**Лекция 7**

Тема. **Фреймы. Формы. Типы элементов формы.**

**Конспект**

**Фреймы -** способ организации сайта, при котором Web - страница дробится на ряд отдельных составляющих и собирается в главном окне браузера из нескольких независимых окон.При использовании фреймов раздел <BODY>заменяется на раздел<FRAMESET>.

В общем виде синтаксис записи <FRAMESET> следующий:

<FRAMESET rows = “” cols = “” frameborder = 1|0 framespacing = “”>

<FRAME name = ”” target = “” scrolling = yes|no| auto src = “URL” frameborder = 1|0 noresize>

……..

<NOFRAMES>

<BODY>

Текст, отображаемый в браузерах, не поддерживающих фреймы

</BODY>

</NOFRAMES>

</FRAMESET>

Атрибуты: frameborder – отображать или нет на экране границы фреймов;

framespacing – толщина разделителей в пикселях;

rows, cols – задают соответственно количество горизонталей и вертикалей во фреймовом наборе в пикселях, процентах, и относительных длинах;

Если не указан rows, то каждый столбец будет занимать всю длину страницы, если cols не указан, каждая строка растягивается на всю видимую ширину экрана.

scrolling – наличие полос прокрутки;

name – уникальное имя для данного конкретного фрейма;

src - адрес страницы, открываемой в данном фрейме;

target – информация о целевом фрейме (рекомендуется использовать значение по умолчанию “contents”)

noresize – запрет на изменение пользователем размеров окна фрейма.

<NOFRAMES>, включающий теги <BODY></BODY>, позволяет записать произвольное количество html - кода, который динамически преобразуется в самостоятельную Web - страницу в случае, если клиентский браузер не поддерживает фреймы.

Тег <FRAMESET>, задающий параметры отображения всех фреймов в документе содержит необходимое количество тегов <FRAME>, описывающих каждый отдельный фрейм. Атрибуты в <FRAME>могут отменять параметры в атрибутах тега <FRAMESET>. Допускается вложение одного элемента <FRAMESET> в другой.

**Форма** - это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку внешнего мира, где информация будет некоторым образом обработана. Создать форму гораздо проще, чем ту "точку внешнего мира", в которую форма будет посылать информацию. В качестве такой "точки" в большинстве случаев выступает программа, написанная на Перл или Си. Программы, обрабатывающие данные, переданные формами, часто называют CGI-скриптами. Сокращение CGI (Common Gateways Interface) означает "общепринятый интерфейс шлюзов". Написание CGI-скриптов в большинстве случаев требует хорошего знания соответствующего языка программирования и возможностей операционной системы. В последнее время определенное распространение получил язык PHP/FI, инструкции которого можно встраивать прямо в HTML-документы (документы при этом сохраняются в виде файлов с расширением \*.pht или \*.php). Формы передают информацию программам-обработчикам в виде пар [имя переменной]=[значение переменной]. Имена переменных следует задавать латинскими буквами. Значения переменных воспринимаются обработчиками как строки, даже если они содержат только цифры.

Форма открывается меткой <FORM> и заканчивается меткой </FORM>. HTML-документ может содержать в себе несколько форм, однако формы не должны находиться одна внутри другой. HTML-текст, включая метки, может размещаться внутри форм без ограничений.

Метка <FORM> может содержать три атрибута, один из которых является обязательным. Вот эти атрибуты:

ACTION

Обязательный атрибут. Определяет, где находится обработчик формы.

METHOD

Определяет, каким образом (иначе говоря, с помощью какого метода протокола передачи гипертекстов) данные из формы будут переданы обработчику. Допустимые значения: METHOD=POST и METHOD=GET. Если значение атрибута не установлено, по умолчанию предполагается METHOD=GET.

ENCTYPE

Определяет, каким образом данные из формы будут закодированы для передачи обработчику. Если значение атрибута не установлено, по умолчанию предполагается ENCTYPE=application/x-www-form-urlencoded.

