

A decorative graphic consisting of a series of vertical black lines of varying thicknesses, creating a striped effect that runs down the left side of the page.

HTML

ТЕХНОЛОГИЯ

Санкт-Петербург
Издательство СПбГЭТУ «ЛЭТИ»
2000

Министерство образования РФ

Санкт-Петербургский государственный
электротехнический университет «ЛЭТИ»

Н. И. Михайлов, Н. Р. Кириллова, В. В. Перепеловский

HTML-ТЕХНОЛОГИЯ

Учебное пособие

Санкт-Петербург
Издательство СПбГЭТУ «ЛЭТИ»
2000

УДК 681.3.06
ББК 32.973
М17

М17 Михайлов Н. И., Кириллова Н. Р., Перепеловский В. В.
HTML-технология: Учеб. пособие. СПб.: Изд-во СПбГЭТУ
«ЛЭТИ», 2000. — 72 с.

Изложены основные положения HTML-технологии. Приведены задачи, упражнения и контрольные вопросы по темам, позволяющие глубже изучить теоретический материал и вырабатывать практические навыки разработки HTML-документов.

Предназначено для студентов, обучающихся по направлению «Физическая электроника» в рамках курса «Введение в Интернет-технологии».

Рецензенты:

*кафедра автоматике и вычислительной техники
(Санкт-Петербургский технический университет);*

*д-р физ.-мат. наук, проф. В. Д. Байков,
(Санкт-Петербургский государственный
университет культуры и искусств)*

Утверждено

*редакционно-издательским советом университета
в качестве учебного пособия*

ISBN 5-7629-0337-0

© СПбГЭТУ «ЛЭТИ», 2000

ВМЕСТО ВВЕДЕНИЯ

К числу наиболее известных стандартов, созданных W3C (World Wide Web — организация, занимающееся разработкой и реализацией стандартов в Internet) относятся HTML и CSS. В данном учебном пособии рассматриваются некоторые компоненты этих стандартов, хотя, как показывает практика, разработка стандарта не приводит к полному его применению, даже среди организаций — членов консорциума. Однако основные положения этих стандартов выполняются и достойны рассмотрения.

■ ПРИНЦИПЫ ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ

За основу модели разметки документов в HTML принята тэговая модель. Тэговая модель описывает документ как совокупность контейнеров, каждый из которых начинается и заканчивается тэгами, то есть документ HTML представляет собой не что иное, как обычный текстовый файл с добавленными в него управляющими HTML-кодами (тэгами).

Общая схема построения контейнера в формате HTML может быть записана в следующем виде:

```
"контейнер" := <"имя тэга" "список атрибутов">
                содержание контейнера
                </"имя тэга">
```

Имя конечного тэга идентично имени начального. Перед именем конечного тэга ставится косая черта (/), например, для тэга курсивного начертания шрифта <i> закрывающая пара представляет собой </i>, для тэга заголовка <TITLE> закрывающей парой будет </TITLE>.

В некоторых случаях конечные тэги в документе можно опустить. Большинство браузеров (программ просмотра) реализованы так, что при обработке текста документа начальный тэг воспринимается как конечный тэг предыдущего. Самый распространенный тэг такого типа — тэг абзаца <P>. Поскольку он используется в документе очень часто, то его обычно ставят только в начале каждого абзаца. Когда один абзац заканчивается,

следующий тэг <P> сигнализирует браузеру о том, что нужно завершить данный абзац и начать следующий.

Атрибуты тэга следуют за именем и отделяются друг от друга одним или несколькими знаками табуляции, пробелами или символами возврата к началу строки.

Спецификация атрибута состоит из расположенных в следующем порядке:

- имени атрибута, например WIDTH;
- знака равенства (=);
- значения атрибута, которое задается строкой символов, например "80".

Пример:

```
WIDTH="80".
```

Порядок записи атрибутов в тэге роли не играет. Значение атрибута, если таковое имеется, следует за знаком равенства, стоящим после имени атрибута. Если значением атрибута является одно слово (или число), то его можно просто указать после знака равенства, не выделяя дополнительно. Все остальные значения необходимо заключать в апострофы или в кавычки, особенно если эти значения содержат несколько разделенных пробелами слов.

Если значение атрибута такое же, как его имя, может быть использован минимальный синтаксис атрибута; например, <UL COMPACT="COMPACT"> можно сократить до <UL COMPACT>. Некоторые браузеры требуют минимизации для ряда атрибутов (COMPACT, ISMAP, CHECKED, NOWRAP, NOSHADE, NOHREF). Целесообразнее использовать минимизированный синтаксис.

Элементы HTML можно разделить на три основные категории:

- заголовочные элементы (элементы, используемые в тэге <HEAD> и содержащие информацию о документе в целом): <TITLE>, <ISINDEX>, <BASE>, <META>, <LINK>, <SCRIPT>, <STYLE>;
- элементы, которые структурируют документ, например, разделяют на части и параграфы (абзацы): <H1>, <H2>, <H3>, <H4>, <H5>, <H6>, <ADDRESS>, и элементы <P>, , , <DL>, <PRE>, <DIV>, <CENTER>, <BLOCKQUOTE>, <FORM>, <ISINDEX>, <HR>, <TABLE>;
- текстовые элементы:

- простой текст, возможно содержащий escape-последовательности (например, &);
- выражения разметки: , , <DFN>, <CODE>, <SAMP>, <KBD>, <VAR>, <CITE>;
- разметка шрифта: <TT>, <I>, , <U>, <STRIKE>, <BIG>, <SMALL>, <SUB>, <SUP>;
- специальные элементы: <A>, , <APPLET>, , <BASEFONT>,
, <SCRIPT>, <MAP>;
- элементы формы: <INPUT>, <SELECT>, <TEXTAREA>.

Структура документа

Структура HTML-документа позволяет использовать вложенные друг в друга контейнеры. HTML-документ — это контейнер, который начинается с тэга <HTML> и заканчивается тэгом </HTML>.

Пример:

```
<HTML>
  <HEAD>
    <TITLE>
      Пример
    </TITLE>
  </HEAD>
  <BODY>
    Пример простого документа
  </BODY>
</HTML>
```

Контейнер HTML состоит из двух вложенных контейнеров — заголовка документа (HEAD) и тела документа (BODY).

Контрольные вопросы

1. Какой контейнер должен открываться первым в HTML-документе?
 - а) BODY;
 - б) HEAD;
 - в) HTML;
 - г) Все перечисленные тэги возможны.

2. Какой тэг синтаксически неверен?

- а) <HTML>...</HTML>;
- б) <Html>...</Html>;
- в) <HtMl>...</hTmL>;
- г) все варианты верны.

<HEAD>

Основное назначение тэга <HEAD> — описание общих для данного документа параметров отображения. К таким параметрам можно отнести: стиль отображения документа, общий базовый адрес гипертекстовых ссылок, идентификатор и имя документа и т. п.

В документе допускается только один элемент <HEAD>, который располагается до элемента <BODY>.

Контейнер HEAD может содержать следующие тэги (в любом порядке): <TITLE>, <ISINDEX>, <BASE>, <META>, <LINK>, <STYLE> и <SCRIPT>. Подробнее тэги будут описаны ниже.

<TITLE>

Содержание тэга <TITLE> отображается в поле названия документа.

<TITLE> имеет следующий синтаксис:

```
<TITLE> Название документа </TITLE>
```

Пример:

```
<HEAD>
  <TITLE>
    Home Page by Petrov
  </TITLE>
</HEAD>
```

<BASE>

Тэг <BASE> служит для указания полного базового URL-адреса (сетевого адреса для связанного ресурса) документа. С его помощью относительная ссылка (см. далее) продолжает работать, если HTML-документ переносится в другой каталог или на другой компьютер. Тэг <BASE> имеет один обязательный

атрибут HREF, после которого указывается полный URL-адрес документа.

Синтаксис тэга <BASE>:

```
<BASE HREF="URL">
```

Например, если в заголовке задано:

```
<BASE HREF="http://eivt.da.ru/>
```

гипертекстовая ссылка вида будет расширена до .

<META>

Назначение тэга <META> — задание метаданных (информации о документе), например, авторства, срока истечения даты, списка ключевых слов и т. д.

Синтаксис тэга <META>:

```
<META NAME=имя элемента метаданных
  CONTENT=содержимое информации>
```

или

```
<META HTTP-EQUIV=имя элемента метаданных
  CONTENT=содержимое метаданных>
```

HTTP-EQUIV является аналогом атрибута NAME на ранних версиях браузеров, CONTENT — содержимое метаданных. META-тэги с атрибутом NAME используются в случаях, когда применение тэга <META> приводит к некорректному результату. Например, атрибут "Keywords" необходимо использовать как с атрибутом NAME, так и с атрибутом HTTP-EQUIV.

Пример:

```
<META NAME=DESCRIPTION CONTENT="Department EIVT">
<META NAME=KEYWORDS CONTENT="MESFET, EIVT, LETI">
```

Атрибут NAME может принимать следующие значения:

- Expires — дата устаревания.

Пример:

```
<META HTTP-EQUIV="expires"
  CONTENT="Wed, 26 Feb 1997 08:21:57 GMT">
```

В Netscape Navigator реализация данного примера выглядит следующим образом: если указанная дата прошла, то очеред-

ной запрос этого документа вызывает повторный сетевой запрос, а не подгрузку документа из кэша (буфера для временного хранения информации). Дата со значением "0" интерпретируется как "сейчас". Такое значение заставляет браузер каждый раз при запросе проверять, изменялся ли этот документ.

- Content-Type — указывает типа документа. Можно указать кодировку страницы (charset). Если же указывать charset в содержании META-тэга, то Netscape Navigator выведет такую страницу уже в заданном charset. Если текст страницы в кодировке Windows, а значение charset=KOI8-R, то пользователь не сможет поменять перекодирование, чтобы увидеть нормальные слова.

Пример:

```
<META HTTP-EQUIV="Content-type"
CONTENT="text/html; charset=ISO-2022-JP">
```

- Content-language — указывает языка документа. Может использоваться поисковыми машинами при индексировании страниц. Комбинация поля Accept-Language (посылаемого браузером) с содержимым Content-language может быть условием выбора сервером того или иного языка.

Пример:

```
<META HTTP-EQUIV="Content-language" CONTENT="en-GB">
```

В спецификации HTML 4.0 появилась возможность явного указания языка — `<html lang="en">` Content-language: en-GB. Язык описывается парой значений (язык — диалект). В приведенном примере: английский — Великобритания.

- Refresh — определяет задержку в секундах, после которой браузер автоматически обновляет документ. Дополнительная возможность — автоматическая загрузка другого документа.

Пример:

```
<META HTTP-EQUIV="Refresh"
Content="3, URL=http://www.name.com/page.html">
```

В данном примере загрузка станицы производится с указанного URL после трехсекундной задержки. В Netscape Navigator это дает такой же эффект, что и нажатие на кнопку Reload.

- Cache-Control — определяет действия кэша по отношению к данному документу. Возможные значения: Public — документ кэшируется в доступных для всех кэшах; Private — только в частном кэше; no-cache — не может быть кэширован; no-store — может быть кэширован, но не сохраняется
- Vary — определяет доступные альтернативы для указанных в CONTENT полей HTTP-заголовка.

Пример:

```
<META HTTP-EQUIV="Vary" CONTENT="Content-language">,
что эквивалентно HTTP-заголовку
Vary: Content-language.
```

- Robots — управляет индексацией страницы для поисковых роботов. Возможные значения: ALL, NONE, INDEX, NOINDEX, FOLLOW, NOFOLLOW.

Пример:

```
<META NAME="Robots" CONTENT="NOINDEX,FOLLOW">
```

- Description — краткая аннотация содержания документа. Используется поисковыми системами для описания документа. Этот тэг полезен в случаях, когда в документе мало текста, когда это управляющий фреймами (см. ниже) файл (frameset) или когда в начале документа используются скрипты (сценарии — последовательности команд). Длина текста — не более 100 символов.

