за другой, в список с горизонтальным представлением, как было показано на рис. 9.4. В этом разделе описываются два распространенных подхода к созданию из списка горизонтальной панели навигации. Первый метод основывается на использовании свойства `display: inline` — он прост, но обычно приводит к появлению маленьких расстояний между кнопками. Проблема решается, но для этого придется возиться с HTML-кодом. Если нужна горизонтальная навигационная панель с соприкасающимися кнопками, обратитесь к методу с применением плавающих элементов `u1`.

ОБХОДНОЙ ПРИЕМ

Если границы соединяются

Если на вертикальной панели навигации кнопки расположены вплотную друг к другу, то линии границ соседних ссылок сливаются между собой и превращаются в двойные. Другими словами, граница одной кнопки касается границы следующей кнопки.

Чтобы избавиться от этого, можно добавить границу только к *верхнему* краю ссылок. Таким образом, мы получим всего одну границу, отделяющую кнопки, расположенные друг над другом.

Однако в результате этого обходного приема последняя, нижняя ссылка панели навигации получается без границы. Чтобы решить проблему, можно создать класс со стилем нижней границы и применить

его к последней ссылке, но при этом придется дописывать HTML-код. Другой вариант — добавить нижнюю границу к элементу `u1`, завершающему панель навигации. Или можете использовать селектор `:last-of-type` для добавления нижней границы к последней ссылке (этот прием будет продемонстрирован на практике этой главы).

То же самое справедливо и для горизонтальных ссылок. В этом случае границы левой и правой кнопок касаются друг друга. Удалите левую границу у ссылок и добавьте ее, к примеру, к элементу-контейнеру `u1` или используйте селектор `:first-of-type`, чтобы выделить первую ссылку на панели и добавить границу к ее левому краю.

Какой бы метод вы ни использовали, сначала удалите маркеры и пространство слева из элементов `u1`, как показано на рис. 9.6, 1.

Использование свойств `display:inline` и `display:inline-block`

Самый простой метод создания горизонтальной панели навигации заключается в смене для элементов списка значения `block` (блочный элемент) свойства `display`

на inline (строчный элемент). Это делается средствами каскадных таблиц стилей.

1. Создайте стиль для маркированного списка, чтобы удалить отступы, поля и маркеры.

```
ul.nav {  
  margin-left: 0px;  
  padding-left: 0px;  
  list-style: none;  
  border-bottom: 1px dashed #000;  
}
```

В данном случае добавляется нижняя граница, которая появится ниже кнопок (см. рис. 9.6, 1).

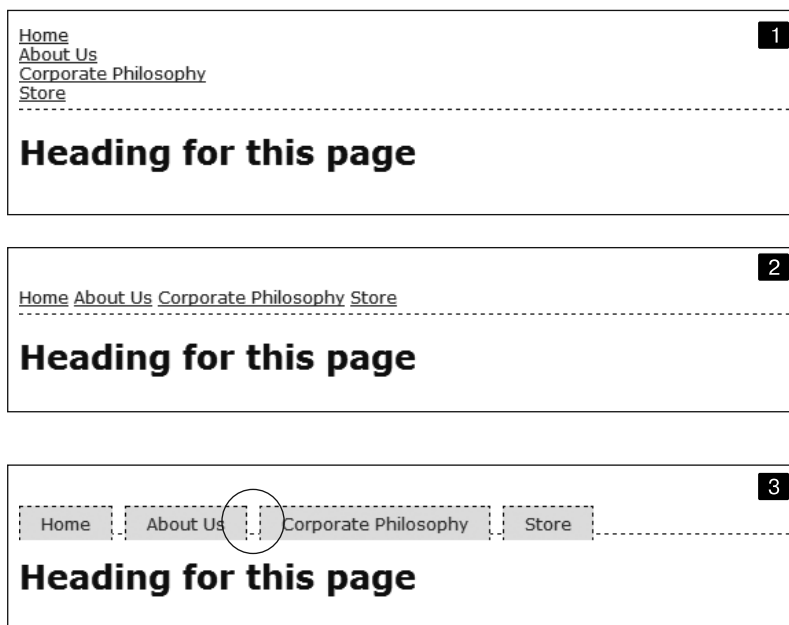


Рис. 9.6. Создание горизонтального меню из списка ссылок потребует выполнения всего нескольких действий

2. Преобразуйте пункты списка в строчные элементы.

Строчные элементы не создают разрывов строк ни перед элементом, ни после него, как это делают блочные. Присвоение значения inline свойству display элементов li приведет к отображению элементов списка в одну строку (см. рис. 9.6, 2).

```
.nav li { display: inline; }
```

Вы должны быть уверены в том, что панель навигации кнопок-ссылок не слишком большая. Те кнопки, которые не поместятся в одну строку, перенесутся на другую.

3. Отформатируйте ссылки.

Можно убрать подчеркивание под ссылками и вместо этого добавить вокруг них окаймляющую границу. Чтобы обеспечить визуальную глубину и реалистичность кнопок, надо изменить цвет фона или добавить рисунок. При необходимости добавьте отступы вокруг текста ссылок. Если требуется разнести кнопки друг от друга, установите для них правое поле. Следующий стиль обеспечивает ссылкам визуальное подобие кнопок, как показано на рис. 9.6, 3:

```
.nav a {  
  display: inline-block;  
  border: 1px dashed #000;  
  border-bottom: none;  
  padding: 5px 15px 5px 15px;  
  background-color: #EAEAEA;  
  text-decoration: none;  
  color: #333;  
}
```

Сначала нужно присвоить свойству `display` ссылок значение `inline-block`. После этого верхние и нижние отступы и поля будут соответствовать требованиям (как уже упоминалось, обычно строчные элементы игнорируют верхние и нижние отступы и поля, а также значения ширины и высоты). Затем можно будет указать стиль кнопок в соответствии с вашим вкусом. Здесь к ссылкам добавляется граница и удаляется нижняя граница, чтобы исключить дублирование нижней границы, применяемой к маркированному списку.

СОВЕТ

Определив ширину ссылок, все кнопки можно сделать одинакового размера.

Чтобы эта навигационная панель отображалась в центре страницы, к стилю элемента `ul` нужно добавить объявление `text-align: center;`. У такой технологии по сравнению с рассматриваемой в следующем разделе есть одно преимущество — при использовании значений `inline` и `inline-block` панель навигации можно отцентрировать, чего нельзя сделать с `float`.

Но можно заметить, что кнопки не соприкасаются (см. область, выделенную на рис. 9.6). Это связано с тем, как браузеры рассматривают пустое пространство между элементами `li`. Возьмем, например, следующий фрагмент HTML-кода:

```
<ul class="nav">  
<li><a href="index.html">Главная</a></li>  
<li><a href="news.html">Новости</a></li>  
<li><a href="reviews.html">Обзоры</a></li>  
</ul>
```

Пробельные символы (символ табуляции, возврата каретки или пробел) между закрывающим тегом `` и следующим открывающим тегом `` браузеры рассматривают как пустое пространство. Чтобы его удалить, доступны два способа.