**Простейшая форма.** Для того, чтобы запустить процесс передачи данных из формы обработчику, нужен какой-то орган управления. Создать такой орган управления очень просто: <INPUT TYPE=submit> Встретив такую строчку внутри формы, браузер нарисует на экране кнопку с надписью Submit (читается "сабмит" с ударением на втором слоге, от английского "подавать"), при нажатии на которую все имеющиеся в форме данные будут переданы обработчику, определенному в метке <FORM>. Надпись на кнопке можно задать такую, какая нравится, путем введения атрибута VALUE="[Надпись]" (читается "вэлью" с ударением на первом слоге, от английского "значение"), например:

<INPUTTYPE=submitVALUE="Поехали!">Теперь достаточно для того, чтобы написать простейшую форму. Она не будет собирать никаких данных, а просто вернет нас к тексту этой главы.

<HTML>

<HEAD>

<TITLE>Примерформы</TITLE>

</HEAD>

<H1>Простейшая форма </H1>

<FORM ACTION="pr0008.htm"><!--Это начало формы-->

<INPUT TYPE=submit VALUE="Назад, кглаве 8...">

</FORM><!--Это конец формы-->

</BODY>

</HTML>

Надпись, нанесенную на кнопку, можно при необходимости передать обработчику путем введения в определение кнопки атрибута NAME=[имя] (читается "нэйм", от английского "имя"), например:

<INPUT TYPE=submit NAME=button VALUE="Поехали!">При нажатии на такую кнопку обработчик вместе со всеми остальными данными получит и переменную button со значением Поехали!.

В форме может быть несколько кнопок типа submit с различными именами и/или значениями. Обработчик, таким образом, может действовать по-разному в зависимости от того, какую именно кнопку submit нажал пользователь.

Существуют и другие типы элементов <INPUT>. Каждый элемент <INPUT> должен включать атрибут NAME=[имя], определяющий имя элемента (и, соответственно, имя переменной, которая будет передана обработчику). Имя должно задаваться только латинскими буквами. Большинство элементов <INPUT> должны включать атрибут VALUE="[значение]", определяющий значение, которое будет передано обработчику под этим именем. Для элементов <INPUT TYPE=text> и <INPUT TYPE=password>, однако, этот атрибут не обязателен, поскольку значение соответствующей переменной может вводиться пользователем с клавиатуры.

Основные типы элементов <INPUT>:

TYPE=text

Определяет окно для ввода строки текста. Может содержать дополнительные атрибуты SIZE=[число] (ширина окна ввода в символах) и MAXLENGTH=[число] (максимально допустимая длина вводимой строки в символах). Пример:

<INPUT TYPE=text SIZE=20 NAME=user VALUE="Иван">

Определяет окно шириной 20 символов для ввода текста. По умолчанию в окне находится текст Иван, который пользователь может редактировать. Отредактированный (или неотредактированный) текст передается обработчику в переменной user.

TYPE=password

Определяет окно для ввода пароля. Абсолютно аналогичен типу text, только вместо символов вводимого текста показывает на экране звездочки (\*). Пример:

<INPUT TYPE=password NAME=pw SIZE=20 MAXLENGTH=10>

Определяет окно шириной 20 символов для ввода пароля. Максимально допустимая длина пароля — 10 символов. Введенный пароль передается обработчику в переменной pw.

TYPE=radio

Определяет радиокнопку. Может содержать дополнительный атрибут checked (показывает, что кнопка помечена). В группе радиокнопок с одинаковыми именами может быть только одна помеченная радиокнопка. Пример:

<INPUT TYPE=radio NAME=modem VALUE="9600" checked> 9600 бит/с<INPUT TYPE=radio NAME=modem VALUE="14400"> 14400 бит/с<INPUT TYPE=radio NAME=modem VALUE="28800"> 28800 бит/с

Определяет группу из трех радиокнопок, подписанных 9600 бит/с, 14400 бит/с и 28800 бит/с. Первоначально помечена первая из кнопок. Если пользователь не отметит другую кнопку, обработчику будет передана переменная modem со значением 9600. Если пользователь отметит другую кнопку, обработчику будет передана переменная modem со значением 14400 или 28800.

TYPE=checkbox

Определяет квадрат, в котором можно сделать пометку. Может содержать дополнительный атрибут checked (показывает, что квадрат помечен). В отличие от радиокнопок, в группе квадратов с одинаковыми именами может быть несколько помеченных квадратов. Пример:

<INPUT TYPE=checkbox NAME=comp VALUE="PC"> Персональные компьютеры <INPUT TYPE=checkbox NAME=comp VALUE="WS" checked> Рабочие станции <INPUT TYPE=checkbox NAME=comp VALUE="LAN"> Серверы локальных сетей <INPUT TYPE=checkbox NAME=comp VALUE="IS" checked> Серверы Интернет

Определяет группу из четырех квадратов. Первоначально помечены второй и четвертый квадраты. Если пользователь не произведет изменений, обработчику будут переданы две переменные: comp=WS и comp=IS.