Пример:

```
<META NAME="Description"
CONTENT="Документ содержит словарь META-тэгов">
```

- Keywords — используется поисковыми системами для индексирования документа (внесения ключевых слов в указатель поисковых систем). Обычно здесь указываются синонимы к словам в заголовке (TITLE) или альтернативный заголовок. Длина списка — до 1000 символов. Не допускается использование одного и того же ключевого слова более семи раз, поисковые системы игнорируют это слово.

Пример:

```
<META NAME="Keywords"
CONTENT="тэги, метаданные, список">
```

- **Document-state** — управляет индексированием документа для поисковых систем. Определяет частоту индексации — или индексировать документ один раз, или реиндексировать документ регулярно. Возможные значения: *Static*, *Dynamic*.

Пример:

```
<META NAME="Document-state" CONTENT="Static">
```

<LINK>

Тэг **<LINK>** предоставляет документу независимый от среды метод определения отношения данного документа к другим документам и ресурсам сети. Используется с аргументами *REL* и *REV*. С помощью тэга **<LINK>** можно:

- привязывать такие ассоциированные ресурсы, как таблицы стилей и скрипты;
- создавать в документе специальные навигационные кнопки или меню;
- управлять процессом отображения набора HTML-файлов в печатные документы;
- предоставлять альтернативные формы для данного документа.

Пример:

```
<LINK rel=help href="http://www.name.com/help.html">
```

где `http://www.name.com/help.html` — страница помощи по данному документу.

Тэги **<LINK>** могут использоваться только в заголовке документа (**HEAD**).

Атрибут **REL** может принимать значения: *top*, *contents*, *index*, *glossary*, *copyright*, *next*, *previous*, *search*.

Рассмотрим некоторые из рекомендованных типов взаимосвязей:

- **rel=top** указывает на вершину в некоторой иерархической структуре, например на первую либо титульную страницу в некотором наборе документов;
- **rel=contents** указывает на файл, где приводится оглавление к данному документу;

- **rel=index** указывает на другой документ, который можно использовать в целях индексного поиска по текущему документу;
- **rel=glossary** указывает на документ, где содержится глоссарий терминов, относящихся к текущему документу;
- **rel=copyright** отсылает к тексту, в котором указаны авторские права на данный документ;
- **rel=next** указывает на следующий документ в заранее определенном маршруте просмотра (например, может использоваться для предупреждающей автоматической загрузки браузером следующей страницы);
- **rel=previous** отсылает к предыдущему документу в некотором определенном маршруте просмотра;
- **rel=help** указывает на документ, предлагающий помощь (например, это может быть текст, дающий более развернутое описание и предлагающий ссылки на другие документы по этой теме);
- **rel=search** ведет к поисковой странице, контролирующей некий набор страниц, связанных общей темой.

Многие поисковые системы вводят свои дополнительные значения аргументов *REL* и *REV*.

<SCRIPT>

Тэг **<SCRIPT>** зарезервирован для будущего использования.

Контрольные вопросы

1. Какой из приведенных вариантов задаст адрес, от которого будут определяться все относительные адреса в документе?
 - а) `<BASE="URL">`;
 - б) `<BASE="URL">`;
 - в) `<BASE HREF="URL">`;
 - г) `<META NAME="BASE" CONTENT="URL">`.
2. Как правильно задать набор ключевых слов документа?
 - а) `<KEYWORDS="EIVT, LETI">`;
 - б) `<META KEYWORDS="EIVT, LETI">`;

- в) `<META NAME=CONTENT KEYWORDS="EIVT, LETI">`;
 г) `<META NAME=KEYWORDS CONTENT="EIVT, LETI">`.
3. Как задать дату, по прошествии которой запрос документа произойдет через сеть вместо подгрузки из кэша?
- а) `<META HTTP-EQUIV="EXPIRES" CONTENT="Wed, 26 Feb 1997 08:21:57 GMT">`;
 б) `<META HTTP-EQUIV="REFRESH" CONTENT="Wed, 26 Feb 1997 08:21:57 GMT">`;
 в) `<META NAME="REFRESH" CONTENT="Wed, 26 Feb 1997 08:21:57 GMT">`;
 г) `<EXPIRES="Wed, 26 Feb 1997 08:21:57 GMT">`.
4. Какой из перечисленных тэгов может встречаться в контейнере HEAD более одного раза?
- а) `<TITLE>...</TITLE>`;
 б) `<META>`;
 в) `<BASE>`;
 г) ни один из перечисленных.
5. Какое действие выполняет тэг `<META NAME="DOCUMENT-STATE" CONTENT="STATIC">`?
- а) Запрет на внесение в документ изменений;
 б) закрепление сетевого адреса документа;
 в) указание поисковым роботам индексировать документ лишь однажды;
 г) указание браузеру выводить документ, не изменяя кодировки.
6. Какая из перечисленных команд задаст кодировку страницы?
- а) `<CHARSET="KOI8-R">`;
 б) `<META NAME="CHARSET" CONTENT="KOI8-R">`;
 в) `<META CHARSET="KOI8-R">`;
 г) `<META HTTP-EQUIV="DOCUMENT-STATE" CONTENT="TEXT/HTML; CHARSET=KOI8-R">`.
7. Какая из перечисленных команд задаст описание сайта?
- а) `<DESCRIPTION>...</DESCRIPTION>`;
 б) `<META NAME="DESCRIPTION" CONTENT="...">`;
 в) `<DESCRIPTION HREF="URL">`;
 г) `<META DESCRIPTION="..." CONTENT="URL">`.

<BODY>

Описание тэгов тела документа следует начать с тэга `<BODY>`. В отличие от тэга `<HEAD>`, тэг `<BODY>` имеет следующие атрибуты:

ID — идентификатор тэга. Используется для именования тэгов, а также в качестве точки перехода по гипертекстовой ссылке. Данный атрибут имеется практически во всех тэгах тела документа. *Практическое использование идентификатора `<BODY>` сомнительно, так как отображаемая часть документа, собственно, и начинается с этого тэга.*

LANG — определяет язык документа в виде двухсимвольного кода ISO-639, за которым следует через точку необязательный код страны в формате ISO-3166. По замыслу разработчиков стандарта языка данный атрибут должен распознаваться программами интерпретации и управлять отображением многоязычных текстов.

CLASS — иерархически организованное имя типа "ADDITION.FIRST". Предназначено для связывания тэга текста с определенным стилем отображения. Реально пока не используется.

Все три перечисленных атрибута являются общими для тэгов тела и в дальнейшем обсуждаться не будут. Если атрибут для тэга тела документа не будет разрешен, то это будет указано специально.

BACKGROUND — определяет цвет фона, на котором отображается текст документа. В приведенном ниже примере в качестве фона был использован небольшой графический образ "bgr.gif".

Пример:

```
<BODYBACKGROUND="bgr.gif">
```

Как видно из этого примера, в качестве значения данного атрибута используется адрес в сокращенной форме URL. В данном случае это адрес локального файла. Следует заметить, что разные интерфейсы пользователя поддерживают различные дополнительные атрибуты для тэга BODY.

Атрибут	Значение
BACKGROUND=#FFFFFF	Цвет фона
TEXT=#0000FF	Цвет текста

VLINK=#FF0000 Цвет пройденных гипертекстовых ссылок

LINK=#00FF00 Цвет гипертекстовой ссылки

Строка #XXXXXX определяет цвет в терминах цветовой модели RGB в шестнадцатеричной нотации. Здесь цвет текста — синий, фона — белый, пройденных ссылок — красный, а новых ссылок — зеленый. Если в качестве атрибутов тэга <BODY> указать:

```
<BODY BACKCOLOR=#FFFFFF
TEXT=#0000FF VLINK=#FF0000 LINK=#00FF00>
```

то цвет фона будет желтым, текста — синим, ссылок — зеленым, а пройденных ссылок — красным. Однако пользоваться этими атрибутами следует крайне осторожно, так как у пользователя может оказаться интерфейс, который эти параметры не интерпретирует.

Microsoft Internet Explorer и Netscape Navigator допускают применение атрибутов **LEFTMARGIN=n** и **TOPMARGIN=n** в тэге <BODY>. Атрибут **LEFTMARGIN** задает левое поле для всей страницы, **TOPMARGIN** определяет верхнее поле. Число n показывает ширину поля в пикселях. Например, тэг <BODY LEFTMARGIN="40"> создаст на всей странице левое поле шириной 40 пикселей. При n = 0 левое поле отсутствует.

■ ФОРМАТИРОВАНИЕ ТЕКСТА

Логическое форматирование

Логическое форматирование и структурирование являются необходимыми элементами хорошо составленного документа.

К элементам логического форматирования относятся:

Тэг	Значение
...	Выделение важных фрагментов
<CITE>...</CITE>	Цитирование
...	Усиление
<CODE>...</CODE>	Отображение примеров программного кода (синоним <pre>)
<SAMP>...</SAMP>	Выделение нескольких символов шрифтом фиксированной ширины

<KBD>...</KBD>	Выделение фрагмента текста, вводимого с клавиатуры
<VAR>...</VAR>	Переменная
<DFN>...</DFN>	Определение
<Q>...</Q>	Текст, заключенный в скобки

Пример:

```
<EM>
выделенный текст, обычно курсив
</EM>
```

Пример программного кода:

```
<code>
<html>
  <body>
    Простой пример
  </body>
</html>
</code>
```

Физическое форматирование

К элементам физического форматирования относятся:

Тэг	Значение
<I>...</I>	Курсив (italic)
...	Полужирный (bold)
<TT>...</TT>	Моноширинный шрифт (Courier)
<U>...</U>	Подчеркнутый текст (underline)
<S>...</S>	Перечеркнутый текст (strikethru)
<BIG>...</BIG>	Заглавные
<SMALL>...</SMALL>	Капитель
_{...}	Подстрочные символы (subscript)
^{...}	Надстрочные символы (superscript)

Пример:

```
<I>
курсив
```

```
</I>
строка
  <SUB>
    подстрочные символы
  </SUB>
```

Контрольные вопросы

1. Указать тэг, позволяющий вывести подстрочные символы:
 - а) строка `_{подстрочные символы}`;
 - б) строка `^{подстрочные символы}`;
 - в) строка `<TT>подстрочные символы </TT>`;
 - г) все перечисленные тэги ошибочны.
2. Указать тэг, позволяющий выделить (усилить) фрагмент текста:
 - а) строка `выделенный фрагмент (bold) `;
 - б) строка `<i>выделенный фрагмент (bold) </i>`;
 - в) строка `<s>выделенный фрагмент (bold) </s>`;
 - г) все перечисленные тэги ошибочны.
3. Каким образом вывести фрагмент программного кода?
 - а) Строка `<KBD><html><body></body></html></kbd>`;
 - б) строка `<code><html><body></body></html></code>`;
 - в) строка `<cite><html><body></body></html></cite>`;
 - г) все перечисленные тэги ошибочны.
4. Как задать белый цвет фона?
 - а) `<body bgcolor = "#FF0000" link = "#000000">`;
 - б) `<body bgcolor = "#FFFFFF" vlink = "#FF0000">`;
 - в) `<body bgcolor = "#000000" link = "#000000">`;
 - г) все перечисленные тэги ошибочны.

■ ШРИФТЫ

Одной из важных особенностей современного HTML является достаточно полный контроль облика создаваемого документа. На пути к полному контролю облика встает проблема шрифтов. Разработчик HTML-страницы ничего не знает о наборе шрифтов, которым располагает программа просмотра. Элемент `` частично решает данную проблему. Главное преимущество тэга

`` состоит в том, что он после окончания своего действия не разбивает строку. Поэтому тэг `` бывает очень полезен для изменения размера шрифта в середине строки (после окончания действия тэга `` восстанавливаются параметры шрифта, заданные как базовые).

