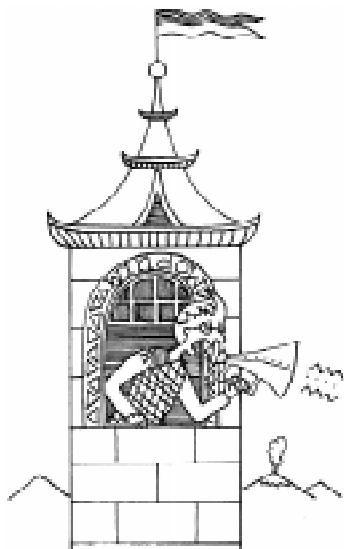


JAVASCRIPT. ЗАНЯТИЕ 4. УПРАВЛЕНИЕ WEB-СТРАНИЦАМИ

ОБЪЕКТЫ БРАУЗЕРА



В языке JavaScript определены объекты, которые называются объектами браузера. Каждый объект соответствует некоторому элементу Web-страницы: окну, документу, изображению, ссылке и т. п. Управлять частями документа можно с помощью методов браузера. Объекты браузера имеют иерархическую структуру. На самом верхнем уровне располагается объект **window**. Он является родителем остальных объектов. Объект **window** представляет окно браузера. Свойство **window.status** можно использовать для изменения вида строки состояния.

Для отображения диалоговых сообщений объект **window** имеет три метода.

МЕТОД ALERT

Метод **window.alert** отображает диалоговое окно, в которое помещается со-

общение для пользователя. Этот метод часто используется при обработке полей формы. Если в какое-либо из полей формы введено неверное значение, то пользователю посылается сообщение, которое изображено на рисунке 1.

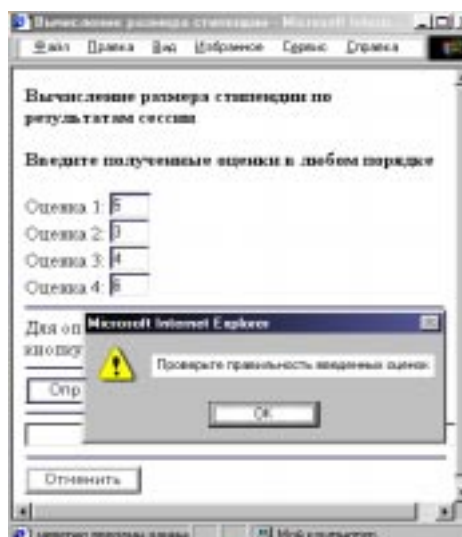
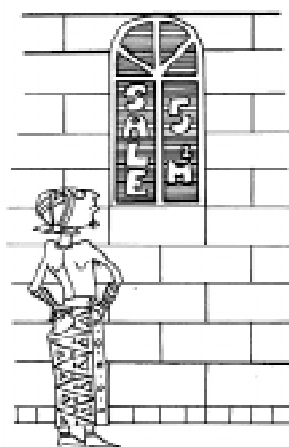
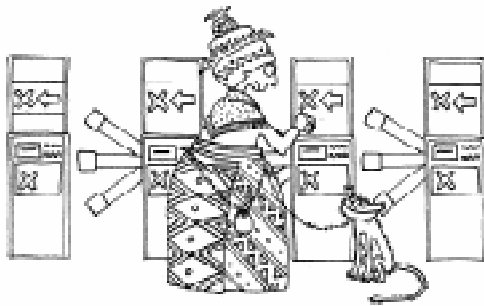


Рисунок 1. Метод **alert** при анализе введенных данных



ИСПОЛЬЗОВАНИЕ МЕТОДА CONFIRM



Метод **confirm** отображает диалоговое окно подтверждения выполнения операции. Оно содержит две кнопки **ОК** и **Cancel (Отмена)**, позволяющие выбрать один из вариантов. Если пользователь щелкнул по кнопке **ОК**, то возвращается значение **true**, после щелчка по кнопке **Отмена** возвращается значение **false**. Вид диалогового окна **confirm** изображен на рисунке 2.

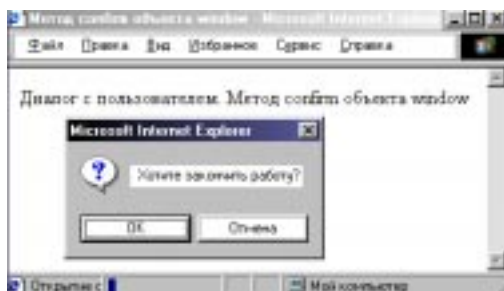


Рисунок 2. Метод **confirm** для организации диалога с пользователем

ОКНО ЗАПРОСА ДАННЫХ



Метод **prompt** используется для вывода диалогового окна запроса данных. При щелчке по кнопке **ОК** введенные пользователем в текстовое поле данные отображаются в документе. Если выбрано значение **Cancel**, то возвращается значение **Null**. Диалоговое окно запроса данных изображено на рисунке 3.

