

МИНОБРНАУКИ РФ  
ФГБОУ ВПО «УДМУРТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
Филиал в г. Нижняя Тура

**«Информационные технологии создания сетевых  
ресурсов»**

Нижняя Тура  
2012

УДК 681.518  
ББК 32.81  
И-74

«УТВЕРЖДАЮ»  
Зам. директора ИЭиУ  
по учебно-методической работе

\_\_\_\_\_ О. А. Воробьева  
«\_\_» \_\_\_\_\_ 20 г.

В учебном пособии «Информационные технологии создания сетевых ресурсов» изложен материал для проведения лабораторных занятий.

Дисциплина «Информационные технологии создания сетевых ресурсов» входит в образовательный модуль раздела «Б.2. Математический и естественнонаучный цикл. Вариативная часть» ФГОС ООП бакалавриата по направлению подготовки «Экономика» профиля подготовки бакалавров.

Цикл лабораторных работ, представленных в данном пособии, помогает постепенно освоить основные технологии создания сетевых ресурсов, материал структурирован по темам курса в соответствии с программой. Лабораторные работы содержат необходимый минимум теоретического материала и задания с подробным алгоритмом решения, что обеспечивает помощь студентам при выполнении заданий для самостоятельной работы.

Для студентов, обучающихся по направлению подготовки «Экономика», профиль подготовки «080112 Экономика предприятий и организаций».

Автор учебного пособия: Лапшина Н.А, старший преподаватель ФГБОУ ВПО «УдГУ», филиал г. Нижняя Тура

Учебно–методические материалы обсуждены и одобрены на заседании кафедры экономики и правовых основ управления «\_\_» \_\_\_\_\_ 20 г., протокол №

Зав. кафедрой экономики и правовых основ управления

\_\_\_\_\_ к. э. н. О.А. Скобелева.

## Оглавление

Введение .....	4
Лабораторная работа №1 Разработка Web-страниц при помощи языка разметки гипертекста HTML.....	5
Лабораторная работа №2: Создание сайтов в программе Microsoft Office SharePoint Designer.....	9
Лабораторная работа №3 «Создание web-сайтов с использованием MS SHAREPOINT DESIGNER (ПЕРЕХОДЫ)» .....	14
Лабораторная работа №4: Графика в web . Создание анимационного баннера в программе Adobe Photoshop. ....	19
Лабораторная работа №5: Работа с формами. ....	23
Лабораторная работа №6: Использование для обработки форм скриптового языка JavaScript.....	31
Лабораторная работа №7: Создание и заполнение блога. ....	38
Лабораторная работа №8: Создание сайта на бесплатном сервисе Yandex - Народ. ....	41
Список литературы:.....	53

## Введение

В настоящее время в связи с развитием глобальной компьютерной сети Интернет, любая компания, предприятие или учреждение обязательно имеют Интернет-портал в сети, либо собираются (а скорее всего обязаны) его создать. В связи умение создавать сетевые ресурсы является достаточно актуальным в наше время.

Существует большое количество технологий, позволяющих создавать сайты. К основным технологиям в наше время относятся:

1. Использование языка гипертекстовой разметки [HTML](#) и каскадных таблиц стилей CSS.
2. Использование программ-конструкторов (например Microsoft Office SharePoint Designer 2007 )
3. Использование скриптовых языков (PHP, Java и др.)
4. Использование флеш-технологии.

Каждая из перечисленных технологий имеет свои достоинства и недостатки и выбор технологии личное дело автора сайта, но как показывает практика оптимальным решением является сочетание всех перечисленных технологий. Например дизайн (шаблон) сайта можно создать в программе – конструкторе. Лишний код, который программа генерирует сама удалить (это сделать не сложно, если Вы знаете язык HTML). Для того чтобы сайт был интерактивным добавить в него элементы созданные при помощи скриптовых языков. Технологию Flash можно применить в качестве элементов дизайна (например, для создания логотипов, флэш-меню, и других анимированных графических элементов) в качестве анимированных flash-баннеров и входных flash-заставок. В результате будет создан современный интерактивный красивый web-портал. Но для того чтобы это реализовать необходимо знать как применяются перечисленные технологии.

Цель данного пособия познакомить студентов с основными технологиями создания сайтов, научить применять их на практике.

Разновидность технологий создания сетевых ресурсов не ограничивается перечисленными, их намного больше, но большинство сайтов работают именно на вышеописанных языках. Время идёт – всё меняется. Интернет не является исключением, скорее наоборот. Технологии создания сайта изменяются ещё быстрее.

## Лабораторная работа №1 Разработка Web-страниц при помощи языка разметки гипертекста HTML.

### Теоретический материал:

Для создания статических Web-страниц и Web-сайтов существует два стандартных способа:

1. Использование языка разметки гипертекста;
2. Использование специальных программ (конструкторов сайтов).

У каждого из этих способов есть свои достоинства и недостатки. Например, в первом способе необходимо полностью прописывать весь код страницы самостоятельно, что является довольно трудоемким процессом. Используя второй способ, достаточно просто используя инструменты конструктора визуально создать страничку, код программа генерирует сама. Недостаток этого способа состоит в том, что программа генерирует много лишнего кода и это приводит к увеличению размера страницы.

Оптимальным решением на данном уровне является использование обоих способов в комплексе. Необходимо визуально создать страничку в любом, на ваш выбор конструкторе. Просмотреть код и самостоятельно его скорректировать. Однако для того чтобы вы могли скорректировать код необходимо иметь представление о самом языке разметки гипертекста на котором этот код записан, познакомиться с основными тегами и атрибутами этого языка, создать простейшую web- страничку на этом языке и только потом переходить к изучению второго способа – работе с конструктором сайтов. В данной работе мы рассмотрим основные структурные элементы этого языка и создадим небольшую личную Web-страничку.

**Язык разметки гипертекста HTML** (Hyper Text Markup Language) был создан Тимом Бернерсом-Ли в 1989 г.

Код страницы набирается в любом текстовом редакторе (например, *Блокнот*) или специальной программе (например, *Homepage Builder*) и оформляется с помощью команд языка HTML. Тестируется html-файл с помощью любого браузера. *Web-страницы (html-файлы) имеют формат \*.htm или \*.html.*

Команды HTML задаются с помощью специальных элементов — *тэгов*, которые заключаются в угловые скобки. Прописные и строчные буквы при написании тэгов не различаются!

Существует два типа тэгов:

- ✓ непарные, например, <BR> — тэг перевода строки;
- ✓ парные, например, <B>мама</B> — выделение слова «мама» полужирным шрифтом.

Признаком окончания действия парного тэга служит слэш / .

Программы, написанные на языке HTML, имеют определенную стандартизированную структуру (листинг1):

```
<html>
  <head>
    <title>Моя первая страничка</title>
  </head>
  <body>
    ...
  </body>
</html>
```

листинг 1

#### **Описание структуры:**

`<html>...</html>` — это первый тэг, который должен находиться в любом html-документе (указывает на то, что данный документ действительно содержит в себе html-текст).

`<head>...</head>` — определяет заголовок страницы, содержит справочную информацию о странице (название, используемая кодировка, ключевые слова и т.д.).

`<title>...</title>` — определяет название страницы, которое будет выведено в строке заголовка браузера.

`<body>...</body>` — определяет содержимое страницы, выводимое на экран монитора.

#### **Практикум:**

##### **Задание №1**

Набрать структуру программы в простейшем текстовом редакторе и сохранить в формате html в своей папке.

#### **Технология работы:**

1. Запустите текстовый редактор Блокнот (либо другой простой текстовой редактор).
2. Наберите в окне редактора код структуры (листинг1).
3. Сохраните этот файл в своей папке, используя команду **Сохранить как**, под именем **index.htm** (имя файла может быть любым на ваше усмотрение, расширения файла обязательно htm или html)

**В текстовом редакторе можно изменять код страницы, а просматривать саму страничку необходимо в программе для просмотра Web-страниц – Web-браузере!**

## Задание №2

Просмотрите созданную вами чистую страничку.

### Технология работы:

1. Сверните окно текстового редактора на панель задач или закройте его.
2. Откройте вашу папку и запустите файл **index.htm**, двойным нажатием левой кнопки мыши.

Если вы все сделали правильно, запустится браузер (например MozillaFirefox), в котором откроется ваша чистая страничка, в строке заголовка браузера будет написано **Моя первая страничка**.

Если страничка отображается неправильно необходимо открыть файл **index.htm** в текстовом редакторе (Блокнот) и проверить правильность написания кода, затем снова просмотреть страницу в браузере.

В данном задании рассмотрено создание чистой странички, для ее оформления существует достаточно большое количество тегов, описание основных тегов приведено в приложении №1 данного практикума.

## Задание №3

Наберите на вашей страничке текстовую информацию для приветствия посетителей.

### Технология работы:

1. Запустите текстовый редактор (Блокнот).
2. Откройте в редакторе файл **index.htm**.
3. Между открывающимся и закрывающимся тегом **body** наберите текст по образцу (Листинг2).

```
<html>
  <head>
    <title>Моя первая страничка</title>
  </head>
  <body>
```

Спасибо Вам, уважаемый посетитель, за то, что Вы нашли время и возможность заглянуть на мою страницу во Всемирной сети. Здесь Вашему вниманию предложена информация о моих увлечениях, моей биографии, моих друзьях, работе и семье. В отдельной рубрике я предлагаю Вам несколько фотографий. Надеюсь, Вам они понравятся. К сожалению, в настоящий момент моя домашняя страничка находится в стадии разработки, поэтому еще далеко не все ее разделы доступны для просмотра. Я искренне надеюсь, что этот ресурс будет обновляться и в ближайшем будущем многие из перечисленных рубрик займут свое место. Заходите на мою страницу почаще! С уважением, Иван Петров.

```
</body>
</html>
```

листинг 2

4. Сохраните изменения внесенные вами в страничку (Файл/Сохранить). Закройте текстовый редактор и просмотрите страничку через браузер. Атрибуты тегов.

Практически каждый тег языка html имеет ряд атрибутов (свойств тегов) которым можно присваивать необходимые значения.

Например, если необходимо изменить цвет всей странички необходимо использовать один из атрибутов тега **body**. *Атрибут изменения цвета - bgcolor* , причем ему необходимо присвоить код цвета (из соответствующей таблицы цветов).

Например:

`<body bgcolor= "#008000">` - это означает, что фон вашей странички будет зеленый.

Если необходимо изменить цвет шрифта текста странички, то используется атрибут - **text**.

Например:

`<body bgcolor= "#008000" text="#0000ff">` - это означает, что фон вашей странички будет зеленый, а текст синий. И так прописываются все элементы странички при помощи различных тегов и их атрибутов.

Для того чтобы вы могли применить другие теги и их атрибуты используйте в работе таблицу тегов (приложение №1) и таблицу цветов (приложение №2)

**Задание №4** Оформите вашу страничку используя таблицы тегов и таблицу цветов по вашему усмотрению. Примерный вид программы приведен в листинге 3.

```
<html>
  <head>
    <title>Моя первая страничка</title>
  </head>
  <body bgcolor= #008000 text=#0000ff >
    <h1 align ="center" > Добро пожаловать на мою домашнюю
    страницу </h1>
    &nbsp; &nbsp; &nbsp;
```

Спасибо Вам, уважаемый посетитель, за то, что Вы нашли время и возможность заглянуть на мою страницу во Всемирной сети. Здесь Вашему вниманию предложена информация о моих увлечениях, моей биографии, моих друзьях, работе и семье. В отдельной рубрике я предлагаю Вам несколько фотографий. Надеюсь, Вам они понравятся.

```
<br>
```

```
&nbsp; &nbsp; &nbsp;
```

*<i>К сожалению, в настоящий момент моя домашняя страничка находится в стадии разработки, поэтому еще далеко не все ее разделы доступны для просмотра. Я искренне надеюсь, что этот ресурс будет обновляться и в ближайшем будущем многие из перечисленных рубрик займут свое место. </i>*

**<br>**

*<i> <u>Заходите на мою страницу почаще! С уважением, Иван Петров. </u></i>*

**</body >**

**</html>**

### листинг 3

**Задание №5** Сохраните изменения в домашней страничке, откройте страничку в браузере и покажите задание преподавателю.

**Задание №6** Используя таблицу тегов и атрибутов (Приложение 1) отформатируйте созданную Вами страничку по Вашему усмотрению разместите фоновый рисунок, картинку, бегущую строку и т.д. Сохраните результаты работы.

**Задание № 7** Используя таблицу тегов и атрибутов (Приложение 1) - раздел **таблицы**, создайте чистую Web-страничку и разместите на ней таблицу состоящую из 3 строк в каждой из которых по две ячейки ( рис.1).

