

ПОИСКОВО-ТВОРЧЕСКИЕ ЗАДАНИЯ ПО ВЕБ-ДИЗАЙНУ

А. М. Сиденко

Кто испытал наслаждение творчества, для того всех других наслаждений не существует!

А. П. Чехов

Представим, что наша задача предельно проста: человек, владеющий базовыми навыками и умеющий работать с компьютером, должен научиться создавать веб-страницы. «Научиться», в моём понимании, не значит приобрести все необходимые умения. Это значит понять методику решения типичных задач и разобраться, где именно искать ключи к решению новых нестандартных задач. Обязательно ли для этого нужна фантазия? Следует отметить, что большинство учителей представляют себе этапы создания элементарного сайта наподобие набора текстового документа в программе MS Word.

Счастливые люди, которые так считают, потому что они не знают основных этапов создания веб-сайта.

В моей практике было множество случаев, когда ожидали принудительного создания нескольких веб-страниц с соответствующим текстовым контентом, который ещё надо было набрать на компьютере и разместить на сервере. На это давали всего 3–4 часа.

А учительское время предусматривает перемены на проведение уроков, общение с учениками и корректировку этого самого материала, который необходимо отобразить в контенте этих самых страниц. Для этого давайте ознакомимся с этапами создания сайта, который должен отображаться в интернете.

ЭТАПЫ СОЗДАНИЯ ПОЛНОФУНКЦИОНАЛЬНОГО САЙТА

1. Проектирование графического интерфейса

Веб-дизайнер создаёт проект будущего веб-сайта. Если предполагается, что разные части сайта должны выглядеть по-разному, для этого необходимо иметь соответствующее количество разных по дизайну веб-страниц. Они должны соответствовать по своему назначению и использоваться названиями или разделам сайта.

2. Создание HTML-шаблонов

После того как перечень и назначение страниц определены, необходимо подготовить шаблоны для будущего веб-сайта, то есть в одном

из HTML-редакторов создать непосредственно веб-страницы, содержащие оптимизированную графику и место для текста, который предполагают разместить. Количество таких шаблонов соответствует количеству по-разному оформленных разделов сайта, которые вы спроектировали.

3. Начальное программирование

Веб-страницы должны быть интерактивными, то есть содержать элементы интерфейса, реагирующие на действия посетителя, которые не сводятся к активированию гиперссылок. Чаще всего используют так называемые JavaScript — скрипты, изменяющие внешний вид элементов страницы, над которыми находится курсор мыши.

4. Отображение контента (содержания)

В готовые шаблоны страниц вводят текст, после чего шаблоны сохраняют под разными именами, соответствующими названиям разделов сайта. Разрабатывают систему взаимных гиперссылок собственно на сайте (внутренние) и за его пределами (внешние). Внешние ссылки можно использовать при необходимости установить некоторые документы, слишком большие по размеру, но обязательные для пользователей данного сайта. Для подобной разработки веб-сайтов необходима полная запись метаданных (meta), которые не только идентифицируют документ, указывают принадлежность сайта определённой организации, набор ключевых слов (выражений), но и авторство общего дизайна сайта или авторство модератора сайта.

5. Служебная часть сервера

Очень часто возможности того или иного сайта обеспечивают соответствующие настройки сервера. Функции клиентских скриптов бывают очень ограниченными, преимущественно по соображениям безопасности. По своим стандартным функциям сервер провайдера может работать с базами данных, генерировать соответствующие страницы, преобразовывать форматы документов, но также может иметь желательные ограничения, сознательно заданные авторами, и собирать информацию, введённую посетителями сайта (данные анкет, тестов, служебная информация — названия почтовых ящиков, IP-адреса и др.).


```

alert('Проблема определена и устранена. ');
{
parent.close();
}
}
// End -->
</SCRIPT>

<BODY>

<CENTER>
<FORM>
<input type="button" value="Тайная информация или раздел - Программные Дополнения" onClick="confirmClose()" >
</FORM>
</CENTER>

