Форма Google

В этой главе мы узнаем, как использовать форму для поиска на своем сайте. Для эффективного использования форм поиска вам необходимо знать основы HTML, поэтому сейчас поговорим об этих самых основах. Знания HTML вам также понадобятся для написания собственных Google-приложений. В этой книге мы также поговорим о том, как создать простое Google-приложение без использования Google API, который мы не будем рассматривать в этой главе. Кроме того, без основ HTML вы даже не сможете создать форму Google, поэтому эту главу вам придется прочитать (или хотя бы просмотреть) полностью.

6.1. Основные теги HTML

Цель данной главы — напомнить вам язык HTML, чтобы у вас во время работы с книгой не возникало потребности обращаться к какому-нибудь другому источнику. Язык HTML (Hyper Text Markup Language — язык разметки гипертекстовой информации) предназначен для разметки Web-страниц. Все браузеры понимают HTML-документы: возможно, вы и найдете какой-нибудь старый браузер, который не будет понимать последние стандарты HTML — HTML4, но основные теги, о которых мы поговорим в этой главе, обязан понимать каждый браузер — иначе это не браузер, а просто telnet-клиент.

Тег обязательно начинается символом "меньше" (<), и заканчивается символом "больше" (>). Большая часть тегов (но не все) подразумевает использование закрывающего тега:

<HTML> - Ter,

</HTML> — закрывающий (конечный) тег.

Вы можете указать атрибуты тега, если он подразумевает их использование, например, <P align="center">. Атрибуты указываются с использованием синтаксиса имя_атрибута=значение. Значение желательно брать в кавычки, хотя многие браузеры лояльны и позволяют указывать значение без кавычек.

6.1.1. Структура документа

Документ HTML имеет такую структуру:

```
<HTML>
<HEAD>
```

```
<TITLE>Это мы увидим в заголовке браузера</TITLE>
<HEAD>
<BODY>
<h1>Информация</h1>
</BODY>
</HTML>
```

Проведем небольшой эксперимент. Создайте HTML-документ, в качестве содержимого которого используйте вышеприведенный листинг. Напомню, для такого рода задачи нам подойдет приложение Блокнот, если вы работаете в Windows, или kedit (или любой другой текстовый редактор), если вы работаете в Linux. Сохраните файл под именем 1.html и откройте его в браузере (рис. 6.1).



Рис. 6.1. Файл 1. html в браузере Opera

Теперь измените наш первый HTML-документ так:

<TITLE>Это мы увидим в заголовке браузера</TITLE><hl>Информация</hl>

Ter <html> обозначает начало HTML-документа, а тег </html> — его конец. Ter <head> содержит некоторую служебную информацию — теги <title> (заголовок), <Meta> и др.

Тег <вору> — это начало тела документа. В этом теге находится вся информация, которая будет отображена в окне браузера.

Самым важным из тегов в разделе HEAD будет тег <META>, определяющий так называемую мета-информацию — ключевые слова, кодировку, время окончания срока использования документа.

Вот примеры использования тега <МЕТА>.

• Указание кодировки документа:

<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">

 Информация об авторе: <meta name="Author" content="Denis Kolisnichenko">

• Информация о композере (программе, с помощью которой был создан до-

кумент):

```
<meta name="GENERATOR" content="Mozilla/4.7 [en] (Win98; I) [Netscape]">
```

• Ключевые слова:

<meta name="keywords" content="Linux, UNIX">

- Описание сайта: <meta name="description" content="Мой сайт">
- Время окончания срока использования документа: <meta http-equiv="Expires" content="Mon, 06 Jan 1990 00:00:01 GMT">

Важно правильно объявить теги МЕТА. Например, правильно указав ключевые слова, вы упростите поиск своего ресурса с помощью поисковых систем, а следовательно, повысите посещаемость сайта. Также рекомендую указать в качестве даты Expire некоторую дату из далекого прошлого. Браузер не загружает из кэша просроченный документ, а старается получить его новую версию. Таким образом, пользователи всегда получат новую версию вашей странички, а не старую, сохраненную в кэше браузера. Данный прием очень полезен, если информация на вашем сайте часто обновляется.

Совет. Удерживайте нажатой клавишу <Shift> при щелчке на кнопке **Обновить** в окне браузера — так вы гарантировано получите новую версию документа.

6.1.2. Типы тегов

Теперь поговорим о тегах, содержащихся в теле документа. В теле документа могут находиться следующие теги.