- Разместить закрывающий тег `` и открывающий тег `` на одной и той же строке:

```
<ul class="nav">
<li><a href="index.html">Главная</a></li><li>
<a href="news.html">Новости</a></li><li>
<a href="reviews.html">Обзоры</a></li>
</ul>
```

Обычно так код не записывают, и такие HTML-редакторы, как Dreamweaver, так бы не сделали. Чтобы удалить пустое пространство, нужно внести изменения в код вручную.

- Добавить к элементам списка правое поле с отрицательным значением. Например, в стиль `li` в показанном выше шаге 2 можно внести следующее изменение:

```
.nav li {
  display: inline;
  margin-right: -5px;
}
```

Отрицательное значение поля приводит к наложению следующего элемента списка на 5 пикселей, закрывая промежуток между кнопками. Проблема при использовании этой технологии заключается в том, что конкретное используемое значение варьируется в зависимости от размера текста — 5 пикселей могут сработать, а могут — и нет, поэтому для получения нужного значения следует провести эксперимент.

Панели навигации на основе плавающих элементов

Другой, более популярной, технологией является применение плавающих элементов списка. Такой метод также позволяет поместить ссылки бок о бок. Кроме того, эта технология не страдает от проблемы пустого пространства, присущей строчному методу.

ПРИМЕЧАНИЕ

Панели навигации на основе плавающих элементов трудно центрировать горизонтально посередине веб-страницы. Если требуется такое выравнивание, то лучше пользоваться методом со строчными элементами, описанным в этой главе, или методом `flexbox`, показанным в практикуме к главе 17.

1. Преобразуем пункты списка в плавающие элементы.

Добавление к элементам `li` свойства `float` с выравниванием по левому краю исключает их из обычного нисходящего потока элементов:

```
.nav li { float: left; }
```

Плавающие элементы списка (вместе с вложенными ссылками) располагаются на одной строке, точно так же, как изображения фотогалереи из практикума главы 8 (при необходимости можно с такой же легкостью установить выравнивание кнопок по правому краю экрана (окна браузера) или боковой панели, в которую заключены кнопки).

ЧАВО

Всплывающие меню

Как создать стильные современные всплывающие меню, которые показывают подменю со ссылками при наведении указателя мыши на кнопки панели навигации?

Панели навигации в виде всплывающих многоуровневых меню чрезвычайно популярны. Они представляют собой отличный способ вместить огромное количество ссылок в компактную панель навигации. Вы можете создать их несколькими способами.

Один из методов создания всплывающих меню использует только средства каскадных таблиц стилей. Основное руководство по процессу создания можно найти по адресу tinyurl.com/o7bhpw5. Как создать изящное меню с CSS-переходами для появления и исчезновения всплывающего меню, можно узнать по адресу tinyurl.com/kv9bg6v.

Если вы не хотите делать меню самостоятельно или ограничены во времени, то можете воспользоваться бесплатным мастером, реализованным в виде веб-страницы, которая автоматически создает необходимый код HTML и CSS (purecssmenu.com).

Подход с применением исключительно средств каскадных таблиц стилей имеет один недостаток: когда посетитель убирает с кнопки-ссылки указатель мыши, подменю исчезает мгновенно, без задержки — остается надеяться, что у посетителей сайта достаточно хорошая реакция. Есть и другой подход: используйте CSS-код для форматирования кнопок-ссылок и JavaScript-код для управления работой подменю. Пример такого использования JavaScript-кода находится по адресу tinyurl.com/qhuupeb.

2. Добавьте в стиль ссылок свойство `display: block`.

Ссылки — это строчные элементы, и значения ширины (как верхние и нижние отступы и поля) к ним неприменимы. Преобразование ссылок в блочные элементы позволит установить точную ширину кнопок и добавить требуемые промежутки для ссылок:

```
.nav a { display: block; }
```

3. Отформатируйте ссылки.

Измените цвет фона, добавьте границы и т. д.

4. Укажите ширину.

Если необходимо, чтобы все кнопки панели управления имели одинаковую ширину, укажите ширину элемента `a`. Точное значение ширины зависит от максимальной длины текста на каждой из кнопок. Очевидно, что для ссылки «Миссия корпорации» нужна кнопка большей ширины. Если вы хотите, чтобы каждая кнопка имела ширину находящегося внутри нее текста, не указывайте значение ширины. Хотя можно добавить отступы слева/справа и таким образом задать для текста немного пространства, избавившись от тесноты.

СОВЕТ

Чтобы расположить текст ссылок по центру кнопок, добавьте в стиль ссылок свойство `text-align: center;`.

5. Добавьте свойство `overflow: hidden` в стиль элемента `ul`.

В случае если для элемента `ul` заданы границы, цвет фона или изображение, вам придется «удержать плавающий элемент», то есть плавающие пункты

списка внутри элемента `ul` будут высываться из-под нижней части списка (и в таком случае находиться за пределами границ и фонового цвета элемента `ul`).

```
.nav {  
    overflow: hidden;  
}
```

Ниже представлен код стилей, требуемых для создания панели навигации, изображенной на рис. 9.7. Обратите внимание, что кнопки имеют одинаковую ширину и текст в них центрирован.

```
.nav {  
    margin: 0px;  
    padding: 0px;  
    list-style: none;  
    border-bottom: 3px solid rgb(204,204,204);  
    overflow: hidden;  
}  
  
.nav li {  
    float: left;  
}  
  
.nav a {  
    width: 12em;  
    display: block;  
    border: 3px solid rgb(204,204,204);  
    border-bottom: none;  
    border-radius: 5px 5px 0 0;  
    padding: 10px;  
    margin-right: 5px;  
    background-color: rgb(95,95,95);  
    background-image: -webkit-linear-gradient(rgb(175,175,175), rgb(95,95,95));  
    background-image: -moz-linear-gradient(rgb(175,175,175), rgb(95,95,95));  
    background-image: -o-linear-gradient(rgb(175,175,175), rgb(95,95,95));  
    background-image: linear-gradient(rgb(175,175,175), rgb(95,95,95));  
    text-decoration: none;  
    color: white;  
    text-align: center;  
    font-family: Arial, Helvetica, sans-serif;  
    font-weight: bold;  
}
```



Heading for this page

Рис. 9.7. Плавающие элементы списка позволяют создавать кнопки панели навигации одинаковой ширины. Вы можете использовать CSS-код, приведенный выше

Использование ролловеров

Очень давно, когда еще не было языка CSS, для смены одной графической ссылки на другую при наведении на нее указателя мыши приходилось привлекать средства JavaScript. Теперь с помощью каскадных таблиц стилей вы можете достичь подобного эффекта посредством псевдокласса `:hover` и фонового изображения. Однако с этим методом есть одна проблема: если браузером не загружено сменное изображение, оно не появится, пока на данную ссылку не будет наведен указатель мыши, — для посетителя заметна задержка этого сменного изображения на время загрузки браузером. Задержка произойдет всего один раз, во время первого наведения указателя мыши посетителем на ссылку, но все-таки ожидание загрузки графики — это издержки XX века.

Метод с применением JavaScript-сценария поможет избежать этой проблемы благодаря возможности *предварительной* загрузки сменных изображений задолго до того, как они понадобятся. В CSS нет таких средств, поэтому придется задействовать другой хитрый трюк, называемый *CSS-спрайтом*, который использует единственное изображение для представления различных состояний одной навигационной кнопки.