**Ваш возраст :**

до 18	да
18-50	нет
старше 50	нет

Рисунок 1

Заполняя текст ячеек таблицы поставьте **Да** в ячейке, которая находится напротив вашего возраста. Сохраните страничку в своей папке, задайте имя страничке –**таблица.htm** .

## Лабораторная работа №2: Создание сайтов в программе Microsoft Office SharePoint Designer.

### Теоретический материал:

Вторым способом создания сайтов является использование специальных программ-конструкторов. Специальные редакторы, ориентированные на визуальную разработку сайтов, позволяют проектировать web-страницы так же легко, как и создавать документ в текстовом редакторе. Вы можете размещать на странице текст, таблицы, рисунки, видео, создавать ссылки,

использовать различные интерактивные компоненты и при этом работать со страницей в режиме WYSIWYG (What You See Is What You Get - Что видишь, то и получаешь), то есть практически в том виде, в каком она будет выглядеть в браузере. Визуальный редактор позволяет скрыть внутреннее представление страницы в формате HTML, что существенно облегчает разработку и избавляет от поиска синтаксических ошибок. Существует большое количество таких программ. Например, достаточно простой конструктор - Web Page Maker - это русифицированный, условно-бесплатный редактор для создания web-страниц. Так же достаточно популярной программой является Micromedia Dreamweaver – очень мощный редактор, обладающий полным набором функций для создания полноценного сайта.

Одним из редакторов для визуального проектирования web-сайтов, обладающим на сегодняшний день, наверное, самым широким спектром возможностей, является программа SharePoint Designer 2007 компании Microsoft, входящая в состав пакета MS Office 2007. Редактор SharePoint Designer содержит большой набор шаблонов и мастеров для создания сайтов по различной тематике. Для форматирования текста можно использовать все возможности, предусмотренные в основном стандарте HTML, а также применять специальные динамические эффекты и анимацию. Редактор имеет удобные инструменты для работы с таблицами, встроенные средства для обработки изображений, позволяет легко размещать на страницах различные мультимедиа-объекты: рисунки, видеофильмы, анимацию, звуковые фрагменты. Тесная интеграция с пакетом MS Office позволяет отображать на страницах документы MS Word, таблицы и графики MS Excel, динамически получать данные из MS Access, использовать язык VBA, средства проверки орфографии и десятки готовых тем для оформления страниц сайта. В программе SharePoint Designer реализована поддержка современных web-технологий, таких, как каскадные таблицы стилей (CSS), динамические эффекты (DHTML), фреймы, активные страницы (ASP), элементы ActiveX и Java-апплеты. Если web-сервер поддерживает технологии SharePoint Designer, то появляются дополнительные возможности при создании страниц - формы поиска, счетчики посещений, оглавление сайта и другие. Программа SharePoint Designer является не только редактором web-страниц, но и содержит средства управления сайтом, такие как схема навигации по страницам, анализ сайта с помощью различных отчетов, коллективная разработка, настройка на определенные браузеры, загрузка сайта на web-сервер по протоколам HTTP и FTP. Все эти возможности позволяют создавать с помощью SharePoint Designer полноценные web-сайты, такие, например, как электронные магазины или игровые сайты.

К недостаткам редактора SharePoint Designer можно отнести некоторую избыточность готового кода HTML, а также его преимущественную

ориентацию на браузер Internet Explorer, поэтому следует проверять работу готового сайта в других браузерах.

Программа SharePoint Designer входит в состав пакета MS Office 2007, поэтому ее можно установить вместе с другими приложениями MS Office или отдельно.

В этой лабораторной работе на примере разработки Вашего личного сайта, состоящего как минимум из 5 страниц мы познакомимся с основными возможностями этой программы.

### Практикум:

**Задание №1** Создать стартовую Web-страничку при помощи программы Microsoft Office SharePoint Designer.

#### Технология работы:

1. Создайте в своей папке папку **Site** для хранения файлов сайта.
2. В папке **Site** создайте две папки:
  - **Images** (для хранения графических файлов).
  - **files** (для хранения файлов других типов).
3. Подберите, используя возможности Интернета, либо другим способом картинки для размещения на сайте (фоновые рисунки, фото, анимационные картинки) и скопируйте их в папку **images** (имена файлов запишите английскими буквами).
4. Запустите программу **MS SharePoint Designer** . (Пуск/Программы/MS office / **MS SharePoint Designer**).
5. Создайте новый Web-узел (Файл/Создать/страница) рис.2.

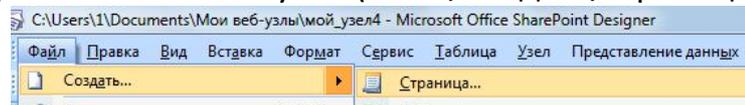


Рисунок 2

6. Используя команду **Файл/Сохранить** сохраните вашу страницу в вашей папке в папке **Site** под именем **index.htm**
7. Убедитесь, что Вы находитесь в режиме конструктора (в левом нижнем углу находятся кнопки переключения режима)рис.3.

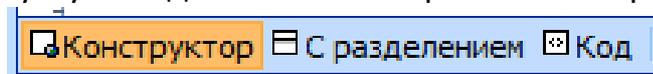


Рисунок 3

8. Создайте дизайн для Вашей странички, для этого:
  - На вкладке **Таблица**, выберите **Макетные таблицы**.
  - На панели **Макетные таблицы** выберите макет (угол, верхний колонтитул, столбец слева и осн. область). Название макета появляется при наведении на него указателя мыши рис.4.

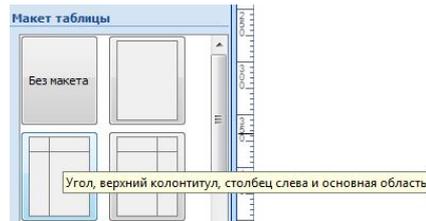


Рисунок 4

При желании можно выбрать другой макет из предложенных. Закройте панель **Макетные таблицы**.

9. Установите фоновый рисунок. Для этого:

- На вкладке **Формат** выберите **Фон**.
- В появившемся окне, установите флажок для **Фоновый рисунок**, нажмите **обзор** и выберите любой понравившийся вам фон из Вашей папки **images**.
- Нажмите **ОК**.
- Либо аналогичным образом можно установить просто фоновую заливку (**Формат/Фон/Цвет фона**).

10. Установите эмблему для Вашей странички. Для этого:

- Установите курсор в верхнюю левую ячейку таблицы.
- На вкладке **Вставка** выберите **рисунок/Из файла**.
- Выберите понравившуюся Вам картинку из папки **images** (можно анимационную) Можно ввести замещающий текст, например **Эмблема**.
- Просмотрите результат в браузере (**Файл/Просмотреть в обозревателе/выбрать обозреватель из списка**).
- Если Вас не устраивает размер рисунка его можно изменить, установить рамку и др. настройки (Для этого необходимо вызвать контекстное меню, щелчком правой кнопки мыши по рисунку и открыть в появившемся окне вкладку **вид**).
- Вернитесь в конструктор.
- Нажмите **ОК**.

11. Рядом с ячейкой с эмблемой разместите текст с названием Вашего сайта (можно вставить картинку). Для этого:

- Установите курсор в ячейку рядом с эмблемой.
- Наберите текст заголовка. Например, **Мой сайт** или **Личный сайт механика Гаврилова** и др.
- Отформатируйте заголовок по вашему желанию.
- Просмотрите результат в браузере.

12. Заполните центральную часть<sup>1</sup> сайта, текст можно взять из предыдущей работы: **Добро пожаловать.....** и т.д.

<sup>1</sup> Обратите внимание, область слева под эмблемой оставлена чистой для навигационной панели.

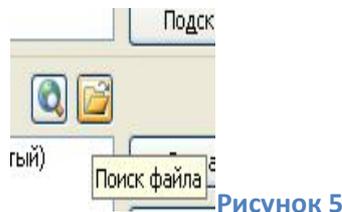
13. Можно добавить анимационные элементы (например бегущую строку). Для этого:

- Установите курсор в необходимое место на странице.
- Зайдите в пункт основного меню Вставка/Веб-компонент.
- Выберите эффект Бегущая строка.
- В появившемся окне введите текст и установите понравившиеся Вам настройки. Нажмите ОК. Просмотрите результат в браузере.

14. Можно добавить и заполнить таблицу и т.д.

15. Так же можете прикрепить любой файл. Для этого:

- Найдите в сети любой файл или просто создайте документ в текстовом редакторе.
- Сохраните этот файл, в папку **files** в вашей папке<sup>2</sup>.
- Напишите, в любом понравившемся Вам месте текст с названием файла. Например: **Просмотреть** или **Скачать** и т.д.
- Для прикрепления файла установим гиперссылку, для этого выделите набранный текст.
- На вкладке **Вставка** выберите пункт **Гиперссылка**.
- На панели **Связать с** выбрать с файлом...
- Нажать кнопку Поиск Файла Рис.5.



- В появившемся окне выберите файл. Нажмите ОК.
- Просмотрите файл в обозревателе.

16. Продолжите оформление странички по Вашему желанию.

17. Сохраните изменения.

**Задание №2** Создайте, оформите в соответствии с темой и сохраните аналогичным образом, еще 4 страницы например: Биография, Фотогалерея, Увлечения, Регистрация и т.п.

Нет необходимости создавать каждую страницу в отдельности, можно изменять эту страницу, удалить центральную часть, заполнить по другому в соответствии с тематикой и сохранить под другим именем, обязательно используя команду **Сохранить как**. В этом случае останется и первоначально созданная страница и новая как 2 разных

<sup>2</sup> Напоминаю о необходимости вводить имена английскими буквами. Это необходимо для того, чтобы избежать проблем при размещении сайта в сети.

файла. Таким образом в Вашей папке должны находиться 2 папки и 5 веб- страниц. Например (index.htm, yvlecheni.htm, biograf.htm, fotogal.htm, registr.htm).

### **Задание №3** Создание навигационной панели.

#### **Технология работы:**

1. Откройте файл index.htm в программе **Microsoft Office SharePoint Designer**.
2. Установите курсор в ячейку таблицы слева, для создания панели.
3. Наберите названия страниц. Например:

Главная

Биография

Увлечения

Фотогалерея

Регистрация

4. Установите понравившийся Вам шрифт.
5. Скопируйте названия страниц на остальные странички сайта.
6. Установите связь между страницами используя гиперссылки, для этого:
  - Перейдите на стартовую страницу (index.htm).
  - Выделите **Главная**
  - На вкладке **Вставка** выберите пункт **Гиперссылка**.
  - На панели **Связать с** выбрать с файлом, веб-страницей.
  - Используя кнопку Поиск файла, откройте Вашу папку site и выделите страницу index.htm.
  - Аналогично установите гиперссылки для других страниц (биография, увлечения, фотогалерея, регистрация)
7. Откройте одновременно с этой все остальные страницы. И установите гиперссылки для навигационной панели аналогичным образом (можно скопировать панель с установленными гиперссылками на остальные страницы).
8. Просмотрите стартовую страницу через обозреватель.
9. Сайт готов.

### **Лабораторная работа №3 «Создание web-сайтов с использованием MS SHAREPOINT DESIGNER (ПЕРЕХОДЫ)»**

Одним из редакторов для визуального проектирования web-сайтов, обладающим на сегодняшний день, наверное, самым широким спектром возможностей, является программа SharePoint Designer 2007

компании Microsoft, входящая в состав пакета MS Office 2007. Редактор SharePoint Designer содержит большой набор шаблонов и мастеров для создания сайтов по различной тематике.

**Практикум:**

**Задание№1** При помощи переходов создать структуру сайта туристической фирмы имеющий следующую структуру (рис.6).

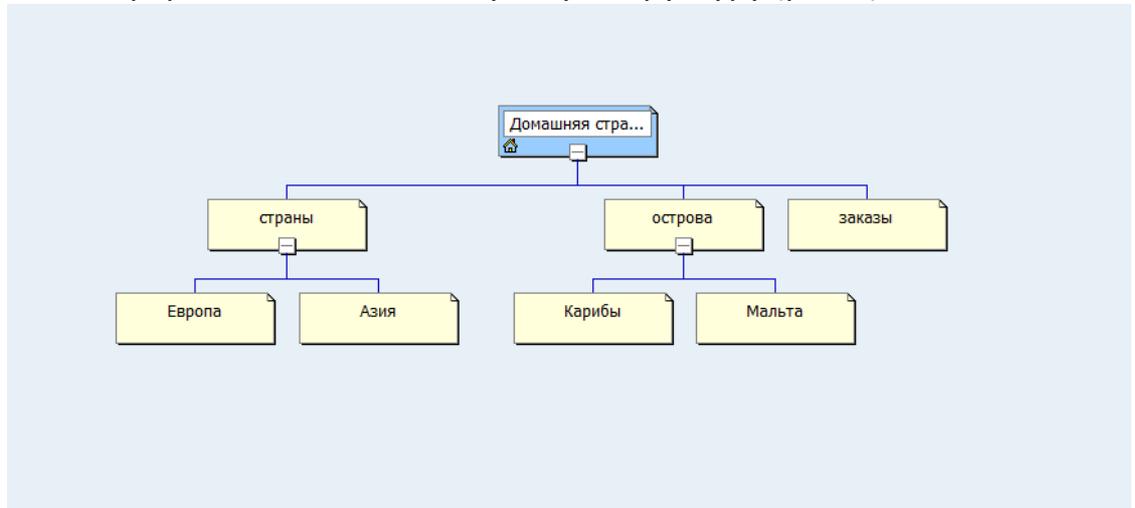


Рисунок 6

**Технология работы:**

1. Запустите программу **MS SharePoint Designer 2007** .  
(Пуск/Программы/MS office / **MS SharePoint Designer 2007**).
2. Создайте пустой web-узел(Файл/Создать/web-узел).
3. Перейдите в режим **Переходы** (рис.7).