```

Изображение веб-страницы или действия скрипта

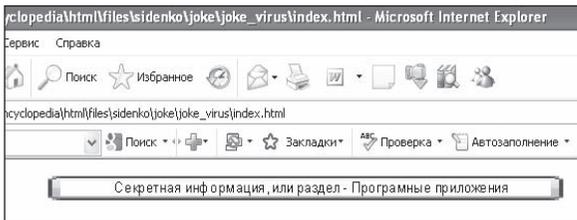


Рис. 1. Сообщение для «пробуждения» заинтересованности пользователя

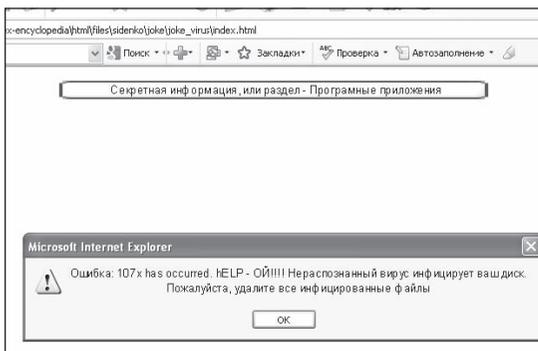


Рис. 2. Сообщение, имитирующее ошибку в работе компьютера и появление неизвестного вируса

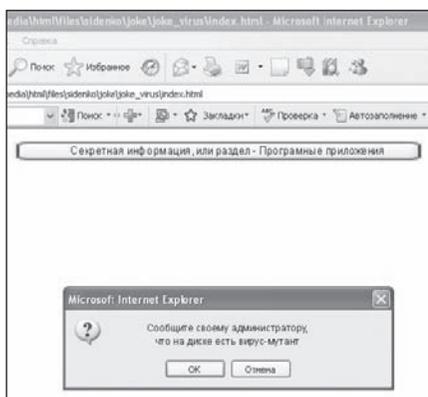


Рис. 3. Сообщение для «распространения паники» с обращением к администратору

Изображение веб-страницы или действия скрипта

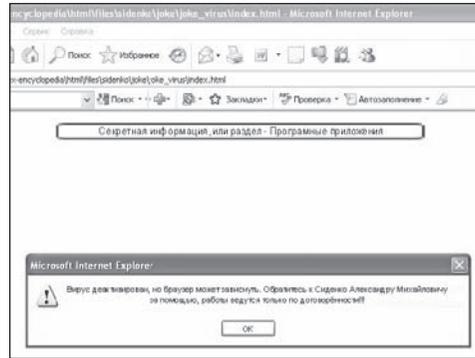
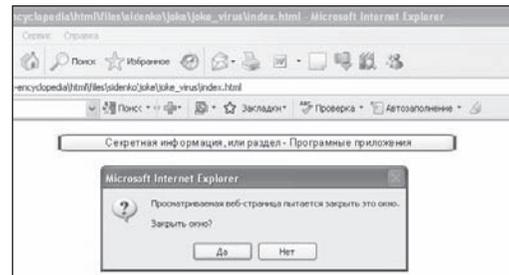


Рис. 4. Сообщение для обращений к «автору вируса» с отображением контактной информации



ИСПОЛЬЗОВАНИЕ ПОДВИЖНОГО ТЕКСТОВОГО БЛОКА, ИЛИ СТАРУЕТ БЕГУНОК

Задача 3

Создать веб-страницу, которая отображала бы подвижный блок текста на цветном фоне и имела своё направление движения:

- а) по горизонтали;
- б) по вертикали.

Вариант а) полная запись веб-страницы;

```

<HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY>
<DIV id="marqueel" style="position: absolute; overflow: hidden; left:255 px; top:1863 px; width:479 px; height:30 px; z-index:35; background-color:#FF0000">
<MARQUEE width="479" height="30">
<DIV class="wpmdd">
<DIV align=center><FONT class="ws14">Пример записи текстового "бегунка"
</FONT></DIV>
</DIV></MARQUEE>
</DIV>
</BODY>
</HTML>

```

ключевой тег в обоих вариантах для создания текстового блока — это «marquee» или «маркер» — тот, который выделяет.

Справка. Тег <MARQUEE> создаёт текстовый блок на странице. В действительности содержимое контейнера <MARQUEE> не ограничивается строками и позволяет перемещать любые элементы веб-страницы: изображение, текст, таблицы, элементы форм и тому подобное.

