

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 08.02.2021 16:53:45
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

1

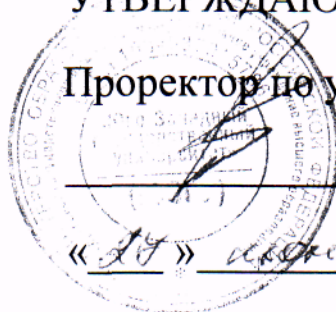
МИНОБРАЗОВАНИЯ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ

Проректор по учебной работе

О. Г. Локтионова



2017 г.

СОЗДАНИЕ САЙТОВ НА ЯЗЫКАХ HTML, JAVASCRIPT И
ОБЕСПЕЧЕНИЕ ИХ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Методические указания по выполнению
лабораторных работ по дисциплине
«Защита информационных процессов в компьютерных
системах» для студентов направления подготовки бакалавров
10.03.01

Курск 2017

УДК 004.56.5(076.5) + 004.432(076.5)

Составитель: К.А. Тезик

Рецензент

Кандидат технических наук, доцент *Т. И. Лапина*

Создание сайтов на языках HTML, JavaScript и обеспечение их информационной безопасности: методические указания по выполнению лабораторных работ / Юго-Зап. гос. ун-т; сост.: К. А. Тезик, Курск, 2017. 70 с.: ил. 12, табл. 1, Библиогр.: с. 70.

Содержат краткие теоретические положения о языке разметки текстов HTML, языке программирования JavaScript, а также методах защиты информации с помощью формы регистрации на сайте и Java-скриптов, предназначенных для защиты контента от несанкционированного копирования.

Методические указания соответствуют требованиям программы по направлению подготовки бакалавров: информационная безопасность.

Предназначены для студентов направления подготовки бакалавров 10.03.01 дневной формы обучения.

Текст печатается в авторской редакции

Подписано в печать *24.06* Формат 60x84 1/16.
Усл. печ. л. *4/1* . Уч. – изд. л. *3/8* . Тираж 100 экз. Заказ. *1364* Бесплатно.
Юго - Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Введение

В сознании большинства пользователей Интернет ассоциируется с тремя основными информационными технологиями:

- электронная почта (e-mail);
- файловые архивы FTP;
- World Wide Web.

Технология World Wide Web предоставляет простой интерфейс для доступа к разнообразным сетевым ресурсам. Это привлекло внимание коммерческих структур и привело к лавинообразному росту числа пользователей WWW. Одним из компонентов создания технологии создания распределенной гипертекстовой системы World Wide Web стал язык гипертекстовой разметки HTML, разработанный Тимом Бернерсом-Ли. Язык HTML позволяет размечать электронный документ, который отображается на экране с полиграфическим уровнем оформления. Итоговый документ может содержать разнообразные метки, иллюстрации, аудио- и видеофрагменты. В состав языка вошли средства для создания заголовков, шрифтовых выделений, списки, таблицы. Вторым важным моментом, обусловившим успех: в качестве основы HTML лежит обычный текстовый файл. Таким образом, гипертекстовая база данных в концепции WWW – это набор текстовых файлов, размеченных на языке HTML, который определяет форму представления информации (разметка) и структуру связей между этими файлами и другими информационными ресурсами (гипертекстовые ссылки).

Однако, только лишь применение HTML позволяет создавать только статические сайты. Статические веб-страницы - это статические файлы (набор текста, таблиц, рисунков и т.д.), которые хранятся в готовом виде в файловой системе сервера.

Динамические сайты, в которых веб-страницы генерируются или формируются (создаются динамически) в процессе исполнения запроса пользователя. Динамические сайты бывают двух типов. В

первом типе сайтов, веб-страницы генерируются или формируются из данных хранящихся на сервере в базе данных. Во втором типе сайтов веб-страницы генерируются на стороне клиентского приложения (в браузере).

Для управления клиентскими сценариями просмотра страниц Web-сайта можно использовать языки программирования этих сценариев, например, JavaScript. Клиентские сценарии широко применяются для решения таких задач, как проверка информации, введенной пользователем в формы, перед ее отправкой на сервер или для программирования различных ответных реакций на действия пользователя, делающих веб-страницы интерактивными.

В данном методическом пособии рассматриваются основы языка гипертекстовой разметки HTML и языка программирования JavaScript. Рассмотрены вопросы защиты информации на сайте с помощью технологий аутентификации (форм регистрации на сайте), а также Java-скрипты для защиты контента от несанкционированного копирования.

1. Язык HTML. Первая Web-страница.

Краткие теоретические положения

HTML – описательный язык разметки документов, в котором используются указатели разметки – теги. Существуют два типа тегов – контейнерные и одиночные, которые заключаются в угловые скобки <Имя_тега>. Контейнерные теги состоят из пары – открывающий и закрывающий тег. Перед именем закрывающего тега необходимо ставить косую черту “ / ” (прямой слэш).

Например, <h1> Моя первая Web – страница </h1> Здесь <h1> тег заголовка первого уровня.

Одиночные теги состоят лишь из открывающего тега. Например, тег <hr> выводит на экран горизонтальную разделительную линию.

В HTML – теги можно помещать другие теги. Например:

<h1> Моя <i> первая </i> Web- страница </h1> Текст, заключенный между тегами <i> и </i> будет выведен на экран курсивом.

Тег может иметь атрибуты и значения атрибутов. Атрибуты добавляют в тег для расширения или модификации его действий. Синтаксис:

<тег Имя_атрибута1= ”Значение” Имя атрибута 2 = “Значение”>..</тег> Например, <h1 align=”center” > Моя первая Web – страница </h1> С помощью атрибута align заголовков будет отцентрирован.

В языке HTML безразлично, в каком регистре записаны названия тегов и их атрибутов.

Практическое задание №1

Цель работы: изучить методику работы с Web-документом.

Назначение программы: программа выводит на экран сообщение “ Моя домашняя страничка. Мой первый HTML – документ ”

Текст программы:

```
<html>
```

```

<head>
<title> Издательство НТ Пресс </title>
</head>
<body>
<h1>Моя домашняя страница</h1>
<p>Мой первый HTML-документ</p>
</body>
</html>

```

2. Структура HTML – документа. Краткие теоретические положения

Любой HTML - документ (страница) заключается между тегами <html> и </html>. HTML – документ состоит из двух частей : заголовков – head и тело - body. В заголовке содержится информация о документе - название, метайнформация. В теле находится само содержимое документа – то, что выводится в окне браузера: текст, картинки, таблицы.

В заголовочной части таблицы страницы находятся теги <title>, <meta>, <style>, <script>. Тег <title> определяет название документа. Он единственный обязательный элемент заголовочной части таблицы.

Теги <meta> (одионочный тег) предоставляет поисковым машинам информацию о документе – ключевые слова и описание документа. Для перекодировки на стороне клиента в заголовок документа необходимо включить meta-тег следующего вида:

```
< meta http – equiv="content-type" content ="text/html; charset=windows-1251">
```

Для описания документа используется два мета-тега. Один определяет список ключевых слов (атрибут keywords), а второй – реферат (краткое содержание документа), который отображается в отчете поисковой машины (атрибут description).

Синтаксис:

```

<meta name="keywords" content="слово1, слово2, слово3">
<meta name="description" content="Краткое описание страницы">

```

Помещая тег `<styles>` в раздел `<head>` можно добавить встроенные таблицы стилей. Встроенная таблица стилей – раздел HTML – документа, где описывается стилевое оформление тегов: каким шрифтом, цветом оформлять заголовки, ссылки, текст в абзацах.

Тег `<script>` предназначен для реализации сценариев на языке JavaScript, данный вопрос будет рассмотрен в параграфе 10.

Практическое задание №2

Цель работы: изучить структуру документа HTML.

Назначение программы: в программе включена таблица стилей в раздел `head` для того, чтобы текст, ограниченный тегами `<h1>` и `</h1>` стал красным.

Текст программы:

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=windows-1251">
<meta name="description" content="В рамках книги основы HTML, структура заголовка и теги meta">
<meta name="keywords" content="учебный курс, Web-технология, документ, МЕТА, HEAD, разметка, методика, description, keywords">
<title>Издательство ИТ Пресс </title>
<style type="text/css">h1 { color : red } </style>
</head>
<body>
<h1>Моя домашняя страница </h1>
<p>Мой первый HTML - документ </p>
</body>
</html>
```

3. Списки в HTML – документах.

Краткие теоретические положения

В языке HTML используются маркированные и нумерованные списки. Маркированные списки вводятся тегом ``. Между открывающим `` и закрывающим `` тегами располагаются элементы списка. Каждый элемент списка вводится тегом `` (контейнерного типа). Используя атрибут `type`, можно принудительно задать тип маркера. Для всего списка `<ul type=" " ">`, для элемента списка - `<li type=" " ">`. Типы маркера (значения атрибута `type`):

`disk, circle, square.`

Нумерованные списки вводятся тегом ``. Между открывающим `` и закрывающим `` тегами располагаются элементы списка. Каждый элемент списка вводится тегом `` (контейнерного типа). Используя атрибут `type`, можно принудительно задать тип нумерации. Для всего списка `<ol type=" " ">`, для элемента списка - `<li type=" " ">`. Типы нумерации:

- 1) `type=1` – арабские цифры (по умолчанию);
- 2) `type=a` – строчные буквы;
- 3) `type=A` – заглавные буквы;
- 4) `type=i` – римские цифры строчными буквами;
- 5) `type=I` – римские цифры заглавными буквами.

Практическое задание №3

Цель работы: изучить методику применения нумерованных списков в языке HTML

Назначение программы: программа выводит на экран нумерованный список

Текст программы:

```
<html>
<head>
<title>Издательство НТ Пресс</title>
<style>
p {color:red}
</style>
</head>
```