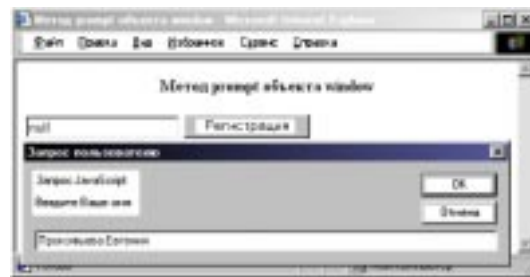


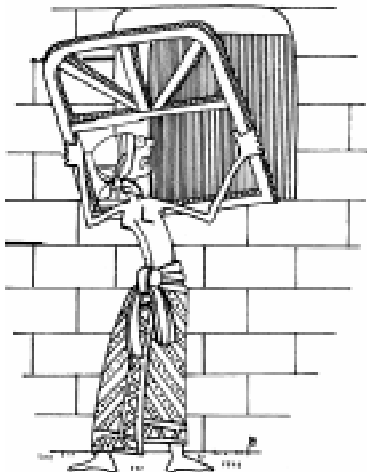
Рисунок 3. Диалоговое окно запроса данных

Листинг 1. Диалоговое окно запроса данных

```
<HTML>
<HEAD>
<TITLE>Метод prompt объекта window</TITLE>
</HEAD>
<SCRIPT>
function reg (obj)
{obj.regname.value = window.prompt ("Введите Ваше имя", "") }
</SCRIPT>
<BODY>
<P>Метод prompt объекта window</P>
<form name="form1">
<input type=text name=regname>
<input type="button" value="Регистрация" onClick= "reg(form1)">
</form></BODY></HTML>
```

Метод **prompt** имеет второй параметр, с помощью которого задается значение по умолчанию. HTML-код со сценарием, использующим метод **prompt**, хранится в листинге 1.

СОЗДАНИЕ НОВОГО ОКНА



При выполнении сценария пользователь может создавать новые окна, загружать в них различные документы и др. Новое окно браузера создается с помощью метода **windows.open**.

Приведем оператор присваивания, в правой части которого используется метод **windows.open**.

```
wint = window.open ("anketa.htm",
                    "mywin", "")
```

Переменная **wint** используется для хранения данных о новом окне (объект **window**). Эту переменную можно использовать для получения свойств и методов нового созданного окна. В качестве первого параметра при создании нового окна задается URL-адрес документа, загружаемого в создаваемое окно. Если значение первого параметра не указано, то созданное окно будет пустым. Второй параметр определяет название окна, это значение соответствует свойству **name** нового объек-

Листинг 2. Окна, определяемые пользователем

```
<HTML>
<HEAD>
<TITLE>Окна, определяемые пользователем</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
// создание нового окна
function opnewWin()
{ wint = window.open ("anketa.htm", "myWin", "") }
// определение свойств созданного окна
function propwin(obj)
{ obj.t0.value= wint.name
  obj.t1.value= wint.location.href
}
//-->
</SCRIPT></HEAD>
<BODY>
<h4>Вывод на экран диалоговых окон, определенных пользователем</h4>
<form name="form1">
<input type=button value="Открыть анкету" onClick="opnewWin()">
<input type=button value="Закрыть анкету" onClick="wint.close()">
</form>
<h5 align=center>Свойства созданного окна</h5>
<pre>
<form name="form2">
<input type=button value="Посмотреть" onClick="propwin(form2)"><br>
<input type=text name=t0> name
<input type=text name=t1 size=40> location.href
</form></pre></BODY></HTML>
```

та **window**. В простейшем случае третий параметр можно не указывать, тогда при создании окна его параметры будут использоваться значения по умолчанию. HTML-код документа, содержащий сценарий создания нового окна, приведен в листинге 2.

При щелчке по кнопке **Открыть анкету** создается новое окно, в которое помещается документ с именем **anketa.htm**. Для того чтобы закрыть новое окно, можно щелкнуть по второй кнопке. В качестве значения параметра обработки события вызывается метод **close()**. Два свойства нового окна выводятся в текстовое поле формы.

Третий параметр содержит список необязательных опций. Если ни один из параметров не указан, как в предыдущем случае, то используются значения по умолчанию. После щелчка по кнопке **Открыть анкету** откроется новое окно, опции которого задаются по умолчанию.

С помощью опций можно, например, задать первоначальную ширину окна в пикселях, наличие или отсутствие меню в верхней части окна, наличие панели инструментов и др.

пок для статических ссылок, свойство **scrollbar** определяет, требуются ли полосы прокрутки, если документ не помещается на экран.