- Теги физического и логического форматирования текста.
- Гиперссылки.
- Таблицы.
- Теги рисунков.
- Теги списков.
- Теги элементов управления.
- Другие теги.

Мы будем рассматривать далеко не все теги, а только самые необходимые нам (первые пять групп). Если вас интересуют все теги стандарта HTML4, то откройте файл tags.chm, который можно скачать по адресу http://dkws.org.ua/files/tags.rar.

6.1.3. Теги физического форматирования текста

Начнем с тегов физического форматирования текста. К этим тегам относятся: В, BR, FONT, Hn, HR, I, P, PRE, SUB, SUP, U.

Теги В, **I** и **U** делают текст полужирным, наклонным (курсив) и подчеркнутым соответственно. Пример:

```
<В>Полужирный текст</В>
<I>Курсив</I>
<U>Подчеркнутый</U>
```

Тег вк вставляет разрыв строки. Тег BR не требует закрывающего тега. Пример:

Это абзац
Новая строкаСледующий абзац

Как вы уже догадались, **тег Р** используется для обозначения абзаца. В окне браузера данный HTML-код будет выглядеть как на рис. 6.2.



Рис. 6.2. Обработка браузером тегов Р и ВR

С помощью **тега FONT** можно указать характеристики используемого шрифта: гарнитуру, размер и цвет. Например:

Tekct

Атрибут size предназначен для управления размером шрифта и может принимать значения от 1 до 7. Значение по умолчанию — 3.

В качестве значение атрибута color можно использовать значение в виде *RRGGBB*, например #FFFFFF — белый, или использовать одну из именованных констант цвета (табл. 6.1).

Unon	IC an amount of	DDCCDD
цвег	KOHCTAHTA	RKGGDD
Белыи ц	black	
Черныи	black	000000
Синии	Dine	0000FF
Зеленый	green	008000
Голубой	cyan	005.5.5.5
Красный	red	F'F'0000
Сиреневый	magenta	FFOOFF
Коричневый	brown	A52A2A
Серый	gray	808080
Оттенок синего	aliceblue	FOF8FF
ла~ Лче.		FAEBD7
 		OOFFFF
Аквамарин	aquamarine	7fffd4
Очень близкий к белому	azure	FOFFFF
Бежевый	beige	F5F5DC
Немного темнее, чем antiquewhite	bisque	FFE4C4
Похож на бежевый	blanchedalmond	FFEBCD
Оттенок фиолетового	blueviolet	8A2BE2
Цвет дерева	burlywood	DEB887
Оттенок синего	cadetblue	5F9EA0
Оттенок зеленого	chartreuse	7FFF00
Шоколадный	chocolate	D2691E
Коралловый	coral	FF7F50
Оттенок синего	cornflowerblue	6495ED
Шелковый	cornsilk	FFF8DC
Темно-синий	darkblue	00008B
Темно-голубой	darkcyan	008B8B
Темное золото	darkgoldenrod	B8860B
Темно-серый	darkgray	A9A9A9
Темно-зеленый	darkgreen	006400
Темный хаки	darkkhaki	BDB76B
Темно-сиреневый	darkmagenta	8B008B
Оливковый	darkolivegreen	556B2F
Темно-оранжевый	darkorange	FF8C00
Похож на blueviolet	darkorchid	9932CC
Темно-красный	darkred	8B0000
"Лосось"	darksalmon	E9967A
Оттенок зеленого	darkseagreen	8FBC8F
Темный фиолетовый	darkviolet	483D8B