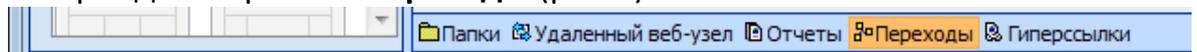


Рисунок 7

4. Щелкните правой кнопкой мыши в рабочей области, выберите Создать/Верхняя страница.
5. Выделите Домашнюю страницу, щелкните по ней правой кнопкой мыши, выберите Создать /страница (рис.8).

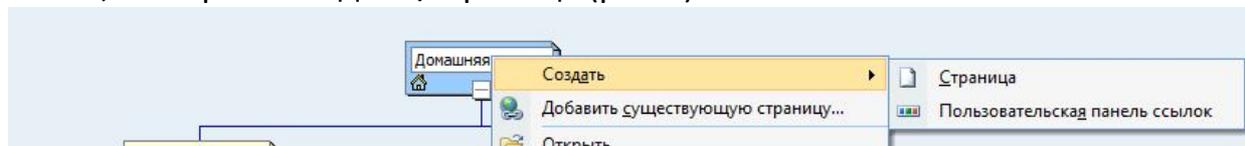


Рисунок 8

6. Переименуйте ее в соответствии с схемой.
  7. Аналогично добавьте другие страницы структуры (рис.6).
- На данном этапе вы автоматически создали 8 чистых web-страниц (Домашняя, страны, острова, заказы, Европа, Азия, Карибы, Мальта) и определили связи между ними.  
Просмотрите их в папке Web- узла.

**Задание№2** Оформите созданные страницы.

### Технология работы:

1. Откройте домашнюю страницу Вашего сайта в **MS SharePoint Designer 2007**.
2. Убедитесь, что Вы находитесь в режиме конструктора (в левом нижнем углу находятся кнопки переключения режима) рис.9.

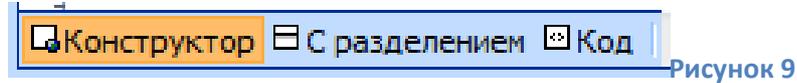


Рисунок 9

3. Создайте дизайн для Вашей странички, для этого:

- На вкладке **Таблица**, выберите **Макетные таблицы**.
- На панели **Макетные таблицы** выберите **макет** (угол, верхний колонтитул, столбец слева и осн. область). Название макета появляется при наведении на него указателя мыши рис.10.

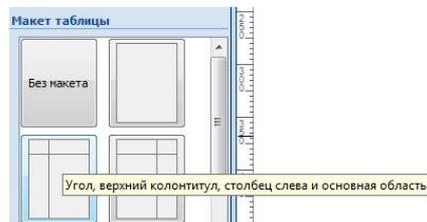


Рисунок 10

- Выровняйте таблицу по центру страницы.

При желании можно выбрать другой макет из предложенных. Закройте панель **Макетные таблицы**.

4. Установите фоновый рисунок. Для этого:

- На вкладке **Формат** выберите **Фон**.
- В появившемся окне, установите флажок для **Фоновый рисунок**, нажмите **обзор** и выберите любой понравившийся вам фон.
- Нажмите **ОК**.
- Либо аналогичным образом можно установить просто фоновую заливку (**Формат/Фон/Цвет фона**).

Установите эмблему для Вашей странички. Для этого:

- Установите курсор в верхнюю левую ячейку таблицы.
- На вкладке **Вставка** выберите **рисунок/Из файла**.
- Выберите понравившуюся Вам картинку (можно анимационную). Можно ввести замещающий текст, например **Эмблема**.
- Если Вас не устраивает размер рисунка его можно изменить, установить рамку и др. настройки (Для этого необходимо вызвать контекстное меню, щелчком правой кнопки мыши по рисунку и открыть в появившемся окне вкладку **вид**).
- Сохраните результат Вашей работы. В окне **Сохранение внедренных файлов** нажмите **ОК**.

- Просмотрите результат в браузере (Файл/Просмотреть в обозревателе/выбрать обозреватель из списка).
  - Вернитесь в конструктор.
  - Нажмите ОК.
5. Рядом с ячейкой с эмблемой разместите текст с названием Вашего сайта (можно вставить картинку). Для этого:
    - Установите курсор в ячейку рядом с эмблемой.
    - Наберите текст заголовка. Например, Турбюро «КОМПАС»
    - Отформатируйте заголовок по вашему желанию.
    - Просмотрите результат в браузере.
  6. Заполните центральную часть<sup>3</sup> сайта, в соответствии с темой.
  7. Аналогичным образом оформите остальные страницы сайта, в соответствии с их названием (лучше использовать тот же шаблон макетной таблицы).

### **Задание№3** Создайте панель навигации.

#### **Технология работы:**

1. Откройте домашнюю страницу Вашего сайта в **MS SharePoint Designer 2007**.
2. Установите курсор в ячейку под эмблемой.
3. Выполните команду Вставка/Веб-компонент (рис.11)

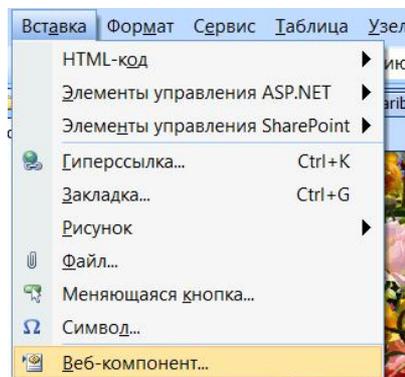


Рисунок 11

4. В открывшемся окне выберите Панели ссылок/Панель основанная на структуре переходов. Нажмите Далее.
5. Выберите стиль кнопок. Нажмите Далее.
6. Выберите вертикальную ориентацию, нажмите Ок.
7. В окне , в разделе Дополнительно установите галочку рядом с домашней страницей. Нажмите ОК

<sup>3</sup> Обратите внимание, область слева под эмблемой оставлена чистой для навигационной панели.

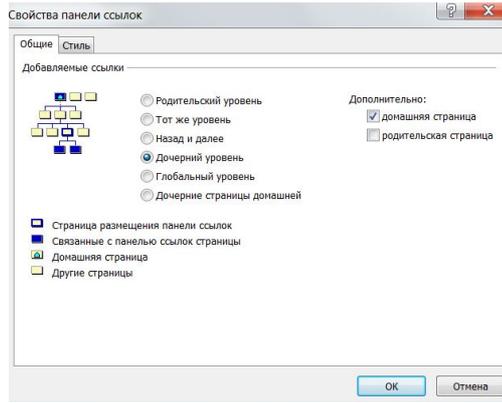


Рисунок 12

8. Просмотрите результат в браузере, при необходимости измените стиль панели.
9. Аналогичным образом разместите панель навигации (тот же уровень) на страницах –Страны, Острова, Заказы, на остальных создайте на них гиперссылки перехода на главную и на предыдущий уровень.
10. Примените на страницах сайта другие возможности Web-компонента (бегущая строка, фотогалерея и т.д)
11. Просмотрите результат в браузере, покажите задание преподавателю.

## Лабораторная работа №4: Графика в web . Создание анимационного баннера в программе Adobe Photoshop.

### Теоретический материал:

Размещение графических элементов обязательно для любого сетевого ресурса. Графические элементы визуально украшают web-страницы и сайты.

Подготавливая графику для web-страницы, необходимо помнить о небольших размерах области, с которой вы должны работать. Изображение не должно превышать 700 пикселей по ширине и 400 пикселей по высоте. Но файл даже такой картинки может оказаться слишком большим для передачи по каналам связи сети. Следует также использовать корректную цветовую гамму, это очень важно при разработке любого графического проекта. Для графики web-страниц необходимо применять цвета модели RGB .

Размер графического файла должен быть сравнительно небольшим, время его загрузки по сети должно не превышает одной минуты, что выражается в ограничении размера величиной примерно 60 килобайт.

### Графические форматы для Интернета

Любая компьютерная информация может храниться только в определенном формате. Каждый вид информации имеет собственные форматы. Для текста используются одни форматы, для электронных таблиц - другие, для графики - третьи. Формат графической информации обычно определяется программой, в которой она создана. Для web-графики стандартными считаются форматы **JPEG**, **GIF** и **PNG**.

Формат **JPEG** (Joint Photographic Experts Group - Объединенная группа экспертов фотографии) - самый популярный формат для хранения фотографических изображений. **JPEG** является стандартом в Интернете и предназначен для сохранения растровых или точечных изображений со сжатием, уменьшающим размер файла от десятых долей процента до 100 раз, за счет отбрасывания избыточной информации, не влияющей на отображение документа.

Формат **GIF** предназначен в основном для “рисованных” изображений: чертежей, графиков и т. д. Не стоит сохранять в формате **GIF**, например, многоцветные фотографии — размер файла останется довольно большим, а качество изображения заметно ухудшится за счет уменьшения количества цветов. Зато файлы, содержащие много одноцветных точек, расположенных рядом, сжимаются с помощью формата **GIF** до небольших размеров. Достоинства этого формата в том, что фон рисунка, сохраненного в формате может быть прозрачным. Это очень удобно для размещения на вашей странице логотипов и других графических элементов произвольной формы, где фоном служит фоновый цвет страницы. Так же загрузка файла **GIF** возможна чересстрочным методом. Если графический файл имеет большой

размер и грузится из Интернета долго, пользователь увидит сначала как бы нечеткие контуры будущего рисунка, а по мере загрузки изображение будет постепенно “проявляться”, что достигается очень простым приемом — изменением порядка загрузки строк изображения. И, наконец, еще одно достоинство **GIF**-файлов — они могут содержать не только статичные рисунки, но и целые анимационные фрагменты! На самом деле эти фрагменты представляют собой последовательности нескольких статичных кадров, а также информацию о том, сколько времени каждый кадр должен задерживаться на экране.

### Типы графических вставок в Web-страницы:

- ✚ Навигационные кнопки используются для содействия посетителям в перемещениях по сайту. Хорошо оформленные, они воспринимаются легче, чем текстовые ссылки.
- ✚ Логотип используется для идентификации корпоративного стиля. Логотип может существенно увеличить узнаваемость бренда или торговой марки.
- ✚ Маркеры списков используются для привлечения внимания посетителей к узловым точкам документов. Также часто используются для разделения параграфов и абзацев в документе.
- ✚ Титульная графика (мастхед) предоставляет посетителям информацию по Web-ресурсу. Также улучшает визуальное восприятие.
- ✚ Разделительные линии обычно используются для разделения монотонных текстов и облегчения их восприятия.
- ✚ Фоновые рисунки улучшают внешний вид страниц. Наиболее часто используется горизонтальная двухцветная полоска.
- ✚ Графический заголовок позволяет использовать независимые от установленных на клиентском компьютере шрифты.
- ✚ Фотографии обычно используются для придания более доброжелательного внешнего вида странице. Взгляд посетителей естественным образом притягивается к фотоизображениям.

### Баннеры

**Баннер** (от англ. banner - знамя, флаг) - это небольшая картинка или анимация, играющая роль гиперссылки на рекламируемый ресурс.

### Типы баннеров.

Существует достаточно большое количество типов баннеров. Это обусловлено тем, что любой баннер представляет собой графическое изображение или анимацию, подходов к созданию которых существует множество.

**GIF-баннер.** Основной плюс использования стандарта GIF — возможность рисовать анимированные баннеры. До широкого

распространения Flash это была единственная возможность создания динамических баннеров без использования программирования.

**JPEG-баннер.** JPEG-стандарт был создан для сохранения изображений фотографического качества в файлах небольшого размера. Он оптимально подходит для создания статичных баннеров с использованием фрагментов фотографий.