Синтаксис тэга ``:

```
<FONT FACE = "type1, type1, type3" size = XX COLOR = xx>
```

ТЕКСТ

```
</FONT>
```

Атрибуты тэга ``:

FACE = "type" — позволяет указать тип шрифта (если таковым располагает программа просмотра).

SIZE = xx — позволяет задавать кегль (размер) шрифта: относительный (относительно базового размера, например `SIZE = +2`) и абсолютный (от 1 до 7), например `SIZE = 5`). Если не пользоваться тэгом `<BASEFONT face = "type1, type1" size = XX COLOR = xx>` для задания определенного размера шрифта на всей странице, то по умолчанию принимается `SIZE = 3`.

COLOR = xx — позволяет задавать цвет шрифта в шестнадцатеричном формате RGB, например `COLOR = "#00cc00"`, либо посредством стандартного имени, например `COLOR = "red"`.

Пример:

```
<FONT FACE="Arial Cyr, Arial, Geneva" SIZE="3" COLOR="#FF9900">
```

Контрольные вопросы

1. Указать тэг, позволяющий вывести текст шрифтом Arial или Geneva:
 - а) ``;
 - б) ``;
 - в) ``;
 - г) все перечисленные тэги ошибочны.
2. Указать тэг, позволяющий вывести текст шрифтом, кегль которого больше стандартного на 3 пункта:
 - а) ``;
 - б) ``;
 - в) ``;
 - г) Все перечисленные тэги ошибочны.

3. Как задать красный цвет ограниченного фрагмента текста?

- а) ``
фрагмент текста ``;
- б) ``
фрагмент текста ``;
- в) `` фрагмент текста ``;
- г) все перечисленные тэги ошибочны.

4. Как задать красный цвет всему последующему тексту?

- а) `<BASEFONT color="#ff0000" size="+3">`
фрагмент текста
``;
- б) `<BASEFONT color="#0000ff" size="3">`
фрагмент текста
``;
- в) `<FONTBASE "red" size="+3">`
фрагмент текста
``;
- г) все перечисленные тэги ошибочны.

■ ВСТАВКА ИЗОБРАЖЕНИЙ В ДОКУМЕНТ

Изображение позволяет лучше передать суть документа — один хороший рисунок ценнее тысячи слов. Для вставки изображения следует воспользоваться тэгом . Графический файл должен быть записан в формате GIF (Graphical Interchange Format) или JPG (Joint Photographic Experts Group). Используя изображения, следует учитывать некоторые особенности растрового формата:

- файлы растровых форматов медленно передаются;
- поисковые системы не работают с изображениями;
- пользователь может отключить графический интерфейс;
- возможны искажения при переносе цветовой гаммы.

Синтаксис тэга :

```
<IMG SRC="filename"
  align=AL width=XX height=XX
  border=XX vspase =XX hspace = XX alt="XXXX">
```

где filename — URL изображения в формате *.gif либо *.jpg; AL — top, middle, bottom — выравнивания изображения относительно базовой линии текста; left, right — задание правил обтекания текстом изображения; width=XX, height=XX — указание размеров выводимого изображения в пикселах; border — указание размеров рамки; vspase, hspace — расстояние от изображения до текста; alt — альтернативное описание изображения.

Контрольные вопросы

1. Какой тэг позволяет вывести изображение "test" в окно браузера?
 - а) ``;
 - б) ``;
 - в) ``;
 - г) все перечисленные тэги ошибочны.
2. Каким образом изображение размером 30X30 пикселей вывести в окно браузера как изображение 60X60 пикселей?
 - а) `vspase=60, hspace=60;`
 - б) `width=60, height=60;`
 - в) `align left=60, align right=60;`
 - г) все перечисленные тэги ошибочны.
3. Как задать альтернативное описание изображения?
 - а) `alt=" описание рисунка ";`
 - б) `src=" описание рисунка";`
 - в) `align=" описание рисунка";`
 - г) все перечисленные тэги ошибочны.
4. Как указать расстояние от изображения до обтекающего его текста?
 - а) `vspase=60, hspace=60;`
 - б) `width=60, height=60;`
 - в) `align left=60, align right=60;`
 - г) все перечисленные тэги ошибочны.
5. Какой атрибут тэга позволяет задать рамку?
 - а) title;
 - б) border;
 - в) bgcolor;
 - г) все перечисленные атрибуты ошибочны.

Задачи**Задача 1**

Разместить изображение на странице в начале текстовой строки.

```
<P>
  <IMG src = "1.gif">
    text text text text text text
</P>
```

Задача 2

Разместить изображение на странице таким образом, чтобы текстовая строка выравнивалась по центру изображения.

```
<P>
  <IMG src = "1.gif" align=middle>
    text text text text text text
</P>
```

Задача 3

Разместить изображение на странице таким образом, чтобы текст обтекал изображение, расположенное в правой стороне окна браузера. Указать размер выводимого изображения 20X20 пикселей.

```
<P>
  <IMG src = "1.jpg" align=right width=20 height=20>
    text text text text text text
    text text text text text text
    text text text text text text
    text text text text text text
    text text text text text text
</P>
```

Задача 4

Дополнить задачу 3. Поместить изображение в рамку шириной 3 пикселя, добавить альтернативное описание изображения и отделить изображение от текста на расстояние 20 пикселей.

```
<P>
  <IMG src = "1.jpg" align=right
    width=20 height=20 border=3
    alt="Пример обтекания текстом изображения">
```

```
vspace=20 hspace=20>
  text text text text text text text
  text text text text text text text
  text text text text text text text
  text text text text text text text
  text text text text text text text
</P>
```

■ ССЫЛКИ

Гипертекстовый документ — это документ, содержащий ссылки на другие документы. Доступ к ним реализуется активизацией гиперссылок. Ссылка состоит из двух частей: указательной (Anchor) и адресной (URL-адрес).

<A>

Синтаксис тега <A>:

```
<A HREF="адресная часть" title = "text" TARGET=name>
  указательная часть
</A>
```

Указатель — слово, группа слов или изображение. Адрес может быть:

- абсолютным — "http://eivt. da. ru";
- относительным — "... /namedir/filename. htm";
- внутренним для текущего документа — "#namemet";
- внутренним для внешнего документа: "filename. htm #namemet".

Для формирования внутренней ссылки используется атрибут **NAME**.

Пример:

```
формирование внутренней ссылки
<A name="namemet">текст </A>
```

При наличии тега <BASE> относительный адрес определяется в тэге <BASE>. Также адрес может содержать e-mail. В этом случае "http://" заменяется на "mailto:".

Возможны следующие форматы адресов:

```
http://address, mailto:address, news:address, ftp://address,
  gopher://address, wais://address, telnet://address.
```

Атрибуты тэга <A>:

TITLE — текст, содержащийся в атрибуте <TITLE>, выводится при подведении курсора к тексту ссылки.

TARGET — атрибут, предназначенный для указания кадра, в котором должен быть размещен файл, указанный атрибутом HREF. Name — имя кадра в который происходит загрузка файла. Зарезервированны следующие имена кадров, в которые происходит загрузка:

- **_blank** — загружает указанный файл в новое окно без названия;
- **_self** — загружает указанный файл в кадр, откуда выполнен вызов;
- **_parent** — загружает указанный файл в старший (родительский) кадр;
- **_top** — загружает указанный файл в полное окно.

Пример:

ссылка с указателем в виде изображения pic.gif

Пример:

ссылка на электронную почту по адресу "art@art.com"

Пример:

загрузка файла в новое окно без названия
 новое окно

Контрольные вопросы

1. Указать тэг, позволяющий организовать ссылку на документ "home.htm":
 - а) home.htm ;
 - б) anchor.htm ;
 - в) anchor.htm ;
 - г) все перечисленные тэги ошибочны.
2. Каким образом можно послать электронную почту?
 - а) контакт ;
 - б) контакт ;
 - в) контакт ;
 - г) все перечисленные тэги ошибочны.

3. Как задать указатель ссылки в виде рисунка?
 - а) <href="pic.gif"> ;
 - б) ;
 - в) <igm href="anchor.htm"><src="pic.gif"> ;
 - г) все перечисленные тэги ошибочны.
4. Указать тэг с правильным синтаксисом:
 - а) ;
 - б) ;
 - в) ;
 - г) все перечисленные тэги ошибочны.
5. Каким образом можно сформировать внутреннюю ссылку для данного документа?
 - а) текст ;
 - б) текст ;
 - в) <a name:cp1>текст ;
 - г) все перечисленные тэги ошибочны.
6. Указать тэг, позволяющий организовать ссылку на внутреннюю метку (cp1) документа "home.htm".
 - а)

Глава 1

 ;
 - б) ;
 - в) ;
 - г) все перечисленные тэги ошибочны.
7. Каким образом можно открыть новое окно с документом home.htm?
 - а) home.htm ;
 - б) anchor.htm ;
 - в) anchor.htm ;
 - г) все перечисленные тэги ошибочны.
8. Как загрузить документ home.htm в кадр с именем "WV"?
 - а) home.htm ;
 - б) anchor.htm ;
 - в) anchor.htm ;
 - г) все перечисленные тэги ошибочны.

■ ИЗОБРАЖЕНИЯ-КАРТЫ, ОБСЛУЖИВАЕМЫЕ БРАУЗЕРОМ

Изображение-карта — это изображение, разные части которого связаны ссылками с различными URL-адресами.

<MAP>

Ссылка на изображение-карту:

```
<IMG src=my_map.gif USEMAP=#my_map>
```

USEMAP — атрибут, указывающий имя контейнера MAP, в котором определяется соответствие между областями изображения и ссылками.

Синтаксис тега <MAP>:

```
<map name="my_map">
```

```
  <area share="sh" coord="x,y,..." href="url-адрес">
```

```
</map>
```

Атрибуты тега <MAP>:

MY_MAP — имя контейнера MAP, использующееся в тэге .

SHARE — определяет форму чувствительной области. Имеет значения: rect, poly, circle, default.

COORDS — содержит координаты чувствительной области.

Пример:

```
<map name="my_map">
```

```
  <area share="rect" coord="0,0,10,10" href="test1.htm">
```

```
  <area share="rect" coord="30,30,50,50" href="test2.htm">
```

```
</map>
```

```
<img src=my_map.gif USEMAP=#my_map>
```

Контрольные вопросы

1. Указать тэг, позволяющий организовать изображение-карту:
 - а) ;
 - б) ;
 - в) ;
 - г) все перечисленные тэги ошибочны.
2. Как указать координаты области изображения, связанной со ссылкой?

- а) <area share="rect" coord="0,0,10,10" href="test1.htm">;
 - б) <area share="rect" left="10", right="20", href="t.htm">;
 - в) <area share="circle" coord="10,10" href="test1.htm">;
 - г) все перечисленные тэги ошибочны.
3. Как сформировать внутреннюю ссылку во внешнем документе, используя технологию изображения-карты?
 - а) <area share="rect";
 coord="0,0,10,10" href="http://s.ru/test1.htm #cp1">;
 - б) <area share="#cp1" coord="0,0,10,10" href="test1.htm ">;
 - в) <area share="rect" coord="0,10,10" href="test1.htm #cp1">;
 - г) все перечисленные тэги ошибочны.

■ СПИСКИ

Упорядоченный (нумерованный) список

Списки — один из наиболее легких для восприятия форматов. Упорядоченный список используется для нумерованного перечисления отдельных пунктов.

Упорядоченный список открывается тэгом .

Пример:

```
<OL type=1>
```

```
  <LI> Заголовок упорядоченного списка<BR>
```

```
    <LI>Первый элемент списка
```

```
    <LI>Второй элемент списка
```

```
</OL>
```

Синтаксис тега :

```
<OL type=XX start=n>
```

- xx= A — устанавливает маркер в виде прописных букв;
 a — устанавливает маркер в виде строчных букв;
 I — устанавливает маркер в виде заглавных римских цифр;
 i — устанавливает маркер в виде строчных римских цифр;
 1 — устанавливает маркер в виде арабских цифр;
 n — устанавливает начальный маркер в текущем списке.

