**Практическая работа 10**

**Создание фреймов в JavaScript**

**Цель занятия:** Повторить приемы создания документов, содержащих фреймы. Ознакомиться с принципами использования адресации для фреймов

**Задания:**

1. Ознакомиться с теоретическими аспектами темы.
2. Создайте веб-страницу с несколькими ссылками в одном фрейме.
3. Создайте веб-страницу, в которой в одном фрейме создайте несколько ссылок, но если посетитель активирует какую-либо из них, соответствующая страница будет помещена не в тот же самый фрейм, а в соседний.

**Необходимые приборы:** ПК, текстовый редактор Блокнот, браузер.

**Методические рекомендации к выполнению лабораторной работы:**

**Методические рекомендации к выполнению задания 1**

В общем случае окно браузера может быть разбито в несколько отдельных фреймов. Это означает, что фрейм определяется как некое выделенное в окне браузера поле в форме прямоугольника. Каждый из фреймов выдает на экран содержимое собственного документа. Таким образом, можно, к примеру, создать два фрейма. В первый такой фрейм загрузить "домашнюю страницу" фирмы Netscape, а во второй - фирмы Microsoft. Для создания фреймов необходимы два тэга: <frameset> и <frame>. HTML-страница, создающая два фрейма, в общем случае может выглядеть следующим образом:

<html>

<frameset rows="50%,50%">

<frame src="page1.htm" name="frame1">

<frame src="page2.htm" name="frame2">

</frameset>

</html>

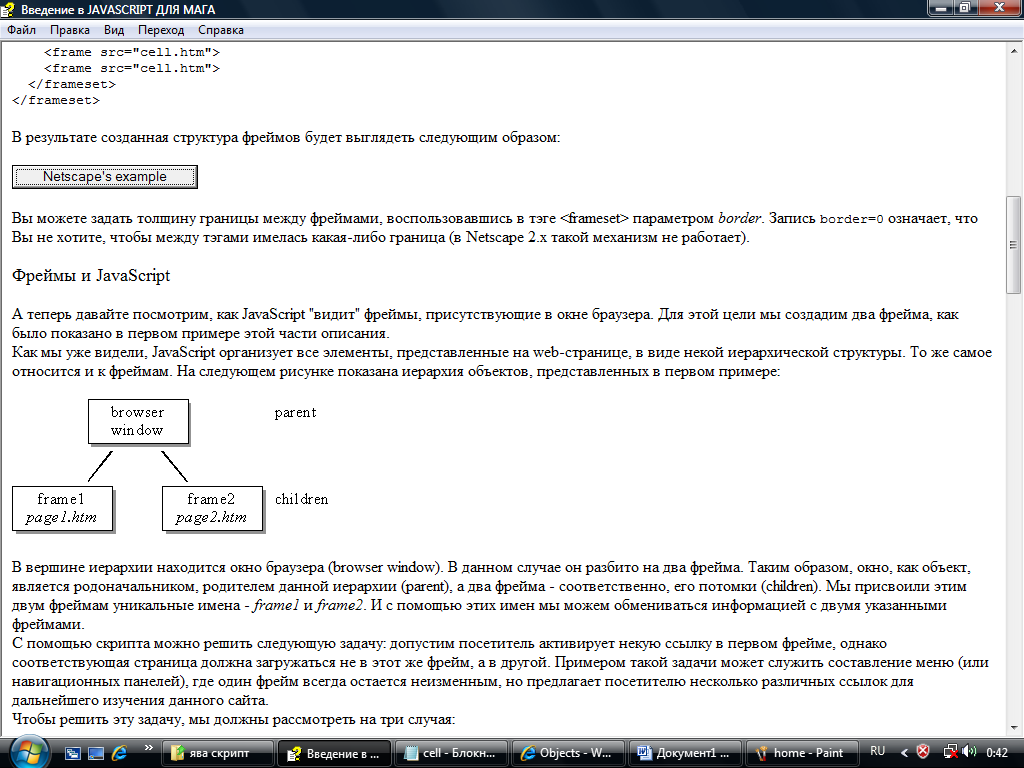
В результате будут созданы два фрейма. Во фрейме <frameset> мы используем свойство *rows*. Это означает, два фрейма будут расположены друг над другом. В верхний фрейм будет загружена HTML-страница *page1.htm*, а в нижнем фрейме разместится документ *page2.htm*.