```

<body>
<br>
<ol>
<lh ><font size="6"><font color=purple>Список
студентов</font></font></lh>
<li>Петров</li>
<li>Иванов</li>
<li>Ивановский</li>
</ol>
<br><br>
<hr color="orange">
А теперь сформируем список с помощью букв
</body>
<br>
<ol type="a">
<lh ><u>Список студентов</u></lh>
<li>Петров</li>
<li>Иванов</li>
<li>Ивановский</li>
</ol>
</html>

```

Практическое задание №4

Цель работы: изучить методику применения маркированных списков в языке HTML

Назначение программы: программа выводит на экран маркированный список

Задание: разработать Web – страницу, которая выводит на экран маркированный список «Специальности факультета фундаментальной и прикладной информатики ЮЗГУ».

4. Форматирование текста и управление шрифтом. Краткие теоретические положения

Выравнивание текста в абзаце осуществляется с помощью атрибута align тега <p>. Атрибут align может иметь следующие значения:

- Align = "left" – выравнивание по левому краю;
- Align="right" – выравнивание по правому краю;
- Align="center" – выравнивание по центру;
- Align = "justify" – выравнивание по ширине.

Теги и позволяют использовать полужирный шрифт. Теги <i> и </i> позволяют отображать курсив. Для подчеркивания текста необходимо использовать теги <u> и </u>. Для перечеркивания текста необходимо использовать теги <s> и </s>. Для выведения текста в виде нижнего индекса его следует пометить между тегами _и, а в виде верхнего – между ^и.

Используя теги и можно изменить тип шрифта, его размер и цвет. Воспользуйтесь атрибутом face=" " , и браузер попытается отобразить текст указанным вами шрифтом.

Шрифт Arial

Если не удастся найти шрифт Arial, браузер выводит текст иным шрифтом по умолчанию, как если бы тега и вовсе не было.

Размер шрифта меняется с помощью атрибута size тега . Например: Размер 1 . Размер 1 означает самый малый размер шрифта, размер 7 самый большой размер шрифта.

Использование атрибута color тега дает возможность задавать один из нескольких цветов для отображения текста. При этом применяются два вида цветовых переменных: HEX-коды и названия цветов.

Например: Красный

 Желтый

Для отображения бегущей строки используется контейнерный тег <marquee> ...</marquee> Пример:

```
<marquee scrollamount=5> <font size=5> Не злоупотребляйте
бегущей строкой </font></marquee>
```

Здесь scrollamount=5 – скорость перемещения текста.

Практическое задание №5

Цель работы: изучить методику применения шрифтов в языке HTML

Назначение программы: программа выводит на экран 4 строки с названиями шрифтов, причем данные строки выведены соответствующими шрифтами.

Текст программы:

```
<html>
<head>
<title>Издательство НТ Пресс </title>
</head>
<body>
<h3 align=center> изменение типа шрифта </h3>
<font face="Arial"> шрифт Arial </font> <br>
<font face="Courirer New"> Шрифт Courier New </font> <br>
<font face="Times"> Шрифт Times</font> <br>
<font face="Ms Sans Serif"> Шрифт Ms Sans Serif </font>
</body>
</html>
```

Дополнительное задание: внесите изменения в программу так, чтобы строка “изменение типа шрифта” была синего цвета.

5. Использование графики в HTML

Краткие теоретические положения

Тегом HTML, который заставляет браузер выводить изображение, является тег с обязательным атрибутом src. Имя файла представляет собой имя выводимого графического файла. Пример вставки изображения:

```

```

Здесь `image.gif` – имя файла изображения и его URL, то есть точный адрес файла на Web-сайте;

`Alt="изображение"` – альтернативная надпись, то есть надпись, которая будет появляться при наведении курсора на изображение.

Тег изображения может иметь необязательные атрибуты `align`, `border`, `width`, `height`.

Атрибут `align` определяет положение относительно окружающего его текста.

`Align="left"`, изображение располагается вдоль левой границы документа, а строки текста огибают его справа.

`Align="right"`, изображение располагается вдоль правой границы документа, а строки текста огибают его слева.

`Align="middle"`, изображение располагается посередине, выравнивает центр изображения по базовой линии строки окружающего текста.

Атрибут `border` – определяет толщину рамки вокруг изображения.

Атрибуты `width` и `height` изменяют размеры изображений, выводимых на Web-страницу.

Приведем пример оператора, который показывает применение атрибутов тега ``.

```

```

Практическое задание №6

Цель работы: изучить методику применения графики в языке HTML.

Назначение программы: программа выводит на экран изображение в левой части экрана и текст в правой части экрана. Файл с именем `roza.bmp`, в котором изображен цветок розы, должен быть предварительно создан в графическом редакторе Paint и этот файл должен быть скопирован в ту же папку, что и файл Web-страницы.

Текст программы:

```

<html>
<head>
<title>Издательство НТ Пресс</title>
</head>
<body>
<h2 align=center> Действие атрибутов графики</h2>

<p>Изображение располагается вдоль левой границы доку-
мента, а последующие строки текста огибают его справа. На
рисунке изображен цветок роза </p>
</body>
</html>

```

6. Использование таблиц в HTML**Краткие теоретические положения**

Для описания таблиц используется тег <table>. Тег <tr > создает строку таблицы. Другие теги, которые требуется поместить в 1 строку, например теги ячеек должны размещаться между тегами <tr> ...</tr>. Каждая ячейка, содержащая текст или изображение, должна быть окружена тегами <td> ...</td>. Заголовки для столбцов и строк таблицы задаются с помощью тега <th>...</th>. Данные теги подобны тегам <td> ...</td>. Отличие состоит в том, что текст заключенный между тегами <th>...</th>, автоматически записывается жирным шрифтом и по умолчанию располагается по середине ячейки. Тег <caption> позволяет создавать заголовки таблицы. По умолчанию заголовки центрируются и размещаются либо над (<caption align=top>), либо под таблицей <caption align=bottom>.

Рассмотрим атрибуты табличных тегов.

Теги </td> и </th> модифицируются с помощью атрибута colspan. При необходимости сделать ячейку шире, чем верхняя или нижняя можно воспользоваться атрибутом colspan, чтобы растянуть ее над любым количеством обычных ячеек. Атрибут rowspan,

подобен атрибуту colspan, только он задает число строк, на которые растягивается ячейка. Атрибут width можно поместить в тег <table> для задания ширины таблицы в пикселях (например width=250) или процентном отношении от ширины страницы (например width=50%). Атрибут align в тегах <tr>, <td>, </th> определяет выравнивание текста в ячейке по левому краю (align= left), по центру (align=center), или по правому краю (align=right). Атрибут border в теге <table> определяет толщину линий рамки таблицы. Атрибут bgcolor позволяет установить цвет фона ячейки, строки или таблицы, применяется совместно с тегами table, tr, td. Например: <td bgcolor=red> текст или данные </td>.

Практическое задание №7

Цель работы: изучить методику использования таблиц в языке HTML.

Назначение программы: программа выводит на экран следующую таблицу:

Таблица

Если в таблице два тега tr, то в ней две строки		
Если в строке три тега td	то в ней	три столбца

Рис 1. Таблица, выводимая на Web-страницу

Текст программы:

```
<html>
<head>
<title> издательство НТ Пресс </title>
</head>
<body>
<h2 align=center> Таблица </h2>
<center>
<table border=4>
<tr>
```

```

<td colspan=3> Если в таблице два тега tr,то в ней две
троки.</td>
</tr>
<tr>
<td> Если в строке три тега td,</td>
<td> то в ней </td>
<td> три столбца </td>
</tr>
</table>
</center>
</body>
</html>

```

7. Гиперссылки в HTML. Пример разработки элементарного сайта.

Краткие теоретические положения

Гиперссылки в языке HTML состоят из двух частей. Это видимый текст ссылки и невидимый URL той страницы, на которую данный текст ссылается. Для создания гиперссылки используется контейнерный тег <a>. В наиболее простом случае синтаксис гиперссылки выглядит следующим образом:

```
<a href=URL страницы> Текст ссылки </a>
```

Приведем пример ссылки на HTML страницу, расположенную на другом Web-узле (ссылка на сайт издательства «НТ Пресс»).

```
<a href=http://www.ntpress.ru target="blank" title="Переход на сайт издательства НТ Пресс"> Издательство "НТ Пресс" </a>
```

Обязательный атрибут href указывает адрес той страницы, на которую необходимо перейти. Атрибут target со значением «blank» указывает браузеру, что новую страницу следует открывать в новом окне. Атрибут title выводит подсказку, которая позволит посетителю Web-сайта оценить необходимость перехода по этой ссылке.

Адрес URL часто не содержит имени файла. В таком случае в корневом каталоге сайта происходит поиск файла с именем

index.htm. Этот файл называется главной, стартовой или домашней страницей сайта.

В качестве средства ссылок на другие Web-страницы, можно использовать не только текст, но и изображения, например:

```
<a href=http://www.ntpress.ru></a>
```

Практическое задание №8

Цель работы: изучить методику использования ссылок и разработки элементарного сайта в языке HTML.

Назначение программы: программа реализует сайт, состоящий из трех страниц: index.htm, first.htm, second.htm. Все три страницы расположены в одной папке. Между этими страницами организованы переходы по гиперссылкам. Главная страница сайта представлена на рис. 2. Остальные две страницы сайта имеют аналогичную структуру.

Это мой первый сайт

Главная страница	
<p><u>Главная</u> <u>Первая</u> <u>Вторая</u></p>	<p>Здравствуйте, это главная страница моего сайта. Он создан мною в ходе изучения основ HTML и содержит еще 2 страницы. Надеюсь, Вам понравится мой сайт</p>

Рис 2. Главная страница сайта

Текст программы:

Страница index.htm