Таблица. 6.1. Именованные константы значения цвета

		Окончание табл. 6.1
Цвет	Константа	RRGGBB
Насыщенный розовый	deeppink	FF1493
Цвет неба	deepskyblue	OOBFFF
Зеленый лес	forestgreen	228B22
"Привидение"	ghostwhite	F8F8FF
Золотой	gold	FFD700
Кислотный зеленый	greenyellow	ADFF2F
Оттенок розового	hotpink	FF69B4
"Индеец"	indianred	CD5C5C
"Индиго"	indigo	4B0082
Слоновый	ivory	FFFFFO
rt'f FO' yellowgreen ç whitesmoke лйка" lawngreen пло-синий lightblue зетло-коралловый Светло-голубо"		F0E68C
Зеленый с желтым оттенком	yellowgreen	9ACD32
Цвет дыма	whitesmoke	F5F5F5
"Лужайка"	lawngreen	7CFC00
Светло-синий	lightblue	ADD8E6
Светло-коралловый	lightcoral	F08080
Светло-голубой	lightcyan	EOFFFF
Светло-зеленый	lightgreen	90EE90
Светло-серый	lightgrey	D3D3D3
Светло-розовый	lightpink	FFB6C1
Цвет неба (светлый)	lightskyblue	87CEFA
"Синяя сталь" светлый	lightsteelblue	B0C4DE
Светло-желтый (близкий к белому)	lightyellow	FFFFEO
Кислотно-зеленый	lime	00FF00
Более темный, чем lime	limegreen	32CD32
"Марон"	maroon	800000
Аквамарин средней насыщенности	mediumauqamarine	66CDAA
Синий средней насыщенности	mediumblue	0000CD
Пурпурный средней насыщенности	mediumpurple	9370D8
"Полночь"	midnightblue	191970
Фирменный цвет ВМФ США	navy	08000
Оливковый	olive	808000
Оранжевый	orange	FFA500
Пурпурный	purple	800080
Серебряный	silver	COCOCO
Цвет неба	skyblue	87CEEB
Снег	snow	FFFAFA
Фиолетовый	violet	EE82EE
Томат	tomato	FF6347
"Голубая сталь"	steelblue	4682B4

Пример. Устанавливаем шрифт Arial, красный, размером 6:

```
<font name="Arial" color="red" size="6">Tect</font>
<font name="Arial" color="#FF0000" size="6">Tect</font>
<font name="Arial" color="FF0000" size="6">Tect</font>
```

Все эти три тега идентичны. Тег FONT обязательно требует закрывающий тег. **Теги Н***n* используются для создания заголовка и подразумевают использование закрывающего тега (рис. 6.3). Примеры:

```
<hl>Самый большой заголовок</hl><h2>Чуть меньше</h2><h3>Еще меньше</h3><h4>Совсем маленький</h4><h5>Меньше</h5><<h6>Самый мелкий</h6>
```



Рис. 6.3. Вид группы тегов Hn из приведенного выше примера в браузере

Теги Hn имеют два атрибута — align и title. Первый управляет выравниванием заголовка, а второй определяет всплывающую подсказку. Выравнивание может быть:

- по центру center;
- по левому краю left;
- по правому краю right;
- по ширине justify.

Пример:

<hl align="justify">Самый большой заголовок</hl>

Тег ня создает горизонтальную линию. Атрибуты тега:

- size определяет высоту линии;
- width определяет ширину линии в пикселях или процентах;

- color задает цвет линии;
- noshade создает линию без тени.

Пример:

```
<hr color="red" width="100%">
```

С тегом **Р** вы уже знакомы — он обозначает создание нового абзаца. Наличие закрывающего тега не обязательно. Для этого тега характерны те же самые атрибуты, что и для тегов Hn, т.е. align u title.

Ter PRE подойдет, если вы надумаете вставить в HTML-документ заранее отформатированный текст так, чтобы сохранилось форматирование, например, код программы — ведь если вы сделаете отступы в тексте программы с помощью двух символов табуляции, то в окне браузера каждый символ табуляции будет заменен одним (!) пробелом. Поэтому в этом случае лучше использовать тег PRE и закрывающий тег PRE (рис. 6.4). Например:



Рис. 6.4. Использование тега PRE

В случае использования тега PRE форматирование сохраняется, но могут исчезнуть некоторые символы. Как правило, исчезают специальные символы (<, > и др.). Для отображения таких символов можно использовать конструкцию &#nnn, где nnn — код символа в стандарте ASCII или ISO-8859-1. Часто используемые специальные символы представлены в табл. 6.2 именованными константами и кодами.

Таблица 6.2.	Специальные	символы

Символ	Конструкция
Больше чем	>
Меньше чем	<
Пробел (неразрывный)	
Прямые кавычки	"
Угловые кавычки	«,»
Амперсанд	&
Знак авторских прав ©©	¢сору
Зарегистрированная торговая марка — ®	®
Торговая марка — тм	™
Коммерческое эт или "ухо" — @	@
Градус — 36,6°	°

Для вставки в HTML-документ формул вам помогут **теги SUB и SUP** — нижний и верхний индексы соответственно. Наличие закрывающего тега обязательно (рис. 6.5).

```
<h2>Математическая формула
y<sup>''</sup> = x<sup>4</sup> + 10y<sup>'</sup>
<h2>Химическая формула
H<sub>2</sub>0
```

1. the installast Cit. toral. Opera		
nale (pass but benars bysens (pass-sector) Hetpeen Control enters 10 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	 Operation International International 	₿. 160
Математическая формула		
$y'' = x^4 + 10y'$		
Химическая формула		
н,о		
1.50		

Рис. 6.5. Запись формул с помощью НТМL-тегов

Совет. Следует заметить, что сложные формулы лучше сохранять в графическом виде и вставлять в документ как графику с помощью тега IMG.