Самый простой вариант записи создания бегунка:

```
<MARQUEE>Содержание текстового блока!!!</marquee>
```

Параметры тега

Возможные параметры	Пояснение
behavior	Задаёт тип движения содержимого контейнера <MARQUEE>
bgcolor	Цвет фона
direction	Указывает направление движения содержимого контейнера <MARQUEE>
height	Высота области прокрутки
hspace	Горизонтальные поля вокруг контента
loop	Задаёт, сколько раз демонстрируется содержимое
scrollamount	Скорость движения контента
scrolldelay	Величина задержки между движениями в миллисекундах
truespeed	Отменяет встроенный ограничитель скорости при низких значениях параметра scrolldelay
vspace	Вертикальные поля вокруг содержимого
width	Ширина области прокрутки

Упрощённый вариант записи с учётом направления движения текста:

```
<marquee direction="down | left | right | up">...</marquee>
```

Таблица аргументов тега marquee

Аргумент	Комментарий
down	Движение сверху вниз
left	Движение справа налево
right	Движение вправо
up	Движение вверх

Перемещение можно задать не только по горизонтали, но и по вертикали. В этом случае указывают размеры области, в которой будет происходить движение.

Обратите внимание на позицию бегунка (текстового блока) на странице: числовые данные можно изменять! Во фрагменте указывают не только координаты размещения блока, но и размеры рамки бегунка.

```
<DIV id="marquee1" style="position: absolute; overflow: hidden; left:255 px; top:1863 px; width:479 px; height:30 px; z-index:35; background-color:#FF0000">
<MARQUEE width="479" height="30">
```

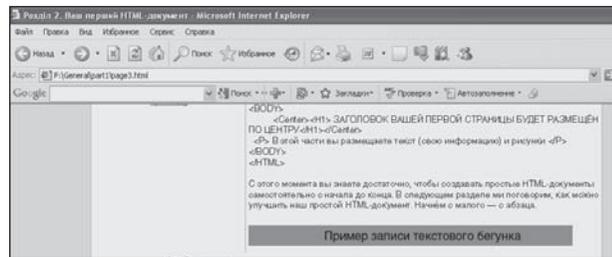


Рис. 5. Общий вид бегунка в горизонтальном направлении

Вариант б) полная запись веб-страницы;

```
<HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY>
<DIV sd="marquee2" style="position: absolute; overflow: hidden; left:340 px; top:2463 px; width:165 px; height:118 px;z-index:38; border:#808080 3px solid; background-color:#FF0000">
<MARQUEE width="165" height="118" direction="Up">
<DIV class="wpmд">
<div align=center><FONT class="ws14"><B>Пример записи текстового бегунка в вертикальном направлении </B></FONT></DIV>
</DIV></MARQUEE>
</DIV>
</BODY>
</HTML>
```

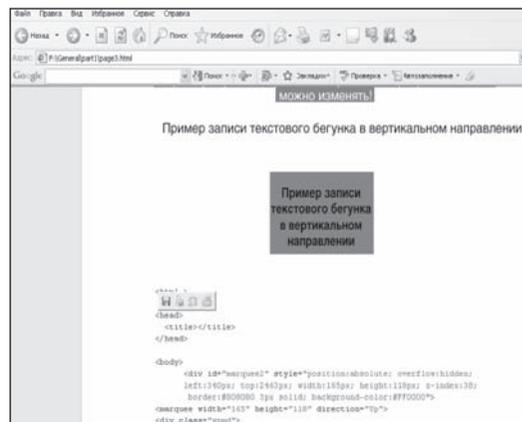


Рис. 6. Общий вид бегунка в вертикальном направлении

Задача 4

Постановка задачи. Создать веб-страницу, которая имела бы полную защиту html-кода и нормальный просмотр в браузере, а также предоставляла возможность корректировать её или просматривать в обычном виде (HTML-кода).

Решение. Используем программу автоматического кодирования содержимого HTML-кода, дающую возможность «спрятать» сам html-код, но оставить полную функциональность веб-страницы. Название программы — sencrypt (Source Code Encrypter 1.0), её размер не превышает 4 Кб.