**Java-баннер.** Представляет собой обычный Java-апплет<sup>4</sup>, встроенный в HTML-документ. Ввиду технической сложности и большого размера использовать баннеры такого типа стоит лишь для достижения интерактивности. Замечено, что если при наведении указателя мыши на баннере будут происходить динамические изменения или проигрываться звук, то его эффективность окажется гораздо выше неинтерактивного аналога. В свое время были популярны баннеры с играми. Основными препятствиями к распространению баннеров такого типа стала сложность их изготовления и несовместимость с большинством баннерных сетей. Кроме того, с появлением Flash эффекты, реализуемые с использованием Java, стало гораздо проще воплощать с помощью ActionScript<sup>5</sup>.

**HTML-баннер.** Представляет собой маленькую интернет-страницу. Может содержать стандартные формы и объекты интерактивности. Оптимален для создания технически сложных баннеров. Например, сайт компании, производящей переводчики, лучше всего «раскручивать» с помощью баннера, содержащего строку, введя в которую русское слово можно будет, нажав соответствующую кнопку, получить его перевод. Недостаток HTML-баннера — техническая сложность изготовления (для этого зачастую требуется знание таких технологий, как PHP или Perl).

**Flash-баннер.** Наиболее молодой, перспективный. Основные преимущества Flash-технологии следующие.

- Позволяет создавать яркие, полноценные анимации при минимуме затраченных усилий.
- Размер Flash-анимаций во много раз меньше аналогичных, но реализованных с использованием, например, GIF. Чем сложнее и правдоподобнее мультипликация, тем больше проявляется это различие.
- Flash позволяет создавать интерактивные баннеры, причем куда проще, чем с использованием Java.

Баннер, выполненный с помощью Flash, по сумме своих достоинств (динамичность, красочность, интерактивность, небольшой размер, информативность) оставляет далеко позади любой другой тип. Поэтому Flash-баннеры намного популярнее у рекламодателей, чем все остальные.

<sup>4</sup> **Java-апплет** — прикладная программа, чаще всего написанная на языке программирования Java

<sup>5</sup> **ActionScript** — объектно-ориентированный язык программирования, который добавляет интерактивность, обработку данных и многое другое в содержимое Flash-приложений

У Flash-баннеров есть еще одно заметное преимущество — они загружаются даже тогда, когда пользователь отключает отображение графических элементов. В этом случае их эффективность резко возрастает: на фоне полного отсутствия на странице графики яркий Flash неизбежно привлечет внимание.

### Практикум

Для работы с графикой в Web существуют специальные программы. Безусловным лидером среди них, на мой взгляд является программа Adobe Photoshop.

При помощи этой программы можно достаточно быстро создавать не только статичные графические элементы, но небольшие анимационные заставки в формате GIF.

Рассмотрим пример создания небольшой графической анимационной заставки.

**Задание №1** Создать небольшой анимационный баннер с движения цвета по тексту.

#### Технология работы:

1. Запустите программу Adobe Photoshop CS5.
2. Создайте новый файл (Файл/Создать).
3. Установите размер 400×100 пикселей.
4. Выберите на панели инструментов **инструмент текст** Рис.13.

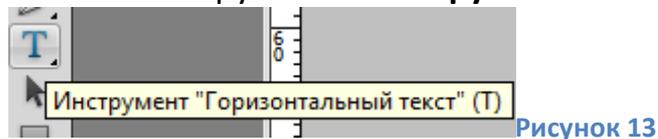


Рисунок 13

5. Щелкните на белом поле и напишите текст, аналогичный тому который был написан на баннере сайта (Например Мой сайт).
6. Выделите текст и отформатируйте по своему усмотрению.
7. Установите понравившийся цвет.
8. Откройте панель слоев (Окно/Слой).
9. Создайте новый слой (Слой/Новый/Слой) Рис.14. Выделите его.

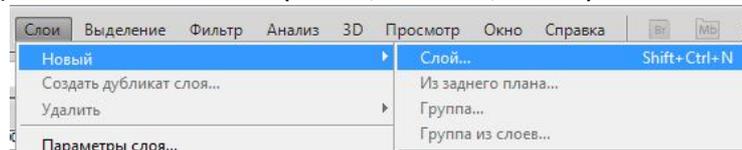


Рисунок 14

10. Выберите в палитре другой цвет и нарисуйте прямоугольник перед текстом (Используя инструмент Прямоугольник). Рис.15.



Рисунок 15

11. Отключите слой фона (на панели слоев).

12. Откройте панель анимации (Окно/Анимация).
13. Перейдите в режим **Покадровая анимация**. Рис.16

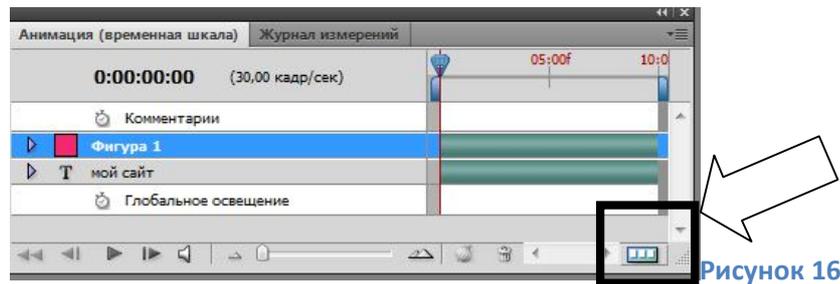


Рисунок 16

14. Установите длительность кадра 0,1 сек.
15. Создайте копию выделенного кадра.
16. Перенесите на этом слое прямоугольник в конец текста.
17. Выполните команду **Слой/Создать обтравочную маску**. Линия исчезнет.
18. На панели анимация нажмите кнопку **Создать промежуточные кадры**.
19. Кол-во кадров равно 20.
20. Запустите воспроизведение, посмотрите результат.
21. Сохраните файл используя команду **Сохранить для Веб и устройств** под именем `Baner.gif` в папке `images` Вашей папки `site`.
22. Откройте созданный Вами сайт и установите на стартовую страницу анимированный баннер.
23. Просмотрите результат в обозревателе.

**Задание№2** Самостоятельно создайте статичный логотип для вашего сайта и разместите его на своем сайте.

### **Лабораторная работа №5: Работа с формами.** **Теоретический материал**

В ранее выполненных лабораторных работах Вы освоили технологии, позволяющие *предоставлять* информацию посетителю Web-сайта. Но при разработке любой программы одна из главных задач, стоящих перед программистом, - это создание механизма *взаимодействия* с пользователем. Вам нужно научиться *получать* информацию от посетителя Web-сайта, а затем обрабатывать эту информацию. В HTML для получения данных посетителя предусмотрены формы и их элементы. В этой лабораторной работе Вы освоите технологию создания форм для сбора информации посетителей и технологию обработки данных формы на сервере .

**Форма** - это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку, где информация будет некоторым образом обработана. Формы превращают Web в интерактивную среду. После того, как посетитель заполняет предложенную ему форму, данные, введенные им в поля формы, отправляются серверной программе для последующей обработки.

### **Создание формы**

Форма организуется с помощью тега `<FORM>` и `</FORM>`. HTML-документ может содержать в себе несколько форм, однако формы не следует вкладывать друг в друга.

С помощью атрибута `ACTION` тега `<FORM>` указывается URL (интернет-адрес), по которому следует передать введенную в форму информацию. Это обязательный атрибут.

С помощью атрибута `METHOD` тега `<FORM>` указывается, каким образом (иначе говоря, с помощью какого метода протокола передачи гипертекстов) данные из формы будут переданы обработчику. Существует два основных метода передачи данных: `POST` и `GET`.

Значение атрибута `METHOD=GET` заставляет браузер передать все данные из формы по URL, заданному в `ACTION`. При этом к адресу добавляется только знак вопроса и соответствующие (закодированные) данные, введенные в форму при заполнении. Таким образом, данные из формы передаются серверу *вместе* с URL сценария обработчика формы. Этот метод удобен для небольших форм.

Значение атрибута `METHOD=POST` заставляет браузер прежде всего связаться с сервером, обрабатывающим форму. Как только связь установится (получено подтверждение), браузер передает сами данные. Таким образом, данные из формы передаются серверу *отдельно* от URL сценария обработчика формы.

Существует три типа элементов формы: `<TEXTAREA>`, `<SELECT>`, `<INPUT>`.

**Элемент `<TEXTAREA>`** Этот элемент служит для создания многострочного поля ввода. Элемент `<TEXTAREA>` парный, у него имеются несколько атрибутов:

- `NAME` - обязательный атрибут, определяющий имя, под которым содержимое поля будет передано обработчику;
- `ROWS` - атрибут, устанавливающий высоту окна в строках;
- `COLS` — атрибут, устанавливающий ширину окна в символах.

Многострочные поля ввода можно использовать, например, для ввода комментариев посетителя.

### **Элемент `<SELECT>`**

Этот тег служит для организации разворачивающихся меню и полей списка. Тег <SELECT> парный, у него имеются следующие атрибуты:

- NAME - обязательный, определяющий имя переменной, которую генерирует меню;
- MULTIPLE - устанавливает режим выбора нескольких значений из списка;
- SIZE - устанавливает число строк, отображаемых одновременно.

Между тегами <SELECT> и </SELECT> находятся значения, которые может выбирать пользователь. Они перечислены в теге <OPTION>. Тег <OPTION> - одиночный элемент, не требующий закрывающего тега.

Тег <OPTION> определяет элемент меню. У тега <OPTION> имеются следующие атрибуты:

- VALUE - обязательный, устанавливающий значение, которое будет передано обработчику, если выбран этот элемент меню;
- SELECTED - показывает, что данный элемент отображается выбранным.

### Элемент <INPUT>

- Этот элемент предоставляет возможность использовать другие типы ввода информации: отдельные строки текста, переключатели, флажки и кнопки передачи запроса и отказа от ввода. Это одиночный элемент, не требующий закрывающего тега.
- У тега <INPUT> имеются следующие атрибуты: NAME, SIZE, MAXLENGTH, VALUE, CHECKED, TYPE.

Типы элементов <INPUT> определяются атрибутом TYPE.

TYPE= "RADIO" - определяет переключатель. Используется в тех случаях, когда необходимо выбрать только один из родственных элементов. Это значение атрибута TYPE используется с атрибутами NAME (обязателен), VALUE (определяет значение, возвращаемое серверу в случае выбора данной кнопки), CHECKED (показывает, что кнопка выбрана по умолчанию). Присвоив одинаковые значения атрибуту NAME разных переключателей их можно объединить в группу. В группе радиокнопок осуществляется выбор только одной позиции (например в тесте для вопроса может быть только один вариант ответа).

TYPE="CHECKBOX" - кнопка с независимой фиксацией. Используется в тех случаях, когда одновременно можно выбрать несколько опций. Атрибуты: NAME (обязателен), VALUE, CHECKED.

### Практикум

**Задание№1** Используя элемент формы <TEXTAREA> создайте многострочное поле для ввода текста.

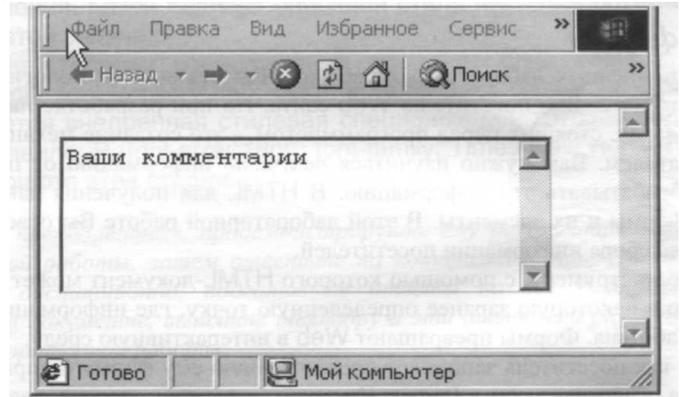
**Технология работы:**

1. Запустите текстовый редактор Блокнот.
2. Наберите в нем код представленный на листинге 4.
3. Внимательно изучите использование тега <TEXTAREA>и его атрибутов.
4. Сохраните этот документ в своей папке под именем 1.htm
5. Просмотрите результат в браузере (рис.17).

```

<HTML>
<HEAD> <TITLE> форма1</TITLE>
</HEAD>
<BODY>
<textarea name="address"
rows="5" cols="30"> Ваши
комментарии </textarea>
</BODY>
</HTML>

```



листинг 4

Рисунок 17

**Задание№2** Используя элемент формы <SELECT> создайте разворачивающееся меню из четырех пунктов (Выбор страны проживания).