Неупорядоченный (маркированный) список****

Неупорядоченный список открывается тэгом .

Пример:

```
<UL type=square>
  <LH>Заголовок упорядоченного списка </LH>
  <LI>Первый элемент списка
  <LI>Второй элемент списка
</UL>
```

Синтаксис тэга :

```
<UL type=XX>
```

где xx = disc (точка), square (квадрат), circle (окружность).

Контрольные вопросы

- Указать тэг, позволяющий организовать упорядоченный список с маркерами в виде строчных букв:
 - <ul type=square>;
 - <ol type=1>;
 - <ol type=a>;
 - все перечисленные тэги ошибочны.
- Указать тэг, позволяющий организовать упорядоченный список с маркерами в виде арабских цифр:
 - <ul type=square>;
 - <ol type=1>;
 - <ol type=a>;
 - все перечисленные тэги ошибочны.
- Указать тэг, позволяющий организовать неупорядоченный список с маркерами в виде точек:
 - <ul type=disc>;
 - <ul type=square>;
 - <ol type=i>;
 - все перечисленные тэги ошибочны.
- Как задать заголовок списка?
 - <href="pic. gif"> ;
 - ;
 - <src="pic. gif"> ;

- все перечисленные тэги ошибочны.
- Указать тэг с правильным синтаксисом:
 - ;
 - ;
 - ;
 - все перечисленные тэги ошибочны.
 - Как сформировать внутреннюю ссылку для данного документа?
 - текст ;
 - текст ;
 - <a name:cp1>текст ;
 - все перечисленные тэги ошибочны.
 - Указать тэг, позволяющий организовать ссылку на внутреннюю метку (cp1) документа "home. htm":
 -

Глава 1

 ;
 -

Глава 1

 ;
 - ;
 - все перечисленные тэги ошибочны.

■ ФРЕЙМЫ

Технология фреймов в HTML позволяет делить главное окно браузера на несколько окон и в каждом из них просматривать свой документ. Один фрейм (окно) отображает только один документ.

Создание фрейма**<FRAMESET>**

Создание фреймов осуществляется с помощью тэга <FRAMESET>:

```
<FRAMESET>...</FRAMESET>
```

Внимание! В HTML-странице, использующей контейнер FRAMESET, контейнер BODY не используется.

В контейнер `<NOFRAMES>...</NOFRAMES>` заносится текст, который будет выполняться, если браузер пользователя не поддерживает фреймы.

Свойства фреймов

Синтаксис тэга `<FRAMESET>`:

```
<FRAMESET XXXX="N1,N2,...">
```

Данный тэг позволяет определить количество фреймов и задать их размер в процентах от размера окна браузера или зафиксировать эти размеры в пикселах.

Для разделения экрана на вертикальные фреймы необходимо вместо XXXX подставить COLS, на горизонтальные фреймы — вместо XXXX подставить ROWS.

Размеры фреймов

Пример

```
<FRAMESET ROWS="50%,50%">
```

где "50%,50%" — размер полученных окон в процентах от размера окна браузера. "50%,*" — размер окон, отмеченных как "*", определяется так, чтобы в сумме все окна составляли 100% от размера окна браузера. Размер фрейма можно задавать и в пикселах, для этого достаточно после числа не ставить символ %.

<FRAME>

Синтаксис тэга `<FRAME>`:

```
<FRAME NAME="framename"
```

```
MARGINWIDTH=X MARGINHEIGHT=X
```

```
SCROLLING=XX
```

```
NORESIZE SRC="filename">
```

Атрибуты тэга `<FRAME>`:

FILENAME имя фрейма

MARGINWIDTH=X горизонтальный отступ

MARGINHEIGHT=X вертикальный отступ между фреймом и его границей(1...6)

SCROLLING=XX прокрутка фрейма.

Атрибут `SCROLLING=YES` позволяет создать полосы прокрутки; `SCROLLING=NO`

указывает браузеру, что полосы прокрутки в данном фрейме отсутствуют, а атрибут `SCROLLING=AUTO` позволяет отображать полосы прокрутки в зависимости от свойств браузера

NORESIZE

фиксированный размер фрейма (во время просмотра изменить размер фрейма нельзя)

SRC="filename"

задает имя файла, содержащего просматриваемый документ

Пример 1:

```
<html>
  <frameset rows="50%,50%">
    <frame src="page1.htm" name="menu">
    <frame src="page2.htm" name="main">
  </frameset>
</html>
```

В данном примере окно браузера разделяется горизонтально на два окна и в каждом из этих окон формируется своя HTML-страница (размеры окон указаны в процентах от размера окна браузера).

Для разделения экрана вертикально вместо `<FRAMESET rows=...>` надо указать `<FRAMESET cols=...>`.

Пример 2:

```
<FRAMESET COLS="136,*">
  <FRAMESET ROWS="197,*">
    <FRAME SRC="1.html" NAME="fr1">
    <FRAME SRC="2.html" NAME="fr2">
  </FRAMESET>
  <FRAMESET ROWS="100,*">
    <FRAME SRC="3.html" NAME="fr3">
    <FRAME SRC="4.html" NAME="fr4">
  </FRAMESET>
</FRAMESET>
```

В данном примере окно браузера разделяется на четыре фрейма.

Пример 3:

Окно браузера делится на два фрейма (окна) и в них загружаются какие-либо документы (см. пример 1): в окно 1 — page1.html, в окно 2 — page2.html.

Для того, чтобы при нажатии на ссылку в фрейме 1 (окно 1) страница загружалась в фрейм 2 (окно 2), при указании ссылки необходимо использовать атрибут TARGET="имя, присвоенное данному фрейму" (в примере 1 имя фрейма 2 — "main"), т. е. в файле page1.html необходимо написать:

```
<A href="nextpage.html" Target="main">
Загрузка в фрейм 2 страницы nextpage.html </A>
```

Контрольные вопросы

- Как задать отступы между фреймами 3 пикселя по горизонтали и 4 пикселя по вертикали?
 - Cols="3,4";
 - Rows="3,4";
 - Rows="4,3";
 - MARGINWIDTH=3 MARGINHEIGHT=4;
 - MARGINHEIGHT=3 MARGINWIDTH=4.
- Какая команда запрещает прокрутку фрейма?
 - NORESIZE;
 - SCROLLING="NO";
 - RESIZENO;
 - NOSCROLLING.
- Как указать HTML-страницу, загружаемую во фрейм?
 - <FRAME SRC="page.html">;
 - <FRAME NAME="page.html">;
 - <FRAME SCR="page.html">;
 - ни один из вариантов не верен.
- Как задать размеры фреймов в процентах?
 - <FRAMESET ROWS="50,50%">;
 - <FRAMESET COLS=%"50,50">;
 - <FRAMESET COLS="50%,50%">;
 - ни один из вариантов не верен.
- Укажите тэг, который разделяет окно браузера на два вертикальных фрейма:

- <FRAMESET COLS="136,234">;
 - <FRAMESET ROWS="136,234">;
 - <FRAMESET COLS="136;234">;
 - ни один из вариантов не верен.
- Как можно указать, в какой фрейм следует загружать страницу?
 - SRC="FrameName";
 - SCR="FrameName";
 - TARGET="FrameName";
 - ни один из вариантов не верен.
 - Какой контейнер нельзя использовать в странице, содержащей <FRAMESET></FRAMESET>?
 - <TITLE>...</TITLE>;
 - <BODY>...</BODY>;
 - <NOFRAMES>...</NOFRAMES>;
 - ни один из вариантов не верен.
 - Какая команда запрещает изменение размеров фрейма при просмотре их в браузере?
 - NORESIZE;
 - SCROLLING="NO";
 - RESIZENO;
 - NOSCROLLING.
 - Как задать размеры двух фреймов так, чтобы первый фрейм занимал 20%, а второй — оставшуюся часть от окна браузера?
 - <FRAMESET ROWS="20,+"%>;
 - <FRAMESET ROWS="20,*"%>;
 - <FRAMESET ROWS="20%,+">;
 - <FRAMESET ROWS="20%,*"%>;

Задачи

Задача 1

Разделить окно браузера горизонтально на два одинаковых окна и в каждое из этих окон загрузить свою html-страницу (размеры окон указать в процентах от окна браузера). Повторить решение задачи для вертикального деления окна.

```
<HTML>
<FRAMESET rows="50%,50%"
<FRAME src="page1.htm">
```

```
<FRAME src="page2.htm">
```

```
</FRAMESET>
```

```
</HTML>
```

Для разделения экрана вертикально вместо `<FRAMESET rows=...>` надо указать `<FRAMESET cols=...>`.

Задача 2

Разделить окно браузера на четыре одинаковых фрейма.

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

```
<FRAMESET ROWS="50%,*">
```

```
<FRAME SRC="1.html" NAME="fr1">
```

```
<FRAME SRC="2.html" NAME="fr2">
```

```
</FRAMESET>
```

```
<FRAMESET ROWS="50%,*">
```

```
<FRAME SRC="3.html" NAME="fr3">
```

```
<FRAME SRC="4.html" NAME="fr4">
```

```
</FRAMESET>
```

```
</FRAMESET>
```

Задача 3

Создать такой HTML-текст, чтобы при нажатии на ссылку в первом фрейме во второй фрейм грузилась какая-либо страница.

В файле `page1.html` необходимо написать:

```
<A href="nextpage.html" Target="main">
```

Загрузка во фрейм 2 страницы `nextpage.html`

```
</A>
```

Задача 4

Разделить окно на два вертикальных фрейма, первый из них разделить на два горизонтальных.

```
<FRAMESET BORDER=1 COLS="136,*">
```

```
<FRAMESET BORDER=1 ROWS="197,*">
```

```
<FRAME SCROLLING="AUTO"
```

```
SRC="menu.html" NAME="menu" BORDER=1>
```

```
<FRAME SCROLLING="AUTO"
```

```
SRC="other.html" NAME="other" BORDER=1>
```

```
</FRAMESET>
```

```
<FRAME SCROLLING="AUTO"
```

```
SRC="main.html" NAME="main" BORDER=1>
```

```
</FRAMESET>
```

Задача 5

Дополнить решение задачи 4 так, чтобы отступ между фреймами был 3 пикселя и чтобы размеры фреймов во время просмотра нельзя было менять.

Прописать свойства `MARGINWIDTH=3` `MARGINHEIGHT=3` `NORESIZE` для каждого фрейма.

Задача 6

Дополнить решение задачи 4 так, чтобы полоса прокрутки присутствовала только в фрейме с именем `main`.

Прописать свойство `SCROLLING="NO"` для фреймов с именами `"menu"` и `"other"` и свойство `SCROLLING="YES"` для фрейма с именем `"main"`.

■ ФОРМЫ

Форма (`form`) — это область HTML-документа, которая задается с помощью пары тэгов: `<FORM>` и `</FORM>`.

Интерактивные формы позволяют читателям Web-страниц связываться с их владельцами. Простота такой связи основывается на использовании атрибута `<mailto:e-mail>`, позволяющего отослать содержание формы (например, текстовые области) на электронный адрес владельца сайта.

Многие тэги HTML, например тэги, определяющие поля ввода (`textfield`), области текста (`textarea`), контрольные переключатели (`checkbox`), селекторные кнопки (`radiobutton`) и списки (`selectionlist`), должны располагаться только внутри контейнера `<FORM>...</FORM>`. Подробнее эти тэги описаны далее.

Синтаксис тэга `<FORM>`:

```
<FORM name = "formName"
target = "windowName"
action = "serverURL"
method = "get" | "post"
enctype = "encodingType"
[onSubmit = "handlerText"]>
</form>
```

Рассмотрим атрибуты более подробно.

NAME — задает имя формы для последующего обращения к ней по этому имени (т. е. это имя может быть использовано для

обращения к данной форме без массива forms). Например, если создать следующую форму:

```
<form name = "myform">...</form>
```

то обратиться к ней теперь можно двумя способами. Первый способ: `document.form[i]`, где `i` — номер формы в документе. Второй способ: `document.myform`. Оба эти способа равнозначны, но второй более нагляден (данные обращения используются для обработки форм языком JavaScript).