```

<html>
<head>
<title>Мой первый сайт</title>
</head>
<body bgcolor="#DCDCDC">
<center>
<h1 align=center>Это мой первый сайт</h1>
<table border width=80% bgcolor="CCFFFF">
<tr>
<td colspan=2><h2 align=center>Главная страница </h2>
</td>
</tr>
<tr>
<td width=35%>
<a href="index.htm"> Главная страница</a><br>
<a href="first.htm"> Первая страница</a><br>
<a href="second.htm"> Вторая страница</a><br>
</td>
<td>
Здравствуйете, это главная страница моего сайта. Он создан
мною в ходе изучения основ HTML и содержит еще 2 страни-
цы. Надеюсь, Вам понравится мой сайт
</td>
</tr>
</table>
<br><br>
</center>
</body>
</html>

```

Страница first.htm

```

<html>
<head>
<title>Мой первый сайт</title>
</head>

```

```

<body bgcolor="#DCDCDC">
<center>
<h1 align=center>Это мой первый сайт</h1>
<table border width=80% bgcolor="CCFFFF">
<tr>
<td colspan=2><h2 align=center><font color="red">Первая
страница </font></h2>
</td>
</tr>
<tr>
<td width=35%>
<a href="index.htm"> Главная страница</a><br>
<a href="first.htm"> Первая страница</a><br>
<a href="second.htm"> Вторая страница</a><br>
</td>
<td>
Это первая страница моего сайта. Отсюда Вы можете перейти
на главную страницу и на вторую страницу
</td>
</tr>
</table>
</center>
</body>
</html>

```

Страница second.htm

```

<html>
<head>
<title>Мой первый сайт</title>
</head>
<body bgcolor="#DCDCDC">
<center>
<h1 align=center>Это мой первый сайт</h1>
<table border width=80% bgcolor="CCFFFF">
<tr>

```

```

<td colspan=2><h2 align=center><font color="gold">Вторая
страница </font></h2>
</td>
</tr>
<tr>
<td width=35%>
<a href="index.htm"> Главная страница</a><br>
<a href="first.htm"> Первая страница</a><br>
<a href="second.htm"> Вторая страница</a><br>
</td>
<td>
Это вторая страница моего сайта. Отсюда Вы можете перейти
на главную страницу и на первую страницу
</td>
</tr>
</table>
</center>
</body>
</html>

```

Практическое задание №9

Цель работы: изучить методику использования ссылок в виде изображения.

Назначение программы: самостоятельно разработайте Web-страницу, которая реализует переход на сайт издательства НТ пресс (адрес сайта <http://www.ntpress.ru>) по ссылке, реализованной в виде изображения.

8. Фреймы в языке HTML.

Краткие теоретические положения

Фреймы(кадры) – это возможность разбиения страницы на независимые окна, каждое из которых может отражать собственную информацию. Прием широко используется при построении

Web-сайтов. Например, левая часть страницы содержит некое оглавление, а оставшееся пространство предназначено для отражения документов при выборе раздела оглавления. Структура фреймов описывается между тегами `<frameset>...<frameset>`. Здесь размещается информация о числе фреймов, их размерах и ориентации (горизонтальной или вертикальной). Атрибут `rows` тега `<frameset>` задает число и размер строк на странице.

Следующий тег `<frameset>` создает экран, на котором верхняя строка занимает 10% высоты экрана, средняя – 60%, а нижняя – 30%:

```
<frameset rows="10%, 60%, 30%">
```

Следующий тег создает экран, на котором верхняя строка имеет высоту 20 пикселей, средняя – 80 пикселей, а нижняя занимает оставшееся место:

```
<frameset rows="20, 80, * ">
```

Атрибут `cols` тега `<frameset>` задает число и размер столбцов на странице. Например, следующий тег задает две колонки шириной 35% и 65% экрана:

```
<frameset cols="35%, 65%">
```

Для каждой строки и столбца, упомянутых в теге `<frameset>`, необходим свой набор тегов `<frame>`. Тег `<frame>` определяет внешний вид и поведение фрейма. Основными атрибутами тега `<frame>` являются атрибуты `name` и `src`. Атрибут `name` служит для создания правильной системы навигации (чтобы при щелчке мышью по ссылке соответствующая страница отображалась в определенном фрейме, необходимо указать этот фрейм). Атрибут `src` применяется для того, чтобы определить, какая страница появится в кадре при первой загрузке страницы фреймов.

Практическое задание №10

Цель работы: изучить методику использования фреймов.

Назначение программы: Программа делит страницу на два фрейма (колонки). В левом фрейме меню с двумя ссылками: «страница 1», «страница 2». Правый фрейм имеет имя «окно». В нем при выборе гиперссылки отображается определенная информация

(см. рис 3). Программа состоит из четырех Web-страниц: index.htm, menu.htm, page_1.htm, page_2.htm. Все 4 Web-страницы расположены в одной папке.

<p><u>Страница 1</u></p> <p><u>Страница 2</u></p>	<p>Это вторая страница сайта</p> <p>Фреймы – это возможность разбиения Web- страницы на независимые окна.</p>
---	--

Рис. 3 Переход ко второй странице сайта

Текст программы:

Страница index.htm

```
<html>
<head>
<title>Издательство НТ Пресс</title>
</head>
<frameset cols="35%, 65%">
<frame src="menu.htm">
<frame src="page_1.htm" name="окно">
</frameset>
</html>
```

Страница menu.htm

```
<html>
<head>
<title>Издательство НТ Пресс</title>
</head>
<body>
<a href=page_1.htm target="окно">Страница 1 </a><br>
```

```
<a href=page_2.htm target="окно">Страница 2 </a>
</body>
</html>
```

Страница page_1.htm

```
<html>
<head>
<title>Издательство НТ Пресс</title>
</head>
<body>
<h1 align=center>Это первая страница сайта </h1>
<p>Сайт создан при помощи фреймов</p>
</body>
</html>
```

Страница page_2.htm

Программный код данной Web-страницы напишите самостоятельно в соответствии с рис. 3.

9. Каскадные таблицы стилей в языке HTML. Краткие теоретические положения

Параметры внешнего вида документа можно задавать с помощью каскадных таблиц стилей (Cascading Style Sheets, CSS). Таблицу стилей, подобно шаблону форматирования текстов, можно разработать отдельно от конкретного документа, а затем применить к нему. CSS содержит наборы стилевых параметров или правила форматирования. Каждое правило форматирования записывают в соответствии с синтаксисом:

```
Список_селекторов {имя_параметра:значение;
имя_параметра: значение; ... }
```

Селекторы предназначены для указания элементов документа, к которым данное правило применяется. Силевые параметры за-

ключают в фигурные скобки и разделяют точкой с запятой; имя параметра и его значение разделяют двоеточием.

Например, если требуется определить для всех заголовков первого и второго уровня шрифт Courier 20 пунктов красного цвета, то соответствующее правило можно записать так:

```
h1, h2 {font-family: Courier; font-size: 20pt; color: red}
```

Правила CSS можно записать непосредственно в HTML документе или сохранить в отдельном файле, чтобы применить их одновременно к нескольким документам.

Наиболее часто для задания стилей в документе используется элемент STYLE, внутри которого задается стиль. Этот элемент содержит единственный атрибут type, который для каскадных таблиц стилей должен иметь значение text/css. Пример:

```
<style type =”text/css”>
p {color:blue; font-size:12pt}
</style>
```

Для максимальной гибкости рекомендуется создавать внешние таблицы стилей, которые могут быть модифицированы без изменения исходного HTML-документа. Для соединения с внешней таблицей стиля необходимо использовать элемент LINK. Он служит для связывания с документом внешних таблиц стилей и имеет три атрибута:

rel – задает тип соединения и всегда равен stylesheet;
 type – задает тип ссылки и всегда равен text/css;
 href – URL – адрес файла с внешней таблицей стилей.

Пример:

```
<head>
<title>Пример подключения внешней таблицы стилей </title>
<link rel=”stylesheet” type=”text/css”
href=http://www.mysite.ru/mystylesheet.css>
</head>
```

Практическое задание №11

Цель работы: изучить методику применения каскадных таблиц стилей для построения таблицы.

Назначение программы: создание таблицы с одинарной рамкой.

Текст программы:

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; char-
set=windows-1251">
<title> Издательство НТ Пресс </title>
<style>
table
{
border-collapse: collapse;
background: gold;
border: 4px solid black;
width:80% }
td, th
{
padding: 5px;
border: 2px solid green
}
</style>
</head>
<body>
<center>
<h3> Создание одинарной рамки </h3>
<table cellpadding="4" cellspacing="1" width=100%>
<tr>
<th> Заголовок 1 </th>
<th> заголовок 2 </th>
</tr>
<tr>
<td> данные </td>
<td> данные </td>
</tr>
</table>
</center>
```



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

Практическое задание №12

Цель работы: изучить методику применения каскадных таблиц стилей для работы со ссылками.

Назначение программы: при наведении курсора ссылка становится красной и подчеркнутой.

```
<html>
<head>
<title> Издательство НТ Пресс </title>
<style>
a:link {text-decoration: none}
a:visited {text-decoration: none}
a:active {text-decoration: none}
a:hover
{
text-decoration: underline;
color: red
}
</style>
</head>
<body>
<h2 align="center"> При наведении курсора на ссылку, она
становится подчеркнутой и красной </h2>
<p><a href="http://www.ntpress.ru"> Ссылка 1 </a></p>

<p><a href="http://www.ntpress.ru"> Ссылка 2 </a></p>
</body>
</html>
```

Дополнительное задание

Решите данную задачу с помощью внешней таблицы стилей. Файл внешней таблицы стилей должен иметь расширение CSS.

Практическое задание №13

Цель работы: получить навыки самостоятельной работы применения каскадных таблиц стилей.