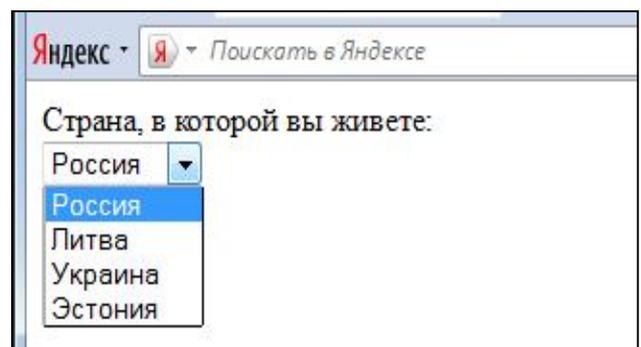
**Технология работы:**

1. Запустите текстовый редактор Блокнот.
2. Наберите в нем код представленный на листинге 5.
3. Внимательно изучите использование тега < SELECT >и его атрибутов.
4. Сохраните этот документ в своей папке под именем 2.htm.
5. Просмотрите результат в браузере (рис.18).

```

HTML>
<HEAD> <TITLE> форма2</TITLE>
</HEAD>
<BODY>
Страна, в которой вы живете:
<form>
<select name="country">
  <option value="Россия">Россия
  <option value="Литва">Литва
  <option value="Украина">Украина
  <option value="Эстония">Эстония </select>
</form>
</BODY>
</HTML>

```



листинг 5

Рисунок 18

**Задание№3** Проверьте самостоятельно работоспособность предложенного в примере HTML-кода. Выясните, как влияют на вид формы атрибуты тега <SELECT> SIZE и MULTIPLE и атрибут тега <OPTION> SELECTED

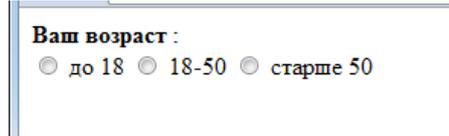
**Задание№4** Самостоятельно создайте разворачивающийся список состоящий из 10 крупных городов России, сохраните web- страницу в своей папке.

**Задание№5** Используя элемент формы <INPUT> создайте часть анкеты, в которой пользователь указывает свой возраст.

**Технология работы:**

1. Запустите текстовый редактор Блокнот.
2. Наберите в нем код представленный на листинге 6.
3. Внимательно изучите использование тега < INPUT >и его атрибутов.
4. Сохраните этот документ в своей папке под именем 3.htm.
5. Просмотрите результат в браузере (рис.19).

```
<HTML>
<HEAD> <TITLE> форма3 </TITLE> </HEAD>
<BODY>
<B> Ваш возраст : </B>
<form>
<input type="radio" name="vozrast" value="до 18"> до 18
<input type="radio" name="vozrast" value="18-50"> 18-50
<input type="radio" name="vozrast" value="старше 50"> старше 50
</form>
</BODY>
</HTML>
```



листинг 6

Рисунок 19

**Задание№6** Оформите предыдущее задание в виде таблицы, используя теги языка HTML для создания таблицы.

**Технология работы:**

1. Запустите текстовый редактор Блокнот.
2. Наберите в нем код представленный на листинге 7.
3. Обратите внимание как теги создания таблицы используются вместе с тегами формы. Таблица состоит из 3 строк (тег <tr>), каждая из которых содержит 2 ячейки (тег <td>).
4. Сохраните этот документ в своей папке под именем 4.htm.
5. Просмотрите результат в браузере (рис.20).

```
HTML>
<HEAD> <TITLE> форма3 </TITLE>
```

```

</HEAD>
<BODY>
<B> Ваш возраст </B> :
<form>
  <table border=1>
    <tr>
      <td> до 18 </td>
      <td> <input type="radio" name="vozrast" value="до 18"> </td>
    </tr>
    <tr>
      <td> 18-50 </td>
      <td> <input type="radio" name="vozrast" value="18-50"><td>
    </tr>
    <tr>
      <td> старше 50 </td>
      <td> <input type="radio" name="vozrast" value="старше 50"> </td>
    </tr>
  </table>
</form>
</BODY>
</HTML>

```

листинг 7

Рисунок 20

И в 5 и 6 задании выбор возможен только одного варианта.

**Задание№7** Используя значение атрибута TYPE="CHECKBOX" напишите программу содержащую часть анкеты с возможностью выбора нескольких вариантов ответа (выбор понравившейся породы собак). Результат представлен на рис.21. Результат сохраните в своей папке под именем 5.htm.

Рисунок 21

### Значения атрибута TYPE

Для того, чтобы в дальнейшем сайт был интерактивен необходимо добавлять на страницы командные кнопки. Для создания их необходимо присвоить атрибуту TYPE определенные значения.

Например **TYPE="SUBMIT"** - командная кнопка, которая отправляет на сервер все введенные данные. Данные отправляются по адресу, указанному в атрибуте ACTION тега <FORM>. Надпись на кнопке можно задать путем введения атрибута VALUE ="Надпись", например:

**<input type="submit" value="отправить">**.

**TYPE= "RESET"** - определяет кнопку, при нажатии на которую форма возвращается в исходное состояние. В этом случае данные обработчику не передаются, поэтому кнопка может и не иметь атрибута NAME. Надпись на кнопке задается при помощи атрибута VALUE="Надпись".

**TYPE= "TEXT"** - определяет окно для ввода строки текста. Это значение атрибута TYPE можно использовать с атрибутами NAME (обязателен), VALUE, SIZE и MAXLENGTH.

**TYPE= "PASSWORD"**- определяет окно для ввода пароля. Абсолютно аналогичен типу TEXT, только вместо символов вводимого текста показывает на экране звездочки (\*).

**TYPE= "BUTTON"**- не выполняет никаких действий при нажатии на кнопку, если к ней не прикреплена программа. Это значение атрибута TYPE используется с атрибутами NAME (имя кнопки), VALUE (надпись на кнопке), ONCLICK (для указания имени программы, которая будет выполняться при нажатии на эту кнопку).

**Задание№8** Создайте часть анкеты, позволяющую клиенту вводить информацию о себе, для этого совместите использование тегов таблицы и формы <INPUT> с атрибутом **TYPE= "TEXT"** Результат на рис15.

#### Технология работы:

1. Запустите текстовый редактор Блокнот.
2. Наберите в нем код представленный на листинге 8.
3. Внимательно изучите использование тега < INPUT >и его атрибутов.
4. Сохраните этот документ в своей папке под именем 6.htm.
5. Просмотрите результат в браузере (рис.22).

```
<HTML>
  <HEAD> <TITLE> форма7 </TITLE> </HEAD>
<BODY>
<B> Заполните поля формы: </B>
<form>
```

```

<table border=1>
  <tr>
    <td> Фамилия, Имя, Отчество* </td>
    <td> <input type="text" name="fio" "> </td>
  </tr>
  <tr>
    <td> Телефон* </td>
    <td> <input type="text" name="tel" ><td>
  </tr>
  <tr>
    <td> E-mail </td>
    <td> <input type="text" name="mail" > </td>
  </tr>
  <tr>
    <td> Адрес </td>
    <td> <input type="text" name="adr" > </td>
  </tr>
</table>
</form>
</BODY>
</HTML>

```

листинг 8

Заполните поля формы :

Фамилия, Имя, Отчество*	<input type="text"/>
Телефон*	<input type="text"/>
E-mail	<input type="text"/>
Адрес	<input type="text"/>

Рисунок 22

**Задание№9** Добавьте в файл, созданный в предыдущем задании 2 кнопки - **Проверка** и **Очистить поля формы**, используя тег формы `input` и его атрибут `type` (`<input type="submit" value="Проверка">` , `<input type="reset" value="Очистить поля формы">`). Результат представлен на рис.23.

Яндекс

Заполните поля формы :

Фамилия, Имя, Отчество*	<input type="text"/>
Телефон*	<input type="text"/>
E-mail	<input type="text"/>
Адрес	<input type="text"/>

Рисунок 23

Сохраните изменения в файле. Просмотрите результат в браузере. Проверьте как работает кнопка **Очистить поля формы**, кнопка **Проверка**

пока не активна, ее настройками мы займемся в следующей лабораторной работе.

**Задание №10** Создайте web- страничку<sup>6</sup>, содержащую форму для регистрации рис.24. Можно обойтись без создания таблицы, используя тег для текстового абзаца <p>. Строка **Дата рождения** содержит разворачивающееся меню, для даты (от 1 до 31), месяца( все месяца) и года (например от 1960 и до 2000) Сохраните ее в своей папке под именем **Анкета.htm** .

Рисунок 24

**Задание №11** Дополните файл Анкета двумя кнопками – **Регистрация** и **Очистить поля формы**, используя тег формы input и его атрибут type. Сохраните изменения в файле. Просмотрите результат в браузере, заполните анкету для себя. Проверьте действие кнопки **Очистите поля формы**. Кнопка Регистрация на данный момент не работает. Ее функции мы настроим в следующей лабораторной работе.

## Лабораторная работа №6: Использование для обработки форм скриптового языка JavaScript.

### Теоретический материал

Выполняя предыдущую лабораторную работу, Вы приступили к процессу создания интерактивной части сайта. У Вас разработана форма для сбора сведений о посетителе сайта. Предполагается, что обработка данных из этих форм будет производиться на сервере и ее целью будет занесение в

<sup>6</sup> Можно копировать код из предыдущих заданий.

базу данных сведений о посетителе. Однако разработчики перед передачей данных на сервер обычно проводят проверку этих данных на компьютере клиента. Это помогает избежать передачи неправильных данных от клиента на сервер и обратно. Для реализации такой проверки обычно используют *клиентские языки сценариев для Web*. Мы с Вами будем использовать JavaScript.

Итак, цель данной лабораторной работы.- познакомиться с технологией проверки данных формы на компьютере клиента средствами JavaScript.

### **Назначение JavaScript**

JavaScript - интерпретируемый язык создания клиентских сценариев.

Предназначен этот язык, как правило, для создания небольших участков кода. Синтаксис похож на синтаксис языка Си. В JavaScript реализованы основные возможности объектно-ориентированных языков.

### **Задачи, решаемые с помощью JavaScript:**

1. Проверка данных в интерактивной форме;
2. Выполнение простейших вычислений, связанных с настройками пользователя и браузера;
3. Контроль за событиями, происходящими в окне браузера;
4. Доступ ко всем HTML-элементам при помощи объектной модели документа.

JavaScript - объектно-ориентированный язык, поэтому в нем определен ряд объектов, относящихся к различным компонентам HTML-страницы и их взаимосвязи. Чтобы узнать о состоянии объекта или управлять им, следует пользоваться определенными свойствами и методами.

JavaScript предоставляет возможность контроля за событиями, происходящими в окне браузера, и доступа к HTML-элементам при помощи объектной модели документа. Язык **JavaScript** используется на стороне клиента, проще говоря в Интернет браузере пользователя. Для проверки сценариев на сервере используются другие языки, такие как язык **php**.

### **Размещение сценария**

Сценарии могут размещаться в четырех различных частях программы документа HTML.

1. *В теле программы.* В этом случае результат сценария отображается на Web-странице при ее загрузке в браузер. JavaScript программа помещается между тегами <SCRIPT> и </SCRIPT>. Встретив тег <SCRIPT>, браузер построчно анализирует содержимое документа до тех пор, пока не будет достигнут тег </SCRIPT>. После этого производится проверка скрипта на наличие ошибок и компиляция JavaScript программы в формат, пригодный для выполнения на компьютере пользователя Например: (листинг 9).  
<html>

```

<body>
  <script>
    alert ("Здравствуйете, я - JavaScript.")
  </script>
</body>
</html> листинг 9

```

2. В заголовке программы между парой тегов <HEAD>. Сценарий, размещенный в заголовке, не выполняется при загрузке страницы. В этом случае он используется как функция - группа операторов JavaScript, выполняемых как одно целое.

3. В теге HTML. Такая конструкция называется обработчиком событий и позволяет выполнять сценарий JavaScript вместе с тегом. Обработчик событий представляет собой отдельный тип сценария, который не требует использования тега <SCRIPT> для его обозначения.

4. В отдельном файле. JavaScript позволяет создавать собственные файлы с расширением .js, содержащие готовые сценарии. В этом случае сценарий указывается в программе в виде имени файла, приведенного между тегами <SCRIPT>. Поскольку следует разделять структуру и содержание Web-страницы, сценарии лучше размещать во внешнем файле с расширением \*.js. Это упрощает процесс обновления сценария, скрывает сценарий от старых версий браузеров.

В этом случае обращение к сценарию из разных HTML-документов выглядит следующим образом: <SCRIPT SRC="\*.js" LANGUAGE="JavaScript">.

Атрибут SRC тега <SCRIPT> определяет имя файла сценария, с которым необходимо связать HTML-документ.

Атрибут LANGUAGE тега <SCRIPT> определяет используемый язык сценариев. С версии HTML 4.0 консорциум W3C предлагает использовать TYPE вместо LANGUAGE, потому что его значения стандартизированы (TYPE="TEXT/JavaScript").