6.1.4. Гиперссылки

К этой группе относится только один тег — А, имеющий четыре атрибута.

- name определяет имя ссылки. Данный атрибут обычно используется при создании ссылок между фреймами.
- href задает URL pecypca. Например, href="http://dkws.org.ua". Для создания ссылки на адрес e-mail используется префикс mailto: href="mailto:root@localhost". Такие ссылки, как правило, обрабатываются почтовым клиентом, например, программой The Bat!
- target цель ссылки.
- title всплывающая подсказка.

Bce атрибуты, кроме target, в особых комментариях не нуждаются. А вот о target нужно поговорить подробнее. Этот атрибут может принимать одно из пяти значений, описанных в табл. 6.3.

Таблица 6.3. Значения атрибута target

Значение	Описание
Имя	Имя ссылки или фрейма. Pecypc URL, указанный в атрибуте href, будет загружен во фрейм с именем <i>имя</i>
_blank	Загружает указанный в атрибуте href pecypc в новое окно
_parent	Загружает указанный в атрибуте href pecypc в родительское окно
_search	Загружает указанный в атрибуте href pecypc в строку поиска браузера. Работает только с браузером Internet Explorer 5 и выше
_ ^{self}	Загружает указанный в атрибуте href pecypc в это же окно, в котором была активизирована ссылка. Значение по умолчанию.

Тег А обязательно требует закрывающий тег. Пример:

Текст ссылки

6.1.5. Таблицы

Для создания таблиц используются теги TABLE, TR и TD. Рассмотрим пример создания простой таблицы.

```
<TABLE border="1" bordercolor="blue" width="100%">
<TR>
<TD>Фамилия</TD><TD>Адрес</TD>TD>Телефон</TD>
</TR>
<TD>Иванов И. П.</TD><TD>Пушкина 6</TD><TD>111-11-11</TD>
</TR>
<TR>
<TD>Петров В. П.</TD><TD>Егорова 9</TD><TD>111-22-22</TD>
</TR>
</TABLE>
```



В браузере эта таблица будет выглядеть так, как показано на рис. 6.6.

Рис. 6.6. Простая таблица

Ter TABLE создает таблицу. Все остальные ее элементы должны быть вложены в этот тег. Под остальными тегами подразумеваются теги TR, TD, TH (заголовок таблицы). В тег TABLE можно вложить еще одну таблицу, т.е. еще один тег TABLE. В него, в свою очередь, еще один и т.д.

Ter TABLE имеет много атрибутов, поэтому начнем по порядку — атрибут align. Данный атрибут используется для выравнивания таблицы. Может принимать три значения:

- left выравнивание по левому краю;
- right по правому;
- center по центру.

Атрибут background определяет рисунок, используемый в качестве фона таблицы, например background="1.gif". А атрибут bgcolor — цвет заднего плана.

Атрибуты border и bordercolor описывают рамку таблицы. Первый задает ширину рамки таблицы в пикселях, а второй — изменяет ее цвет.

Параметры cellspacing и cellpadding управляют отображением ячеек. Первый задает расстояние между ячейками таблицы, а второй — между содержимым ячейки и ее рамкой.

Отображением самих рамок занимается атрибут rules. Он может принимать значения, указанные в табл. 6.4.

Атрибут title, как обычно, задает текст для всплывающей подсказки. А с помощью атрибута width можно установить ширину таблицы в пикселях или в процентах. В нашем примере мы растянули нашу таблицу на все окно браузера, установив ширину 100%.

Значение	Описание
All	Будут видны все рамки
Cols	Будут показаны только вертикальные рамки
Rows	Будут показаны только горизонтальные рамки
None	Рамки не отображаются
Groups	Будут показаны горизонтальные рамки, отделяющие группы THEAD, TBODY и TFOOT

Ter TR означает начало новой строки таблицы, а **тег TD** предназначен для создания новой ячейки в этой строке. Данные теги требуют закрывающий тег. Из всех атрибутов данных тегов вам пригодятся только следующие атрибуты тега TD:

- bgcolor определяет цвет фона ячейки;
- colspan и rowspan объединение ячеек в строке или столбце.