TARGET — задает имя окна, в котором должны обрабатываться события, связанные с изменением элементов формы. Естественно для этого требуется наличие окна или фрейма с данным именем. В качестве значений данного атрибута можно указывать зарезервированные слова, такие как `top`, `self`, `parent` и `blank`.

ACTION — задает адрес URL сервера, который будет получать данные из формы и запускать соответствующий CGI-скрипт. Значением данного атрибута можно взять `mailto:...`, тогда данные из формы будут отсылаться на заданный электронный почтовый адрес.

METHOD — определяет метод передачи формы на сервер. Значениями данного атрибута могут быть методы `GET` и `POST`. При использовании метода `POST` информация из формы посылается отдельным потоком. При приеме данных на сервере CGI-скрипт считывает эти данные из стандартного входного потока. Метод `GET` просто присоединяет данные из формы к строке адреса URL, заданной в атрибуте `action`.

ENCTYPE — задает тип кодировки MIME (Multimedia Internet Mail Extensios) для посылаемых данных.

В формах можно использовать только один тип передаваемой информации — `"text/plain"`. Типы передаваемой информации более подробно рассмотрены в разделе "Способы передачи файлов в почтовом сообщении".

Рассмотрим тэги, используемые в контейнере `<FORM>...</FORM>`

<TEXTFIELD>

Тэг `<TEXTFIELD>` задает строку ввода. Применяется для ввода буквенно-цифровой информации.

Синтаксис:

```
<input name="textname" [type="text"] > [text to display]
```

<TEXTAREA>

Тэг `<TEXTAREA>` задает буквенно-цифровое поле из нескольких строк.

Синтаксис:

```
<textarea name="textareaname" rows='integer' cols='integer',
 [text to display] [wrap='hard | soft']
</textarea>
```

<CHECKBOX>

Тэг `<CHECKBOX>` задает контрольный переключатель (кнопку, которую можно установить в одно из двух состояний: включено или выключено (`on/off`)).

Синтаксис:

```
<input name="checkboxname" type="checkbox"
 value="checkboxvalue" [checked]> [text to display]
```

<RADIOBUTTON>

Тэг `<RADIOBUTTON>` задает селекторную кнопку.

Синтаксис:

```
<input name="radioname" type="radio"
 value="buttonvalue" [checked]> text to display
```

В тэге `<RADIOBUTTON>` для того, чтобы выбрать только один переключатель или одну кнопку, соответственно, нужно использовать одно и то же значение параметра `"name"`.

Пример:

```
<input type='radio' name='mycheck'>
<input type='radio' name='mycheck'>
```

Для того чтобы отослать заполненную форму на сервер (адрес которого задан в атрибуте `action`), используется кнопка `Submit`. Синтаксис данной кнопки следующий:

```
<input type='submit' value='Send'>
```

где параметр `value` задает текст, который будет отображен на кнопке.

Если в форме предусматривается возможность очистки всех полей ввода, то в код формы необходимо добавить кнопку Reset. Ее синтаксис:

```
<input type='reset' value='Reset'>
```

Задачи

Задача 1

Создать форму, состоящую из полей для ввода фамилии, имени и двух контрольных переключателей для выбора пола.

```
<form>
  <input type='textfield'> Введите вашу фамилию <br>
  <input type='textfield'> Введите ваше имя <br>
  <input type='radio' name='check'> Муж. <br>
  <input type='radio' name='check'> Жен.
</form>
```

Задача 2

Усовершенствовать предыдущую форму так, чтобы она отсылалась на ваш e-mail.

```
<form action='mailto:email' enctype='text/plain'>
  <input type='textfield'> Введите вашу фамилию <br>
  <input type='textfield'> Введите ваше имя <br>
  <input type='radio' name='check'> Муж. <br>
  <input type='radio' name='check'> Жен.
  <input type="submit" name="Submit" value="Послать">
  <input type="reset" name="Reset" value="Очистить">
</form>
```

Контрольные вопросы

- Как отправить форму на адрес электронной почты?
 - <form mailto='email'>...</form>;
 - <form action='email'>...</form>;
 - <form action='mailto:e-mail'>...</form>;
 - ни один из вариантов не верен.
- Если задана форма <form name='form1'>...</form>, то к ней можно обратиться:
 - только как document.form[i];
 - только как document.form1;

- можно и так, и так;
 - ни один из вариантов не верен.
- Если в контейнере <form>...</form> задать тэг <input name='text1'>, будет выведено:
 - сообщение об ошибке;
 - буквенно-цифровая строка;
 - буквенно-цифровое поле стандартного размера;
 - ни один из вариантов не верен.
 - Для получения селекторной кнопки надо выбрать тип:
 - radio;
 - button;
 - radiobutton;
 - ни один из вариантов не верен.
 - При каком значении атрибута method посылаемые формой данные будут передаваться отдельным потоком?
 - GET;
 - POST;
 - FLOW;
 - ни один из вариантов не верен.
 - Если задать атрибут type равным textarea, то в окне браузера:
 - будет отображена текстовая строка;
 - будет отображено текстовое поле;
 - будет выведено сообщение об ошибке;
 - ни один из вариантов не верен.
 - Что задается атрибутом action?
 - действие, которое будет производить форма при нажатии кнопки отсылки;
 - такого атрибута не существует;
 - определяется адрес сервера, на который будет отсылаться форма;
 - ни один из вариантов не верен.

■ ТАБЛИЦЫ

Таблицы используются для повышения удобочитаемости документов. Они создаются с помощью контейнера TABLE, который не имеет обязательных атрибутов. Если атрибуты не заданы, то разметка таблицы осуществляется автоматически.

<TABLE>

Атрибуты тега **<TABLE>**:

BORDER — задает толщину рамки таблицы в пикселях.

FRAME — определяет, какие границы таблицы будут видны (Explorer 5 и выше). Возможны следующие значения этого атрибута:

- **VOID** — все границы невидимы (по умолчанию);
- **ABOVE** — только верхняя;
- **BELOW** — только нижняя;
- **HSIDES** — только верхняя и нижняя;
- **VSIDES** — только правая и левая;
- **LHS** — только левая;
- **RHS** — только правая;
- **BOX** — все границы ячеек видимы;
- **BORDER** — все внешние границы таблицы видимы.

CELLPADDING — задает (в пикселях) промежуток между содержимым ячейки и ее границами.

CELLSPACING — задает (в пикселях) промежуток между границами соседних ячеек, между общей границей таблицы и границами крайних ячеек.

RULES — определяет область таблицы, в пределах которой действуют заданные атрибуты (Explorer 4 и выше). Возможные значения:

- **NONE** — правил нет (по умолчанию), т. е. для каждой ячейки таблицы действуют только свои атрибуты;
- **GROUPS** — правила определены между группами столбцов и строк;
- **ROWS** — правила определены только между строками;
- **COLS** — правила определены только между столбцами;
- **ALL** — правила определены между всеми строками и столбцами.

SUMMARY — невидимый заголовок таблицы (используется некоторыми программами).

WIDTH — задает ширину таблицы в пикселях или в процентах от ширины экрана.

HEIGHT — задает высоту таблицы в пикселях или в процентах от ширины экрана.

В контейнер **TABLE** могут входить следующие теги.

<CAPTION>

<CAPTION> — заголовок таблицы, содержит только текст и имеет единственный атрибут **ALIGN**.

Атрибут **ALIGN** может принимать значения:

- **TOP** — заголовок наверху;
- **BOTTOM** — заголовок внизу;
- **LEFT** — заголовок слева;
- **RIGHT** — заголовок справа.

<TR>

<TR> — тег начала строки, имеет два атрибута:

ALIGN — выравнивает содержимое ячейки по горизонтали и может принимать значения:

- **LEFT** — для выравнивания по левой границе ячейки;
- **CENTER** — для выравнивания по центру ячейки;
- **RIGHT** — для выравнивания по правой границе ячейки;
- **JUSTIFY** — для выравнивания по ширине ячейки;
- **CHAR** — для выравнивания относительно символа

и **VALIGN** — выравнивает содержимое ячейки по вертикали и может принимать значения:

- **TOP** — для прижатия к верхней границе ячейки;
- **MIDDLE** — для центрирования по высоте ячейки;
- **BOTTOM** — для прижатия к нижней границе ячейки;
- **BASELINE** — для выравнивания по базовой линии ячейки.

<TD>, <TH>

<TD>, **<TH>** — теги разметки таблицы на уровне ячеек.

<TH> используется для создания заголовков строк и столбцов, а **<TD>** — для создания ячеек с данными. Эти теги имеют следующие атрибуты:

ABBR — задает аббревиатуру заголовка.

ALIGN — выравнивает данные в ячейке по горизонтали.

COLSPAN — задает число колонок, которые занимает ячейка.

HEADERS — создает список идентификационных текстов (**id**) для заголовков.

HEIGHT — задает высоту ячейки в пикселях.

— **NOWRAP** — запрещает автоматический перенос слов в ячейке.

ROWSPAN — задает число строк, которые занимает ячейка.

— **SCOPE** — задает область, занимаемую ячейками заголовка.

VALIGN — выравнивает данные в ячейке по вертикали.

WIDTH — задает ширину ячейки в пикселях.

BGCOLOR — задает цвет фона ячейки таблицы.

<THEAD>, <TFOOT>, <TBODY>

Строки таблицы могут быть сгруппированы в заголовки, колонтитулы либо в одну секцию или более. Это осуществляется с помощью элементов <THEAD>, <TFOOT> и <TBODY> соответственно. <TFOOT> должен быть определен до <TBODY>.

<COLGROUP>

<COLGROUP> — тэг группировки столбцов таблицы, имеющий два атрибута:

SPAN — число столбцов в группе;

WIDTH — ширина столбцов группы.

<COL>

Отдельный столбец группы задается тэгом <COL>, имеющим те же атрибуты, что и <COLGROUP>.

Контрольные вопросы

1. Какие атрибуты тэга <TABLE> являются обязательными?
 - а) ALIGN;
 - б) никакие;
 - в) TR;
 - г) TH.
2. С помощью каких тэгов осуществляется деление таблицы на строки?
 - а) TR;
 - б) TH;
 - в) TH и TD;
 - г) TD.

3. Как прижать текст к нижней границе ячейки?
 - а) ALIGN=RIGHT;
 - б) VALIGN=BOTTOM;
 - в) ALIGN=BOTTOM;
 - г) VALIGN=TOP.
4. В каких единицах задается ширина таблицы?
 - а) в пикселях;
 - б) в процентах;
 - в) в пикселях или в процентах;
 - г) в числе ячеек фиксированной ширины.
5. Значением какого атрибута табличного элемента является JUSTIFY?
 - а) SCOPE;
 - б) SPAN;
 - в) ALIGN;
 - г) WIDTH.
6. Какой атрибут тэга <TABLE> является чувствительным к регистру?
 - а) width;
 - б) height;
 - в) cellspasing;
 - г) summary.
7. Найдите ошибку:
 - а) ALIGN=LEFT;
 - б) CELLPADDING=2;
 - в) VALIGN=CHAR;
 - г) height=5.
8. Что делает элемент <THEAD>?
 - а) Формирует заголовок таблицы;
 - б) формирует заголовок строки;
 - в) группирует столбцы;
 - г) группирует строки в заголовки.
9. Какой элемент группирует столбцы?
 - а) COL;
 - б) COLS;
 - в) COLGROUP;
 - г) COLGROUPS.

10. Как создать эффект слияния соседних ячеек строки?

- + а) COLSPAN=X;
- б) ROWSPAN=X;
- в) COL=X;
- г) ROWS=X.