### **Выполнение сценария**

Существует два варианта сценариев:

- 1) сценарий, который нужно выполнять при загрузке страницы;
- 2) сценарий, который выполняется в ответ на действия пользователя.

Чтобы не дать коду сценария быть выполненным сразу же после загрузки, его помещают внутрь функции.

### **Обработка браузером события**

Когда пользователь открывает Web-страницу, браузер отмечает наличие каждого сценария в HTML-документе и выполняет различные сценарии в разное время. Обычно он ждет появления события, которое Вы связали с функцией, чтобы выполнить ее. *События - это все, что может происходить на Web-странице.* Причиной большинства событий являются

действия мыши - щелчок и перемещение. На рис 25. проиллюстрирована концепция управления событиями

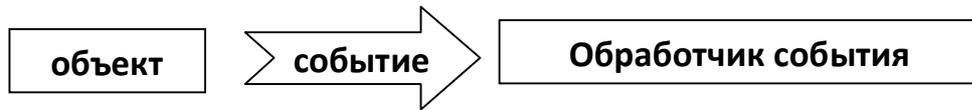


Рисунок 25

На этом рисунке представлены три ключевые области: объект, собственно событие и обработчик события. Их краткое описание:

- ✚ объектом может быть все, что угодно на Web-странице - прежде всего кнопки, однако генерировать события может также любой HTML-элемент;
- ✚ объекты вызывают события в ответ на действия пользователя. Например, если пользователь щелкает по кнопке, кнопка вызывает событие click (щелчок);
- ✚ обработчики событий предпринимают определенные действия в ответ на события. Для создания обработчика события в соответствии этому событию ставится некоторая функция, которую браузер будет выполнять всякий раз, когда произойдет событие.

Предположим, что необходимо связать определенный объект с обработчиком события объекта. Для этого нужно добавить атрибут события к объекту HTML и установить его значение равным функции (или тексту функции). Например, чтобы связать функцию JavaScript по имени Myfunction() с событием кнопки OnClick, описываем тег кнопки следующим образом:

```
<input type="button" name="button" onclick="Myfunction()">
```

Примеры событий с которыми можно связать сценарий приведены в таблице №1 .

Название	Когда вызывается событие
On Load	Браузер заканчивает открытие документа HTML
OnUnload	Браузер выгружает документ HTML
OnClick	Пользователь щелкнул мышью по элементу
OnSelect	Пользователь выбирает текст в текстовом поле

Таблица 1

Итак, событие - это результат определенных действий пользователя. Событие генерируется тогда, когда пользователь совершает действие. События могут *перехватываться* и обрабатываться обработчиками событий JavaScript. Для создания обработчика события в соответствии этому событию ставится некоторая функция, которую браузер будет выполнять всякий раз, когда произойдет событие. Обычно функции, составляющие программу, написанную на языке JavaScript, располагаются в секции **<HEAD>** и **</HEAD>**. Так как эта секция загружается первой, гарантируется, что эти функции будут

загружены раньше, чем пользователь сможет их вызвать с помощью средств располагаемых в секции <BODY> и </BODY>.

Шаблон описания функции:

```
function имя_функции (список параметров)
{
    Описание функции
}
```

Тело функции заключается в операторные скобки. Имя функции должно быть уникальным для сценария и состоять из латинских букв, цифр (0-9) и символа подчеркивания. Имя не должно начинаться с цифры. *Прописные и строчные буквы различаются.*

### Практикум

**Задание №1** Используя язык JavaScript, создайте клиентский сценарий с функцией проверки заполнения обязательных текстовых полей и разместите его в коде программы, созданном в заданиях №8-9 предыдущей лабораторной работе файл (**6.htm**) .

#### Технология работы:

1. Откройте в браузере файл **6.htm** из Вашей папки, который содержит форму для ввода сведений о посетителе сайта (ФИО, адрес, телефон, E-mail) и две кнопки: **Проверка** и **Очистить поля формы**, обратите внимание, что обязательные поля для заполнения отмечены звездочкой.
2. Откройте этот же файл при помощи блокнота и просмотрите код программы.

В этой программе требуется, чтобы обязательно были заполнены два поля: ФИО и телефон. Если оба поля, либо одно из них не заполнено, то регистрация на данном сетевом ресурсе невозможна, должно быть выведено сообщение типа: «пожалуйста, заполните обязательные поля для заполнения».

 **Объектом** в данном случае является кнопка **Проверка**.

 **Событием** является щелчок мыши по этой кнопке.

 **Обработчик событий** (проверка заполнения полей) реализуется через функцию **prov ()**.

Для данной задачи эта функция записывается следующим образом (листинг 10).

```
<script>
function prov()
{
    if (document.forms[0].fio.value=="" |
        document.forms[0].tel.value=="")
        {alert ("пожалуйста, заполните обязательные для заполнения поля")}
}
</script>
```

### листинг 10

Функция `prov()` использует три оператора:

- ✚ оператор сравнения `"=="`;
  - ✚ логический оператор `«|»` - «или», который сравнивает несколько булевых значений (результатов работы операторов сравнения) и возвращает `TRUE`, если хотя бы одно значение истинно;
  - ✚ оператор принятия решений `if` (если).
3. Необходимо разместить сценарий, записанный в листинге 10 в коде программы файла `6.htm`. В теоретическом материале были рассмотрены четыре способа размещения. Мы используем второй: размещение в заголовке программы между парой тегов `<HEAD>`. Для этого:
    - ✚ Установите курсор, после закрывающегося тега `</TITLE>` (лучше перейти на новую строку).
    - ✚ Наберите сценарий (листинг 10).
  4. Теперь необходимо связать кнопку Проверка с обработчиком событий `OnClick` (таб.1) этой кнопки. Для этого необходимо добавить атрибут события `onclick` к объекту `button` и установить его значение равным результату функции `prov()`:
 

```
<input type="button" value="проверка" onclick="prov()">
```

 Внесите эти изменения в код программы.
  5. Сохраните изменения в файле.
  6. Проверьте результат в браузере, заполните форму и оставьте свободным поле фамилия (или телефон), нажмите кнопку Проверка. Если все выполнено правильно, появится сообщение: *пожалуйста, заполните обязательные для заполнения поля.*

Давайте разберем подробно работу функции `prov()` и синтаксис используемых операторов.

Проверка данных реализована с помощью оператора принятия решений `if`. Этот оператор оценивает условие и принимает решение о выполнении блока операторов в зависимости от результата. В нашей форме, если не заполнено хотя бы одно из обязательных полей, с помощью метода `ALERT` будет выдано сообщение *"пожалуйста, заполните обязательные для заполнения поля"*. Оператор `if` применяется следующим образом:

```
if (условие)
  {{операторы1}}
else
  {{операторы2}}
```

В качестве условия может указываться любое логическое выражение. Условие представляет собой критерий, на основании которого производится

выбор. Это выражение, которое равно TRUE или FALSE. Если результат условия равен TRUE, выполняются *операторы1*. Если условие возвращает FALSE, выполняются *операторы2*. Обычно набор операторов заключается в фигурные скобки. Это придает сценарию логический вид и оказывается особенно полезным в случае вложенных операторов if.

Теперь разберем строку кода `document.forms[0].fio.value`. С помощью данного оператора осуществляется доступ к свойству `value` объекта TEXT с именем `fio` (свойство VALUE определяет содержимое поля). Доступ осуществляется в соответствии с точечной нотацией. Точечная нотация обеспечивает иерархический способ доступа к свойствам объектов.

*Примечание:* если объект является свойством другого, большего объекта, его имя упоминается вслед за именем большего объекта. Имена свойств, методов и обработчиков событий объекта указываются после имени самого объекта.

Следовательно, строка кода `document.forms[0].fio.value` содержит следующие компоненты:

1. Объект `document` (здесь первый в иерархии) создается при загрузке страницы и содержит информацию о текущем документе (заголовок, фон, цвет, формы). Объект `document` содержит массивы `anchors`, `forms` и `links`. С помощью этих массивов можно обратиться к каждому из объектов типа `anchor`, `form` и `link`, содержащихся в документе. В нашем случае обращение происходит к объекту `form`.
2. Объект `form` является свойством объекта `document` и представляет собой массив, содержащий все формы, определенные в документе. К нему можно обратиться либо по имени, определенному его атрибутом NAME, либо как к элементу массива: `document.forms[N]`, где N - номер формы в документе, считая от нуля. (У нас форма в документе одна, поэтому ее номер - 0);

**Задание №2** В задании №6 предыдущей лабораторной работы была создана форма, содержащая сведения о возрасте посетителя сайта (файл 4.htm). Дополните код программы сценарием на языке JavaScript (Листинг11) и кнопкой **Проверка**, с соответствующими настройками. Необходимо, чтобы при регистрации несовершеннолетних, нажимая кнопку **Проверка** выходило сообщение: Извините заказы принимаются только у совершеннолетних. Задание выполнить аналогично **Заданию №1**.

```
script>
    function prov()
    {if (document.forms[0].vozrast [0].checked)
      {alert ("Извините, заказы принимаются только у совершеннолетних")}}
</script>
```

листинг 11

При помощи языка **JavaScript** можно не только проверять правильности заполнение форм регистрации, реализации обратной связи. Этот язык позволяет достаточно просто создавать небольшие визуальные эффекты на сайте (такие как снег или плавное появление элементов). Причем программный код этих эффектов не обязательно писать самостоятельно. Существует большое количество Интернет-ресурсов (<http://syblog.ru/sneg-na-sajte>), на которых размещаются уже написанные скрипты различных эффектов, достаточно просто скопировать их и вставить в код своей страницы.

**Задание №3** Разместить на домашней страничке, созданной в первой лабораторной работе визуальный эффект – падающий снег.

**Технология работы:**

1. Откройте при помощи блокнота свою страничку, созданную в первой лабораторной работе.
2. Напишите после тега <BODY> и его атрибутов, код скрипта падающего снега (Приложение №3), можно скачать код с Интернет-ресурса<sup>7</sup>.
3. Сохраните изменения в файле.
4. Просмотрите результат в браузере.

### **Лабораторная работа №7: Создание и заполнение блога.**

#### **Теоретический материал:**

Термин "Блог" - blog - происходит от английского слова, обозначающего действие - Web-logging или блоггинг - вход во Всемирную паутину или веб, в которой человек ведет свою коллекция записей. Как правило, это личные записи, напоминающие дневник. Часто в записях содержатся аннотированные ссылки на другие ресурсы, опубликованные в сети.

**Каждое сообщение, опубликованное внутри блога, имеет свой URL - адрес, по которому к сообщению можно обратиться.** Этот простой признак – устойчивость ссылки - играет важную роль при установлении отношений между людьми и сообщениями. Если у сообщения нет устойчивого сетевого адреса, то оно не имеет статуса сетевого документа. На такое сообщение нельзя сослаться из другого сетевого документа, и оно не может быть найдено программными агентами.

Простота публикации и внесения записей в сетевой дневник обеспечили приток новых авторов. Кроме того, выяснилось, что люди с большим интересом читают новости и заметки, подготовленные не в целях рекламы и продвижения собственного имени. Ясность и доступность блога вызывают интерес многих исследователей, которые рассматривают его как

<sup>7</sup> Например: <http://syblog.ru/sneg-na-sajte>

вариант личного образовательного пространства. В блоге принят обратный порядок записи: самые свежие записи публикуются сверху. Для ведение блога нужен только доступ к сети и желание представлять свои материалы. Как правило, автором записей в блоге является один человек. Авторы нескольких блогов часто объединяются в социальную сеть, отслеживают записи друг друга, оставляют отзывы и заметки на полях чужих дневников. Возможность создать блог существует на многих ресурсах. Популярные платформы для ведения блогов – это: [Blogger](http://www.blogger.com/) (<http://www.blogger.com/>), [ЖЖ](http://www.ctrlc.ru/?p=50) (LiveJournal) (<http://www.ctrlc.ru/?p=50>), [Wordpress.com](http://wordpress.com/) (<http://wordpress.com/>), [MySpace](http://www.ctrlc.ru/?p=78) (<http://www.ctrlc.ru/?p=78>). Уже существуют и появляются социальные образовательные сети, в которых у каждого участника есть возможность вести свой собственный личный и профессиональный блог, делаясь с другими не только новостями, но и новыми знаниями, например: <http://www.openclass.ru/>. Для пользователей предоставляются бесплатные: учетная запись и аренда ресурсов сервера.

Для создания блога в данной лабораторной работе будем использовать платформу [Blogger](http://www.blogger.com/) (<http://www.blogger.com/>). Преимущество этой платформы:

- Blogger прост и удобен в обращении.
- Платформа Blogger по большинству критериев значительно безопаснее, чем, например, LiveJournal.
- Платформа Blogger интегрирована с почтой Gmail.
- В настройках Blogger есть возможность выбора множества языков, в том числе, русского.