При задании третьего параметра опции, соответствующие свойствам окна, могут принимать одно из двух значений **yes** (1) или **no** (0). Для того чтобы создать окно с первоначальной заданной шириной в 400 пикселей и высотой в 300 пикселей, следует воспользоваться методом **open** с параметрами:

```
wint = window.open ("anketa.htm",
    "myWin", "width=400,height=300").
```

Для задания окна, не содержащего меню браузера и URL-адреса страницы, загруженной в окно, следует выполнить:

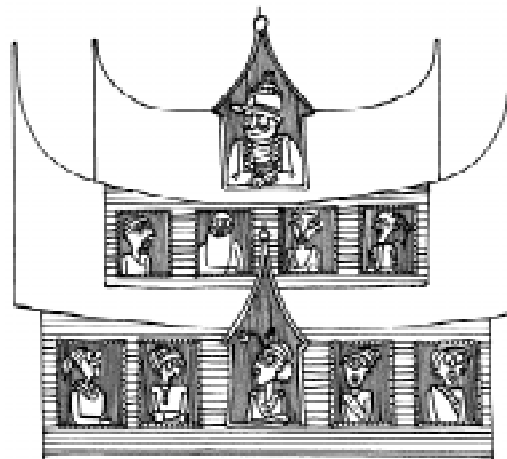
```
wint = window.open ("anketa.htm",
    "myWin", "menubar=no,location=no").
```

УПРАВЛЕНИЕ ВИДОМ СОЗДАВАЕМОГО ОКНА



Рассмотрим некоторые свойства объекта **window**, которые можно задавать при открытии окна. Свойство **location** содержит текущий URL-адрес страницы, загруженной в окно. Свойство **status** отражает строку состояния в нижней части окна. Свойство **menubar** соответствует меню браузера, а свойство **toolbar** - панели инструментов. Свойство **directories** определяет, отображается ли набор кно-

ИЕРАРХИЧЕСКАЯ СТРУКТУРА ОБЪЕКТОВ



Объект **window** расположен на самом верхнем уровне иерархии объектов. Подчиненными объектами (или объектами нижнего уровня) являются объекты **history**, **location**, **frame**, **document**. Свойства объекта **history** представляют адреса ранее загруженных HTML-страниц. Свойства объекта **location** связаны с URL-адресом отображаемого документа. Свойства объекта **frame** связаны со специальным способом представления данных, который будет рассмотрен в дальнейшем.

Объект **history** содержит сведения о страницах, которые отображались или отображаются в браузере. Свойство **history.length** содержит длину списка адресов посещаемых страниц. К элементам объекта **history** можно обращаться по индексу. В этом случае объект **history** рассматривается как массив, каждый элемент которого содержит адрес посещаемой страницы, текущая страница представлена элементом **history[0]**, предыдущая **history[-1]**, следующая **history[1]**. Объект **history** имеет три метода. Можно открыть страницу, адрес которой хранится в списке посещаемых страниц, например, так: **history.go(-1)** или **history.go(2)**. Использование метода **history.back** приведет к загрузке предыдущей страницы, метод **history.forward** обеспечивает переход к следующей странице.

СВОЙСТВА ОБЪЕКТА DOCUMENT

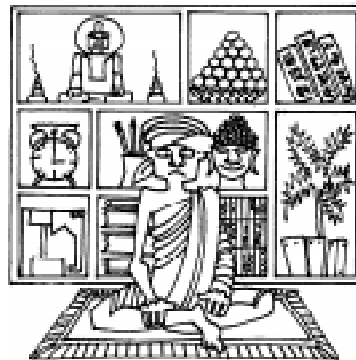


Свойства объекта **document** определяются содержимым самого документа: шрифт, цвет фона, формы, изображения и т. д. Объект **document**, в зависимости от своего содержимого, может иметь объекты, являющиеся для него подчиненными или дочерними. В частности, подчиненным объектом для объекта **документ** являются, объекты **form**, **image**, **link**, **area** и др. Для каждой страницы создается один объект **document**, некоторые его свойства соответствуют параметрам тега **<body>**: **bgColor**, **fgColor**, **linkColor**, **alinkColor**, **vlinkColor**.

Объекты **form** рассматривались при обработке данных, представленных элементами форм, доступ к объектам **image**

осуществлялся при работе с изображениями, при создании графических меню и при программировании визуальных эффектов.

СОЗДАНИЕ ПРОСТЫХ ТАБЛИЦ В HTML



В языке HTML таблицы используются в двух случаях:

- Для представления числовых данных, разбитых по строкам и столбцам.
- Как средство форматирования Web-страниц, задания взаимного расположения элементов страницы.

Описание таблиц должно располагаться внутри раздела документа **<BODY>**. Документ может содержать произвольное число таблиц, допускается вложенность таблиц друг в друга. Каждая таблица должна начинаться тегом **<TABLE>** и завершаться тегом **</TABLE>**. Внутри этой пары тегов располагается описание содержимого таблицы. Любая таблица состоит из одной или нескольких строк, в каждой из которых задаются данные для отдельной ячейки.

Каждая строка начинается тегом **<TR>** и завершается тегом **</TR>**. Если в таблице содержатся две пары тегов **<TR>** и **</TR>**, то это означает, что таблица состоит из двух строк. Отдельная ячейка в строке обрамляется парой тегов **<TD>** и **</TD>** или **<TH>** и **</TH>**. Тег **<TH>** используется обычно для ячеек-заголовков, тег **<TD>** для ячеек-данных. Различие в использовании заключается лишь в типе шрифта, используемого по умолчанию, и в расположении данных внутри ячейки. Теги **<TD>** и **<TH>** должны содержаться внутри тега **<TR>**. Число тегов **<TD>** и

`</TD>` определяет число ячеек в строке. Строка с тремя парами тегов `<TD>` и `</TD>` будет содержать три ячейки.