Назначение программы: программа создает таблицу, в которой отсутствует внешняя рамка. Правила, которые определяют отсутствие внешней границы необходимо определить с помощью каскадной таблицы стилей.

	Март	Апрель	Май
Давление	747	742	745
Влажность (%)	64	81	79
Температура	5	12	17

Рис. 4 Сетка внутри таблицы

10. Самостоятельная работа по языку HTML.

Практическое задание №14

Цель работы: разработать Web-сайт на языке HTML.

Назначение программы: Программа реализует Web-сайт рекламы санатория (пансионата, базы отдыха, гостиницы). Web-сайт должен состоять из трех страниц. Главная страница сайта содержит несколько фреймов. В левой части первой страницы располагаются фотографии, в правой текст рекламы санатория (пансионата, базы данных, гостиницы). Либо наоборот, можно текст рекламы разместить слева, а фотографии справа. На первой странице должна быть гиперссылки «С ценами на номера можно ознакомиться здесь», «Контакты». По ним осуществляются переходы на

вторую и третью страницу. На второй странице приведена таблица с ценами на различные типы номеров в различное время года. На третьей странице находятся телефоны директора, бухгалтерии, менеджера и адрес электронной почты организации.

Примерно сайт может выглядеть следующим образом:

Главная (первая) страница сайта имеет вид



	<p>Предлагаем вам отдохнуть в Крыму на базе отдыха Магарач в поселке Песчаное Бахчисарайского района. Доехать до поселка Песчаное можно за 1 час от Симферополя на автобусе. Наша база отдыха находится в сосновом лесу. В поселке Песчаное имеются как песчаные, так и галечные пляжи. Недалеко от поселка Песчаное находятся следующие достопримечательности:</p> <ol style="list-style-type: none"> 1) город Севастополь 2) город Бахчисарай 3) большой каньон Крыма
	

Рис. 5 Главная страница сайта

Вторая страница сайта имеет вид

На нашей базе отдыха имеются деревянные домики и каменный корпус. В домиках имеются одноместные, двухместные, трехместные номера. Удобства общие на домик, находятся в коридоре. В корпусе имеются двухместные и трехместные номера, удобства в номере. Питание в стоимость путевки не входит и составляет 500 рублей в день на человека. Цены на проживание за сутки с человека смотрите в таблице:

Цены на проживание на базе отдыха Магарач

Месяц	МАЙ	ИЮНЬ	ИЮЛЬ	АВГУСТ	СЕНТЯБРЬ
Тип номера					
Домики	500	550	600	600	500
Корпус	700	750	800	800	700

Рис. 6 Вторая страница сайта

Третья страница сайта имеет вид

Адрес базы отдыха Магарач: Бахчисарайский район, поселок Песчаное, ул. Набережная 15.

Телефон директора: (06554) -9-19-56

Телефон менеджера: (06554) -9-23-78

Телефон бухгалтера: (06554) – 9-34-12

Адрес электронной почты: magarach25@jandex.ru

11. Методы ввода и вывода информации в языке JavaScript

Краткие теоретические положения

Для ввода и вывода информации в языке JavaScript используются методы `alert()`, `prompt()`, `confirm()`. Эти методы генерируют различные окна сообщений. Метод `alert()` отображает окно предупреждения с соответствующим сообщением. Окно содержит кнопку `Ок`. Аргументом данного метода является строка. Метод `prompt()` служит для получения данных от пользователя. При его вызове отображается окно приглашения с текстовым полем. Метод может

содержать два аргумента. Оба эти аргумента должны быть строками. Первый аргумент – сообщение, которое отображается в окне. Второй аргумент – это текст по умолчанию, который должен появиться в соответствующем поле. Также этот метод возвращает значение, которое также является строкой – это текст, который ввел пользователь. Метод `confirm()` отображает окно подтверждения, которое сходно с окном предупреждения, генерируемое методом `alert()`, но содержит 2 кнопки: Ok и Cancel.

Также для вывода информации в языке JavaScript используются метод `document.write ()`, который позволяет дописывать строку текста прямо в страницу.

Практическое задание №15

Цель работы: изучить методику ввода и вывода информации в языке JavaScript.

Назначение программы: в данной программе выдается окно с запросом “Здравствуйте, как Вас зовут?”. Пользователь вводит имя, например, Сергей. Программа выдает окно с сообщением: “Рад Вас видеть, Сергей. Спасибо Вам, что зашли на мой сайт”.

Текст программы:

```
<html>
<head>
<title>сценарий 1</title>
<script>
var nameUser;
nameUser=prompt("Здравствуйте, как вас зовут?","аноним");
alert("Рад Вас видеть"+nameUser+"Спасибо Вам, что зашли на мой
сайт");
</script>
</ head>
<body>
</ body>
</ html>
```

Практическое задание №16

Цель работы: изучить методику применения метода `document.write ()`

Назначение программы: программа 10 раз выводит сообщение “Привет, мир!!!”

Текст программы:

```
<html>
<head>
<title> Сценарий вывода информации </title>
</head>
<body>
<script type="text/javascript">
for (var i=0; i<10; i++)
{
document.write("Привет, мир!!!");
}
</script>
</body>
</html>
```

12. Переменные и массивы в языке JavaScript

Краткие теоретические положения

Рассмотрим примеры объявления и инициализации переменных в языке JavaScript:

```
var x,y,z;
var s=1.34e-5, msg="Строка", Flag=false;
```

Пока переменная не получит значение, она не определена – `undefined`. Чтобы определить тип переменной можно использовать операцию `typeof`, например, `typeof(msg)`. Она возвращает возвращает строку с типом переменной, в данном случае - `string`. Следующий оператор выводит сообщение о значении переменной и типе переменной:

```
Document.write(“Переменная msg=”+msg+” - ”+typeof(msg)
+”\n”);
```

Сообщение будет иметь вид:

Переменная msg=Строка - string

В языке JavaScript для использования нескольких данных одного типа используются массивы. Нумерация элементов массива начинается с нуля. Язык JavaScript позволяет не перечислять элементы по одному, а сразу вывести все элементы массива. Для этого необходимо обратиться к самому массиву, в результате чего получится строка, в которой все элементы массива перечислены через запятую. Узнать длину массива можно с помощью свойства length. Например, следующая строка выведет размер массива stars:

```
alert(stars.length)
```

Пример объявления массива, задания значений элементов массива и вывода элементов массива рассмотрены в практическом занятии № 17.

Практическое задание №17

Цель работы: изучить методику обработки массивов в языке JavaScript

Назначение программы: в программе реализован массив из 5 элементов (звезды), показан вывод первого элемента массива и всех элементов массива.

Текст программы:

```
<html>
<title>Инициализация массива</title>
<script>
//Объявляем массив и инициализируем значения элементов
var stars=new Array('Сириус','Канопус', 'Арктур', 'Капелла',
'Вега');
//Вывод первого элемента массива
```

```
alert(stars[0]);  
//Вывод всех элементов массива  
alert(stars);  
</script>  
</head>  
<body>  
</body>  
</html>
```

Практическое задание №18

Цель работы: разработать программу обработки массива в языке JavaScript

Назначение программы: программа должна находить сумму элементов массива, длину массива и среднее арифметическое значение элементов массива. Программа должна выводить результаты с помощью метода `document.write()` или метода `alert()`.

13. Основные операторы в языке программирования JavaScript

Краткие теоретические положения

Оператор `if..else` управляет механизмом ветвления, позволяет выбирать одно из альтернативных действий в зависимости от значения условия. Синтаксис данного оператора имеет вид:

```
if (логическое выражение)  
{  
операторы 1  
}  
else  
{  
операторы 2  
}
```

Сначала вычисляется логическое выражение, затем если оно равно `true`, выполняются операторы 1, если оно равно `false`, то выполняются операторы 2. Альтернативная ветвь `else` может отсут-

ствовать. В этом случае, если логическое выражение равно false, то управление передается следующему оператору и продолжается выполнение сценария.

Оператор switch позволяет сравнить значение с множеством других. Синтаксис данного оператора имеет вид:

```
switch (выражение)
{
  case значение 1:
    операторы
  case значение 2:
    операторы
  .....
  default:
    операторы
}
```

В данном операторе определяется значение выражения. Значение, возвращаемое выражением, сравнивается со значениями, указанными после ключевых слов case. Если имеет место совпадение, выполняется соответствующий блок операторов. Если не найдено ни одно совпадение, то выполняются операторы после инструкции default.

Оператор for служит для создания цикла. Он имеет следующий синтаксис:

for (выражение инициализации; выражение условия; выражение цикла)

```
{
  операторы
}
```

Выражение инициализации служит для задания начального значения цикла. Выражение условия позволяет прекратить цикл, когда условие перестанет выполняться, то есть примет значение false. Выражение цикла осуществляет инкремент или декремент счетчика цикла.

Оператор `while` сходен с оператором `for`, но он не производит инициализацию и инкремент счетчика в своем объявлении. Синтаксис этого оператора следующий:

```
while (выражение условия)
{
    Операторы
}
```

Оператор `do..while` практически идентичен оператору `while`, но, поскольку в нем проверка условия осуществляется в конце, он гарантирует выполнение операторов по крайней мере один раз:

```
do
{
    Операторы
} while (выражение условие)
```

Практическое задание №19

Цель работы: изучить методику применения оператора `switch`

Назначение программы: программа осуществляет перевод названия животных с английского языка на русский.

Текст программы:

```
<html>
<head>
<title>переводчик</title>
<script>
var trans;
var beast=prompt ("Введите название животного","dog");
switch (beast)
{
    case "dog":
trans="собака";
```

```

break;
case "cat":
trans="кошка";
break;
case "cow":
trans="корова";
break;
default:
trans="неизвестное животное"
}
alert(beast+"---"+trans);
</script>
</ head>
<body>
</ body>
</ html>

```

Практическое задание №20

Цель работы: изучить методику применения оператора while.