Задачи

Задача 1

Построить таблицу 2X2, заполнить и озаглавить ее.

```
<TABLE HEIGHT=50% WIDTH=50%>
  <CAPTION>
    <H3>
      Пример таблицы 2X2
    </H3>
  </CAPTION>
  <TR ALIGN=CENTER>
    <TD>Text1</TD>
    <TD>Text2</TD>
  </TR>
  <TR ALIGN=CENTER>
    <TD>Text3</TD>
    <TD>Text4</TD>
  </TR>
</TABLE>
```

Задача 2

Перенести заголовок таблицы вниз, задать толщину рамки в 3 пикселя.

```
<TABLE HEIGHT=50% WIDTH=50% BORDER=3>
  <CAPTION ALIGN=BOTTOM>
    <H3>Пример таблицы 2X2</H3>
  </CAPTION>
  <TR ALIGN=CENTER>
    <TD>Text1</TD>
    <TD>Text2</TD>
  </TR>
  <TR ALIGN=CENTER>
    <TD>Text3</TD>
```

```
<TD>Text4</TD>
```

```
</TR>
```

```
</TABLE>
```

Задача 3

Выровнять данные, находящиеся в ячейках слева, к левому краю ячеек, а содержимое ячеек справа сделать ссылками на другие страницы.

```
<TABLE HEIGHT=50% WIDTH=50% BORDER=3>
  <CAPTION ALIGN=BOTTOM>
    <H3>Пример таблицы 2X2</H3>
  </CAPTION>
  <TR ALIGN=CENTER>
    <TD ALIGN=LEFT>Text1</TD>
    <TD>
      <A HREF="a.htm">Text2</A>
    </TD>
  </TR>
  <TR ALIGN=CENTER>
    <TD ALIGN=LEFT>
      Text3
    </TD>
    <TD>
      <A HREF="b.htm">Text4</A>
    </TD>
  </TR>
</TABLE>
```

Задача 4

Выровнять картинку по центру с помощью таблицы.

```
<TABLE BORDER=2 WIDTH="80%" HEIGHT="80%"
  BGCOLOR="#FFCCCC">
  <CAPTION>
    Пример использования таблиц
  </CAPTION>
  <TR ALIGN=CENTER VALIGN=CENTER>
    <TD>
      <IMG SRC="mf3.gif">
    </TD>
  </TR>
</TABLE>
```

Задача 5

Создать таблицу, имеющую объединенные ячейки, цветной бордюр, заголовки и данные. Границы ячеек не должны совпадать с границами таблицы. Задать цвета столбцов и строк.

```
<CENTER>
<TABLE frame="box" cellspacing="1" cellpadding="1"
border="2" width="80%" height="80%"
bordercolor="crimson"
rules="all" summary="более сложный пример">
<caption>
  <h3>Пример использования таблиц</h3>
</caption>
<!-- задаются свойства первого столбца (Explorer 4 выше) -->
  <colgroup bgcolor="whitesmoke">
<!-- задаются свойства второго столбца (Explorer 4 выше) -->
  <colgroup align="center" bgcolor="skyblue">
<!-- задаются свойства третьего столбца (Explorer 4 выше) -->
  <colgroup align="center" bgcolor="lightpink">
<tr>
  <th rowspan="2">Вид расчета</th>
  <th colspan="2" bgcolor="linen">Наименование</th>
</tr>
<tr>
  <th>Все про TeX, Дональд Кнут</th>
  <th>Язык программирования Java, Кен Арнольд,
    Джеймс Гослинг
  </th>
</tr>
<tr>
  <th>Наличный</th>
  <td>45 руб.</td>
  <td>70 руб.</td>
</tr>
<tr>
  <th>Безналичный</th>
  <td>40 руб.</td>
  <td>64 руб.</td>
```

```
</tr>
  <!-- закрытие группы столбцов -->
  </colgroup>
  </colgroup>
  </colgroup>
</table>
</center>
```

■ ТЭГИ ФОРМАТИРОВАНИЯ ТЕКСТА

При форматировании HTML-документа часто используют тэги <P>, <HR>,
, <BLOCKQUOTE>.

<P>

<P> — тэг начала абзаца.

<HR>

<HR> — тэг горизонтальной линейки (линии). Закрывающего тэга нет.

Возможные атрибуты:

ALIGN — выравнивает линейку по горизонтали. Значения: LEFT, RIGHT, CENTER, JUSTIFY.

NOSHADE — отключает оттенки у линейки (линейка становится одноцветной).

SIZE — задает толщину линейки в пикселах.

WIDTH — задает ширину линейки в пикселах или в процентах.

 — тэг перевода строки. Закрывающего тэга нет. Может иметь атрибут **CLEAR**, управляющий текстовым потоком и принимающий одно из значений:

- LEFT сдвигает текст к левому краю;
- RIGHT сдвигает текст к правому краю;
- ALL сдвигает текст к левому и правому краям одновременно;
- NONE значение по умолчанию (сдвига текста нет).

<BLOCKQUOTE>

<BLOCKQUOTE> — тэг блока цитирования. Представляется в виде отдельного абзаца, не имеет атрибутов.

Контрольные вопросы

- С помощью какого тэга можно прервать строку и перейти к следующей?
 - HR;
 - BR;
 - NOSHADE;
 - CLEAR.
- Что делает атрибут SIZE тэга <HR>?
 - Задаёт границы параграфа;
 - задаёт длину горизонтальной линии;
 - задаёт толщину горизонтальной линии;
 - задаёт размер текста под горизонтальной линией.
- Для чего используется тэг <BLOCKQUOTE>?
 - Для создания блока цитирования;
 - для изменения цвета текста;
 - для перевода строки;
 - для вставки комментариев;
- Какой из перечисленных тэгов должен иметь закрывающий тэг обязательно?
 - HR;
 - BR;
 - BLOCKQUOTE;
 - P.
- Какое значение атрибута CLEAR имеет тэг
 по умолчанию?
 - ALL;
 - NONE;
 - LEFT;
 - RIGHT.

Задачи**Задача 1**

Ниже приведен текст стихотворения Есенина. Набрать его в HTML без использования тэга <PRE>.

ПОБИРУШКА

Плачет девочка-малютка у окна больших хором,
А в хоромах смех веселый так и льется серебром.
Плачет девочка и стынет на ветру осенних гроз,
И ручонкою изябшей вытирает капли слез.

Со слезами она просит хлеба черствого кусок,
От обиды и волненья замирает голосок.
Но в хоромах этот голос заглушает шум утех,
И стоит малютка, плачет под веселый, резвый смех.

(1915)

<P>

<CENTER> ПОБИРУШКА </CENTER>

 Плачет девочка-малютка у окна больших хором,

 А в хоромах смех веселый так и льется серебром.

 Плачет девочка и стынет на ветру осенних гроз,

 И ручонкою изябшей вытирает капли слез.

 Со слезами она просит хлеба черствого кусок,

 От обиды и волненья замирает голосок.

 Но в хоромах этот голос заглушает шум утех,

 И стоит малютка, плачет под веселый, резвый смех.

 И стоит малютка, плачет под веселый, резвый смех.

 <1915>
</P>

Задача 2

Привести цитату какого-нибудь известного деятеля науки. Набрать ту же цитату, но с использованием тэга <BLOCKQUOTE>. Отделить эти цитаты друг от друга горизонтальной чертой шириной в 70% от размера экрана.

Математику уже затем учить надо, что она ум в порядок приводит.

М. В. Ломоносов

<HR WIDTH=70% SIZE=2>

<BLOCKQUOTE>

Математику уже затем учить надо,
что она ум в порядок приводит.

<P ALIGN=RIGHT> М. В. Ломоносов

</BLOCKQUOTE>

■ СЛОИ

Слоем называется точно позиционированный прямоугольный блок объектов web-страницы.

Слои обладают рядом полезных свойств: они могут быть наложены друг на друга, позволяют перемещать объекты в пределах HTML-страниц (с помощью JavaScript), могут быть видимыми и невидимыми.

В данном разделе рассматривается часть свойств объекта Layer. Более полное рассмотрение требует использования JavaScript.

<LAYER>, <ILAYER>

Чтобы создать слой, следует использовать тэги <LAYER> или <ILAYER> со следующими атрибутами (NetScape):

NAME="layerName" — название слоя.

LEFT=xPosition — абсцисса левого верхнего угла.

TOP=yPosition — ордината левого верхнего угла.

Z-INDEX=layerIndex — номер индекса для слоя.

WIDTH=layerWidth — ширина слоя в пикселах или в процентах от ширины окна браузера.

CLIP="x1_offset, y1_offset, x2_offset, y2_offset" — задание видимой области слоя (рисунок 1).

ABOVE="layerName" — определение слоя под данным.

BELOW="layerName" — определение слоя над данным.

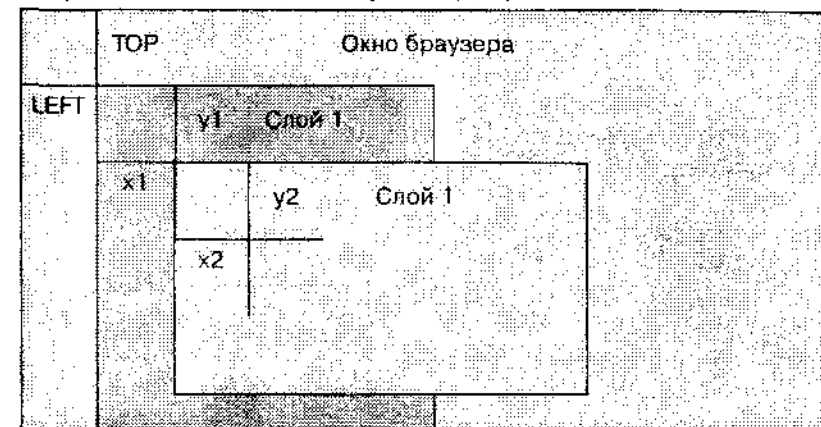
VISIBILITY=show | hide — видимость слоя.

BGCOLOR="rgbColor" — цвет фона (название стандартного цвета или RGB-запись).

BACKGROUND="imageURL" — фоновая картинка.

Тэг <LAYER> используется для тех слоев, которые требуется точно позиционировать с помощью параметров LEFT и TOP. Если

Координаты левого верхнего угла — (0; 0)



положение слоя не указывается, то по умолчанию он помещается в верхний левый угол окна, а при наличии текста — в текстовую строку, сразу за последним символом.

Тэг <ILAYER> создает слой, положение которого определяется при формировании документа.

Пример:

использование слоев (текст расположен поверх изображения)

```
<html>
  <layer name=pic z-index=10 left=222 top=111>
    
  </layer>
  <layer name=txt z-index=20 left=222 top=111>
    <font size=+3>
      Этот текст поверх изображения pic.gif
    </font>
  </layer>
</html>
```


Символ	Десятичный код	HTML код	Описание
"	"	"	Знак кавычек
&	&	&	Знак амперсента
<	<	<	Меньше
>	>	>	Больше
	 	 	Неразрывный пробел
	¦	¦	Разорванная вертикальная черта
©	©	©	Знак авторского права (копирайт)
®	®	®	Зарегистрированный товарный знак
°	°	°	Знак градуса
±	±	±	Плюс—минус
²	²	²	Вторая степень
³	³	³	Третья степень
´	´	´	Знак ударения
µ	µ	µ	Знак микро
¹	¹	¹	Первая степень
¼	¼	¼	Одна четвертая
½	½	½	Одна вторая
¾	¾	¾	Три четвертых
×	×	×	Знак умножения
÷	÷	÷	Знак "от ... до"

Кроме приведенных выше спецсимволов часто используется символ перевода строки — "\n"

Контрольные вопросы

- Какой спецсимвол отвечает за неразрывный пробел?
 -
 - ®
 - °
 - ни один из вышеперечисленных.
- Какой спецсимвол отвечает за знак умножения?
 - ¾
 - ×
 - ÷

- ни один из вышеперечисленных.
- Какой спецсимвол отвечает за знак градуса?
 - ®
 - °
 - ±
 - ни один из вышеперечисленных.
 - Какой спецсимвол отвечает за знак "больше"?
 - &
 - <
 - >
 - ни один из вышеперечисленных.
 - Какой спецсимвол отвечает за знак "меньше"?
 - &
 - <
 - >
 - ни один из вышеперечисленных.
 - Какой спецсимвол отвечает за знак кавычек?
 - &
 - <
 - >
 - ни один из вышеперечисленных.

■ КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ (CSS)

CSS — расширение языка HTML, с помощью которого можно детально и эффективно менять внешний вид страницы.

Пример:

использование CSS

```
<html>
  <head>
    <title>
      Пример CSS
    </title>
  </head>
  <style>
    h1 {color:red;size:20px;}
  </style>
```

css.css

```

<body>
  <h1> Hello! CSS </h1>
</body>
</html>

```

В данном случае строчка «Hello! CSS» выведется шрифтом красного цвета размером в 20 пикселей.

Основные определения

Селектор — запись, определяющая свойства тега.

Пример:

```
H1 {color: red; size: 20pt}
```

Элементы, расположенные в контейнерах <H1>, приобретут свойства, заданные в фигурных скобках.

Класс — набор свойств, связанных общим именем — именем класса. Класс можно рассматривать как одно сложное свойство, требующее объявления и сочетающее в себе множество простых свойств.

Селектор класса — запись, объявляющая класс и определяющая свойства этого класса.

Пример:

```
class1 {color: blue; font-style: italic}
```

Все теги, принадлежащие классу class1, унаследуют свойства, определенные в классе, — class1.

Селектор закрепленного класса — запись, объявляющая класс, который закреплен за указанным тегом.

Пример:

```
H1.blue {color: blue; size: 20pt}
```

В данном случае классу blue принадлежит единственный элемент — тег <H1>. Другие теги не будут наследовать свойства класса blue.

Идентификатор используется для придания элементам класса индивидуальных свойств.

Пример

```

<html>
  <head>
    <title>
      Пример

```

```

</title>
</head>
<style>
  blue {color:blue; font-style:italic}
  #boldunderline
  {text-decoration: underline; font-weight: bold}
</style>
<body>
  <p class="blue">
    Hello! Welcome To My Homepage!
  </p>
  <p class="blue" id="boldunderline">
    Will Be Soon
  </p>
  <p id="boldunderline">
    Under Construction
  </p>
</body>
</html>

```

В результате, для элементов класса blue с id="boldunderline" произойдет синтез свойств класса blue и идентификатора boldunderline.

Контекстные селекторы — это сочетания нескольких обычных селекторов.

Стиль задается только тегам, расположенным в определенном (каскадном порядке) порядке.

Пример:

```
P EM {color:silver}
```

Содержимое контейнера , вложенного в контейнер <P>, будет иметь свойства «color:silver».

Пример:

```
P, EM, H1, H2, H3 {color: silver; font-size: 12pt}
```

Все элементы, перечисленные через запятую, будут обладать перечисленными свойствами.

Псевдокласс — закрепленный класс, применимый только к элементам определенного типа. Синтаксис аналогичен синтаксису закрепленного класса, но вместо ".", разделяющей элемент

и класс, используется ":" (element: class). Существует три типа псевдоклассов:

- Anchor Pseudo Classes — псевдоклассы элемента <a> (ссылки). Псевдоклассы элемента <a>:
 - link (линк);
 - active (активная ссылка);
 - visited (посещенный ранее URL);
 - hover (свойства срабатывают при поднесении курсора, не работает в Netscape).

Пример:

```
<style>
a: link {color: red}
a: visited {color: blue; text-decoration: none}
</style>
```

- First-Line — изменене стилия первой строки элементов. Этот псевдоэлемент может быть использован с block-level элементами (p, h1 и т. д.).
- First-Letter — изменение стилия первого символа.

Способы использования CSS

Способы использования таблиц стилей:

- Использование внутренних таблиц стилей (для каждого элемента задается свой стиль).

Пример:

```
<font style="color: blue; font-size: 12pt; font-family: Arial">
Hello World
</font>
```

- Использование глобальных таблиц стилей (стиль задается для всей страницы).

Пример:

```
<html>
<head>
<title> Пример Of Global Style Sheets </title>
</head>
<style>
h1 {color: red; font-style: italic; font-size: 32px}
.blue {color: blue}
```

```
#bold {font-weight: bold}
</style>
<body>
<h1>
Заголовок набран крупным красным курсивом
</h1>
. Вот <font class="blue"> это </font> слово —
синее, а <font id="bold"> это </font> — жирное!!!
</body>
</html>
```

В данном примере все элементы H1 будут написаны крупным красным курсивом, все элементы с указанным классом BLUE будут синими, а все элементы с идентификатором ID="Bold" станут жирными.

- Использование внешних таблиц стилей (стиль задается во внешнем файле и может использоваться несколькими страницами).

Пример:

Содержимое файла style.css:

```
body {background: black;
font-size: 9pt; color: red;
font-family: Arial Black}
.base {color: blue; font-style: italic}
h1 {color: white}
#bold {font-weight: bold}
```

Ссылка на стилевой файл выполняется при помощи тега <link>:
<link REL="STYLESHEET" TYPE="text/css" HREF="URL">.

Пример:

```
<html>
<head>
<title>
Пример <META>
<Link...>
</title>
</head>
<link rel="stylesheet" type="text/css" href="styles.css">
<body> Содержание документа </body>
</html>
```

Контрольные вопросы

1. Для чего используются таблицы стилей?
 - а) Для построения таблиц;
 - б) для стилизации HTML-страницы;
 - в) для возможности работы со старыми браузерами;
 - г) для всего вышеперечисленного.
2. Что такое селектор закрепленного класса?
 - а) Запись, закрепляющая класс только за объявленным элементом;
 - б) запись, запрещающая изменение свойств класса;
 - в) запись, запрещающая изменение свойств элементов класса;
 - г) то же, что и псевдокласс.
3. Что такое псевдокласс?
 - а) Класс, закрепленный за любым элементом;
 - б) класс, закрепленный только за определенным элементом;
 - в) класс с неопределенными свойствами;
 - г) любой класс с произвольным именем.
4. С помощью CSS был задан красный цвет текста в контейнере H3. Какого цвета будет текст между тегами и в конструкции вида <H3>...</H3>?
 - а) Красного;
 - б) черного;
 - в) текст не будет отображаться, так как не заданы его свойства;
 - г) зеленого.
5. Какой результат обеспечивает конструкция H3 EM {color: yellow}?
 - а) Делает текст в контейнерах H3 и EM желтым;
 - б) делает текст в контейнере H3 желтым;
 - в) делает текст в контейнере EM желтым;
 - г) делает текст желтым при условии вложенности EM в H3.
6. Как задать толщину шрифта?
 - а) font-size: "...";
 - б) font-family: "...";
 - в) font-weight: "...";
 - г) font-style: "...".

7. Что определяет свойство line-height?
 - а) Толщину горизонтальной линии;
 - б) высоту шрифта;
 - в) высоту текущей строки;
 - г) высоту вертикальной линии.
8. Найдите ошибку:
 - а) P{font-family: "Arial", font-size: 10pt}
P{line-height=200%};
 - б) P{font-family: "Arial"; font-size: 10pt;
line-height=200%};
 - в) P{font-family: Arial; font-size: 10px;
line-height=200px};
 - г) P{font-style: "italic"; font-weight: "bolder"}.
9. Какая из перечисленных записей запрещает прокрутку фоновой картинки при прокрутке страницы?
 - а) background-color: transparent;
 - б) background-attachment: fixed;
 - в) background-image: none;
 - г) background-position: center.
10. Как перевести все буквы в тексте в нижний регистр?
 - а) letter-style: little;
 - б) letter-transform: lowercase;
 - в) text-style: small;
 - г) text-transform: lowercase.
11. Как задать один набор свойств для разных тэгов?
 - а) тэг1{ color="#FF0000"; font-style: "italic"};
тэг2{ color="#FF0000"; font-style: "italic"};
 - б) тэг1, тэг2{ color="#FF0000"; font-style: "italic"};
 - в) .s{ color="#FF0000"; font-style: "italic"}
тэг1 class="s"
тэг2 class="s";
 - г) любым из перечисленных способов.

Задачи для самостоятельного решения по теме «Таблицы»

Создайте таблицу и заполните текстом ее ячейки.

(1)

1. 3X4, бордюр 2 пикселя, фон светлый.
2. 5X6, заголовок таблицы внизу.
3. 7X8, фон темный, отображены только правая и левая границы.

(2)

1. В верхних ячейках текст отображен курсивом.
2. В верхних ячейках помещены картинки.
3. В верхних ячейках помещены ссылки.

(3)

1. Ячейки в средней колонке объединены.
2. Заголовки столбцов темнее заголовков строк.
3. Текст в ячейках отцентрирован по горизонтали и по вертикали, данные отстоят от границ ячеек на 4 пикселя.

Пример задачи

Задача 121: создать таблицу 3X4, бордюр 2 пикселя, фон светлый, в верхних ячейках помещены картинки, ячейки в средней колонке объединены.

Задачи для самостоятельного решения по теме «Фрейм»

(1)

1. Создать два горизонтальных фрейма.
2. Создать два вертикальных фрейма.

(2)

1. Размерами 20 и 80% от окна браузера.
2. Размерами 150 пикселей — первый, а второй — оставшуюся часть от окна браузера.

(3)

1. Разделить первый фрейм на два горизонтальных фрейма с любым соотношением размеров.
2. Разделить второй фрейм на два вертикальных фрейма с любым соотношением размеров.
3. Разделить второй фрейм на два горизонтальных фрейма с любым соотношением размеров.
4. Разделить первый фрейм на два вертикальных фрейма с любым соотношением размеров.

(4)

1. Запретить прокрутку в любом фрейме.
2. Запретить изменение размеров одного из фреймов.
3. Создать отступ между фреймами в 4 пикселя.
4. Сделать так, чтобы при нажатии в одном фрейме на ссылку в другой загружалась страница page2.htm.

■ ПРАВИЛА ОФОРМЛЕНИЯ WEB-ЗАДАНИЙ

- В пределах одного задания используйте шрифты с одинаковыми характеристиками: гарнитурой (начертанием), кеглем (высотой), цветом.
- Желательно, чтобы преобладал один из видов выравнивания абзаца, например, текст дан с отступом от левого края и выравниванием влево.
- Цветовую схему повторите на всех страницах задания — это создаст ощущение связности сайта.
- Цвета ссылок старайтесь выбирать таким образом, чтобы, с одной стороны, читатель видел, что это ссылка, с другой — она бы не мешала ему читать основной текст.
- Сделайте цвет уже посещенных ссылок чуть темнее, чем цвет непосещенных. Не используйте в публикации подчеркнутый текст, воспользуйтесь другим способом его выделения.
- Предусмотрите для посетителя возможность перехода на главную страничку публикации.
- При выполнении навигационной панели графическими средствами сделайте ее текстовую копию.

■ ОПТИМИЗАЦИЯ ИЗОБРАЖЕНИЙ В ФОРМАТЕ JPG

Данный формат хранит изображения с цветовой глубиной 24 bpp (24 бита на пиксель) и использует сжатие с потерей информации. Для оптимизации графики лучше воспользоваться программами PhotoPaint или PaintShopPro.

Как определить оптимальную степень сжатия?

Для определения оптимальной степени сжатия фотографию нужно сохранить при нескольких значениях коэффициента сжатия. Критериями потери качества будут так называемые дрожащие контуры, наиболее заметные в местах контрастных переходов, и появление пятен в областях с плавными переходами.

Еще один метод, позволяющий сократить время просмотра JPG-картинки, — метод прогрессивной развертки. При записи файла в прогрессивном формате изображение появляется сразу целиком в расплывчатой форме и постепенно улучшается. Это дает посетителям возможность сразу же оценить фотографию и решить, стоит ли ожидать ее докачки.