ПРИМЕЧАНИЕ

Эволюционировавший метод CSS-спрайтов в настоящее время широко используется такими компаниями, как Yahoo! и Google, причем не только для ролловеров, но и для оптимизации скорости загрузки сайтов. Вы можете прочитать об этом больше на странице tinyurl.com/nebu3km. По адресу tinyurl.com/oymwld8 находится список инструментов, примеров использования и полезных приложений для работы со спрайтами.

Рассмотрим, как реализуется метод.

1. В программе редактирования изображений создайте один рисунок с различными вариантами отображения кнопки.

Нужно создать изображения, представляющие собой различные состояния кнопки: обычное состояние, состояние при наведении указателя мыши и при необходимости состояние нажатой кнопки. Далее поместите изображения вертикально одно над другим. У нас будет два состояния кнопки: изображение в обычном состоянии должно располагаться сверху, а изображение-ролlover при наведении указателя мыши — снизу.

ПРИМЕЧАНИЕ

Существует множество онлайн-инструментов для создания CSS-спрайтов, например SpritePad (tinyurl.com/6ut2p7j) или Sprite Cow (spritecow.com).

2. Измерьте расстояние от верхнего края получившегося комбинированного изображения до верхнего края каждого следующего изображения.

На рис. 9.8, *вверху*, верхняя граница сменного изображения-роллера имеет смещение от верхнего края общего изображения 39 пикселей.

3. Создайте стиль для ссылки в обычном ненажатом состоянии. Поместите изображение в качестве фонового в левый верхний угол формируемой ссылки (см. рис. 9.8, *посередине*).

Стиль может выглядеть следующим образом:

```
a { background: #E7E7E7 url(images/pixy.png) no-repeat left top; }
```

4. Создайте стиль :hover.

Примените свойство background-position, чтобы сместить изображение *вверх*. Таким образом, первое изображение исчезнет, а второе, нижнее изображение-ролlover окажется видимым (см. рис. 9.8, *внизу*).

```
a:hover { background-position: 0 -39px; }
```

Такая методика, помимо предотвращения длительной задержки в загрузке дополнительного изображения-роллера, обеспечит хранение всех изображений кнопки, отражающих ее состояния, в единственном файле.

ПРИМЕЧАНИЕ

Некоторые сайты довели использование этого метода до крайности. Ресурсы Yahoo!, Amazon и Google (и многие другие) часто собирают вместе в одном файле множество маленьких изображений и показывают лишь часть файла, содержащую необходимую кнопку. Вы можете посмотреть пример спрайта с сайта Amazon по адресу tinyurl.com/o9p9oob. Сайты с миллионами посещений в день активно используют сложные спрайты, поскольку веб-серверу проще и быстрее отправить один (хотя и больший по размеру) файл изображения, чем десяток меньших по размеру файлов.

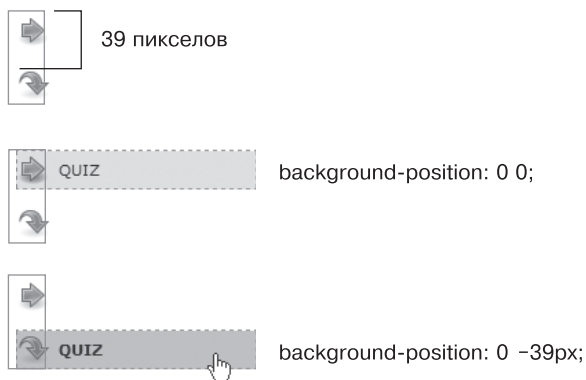


Рис. 9.8. Используя метод CSS-спрайтов, можно избежать задержки во время загрузки браузером рисунка-роллера в первый раз, поместив все изображения различных состояний кнопки-ссылки в единственный графический файл

Форматирование ссылок определенного типа

Веб-дизайнеры используют ссылки на все что угодно: на другие веб-страницы своих сайтов, веб-страницы посторонних сайтов, PDF-файлы, документы Word, ZIP-архивы и многое другое. Чтобы помочь ориентироваться посетителям сайта, вы можете подсказать им, к какому объекту ведет ссылка, прежде чем они щелкнут на ней. Расширенные селекторы представляют отличный способ решения данной проблемы.

Ссылки на другие сайты

Вы можете легко создать стиль, определяющий ссылки на другие сайты, используя селектор атрибута. Как рассказывалось в главе 3, селекторы атрибутов позволяют форматировать HTML-элементы, у которых есть определенный атрибут, например изображение `img` с атрибутом `alt` со значением `Наша компания`. Вы также можете форматировать элементы, атрибуты которых *содержат* определенные значения. Любая ссылка, которая ведет за пределы вашего сайта, должна быть абсолютным URL-адресом, то есть должна начинаться с `http://` или `https://`, например `https://css-tricks.com`.

ПРИМЕЧАНИЕ

В настоящее время все крупные сайты используют протокол SSL (secure sockets layer — уровень защищенных сокетов), поэтому ссылки на такие сайты всегда начинаются со значения `https://`. На самом деле протокол SSL применяется практически повсеместно, поэтому вы, вероятно, чаще встречаете ссылки `https://`, чем `http://`.

Поскольку вам необходимо учесть ссылки как `http://`, так и `https://`, существует два выхода, каждый из которых обладает недостатком. Сначала вы могли бы заполнить поиск начала свойства `href` для `http` следующим образом:

```
a[href^='http']
```

Эта запись соответствует URL-адресу, который начинается как с `http://`, так и с `https://`. Тем не менее данный способ также выбирает относительные ссылки, которые указывают на адрес `http.html` или `http-information.html`. Если вы не планируете ссылаться на любые страницы своего сайта, которые начинаются с `http`, все будет работать прекрасно.

Другой способ заключается в применении селектора для поиска значения `://` среди URL-адресов, например:

```
a[href*='://']
```

Эта запись соответствует ссылкам `http://` и `https://`, а поскольку значение `://` не встречается больше нигде в свойстве `href`, данный способ должен работать для всех внешних ссылок.

Вы могли бы форматировать их любым образом, но распространенный способ — добавить рядом со ссылкой маленькое изображение — значок, который указывает на то, что это внешняя ссылка. Этот метод представлен в практикуме в конце текущей главы.

Если вы решили использовать абсолютные ссылки на другие страницы *вашего* сайта, то, вероятно, не захотите применять тот же стиль для них. На этот случай *отрицающий* псевдокласс `:not()` легко выполняет эту функцию. С помощью приведенного ниже кода вы легко можете выделить абсолютные ссылки, которые *не* ведут на страницы вашего сайта:

```
a[href*='://']:not(a[href*='www.mysite.com']) {  
  background: url(images/globe.png) no-repeat center right;  
  padding-right: 15px;  
}
```


Ссылки на адреса электронной почты

Ссылки на адреса электронной почты — еще один особый вид ссылок. Обычно они выглядят как любые другие ссылки: имеют синий цвет и подчеркивание. Тем не менее они действуют не так, как другие ссылки. Переход по ссылке на адрес электронной почты запускает ассоциированный почтовый клиент на компьютере посетителей, что некоторых из них действительно отвлекает. Поэтому желательно как-то подсказать пользователю, что эта ссылка ведет на адрес электронной почты.