Назначение программы: программа осуществляет расчет факториала числа.

Текст программы:

```

<html>
<head>
<title> факториал с помощью оператора while </title>
<script>
var f=1;
var x=prompt("введите число", "5");
x=+x;
var i=1;
while(i<=x)
{
f=f*i;
i++;

```

```

}
alert(f);
</script>
</head>
<body>
</body>
</html>

```

Практическое задание №21

Цель работы: изучить методику применения оператора `if..else`, программу разработайте самостоятельно.

Назначение программы: программа рассчитывает значение функции:

$$F(x) = \begin{cases} x+10, & \text{при } x < 0 \\ x^2 + 4, & \text{при } 0 \leq x \leq 5 \\ 5/x, & \text{при } x > 5 \end{cases}$$

14. Функции в языке программирования JavaScript Краткие теоретические положения

Функция – это набор команд, объединенных под общим именем для выполнения определенной задачи. Синтаксис функции имеет следующий вид:

```

function ИмяФункции ([аргумент1] [,..аргумент N])
{
операторы
}

```

Ключевое слово `function` объявляет функцию с именем `ИмяФункции`. Функциям могут передаваться аргументы, которые перечисляются через запятую в скобках рядом с именем функции. В качестве аргументов могут выступать переменные, значения и выражения. Функции можно объявить в любом месте внутри эле-

мента SCRIPT. Тем не менее, рекомендуется объявить функции в блоке HEAD до начала выполнения основного кода, иначе функция может быть недоступной в момент ее вызова.

Функции могут возвращать значение. Для этого необходимо использовать оператор возврата return. Функции можно использовать в выражениях. Функция может объявляться в выражении. Обычно в этом случае функция присваивается переменной, которая затем может использоваться в выражении:

```
var cube =function(x) {return x*x*x }
alert(cube(2)+cube(3));
```

В данном случае объявляется переменная-функция, вычисляющая куб числа.

Практическое задание №22

Цель работы: изучить методику использования функций

Назначение программы: программа рассчитывает в зависимости от выбранного режима сумму или произведение элементов массива. Для расчета суммы и произведения использованы две функции sum() и product().

Текст программы:

```
<html>
<head>
<title>Пример использования функций</title>
<script>
//Глобальный массив
var a= new Array (1,2,4,8);
function sum()
{
var s=0; //локальная переменная для хранения суммы
//Цикл по всем элементам массива a
for (var i in a)
{
```

```
s=s+a[i];
}
alert("Сумма элементов массива="+s);
}
function product()
{
var p=1; //локальная переменная для хранения произведения
for (var i in a)
{
p=p*a[i];
}
alert("Произведение элементов массива="+p);
}
</script>
</head>
<body>
<script>
var x=prompt("Найти сумму или произведение (0 или 1)?","0");
if (x=="0")
{
sum();
}
if (x=="1")
{
product();
}
</script>
</body>
</html>
```

Практическое задание №23

Цель работы: изучить методику использования значения функции

Назначение программы: функция `sum(a,b)` возвращает сумму `a` и `b`, затем значение функции используется в выражении и результат выражения выводится на экран методом `alert`:

Текст программы:

```
<html>
<head><title>Использование значения функции</title>
<script>
function sum(a,b)
{
return (a+b);
}
</script>
</head>
<body>
<script>
var x1=5, x2=6, x3=3, x4=3;
//Отображаем (x1+x2)*(x3+x4)
alert (sum(x1,x2)*sum(x3,x4));
</script>
</body>
</html>
```

Практическое задание №24

Цель работы: получить навыки самостоятельной разработки программ с использованием функций на языке JavaScript

Назначение программы: программа рассчитывает число сочетаний без повторений из n элементов по m элементов по формуле: $C_n^m = \frac{n!}{m!(n-m)!}$.

При решении задачи расчет факториала надо выполнить с помощью специальной функции.

15. Обработчики событий в языке программирования JavaScript

Краткие теоретические положения

В языке JavaScript часто используются реагирования на события, которые могут быть вызваны пользователем или браузером. К событиям относятся открытие новой страницы, перемещение указателя мыши, щелчок кнопкой мыши. Каждое событие имеет соответствующий обработчик, который автоматически реагирует на возникшее событие. Например, если посетитель веб-страницы нажимает некоторую кнопку, происходит событие Click и вызывается обработчик событий onClick. Если указатель мыши пересекает гипертекстовую ссылку, происходит событие MouseOver, которому соответствует обработчик событий onMouseOver. Рассмотрим сценарии с данными обработчиками событий.

Практическое задание №25

Цель работы: изучить методику использования обработчиков событий onMouseOver и onMouseOut для динамической смены изображений-ссылок на веб-странице

Назначение программы: изначально на веб-странице отображается рисунок a.jpg. Когда посетитель наводит указатель мыши на изображение, картинка меняется на b.jpg. Это результат действия обработчика onMouseOver. Когда пользователь убирает указатель с картинки, снова появляется изображение a.jpg. Здесь действует команда onMouseOut.

Текст программы:

```
<html>
<head>
<title>JavaScript</title>
</head>
<body topmargin=70>
<center>
```



```

<h1> Демонстрация обработчика события наведения указателя
мыши </h1>
<br>
<a href="primer1.html" onMouseOver="document.pic.src='b.jpg'"
onMouseOut="document.pic.src='a.jpg'"> <img SRC="a.jpg"
name="pic" hspace=40 border=2></a>
</center>
</body>
</html>

```

Практическое задание №26

Цель работы: изучить методику использования обработчика событий onClick

Назначение программы: программа по ссылке переходит на поисковую систему Rambler. Но предварительно по обработчику события onClick выдается сообщение «Желаем удачного поиска информации»

Текст программы:

```

<html>
<head>
<title>JavaScript</title>
</head>
<body topmargin=70>
<center>
<h1> Демонстрация обработчика щелчка кнопки мыши </h1>
<br>
<a href="http://www.rambler.ru" onClick="alert('желаем удачного по-
иска информации');"> Поиск </a>
</center>
</body>
</html>

```

16. Окна в языке программирования JavaScript

Краткие теоретические положения

Объект `window` представляет окно браузера или фрейм. С помощью сценария можно создавать или удалять окна, загружать в них документы и записывать данные. Окно создается методом `open ()` объекта `window`.

`Window.open (URL-адрес, имя, опции, замена);`

URL - адрес – строка с URL – адресом документа, который следует загрузить в открываемое окно; если вы не хотите загружать в окно никакой документ, то укажите пустую строку “ ”.

Имя - строка с именем окна, которое позже может стать значением атрибута `target` в тегах `<a>` ссылок, если вы не будете ссылаться на данное окно, то укажите пустую строку “ ”. Если вам нужно, чтобы страница загружалась в маленьком окне, то данный параметр должен иметь значение `Joe`.

Опции – строка, в которой через запятую указаны дополнительные параметры оформления окна.

Замена – необязательное логическое значение, указывающие, должен ли документ по заданному URL – адресу заменить содержимое окна (применительно к уже существующим окнам).

Пример открытия окна:

```
mywindow= open(http://www.yandex.ru, “Yandex”, “width=400, height=400”)
```

Для закрытия окна служит метод `close()`. Однако выражение `window.close()` закрывает главное окно, а не то окно, которое создано методом `open()`. В этом случае необходима ссылка на созданное окно, которую следует сохранить в глобальной переменной.

Пример закрытия окна:

```
Var objwin=window.open(“mypage.htm”, “Моя страница”)
Objwin.close()
```

При создании окна методом `open()` с помощью третьего параметра можно указать в строке множество дополнительных параметров, разделенных запятыми.

Наиболее популярными являются следующие параметры:

`width, height` – ширина и высота окна в пикселях;

left, top – горизонтальная и вертикальная координаты относительно левого верхнего угла экрана в пикселях;
 menubar – разрешение/запрет отображения панели меню;
 toolbar – разрешение/запрет отображения панели инструментов;
 scrollbars – разрешение/запрет отображения полос прокрутки;
 resizable – разрешение/запрет изменения размеров окна пользователем;
 location – параметр определяет, будет ли отображаться поле ввода адреса HTML – документа;
 status – отображение строки состояния.

Логические значения для данных параметров указывают как yes и no.

Также возможен следующий вариант синтаксис команды открытия окна:

Window.open('адрес URL', 'имя окна', config='параметр 1, параметр 2, параметр n')

Параметр config показывает, что следующие атрибуты относятся к конфигурации нового окна и определяют его внешний вид. Все атрибуты параметра config перечисляются через запятую без пробелов и заключаются в одинарные кавычки.

У объекта window имеется синоним self, используемый при обращении к окну, содержащему текущий элемент.

Практическое задание №27

Цель работы: изучить методику создания и использования окон в языке JavaScript

Назначение программы: в маленькое окно загружается фотография (файл foto.jpg). По нажатию кнопки «Увеличить» создается новое окно большого размера и в него загружается данная фотография.

Текст программы: программа реализована в виде двух файлов. Имя второго файла – foto2.html.

```
<html>
```

```
<head>
```

```

<title>JavaScript</title>
<script language="JavaScript">
function openwin()
{
window.open('foto2.html','joe', config='height=300, width=400,
toolbar=no, menubar=no, scrollbars=no, resizable=no, loca-
tion=no, status=no')
self.name="main window"
}
</script>
</head>
<body>
<h3>Фотография</h3>
 <br> <br>
<form>
<button type="SUBMIT" onClick="openwin()">Увеличить
</button>
</form>
</body>
</html>

```

Файл foto2.html

```

<html>
<head>
<title> Фотография </title>
</head>
<body background="foto.jpg">
</body>
</html>

```

Практическое задание №28

Цель работы: получить навыки самостоятельной работы создания фотогалереи в языке JavaScript.

