

Приложение 1

HTML

Язык HTML (HyperText Markup Language) позволяет публиковать в Internet документы с помощью заголовков, текста, списков, таблиц, получать доступ к документам с помощью гиперссылок, включать видеоклипы, звук и т.д. Страницы JSP, привнося свои теги, активно используют уже существующие HTML-теги.

Существует несколько версий HTML. В 1999 году вышла последняя редакция языка – HTML 4.01. С тех пор развиваются языки разметки, базирующиеся на XML. Среди них – XHTML 1.0/ XHTML 2.0, MathML (язык разметки математических формул) и некоторые другие.

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, должен начинаться со строки объявления версии HTML, которая выглядит так:

Строгий (Strict) – не содержит элементов, помеченных как «устаревшие» или «не одобряемые» (deprecated):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Переходный (Transitional) – содержит устаревшие теги в целях совместимости и упрощения перехода со старых версий HTML:

```
<!DOCTYPE HTML PUBLIC
"-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

С фреймами (Frameset) – аналогичен переходному, но содержит также теги для создания наборов фреймов:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

Документ, соответствующий XHTML 1.0, автоматически соответствует также и HTML 4.0. XHTML 2.0 обеспечивает полное отделение содержания документа от его представления. В нем нет ничего отвечающего за внешний вид документа – нет элементов **font**, **i**, **b**, **center**, нет атрибутов **align**, **size** и **bgcolor**. Предполагается, что вся информация об оформлении документа содержится в отдельной таблице стилей. Кроме того, в XHTML 2.0 вводится много новых элементов – например элемент **section**, обозначающий структурную часть документа, и элемент **h** – заголовок общего вида.

Таким образом, если предполагается использование документа вместе с расширениями в XHTML 2.0, базирующемся на XML, следует учитывать, что в документе нельзя использовать устаревшие или нежелательные элементы. При этом:

- верхний и нижний регистры различаются, все теги следует писать с маленькой буквы;
- теги **center**, **font**, **s**, **u**, атрибуты **align**, **bgcolor**, **background**, **clear**, **color**, **face**, **size**, т.е. почти все теги и атрибуты, определяющие представление документа HTML (цвета, выравнивание, шрифты, графика и т.д.), являются нежелательными, взамен рекомендуется использовать таблицы стилей;

- если пишется атрибут, то обязательно должно быть указано его значение. Все значения атрибутов должны писаться в кавычках.
- обязательно прописывать и закрывающие теги! (например у тега **li**). Примеры тегов, у которых отсутствуют закрывающие теги: **
, ** и т.д. В случае если закрывающий тег отсутствует, то в конце тега пишется «/».

HTML-документ создается с помощью HTML-редактора и сохраняется в виде текстового файла с расширением **html** или **htm**. Для просмотра HTML-документов используются Web-браузеры, интерпретирующие документы.

HTML-документ состоит из вложенных друг в друга элементов (тегов). Теги **<html>** начинаются со стартового тега ("**< >**") и заканчиваются завершающим тегом ("**</>**"). Сам документ – это один большой элемент вида:

```
<html>
<!--Содержание документа-->
</ html>
```

HTML не реагирует на регистр символов, описывающих тег (в XML используются строчные буквы).

Заголовочная часть документа **<HEAD>**

Тег заголовочной части документа используется сразу после тега **<HTML>**. Данный тег содержит общее описание документа. Например:

```
<HTML><HEAD>
      <TITLE> Список сотрудников </TITLE>
</HEAD></HTML>
```

Заголовок документа **<TITLE>**

Большинство Web-браузеров отображают заголовок документа, ограниченный тегами **<TITLE>** и **</TITLE>** вверху экрана, отдельно от содержимого документа.

Тело документа **<BODY>**

Тело документа должно находиться между тегами **<BODY>** и **</BODY>**. Это та часть документа, которая отображается как текстовая и графическая информация документа. Технически стартовые и завершающие теги типа **<HTML>**, **<HEAD>** и **<BODY>** необязательны. Но настоятельно рекомендуется их использовать, поскольку использование данных тегов позволяет Web-браузеру разделить заголовочную и смысловую часть документа.