Для этого используется тот же базовый метод, который был описан применительно к внешним ссылкам выше. Поскольку все ссылки на почтовые адреса начинаются со значения `mailto:`, вы можете создать селектор наподобие следующего для стиля, форматирующего только определенный тип ссылок:

```
a[href^='mailto:']
```

Скоро вы увидите такой пример в действии.

Ссылки на определенные типы файлов

Некоторые ссылки указывают на файлы, а не на другие веб-страницы. Вы могли видеть ежегодные отчеты различных компаний в виде загружаемого PDF-файла или ZIP-архива файлов (как примеры для этой книги) на сайте. Ссылки на такие типы файлов, как правило, вынуждают браузер приступить к загрузке соответствующего файла на компьютер посетителя или, в случае PDF-файлов, запустить плагин (подключаемый модуль), который позволит просматривать документ прямо в браузере. Выбрав ссылку, посетитель может оказаться в шоке от того, что на самом деле началась загрузка файла размером 100 Мбайт!

Вы можете идентифицировать определенные типы файлов наподобие внешних ссылок или ссылок на адреса электронной почты. Но вместо того, чтобы пытаться найти какую-то конкретную информацию в начале URL-адреса ссылки, поищите ее в конце. Например, ссылка на PDF-документ может выглядеть так: ``, а ссылка на ZIP-архив так: ``. В каждом случае конкретный тип файла определяется расширением в конце URL-адреса: `.pdf` или `.zip`.

Каскадные таблицы стилей предоставляют селектор, который позволяет искать атрибуты, заканчивающиеся какой-либо конкретной информацией. Так, чтобы создать стиль ссылки на PDF-файлы, используйте следующий селектор:

```
a[href$='.pdf']
```

Сочетание `$=` означает «заканчивается на», и, соответственно, данный селектор позволяет выбрать все ссылки, значение атрибута `href` которых заканчивается на `.pdf`. Вы можете создать аналогичные стили и для других типов файлов:

```
a[href$='.zip'] /* zip-архив */  
a[href$='.doc'] /* документ Word */
```

Далее вы увидите примеры этого метода.

Практикум: форматирование ссылок

В этом практикуме мы потренируемся в форматировании ссылок разнообразными способами, в том числе путем добавления изображений-ролловеров и фоновых рисунков.

Чтобы начать обучение, вы должны иметь в распоряжении файлы с учебным материалом. Для этого нужно загрузить файлы для выполнения заданий практикума, расположенные по адресу github.com/mrightman/css_4e. Перейдите по ссылке и загрузите ZIP-архив с файлами (нажав кнопку **Download ZIP** в правом нижнем углу страницы). Файлы текущего практикума находятся в папке с именем **09**.

Простейшее форматирование ссылок

Начнем с простого форматирования ссылок.

1. Запустите браузер и откройте файл веб-страницы `links.html` из папки **09\links**. Эта страница содержит множество ссылок (рис. 9.9), которые указывают на другие веб-страницы текущего или иных сайтов, а также ведут на адреса электронной почты. Сначала изменим цвет ссылок данной страницы.
2. Откройте файл `links.html` в редакторе HTML-кода и поместите указатель между открывающим и закрывающим тегами элемента `<style>`. Эта веб-страница уже имеет внешнюю таблицу стилей, придающую ей какое-то базовое форматирование и содержащую элемент `style` внутренней таблицы.

ПРИМЕЧАНИЕ

Вы разместите стили для этого упражнения во внутренней таблице, чтобы было легче писать код и просматривать страницу. Когда все будет готово, рекомендуется переместить стили во внешнюю таблицу стилей.

3. Добавьте во внутреннюю таблицу новый стиль:

```
<style>
a {
  color: #207EBF;
}
</style>
```

Он будет применяться ко всем элементам `a` на странице. С него хорошо начинать, поскольку он устанавливает общий внешний вид для ссылок на странице. Вы добавите больше стилей, которые позволят вам форматировать ссылки в определенных областях страницы. Теперь пришло время удалить это надоевшее подчеркивание под ссылкой.

4. Добавьте в только что созданный стиль свойство `text-decoration: none;`.

Он убирает подчеркивание, но в то же время ссылка на веб-странице становится менее заметной. Не забывайте, что ссылки всегда должны визуально выделяться на общем фоне, чтобы у посетителей сайта не возникало сомнений в том, что это именно ссылки и на них можно щелкнуть кнопкой мыши.



Рис. 9.9. Простейшая веб-страница со стандартным для браузеров форматированием — ссылки выделены синим цветом (или фиолетовым, если посетители ранее посещали страницы, на которые они ссылаются) и подчеркнуты

5. Добавьте в стиль `a` свойство `font-weight: bold;`.

Теперь ссылки отображаются полужирным шрифтом (для остального текста он также может быть установлен). Далее заменим подчеркивание на выделение другим способом, но сделаем это творчески, используя вместо свойства `text-decoration` границы.

6. Добавьте свойство `border`, чтобы стиль принял следующий вид:

```
a {
  color: #207EBF;
  text-decoration: none;
```

```
font-weight: bold;
border-bottom: 2px solid #F60;
}
```

Теперь ссылки выделены, а использование границ вместо обычного подчеркивания позволило изменить стиль, цвет и толщину линий. А сейчас вы измените вид посещенных ссылок.

7. Добавьте псевдокласс `:visited` для посещенных ссылок следующим образом:

```
a:visited {
  color: #6E97BF;
}
```

Стиль изменяет внешний вид посещенных ссылок на более светлый с серым оттенком — это искусный способ отвлечь внимание от уже посещенной страницы. Если вы просматриваете страницу, щелкните на одной из ссылок (попробуйте, например, одну из тех, что находятся в центре страницы), а затем вернитесь к странице `links.html`. Вы должны увидеть, что ссылка стала светлее. Но вы также заметите, что остается выделение полужирным и все еще присутствует оранжевое подчеркивание, которое вы назначили стилем в шаге 6. Это пример каскадности в действии (см. главу 5): стиль `a:visited` более специфичен, чем простой селектор, поэтому его свойство цвета переопределяет тот цвет, который был назначен стилем.

Сейчас добавим эффект ролловера, чтобы фоновый цвет ссылки изменялся при наведении на нее указателя мыши.

8. Добавьте в таблицу стилей псевдокласс `:hover`:

```
a:hover {
  color: #FFF;
  background-color: #6E97BF;
  border-bottom-color: #6E97BF;
}
```

Этот псевдокласс применяется только на время, пока указатель мыши находится поверх ссылки. Свойство интерактивности эффекта ролловера позволяет посетителям узнать, что выполняется какое-то действие (рис. 9.10).