Предположим, что нам нужно создать сложную таблицу, например, такую, как показано на рис. 6.7.

	10
	-
K B	2991
31 2	121,50
the second se	
	JI I

Рис. 6.7. Пример сложной таблицы

Обратите внимание, что мы объединили две ячейки по вертикали в первом столбце (Доходы тыс.руб./мес) и целых пять ячеек — по горизонтали в верхней строке (Год). Подобный эффект достигнут с помощью атрибутов rowspan и colspan тега TD. Bot HTML-код этой таблицы:

```
<TABLE border="1" width="100%">
<TR>
<TD rowspan=2>Доход, тыс. руб./мес. </TD><TD colspan=5>Год</TD>
</TR>
19971998200020012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001200120012001<
```

```
<TD>Январь</TD><TD>200,34</TD><TD>195,45</TD><TD>234,11</TD>
<TD>189,31</TD><TD>121,98</TD>
</TR>
<TR>
<TD>Февраль</TD><TD>211,34</TD><TD>205,45</TD><TD>210,11</TD>
<TD>193,31</TD><TD>145,98</TD>
</TR>
</TR>
```

6.1.6. Теги рисунков

К этой группе относит один, но очень важный тег, — IMG. Этот тег позволяет вставлять в HTML-страницы графику. А какая же Web-страница обходится без графики?

Данный тег имеет два основных атрибута — src и alt. Первый задает URL рисунка, а второй — альтернативный текст, который будет отображен браузером, если пользователь отключил отображение рисунков. Пример:

```
<img src="pl.png" alt="Kartinka 1">
```

6.1.7. Теги списков

Для создания нумерованных списков используется тег OL (Ordered List), а ненумерованных — UL (Unordered List). Элементы списка заключаются в теги LI. Пример нумерованного списка:

```
<OL>
<LI>Первый элемент</LI>
<LI>Второй элемент</LI>
<LI>Третий элемент</LI>
</OL>
```

Ненумерованные списки допускают указание атрибута type, задающего тип маркера. Атрибут type может принимать такие значения:

- disk закрашенный кружок (по умолчанию):
- circle незакрашенный кружок:
- square квадрат.

Пример определения ненумерованного (маркированного) списка:

```
<UL type="circle">
<LI>Первый элемент</LI>
<LI>Второй элемент</LI>
<LI>Третий элемент</LI>
</UL>
```

6.2. Формы

Формы предназначены для отправки информации CGI-приложению, т.е. программе на сервере, которая будет обрабатывать переданную пользователем информацию. Рассмотрим простую форму ввода имени:

```
<form action="http://localhost/login.php" method="GET">
MMя: <input type=text name=user>
<br><input type=submit value="Bxog">
</form>
```

Данная форма содержит всего два элемента — поле для ввода имени и кнопку с надписью **Вход**. Что же происходит, когда пользователь вводит имя и щелкает на кнопке **Вход**? Браузер передает сценарию login.php на сервере параметр user. В качестве значения параметра будет отправлено введенное пользователем в текстовое поле имя. Поскольку используется метод GET, в строке адреса вы увидите:

```
http://localhost/login.php?user=Denis
```

Существует еще один метод доставки информации на сервер — метод POST, который использует другой способ передачи данных. При его использовании передаваемую информацию в адресной строке браузера вы не увидите. Данный способ используется обычно для передачи большого количества информации или же когда пользователю нежелательно видеть передаваемую информацию в строке адреса.

Все элементы формы делятся на три группы:

- поля ввода (input);
- многострочные поля ввода (textarea);
- списки выбора.

Чаще всего используются поля ввода — ведь даже кнопка является полем ввода. Мы уже знакомы с полем ввода текстовой информации, поэтому обзор полей ввода начнем именно с текстового поля. Формат соответствующего HTML-тега INPUT следующий:

```
<input type=text
name=имя_параметра
[value=значение]
[size=размер]
[maxlen=длина]
```

>

Данный тег создает в HTML-форме поле ввода с максимально допустимой длиной текста maxlen и видимым размером в size знакомест. Если задан атрибут value, то изначально в поле будет отображаться указанная текстовая строка.