Если нужно, чтобы документы располагались не друг над другом, а рядом, то следует в тэге <frameset> писать не *rows*, а *cols*. Фрагмент *"50%,50%"* сообщает, насколько велики должны быть оба получившихся окна. Вы имеете также возможность записать *"50%,\*"*, если не хотите утруждать себя расчетами, насколько велик должен быть второй фрейм, чтобы в сумме получалась все те же 100%. Вы можете также задать размер фрейма в пикселях, для чего достаточно после числа не ставить символ %.

Любому фрейму можно присвоить уникальное имя, воспользовавшись в тэге <frame> атрибутом *name*. Такая возможность пригодится в языке JavaScript для доступа к фреймам.

При создании web-страниц можно использовать несколько вложенных тэгов <frameset>.

JavaScript организует все элементы, представленные на web-странице, в виде некой иерархической структуры. То же самое относится и к фреймам. На следующем рисунке показана иерархия объектов, представленных в первом примере:



В вершине иерархии находится окно браузера (browser window). В данном случае он разбито на два фрейма. Таким образом, окно, как объект, является родоначальником, родителем данной иерархии (parent), а два фрейма - соответственно, его потомки (children). Мы присвоили этим двум фреймам уникальные имена - *frame1* и *frame2*. И с помощью этих имен можно обмениваться информацией с двумя указанными фреймами.

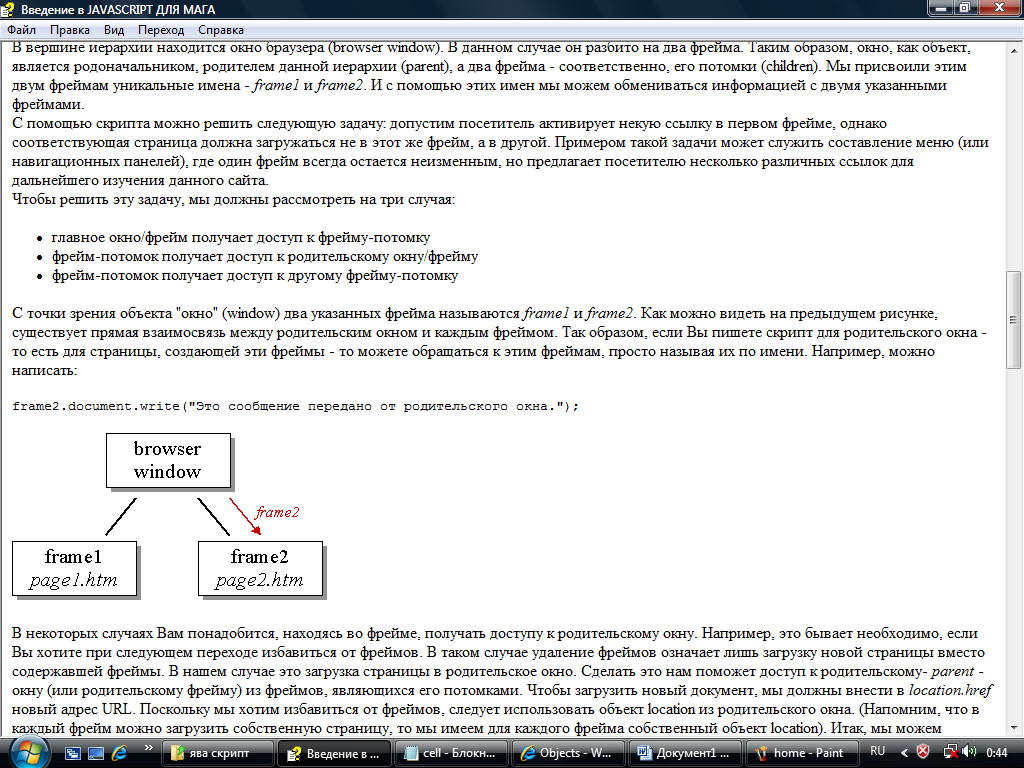
С помощью скрипта можно решить следующую задачу: допустим посетитель активирует некую ссылку в первом фрейме, однако соответствующая страница должна загружаться не в этот же фрейм, а в другой. Примером такой задачи может служить составление меню (или навигационных панелей), где один фрейм всегда остается неизменным, но предлагает посетителю несколько различных ссылок для дальнейшего изучения данного сайта.