Назначение программы: на веб-странице имеются несколько фотографий маленького размера. При щелчке мышью на фотографии увеличенная фотография должна открываться или в отдельном фрейме или в отдельном окне. Добавьте кнопки перехода к следующей и предыдущей фотографии.

17. Анимация изображения в языке программирования JavaScript

Краткие теоретические положения

Для анимации изображения необходимо через равные промежутки времени изменять атрибут `src` в изображении. Для создания анимации используют одну из функций объекта `Window`: `setTimeout ()` или `setInterval ()`. Рассмотрим пример анимации изображения.

Практическое задание №29

Цель работы: изучить методику использования анимации изображений в языке JavaScript.

Назначение программы: В данном сценарии последовательно происходит анимация изображений `img5.jpg`, `img4.jpg`, `img3.jpg`, `img2.jpg`, `img1.jpg`, `img0.jpg`. Это происходит с помощью функции `animation ()`. Данная функция получает управление после щелчка кнопкой мыши на рисунке. После чего функция изменяет рисунок, уменьшает счетчик и порождает другой свой экземпляр, выполнение которого откладывается на одну секунду. Когда счетчик становится отрицательным, процесс анимации прекращается.

Текст программы:

```
<html>
<head>
<meta charset="utf-8">
<title>Анимация изображения</title>
<script>
```

```

//Создаём массив для хранения изображений
numerals=new Array(6);
//В цикле создаём объекты изображений и загружаем рисунки
for(var i=1;i<6;i++){
    //Создаём объекты изображений
    numerals[i]=new Image();
    // Загружаем все рисунки
    numerals[i].src="img"+i+".jpg";
}
//Функция, осуществляющая анимацию
function animation(count){
    //Загружаем текущий рисунок
    document.images[0].src="img"+count+".jpg";
    count--; //Уменьшаем счётчик
    //Пока есть изображения, рекурсивно вызываем функцию
    animation()
    if (count>=0){
        //Открываем вызов функции на секунду
        setTimeout("animation("+count+");",1000);
    }
}

</script>
</head>
<body>
<h2 style="text-align : center;">Анимация изображения</h2>
<p style="text-align : center;" >
 </p>
</body>
</html>

```

18. Создание меню в языке программирования JavaScript

Краткие теоретические положения

Одна из главных задач, которую необходимо решать программисту при разработке сайта, это создание удобной системы навигации (меню). Для этого можно использовать обычные гиперссылки, а можно воспользоваться средствами языка JavaScript. Рассмотрим пример создания простого горизонтального меню. Для создания такого меню использовался обычный маркированный список со ссылками внутри, а также каскадные таблицы стилей: стиль `ul li` позволяет выстроить список в линию и убрать маркер; стиль `li` а описывает внутренние отступы и внешний вид меню. Переход по ссылкам в меню осуществляется при помощи обработчика событий `onClick`.

Практическое задание №30

Цель работы: изучить методику создания меню в языке JavaScript.

Назначение программы: программа создает горизонтальное меню следующего вида:



Рис 7. Горизонтальное меню.

Текст программы:

```
<html>
<head>
<title>Простое горизонтальное меню</title>
<!--Определяем стиль -->
<style type="text/css">
ul li{list-style-type:none;display:inline}
li a{font-family:Tahoma,Helvetica,sans-serif;text-decoration:none;color:#fff}
li a{display:block;float:left;padding:4px 10px;background:193D6A}
li a:hover{background:#408BE8;padding:7px
14px;position:relative;top:-3px}
```

```

</style>
</head>
<body>
<ul>
<li><a href="#" onClick="location.href='about.html';">
О нас</a></li>
<li><a href="#" onClick="location.href='news.html';">
Новости</a></li>
<li><a href="#" onClick="location.href='serv.html';">
Услуги</a></li>
<li><a href="#" onClick="location.href='cont.html';">
Контакты</a></li>
</ul>
</body>
</html>

```

Дополнительные задания:

1. Создайте веб-страницы, на которые указывают пункты меню и убедитесь в правильной работе сайта.
2. Измените цветовую раскраску меню следующим образом: фон меню – зеленый цвет, маркер, выбирающий пункт меню – красный цвет, текст – черный цвет.

Практическое задание №31

Цель работы: получить навыки самостоятельной разработки меню в языке JavaScript.

Назначение программы: реализация вертикального раскрывающегося меню. В исходном состоянии меню выглядит подобно кнопке (прямоугольник с надписью). При наведении указателя мыши на данный прямоугольник меню раскрывается, показывая опции (пункты) своего подменю. При уходе указателя мыши с прямоугольника раскрывшегося меню последнее сворачивается и принимает исходное состояние. Внешний вид меню представлен на рис. 8. При щелчке мышью на пункте меню загружается соответствующая поисковая система.

Поисковики
Rambler
Яндекс
Google

Рис 8. Вертикальное раскрывающееся меню.

19. Защита информации с помощью аутентификации в языке программирования JavaScript

Краткие теоретические положения

Объект `Form` служит контейнером для пересылки данных от пользователя на сервер. Объект `Document` является контейнером всех форм на странице. Чтобы обратиться к отдельному элементу в конкретной форме, необходимо воспользоваться конструкцией `document.Имя Формы.ИмяЭлемента`.

```
Document. Myform.myelement.value = "Мое текстовое поле";
```

Рассмотрим основные свойства, методы и события формы

Таблица 1. Свойства, методы и события формы

Характеристика	Название
Свойства	<code>action</code> , <code>elements[]</code> , <code>length</code> , <code>method</code> , <code>name</code> , <code>target</code>
Методы	<code>reset()</code> , <code>submit()</code>
События	<code>onReset</code> , <code>onSubmit</code>

Свойство **action** задает URL-адрес сценария на сервере, которому отправляется форма:

```
<form name="myform"
action=http://www.myserver.com/cgi-bin/registration.asp
method="post">
<!--Объекты формы- - >
</form>
```

Свойство **elements[]** представляет собой массив, в который включены все элементы формы. Ссылаться на элементы можно по их индексу, который начинается с нуля.

```
formName.element[2].value
```

Свойство **length** определяет количество элементов на форме.

Свойство **method** определяет способ отправки формы серверу. На практике чаще всего используются методы GET и POST. В методе GET данные передаются на адрес, указанный в свойстве **action**, через строку поиска (часть адреса, идущая после знака ?) в URL. В методе POST данные передаются в теле HTML – запроса. Этот метод следует использовать при передаче серверу большого объема данных.

Свойство **name** ссылается на внутренние имена форм.

Свойство **target** указывает окно назначения браузера, в котором должен отображаться результат отправки формы.

Метод **reset** сбрасывает значения всех элементов формы в исходное состояние.

Метод **submit ()** используется для передачи данных формы.

Событие **onReset** восстанавливает значения по умолчанию в полях формы и возникает при нажатии кнопки Reset или при выполнении метода **reset ()**.

Событие **onSubmit** запускается в момент отправки пользователем данных формы на сервер.

Рассмотрим элементы управления в формах.

Поля ввода данных предназначены для ввода однострочной информации. Пример данного элемента:

```
<input type="text" name="CustName" size=40 maxlength=60>
```

Поля для паролей предназначены для ввода паролей. Обычно браузеры вместо вводимых символов на экран выводят звездочки (*). Пример этого элемента:

```
<input type="password" name="pass" size=20>
```

Многострочные поля ввода предназначены для ввода информации, которая не помещается в обычных полях ввода. Чтобы создать текстовое поле из десяти строк по 80 символов в каждом, наберите следующий код:

```
<textarea name="Comments" rows = 10 cols=80></textarea>
```

Флажки предназначены для обозначения логических данных, принимающих значение true или false. Пример этого элемента:

```
<input type =”checkbox” name=”hobby” value=”Литература”>
```

Использование флажка на форме:

Литература

Переключатели предназначены, чтобы пользователь выбрал только один вариант из нескольких возможных. Пример этого элемента:

```
< input type =”radio” name =”trade”  
value =”student”> Студент <br>
```

Использование переключателя на форме:

Студент

Списки также предназначены для выбора элемента. Однако в отличие от переключателей из списка можно выбрать несколько элементов. Пример списка:

```
<select name =”writer” size=2 multiple>  
<option> Иванов </option>  
<option> Петров </option>  
<option> Сидоров </option>  
</select>
```

Если параметр size=1, то список раскрывающийся. При size=2 список будет прокручиваться внутри окна. Параметр multiple служит для выбора нескольких элементов из списка.

Кнопки

В стандарте HTML определены четыре типа кнопок, которые можно включать в формы: Button (обычная кнопка), Submit (кноп-

ка отправки данных на сервер), Reset (кнопка сброса полей формы), Image (кнопка с изображением).

Практическое задание №32

Цель работы: изучить методику создания формы регистрации на сайте

Назначение программы: программа создает типовую форму регистрации на сайте следующего вида

Регистрация на сайте

Имя: *	<input type="text"/>
Фамилия: *	<input type="text"/>
Адрес Email: *	<input type="text"/>
Логин: *	<input type="text"/>
Пароль: *	<input type="text"/>
Повтор Пароля: *	<input type="text"/>
Страна:	<input type="text" value="Россия"/> v
Ваш пол:	<input type="radio"/> Мужской <input type="radio"/> Женский
<input type="checkbox"/>	С условием регистрации согласен.

Поля, помеченные звездочкой (*), являются обязательными для заполнения.

Рис. 9 Форма регистрации на сайте.

С помощью функции `validateForm ()` при отправке формы на сервер (см. текст программы) происходит проверка заполнения всех полей формы.

