**Частное учреждение образования**

**«Гродненский колледж бизнеса и права»**

**Лабораторная работа № 16**

**по дисциплине**

**«Программные средства создания Internet - приложений»**

**Тема:** «Открытие, закрытие окон. Определение их параметров. Динамическое формирование HTML-документов»

для учащихся 4 курса специальности

2-40 01 01 «Программное обеспечение информационных технологий»

**Тема:** Открытие, закрытие окон. Определение их параметров. Динамическое формирование HTML-документов.

**Цель:** обучить приемам использования основных функций для работы с окнами. Сформировать умение производить динамическое формирование HTML-документа.

**Вид занятия:** Практическая работа.

**Тип занятия:** Формирование навыков.

**Время выполнения:** 2 часа

**Оборудование и методическое обеспечение:** IBM PC, приложение Блокнот, MS Internet Explorer карточки с заданием.

**Литература:**

1. Java Script.. Г. В. Гудман СПб.: Питер, 2004.

**План работы:**

1. Организационный этап занятия
2. Проверка домашнего задания
3. Информационно-подготовительный этап занятий
4. Исполнительский этап занятий
5. Постановка домашнего задания
6. Оценочно-рефлексивный этап занятия

**Ход занятия**

1. **Организационный этап занятия**

Проверка готовности учащихся к занятию, отметка отсутствующих, объявление темы и цели урока.

1. **Проверка домашнего задания**

Устный опрос учащихся по следующим вопросам:

1. Дайте характеристику методу open объекта window.
2. Опишите основные свойства окна, которыми можно управлять на JavaScript.
3. Дайте характеристику методу close объекта window.
4. В чем состоит преимущество управления фреймами на JavaScript?
5. **Информационно-подготовительный** **этап занятия**

***Создание Окна***

Начнем с того, что, когда Вы запускаете браузер, Вы, тем самым, уже создаете окно браузера Иногда же возникает необходимость создать еще одно окно *(или несколько)*, чтобы загрузить туда другой документ или даже поместить туда документ, сгенерированный Вашей **JavaSript** программой "на лету".

Для этого используется метод **open** объекта **window**.

open(*url*, *имя окна*, *параметры*);

первый аргумент - документ, который следует загрузить во вновь создаваемое окно. Второй - имя окна (значение свойства **name** нового объекта **window**). Третий - параметры создания окна.

Параметры создания окна:

* **height**=*высота в пикселах;*
* **width**=*ширина в пикселах*
* **location**=*yes|no (1|0);* - указывает, следует ли создаваемому окну иметь поле для ввода алреса;
* **menubar**=*yes|no (1|0)* - будет ли у создаваемого окна меню;
* **resizable**=*yes|no (1|0)* - можно ли будет менять размер окна после его создания;
* **scrollbars**=*yes|no (1|0)* - будут ли у создаваемого окна линейки прокрутки;
* **status**=*yes|no (1|0)* - будет ли у создаваемого окна поле статуса;
* **toolbar**=*yes|no (1|0)* - будет ли у создаваемого окна панель инструментов.

Например:

open("http://joker.botik.ru/ip/images/window.gif","ps",

"width=276 height=210 scrollbars=no resizable=no "+

"menubar=no location=no status=no toolbar=no");

Как всегда, мы опускаем указание на объект **window**, т.к. он используется по умолчанию.

***Доступ с созданному окну***

Что же мы может теперь сделать с этим окном? На самом деле, более важным является вопрос, а что мы хотим с ним сделать? Если мы хотели просто загрузить документ, что бы пользователь его посмотрел, то цель уже достигнута и больше нам ничего делать и не нужно.

Другое дело, если мы хотим, из своей программы, иметь доступ к объектам, расположенным в новом окне или, например, иметь возможность закрыть его, когда оно нам более не будет нужно.

В этом случае нам необходимо воспользоваться значением, которое возвращает метод **open**. С помощью этого значения, мы и будем впредь работать с вновь созданным окном.

Например:

**var** newWindow = open("","myWin","location=no");

newWindow.document.writeln("<H1>Hello world!</H1>Содержимое этого "+

"окна сгенерировано <B>JavaScript</B> программой.<P>");

Начало формы

, что из этого получится.

Конец формы

Обратите внимание, что на этот раз, мы явно указали к какому окну относится **document.writeln()**. Если раньше, мы всегда писали просто **document.writeln()** *(или, что тоже самое -* ***window.document.writeln()****)* и это относилось к ***нашему текущему окну***, то теперь мы явно указали, к документу какого именно окна мы обращаемся!

В этом и состоит вся *"хитрость"* работы с окнами. Окно - это объект. Значит, если мы укажем с каким объектом мы намерены иметь дело, мы можем использовать все свойства и методы этого объекта! Таким образом, ***все***, что мы усеем делать со своим родным окном, можно делать и посторонним, нужно только всегда указывать с каким именно окном мы работаем.

