

Лабораторная работа №7

Флажки

Элемент управления "флажок" используется в случае, когда из предложенных вариантов можно выбрать как один, так и несколько. Каждый вариант выбора задается флажком, который можно либо установить, либо сбросить. Флажок определяется в теге `<input>` значением `checkbox` параметра `type`. Обязательным параметром является параметр `value`, значение которого будет передано на обработку в случае выбора нажатием кнопки.

Пример 1. Выбор характеристик издания

Предположим, читателю предлагается заполнить анкету, в которой требуется указать название любимого издания и выбрать из предложенного списка характеристики, которые присущи рассматриваемому изданию.

Для задания характеристик издания можно воспользоваться флажком. Пользователь устанавливает флажки для тех свойств, которыми, по его мнению, обладает издание. Обработка анкеты будет состоять в том, что выбранные свойства будут отражены в поле ввода многострочного текста.

При щелчке мышью по флажку возникает событие `click`, обработка которого состоит в вызове функции `set` с одним параметром, принимающим значение параметра `value` флажка. Для формирования строки результата служит глобальная переменная `s`; к имеющемуся значению добавляется значение параметра функции и помещается в текстовое поле. Если нажать ни кнопку Отмена, то очистятся все поля формы. Однако следует позаботиться о том, чтобы значение переменной `s` изменилось на начальное. Значение параметра реакции на событие `click` при щелчке по кнопке Отмена задается оператором присваивания, обеспечивающим начальные условия.

HTML-код представлен в листинге 1.

Листинг 1. Анкета читателя

```
<HTML>
<HEAD>
<TITLE>Анкета читателя</TITLE>
<script>
<!--
var s="Вас
привлекает: \r\n"
function set(vch)
{ s=s+vch + "\r\n"; document.form1.area.value=s }
//-->
</script>
</HEAD>
<BODY bgcolor="F8F8FF">
<CENTER>
```

```

<H3 align="center">Анкета читателя</H3>
<FORM name="form0">
<H4>Введите название любимого журнала или газеты</H4>
<input type="text" name="n1" size=45><br>
</FORM>
<FORM name="form1">
<H4>Что Вас привлекает в издании?</H4>
<TABLE border=3 align=center> <TR>
<TD></TD>
<TD><input type="checkbox" name="m1" value="Стиль
подачи материала"
onClick="set(form1.elements[0].value)">
Стиль подачи материала<br>
<input
type="checkbox" name="m2"
value="Достоверность информации"
onClick="set(form1.elements[1].value)">
Достоверность информации<br>
<input type="checkbox" name="m3"
value="Дизайн и оформление"
onClick="set(form1.elements[2].value)">
Дизайн и оформление<br>
</TD></TR></TABLE>
<textarea name="area" cols=35 rows=7> </textarea><br>
<input type="reset"
value="Отмена" onclick=
"s='Вас привлекает:
\r\n'">
</FORM>
</BODY>
</HTML>

```

В рассмотренных примерах значения параметра name флажков были различны, поскольку каждый флажок существовал независимо от других. Флажки можно объединить в группу. Для этого следует всем флажкам присвоить одно и то же значение параметра name.

Пример 2. Использование флажков в анкете переводчика

В анкете требуется указать те языки, которыми владеет переводчик. Предположим, что за знание каждого языка назначается определенная сумма. Размер вознаграждения определяется после заполнения анкеты в зависимости от тех языков, которыми пользователь владеет. По результатам заполненной переводчиком анкеты напишите сценарий определения размера вознаграждения.

Для задания сведений о том, владеет ли пользователь определенным языком, удобно применять флажок. При щелчке мышью по кнопке Вознаграждение выполняется функция grant(). Требуется проанализировать состояние флажков. Свойство checked возвращает логическое значение,

представляющее текущее значение отдельного флажка (true или false).
Воспользуемся тем, что каждый объект form имеет свойство-массив elements,

получим доступ к каждому флажку формы. Состояние первого флажка можно определить с помощью следующей конструкции:

```
(document.form1.elements[0]).checked  
второго -  
(document.form1.elements[1]).checked
```

и т. д. В переменной k накапливается сумма. Шаг увеличения этой переменной задается в качестве значения параметра value. После анализа всех флажков полученная сумма выводится в документ.

HTML-код представлен в листинге 2.