Перечисленные теги могут иметь параметры, число и значения которых различны. В простейшем случае теги используются без параметров, которые принимают значения по умолчанию. Тег `<TABLE>` может использоваться с рядом параметров, каждый из которых можно опустить. Набор допустимых параметров зависит от браузера.

Каждую ячейку таблицы можно рассматривать как отдельную область для независимого форматирования. Все правила для форматирования текста могут быть использованы для форматирования текста внутри ячейки. Область действия тегов, записанных внутри отдельной ячейки, ограничивается пределами этой ячейки, независимо от наличия завершающего тега. Для форматирования данных внутри ячейки таблицы предусмотрен ряд параметров. С помощью таблицы можно представить, например, шахматную доску с расставленными фигурами.

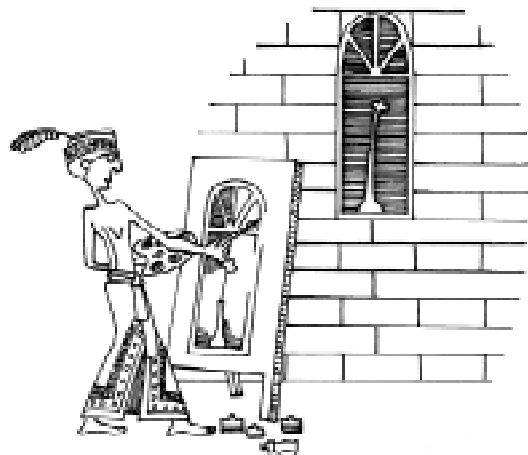
ФРЕЙМЫ. СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ



Окно просмотра браузера можно разбить на прямоугольные области, называемые фреймами. Области соприкасаются друг с другом, и в каждую из областей можно загрузить отдельный HTML-документ и работать с этим документом, независимо от документов, загруженных в другие области окна или фрейма. Между

фреймами можно организовать взаимодействие, например, выбор ссылки в одном из фреймов позволит изменить содержимое других фреймов. Фреймы часто используются в случаях, когда возникает необходимость загрузить документ в одну из областей при работе в другой области, или когда требуется расположить информацию, которая должна постоянно находиться на экране.

ПРАВИЛА ОПИСАНИЯ ФРЕЙМОВ



Фреймовая структура определяет только способ организации экрана с фреймами и указывает, где находится начальное содержимое каждого кадра. Для всех кадров должны быть заданы адреса, описывающие местонахождение данных. Как правило, на странице с фреймовой структурой нет содержимого кадров. Такая страница обычно невелика, описывает только фреймовую структуру экрана. Для описания структуры фреймов используются теги `<FRAMESET>`, `<FRAME>`, `<NOFRAMES>`.

ТЕГ `<FRAMESET>`

Фреймы определяются в структуре, называемой `FRAMESET`, которая используется для страниц, содержащих фреймы, вместо раздела `BODY` обычного документа. Web-страницы, составленные из фреймов, не могут содержать раздела `BODY`, страницы с разделом `BODY` не могут содержать фреймы. Теги `<FRAMESET>` и `</FRAMESET>` обрамляют каждый блок



определения фреймов. Внутри него могут содержаться только теги **<FRAMESET>** и **<FRAME>**.

Тег **<FRAMESET>** имеет два параметра **ROWS** (строки) и **COLS** (столбцы). В общем случае тег **<FRAMESET>** записывается в виде:

```
<FRAMESET ROWS="список значений"
COLS="список значений">
```

Можно определить значения для **ROWS** или **COLS** или обоих вместе. Необходимо определить два значения хотя бы одного из этих двух параметров. Если другой параметр опущен, то его значение принимается равным 100%. Нельзя определить **<FRAMESET>**, состоящий из одного фрейма.

Список значений параметров **ROWS** и **COLS** представляет собой разделенный запятыми список значений, которые могут задаваться:

- в пикселях,
- в процентах,
- в относительных единицах

Число строк или столбцов определяется числом значений в соответствующем списке.

Приведем следующие примеры:

При описании фреймовой структуры

```
<FRAMESET ROWS="100, 220, 160">
```

окно просмотра браузера разбивается по горизонтали на три прямоугольные области размером 100, 220 и 160 пикселей, соответственно.

Определим фреймовую структуру

```
<FRAMESET COLS="20%, 50%, 30%">.
```

В этом случае окно просмотра браузера будет разбито по вертикали на три прямоугольных области, размер которых по отношению к окну браузера составляет 20%, 50%, 30%, соответственно.

Символ (*) используется для пропорционального деления пространства. Каждая звездочка представляет собой одну часть целого. Опишем фреймовую структуру

```
<FRAMESET COLS="*, 2*, 3*"> .
```

В этом случае первый столбец занимает 1/6 часть от общей ширины окна, второй – 1/3, третий – 1/2.