Давайте, например, закроем недавно созданное окно. Для этого запишем:

newWindow.close();

Начало формы

Конец формы

Чтобы эффективно работать с окнами, нужно знать еще несколько вещей:

1. у объекта **document**, тоже есть методы **open** и **close**. Первый из них следует вызывать перед тем, как начинать запись в документ методами **write** и **writeln**. Мы не делали этого раньше потому, что при создании окна, там уже есть открытый документ. Метод **close** следует вызывать после завершения записи. Комбинация **close** и, затем, **open** приведет к тому, что документ будет очищен и Вы сможете начинать писать в него заново;
2. вновь созданное окно автоматически получает фокус ввода. Можно переключить фокус на любое окно с помощью метода **focus()** объекта **window**;
3. для манипулирования окнами в объекте **window** имееется два полезных свойства: **self** - указывает на текущее окно и **opener** - указывает на то окно, из которого было открыто данное;
4. для того чтобы проверить, существует ли созданное ранее окно или пользователь уже закрыл его (нажатие на **X** в верхнем правом углу), можно воспользоваться свойство **closed** объекта **window**.

***Динамическое создание документа.***

<html><head>

<script language="JavaScript">

function openWin () {

myWin= open("", "displayWindow",

"width=500,height=400,status=yes,toolbar=yes,menubar=yes");

myWin.document.open();

myWin.document.write("<html><head><title>On-the-fly");

myWin.document.write("</title></head><body>");

myWin.document.write("<center><font size=+3>");

myWin.document.write("Данный документ HTML был создан ");

myWin.document.write("с помощью JavaScript!");

myWin.document.write("</font></center>");

myWin.document.write("</body></html>");

myWin.document.close();}

</script>

</head>

<body>

<form>

<input type=button value="On-the-fly" onClick="openWin()">

</form>

</body>

</html>

### *Создание фреймов*

Фреймы создаются в HTML с помощью тэга FRAME. В JavaScript доступ к фреймам осуществляется через массив frames - свойство объекта window.

Мы не будем рассматривать здесь способы описания фреймов, атрибут TARGET и т.п. Все это должно быть Вам известно из курса HTML, сосредоточимся на доступе к фреймам из программ на JavaScript.

### *Работа с фреймами*

Когда мы говорили об окнах, то мы отмечали, что они все равноправны. Елинственное, что связывало порожденное окно с породившим - свойство opener.

С фреймами дело обстоит иначе. Все фреймы созданные в окне, являются потомками этого окна. Если внутри фрейма загружен документ, который, в свою очередь, содержит FRAMESET то фреймы этого FRAMESET являются потомками родительского фрейма и т.д. В остальном же фреймы подобны окнам. Они обладают всеми свойствами объекта window.

Для организации связи родитель-потомок служат свойства объекта window: parent - окно-родитель *(фрейм-родитель)* и top - окно самого верхнего уровня.

Рассмотрим окно в котором определены следующие тэги.

<FRAMESET COLS="50%,\*">

<FRAMESET ROWS="30%,\*">

<FRAME NAME="fr1" SRC="javascript:void(0)"></FRAME>

<FRAME NAME="fr2" SRC="javascript:void(0)"></FRAME>

</FRAMESET>

<FRAME NAME="fr3" SRC="javascript:void(0)"></FRAME>

</FRAMESET>

В этом окне определены три фрейма. Все они являются потомками текущего окна. Значит, все они являются элементами массива frames текущего окна. Таким образом, мы можем обращаться к ним по их именам, через массив frames, а именно frames["fr1"], frames["fr2"] и frames["fr3"].

На самом деле, поскольку их имена уникальны, можно обращаться и просто по именам, минуя массив frames - fr1, fr2 и fr3.

Ну, а что делать, если программа определенная в документе, загруженном во фрейм fr1, захочет добраться до фрейма fr2? Да, именно так, как Вы подумали, она должна обратиться к fr2 через главное окно: top.fr2 или parent.fr2, поскольку в данном случае окно - родитель одновременно является и окном самого верхнего уровня.

Ниже приведен текст некой страницы. Посмотрите и попытайтесь понять что там к чему.

<HTML>

<HEAD>

<META HTTP-EQUIV="Content-Type"

CONTENT="text/html; charset=windows-1251">

<TITLE>Пример фреймов</TITLE>

<SCRIPT LANGUAGE="JavaScript">

function writeToFrames() {

fr1.document.write("<H1>Фрейм 1</H1>Попали?");

fr2.document.write("<H1>Фрейм 2</H1>Попали?");

fr3.document.write("<H1>Фрейм 3</H1>Попали?");

}

onload = writeToFrames;

</SCRIPT>

</HEAD>

<FRAMESET COLS="50%,\*">

<FRAMESET ROWS="30%,\*">

<FRAME NAME="fr1"></FRAME>

<FRAME NAME="fr2"></FRAME>

</FRAMESET>

<FRAME NAME="fr3"></FRAME>

</FRAMESET>

</HTML>

Теперь, когда мы знаем как именовать фреймы и как добираться из одного в другой, мы можем делать с фреймами абсолютно все, что мы делали с окнами - писать в них, загружать в них другие документы и т.д. и т.п.