**Задание** Создать личный блог, используя платформу для ведения блогов [blogger.com](http://www.blogger.com).

**Технология работы:**

1. Запустите браузер в строке адреса, наберите адрес платформы <http://www.blogger.com>.
2. В появившемся окне нажмите **Начать**.
3. Для создания блога на этой платформе необходимо создать **аккаунт Google**. Если аккаунт<sup>8</sup> уже есть, зайдите в него.
4. Заполните предложенную Вам анкету.
5. Нажмите продолжить.
6. Следующий шаг **Введите название блога**. Придумайте свой заголовок и введите его в поле Название блога.

---

<sup>8</sup> **Учётная запись** — запись, содержащая сведения, которые пользователь сообщает о себе некоторой компьютерной системе. (Материал из Википедии).

7. Если Вы делаете блог для себя, то заголовок может быть простым, но если Вы хотите привлечь внимание к своему блогу, постарайтесь дать ему яркое название, которое будет притягивать к нему посетителей.
8. Придумайте и введите английскими буквами Адрес блога (URL)(Рис.26).

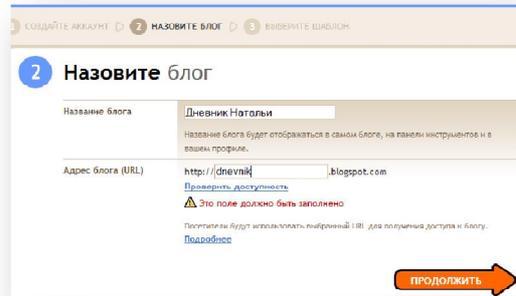


Рисунок 26

1. Нажмите **Проверить доступность**, для того чтобы выяснить свободен он или нет. Если этот адрес недоступен (занят) придумайте другой.
2. Если адрес блога не занят выпишите его к себе в тетрадь (полный адрес, например (dnevnic.blogspot.com)). Нажмите Продолжить.
3. Необходимо выбрать **Начальный шаблон** (Можно изменить шаблон позже или даже настроить его с помощью Редактора шаблонов). Выберите любой из понравившихся. Нажмите Продолжить.
4. Ваш блог создан, теперь вы можете добавить в него сообщения или настроить его внешний вид.
5. Нажмите Добавить сообщение.
6. Откроется встроенный текстовый редактор.
7. Введите название для сообщения и в текстовом поле текст сообщения, например (рис.27):

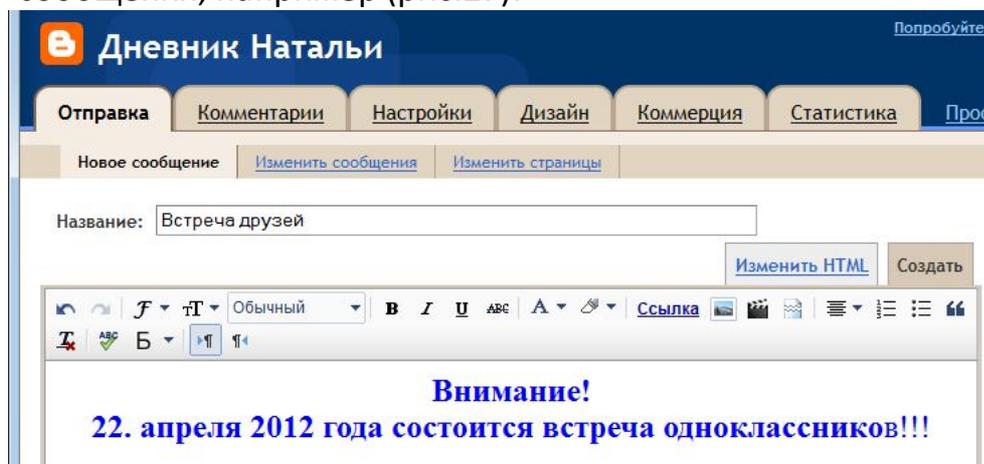


Рисунок 27

8. Нажмите **Опубликовать сообщение**.

ОПУБЛИКОВАТЬ СООБЩЕНИЕ

СОХРАНЕНО

ПРОСМОТР

9. Просмотрите сообщение и войдите обратно в блог. Можно просмотреть другие вкладки: Комментарии, Настройки, Дизайн и т.д.
10. Обменяйтесь с другими студентами адресами блогов, просмотрите адреса и добавьте свои комментарии.
11. Выйдите из блога. Для дальнейшей работы с блогом Вам достаточно знать адрес блога, логин (название почты) и пароль.
12. Разместите ссылку на свой блог на любой из страниц Вашего сайта (созданного в прошлой лабораторной работе).
13. Для активации аккаунта google, необходимо зайти в свой почтовый ящик и перейти по ссылке, которую администрация google Вам пришлет. Можно дополнительно настроить функции аккаунта.

## Лабораторная работа №8: Создание сайта на бесплатном сервисе Yandex - Народ.

**Задание №1:** Выполнить регистрацию на сайте narod.yandex.ru .

Почему-то считается, что конструктор бесплатных сайтов Яндекс.Народ является уделом новичков, но на самом деле Яндекс.Народ позволяет создавать профессиональные сайты и блоги. Тем более, что размещение сайтов на этом сервисе бесплатно.

**Технология работы:**

1. Зайдите на сайт <http://narod.yandex.ru>, внимательно ознакомьтесь с интерфейсом сайта.
2. Выберите раздел **Войти**, затем **Зарегистрироваться**<sup>9</sup>.
3. Заполните Яндекс-паспорт для получения логина и пароля доступа к Вашему сайту. Это очень важный момент, поскольку пароль - это единственная и самая важная защита вашей регистрации на Яндексе.
4. Запомните и запишите полученные логин и пароль.

**Задание №2:** Используя конструктор сайтов на сервисе народ создать свой сайт.

**Технология работы:**

1. Зайдите на сайт <http://narod.yandex.ru> под своим логином и паролем.
2. Просмотрите доступные сервисы и нажмите кнопку **Создать сайт** (рис.28).



Рисунок 28

<sup>9</sup> Придумайте логин и пароль. К этой задаче следует относиться серьезно, поскольку придуманный логин станет именем Вашего сайта. Полное имя будет выглядеть так **ваш\_логин.narod.ru**.

3. Выберите один из вариантов сайта и нажмите **Дальше** (лучше выбрать с чистого листа, оригинальный).
4. Заполните общие параметры сайта. Название, например **Личный**. Загрузите логотип сайта (заранее заготовленный логотип и сохраненный на Вашем компьютере, можно анимационную картинку). Нажмите **Дальше**.
5. Выберите один из стилей оформления (Например - авангардный), понравившийся шрифт и макет сайта. Нажмите **Начать заполнять**.
6. Используя Конструктор сайтов заполните эту страницу необходимым содержанием.
7. Добавьте новую страницу и заполните ее в соответствии с тематикой и т.д.

## Таблица основных тегов и атрибутов HTML

Основные теги	
<html></html>	Указывает программе просмотра страниц что это HTML документ.
<head></head>	Определяет место, где помещается различная информация не отображаемая в теле документа. Здесь располагается тег названия документа и теги для поисковых машин.
<body></body>	Определяет видимую часть документа
Теги оглавления	
<title></title>	Помещает название документа в оглавление программы просмотра страниц
Атрибуты тела документа	
<body bgcolor=?>	Устанавливает цвет фона документа, используя значение цвета в виде RRGGBB - пример: FF0000 - красный цвет.
<body text=?>	Устанавливает цвет текста документа, используя значение цвета в виде RRGGBB - пример: 000000 - черный цвет.
<body link=?>	Устанавливает цвет гиперссылок, используя значение цвета в виде RRGGBB - пример: 00FF00 - зеленый цвет.
<body vlink=?>	Устанавливает цвет гиперссылок на котох вы уже побывали, используя значение цвета в виде RRGGBB - пример: 333333 - серый цвет.
<body alink=?>	Устанавливает цвет гиперссылок при нажатии.
Теги для форматирования текста	
<pre></pre>	Обрамляет предварительно отформатированный текст.
<h1></h1>	Создает САМЫЙ БОЛЬШОЙ заголовок
<h6></h6>	Создает самый маленький заголовок
<b></b>	Создает жирый текст
<i></i>	Создает наклонный текст
<tt></tt>	Создает текст - имитирующий стиль печатной машинки.
<cite></cite>	Используется для цитат, обычно наклонный текст.
<em></em>	Используется для выделения из текста слова (наклонный или жирный текст)
<strong></strong>	Используется для выделения наиболее выжных частей текста (наклонный или жирный текст)
<font size=?></font>	Устанавливает размер текста в пределах от 1 до 7.
<font color=?></font>	Устанавливает цвет текста, используя значение цвета в виде RRGGBB.
Гиперссылки	
<a href="URL"></a>	Создает гиперссылку на другие документы или часть текущего документа.
<a href="mailto:EMAIL"></a>	Создает гиперссылку вызова почтовой программы для написания письма автору документа.
<a name="NAME"></a>	Отмечает часть текста как цель для гиперссылок в документе.
<a href="#NAME"></a>	Создает гиперссылку на часть текущего документа.

<b>Форматирование</b>	
<p>	Создает новый параграф
<p align=?>	Выравнивает параграф относительно одной из сторон документа, значения: left, right, или center
 	Вставляет перевод строки.
<blockquote> </blockquote>	Создает отступы с обеих сторон текста.
<dl></dl>	Создает список определений.
<dt>	Определяет каждый из терминов списка
<dd>	Описывает каждое определение
<ol></ol>	Создает нумерованный список
<li>	Определяет каждый элемент списка и присваивает номер
<ul></ul>	Создает ненумерованный список
<li>	Предваряет каждый элемент списка и добавляет кружок или квадратик.
<div align=?>	Важный тег используемый для форматирования больших блоков текста HTML документа, также используется в таблицах стилей
<b>Графические элементы</b>	
	Добавляет изображение в HTML документ
	Выравнивает изображение к одной из сторон документа, принимает значения: left, right, center; bottom, top, middle
	Устанавливает толщину рамки вокруг изображения
<hr>	Добавляет в HTML документ горизонтальную линию.
<hr size=?>	Устанавливает высоту(толщину) линии
<hr width=?>	Устанавливает ширину линии, можно указать ширину в пикселах или процентах.
<hr noshade>	Создает линию без тени.
<hr color=?>	Задаст линии определенный цвет. Значение RRGGBB.
<b>Таблицы</b>	
<table></table>	Создает таблицу.
<tr></tr>	Определяет строку в таблице.
<td></td>	Определяет отдельную ячейку в таблице.
<th></th>	Определяет заголовок таблицы (нормальная ячейка с отцентрованным жирным текстом)
<b>Атрибуты таблицы</b>	
<table border=#>	Задаст толщину рамки таблицы.
<table cellspacing=#>	Задаст расстояние между ячейками таблицы.
<table cellpadding=#>	Задаст расстояние между содержимым ячейки и ее рамкой.
<table width=#>	Устанавливает ширину таблицы в пикселах или процентах от ширины документа.