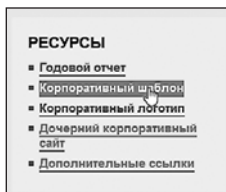


Рис. 9.10. С помощью нескольких стилей вы можете изменить внешний вид любой ссылки, а используя псевдокласс `:hover`, можно установить отдельный стиль, активируемый при нахождении указателя мыши над ссылкой

Добавление фонового изображения

Ссылка на адрес электронной почты в нижней части веб-страницы ничем не отличается от других ссылок на странице (рис. 9.11, *вверху*). Поскольку значение `mailto:`

указывает на адрес электронной почты, при щелчке на ней посетитель не перейдет на другую веб-страницу, а запустит почтовую программу. Чтобы обеспечить визуальное выделение этой ссылки, добавим небольшой значок почтового конверта.

1. Добавьте во внутреннюю таблицу стилей файла `links.html` селектор потомков:

```
a[href^="mailto:"] {
  color: #666666;
  border: none;
  background: url(images/email.gif) no-repeat left center;
}
```

Здесь используется расширенный селектор атрибута, выбирающий любые ссылки, начинающиеся со значения `mailto:` (то есть он выбирает ссылки на адреса электронной почты). Ссылке назначен серый цвет, а код `border: none` убирает подчеркивание, определенное в шаге 6. Свойство `background` добавляет фоновое изображение с левой стороны, а параметр `no-repeat` обеспечивает однократное отображение рисунка. Здесь трудность состоит в том, что фоновое изображение (значок конверта) располагается на заднем плане позади текста ссылки, и он становится трудночитаемым (см. рис. 9.11, *посередине*).

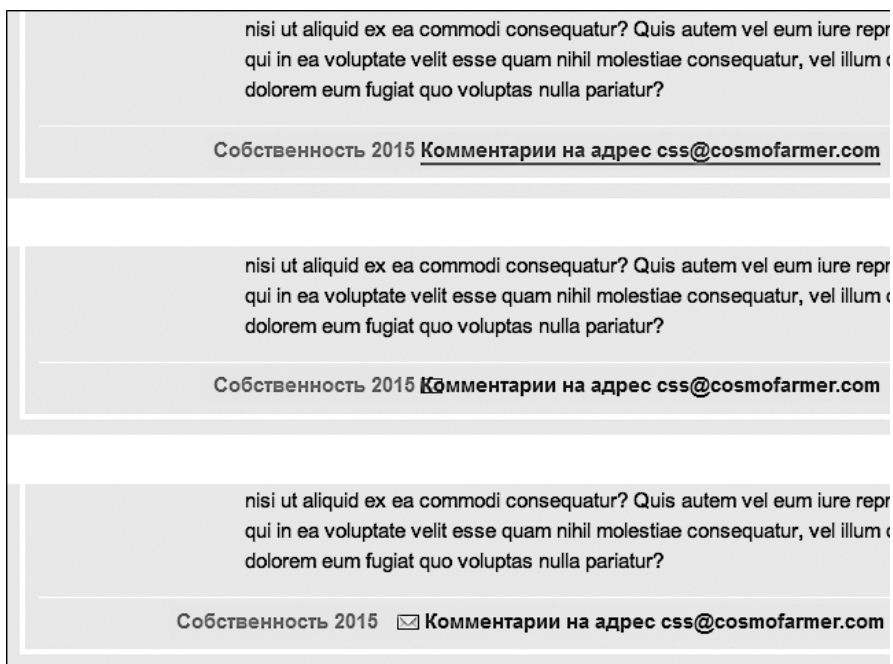


Рис. 9.11. Несколько тонких изменений сделают назначение ссылки понятным: простая ссылка (*вверху*) превращается в явную, узнаваемую владельцами электронной почты (*внизу*)

2. Добавьте в только что созданный стиль атрибута левый отступ размером 20 пикселей:

```
padding-left: 20px;
```

Помните, что отступ добавляет промежуток между содержимым и границей элемента, поэтому установка небольшого отступа слева смещает текст ссылки на 20 пикселей, оставляя фоновое изображение на месте. И последний штрих: нужно немного отодвинуть всю ссылку от упоминания об авторском праве.

3. Добавьте в стиль поле слева размером 10 пикселей. Должен получиться следующий стиль:

```
a[href^="mailto:"] {
  color: #666666;
  border: none;
  background: url(images/email.gif) no-repeat left center;
  padding-left: 20px;
  margin-left: 10px;
}
```

Эта небольшая поправка обеспечивает визуальное отделение изображения-значка почтовой ссылки от упоминания об авторском праве (см. рис. 9.11, *внизу*). Таким образом, ссылка воспринимается посетителем как сочетание значка с текстом.

Выделение внешних ссылок

Иногда требуется визуально показать, что ссылка ведет на внешний сайт. Этим можно сказать посетителям, что дополнительная информация по теме находится где-то во Всемирной паутине, на другом сайте, или предупредить, что выбор этой ссылки приведет к переходу на другой сайт. Кроме того, вы, возможно, захотите идентифицировать ссылки, указывающие на загружаемые файлы или другие документы, не являющиеся веб-страницами.

На веб-странице, над которой вы работаете, боковая панель, размещенная справа, содержит различные типы ссылок. Вы их выделите значками, используя отдельный значок для каждого типа. Для начала вы настроите базовый стиль, который применяется ко всем ссылкам.

1. Добавьте во внутреннюю таблицу стилей веб-страницы `links.html` следующий код:

```
.resources a {
  border-bottom: none;
}
```

Поскольку все ссылки, которые нужно отформатировать, находятся внутри элемента `div` с классом `resources`, селектор потомков `.resources a` воздействует только на эти ссылки. Стиль избавляет от подчеркивания, которое было добавлено в общем стиле ранее.

Далее вы добавите значок, размещаемый рядом с внешними ссылками.

2. Добавьте еще один стиль в конец внутренней таблицы стилей веб-страницы `links.html`:

```
.resources a[href*='://'] {
  background: url(images/globe.png) no-repeat right top;
}
```

В этом стиле селектора потомков используется расширенный селектор атрибута. В основном он воздействует на любую ссылку, которая содержит символы `://` (но только на ту, которая находится внутри элемента с классом `resources`). Как и стиль ссылки на адрес электронной почты, который вы создали ранее, этот стиль добавляет фоновое изображение. Он помещает рисунок в правую часть ссылки.

Тем не менее у этого стиля есть проблема, аналогичная проблеме стиля ссылки на адрес электронной почты, — изображение появляется позади текста ссылки. Решение такое же — нужно добавить отступы, чтобы переместить изображение в нужное место. В этом случае, однако, вместо левого отступа вы добавите правый (поскольку значок появляется в правой части ссылки). Кроме того, так как все ссылки в разделе ресурсов будут иметь значки схожих размеров, вы можете уменьшить объем кода, добавив отступы с помощью стиля `.resources a`, созданного на шаге 1.

3. Отредактируйте стиль `.resources a` так, чтобы он выглядел следующим образом:

```
.resources a {  
    border-bottom: none;  
    padding-right: 22px;  
}
```

Если вы сохраните страницу и просмотрите ее в браузере, то увидите маленькие значки с земным шаром в правой части двух нижних ссылок боковой панели. Теперь отформатируем другие ссылки.