Старайтесь размещать Ваши JavaScript функции в самом главном документе, там где у Вас находится FRAMESET. Это даст Вам уверенность, что когда бы некая функция ни понадобилась документу расположенному во фрейме, она уже будет доступна, ведь главный документ грузится из сети первым.

И последний пример. Допустим, Вы разработали страницу, состоящую из нескольких фреймов и Вам бы очень не хотелось, чтобы документы, предназначенные для показа во фреймах, кто-то загружал бы как главные - прямо в окно. Например, они при этом непонятны или еще что-нибудь.

Как с этим бороться? Очевидно, документ, предназначенный для показа во фрейме, должен проверять, а не загрузили ли его как самостоятельный? И, если так, то немедленно загружать, вместо себя, свою главную - фреймосодержащую страницу.

Для решения этой задачи, достаточно, в самом начале документов, предназначенных для показа во фреймах, расположить такой простой код:

if (self==parent) location.replace("mainpage.html");

Здесь предполагается, что главная страница называется mainpage.html.

1. **Исполнительский этап занятия**

***Вариант 1***

*Уровень 1.* Дайте характеристику объекта location.

Какую роль играют значения параметра name, используемого в тегах HTML, при создании скриптов?

*Уровень 2.* Создать web-документ (1.html). В данном документе создать форму, состоящую из строки ввода и кнопки.

При нажатии кнопки содержимое созданной строки должно появиться в строке состояния.

Создать фреймовую структуру:

|  |  |
| --- | --- |
| 30% | 70% |

*Уровень 3.* В правый фрейм загрузить пустую web-страницу(2.html), в левый фрейм загрузить web-страницу (3.html), содержащую ссылку на 1.html. При выборе ссылки, документ должен загрузиться в правый фрейм.

На странице 2.html установите кнопку. При нажатии кнопки должно появиться новое окно с документом 4.html, содержащим фамилию и номер варианта.

Для окна с документом 4.html установите: ширина – 300, высота – 200, отсутствие панели инструментов и полосы прокрутки, отступ сверху – 150.

Для документа 4.html установить ссылку, при нажатии которой, данное окно закроется.

*Уровень 4.* Для документа 1.html создать кнопку, при нажатии которой должен динамически сформироваться новый документ, содержащий ответы на вопросы первого уровня.

Создать фреймовую структуру:

|  |  |  |
| --- | --- | --- |
| 1 | | |
| 2 | 3 | 4 |

Во фреймы 2,3,4 загрузить пустую web-страницу, в первый фрейм загрузить страницу (5.html) содержащую ссылку. При выборе ссылки документ 1.html должен загрузиться во второй фрейм, документ 2.html– в третий фрейм.

В документе 5.html создать кнопку, при нажатии которой должно появиться окно сообщений, содержащее текст, набранный в строке ввода документа 1.html.

В документе 2.html необходимо создать ссылку. При наведении мыши на данную ссылку, в строке состояния должен появиться текст: «Это задание на 10 баллов».

***Вариант 2***

*Уровень 1.* Дайте характеристику объекта status.

Опишите особенности управления фреймами на JavaScript.

*Уровень 2.* Создать web-документ. В данном документе создать форму, состоящую из кнопки.

При нажатии кнопки содержимое контейнера <title> должно появиться в строке состояния.

Создать фреймовую структуру:

|  |
| --- |
| 20% |
| 80% |

*Уровень 3.* В верхний фрейм загрузить пустую web-страницу(2.html), в нижний фрейм загрузить web-страницу (3.html), содержащую ссылку на 1.html. При выборе ссылки, документ должен загрузиться в текущий фрейм.

На странице 2.html установите ссылку, при нажатии которой должно появиться новое окно с документом 4.html, содержащим фамилию и номер варианта.

Для окна с документом 4.html установите: ширина – 500, высота – 400, отсутствие строки меню и строки состояния, отступ слева – 100.

Для документа 4.html установить кнопку, при нажатии которой, данное окно закроется.

*Уровень 4.* Для документа 1.html создать кнопку, при нажатии которой должен динамически сформироваться новый документ, содержащий ответы на вопросы первого уровня.

Создать фреймовую структуру:

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |

Во фреймы 2,3,4 загрузить пустую web-страницу, в первый фрейм загрузить страницу (5.html) содержащую кнопку. При нажатии кнопки документ 1.html должен загрузиться во второй фрейм, документ 2.html– в третий фрейм.

В документе 5.html создать кнопку, при нажатии которой текст в документе 2.html должен стать зелёного цвета.

В документе 2.html необходимо создать ссылку. При наведении мыши на данную ссылку, в строке состояния должен появиться текст: «Это задание на десятку».

1. **Постановка домашнего задания**

Из литературы [2], с.80-83

1. **Оценочно-рефлексивный этап занятия**

Учащимся необходимо ответить на поставленные вопросы:

1. Что нового Вы узнали на занятии?
2. Что вызвало затруднение при выполнении задания?
3. Было ли Вам интересно на занятии?