```
<!--пример # 1: простой HTML документ-->
<HTML>
<HEAD><TITLE>Домашняя страница </TITLE></HEAD>
<BODY>
<!-- Это комментарий-->
<H1>Пример заголовка, размер 1</H1>
<H6>Пример заголовка, размер 6</H6>
<ADDRESS>Романчик - e-mail:rom@bsu.by</ADDRESS> <P>
</BODY></HTML>
```

В этом примере использовались теги: **<P>** – тег нового абзаца. Тег **<ADDRESS>** позволяет сформировать информацию о связи с автором документа и определяет вид сообщения. HTML позволяет вставлять в тело документа комментарии. Синтаксис комментария:

```
<!-- Это комментарий -->
```

В тексте HTML-документа структурно выделяются собственно текст, заголовки частей текста, заголовки более высокого уровня и т.д. Самый большой заголовок обозначается цифрой 1, и т.д. до шести. Синтаксис заголовка уровня 1 следующий:

```
<H1> Заголовок первого уровня </H1>
```

В языке описания гипертекстовых документов все теги – парные (start-тег и stop-тег). В конечном теге присутствует слэш, который сообщает обозревателю о завершении тега. Существуют исключения из этого правила пар, например: тег **<P>**, определяющий абзац, не требует завершающего тега, хотя может его иметь. Если браузер не понимает тега, то он его просто пропускает.

Разбивка текста на смысловые группы

В HTML-документе браузер игнорирует символы возврата каретки и перехода к новой строке. Браузер разделяет абзацы при наличии тега **<P>**, который вставляет новую строку и осуществляет переход к следующей строке.

Дополнительные параметры тега **<P>** позволяют выравнивать абзац по левому краю, центру и правому краю соответственно:

```
<P ALIGN=left|center|right>
```

Тег **
** извещает браузер о разрыве строки. Дополнительный параметр **CLEAR** тега **
** позволяет не просто выполнить перевод строки, но и разместить следующую строку, начиная с чистой левой (**left**), правой (**right**) или обеих (**all**) границ окна браузера:

```
<BR CLEAR=left|right|all>
```

Например, если рядом с текстом слева встречается рисунок, то можно использовать тег **
** для смещения текста ниже рисунка:

```
<!--пример # 2: разбивка текста-->
<HTML><HEAD>
<TITLE>Домашняя страница </TITLE></HEAD>
<BODY>
<H1>Пример заголовка, размер 1</H1>
Шура Балаганов <P>   <!--Пропуск строки-->
Октябрьская улица, <BR> <!--Разрыв строки-->
10а, офис 326 <BR>
<p> данная строка демонстрирует <BR CLEAR=left>