Текст программы:

```
<html>
<head>
  <title>Регистрация на сайте</title>
<script>
//Функция проверки формы на наличие пустых полей
function validateForm()
{
var validateFlag=true;//возвращаемая функция логическая переменная
for (var i=0; i<6;i++)//Проверяем первые шесть элементов
{
// В действительности переменная countElem не нужна. Она введена для
// улучшения читабельности кода
var countElem=document.regform.elements[i];
if (countElem.value=="") //Текущий элемент пустой?
{//Если да, то
  validateFlag=false; //Устанавливаем флаг в false
//Выдаем предупреждение для текущего элемента
alert ("Введите значение в поле"+countElem.name);
break; // Цикл уже можно не продолжать
}
}
alert('Поздравляем, Вы успешно прошли регистрацию');
return validateFlag;
}
</script>
</head>
<body>
<h2>Регистрация на сайте</h2>
<!--Создаем форму-->
```

```

<form name="regform" method="post" onsubmit="return validate-
Form();">
<!--Для удобства форматирования элементы формы располагаем в
таблице-->
<table width="100%">
<tr>
<!--Создаем поля ввода данных-->
<td>Имя: <sup>*</sup></td>
<!--Поле Имя-->
<td><input type="text" size="35" name="firstname"></td>
</tr>
<tr>
<td>Фамилия: <sup>*</sup></td>
<!--Поле Фамилия-->
<td><input type="text" size="35" name="lastname"></td>
</tr>
<tr>
<td>Адрес Email: <sup>*</sup></td>
<!--Поле Email-->
<td><input type="text" size="35" name="email"></td>
</tr>
<tr>
<td>Логин: <sup>*</sup></td>
<!--Поле Логин-->
<td><input type="text" size="35" name="login"></td>
</tr>
<tr>
<td>Пароль: <sup>*</sup></td>
<!--Поле Пароль-->
<td><input type="password" size="20" name="pass1"></td>
</tr>
<tr>
<td>Повтор Пароля: <sup>*</sup></td>
<!--Поле Повтор Пароля-->
<td><input type="password" size="20" name="pass2"></td>
</tr>
<tr>

```

```

<td>Страна:</td>
<!--Далее список-->
<td><select size="1" name="country">
<!--Вариант списка-->
<option value="Russia">Россия</option>
<option value="Ukraine">Украина</option>
<option value="Belorussia">Белоруссия</option>
<option value="USA">США</option>
<option value="France">Франция</option>
<option value="Great Britain">Великобритания</option>
<option value="other">Другая</option>
</td>
</tr>
<tr>
<td>Ваш пол:</td>
<td>
<!--Формируем переключатели -->
<input type="radio" name="sex" value="male">Мужской<br/>
<input type="radio" name="sex" value="female">Женский<br/>
</td>
</tr>
<tr>
<td colspan=2>
<!--Флажок со своим описанием занимает всю строку в таблице -->
<input type="checkbox" name="consent">
С условием регистрации согласен.</td>
</tr>
<tr>
<td colspan=2> Поля, помеченные звездочкой (<sup>*</sup>), яв-
ляются обязательными для заполнения.</td>
</tr>
<tr>
<!--Заканчивают форму две стандартные кнопки-->
<td><input type="submit" value="Зарегистрировать"></td>
<td><input type="reset" value="Очистить"></td>
</tr>
</table>

```

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

Дополнительные задания:

1. Усовершенствуйте форму регистрации на сайте таким образом, чтобы происходила проверка на совпадение полей: «пароль» и «повтор пароля».

2. Усовершенствуйте форму регистрации на сайте таким образом, чтобы происходила проверка поля адрес E-mail на наличие символа @.

Практическое задание №33

Цель работы: изучить методику защиты паролем конкретной страницы сайта.

Назначение программы: если пароль неверный, то пользователь переадресовывается, если верный, то выдается сообщение и загружается текущая страница. Данный Java-скрипт следует вставить между тегам `<head>` и `</head>` той Web-страницы, которую следует защитить паролем. В данном демонстрационном примере пароль `password`, в случае неверного ввода пользователь переадресовывается на Web-страницу `http://studia.scriptic.ru/`

Текст программы:

```
<head>
<title>Защита паролем. Если пароль неверный то пользователь переадресовывается, если верный то выдается сообщение и загружается страница.</title>
<script language=JavaScript>
<!--//
p = prompt('Пожалуйста, введите пароль');
// Запрос пароля
if (p=='password') {
// Не забудьте сменить пароль
```



```

alert('Вход разрешен')
// текст сообщения о том что пароль принят
} else {
top.location.href="http://studia.scriptic.ru/"
// Страница на которую будет направлен пользователь если пароль
не верен
}
//-->
</script>
</head>

```

Практическое задание №34

Цель работы: изучить методику защиты паролем сайта, при которой на каждой странице сайта используется отдельный логин и пароль.

Назначение программы: В данном демонстрационном примере рассматривается защита сайта из трех страниц. Например, для страницы news.htm используется логин «пример1» и пароль «пример1». Данный Java-скрипт может быть реализован на главной странице сайта.

Текст программы:

```

< head>
< script language="JavaScript">
<!-- Begin
function Login(){
var done=0;
var username=document.login.username.value;
username=username.toLowerCase();
var password=document.login.password.value;
password=password.toLowerCase();
if (username=="пример" && password=="пример") { win-
dow.location="contakt.htm"; done=1; }
if (username=="пример1" && password=="пример1") { win-
dow.location="news.htm"; done=1; }

```

```

if (username=="пример2" && password=="пример2"){ win-
dow.location="serv.htm"; done=1;}
// и так далее...
if (done==0) { alert("Неверный пароль или имя пользователя!"); }
}
// End -->
</ script>
</head>
<body>
<center>
<form name=login>
<table width=225 border=1 cellpadding=3 bgcolor="#808080">
<tr><td colspan=2>
<p align="center"><b><font color="#FFFFFF"
size="3"></font></b></p>
</td></tr>
<tr><td>
<font size="3" color="#FFFFFF">
Имя:
</font>
</td><td>
<font size="3" color="#FFFFFF">
<input type=text name=username>
</font>
</td></tr>
<tr><td>
<font size="3" color="#FFFFFF">
Пароль:
</font>
</td><td>
<font size="3" color="#FFFFFF">
<input type=text name=password>
</font>
</td></tr>
<tr><td colspan=2 align=center>
<font size="3" color="#FFFFFF">
<input type=button value="Перейти на сайт" onClick="Login()">

```

```
</font>  
</td></tr>  
</table>  
</form>  
</center>  
</body>
```

20. Использование форм в языке программирования JavaScript

Краткие теоретические положения по данному вопросу представлены в п. 19. В данном параграфе рассмотрим дополнительные примеры использования форм в задачах выбора вариантов и передачи информации на сервер.

Практическое задание №35

Цель работы: изучить методику решения задач выбора вариантов с использованием форм и флажков "checkbox".

Назначение программы: программа создает форму, в которой пользователю предлагается выбрать хобби. Программа анализирует ответ пользователя и выдает сообщение, каковы хобби пользователя.

Ваше хобби

- литература

- спорт

- туризм

Рис. 10 Форма выбора варианта с помощью флажков "checkbox"

Текст программы:

```

<script>
function raport() {
var num=document.myform.length;
var hob="Ваше хобби";
for (var i=0; i<num-1; i++) {
if (document.myform.hobby[i].checked==true) {
hob=hob+" " + document.myform.hobby[i].value;}
}
window.alert(hob);
}
</script>
<form name="myform">
<input type="checkbox" name="hobby" value="литература">
литература</br>
<input type="checkbox" name="hobby" value="спорт">
спорт</br>
<input type="checkbox" name="hobby" value="туризм">
туризм</br>
<input type="button" value="Отправить" onClick="raport()">
</form>

```

Практическое задание №36

Цель работы: изучить методику передачи информации на сервер с помощью формы.

Назначение программы: программа передает на сервер следующую информацию: e-mail пользователя и сообщение. Но поскольку программы “anuprogramm.php”, которая обрабатывает информацию на сервере не существует, при нажатии кнопки «**Отправить**» будет выдано сообщение “Невозможно найти удалённый сервер”.

Ваш e-mail

Сообщение

Сброс

Отправить

Рис. 11 Форма отправки сообщения на сервер

Текст программы:

```

<html>
<head>
<title>Форма</title>
<style type="text/css">
form
{
width:300px;
padding:20px;
border-with:1px;
border-style:solid;
background:#e0e0e0
}
</style>
</head>
<body>
<form action="http://anyserver.ru/anyprogramm.php" method="post">
<h2>Сообщение автору </h2>
Ваш E-mail:
<input type="text" name="from" value="" size=30/>
<br/>
Сообщение: <br>

```

```
<textarea name='text'cols=30 rows=8></textarea>  
<br/>  
<input type="reset"/>  
<input type="submit" name="Отправить" value="Отправить"/>  
</form>  
</body>  
</html>
```

Практическое задание №37

Цель работы: получить навыки самостоятельной разработки программ для решения задач выбора варианта с использованием форм и переключателей "radio".

Назначение программы: программа реализует тестовый вопрос, в котором из 4 возможных вариантов ответов надо выбрать один правильный. Программа должна проанализировать ответ пользователя и выдать сообщение:

1. Ответ правильный.
2. Ответ неправильный. Вам надо прочитать теоретический материал по сетевым атакам.

Какая сетевая атака связана с превышением допустимых пределов функционирования сети?

- Подслушивание (Sniffing).
- Отказ в обслуживании (DoS –атака).
- Атака Man in – the – Middle (человек в середине).
- Угадывание ключа.

Отправить

Рис. 12 Форма выбора варианта с помощью переключателей "radio".

21. Слои в языке программирования JavaScript

Краткие теоретические положения

Для создания слоев используется элемент `DIV`, являющийся блочным элементом разметки. Рассмотрим свойства, указывающие способ размещения и точное позиционирование слоев. Свойство `position` указывает способ размещения элементов. Оно может принимать четыре значения:

`absolute` – задает абсолютное свободное позиционирование;

`fixed` – аналогично `absolute`, но при прокрутке элемент не перемещается;

`relative` – определяет относительное свободное позиционирование;

`static` – задает статическое позиционирование, при котором элемент страницы отображается внутри общего «потока» текста.