4. Добавьте еще три стиля во внутреннюю таблицу стилей:

```
.resources a[href$='.pdf'] {  
    background: url(images/acrobat.png) no-repeat right top;  
}  
.resources a[href$='.zip'] {  
    background: url(images/zip.png) no-repeat right top;  
}  
.resources a[href$='.doc'] {  
    background: url(images/word.png) no-repeat right top;  
}
```

Эти три стиля проверяют, каким значением заканчивается атрибут `href`, идентифицируют ссылки как файлы Adobe Reader (`.pdf`), ZIP-архивы (`.zip`) или документы Word (`.doc`) и назначают различные значки в каждом конкретном случае.

5. Наконец, добавьте состояние наведения указателя мыши для ссылок на ресурсы:

```
.resources a:hover {  
    color: #000;  
    background-color: rgba(255,255,255,.8);  
}
```

Этот стиль изменяет цвет текста и добавляет цвет фона (рис. 9.12).

Окончательную версию этого урока вы найдете в файле `09_finished/links/links.html`.

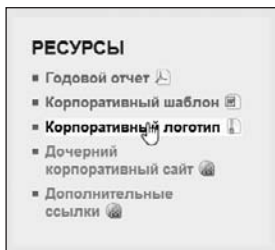


Рис. 9.12. Используя расширенные селекторы атрибутов, вы можете легко идентифицировать и форматировать различные типы ссылок — внешние, ссылки на PDF-файлы, текстовые документы Word и архивы

Практикум: создание панели навигации

На этом практическом занятии мы превратим простой список ссылок во впечатляющую панель навигации с эффектом ролловера и выделением нажатой кнопки текущего раздела сайта.

1. Откройте файл веб-страницы `nav_bar.html` из папки `09\nav_bar` в редакторе HTML-кода.

Как видите, в этом файле содержится совсем короткий исходный код. Здесь присутствует внутренняя таблица со сбросом стандартных стилей CSS и одним правилом, определяющим простейшее форматирование тела веб-страницы. HTML-код создает маркированный список, состоящий из шести ссылок. Веб-страница имеет вид, показанный на рис. 9.13, 1. Первым шагом будет добавление HTML-кода, обеспечивающего целенаправленное форматирование ссылок списка средствами каскадных таблиц стилей.

2. Найдите открывающий тег `` и добавьте в него код `class="mainNav"` следующим образом:

```
<ul class="mainNav">
```

Атрибут `class` назначает этот список основной областью навигации. Мы будем использовать его для создания селекторов потомков, избирательно форматизирующих только ссылки этого списка (а не все ссылки веб-страницы).

3. Добавьте во внутреннюю таблицу после стиля `body` следующий код:

```
.mainNav {
  margin: 0;
  padding: 0;
  list-style: none;
}
```

Этот стиль будет применен лишь к элементу с классом `mainNav`, в данном случае элементу `ul`. Он удаляет отступ и маркеры, которые браузер добавляет в маркированные списки, как показано на рис. 9.13, 2. Теперь приступим к форматированию ссылок.



Рис. 9.13. Вам может показаться, что преобразование простого списка в сложную панель навигации требует выполнения объемной работы, но на самом деле нужно лишь создать несколько стилей

4. Добавьте селектор потомков для форматирования ссылок списка:

```
.mainNav a {
  color: #000;
  font-size: 11px;
  text-transform: uppercase;
  text-decoration: none;
}
```

Этот стиль определяет базовое форматирование текста ссылок: устанавливает цвет и размер шрифта, определяет все буквы как прописные и удаляет линию подчеркивания (см. рис. 9.13, 3). Сейчас придадим ссылкам вид кнопок.

5. Добавьте в стиль `.mainNav a` свойства `border` и `padding`:

```
border: 1px dashed #999;
padding: 7px 5px;
```

Теперь при просмотре веб-страницы в браузере вы заметите пару проблем (см. рис. 9.13, 4): границы накладываются друг на друга, а ссылки имеют различную ширину. Это происходит потому, что элемент `a` является строчным и ширина ссылки равна длине ее текста. Кроме того, верхний и нижний отступы не обеспечивают увеличения высоты строчным элементам, поэтому границы накладываются друг на друга (о строчных элементах читайте в разделе «Управление размерами полей и отступов» главы 7). Решить эти проблемы со ссылками можно путем изменения их способа отображения браузером.

6. Добавьте свойство `display: block`; в стиль `.mainNav a`.

Мы изменяем параметры отображения элемента `a` таким образом, чтобы он обрабатывался браузером как блочный элемент (например, абзац текста), с расположением ссылок в виде списка, в точности одна над другой. Осталась

единственная проблема — ссылки растянуты по всей ширине окна браузера — слишком много для кнопок. Исправим ситуацию, ограничив ширину элемента `ul` соответствующим стилем.

7. Во внутренней таблице найдите стиль `.mainNav` и добавьте в него свойство `width: 175px;`.

```
.mainNav {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
  width: 175px;  
}
```

При ширине списка 175 пикселей ссылки по-прежнему слишком широки, хотя и ограничиваются шириной элемента-контейнера (`ul`). В большинстве случаев список ссылок заключен в какой-либо элемент разметки (например, боковую панель), для которого уже определена ширина, поэтому можете пропустить этот шаг (о том, как создавать боковые панели, читайте в части III).

Сейчас перейдем к самому интересному.

8. Добавьте в стиль `.mainNav a` свойства фона следующим образом:

```
.mainNav a {  
  color: #000;  
  font-size: 11px;  
  text-transform: uppercase;  
  text-decoration: none;  
  border: 1px dashed #999;  
  padding: 7px 5px;  
  display: block;  
  background-color: #E7E7E7;  
  background-image: url(images/nav.png);  
  background-repeat: no-repeat;  
  background-position: 0 2px;  
}
```

Эти строки кода изменяют цвет фона ссылок на серый и устанавливают одиночное изображение с левой стороны каждой кнопки (см. рис. 9.13, 5). Здесь тоже нужно кое-что исправить: текст ссылки накладывается на значок, а линии границ между кнопками имеют толщину, равную 2 пикселям (теоретически границы имеют толщину 1 пиксел, но сливающиеся воедино линии соседних ссылок образуют линию толщиной 2 пикселя).

ПРИМЕЧАНИЕ

Используя сокращенную запись свойства `background`, можно изменить код на шаге 8 на следующий: `background: #E7E7E7 url(images/nav.png) no-repeat 0 2px;`.

9. Удалите верхнюю границу и измените отступ в стиле `.mainNav a`, чтобы его код выглядел следующим образом:

```
.mainNav a {  
  color: #000;
```

```
font-size: 11px;
text-transform: uppercase;
text-decoration: none;
border: 1px dashed #999;
border-bottom: none;
padding: 7px 5px 7px 30px;
display: block;
background-color: #E7E7E7;
background-image: url(images/nav.png);
background-repeat: no-repeat;
background-position: 0 2px;
}
```

Выглядит неплохо: текст каждой ссылки отделен от значка, границы выделяются. Однако нижняя граница последней ссылки исчезла. Существует несколько способов разобраться с этим. Один состоит в том, чтобы применить нижнюю границу к элементу `ul`, содержащему список ссылок (поскольку этот элемент не имеет отступов, нет промежутка, отделяющего верх элемента `ul` от верхней стороны этой первой ссылки). Второй способ заключается в использовании псевдокласса `:last-of-type`. Вам нужно выделить ссылку, которая находится внутри последнего элемента списка на панели навигации, и добавить к ней нижнюю границу.