Чтобы решить эту задачу, нужно рассмотреть на три случая:

* главное окно/фрейм получает доступ к фрейму-потомку
* фрейм-потомок получает доступ к родительскому окну/фрейму
* фрейм-потомок получает доступ к другому фрейму-потомку

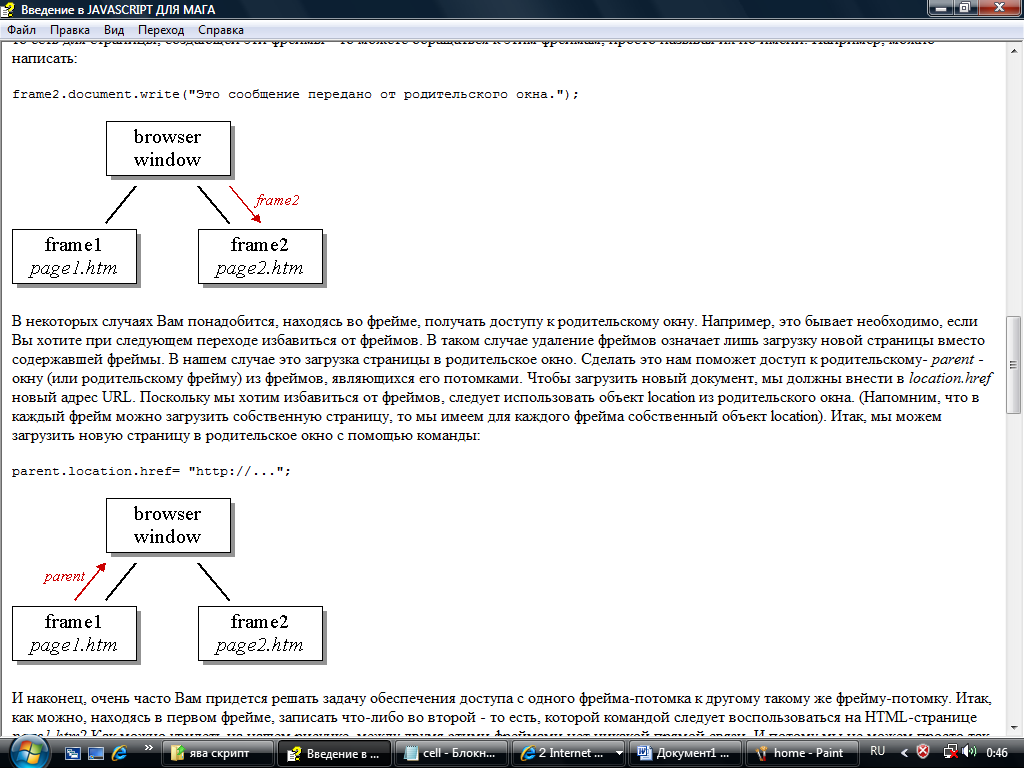
С точки зрения объекта "окно" (window) два указанных фрейма называются *frame1* и *frame2*. Как можно видеть на предыдущем рисунке, существует прямая взаимосвязь между родительским окном и каждым фреймом. Так образом, если Вы пишете скрипт для родительского окна - то есть для страницы, создающей эти фреймы - то можете обращаться к этим фреймам, просто называя их по имени. Например, можно написать:

frame2.document.write("Это сообщение передано от родительского окна.");