Описание фреймовой структуры, в которой первый столбец займет ширину 100 пикселей, второй – 25% от всей ширины окна, третий столбец – 1/3 часть оставшегося пространства, четвертый столбец – 2/3, выглядит следующим образом:

```
<FRAMESET ROWS="100, 25%, *, 2*">.
```

Задание фреймовой структуры **<FRAMESET ROWS="*, 2*, *" COLS="2*, *">** разобьет окно просмотра на шесть прямоугольных областей.

ТЕГ <FRAME>



Тег **<FRAME>** определяет одиночный фрейм. Он должен располагаться внутри пары тегов **<FRAMESET>** и **</FRAMESET>**. Необходимо записать столько тегов **<FRAME>**, сколько отдельных фреймов определено при задании тега **<FRAMESET>**. Тег **<FRAME>** имеет шесть

параметров, которые редко используются одновременно. Пример записи тега со всеми параметрами:

```
<FRAME SRC="f1.htm" NAME="leftframe"
      SCROLLING= YES|NO|auto
      MARGINWIDTH=n MARGINHEIGHT=m
      NORESIZE>
```

Значение параметра **SRC** определяет URL-адрес документа, который изначально будет загружен в заданный фрейм. В приведенном примере первоначально будет загружен документ с именем **f1.htm**, расположенный в текущей папке.

Параметр **NAME** определяет имя фрейма, которое может использоваться для ссылки к данному фрейму, например, при указании, в какой фрейм загрузить документ. Если для фрейма не задано имя, то будет создан фрейм без имени, но не будет возможности использовать ссылки на него из других фреймов.

Параметры **MARGINWIDTH** и **MARGINHEIGHT** дают возможность устанавливать ширину полей фрейма. Они определяют пространство, в пределах которого не будет располагаться никакая информация.

Для фреймов будут автоматически задаваться и отображаться полосы прокрутки, если содержимое фрейма не помещается полностью в отведенное пространство, если значение параметра **SCROLLING** равно **auto**. Если задать значение параметра **SCROLLING** равным **YES**, то полосы прокрутки будут отображаться даже в случае, если они не нужны. Если же значение параметра **SCROLLING** равно **NO**, то полосы прокрутки не будут отображены, даже если они необходимы. По умолчанию, значение параметра **SCROLLING** равно **auto**.

Все, что расположено между тегами **<NOFRAMES>** и **</NOFRAMES>**, будет отображаться браузерами, не поддерживающими фреймы. Браузеры с поддержкой фреймов проигнорируют всю информацию, расположенную между этими тегами.

При записи тега **<FRAME>** следует указывать имя пустого файла, если сразу

невозможно указать имя конкретного файла. Такой файл можно назвать **empty.htm**, содержимым которого будет являться корректный HTML-документ, представленный в листинге 3.

Листинг 3. Пустой документ

```
<HTML>
<HEAD>
<TITLE>Пустой документ</TITLE>
</HEAD>
<BODY></BODY>
</HTML>
```

ПРОСТАЯ ФРЕЙМОВАЯ СТРУКТУРА



Создадим документ, который разбивает область экрана на две части. Левая часть содержит оглавление разделов документа, который располагается в правой части. При выборе пункта оглавления в левой части появляется соответствующий раздел документа в правой части.

Разобьем область экрана на два фрейма. Левый фрейм занимает 25% ширины всего окна и будет содержать оглавление разделов документа, который будет загружен в правый фрейм. Пусть имя файла, содержащего оглавление, – **contents0.htm**, имя документа – **ch.htm**. Фреймовая структура задает способ организации экрана и определяет, какие документы должны быть первоначально загружены во фреймы. Создать описанную фреймовую структуру можно, если использовать документ, содержащий следующий HTML-код (см. листинг 4a).

Листинг 4 а. Создание простой фреймовой структуры

```
<html>
<head>
<title>простая фреймовая структура</title>
</head>
<frameset cols="25%,75%">
  <frame src=contents0.htm name=left>
  <frame src= ch.htm name=right>
</frameset>
</html>
```

Связь между фреймами и документами для описанного примера изображена на рисунке 4.

Файл **ch.htm** содержит так называемые внутренние ссылки, то есть разбит на четыре части.

Перед заголовком первой части расположен тег ****, помечающий начало раздела. К этому разделу можно осуществить переход, например, по ссылке Возврат, заданной следующим образом:

```
<a href="#chapter_1">К главе 1</a>.
```

HTML-код документа, представляющего оглавление, которое загружается в левый фрейм, приведен в листинге 4 б.

Ссылка устанавливается на соответствующий раздел документа заданием параметра **href="ch.htm#chapter_1"**. К каждой из частей осуществляется переход по ссылке из оглавления, расположенного в левом фрейме.

Параметр **target** определяет имя фрейма, в который загружается документ, на который установлена ссылка. В рас-

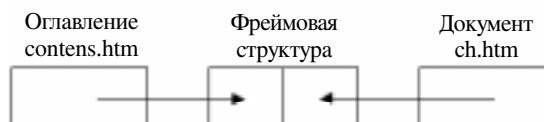


Рисунок 4. Связь между фреймами и документами

сматриваемом случае это фрейм с именем **right**. По умолчанию или при отсутствии параметра **target** документ загружается в текущий фрейм или окно.