Возможно, вы создаете какой-нибудь серьезный проект, требующий аутентификации, т.е. проверки имени пользователя и пароля. Конечно, имя пользователя можно ввести с помощью обыкновенного текстового поля. А вот как быть с паролем, ведь он не должен отображаться на экране? В этом случае вам поможет специальное поле ввода пароля:

```
<input type=password
name=имя_параметра
[value=значение]
[size=размер]
[maxlen=длина]
```

>

Принцип работы данного тега точно такой же, как и текстового. Но следует иметь в виду, что из соображений безопасности не рекомендуется задавать значение value. Да, в окне браузера данное значение не отображается, но стоит просмотреть исходный HTML-код, и пароль будет виден "невооруженным глазом".

Для передачи служебной информации (о которой пользователь даже не должен подозревать) используются скрытые поля:

```
<input type=hidden
name=имя
value=значение
```

Очень часто нам нужно предоставить пользователю возможность включить или выключить какой-нибудь режим работы, например: "Да, я хочу получать информацию о ...". Это можно сделать с помощью переключателя, описываемого следующим образом:

```
input type=checkbox
name=имя
value=значение
[checked]
```

Если атрибут checked указан, то данный флажок по умолчанию будет в установленном состоянии. Если флажок в форме был установлен до щелчка пользователя на кнопке отправки данных, серверному сценарию будет передан параметр имя=значение. Если же флажок сброшен, то никаких сведений о нем сценарию вообще не будет передано — как будто этого флажка вообще нет.

Переключатель checkbox называется независимым, так как его свойство checked не зависит от состояния других переключателей типа checkbox в данной форме. И наоборот, переключатель типа radio будет зависимым, поскольку в форме только один из переключателей этого типа может быть в установленном состоянии. Точнее, в форме может быть только один активный переключатель в *своей группе*, а независимых групп переключателей типа radio в одной форме допускается несколько. Чтобы объединить переключатели radio в группы, нужно задать для них одно и то же свойство (атрибут) **name**. Рассмотрим следующий пример:

```
<form action="http://localhost/search.php" method="post">
    <input type=radio
        name=area
        value=site
        checked
        >∏o caйтy
    <input type=radio
        name=area
        value=internet
        >B Интернете
    <input type=submit
        name=go
        value=Поиск
    >
</form>
```

Первый переключатель (со значением site) активен по умолчанию (для него задан атрибут checked). Как только пользователь щелкнет на кнопке Поиск, сценарию search.php будет передан параметр area (значение атрибута name обоих переключателей) со значением site. Если же пользователь выберет другой вариант (internet), сценарию будет передано это значение параметра area.

Последней группой элементов управления типа INPUT являются кнопки. О кнопке submit, передающей введенные пользователем данные на сервер, уже мы знаем практически все. Подскажем только, что атрибут value определяет текст, отображаемый на кнопке:

```
<input type=submit
name=go
value=Передать
```

Примечание. Мы уже знаем, что значения атрибутов тегов желательно заключать в кавычки. Желательно, но необязательно — большая часть браузеров нормально обрабатывает значения атрибутов без кавычек.

Кроме кнопки submit, есть еще кнопка recset, которая предназначена для сброса формы — установки всех ее элементов управления в состояние по умолчанию. Желательно, чтобы на вашей форме была такая кнопка, особенно, если это большая форма. Наличие данной кнопки упрощает очистку формы, если пользователем были введены неправильные параметры. Описывается такая кнопка очень просто:

```
<input type=reset
value=Cброс
```

Вместо кнопки submit для отправки данных можно использовать рисунок, чтобы сделать ее более привлекательной. После щелчка на этом рисунке произойдет то же самое, что и после щелчка на кнопке типа submit. Кроме того, сценарию будут переданы координаты места выполнения щелчка в пределах рисунка. Координаты будут переданы в формате имя.x=кооp_X, имя.y=коор_У. Пример:

```
<input type=image
   name=uMA
   src=pucyHok
>
```

На этом обсуждение полей ввода заканчивается, и мы переходим к рассмотрению многострочного поля ввода — textarea. Поле textarea позволяет отправить на сервер не одну строку, а сразу несколько:

```
<textarea
name=имя
[cols=ширина_B_символах]
```

[rows=высота_в_символах] wrap=тип_переноса >Текст по умолчанию </textarea>

Несколько замечаний относительно использования этого поля: необязательные параметры cols и rows желательно указывать. Первый из них задает количество символов в строке, а второй — количество строк в области. Атрибут wrap определяет тип переноса текста и имеет следующие допустимые значения.