Листинг 2. Данные, представленные флажком. Анкета переводчика

```
<HTML>  
<HEAD>  
<TITLE>Данные, представленные флажком. Анкета  
переводчика</TITLE>  
<script language="JavaScript">  
<!-- //  
function grant()  
{ var d= document var k=0;  
if ((d.form1.elements[0]).checked)  
k=k+Number(d.form1.elements[0].value) if  
((d.form1.elements[1]).checked)  
k=k+Number(d.form1.elements[1].value) if  
((d.form1.elements[2]).checked)  
k=k+Number(d.form1.elements[2].value)  
form1.ww.value="Вам полагается вознаграждение "+k+"  
y.e."  
}  
//-->  
</script>  
</HEAD>  
<BODY>  
<H3>Анкета для переводчиков</H3>  
Укажите те языки, которыми Вы владеете в совершенстве:  
<br>  
<FORM name="form1">  
<input type="checkbox" name="lan" value=100>русский<br>  
<input type="checkbox" name="lan"  
value=200>английский<br>  
<input type="checkbox" name="lan"  
value=300>французский<br>  
<input type="button" value=Вознаграждение  
onClick="grant()"> <hr>  
<input type="Text" size=50 name="ww" value=""><br>  
<input type="reset" value="Отменить">  
</FORM><hr>
```

```
</BODY>  
</HTML>
```

Списки

Если элементов много, то представление их с помощью флажков или переключателей увеличивает размер формы. В этом случае варианты выбора могут быть представлены в окне браузера более компактно с помощью тега `<select>`. Тег имеет несколько параметров. Параметр `name` является обязательным. Для того чтобы установить число одновременно видимых элементов, следует задать параметр `size=n`. Когда `n` равно 1, то отображается ниспадающее меню или список выбора; при `n>1` выводится список с `n` одновременно видимыми значениями. Если параметр `size` не задан, то по умолчанию принимается значение равное 1. Указание параметра `multiple` означает, что из меню или списка можно выбрать несколько элементов. Элементы меню задаются внутри тега `<select>` с помощью тега `<option>`. Общий вид тега таков:

```
<option selected value=строка>
```

Параметр `selected` означает, что данный элемент списка считается выбранным по умолчанию. Параметр `value` содержит значение, которое передается, если данный элемент выбран из списка или меню.

Пример 1. Обработка анкеты переводчика

Напишем сценарий обработки анкеты переводчика. Сведения о тех языках, которыми владеет переводчик, требуется задать с помощью списка. Выбранные языки следует помещать в поле ввода многострочного текста.

Напомним, что событие `change` происходит в тот момент, когда значение элемента формы `text`, `select` или `textarea` изменилось, и элемент потерял фокус. Будем обрабатывать анкету переводчика следующим образом. Параметр обработки события поместим в тег `<select>`. Как только выбран конкретный язык, т. е. произошло событие `change`, выполняется функция `gr`:

```
function gr(obj,m)
{ var r=100*(Number((obj.elements[0])[m]).value)+1)
s+=((obj.elements[0])[m]).text+"\r\n"
obj.restext.value=s
sum+=r obj.res.value=r}
```

Первый параметр - имя формы, второй - значение параметра `value` выбранного элемента. Второй оператор обеспечивает формирование строки всех выбранных пользователем языков. Третий оператор помещает вычисленное значение в текстовое поле. В результате выполнения четвертого оператора присваивания в переменной `sum` формируется значение, которое, затем при нажатии кнопки Сумма будет выведено в текстовое поле.

Последний оператор помещает значение для выбранного языка в соответствующее поле формы. Полностью документ со сценарием и формами может быть описан так, как указано в листинге 1.

Листинг 1. Реакция на событие change в теге <select>

```
<HTML>
<HEAD>
<TITLE>Реакция на событие Change в теге select</TITLE>
<script language="JavaScript">
<!-- // var s=""
var sum=0 function gr(obj,m)
{ var r=100*(Number((obj.elements[0])[m]).value)+1)
s+=((obj.elements[0])[m]).text+"\r\n"
obj.restext.value=s
sum+=r obj.res.value=r
}
//-->
</script>
</HEAD>
<BODY>
<FORM name="form1">
<H3>Анкета переводчика</H3>
<TABLE border=3 bgcolor=silver>
<TR><TH>Выбранный язык</TH><TH>Результат</TH></TR>
<TR>
<TD valign=top>
<select name="data" size=3
onChange="gr(form1,form1.data.value)">
<option value=0>русский
<option value=1>английский
<option value=2>французский
</select><P>
<input type="text" name="res" size=15>
</TD>
<TD><TEXTAREA name="restext" cols=15 rows=6>
</TEXTAREA><P>
<input type="button" value=Сумма
onClick="form1.resgr.value=sum">
<input type="text" name="resgr" size=10>
</TD></TR></TABLE><p>
<input type="reset" value="Отменить" onClick="sum=0;
s=' '>
</FORM>
</BODY>
```

</HTML>

Пример 2. Тест "Города и памятники"

Напишем сценарий обработки теста "Города и памятники". Названия городов и памятников задаются с помощью списков. Пользователь выбирает в левом перечне название города, а в правом - памятник, расположенный в этом городе. После нажатия кнопки Результат в текстовое поле выводится количество правильных ответов.