разрыв строки и вывод слева после рисунка </p>
</BODY></HTML>
```

Здесь тег **** включает рисунок.

Если не требуется, чтобы браузер автоматически переносил строку, то можно обозначить ее тегами **<NOBR>** и **</NOBR>**. В этом случае браузер не будет пере-

носить строку, даже если она выходит за границы экрана. Вместо этого браузер позволит горизонтально прокручивать окно. Например: **<NOBR>**. Данная строка является самой длинной строкой документа, которая не допускает какой-либо разбивки где бы то ни было: **</NOBR>**. Если необходимо всё же позволить разбивку данной строки на две, но в строго запланированном месте, то следует вставить тег **<WBR>** в это место.

Линии

Тег **<HR>** проводит контурную горизонтальную линию (опция **SHADE** по умолчанию). Например:

<HR NOSHADE> – горизонтальная линия с тенью;

<HR WIDTH=75% ALIGN=LEFT|CENTER|RIGHT> – ширина 75%, выравнивание влево, по центру, вправо;

<HR SIZE=n> устанавливает толщину линии в **n** пикселей, где **n** от 1 до 175 (по умолчанию **n=2**).

<!--пример # 3: линии с разным выравниванием-->

<HTML><HEAD>

<TITLE>Примеры горизонтальных линий **</TITLE></HEAD>**

<BODY>

**** Стандартная линия, задаваемая тегом **<HR>**: ****

<HR><P>

**** Линия, заданная тегом **<HR>** с параметром **NOSHADE**: ****

<HR NOSHADE>

**** Линия шириной 50% и с выравниванием по левому краю: ****

<HR WIDTH=50% ALIGN=LEFT><P>

**** Линия шириной 25% и с выравниванием по центру: ****

<HR WIDTH=25% ALIGN=CENTER><P>

</BODY></HTML>

Предварительное форматирование

Дополнительные пробелы, символы табуляции и возврата каретки в исходном тексте HTML-документа будут проигнорированы Web-браузером при интерпретации документа. HTML-документ может включать вышеописанные элементы, только если они помещены внутрь тегов **<PRE>** и **</PRE>**. Эти теги используются, чтобы текст выглядел так, как набран, например при создании таблиц.

Стилевое оформление текста

Приведенные ниже теги **<center>**, ****, **<s>**, **<u>** для оформления стиля текста применяются в настоящее время крайне редко и являются нежелательными элементами. Вместо них широко используются таблицы стилей CSS.

Тег **<CENTER>** позволяет центрировать все элементы документа в окне браузера. Например: **<CENTER><H1>**. Все элементы между тегами будут находиться в центре окна **</H1></CENTER>**. Гипертекстовый документ может быть оформлен с использованием следующих стилей:

**** Полужирный ****, **<I>** Курсив **</I>**, **<TT>** Моноширинный **</TT>**,

<S> Зачеркнутый текст **</S>**,
<U> Подчеркнутый текст **</U>**, **<BIG>** КРУПНЫЙ ТЕКСТ **</BIG>**,
<SMALL> мелкий текст **</SMALL>**, **_{** Нижний индекс **}**,
^{ Верхний индекс **}**.

```

<!--пример # 4: различные стили форматирования-->
<HTML><HEAD>
<TITLE>Домашняя страница </TITLE></HEAD>
<BODY>
<CENTER> <H1>Добро пожаловать </H1> </CENTER><HR>
Я рад <B>приветствовать Вас</B>
на моей <I>странице</I>. <P>
Вот что я <SUP>люблю</SUP> делать в <SUB>свободное</SUB>
время: <BR>
<U> - Читать книги </U><BR> <!--Подчеркивание текста -->
<S>- Исследовать Интернет</S><BR> <!--Зачеркивание текста -->
</BODY></HTML>

```

Логический стиль документа

Текст в документе HTML может быть логически выделен одним из следующих тегов:

<DFN> – определить слово. Как правило – курсив;
**** – усилить акцент. Как правило – курсив;
<CITE> – заголовок чего-то большого. Курсив;
<CODE> – компьютерный код. Моноширинный шрифт;
<KBD> – текст, введенный с клавиатуры. Жирный шрифт;
<SAMP> – сообщение программы. Моноширинный шрифт;
**** – очень важные участки. Жирный шрифт;
<VAR> – замена переменной на число. Курсив;
<BLOCKQUOTE> – позволяет включить цитату в объект.

```

<!--пример # 5: логический стиль документа-->
<HTML><HEAD>
<TITLE> Элементы содержания </TITLE></HEAD>
<BODY bgcolor="white" >
<CENTER><H5 > Элементы содержания </H5></center> <HR>
<P><BLOCKQUOTE> Это цитата </blockquote>
<P> <INS> Использование элемента INS </ins>
<P> <DEL> Использование элемента DEL </del>
<P> <Q> Использование элемента Q </q>
<P> <EM> Использование элемента EM </em>
<P> <STRONG> Использование элемента STRONG </strong>
</BODY></HTML>

```

Работа с тегами FONT

Тег **** позволяет установить вид, размер и цвет шрифта.