<code>&lt;tr align=?&gt;</code> или <code>&lt;td align=?&gt;</code>	Устанавливает выравнивание ячеек в таблице, принимает значения: left, center, или right.
<code>&lt;tr valign=?&gt;</code> или <code>&lt;td valign=?&gt;</code>	Устанавливает вертикальное выравнивание для ячеек таблицы, принимает значения : top, middle, или bottom.
<code>&lt;td colspan=#&gt;</code>	Указывает кол-во столбцов которое объединено в одной ячейке. (по умолчанию=1)
<code>&lt;td rowspan=#&gt;</code>	Указывает кол-во строк которое объединено в одной ячейке. (по умолчанию=1)
<code>&lt;td nowrap&gt;</code>	Не позволяет программе просмотра делать перевод строки в ячейке таблицы.
<b>Кадры</b>	
<code>&lt;frameset&gt;&lt;/frameset&gt;</code>	Предваряет тег <code>&lt;body&gt;</code> в документе, содержащем кадры;
<code>&lt;frameset rows="value,value"&gt;</code>	Определяет строки в таблице кадров, высота которых определена кол-вом пикселей или в процентном соотношении к высоте таблицы кадров.
<code>&lt;frameset cols="value,value"&gt;</code>	Определяет столбцы в таблице кадров, ширина которых определена кол-вом пикселей или в процентном соотношении к ширине таблицы кадров.
<code>&lt;frame&gt;</code>	Определяет единичный кадр или область в таблице кадров.
<code>&lt;noframes&gt;&lt;/noframes&gt;</code>	Определяет, что будет показано в окне браузера если он не поддерживает кадры.
<b>Атрибуты кадров</b>	
<code>&lt;frame src="inet/m1/URL"&gt;</code>	Определяет какой из HTML документов будет показан в кадре.
<code>&lt;frame name="name"&gt;</code>	Указывает Имя кадра или области, что позволяет перенаправлять информацию в этот кадр или область из других кадров.
<code>&lt;frame marginwidth=#&gt;</code>	Определяет величину отступов по левому и правому краям кадра; должно быть равно или больше 1.
<code>&lt;frame marginheight=#&gt;</code>	Определяет величину отступов по верхнему и нижнему краям кадра; должно быть равно или больше 1.
<code>&lt;frame scrolling=VALUE&gt;</code>	Указывает будет-ли выводиться линейка прокрутки в кадре; значение value может быть "yes," "no," или "auto". Значение по умолчанию для обычных документов - auto.
<code>&lt;frame noresize&gt;</code>	Препятствует изменению размеров кадра.
<b>Формы</b>	
<code>&lt;form&gt;&lt;/form&gt;</code>	Создает формы
<code>&lt;select multiple name="NAME" size=?&gt;&lt;/select&gt;</code>	Создает скролируемое меню. Size устанавливает кол-во пунктов меню, которое будет показано на экране, остальные будут доступны при использовании прокрутки.
<code>&lt;option&gt;</code>	Указывает каждый отдельный элемент меню
<code>&lt;select name="NAME"&gt;&lt;/select&gt;</code>	Создает ниспадающее меню
<code>&lt;option&gt;</code>	Указывает каждый отдельный элемент меню
<code>&lt;textarea name="NAME" cols=40 rows=8&gt;&lt;/textarea&gt;</code>	Создает окно для ввода текста. Columns указывает ширину окна; rows указывает его высоту.
<code>&lt;input type="checkbox" name="NAME"&gt;</code>	Создает checkbox. За тегом следует текст.
<code>&lt;input type="radio" name="NAME" value="x"&gt;</code>	Создает radio кнопку. За тегом следует текст.

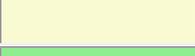
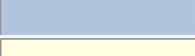
<code>&lt;input type="text" name="foo" size=20&gt;</code>	Создает строку для ввода текста. Параметром Size указывается длина в символах.
<code>&lt;input type="submit" value="NAME"&gt;</code>	Создает кнопку "Принять"
<code>&lt;input type="image" border=0 name="NAME" src="inet/m1/name.gif"&gt;</code>	Создает кнопку "Принять" - для этого используется изображение
<code>&lt;input type="reset"&gt;</code>	Создает кнопку "Отмена"

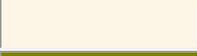
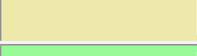
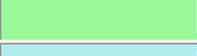
## Приложение №2

## Таблица цветов HTML

<i>Предопределенное имя</i>	<i>Шестнадцатеричное значение</i>	<i>Цвет</i>
aliceblue	#F0F8FF	
antiquewhite	#FAEBD7	
aqua	#00FFFF	
aquamarine	#7FFFD4	
azure	#F0FFFF	
beige	#F5F5DC	
bisque	#FFE4C4	
black	#000000	
blanchedalmond	#FFEBCD	
blue	#0000FF	
blueviolet	#8A2BE2	
brown	#A52A2A	
burlywood	#DEB887	
cadetblue	#5F9EA0	
chartreuse	#7FFF00	
chocolate	#D2691E	
coral	#FF7F50	
cornflowerblue	#6495ED	
cornsilk	#FFF8DC	
crimson	#DC143C	
cyan	#00FFFF	
darkblue	#00008B	
darkcyan	#008B8B	

darkgoldenrod	#B8860B	
darkgray	#A9A9A9	
darkgreen	#006400	
darkkhaki	#BDB76B	
darkmagenta	#8B008B	
darkolivegreen	#556B2F	
darkorange	#FF8C00	
darkorchid	#9932CC	
darkred	#8B0000	
darksalmon	#E9967A	
darkseagreen	#8FBC8F	
darkslateblue	#483D8B	
darkslategray	#2F4F4F	
darkturquoise	#00CED1	
darkviolet	#9400D3	
deeppink	#FF1493	
deepskyblue	#00BFFF	
dimgray	#696969	
dodgerblue	#1E90FF	
firebrick	#B22222	
floralwhite	#FFFAF0	
forestgreen	#228B22	
fuchsia	#FF00FF	
gainsboro	#DCDCDC	
ghostwhite	#F8F8FF	
gold	#FFD700	
goldenrod	#DAA520	
gray	#808080	
green	#008000	
greenyellow	#ADFF2F	
honeydew	#F0FFF0	
hotpink	#FF69B4	
indianred	#CD5C5C	
indigo	#4B0082	

ivory	#FFFFFF0	
khaki	#F0E68C	
lavender	#E6E6FA	
lavenderblush	#FFF0F5	
lawngreen	#7CFC00	
lemonchiffon	#FFFACD	
lightblue	#ADD8E6	
lightcoral	#F08080	
lightcyan	#E0FFFF	
lightgoldenrodyellow	#FAFAD2	
lightgreen	#90EE90	
lightgrey	#D3D3D3	
lightpink	#FFB6C1	
lightsalmon	#FFA07A	
lightseagreen	#20B2AA	
lightskyblue	#87CEFA	
lightslategray	#778899	
lightsteelblue	#B0C4DE	
lightyellow	#FFFFE0	
lime	#00FF00	
limegreen	#32CD32	
linen	#FAF0E6	
magenta	#FF00FF	
maroon	#800000	
mediumaquamarine	#66CDAA	
mediumblue	#0000CD	
mediumorchid	#BA55D3	
mediumpurple	#9370DB	
mediumseagreen	#3CB371	
mediumslateblue	#7B68EE	
mediumspringgreen	#00FA9A	
mediumturquoise	#48D1CC	
mediumvioletred	#C71585	
midnightblue	#191970	

mintcream	#F5FFFA	
mistyrose	#FFE4E1	
moccasin	#FFE4B5	
navajowhite	#FFDEAD	
navy	#000080	
oldlace	#FDF5E6	
olive	#808000	
olivedrab	#6B8E23	
orange	#FFA500	
orangered	#FF4500	
orchid	#DA70D6	
palegoldenrod	#EEE8AA	
palegreen	#98FB98	
paleturquoise	#AFEEEE	
palevioletred	#DB7093	
papayawhip	#FFEFD5	
peachpuff	#FFDAB9	
peru	#CD853F	
pink	#FFC0CB	
plum	#DDA0DD	
powderblue	#B0E0E6	
purple	#800080	
red	#FF0000	
rosybrown	#BC8F8F	
royalblue	#4169E1	
saddlebrown	#8B4513	
salmon	#FA8072	
sandybrown	#F4A460	
seagreen	#2E8B57	
seashell	#FFF5EE	
sienna	#A0522D	
silver	#C0C0C0	
skyblue	#87CEEB	
slateblue	#6A5ACD	

slategray	#708090	
snow	#FFFAFA	
springgreen	#00FF7F	
steelblue	#4682B4	
tan	#D2B48C	
teal	#008080	
thistle	#D8BFD8	
tomato	#FF6347	
turquoise	#40E0D0	
violet	#EE82EE	
wheat	#F5DEB3	
white	#FFFFFF	
whitesmoke	#F5F5F5	
yellow	#FFFF00	
yellowgreen	#9ACD32	

## Эффект падающего снега на JavaScript (материал с сайта <http://syblog.ru/sneg-na-sajte>)

```

<SCRIPT type="text/javascript">
// Количество снежинок на странице (Ставьте в границах 30-40, больше не
рекомендую)
var snowmax=35;
// Установите цвет снега, добавьте столько цветов сколько пожелаете
var snowcolor=new
Array("#AAAACC", "#DDDDFF", "#CCCCDD", "#F3F3F3", "#F0FFFF", "#FFFFFF", "#EFF5FF")
// Поставьте шрифты из которых будет создана снежинка ставьте столько
шрифтом сколько хотите
var snowtype=new Array("Arial Black", "Arial Narrow", "Times", "Comic Sans MS");
// Символ из какого будут сделаны снежинки (по умолчанию * )
var snowletter="*";
// Скорость падения снега (рекомендую в границах от 0.3 до 2)
var sinkspeed=0.6;
// Максимальный размер снежинки
var snowmaxsize=22;
// Установите минимальный размер снежинки
var snowminsize=8;
// Устанавливаем положение снега
// Впишите 1 чтобы снег был по всему сайту, 2 только слева
// 3 только по центру, 4 снег справа
var snowingzone=1;

/*
// * ПОСЛЕ ЭТОЙ ФРАЗЫ БОЛЬШЕ НЕТ КОНФИГУРАЦИИ*
*/
// НИЧЕГО НЕ ИЗМЕНЯТЬ
var snow=new Array();
var marginbottom;
var marginright;
var timer;
var i_snow=0;
var x_mv=new Array();
var crds=new Array();
var lftrght=new Array();
var browserinfos=navigator.userAgent;
var ie5=document.all&&document.getElementById&&!browserinfos.match(/Opera/);
var ns6=document.getElementById&&!document.all;
var opera=browserinfos.match(/Opera/);
var browserok=ie5||ns6||opera;
function randommaker(range) {
    rand=Math.floor(range*Math.random());
    return rand;
}
function initsnow() {
    if (ie5 || opera) {
        marginbottom=document.body.clientHeight;
        marginright=document.body.clientWidth;
    }
    else if (ns6) {
        marginbottom=window.innerHeight;
        marginright=window.innerWidth;
    }
    var snowsizerange=snowmaxsize-snowminsize;
    for (i=0;i<=snowmax;i++) {
        crds[i]=0;
        lftrght[i]=Math.random()*15;
    }
}

```

```

x_mv[i]=0.03+Math.random()/10;
snow[i]=document.getElementById("s"+i);
snow[i].style.fontFamily=snowtype[randommaker(snowtype/length)];
snow[i].size=randommaker(snowsizerange)+snowminsize;
snow[i].style.fontSize=snow[i].size+"px";
snow[i].style.color=snowcolor[randommaker(snowcolor.length)];
snow[i].sink=sinkspeed*snow[i].size/5;
if (snowingzone==1) {snow[i].posx=randommaker(marginright-
snow[i].size)}
if (snowingzone==2) {snow[i].posx=randommaker(marginright/2-
snow[i].size)}
if (snowingzone==3) {snow[i].posx=randommaker(marginright/2-
snow[i].size)+marginright/4}
if (snowingzone==4) {snow[i].posx=randommaker(marginright/2-
snow[i].size)+marginright/2}
snow[i].posy=randommaker(2*marginbottom-marginbottom-
2*snow[i].size);
snow[i].style.left=snow[i].posx+"px";
snow[i].style.top=snow[i].posy+"px";
}
movesnow();
}
function movesnow() {
for(i=0;i<=snowmax;i++) {
crds[i]+=x_mv[i];
snow[i].posy+=snow[i].sink;
snow[i].style.left=snow[i].posx+lftrght[i]*Math.sin(crds[i])+"px";
snow[i].style.top=snow[i].posy+"px";
if (snow[i].posy>=marginbottom-2*snow[i].size ||
parseInt(snow[i].style.left)>(marginright-3*lftrght[i])) {
if (snowingzone==1) {snow[i].posx=randommaker(marginright-
snow[i].size)}
if (snowingzone==2) {snow[i].posx=randommaker(marginright/2-
snow[i].size)}
if (snowingzone==3) {snow[i].posx=randommaker(marginright/2-
snow[i].size)+marginright/4}
if (snowingzone==4) {snow[i].posx=randommaker(marginright/2-
snow[i].size)+marginright/2}
snow[i].posy=0;
}
}
var timer=setTimeout("movesnow()",50);
}
for (i=0;i<=snowmax;i++) {
document.write("<span id='s"+i+"' style='position:absolute;top:-
"+snowmaxsize+"px;'>"+snowletter+"</span>");
}
if (browserok) {
window.onload=initsnow;
}
</SCRIPT>

```

**Список литературы:**

1. Гурский Д.А, Гурский Ю.А. Flash 8 и ActionScript. Библиотека пользователя - СПб.: Питер, 2006-528с.
2. Костин С.П. Самоучитель создания web-сайтов:быстрый старт. / С.П. Костин – М.: Изд-во Тримф, 2008-176с.
3. Супрун С.В., Долинер Л.И., Ченушкина С.В. Основы Web-программирования/ Учеб. пособие: В 2 ч. Ч.1.-Екатеринбург: Центр «Учебная книга», 2007.-82с.

**Интернет ресурсы:**

1. <http://www.vsetabl.ru/192.htm>
2. <http://www.javaportal.ru/htmlandcss/tablecolors.html>
3. <http://syblog.ru/sneg-na-sajte>