TYPE=hidden

Определяет скрытый элемент данных, который не виден пользователю при заполнении формы и передается обработчику без изменений. Такой элемент иногда полезно иметь в форме, которая время от времени подвергается переработке, чтобы обработчик мог знать, с какой версией формы он имеет дело. Другие возможные варианты использования Вы вполне можете придумать сами. Пример:

<INPUT TYPE=hidden NAME=version VALUE="1.1">

Определяет скрытую переменную version, которая передается обработчику со значением 1.1.

TYPE=reset

Определяет кнопку, при нажатии на которую форма возвращается в исходное состояние. Поскольку при использовании этой кнопки данные обработчику не передаются, кнопка типа reset может и не иметь атрибута name. Пример:

<INPUT TYPE=reset VALUE="Очистить поля формы">

Определяет кнопкуОчистить поля формы, при нажатии на которую форма возвращается в исходное состояние.

Помимо элементов <INPUT>, формы могут содержать меню <SELECT> и поля для ввода текста <TEXTAREA>.

Меню <SELECT> из n элементов выглядит примерно так:

<SELECTNAME="[имя]">

<OPTIONVALUE="[значение 1]">[текст 1]

<OPTIONVALUE="[значение 2]">[текст 2]

...

<OPTIONVALUE="[значениеn]">[текстn]

</SELECT>

Как Вы видите, меню начинается с метки <SELECT> и заканчивается меткой </SELECT>. Метка <SELECT> содержит обязательный атрибут NAME, определяющий имя переменной, которую генерирует меню.

Метка <SELECT> может также содержать атрибут MULTIPLE, присутствие которого показывает, что из меню можно выбрать несколько элементов. Большинство браузеров показывают меню <SELECT MULTIPLE> в виде окна, в котором находятся элементы меню (высоту окна в строках можно задать атрибутом SIZE=[число]). Меню <SELECT> в большинстве случаев показывается в виде выпадающего меню.

Метка <OPTION> определяет элемент меню. Обязательный атрибут VALUE устанавливает значение, которое будет передано обработчику, если выбран этот элемент меню. Метка <OPTION> может включать атрибут checked, показывающий, что данный элемент отмечен по умолчанию. Пример.

<SELECT NAME="selection">

<OPTION VALUE="option1" checked>Вариант 1

<OPTION VALUE="option2">Вариант 2

<OPTION VALUE="option3">Вариант 3

</SELECT>

Такой фрагмент определяет меню из трех элементов: Вариант 1, Вариант 2 и Вариант 3. По умолчанию выбран элемент Вариант 1. Обработчику будет передана переменная selectionзначение которой может быть option1 (по умолчанию), option2 или option3.

Элемент <TEXTAREA> может показаться совсем простым. Например:

<TEXTAREA NAME=address ROWS=5 COLS=50>

А здесь - Ваш адрес...

</TEXTAREA>

Все атрибуты обязательны. Атрибут NAME определяет имя, под которым содержимое окна будет передано обработчику (в примере — address). Атрибут ROWS устанавливает высоту окна в строках (в примере — 5). Атрибут COLS устанавливает ширину окна в символах (в примере — 50).

Текст, размещенный между метками <TEXTAREA> и </TEXTAREA>, представляет собой содержимое окна по умолчанию. Пользователь может его отредактировать или просто стереть. Важно знать, что русские буквы в окне <TEXTAREA> при передаче обработчику могут быть конвертированы в соответствующие им символьные объекты.

**Основная литература:**1 [20-39, 76-96]; 5 [21-40]; 6 [13-32]; 8 [31-50]; 9 [25-47].

**Дополнительная литература:**12 [21-50]; 13 [15-30].

**Контрольные вопросы:**

1. Какие существуют форматы звуковых файлов?
2. Какие существуют форматы видеофайлов?
3. Как встраиваются в страницу видеоклипы?
4. Как встраиваются в страницу аудио- и видеоклипы?
5. Какие существуют форматы звуковых и видеофайлов?
6. Как определяются фреймы? Назначение формы.Какие бывают элементы формы?