Свойства `left`, `right`, `top` и `bottom` определяют точное позиционирование элементов:

`left` – задает горизонтальную позицию левой границы свободно позиционированного элемента;

`right` – определяет горизонтальную позицию правой границы свободно позиционированного элемента;

`top` – задает вертикальную позицию верхней границы свободно позиционированного элемента;

`bottom` – определяет вертикальную позицию нижней границы свободно позиционированного элемента.

Например, следующий фрагмент программы выводит слой в правом нижнем углу:

```
<div id=third style="position:absolute; bottom:50px;right:50px;
Background-color:red;">
```

Слой в правом нижнем углу

```
</div>
```

Свойства `height` и `width` определяют высоту и ширину блока, определяющего слой.

Свойство `visibility` используется для того, чтобы делать слои видимыми и скрытыми. Оно может принимать следующие значения:

hidden – скрывает элемент;
collapse – если не используется с таблицами, то аналогично значению hidden;
inherit – значение по умолчанию, которое указывает, что видимость наследуется от родителя;
visible – делает элемент видимым.

Практическое задание №38

Цель работы: изучить методику использования и позиционирования слоев.

Назначение программы: программа выводит на экран три слоя, причем третий слой является прозрачным, поэтому сквозь него виден второй слой.

Текст программы:

```
<html>
<head>
<title>Перекрывающиеся слои</title>
</head>
<body>
<h2>Перекрывающиеся слои </h2>
<div id=first style="position:absolute; top:50px;left:50px;
height:150px; width:150px; background-color:orange;">
1 слой
</div>
<div id=second style="position:absolute; top:70px; left:70px;
height:150px; width:150px; background-color:red;">
2 слой
</div>
<div id=third style="position:absolute; top:90px;left:90px;
height:150px; width:150px;">
3 слой
</div>
</body>
</html>
```


Практическое задание №39

Цель работы: изучить методику управления видимостью слоев.

Назначение программы: программа создает невидимый слой, который можно отобразить или вновь скрыть щелчком кнопки мыши по ссылке.

Текст программы:

```
<html>
<head>
<title> Управление видимостью </title>
<script>
//Функция, осуществляющая изменение видимости слоя
function vis(){
//Получаем доступ к слою
var Layer=document.getElementById('lay');
//выясняем состояние слоя
if (Layer.style.visibility=='hidden') {
//если слой невидим, то делаем видимым
Layer.style.visibility='visible';
}
else{
//если слой видим, то делаем невидимым
Layer.style.visibility='hidden';
}
}
</script>
</head>
<body>
<h2> Упраление видимостью </h2>
<div id=control style="position:absolute; top:50px;left:50px;">
<a href="javascript:vis();">Скрыть/Показать слой </a>
</div>
<div id=lay style="position:absolute; top:70px;left:50px;
height:100px;width:250px; background-color:orange;
visibility:hidden">
```

Этот слой по умолчанию невидим. Чтобы его скрыть или увидеть, необходимо щелкнуть кнопкой мыши на ссылке

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

22. Защита контента от несанкционированного копирования информации в языке программирования JavaScript

Краткие теоретические положения

Рассмотрим 3 варианта Java-скриптов, предназначенных для защиты контента сайта от несанкционированного копирования. Java-скрипты необходимо вставлять между тегами <head> и </head>, например после тега </title>.

1. Рассмотрим Java-скрипт, позволяющий скопировать текст, но после вставки текста в конце появляется ссылка на источник, например: Источник: © www.swsu.ru

```
<script type="text/javascript">
function addLink() {
var body_element = document.getElementsByTagName('body')[0];
var selection;
selection = window.getSelection();
var pagelink = "<p>Источник: <a
href='"+document.location.href+"'>"+"</a>    © www.swsu.ru";
var copytext = selection + pagelink;
var newdiv = document.createElement('div');
newdiv.style.position='absolute';
newdiv.style.left='-99999px';
body_element.appendChild(newdiv);
newdiv.innerHTML = copytext;
selection.selectAllChildren(newdiv);
window.setTimeout(function() {
body_element.removeChild(newdiv);
```

```

},0);
}
document.oncopy = addLink;
</script>

```

2. Рассмотрим второй Java-скрипт, который не только запрещает выделять текст на странице, но и пользоваться правой кнопкой мышки.

```

<script language="JavaScript">
document.onselectstart=function(){return false}
document.oncontextmenu=function(){return false}
document.onmousedown=function(){return false}
</script>

```

3. Рассмотрим третий Java-скрипт, который защищает контент от несанкционированного копирования прозрачным слоем. При попытке копирования пользователь копирует прозрачный слой.

```

<html>
<head>
<title>Издательство НТ Пресс</title>
<div style="position: relative;">
//далее защищаемый прозрачной картинкой контент сайта


</div>
</head>
<body>
</body>
</html>

```

Практическое задание №40

Цель работы: изучить методику защиты контента сайта от несанкционированного копирования

Назначение программы: программа реализует сайт, состоящий из трех страниц. Сайт на произвольную тему разработайте самостоятельно или используйте сайт из практического задания №8. Каждую страницу сайта защитите от несанкционированного копирования разными видами Java-скриптов, рассмотренными в кратких теоретических положениях.

Список контрольных вопросов

1. Является ли HTML языком программирования? Можно ли на HTML разрабатывать динамические сайты?
2. Какова структура HTML документа? Какой тег в заголовочной части документа HTML является обязательным ?
3. Каким образом в документе HTML поисковым машинам предоставляется информация о тематике документа?
4. Каким образом можно оформить стиль документа HTML ? Например, как определить стиль при котором все заголовки, ограниченные тегом <h3> будут выделены синим цветом?
5. Сколько уровней заголовков предусмотрено в спецификации HTML? Каким образом можно центрировать заголовок в HTML? Приведите пример.
6. Как осуществляется разметка абзацев в HTML? Приведите пример.
7. Какие виды списков существуют в HTML? Приведите примеры.
8. Запишите тег, который осуществляет вывод текста шрифтом Arial.
9. Каким образом картинку на веб-странице можно изобразить слева, а поясняющий картинку текст справа ? Приведите пример.
10. Каким образом можно изменить размеры картинки на веб – странице? Поясните на конкретном примере.

11. Как определить ширину ячейки таблицы при ее создании в HTML? Поясните на примере.

12. Какой атрибут необходимо использовать для создания в HTML таблицы следующего вида?

13. Запишите тег ссылки на поисковую систему Yandex.

14. Поясните, в чем сущность относительного и абсолютного пути к файлу при ссылке на веб-страницу?

15. Дайте определение фрейма. Запишите тег, который разбивает экран на 3 строки в соотношении 30%, 50% и 20%.

16. Как осуществляется навигация во фреймах? То есть, каким образом определяется, какая HTML-страница и в каком окне отобразится при щелчке по ссылке? Приведите пример.

17. Какие методы ввода и вывода информации существуют в языке JavaScript? Поясните на примерах.

18. Нужно ли при объявлении переменной в языке JavaScript указывать ее тип?

19. Каким образом можно вывести значение всех элементов массива в языке JavaScript?

20. Какие операторы языка JavaScript служат для реализации механизмов ветвления. Запишите общий вид этих операторов.

21. Какие операторы языка JavaScript служат для реализации механизмов цикла. Запишите общий вид этих операторов.

22. Дайте определение функции и запишите ее общий вид в языке JavaScript.

23. Для какой цели в языке JavaScript используется оператор return?

24. Приведите примеры двух-трех событий и соответствующих им обработчиков этих событий.

25. Каким образом можно программным путем обратиться к полю формы и присвоить ему определенное значение? Приведите пример.

26. Каким образом программа определяет адрес URL-сценария на сервере по которому передаются данные формы?

27. Какие методы передачи данных формы на сервер существуют и в чем их различие?
28. Дайте краткую характеристику основных элементов управления, используемых в формах (поля ввода данных, поля ввода паролей, флажки, переключатели, списки)
29. С какой целью при работе с формами используются метод Reset и событие onReset?
30. Какие существуют методы защиты контента сайта от не-санкционированного копирования информации. В чем их различие?

Список литературы

1. Мухаметтов Г. В. HTML и CSS/ Г. В. Мухаметтов. – М.: ИТ Пресс, 2008. – 365с.
2. Дунаев В. В. Сценарии для Web-сайта: PHP и JavaScript. – 2-е изд., перераб. и доп. – СПб.: БХВ – Петербург, 2008. -576 с.
3. Днепров А. Г. JavaScript на 100%. – СПб.: Питер, 2008. - 304 с.
4. Колисниченко Д.Н. Современный сайт на PHP и JavaScript. - СПб.: Питер, 2009. – 176 с.
5. Дронов В.А. JavaScript и AJAX в Web-дизайне. - СПб.: БХВ-Петербург, 2008. – 728 с.
6. Прохоренок Н.А., Дронов В. А.: HTML, JavaScript, PHP и MySQL. - СПб.: БХВ-Петербург, 2015. – 766 с.
7. Гарнаев А.Ю. Web-программирование на Java и JavaScript. - СПб.: БХВ-Петербург, 2005. – 1040 с.
8. Дуванов А.А. Web-конструирование. DHTML. - СПб.: БХВ-Петербург, 2003. – 512 с.
9. Кисленко Н. П. HTML. Самое необходимое. БХВ - Петербург, 2008. – 353с.
10. Гаевский А. Ю., Романовский В. А. 100% самоучитель по созданию Web – страниц и Web-сайтов. HTML и JavaScript. - Триумф, 2008. -454 с.