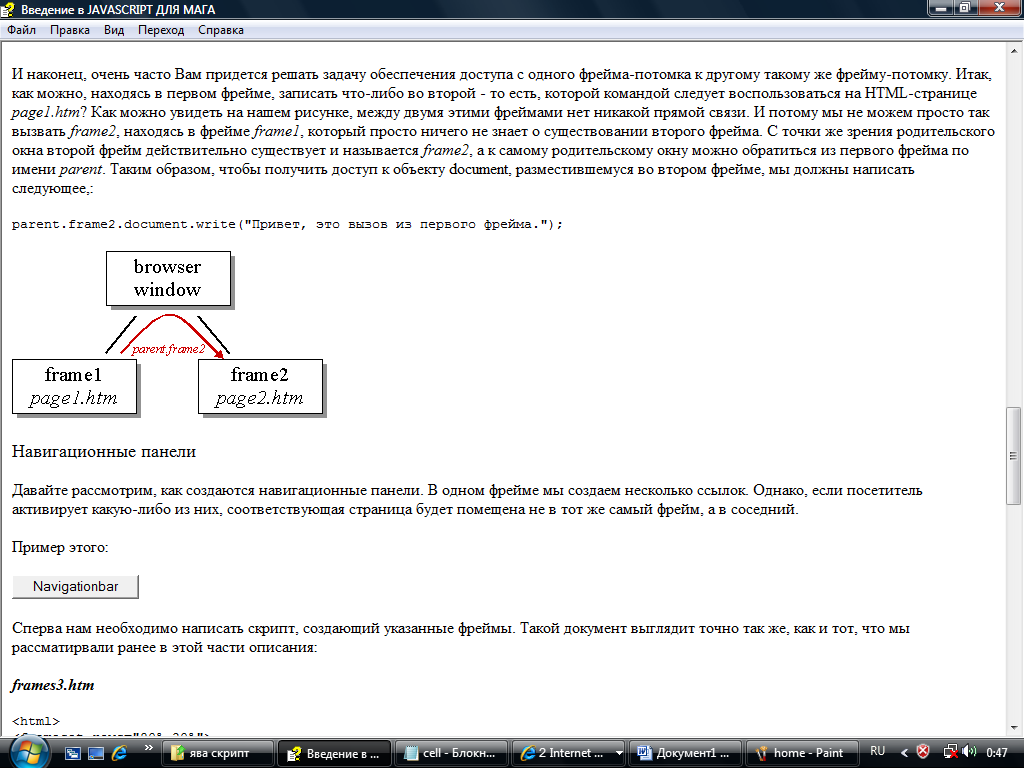
В некоторых случаях понадобится, находясь во фрейме, получать доступу к родительскому окну. Например, это бывает необходимо, если нужно при следующем переходе избавиться от фреймов. В таком случае удаление фреймов означает лишь загрузку новой страницы вместо содержавшей фреймы. В нашем случае это загрузка страницы в родительское окно. Сделать это нам поможет доступ к родительскому- *parent* - окну (или родительскому фрейму) из фреймов, являющихся его потомками. Чтобы загрузить новый документ, мы должны внести в *location.href* новый адрес URL. Поскольку нужно избавиться от фреймов, следует использовать объект location из родительского окна. (Напомним, что в каждый фрейм можно загрузить собственную страницу, то мы имеем для каждого фрейма собственный объект location). Итак, можно загрузить новую страницу в родительское окно с помощью команды:

parent.location.href= "http://...";



И наконец, очень часто Вам придется решать задачу обеспечения доступа с одного фрейма-потомка к другому такому же фрейму-потомку. Итак, как можно, находясь в первом фрейме, записать что-либо во второй - то есть, которой командой следует воспользоваться на HTML-странице *page1.htm*? Как можно увидеть на рисунке, между двумя этими фреймами нет никакой прямой связи. И потому мы не можем просто так вызвать *frame2*, находясь в фрейме *frame1*. С точки же зрения родительского окна второй фрейм действительно существует и называется *frame2*, а к самому родительскому окну можно обратиться из первого фрейма по имени *parent*. Таким образом, чтобы получить доступ к объекту document, разместившемуся во втором фрейме, мы должны написать следующее,:

parent.frame2.document.write("Привет, это вызов из первого фрейма.");



Предположим, мы имеем три фрейма с именами *frame1*, *frame2* и *frame3*.Допустим посетитель активирует ссылку в *frame1*. Нужно, чтобы при этом в два других фрейма загружались две различные web-страницы. В качестве решения этой задачи Вы можете, например, воспользоваться функцией:

function loadtwo() {

parent.frame1.location.href= "first.htm";

parent.frame2.location.href= "second.htm";}

Если же Вы хотите сделать функцию более гибкой, то можете воспользоваться возможностью передачи переменной в качестве аргумента. Результат будет выглядеть как:

function loadtwo(url1, url2) {

parent.frame1.location.href= url1;

parent.frame2.location.href= url2;}

Послеэтогоможноорганизоватьвызовфункции: loadtwo("first.htm", "second.htm") или loadtwo("third.htm", "forth.htm"). Очевидно, передача аргументов делает Вашу функцию более гибкой. В результате Вы можете использовать ее многократно и в различных контекстах.