- Virtual. Справа от текстового поля выводится полоска прокрутки. Вводимый пользователем текст разбивается на строки, причем символ новой строки вставляется после нажатия клавиши <Enter>.
- Physical. Этот тип зависит от браузера и в разных браузерах может обрабатываться по-разному.
- None. Текст выглядит в поле в том виде, в котором пользователь его вводит. Если текст не умещается в одну строку, появляется горизонтальная полоска прокрутки.

Стоит заметить, что самым удобным считается тип Virtual.

Довольно часто нам нужно представить какие-нибудь данные в виде списка выбора, позволяющего выбрать один вариант из множества. В принципе для этой цели можно использовать группу переключателей типа radio, но это нерационально в случае большого количества вариантов. Например, если нам нужно, чтобы пользователь выбрал месяц, то один список будет занимать намного меньше места в окне браузера, чем группа из двенадцати переключателей типа radio. Вот пример подобного списка, определяемого с помощью тегов select и option:

```
<select name=country size=1>
  <option value=01 selected>Poccus</option>
  <option value=02>Vkpauha</option>
  <option value=03>CUA</option>
  <option value=04>Kahaga</option>
  <option value=05>Φpahuus</option>
  <option value=06>Италия</option>
  <option value=07>Германия</option>
  <option value=09>Испания</option>
  <option value=10>Швеция</option>
  <option value=12>Греция</option>
  <option value=13>Турция</option>
  <option value=14>Другая...</option>
</option>
</option</pre>
```

</select>

Атрибут name определяет имя параметра, который будет передан сценарию — country. Если атрибут size paвeн 1, то список будет раскрывающимся, в противном случае он займет n строк и будет снабжен полоской прокрутки. В нашем примере первый элемент списка (Россия) будет выбран в списке по умолчанию (атрибут selected). Когда пользователь выберет один из вариантов, серверу будет передано соответствующее значение value. Например, если пользователь

выберет США, сценарию будет передан параметр country=03. Атрибут value необязателен. Если его не указать, то будет передана строка, заключенная в тег option; в нашем примере сценарию будет передан параметр country=CША.

Список выбора позволяет выбирать не только один вариант, но и сразу несколько. Для этого используется атрибут multiple. Ясное дело, что этот параметр не имеет смысла, если атрибут size=1. Если пользователь выберет несколько вариантов, то серверному сценарию будут переданы параметры в виде:

имя=значение&имя=значение...&имя=значение

Выше я говорил, что мы рассмотрели уже все поля типа input, однако это неверно: существует еще одно поле этой группы — поле загрузки файла. Его определение выглядит так:

```
<input type=file
name=имя
[value=имя_файла]
```

Однако подробнее на этом поле мы останавливаться не будем, поскольку сейчас оно нас совсем не интересует.

6.3. Форма поиска

Чтобы разместить форму поиска на вашем сайте, достаточно вставить следующий код в основную HTML-страницу:

```
<!-- Search Google -->
<form method="get" action="http://www.google.com/search">
<input type="text" name="q" size=31 maxlength=255 value="">
<input type="submit" name="sa" value="Поиск в Google">
</form>
<!-- Search Google -->
```

Но от такой формы толку будет мало, конечно, если у вас не какой-нибудь информационный сайт. В этом случае пользователям просто будет предоставлен быстрый доступ к поисковой машине Google. Но в большинстве случаев у вас будет сайт определенной тематики и вам предстоит немного изменить эту форму, например, попросив Google искать файлы определенного типа. А также, ясное дело, вам захочется, чтобы Google искал информацию и на вашем сайте. Все это можно организовать с помощью *скрытых полей*, которые предназначены для передачи параметров по умолчанию. Пользователи не видят этих полей и не могут изменить содержащиеся в них данные. Реализация поиска на определенном сайте:

<input type="hidden" name="as_sitesearch" value="ваш_сайт">

• зададим количество результатов:

<input type="hidden" name="num" value="100">

• определяем тип файла (ищем только по PDF-файлам):

```
<input type="hidden" name="as filetype" value="PDF">
```

устанавливаем возраст результата (в месяцах):
 <input type="hidden" name="as qdr" value="m7">