Не редактируйте изображение в формате JPG, а используйте этот формат как конечный для предварительно отредактированного изображения, так как оптимизация осуществляется при каждой перезаписи изображения.

Не рекомендуется использовать формат JPG для контрастных изображений и изображений с четкими границами между отдельными цветами.

■ Оптимизация изображений в формате GIF

Формат GIF предназначен для хранения изображений с количеством цветов не более 256, поддерживает палитру и использует сжатие без потерь.

Существуют следующие виды оптимизации изображений:

- уменьшение количества цветов;
- оптимизация палитры изображения;
- изменение размеров изображения;
- фрагментарная оптимизация;
- оптимизация "прозрачных" изображений;
- использование чересстрочной развертки;

Уменьшение количества цветов

Для большинства изображений, не являющихся фотографиями, количество необходимых для воспроизведения цветов часто меньше 256. Лишние цвета можно убрать из изображения, тем самым уменьшив его размер. Количество цветов в изображении выбирается из ряда: 2, 4, 8, 16, 32, 64, 128, 256.

Оптимизация палитры изображения

Преобразование полноцветного изображения в индексное (т. е. с использованием палитры) является важной операцией. Рассмотрим, как эту операцию выполняют современные графические редакторы.

Палитра результирующего изображения может быть либо фиксированной, либо оптимизированной. В первом случае графический редактор просматривает каждую точку изображения и подбирает ей ближайшую по цвету из палитры. Этот способ дает плохие результаты с точки зрения верности воспроизведения цветов, однако применяется для того, чтобы изображения приемлемо выглядели на мониторах с малым количеством цветов (256). В этом случае применяют так называемую безопасную палитру, состоящую из набора часто используемых цветов и их оттенков. Такая палитра гарантирует, что изображения будут показаны одинаково всеми браузерами.

При использовании оптимизированной палитры графический редактор вначале анализирует рисунок и составляет список всех используемых в изображении цветов. Далее на основании частоты появления цветов составляется палитра, которая называется оптимизированной. После этого цвет пикселя изменяется на ближайший к нему из палитры. Этот способ дает наилучшие результаты.

При оптимизации изображения задача состоит в том, чтобы выбрать в палитре оптимальное количество цветов. Уменьшение количества цветов может привести к дизерингу (dithering) — замене "чистых" цветов рисунка, отсутствующих в палитре, на последовательность точек соответствующих цветов, размещенных в шахматном порядке. Дизеринг может увеличить размер файла, причем файл может стать больше, чем до оптимизации. Увеличение размера файла связано с методом сохранения файла на диске. Основная особенность этого метода (LZW) заключается в том, что сжатию лучше всего поддаются области, заполненные однородным цветом, и хуже всего — области, состоящие из набора разноцветных точек.

Фрагментарная оптимизация

Фрагментарная оптимизация применяется в случае, если изображение содержит сильно отличающиеся по количеству цветов области. В этом случае изображение можно разрезать на фрагменты и поместить их в таблицу, тогда в браузере оно будет выглядеть как единое целое. Теперь каждый фрагмент исходного рисунка мы оптимизируем отдельно. Для фрагментов, содержащих мало оттенков, вполне может хватить палитры в 8 или 16 цветов, а для богатых цветами фрагментов можно использовать полную палитру в 256 цветов или сохранение в формате JPG. Этот способ позволяет уменьшить объем, занимаемый исходным файлом, примерно раза в 2–3 (в зависимости от конкретного изображения); кроме того, изображение будет загружаться более равномерно, так как все фрагменты будут загружаться параллельно. Если количество фрагментов превысит 10, то, возможно, выигрыша и не получится, так как каждый фрагмент должен будет хранить свою копию палитры и служебной информации.

Оптимизация "прозрачных" изображений

Формат GIF89a позволяет сохранять так называемые прозрачные изображения. На самом деле GIF не поддерживает прозрачность в изображениях (альфа-канал), а лишь позволяет назначить одному (любому) цвету в палитре атрибут прозрачности. Когда браузер рисует на экране такой GIF, то, встречая "прозрачный" пиксель, он просто игнорирует его и не выводит на экран. Само по себе это хорошо, но в случае использования метода сглаживания краев изображения (anti-aliasing) и последующего сохранения его в формате GIF возникает проблема — вокруг прозрачного изображения появляется ореол, если GIF показывается на фоне, отличном от того, на котором происходил anti-aliasing.

Единственным способом уменьшить проявления этого эффекта является назначение прозрачным пикселя, близкого по цвету к фону, на котором будет использоваться GIF, и проведение anti-aliasing'a на этом же фоне.

Использование чересстрочной развертки

Следует отметить еще одну особенность формата GIF. Версия GIF89a позволяет сохранять файлы с использованием чересстрочной развертки. При использовании этого способа строки, составляющие изображение, перемежаются и при загрузке браузер вначале показывает каждую восьмую строчку, потом каждую четвертую, вторую, и, наконец, загружается полное изображение. Посетитель странички сможет понять, что нарисовано на данной картинке, не дожидаясь ее полной загрузки.

■ СПОСОБЫ ПЕРЕДАЧИ ФАЙЛОВ В ПОЧТОВОМ СООБЩЕНИИ

В данном разделе рассматриваются элементы стандарта передачи почтового сообщения. В формах возможно использование только одного типа передаваемой информации — «text/plain».

Стандарт MIME (Multipurpose Internet Mail Extension) предназначен для описания тела почтового сообщения Internet. MIME в основном ориентирован на описание в заголовке письма структуры тела почтового сообщения и возможность составления письма из информационных единиц различных типов.

В стандарте зарезервировано несколько способов представления разнородной информации. Для этой цели используются специальные поля заголовка почтового сообщения:

- поле версии MIME;
- поле, в котором указывается тип почтового сообщения (Content-Type).

Поле версии MIME (MIME-Version)

Поле версии указывается в заголовке почтового сообщения и позволяет программе рассылки почты определить, что сообщение подготовлено в стандарте MIME.

Формат поля выглядит как

MIME-Version: 1.0

Поле версии указывается в общем заголовке почтового сообщения и относится к всему сообщению целиком.

Стандарт MIME позволяет перемешивать поля заголовка сообщения с телом сообщения. Поэтому все поля делятся на два класса:

- общие поля заголовка, которые записываются в начале почтового сообщения;
- частные поля заголовка, которые относятся только к отдельным частям составного сообщения и записываются перед ними.

Поле, в котором указывается тип почтового сообщения (Content-Type)

Синтаксис поля Content-Type:

Content-Type: = type «/» subtype *[*;» parameter]

Поле используется для описания типа данных, которые содержатся в теле почтового сообщения. Типы передаваемой информации необходимы для правильной интерпретации полученной информации.

Стандарт MIME определяет семь типов данных (type), которые можно передавать в теле письма:

- текст (text);
- смешанный тип (multipart);
- почтовое сообщение (message);
- графический образ (image);
- аудиоинформация (audio);
- фильм или видео (video);
- приложение (application).

Основные типы передаваемой информации

Тип «Text» — указывает на то, что в теле сообщения содержится текст. Основным подтипом типа «text» является «plain», что обозначает так называемый планарный текст. Понятие планарного текста появилось в связи с тем, что существует еще и размеченный текст (текст со встроенными в него символами управления отображением), и гипертекст (текст, который можно просматривать не последовательно, а произвольно, следуя гипертекстовым ссылкам). Для обозначения размеченного текста используют подтип «richtext», а для обозначения гипертекста — подтип «html». В формах доступен только подтип «plain».

Тип «Multipart» — тип содержания тела почтового сообщения — определяет смешанный документ, который может состоять из фрагментов данных разного типа. Указанный тип имеет ряд подтипов.

Тип «Message» — предназначен для работы с обычными почтовыми сообщениями, которые, однако, не могут быть переданы по почте по разного рода причинам.

Типы описания нетекстовой информации

Тип «Image» — служит для описания графических образов. Наиболее часто используются файлы форматов GIF и JPEG.

Тип «Audio» — служит для описания аудиоинформации. Для воспроизведения сообщения данного типа требуется специальное оборудование.

Тип «Video» — служит для передачи фильмов. Наиболее популярным является формат MPEG.

Тип «Application» — служит для передачи данных любого другого формата. Обычно используется для передачи двоичных данных с последующим промежуточным преобразованием.

СОДЕРЖАНИЕ

ВМЕСТО ВВЕДЕНИЯ	3
■ Принципы гипертекстовой разметки	3
Структура документа	5
Контрольные вопросы	5
✓ <HEAD>	6
✓ <TITLE>	6
<BASE>	6
✓ <META>	7
✓ <LINK>	10
<SCRIPT>	11
Контрольные вопросы	11
✓ <BODY>	13
■ Форматирование текста	14
Логическое форматирование	14
Физическое форматирование	15
Контрольные вопросы	16
■ Шрифты	16
	16
Контрольные вопросы	17
■ Вставка изображений в документ	18
✓ 	18
Контрольные вопросы	19
Задачи	20
■ Ссылки	21
✓ <A>	21
Контрольные вопросы	22
■ Изображения-карты, обслуживаемые браузером	24
<MAP>	24
Контрольные вопросы	24
■ Списки	25
Упорядоченный (нумерованный) список	25
	25
Неупорядоченный (маркированный) список	26
	26
Контрольные вопросы	26
■ Фреймы	27
Создание фрейма	27
<FRAMESET>	27

Свойства фреймов	28
Размеры фреймов	28
<FRAME>	28
Контрольные вопросы	30
Задачи	31
■ Формы	33
<TEXTFIELD>	34
<TEXTAREA>	35
<CHECKBOX>	35
<RADIOBUTTON>	35
Задачи	36
Контрольные вопросы	36
■ Таблицы	37
✓ <TABLE>	38
<CAPTION>	39
✓ <TR>	39
✓ <TD>, <TH>	39
<THEAD>, <TFOOT>, <TBODY>	40
<COLGROUP>	40
<COL>	40
Контрольные вопросы	40
Задачи	42
■ Тэги форматирования текста	45
✓ <P>	45
<HR>	45
✓ 	45
<BLOCKQUOTE>	46
Контрольные вопросы	46
Задачи	47
■ Слои	48
<LAYER>, <ILAYER>	48
Задачи	50
Контрольные вопросы	51
■ Спецсимволы	51
Контрольные вопросы	52
✓ ■ Каскадные таблицы стилей (CSS)	53
Основные определения	54
Способы использования CSS	56
Задачи для самостоятельного решения по теме «Таблицы»	60
Задачи для самостоятельного решения по теме «Фрейм»	61

ПРИЛОЖЕНИЯ	62
■ Правила оформления Web-заданий	62
■ Оптимизация изображений в формате JPG	63
Как определить оптимальную степень сжатия?	63
■ Оптимизация изображений в формате GIF	64
Уменьшение количества цветов	64
Оптимизация палитры изображения	64
Фрагментарная оптимизация	65
Оптимизация "прозрачных" изображений	66
Использование чересстрочной развертки	66
■ Способы передачи файлов в почтовом сообщении	67
Поле версии MIME (MIME-Version)	67
Поле, в котором указывается тип почтового сообщения (Content-Type)	68
Основные типы передаваемой информации	68
Типы описания нетекстовой информации	69

*Михайлов Николай Иванович,
Кириллова Наталья Рэмовна,
Перепеловский Вадим Всеволодович*

HTML-ТЕХНОЛОГИЯ
Учебное пособие

Редактор И. Б. Синишева

ЛР № 020617 от 24.06.98

Подписано в печать 11.10.2000. Формат 60x84 1/16.
Бумага офсетная. Печать офсетная.
Усл. печ. л. 4,18. Тираж 60 экз. Заказ

Издательство СПбГЭТУ "ЛЭТИ"
197376, Санкт-Петербург, ул. Проф. Попова, 5

Отпечатано с готового оригинал-макета

Отпечатано в АОЗТ "КопиСервис",
194017, Санкт-Петербург, Скобелевский пр., 16