**Методические рекомендации к выполнению задания 2**

1. Создайте документ html.
2. Создайте произвольную html-страничку и сохраните ее в отдельной папке на рабочем столе под названием cell.htm.
3. Создайте второй html-документ. Введите:

<frameset cols="50%,50%">

<frameset rows="50%,50%">

<frame src="cell.html">

<frame src="cell.html">

</frameset>

<frameset rows="33%,33%,33%">

<frame src="cell.html">

<frame src="cell.html">

<frame src="cell.html">

</frameset>

</frameset>

Можно задать толщину границы между фреймами, воспользовавшись в тэге <frameset> параметром *border*. Запись border=0 означает, что между тэгами нет какой-либо границы.

**Методические рекомендации к выполнению задания 3**

1 В одном фрейме создайте несколько ссылок. Однако, если посетитель активирует какую-либо из них, соответствующая страница будет помещена не в тот же самый фрейм, а в соседний. Сперва необходимо написать скрипт, создающий указанные фреймы:

<html>

<frameset rows="80%,20%">

<frame src="start.htm" name="main">

<frame src="menu.htm" name="menu">

</frameset>

</html>

Здесь *start.htm* - это та страница, которая первоначально будет показана в главном фрейме (*main*). У нас нет никаких специальных требований к содержимому этой страницы.

2 Сохраните документ под названием Frames.htm.

3 Создайте три документа с произвольным содержанием: first.htm, second.htm, third.htm.

4 Создайте документ ***menu.htm***:

<html>

<head>

<script language="JavaScript">

<!-- hide

function load(url) {

parent.location.href= url;

}

// -->

</script>

</head>

<body>

<a href="javascript:load('first.htm')">first</a>

<a href="second.htm" target="main">second</a>

<a href="third.htm" target="\_top">third</a>

</body>

</html>

В данном примере использованы несколько способов загрузки новой страницы во фрейм *main*. В первой ссылке для этой цели используется функция *load()*. Давайте посмотрим, как это делается:

<a href="javascript:load('first.htm')">first</a>

Вместо явной загрузки новой страницы сдесь предлагается браузеру выполнить некую команду на языке JavaScript - для этого используется параметр *javascript:* вместо обычного *href*. Далее, внутри скобок можно увидеть *'first.htm'*. Эта строка передается в качестве аргумента функции *load()*. Сама же функция *load()* определяется следующим образом:

function load(url) {

parent.main.location.href= url;}

Здесь внутри скобок написано *url*. Это означает, что строка 'first1.htm' при вызове функции заносится в переменную *url*. И эту новую переменную теперь можно использовать при работе внутри функций *load()*.

Во второй ссылке присутствует параметр *target*. Это одна из конструкций языка HTML. Как видно, здесь всего лишь указывается имя необходимого фрейма. Заметим, что в этом случае не обязательно ставить перед именем указанного фрейма слово *parent*. Причина такого отступления от правил в том, что параметр *target* - это функция языка HTML, а не JavaScript.

На примере третьей ссылки можно увидеть, как с помощью *target* можно избавиться от фреймов. Если нужно избавиться от фреймов с помощью функции *load()*, то необходимо написать в ней лишь parent.location.href= url.

**Вопросы для самоконтроля:**

1. Что называют фреймом?
2. Какую роль в веб-документе играют фреймы?
3. Какие теги необходимы для создания фреймов?
4. Какие атрибуты и параметры можно задать относительно фреймов?
5. Можно ли присвоить имя фреймам?
6. Каким образом организуется обеспечение доступа с одного фрейма-потомка к другому такому же фрейму-потомку?
7. Для чего предназначена функция loadtwo?