Теперь обобщим сказанное в конкретном примере. Форма для поиска на сайте dkws.org.ua PDF-файлов не старше семи месяцев будет выглядеть так:

```
<!-- Search Google -->
<form method="get" action="http://www.google.com/search">
<input type="text" name="q" size=31 maxlength=255 value="">
<input type="hidden" name="as_sitesearch" value="dkws.narod.ru">
<input type="hidden" name="as_filetype" value="PDF">
<input type="hidden" name="as_filetype" value="PDF">
<input type="hidden" name="as_filetype" value="m7">
<input type="bidden" name="as_filetype" value="m7">
```

Попробуем создать форму для поиска файлов MS Office. Данная форма должна обеспечивать выбор тип файла (Excel, Word, Power Point), а также указание сайта, на котором будет проводиться поиск, — Lib.Ru или Rada.gov.ua.

```
<!-- Поиск файлов Office-->
<h2>Поиск файлов MS Office</h2>
<br>Ключевое слово:
<form method="get" action="http://www.google.com/search">
<input type="text" name="q" size=31 maxlength=255 value="">
```

Реализация возможности выбора типа файла:

```
<select name="as_filetype">
   <option value="xls">Excel</option>
   <option value="doc">Word</option>
   <option value="ppt">Power Point</option>
</select>
```

Организация поиска на требуемом сайте:

```
<select name="as_sitesearch"></option>
        <option value="lib.ru">Lib.Ru</option>
        <option value="rada.gov.ua">Rada</option>
        </select>
        <input type="hidden" name="num" value="100">
        <input type="submit" name="sa" value="Поиск в Google">
</form>
<!-Поиск файлов Office-->
```

В завершенном виде эта форма выглядит так, как показано на рис. 6.8.



Рис. 6.8. Форма поиска файлов MS Office на заданных сайтах

Использовать эту форму очень просто: можно вставить ее код в одну из HTMLстраниц вашего сайта или же просто сохранить ее в HTML-файле на жестком диске и запускать при необходимости. Учтите, что размещать форму на сайте вовсе необязательно — она прекрасно работает и при запуске непосредственно с жесткого диска. Все, что вам нужно, — это подключиться к Интернету, запустить форму, ввести ключевое слово и щелкнуть на кнопке Поиск в Google.

6.4. Форма для переводчика

Предположим, что вы работаете переводчиком и вам часто приходится искать различные словари, содержащие толкования и переводы различных терминов. Следующая форма (рис. 6.9) поможет найти словари, содержащие ваш термин, а также где именно вы хотите искать заданный термин — в заголовке, URL или везде.

```
<!-- Форма будет открываться в новом окне ( blank) -->
<form method="GET" target=" blank"
     action="http://www.google.com/search">
   <!-- Ключевое слово -->
   <input type="text" name="q" size="12">
   <select name="q" size="1">
   <!-- Несколько модификаторов, позволяющих найти словари
                                                               -->
      <option selected value="intitle:dictionary</pre>
             OR intitle:glossary OR intitle:lexicon
             OR intitle:definitions"> синонимы
             "glossary" в TITLE - 1</option>
      <option value="intitle:terminology</pre>
              OR intitle:vocabulary OR intitle:definition
              OR intitle:jargon"> синонимы "glossary"
              в TITLE - 2</option>
```

```
<option value="inurl:dictionary OR inurl:glossary
OR inurl:lexicon OR inurl:definitions">
cинонимы "glossary" в URL = 1</option>
<option value="inurl:terminology OR inurl:vocabulary
OR inurl:definition OR inurl:jargon">
cинонимы "glossary" в URL = 2</option>
<option value="inurl:dict OR inurl:gloss OR inurl:glos
OR inurl:dic"> аббревиатуры "glossary"
в URL</option>
<option value="dictionary OR glossary OR lexicon
OR definitions">синонимы "glossary" BE3ДE</option>
</select>
<input type="submit" value="Поиск">
<input type="reset" value="C6poc" name="B2"></form>
```

```
        • The Mandalaser, Cit Navel Opena
        ■ ● ○

        exits frame that Sectors Bytestic Opena
        ● ● ○

        • Onders instant
        Opena

        • Onders instant
        Opena

        • ● ● ○
        ● ○

        • Onders instant
        • ●

        • ● ● ○
        ● ●

        • ● ● ○
        ● ●

        • ● ● ○
        ● ●

        • ● ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

        • ●
        ●

    <t
```

Рис. 6.9. Форма поиска словарей