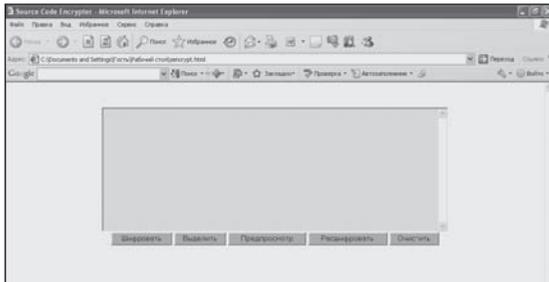


Рис.7. Общий вид программы sencrypt (Source Code Encrypter 1.0), которая запускается в оболочке Internet Explorer

Рабочая оболочка программы Source Code Encrypter 1.0 имеет пять функциональных клавиш:

- кодирование;
- выделение;
- просмотр;
- дешифровка;
- удаление.

Все эти клавиши используют для работы с кодовой частью веб-страниц. Приведём примеры использования программы:

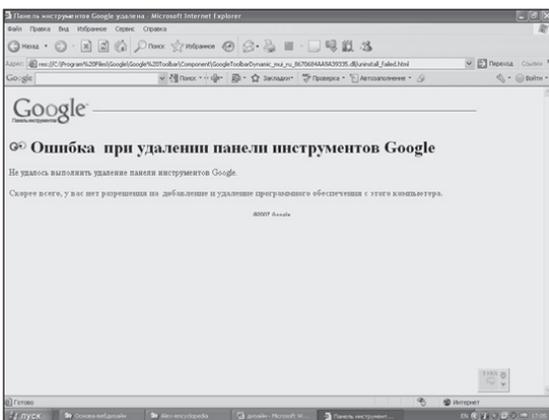


Рис. 8. Вид страницы после кодирования

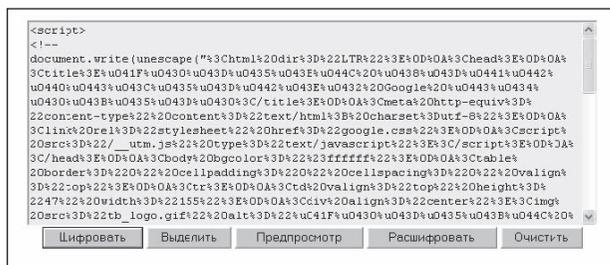


Рис. 9. Вид кодовой части в HTML

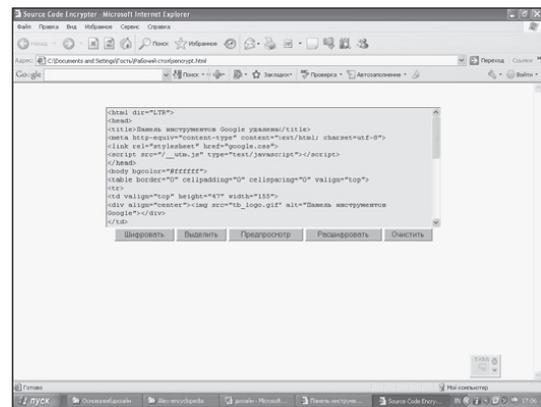


Рис.10. Вид страницы после кодирования

Полная функциональность веб-страницы остаётся прежней, но просмотр HTML-кода будет невозможен без использования программы-ключа. В таком случае заимствовать авторский дизайн страницы или сайта очень сложно человеку, который собирается использовать его для себя.

Поэтому совокупность гипертекстов, объединённых общим информационным наполнением, оформлением, а также авторством, называется веб-страницей. Физически это не обязательно один гипертекст, это любое количество гипертекстов, содержащих мультимедиа информацию. Веб-страницу можно сравнить с обычной бумажной публикацией, в которой количество страниц и дизайн определены по желанию автора. Иногда в качестве синонима термина «веб-страница» используют термин «веб-публикация», который намного точнее отображает суть предмета.

Создание собственного веб-сайта открывает перед вами широкий спектр коммуникационных, информационных и рекламных возможностей по цене значительно ниже, чем при использовании традиционных издательских технологий, но при условии, что ваша фантазия и умение соответствуют современным требованиям.

Поэтому начинающему пользователю придётся осваивать огромное количество новых понятий, терминов, привыкать к поиску новых возможностей, которые вдруг откроются для него сегодня. И тогда потребность разместить в интернете собственную информацию всегда будет прямо пропорциональна вашим сформированным знаниям, которые будут соответствовать скорости создания или обновления сайта.

Литература

1. *Гладкий А.* Веб-Самоделкин. Как самому создать сайт быстро и профессионально. — 2012.
2. *Кудлай А.* Кодировки сайта: проблемы, вопросы, решения. — 2011.