Для всех ссылок в документе с именем **contents.htm** указано одинаковое значение параметра **target**. Если все документы, загружаемые по ссылкам, должны загружаться в один и тот же фрейм, то можно задать параметр **target** в теге **<BASE>**.

ФРЕЙМОВАЯ СТРУКТУРА С ЗАГРУЖАЕМЫМИ ДОКУМЕНТАМИ

Создадим документ, левая часть которого представляет оглавление, а в пра-

Листинг 4 б. Содержимое документа, являющегося оглавлением

```
<html>
<head>
<title>Ссылки внутри документа</title>
</head>
<body>
<body background="decor.gif" bgcolor=silver>
<h4 align=center>Содержание</h4>
  <a href="ch.htm#chapter_1" target=right>Объекты браузера</a><br>
  <a href="ch.htm#chapter_2" target=right>Создание окон</a><br>
  <a href="ch.htm#chapter_3" target=right>Таблицы</a><br>
  <a href="ch.htm#chapter_4" target=right>Фреймы</a><br>
</body></html>
```

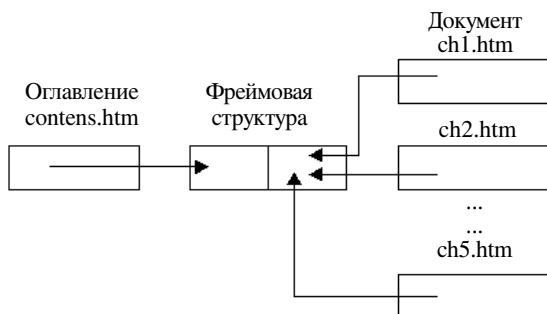


Рисунок 5. Фреймовая структура с документами из разных файлов

вую часть загружаются документы выбранного пункта оглавления. Документы, соответствующие пунктам оглавления, хранятся в разных файлах.

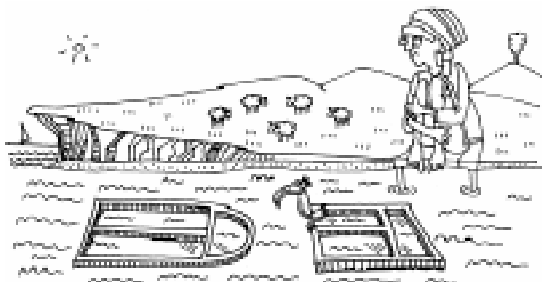
При решении задачи экран разбивается по-прежнему на два фрейма. Левый фрейм занимает 30% ширины всего окна и будет содержать оглавление документов, которые могут быть просмотрены пользователем при выборе соответствующего пункта оглавления. Правый фрейм занимает большую часть окна просмотра и предназначен для отображения самих документов. При первоначальной загрузке эти два фрейма делят окно просмотра по вертикали в соотношении 30% и 70%. Это соотношение может меняться при просмотре. Каждый из фреймов имеет свою полосу прокрутки, обеспечивающую просмотр всего документа. При выборе ссылки в левом фрейме соответствующий документ будет загружен в правый фрейм. Такая структура позволяет одновременно видеть на экране и оглавление документов, и сами документы.

Пусть оглавление документа содержит шесть пунктов и располагается в файле с именем **contents.htm**. Требуется, чтобы файл, содержащий оглавление, загружался в левый фрейм. Файлы с именами **ch1.htm**, **ch2.htm**, ..., **ch6.htm** содержат документы, соответствующие пунк-

там оглавления. Взаимодействие между фреймами и документами представлено на рисунке 5.

Фреймовая структура мало отличается от той, которая была рассмотрена в предыдущем примере.

ПЛАВАЮЩИЕ ФРЕЙМЫ



Обозреватель Microsoft Internet Explorer разрешает использование так называемых плавающих фреймов, описание которых может быть расположено в тексте обычного HTML-документа. Для определения плавающего фрейма используется тег **<iframe>**. В теге **<iframe>** могут встречаться те же параметры, что и в теге описания обычных фреймов, кроме параметра **NORESIZE**, так как размер плавающего фрейма не может быть изменен пользователем. Браузеры, не поддерживающие тег **<iframe>**, отображают информацию, записанную между тегами **<iframe>** и **</iframe>**. В следующем примере HTML-документа используется описание плавающего фрейма. Описание плавающе-

го фрейма напоминает описание встраиваемого в документ изображения. Параметр **src** определяет адрес загружаемого во фрейм изображения. Параметр **name** задает имя фрейма для того, чтобы можно было получить доступ к фрейму, например, из сценария.

СЦЕНАРИИ И ФРЕЙМЫ

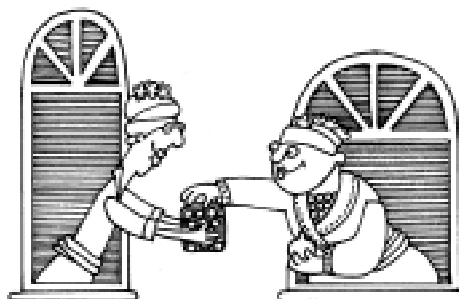


В рассмотренных примерах главы окно просмотра документов разбивалось по вертикали на два фрейма: левый с именем **left** и правый с именем **right**. Доступ к каждому из фреймов осуществлялся с помощью значения параметра **target**.