В сценарии используются три глобальные переменные. Переменная q хранит последнее выбранное значение в левом столбце; переменная a - выбранное значение правого столбца; значение переменной sum содержит число правильных ответов. В двух списках для правильной пары "вопрос/ответ" совпадают соответствующие значения параметра value. Эти значения проверяются после выбора элемента списка правого столбца. Результат тестирования можно посмотреть, если нажать кнопку Результат.

Сценарий, реализующий простую обработку теста, представлен в листинге 2.

Листинг 2. Простая тестирующая программа

```
<HTML>
<HEAD>
<TITLE>Города и памятники.</TITLE>
<script>
<!--
var sum=0 var q
var a function eq()
{ q=test.question.value a=test.answer.value;
if (a==q) sum +=1
}
function result()
{ document.test.res.value=sum}
//-->
</script>
<BODY background="fon3.gif">
<h3 align=center>Города и памятники</h3>
<FORM name="test">
<TABLE border=3 align=center cellpadding=5
cellspacing=6 bgcolor= silver>
<TR><TH>Памятник</TH><TH>Находится в городе</TH>
<TR><TD>
```

```

<select size =7 name="question"
onChange="q=test.question.value">
<option value="mad">Музей Прадо<br>
<option value="ber" >Рейхстаг<br>
<option value="mil">Оперный театр Ла Скала<br>
<option value="ier">Стена Плача<br>
<option value="mek">Священный камень Кааб<br>
<option value="spb">Медный Всадник<br>
<option value="mos">Третьяковская галерея<br>
<option value="par">Триумфальная Арка<br>
<option value="new">Статуя Свободы<br>
<option value="lon">Тауэр<br>
</select>
</TD>
<TD>
<select size=7 name="answer" onChange="eq()" >
<option value="spb">Санкт-петербург<br>
<option value="mos">Москва<br>
<option value="mek">Мекка<br>
<option value="ier">Иерусалим<br>
<option value="mil">Милан<br>
<option value="par">Париж<br>
<option value="mad">Мадрид<br>
<option value="lon">Лондон<br>
<option value="new">Нью-Йорк<br>
<option value="ber">Берлин<br>
</select>
</TD></TR>
</TABLE>
<CENTER><P>
<input type="button" value="Результат"
onclick="result()"><br> Количество правильных ответов
<input type="text" name="res" size="5"><P>
<input type="reset" value="Обновить" onclick="sum=0">
</FORM>
</BODY>
</HTML>

```

Фреймы

Окно просмотра браузера можно разбить на несколько прямоугольных областей, называемых фреймами. Области соприкасаются друг с другом и в каждую из областей можно загрузить отдельный HTML-документ, и работать с ним 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 следует отобразить информацию, которая должна постоянно находиться на экране.

Пример 1. 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 загрузим в правый фрейм. Пусть имя файла, содержащего оглавление - contents.htm, а 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 HTML-код, представленный в листинге 1.

Листинг 1. 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>
```

Параметр cols тега <frameset> имеет вид cols="список значений". В списке через запятую перечисляются значения, которые определяют размеры фреймов. Список должен содержать не менее двух значений. Значения могут задаваться в процентах, в пикселах, в относительных единицах.

Тег `<frame>` определяет один фрейм. Он должен располагаться внутри парного тега `<frameset>` и `</frameset>`. Число тегов `<frame>` должно совпадать с количеством тегов, определенных при описании фреймовой структуры. В рассматриваемом примере в теге `<frameset cols="25%,75%">` определено два фрейма, поэтому в дальнейшем следует описание каждого из фреймов с помощью тега `<frame>`.

Значение параметра `src` тега `<frame>` определяет адрес документа, который первоначально загружается во фрейм. В рассматриваемом случае в левый фрейм загружается документ с именем `contents0.htm`, а в правый фрейм - документ с именем `ch.htm`. В теге параметр `name` определяет имя фрейма, необходимое для указания, в какой фрейм загрузить документ. Если имя фрейма не задавать, то будет создан фрейм без имени, но сослаться на него из других фреймов будет нельзя.

Пример 2. Фреймовая структура с загружаемыми документами

Создадим документ, левая часть которого представляет оглавление, а в правую часть загружаются документы выбранного пункта оглавления. Документы, соответствующие пунктам оглавления, хранятся в разных файлах.

При решении задачи экран по-прежнему разбивается на два фрейма. Левый фрейм занимает 30% ширины всего окна и будет содержать оглавление документов, которые могут быть просмотрены пользователем при выборе соответствующего пункта. Правый фрейм занимает большую часть окна просмотра и предназначен для отображения самих документов. При первоначальной загрузке оба фрейма делят окно просмотра по вертикали в соотношении 30% и 70%. Данное 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 пунктам оглавления.

Фреймовая структура мало отличается от той, которая была рассмотрена в предыдущем примере (листинг 2, а).

Листинг 2, а. Задание фреймовой структуры