10. Между стилями `.mainNav` и `.mainNav a` добавьте следующий стиль:

```
.mainNav li:last-of-type a {
  border-bottom: 1px dashed #999;
}
```

Эти стили селектора потомка форматируют ссылку (`a`), расположенную внутри последнего элемента списка (`li:last-of-type`) или списка навигации (`.mainNav`).

Вот и все, мы создали простейшую панель навигации с применением границ, отступов, фоновой окраски и изображений (см. рис. 9.13, б).

Добавление ролловеров и выделение текущего раздела сайта

Пришло время усовершенствовать панель навигации и придать ей некоторую интерактивность. Во-первых, обеспечим кнопкам панели навигации эффект ролловера. Они должны изменять свой вид, акцентируя внимание посетителя на том, какую кнопку он собирается нажать.

Неплохо было бы оповещать посетителя, в каком разделе (на какой странице) сайта он находится. Мы можем обеспечить созданной панели навигации автоматическую интерактивность. Она будет изменять свой стиль в соответствии с выбранным разделом страницы. Звучит совсем просто, но в действительности это потребует некоторой разметки и настроек, как вы увидите в последующих шагах.

Эффект ролловера реализовать легко, но начнем по порядку.

1. Добавьте в конце таблицы стилей файла `nav_bar.html` следующий стиль:

```
.mainNav a:hover {
  font-weight: bold;
```

```
background-color: #B2F511;
background-position: 3px 50%;
}
```

Он определяет внешний вид ссылки-кнопки в состоянии наведения указателя мыши. Стиль изменяет шрифт текста кнопки с обычного на полужирный, а также цвет фона на ярко-зеленый. Кроме того, он использует метод CSS-спрайтов, упоминавшийся ранее. То же самое изображение применяется на шаге 8 в прошлом разделе — оно в действительности содержит три различных значка (рис. 9.14). В этом случае изображение центрируется внутри кнопки, отображая средний значок файла.

Теперь при появлении указателя мыши поверх любой кнопки она моментально изменяет свой вид (откройте веб-страницу в своем браузере).

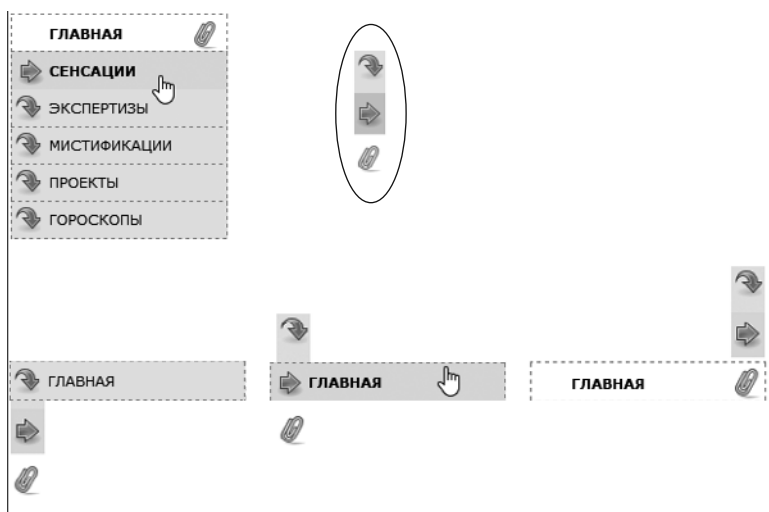


Рис. 9.14. С помощью несложного CSS-кода можно создать интерактивный эффект ролловера для кнопок панели навигации, а также автоматически подсветить раздел сайта текущей страницы

Теперь сделаем панель навигации более информативной, выделив кнопку, соответствующую текущему разделу, страницу которого открыл посетитель сайта. Чтобы сделать это, нам потребуется идентифицировать в HTML-коде панели навигации две вещи: раздел, к которому принадлежит страница и разделы, на которые указывает каждая ссылка. В нашем примере предположим, что открыта домашняя страница.

ПРИМЕЧАНИЕ

Другим вариантом будет создание класса, который изменяет внешний вид ссылки, представляющей раздел страницы. Для веб-страницы гороскопа ссылка панели навигации может выглядеть следующим образом: `Гороскопы`.

- Найдите открывающий тег `<body>` и добавьте в него класс `class="home"`:
`<body class="home">`

Теперь, когда мы знаем, какому разделу сайта принадлежит текущая открытая страница, можно использовать селектор потомков для создания конкретного стиля, который будет применен к веб-странице раздела «Сенсации». Далее мы должны маркировать разделы, на которые указывает каждая ссылка-кнопка, что достигается путем добавления классов.

3. Найдите в HTML-коде панели навигации ссылку «Главная» и добавьте в нее класс `class="homeLink"`, чтобы код элемента выглядел следующим образом:

```
<a href="/index.html" class="homeLink">Главная</a>
```

Этот класс определяет ссылку, предоставляя возможность создания стиля, который будет применен только к ней.

Естественно, нам потребуется добавить классы для всех ссылок панели навигации.

4. Повторите шаг 3 для каждой из ссылок, используя следующие классы: `featureLink`, `expertLink`, `quizLink`, `projectLink` и `horoscopeLink`.

С HTML-кодом закончили. Настало время создать CSS-код. Теперь у нас есть маркированные страница и ссылка, и мы можем создать селектор потомков, выделяющий кнопку-ссылку «Главная».

5. Добавьте в таблицу стилей веб-страницы следующий код:

```
.home .homeLink {  
    background-color: #FFFFFF;  
    background-position: 97% 100%;  
    padding-right: 15px;  
    padding-left: 30px;  
    font-weight: bold;  
}
```

Мы уже рассматривали эти свойства ранее. Снова вы используете метод CSS-спрайтов для управления позицией фонового изображения. На этот раз изображение смещается на 97 % вправо, а его нижняя часть помещается в нижнюю часть кнопки. Другими словами, значок будет отображаться внизу изображения (см. рис. 9.14). О том, как использовать процентные значения при работе с фоновыми изображениями, мы говорили в главе 8.

В данном случае наибольший интерес представляет селектор `.home .homeLink`. Это очень специфичный селектор, применяемый только к ссылке с классом `homeLink`, которая *также* заключена внутрь элемента `body` с классом `home`. Если вы измените класс страницы, например, на `quiz`, с кнопки-ссылки раздела «Главная» исчезнет выделение.

Просмотрите веб-страницу в браузере, чтобы увидеть результат: теперь ссылка «Главная» имеет белый фон и значок скрепки. Чтобы проделать все то же самое с остальными ссылками, вы должны немного *расширить* селектор.