Фреймы образуют иерархическую модель объектов. На верхнем уровне расположен объект **top**, являющийся родителем фреймов. Для ссылки на фреймы можно использовать имена. Для того, чтобы

сослаться на левый фрейм надо написать **top.left**, для ссылки на правый фрейм следует написать **top.right**. Свойство фрейма **location** определяет URL-адрес загруженного во фрейм документа. Если при организации сайта используется небольшое число Web-страниц, то достаточно создать только меню верхнего уровня. Часто бывает удобно работать с меню двух уровней. В одном случае в левом фрейме представляется меню верхнего уровня, в другом случае все пункты меню раскрываются, то есть меню представляется в «раскрытом» виде.

ОБМЕН СОДЕРЖИМЫМ ФРЕЙМОВ



Создадим документ, разбивающий окно просмотра с помощью фреймов на три прямоугольные области. Верхняя область содержит два фрейма, нижняя область состоит из одного фрейма. В нижней области располагается кнопка, которая обменивает содержимое верхних фреймов.

Листинг 5 а. Фреймовая структура для обмена содержимого фреймов

```
<html>
<head>
<title> 3 фрейма с кнопкой для обмена</title>
</head>
<frameset rows="*,50">
  <frameset cols="55%,45%">
    <frame src=lpict.htm name=left>
    <frame src=rtext.htm name=right>
  </frameset>
<frame src=but1.htm name=butt>
</frameset>
</html>
```

Область окна просмотра сначала разбивается на два фрейма по горизонтали. При описании следующего тега определяется `<frameset rows="*,50">`, что размер нижнего фрейма 50 пикселей, остальное пространство отводится под верхний фрейм. Верхний фрейм, в свою очередь, разбивается по вертикали на два фрейма, размеры которых заданы в процентах к области просмотра. HTML-код, описывающий для данного случая фреймовую структуру, приведен в листинге 5 а.

Для ссылки на фреймы можно, как и ранее, использовать имена. Сначала следует запомнить имя файла, загруженного в левый фрейм. Для этого используется переменная `I`. Затем в левый фрейм загружается документ, расположенный в правом фрейме. Это достигается выполнением оператора присваивания

```
top.left.location=top.right.location
```

И, наконец, в правый фрейм загружается тот документ, адрес которого запомнили в переменной `I`.

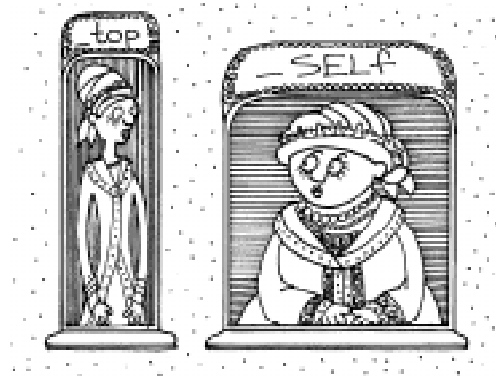
Для ссылки на фреймы страницы можно использовать свойство `frames` объекта `top`, в котором содержатся ссылки на все фреймы страницы. К адресу документа, загруженного в левый фрейм,

в нашем случае можно было обратиться следующим образом: `top.frames[0].location`, поэтому и функция, осуществляющая обмен содержимым фреймов, может быть описана следующим образом.

```
function chframe ()
{ var l=top.frames[0].location
  top.frames[0].location=top.frames[1].location
  top.frames[1].location=l }
```

HTML-код загружаемого в нижний фрейм документа приведен в листинге 5 б.

СТАНДАРТНЫЕ ИМЕНА ДЛЯ ФРЕЙМОВ



Имена фреймов должны начинаться с латинской буквы или цифры. Определены четыре специальных имени, при зада-

Листинг 5 б. Документ, содержащий сценарий для обмена содержимым

```
<HTML>
<HEAD>
<TITLE>Кнопка для смены содержимого фреймов</TITLE>
<script>
<!--//
function chframe ()
{ var l=top.frames[0].location
  top.frames[0].location=top.frames[1].location
  top.frames[1].location=l
}
//-->
</script>
</HEAD>
<body background="pap2.gif" bgcolor=silver>
<center>
<form name=form1>
<input type="button" value=обмен onclick="chframe () ">
</form>
</center></BODY></HTML>
```

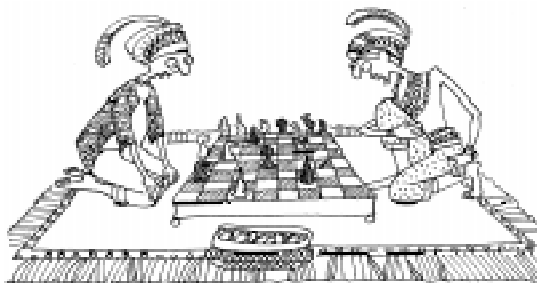
нии которых выполняются специальные действия. Эти имена начинаются с символов подчеркивания: "**_blank**", "**_self**", "**_top**", "**_parent**".