```
<HTML>
<HEAD>
<TITLE>Простая фреймовая структура</TITLE>
</HEAD> <frameset cols="30%,70%">
```

```
<frame src=contents.htm name=left>
<frame src=empty.htm name=right>
</frameset>
</HTML>
```

В правый фрейм первоначально загружается файл с именем empty.htm. Если сразу неизвестно, какой файл загружать во фрейм, то можно использовать файл, содержащий HTML-код (листинг 2, б).

Листинг 2, б. Документ для первоначальной загрузки

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

В левый фрейм помещается Оглавление, которое содержит ссылки на документы, расположенные в различных файлах. Оглавление может быть сформировано так, как указано в листинге 2, в.

Листинг 2, в. Оглавление, загружаемое в левый фрейм

```
<HTML>
<HEAD>
<TITLE>Оглавление</TITLE>
</HEAD>
<BODY background="decor.gif" bgcolor=silver>
<base target=right>
<h3>Оглавление</h3>
<OL>
<LI><A href="ch1.htm">Основы языка HTML </A>
<LI><A href="ch2.htm">Графика </A>
<LI><A href="ch3.htm">Изображение-карта </A>
<LI><A href="ch4.htm">Списки </A>
<LI><A href="ch5.htm">Таблицы </A>
<LI><A href="ch6.htm">Фреймы </A>
</OL>
</BODY>
</HTML>
```

Пример 3. 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.

Область окна просмотра сначала разбивается на два фрейма по горизонтали. Параметр `rows="список значений"` задает количество и размеры фреймов по горизонтали. При описании тега `<frameset rows="*,50">` определяется, что размер нижнего фрейма 50 пикселей, остальное пространство отводится под верхний фрейм. Верхний фрейм в свою очередь делится по вертикали на два фрейма, размеры которых заданы в процентах к области просмотра. HTML-код документа, создающий описанную фреймовую структуру, выглядит так, как указано в листинге 3, а.

Листинг 3, а. Три фрейма с кнопкой для обмена содержимого фреймов

```
<HTML>
<HEAD>
<TITLE>Три 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>
```

Для ссылки на фреймы будем, как и ранее, использовать имена. Сначала следует запомнить имя файла, загруженного в левый фрейм. Для этого используется переменная `l`. 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 `l`.

HTML-код для нижнего фрейма представлен в листинге 3, б.

Листинг 3, б. Сценарий для нижнего фрейма

```
<HTML>
<HEAD>
<TITLE>Кнопка для смены содержимого фреймов</TITLE>
```

```

<script>
<!--//
function chframe()
{ var l=top.left.location
top.left.location=top.right.location
top.right.location=l
}
//-->
</script>
</HEAD>
<BODY background="decor.gif" bgcolor=silver>
<CENTER>
<FORM name=form1>
<input type="button" value=Обмен onclick="chframe()">
</FORM>
</CENTER>
</BODY>
</HTML>

```

При щелчке по кнопке Обмен содержимое верхних фреймов поменяется местами.

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
}

```

Браузер 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 указанном теге могут встречаться те же параметры, что и в теге описания обычных фреймов, кроме параметра noresize, т. к. размер плавающего фрейма не может быть изменен пользователем. Браузеры, не поддерживающие тег <iframe>, отображают записанную между тегами <iframe> и </iframe> информацию.

Для плавающих фреймов с помощью параметров можно задавать размеры фрейма, горизонтальное выравнивание, размер отступа содержимого фрейма от границ. В следующем примере приведено описание плавающего фрейма и заданы параметры. Фрейм имеет высоту 320 пикселей, занимает по ширине 60% окна, расположен справа от текста, полосы прокрутки будут установлены в случае, если документ не станет помещаться во фрейм. Содержимое фрейма должно быть отделено от границы по

горизонтали и вертикали на заданное число пикселей.

```
<iframe src=pictsh.htm name="flframe" height=320  
width=60% hspace=50 vspace=10 scrolling=auto  
align=right>
```

Задания. Сайт разработчика

1. Из ранее созданного резюме создайте html документ, разбивающий окно просмотра с помощью фреймов на две области (прямоугольные или вертикальные): верхнюю и нижнюю. В верхней области поместите оглавление в виде списка (резюме, контакты, репетитор) при выборе пунктов которого соответствующий раздел должен появляться в нижней части окна.

Оглавление представьте с помощью графического вертикального меню.

2. На странице «Репетитор» напишите сценарий обработки выбранных параметров для ученика. Пользователь может:

- выбрать курс (HTML, JavaScript, Си++) – использовать select
- количество занятий в неделю
- форма занятий (очная, дистанционная) – использовать checkbox

В зависимости от этих параметров определяется стоимость обучения в месяц.