6. Отредактируйте селектор только что созданного стиля:

```
.home .homeLink,  
.feature .featureLink,  
.expert .expertLink,
```

```
.quiz .quizLink,  
.project .projectLink,  
.horoscope .horoscopeLink{  
    background-color: #FFFFFF;  
    background-position: 97% 100%;  
    padding-right: 15px;  
    padding-left: 30px;  
    font-weight: bold;  
}
```

Конечно, получился довольно объемный CSS-код. Этот стиль теперь адресует-ся ко всем ссылкам панели навигации, но только при выполнении определенных условий. Если вы в дальнейшем измените класс элемента `body`, например, на `quiz`, то ссылка раздела «Мистификации» приобретет такое же форматирование, каким была выделена ссылка раздела «Сенсации».

Теперь пришло время проверить результаты работы.

ПРИМЕЧАНИЕ

Этот длинный селектор является примером группового селектора, который мы обсуждали в главе 3.

7. Измените атрибут `class` элемента `body`, присвоив ему значение `feature`, следующим образом:

```
<body class="feature">
```

Просмотрев веб-страницу, вы увидите, что ссылка «Сенсации» теперь выделена белым цветом фона и значком скрепки. Весь секрет в том, что нужно изменить атрибут `class` элемента `body`, чтобы указать, к какому разделу сайта принадлежит страница. Например, для страницы гороскопа измените атрибут класса `body` на `class="horoscope"`, а для главной страницы используйте атрибут `class="home"` (см. рис. 9.14).

ПРИМЕЧАНИЕ

Готовы продолжить форматирование? Попытайтесь добавить эффект ролловера, чтобы закончить стиль, созданный на шаге 6 (используйте псевдокласс `:hover` в качестве компонента селектора: `.quiz .quizLink:hover`). Попробуйте также добавить другое изображение для ссылки главной страницы (можете использовать файл `home.png` из папки `images`).

Файл с финальным вариантом панели навигации `nav_bar_vertical.html` можно найти в папке `09_finished\nav_bar`.

От вертикальной к горизонтальной панели навигации

Предположим, вы хотите создать горизонтальную панель навигации в верхней части веб-страницы. Никаких проблем — большую часть работы мы уже выполнили. Чтобы расположить кнопки горизонтально в одну строку, необходимо немного изменить уже созданную веб-страницу (мы будем дорабатывать файл `nav_bar.html`, поэтому, если вы хотите сохранить вертикальную панель навигации, прежде чем продолжить, сделайте копию файла).

1. Убедитесь в том, что вы выполнили все шаги по созданию вертикальной панели навигации и откройте файл `nav_bar.html` в редакторе HTML-кода.

Сейчас вы увидите, как легко можно изменить ориентацию панели. Сначала подчистим кое-что из того, что мы уже сделали. Вам нужно удалить свойство `width`, которое вы установили для элемента `ul` ранее. Оно препятствует тому, чтобы навигационные кнопки охватывали всю длину страницы.

2. Найдите стиль `.mainNav` и удалите из него свойство `width: 175px;`.

Теперь приведем секрет преобразования вертикальной панели навигации в горизонтальную.

3. Добавьте в таблицу новый стиль (сразу после `.mainNav`):

```
.mainNav li {  
  float: left;  
  width: 14em;  
}
```

Стиль применяется к элементу `li` (представляющему собой элементы списка, каждый из которых содержит свою ссылку). Первая команда устанавливает для элемента выравнивание по левому краю. Таким образом, каждый последующий элемент `li` располагается с правой стороны предыдущего (такой же эффект вы могли наблюдать в практикуме по созданию фотогалереи в главе 8). Кроме того, устанавливая ширину элемента `li`, мы одновременно определяем ширину для всех кнопок панели навигации. В данном случае значение `14em` обеспечивает достаточный размер, чтобы вместить самый длинный текст названия ссылки. Если будет необходимо увеличить длину, вы должны будете увеличить это значение.

При просмотре веб-страницы в браузере вы увидите, что с панелью навигации в основном все в порядке. Требуются только небольшие корректировки (рис. 9.15, 1). Нижняя граница под кнопками отсутствует. Кроме того, поскольку левая граница кнопки расположена рядом с границей ссылки, их левые и правые границы сливаются между собой, образуя линии толщиной 2 пикселя. Сейчас мы решим эти проблемы.

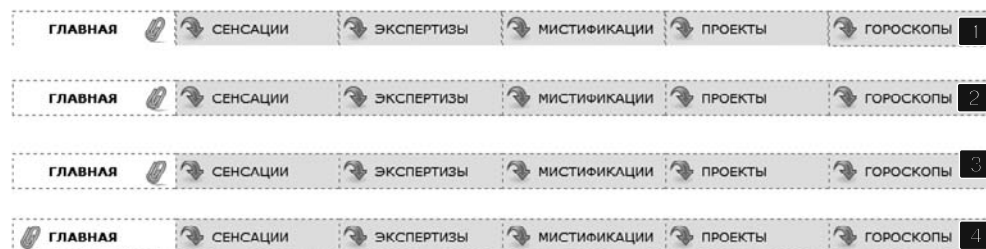


Рис. 9.15. Преобразование вертикальной панели в более компактную с горизонтальным расположением кнопок требует выполнения всего нескольких действий; гораздо больше усилий нужно приложить для настройки стилей, определяющих параметры границ и позиционирование фоновое изображения

4. В стиле `.mainNav` а измените свойство `border-bottom: none;` на `border-right: none::` Это действие приводит к удалению левых границ кнопок, чтобы они не удваивались, и в то же время к нижнему краю добавляется граница. Однако теперь граница с правой стороны последней кнопки на панели навигации исчезла (см. рис. 9.15, 2). Чтобы исправить эту проблему, вы можете сделать то же, что делали в предыдущей части практикума, — использовать селектор `:last-of-type`.
5. Измените свойство `border-bottom` стиля `.mainNav li:last-of-type` а на `border-right`:

```
.mainNav li:last-of-type a {  
    border-right: 1px dashed #999;  
}
```

Это изменение добавит правую границу, но только к последней ссылке на панели навигации (см. рис. 9.15, 3). Наконец, скрепки, выровненные по правому краю выбранной кнопки, выглядят странно. Сейчас вы измените их положение, разместив у левого края кнопки.

6. Найдите стиль выбранной кнопки, созданный в шаге 6 (стиль с длинным селектором). Измените координаты его свойства `background` с `97% 100% на 3px 100%`. Теперь стиль должен выглядеть следующим образом:

```
.home .homeLink,  
.feature .featureLink,  
.expert .expertLink,  
.quiz .quizLink,  
.project .projectLink,  
.horoscope .horoscopeLink {  
    background-color: #FFFFFF;  
    background-position: 3px 100%;  
    padding-right: 15px;  
    padding-left: 30px;  
    font-weight: bold;  
}
```

Просмотрите веб-страницу в браузере, и вы обнаружите, что горизонтальная панель навигации полностью работоспособна и замечательно выглядит (см. рис. 9.15, 4).

Окончательную версию этого урока вы найдете в файле `09_finished/links/links.html`.

ПРИМЕЧАНИЕ

Возможно, вы захотите центрировать текст кнопок-ссылок панели навигации. Для этого нужно сделать две вещи: добавить в стиль `.mainNav` а свойство `text-align: center;` и изменить свойство `left-padding` этого же стиля, чтобы текст был расположен точно по центру кнопок.