Использование имени **target="_blank"** обеспечит загрузку документа в новое окно. Для загрузки документа в текущий фрейм или окно следует использовать конструкцию имени **target="_self"**. Эта конструкция используется для того, чтобы обойти умолчание, задаваемое тегом **<BASE>**. Если требуется загрузить документ в полное окно, то можно воспользоваться **target="_top"**. Загрузка документа в область, занимаемую фреймом-родителем, происходит при выполнении **target="_top"**. Любое другое имя, начинающееся с символа подчеркивания, недопустимо.

Задания

1. Создайте документ, изображающий шахматную доску с расставленными на ней фигурами, используя таблицы HTML.

2. Создайте документ, разбивающий окно просмотра с помощью фреймов на три прямоугольные области. Верхняя область (А) занимает по ширине весь экран, нижняя область окна разбивается на две части: левую (В) и правую (С). В верхнюю область А поместите графическое изображение и информацию о себе. В область В поместите меню решенных задач, при выборе пунктов которого соответствующий документ должен появляться в области С. Меню может быть графическим или текстовым. Предусмотрите в документе ссылки на условия задач, которые появляются в новом окне.



Наши авторы, 2002.
Our authors, 2002.

*Дмитриева Марина Валерьевна,
доцент кафедры информатики
математико-механического
факультета Санкт-Петербургского
государственного университета.*

**КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ – ШКОЛЬНИКАМ РОССИИ.
РОССИЙСКАЯ ИНТЕРНЕТ-ШКОЛА ИНФОРМАТИКИ И
ПРОГРАММИРОВАНИЯ**

УСЛОВИЯ ОБУЧЕНИЯ

Программа интернет-школы информатики и программирования предусматривает многоуровневый цикл обучения и ориентирована на школьников 8–11-х классов, успешно справляющихся с базовой программой средней школы и имеющих отметки по информатике и математике не ниже четырех баллов.

Учащиеся интернет-школы самостоятельно осваивают теоретический материал и выполняют практические задания, имея возможность через Интернет обратиться за разъяснениями к высококвалифицированному педагогу. Предусмотрена возможность организации индивидуальных интернет-консультаций для учащихся школы.

Промежуточная аттестация будет проходить в режиме online одновременно для всех учащихся после завершения изучения каждой темы по методике автоматического тестирования программ, применяемой на чемпионатах мира по программированию.

В рамках работы школы для желающих будут проводиться специальные интернет-туры по программированию для подготовки учащихся к Всероссийским олимпиадам по информатике и программированию.

КАК ПОСТУПИТЬ В ИНТЕРНЕТ-ШКОЛУ

Зачисление в школу проводится по итогам собеседования и отборочного тестирования в учебных центрах Межвузовского подготовительного отделения в вашем регионе.

После оформления документов учащимся высылаются учебные пособия и программа занятий.

Дополнительную информацию о порядке поступления в интернет-школу можно получить на сайте: <http://www.mpo.spb.ru>

ПРОГРАММА ИНТЕРНЕТ-ШКОЛЫ

- Базовый курс информатики.
- Языки программирования (Паскаль, Си++).
- Алгоритмы программирования.
- Основы Web-программирования.
- Компьютерный дизайн.
- Компьютерная графика.
- Дизайн и реклама.

Учащимся, успешно прошедшим итоговую аттестацию, будут выданы свидетельства об окончании Российской интернет-школы информатики и программирования.

190068, Санкт-Петербург, Вознесенский пр., д. 25, оф. 30

Тел.: (812) 312-71-84, 315-66-96, 320-00-51

E-mail: tharchenko@yandex.ru

URL: <http://mpo.ifmo.ru>

координатор проекта:

Харченко Татьяна Владимировна

**ОЧЕРЕДНАЯ ЧЕТВЕРТАЯ
МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ
«КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ 2003»,
БЕСПЛАТНАЯ ШКОЛА-СЕМИНАР
«ВИРТУАЛЬНЫЕ ЛАБОРАТОРИИ
ДЛЯ ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН»
И СПЕЦИАЛЬНАЯ СЕКЦИЯ
«КОМПЬЮТЕРНОЕ ОБРАЗОВАНИЕ В ШКОЛЕ»**

**состоится
в Санкт-Петербургском Политехническом университете
в июне 2003 года.**

ПРОГРАММА КОНФЕРЕНЦИИ:

24-25 июня 2003 г. Специальная секция «Компьютерное образование в школе» и бесплатная школа-семинар «Виртуальные лаборатории для естественнонаучных дисциплин» для учителей школ.

26-28 июня 2003 г. Конференция «Компьютерное моделирование 2003», включающая в себя бесплатную школу-семинар «Виртуальные лаборатории для естественнонаучных дисциплин» для преподавателей университетов.

Предполагается издание сборника трудов конференции.

Приглашение и бланки заявок в электронной форме можно получить у Шиловой Татьяны Васильевны – shil@onti.tu.neva.ru

*Председатель организационного комитета
Сениченков Ю.Б.
sen@dcn.nord.nw.ru*