```

<FONT SIZE=n> размер шрифта n=1..7, стандартный размер n=3
</FONT>
<FONT SIZE= +3> относительный размер, 3+3=6 </FONT>

```

Кроме размера, могут устанавливаться цвет и тип шрифта, например:

```
<FONT COLOR=RED SIZE=6 FACE="Arial"> Пример шрифта </FONT>
```

<!--пример # 6: различные виды шрифтов-->

```
<HTML><HEAD>
<TITLE> Элементы форматирования текста</TITLE></HEAD>
<BODY>
<HR> <H3>Задание абсолютных размеров шрифтов</h3>
<P><FONT size=7> Шрифт размера 7</font>
<P><FONT size=1> Шрифт размера 1</font> <HR>
<H3>Задание относительных размеров шрифтов</h3>
<P><FONT size=+4> Шрифт размера +4</font>
<FONT color="green"> Задан зеленый цвет шрифта</font>
<P><FONT size=+1 face="Courier" color="red"> Шрифт Courier
</font>
</BODY></HTML>
```

Цвет символов на всей странице можно изменить с помощью аргумента **TEXT** тега **<BODY>**: **<BODY TEXT="цвет">...</BODY>** Аргумент **BGCOLOR="цвет"** изменяет цвет фона.

<!--пример # 7: управление цветом текста-->

```
<HTML><HEAD>
<TITLE> Цветовое оформление </TITLE></HEAD>
<BODY bgcolor="white" TEXT="blue">
<CENTER>
<FONT size=6 color="red">Управление цветом текста</font>
<HR color="red">
<TABLE border=3 >
<TR><TD>Аквамарин - aqua<TD bgcolor="aqua" width=200>
<TR><TD>Белый - white<TD bgcolor="white" width=200>
<TR><TD>Желтый - yellow<TD bgcolor="yellow" width=200>
<TR><TD>Синий - blue<TD bgcolor="blue" width=200>
<TR><TD>Ультрамарин - navy<TD bgcolor="navy" width=200>
<TR><TD>Фиолетовый - violet<TD bgcolor="violet" width=200>
<TR><TD>Фуксиновый - fuchsia<TD bgcolor="fuchsia"
width=200>
<TR>
<TD>Черный - black<TD bgcolor="black" width=200>
</table> </center> <HR color="red">
<HR color="lime" size=15 width=195 align="left">
</BODY></HTML>
```

Специальные символы

Символы, которые не могут быть введены в текст документа непосредственно через клавиатуру, называются специальными символами. Для них существуют особые теги. Четыре символа – знак меньше **<**, знак больше **>**, амперсant **&** и двойные кавычки **"** имеют специальное значение внутри HTML и, следовательно, не могут быть использованы в тексте в своем обычном значении. Для использования одного из этих символов введите одну из следующих последовательностей:

< - **<**; **>** - **>**; **&** - **&**; **"** - **"**.

Списки и таблицы

Списки подразделяются на нумерованные, создаваемые с тегом ****, и нумерованные, создаваемые с тегом ****:

```
<!--пример # 8: нумерованные и нумерованные списки-->
<HTML><HEAD>
<TITLE>Использование списков </TITLE></HEAD>
<BODY>
<CENTER><H3>Домашняя страница </H3></CENTER>
<h4>Ненумерованный список: Мои интересы:</h4>
<UL>
<LI><B>Занятия в свободное время:</B></LI>
<LI> Компьютеры
<LI> Чтение книг
<LI> Бассейн
<LI> Отдых на природе
</UL> <HR>
<H4> Нумерованный (упорядоченный) список.</H4>
<OL TYPE=1>
<LI><B>Мое путешествие</B></LI>
<LI> Прибытие в Варшаву
<LI> Автобусом в Будапешт
<LI> Самолетом в Рим
</OL> <HR>
<OL TYPE=A>
<LI><B>Продолжение путешествия</B></LI>
<LI> Автобусом в Берлин
<LI> Поездом в Варшаву
<LI> Пешком в Минск
</OL> <HR>
</BODY></HTML>
```

Атрибуты ненумерованного списка: **type="disk | circle | square"**. Атрибут нумерованного списка **start** - устанавливает число, с которого будет начинаться отсчет. При этом вид нумерации устанавливается аргументом **TYPE**: **TYPE=1** – стандартная нумерация 1,2,3,4...; **TYPE=A** – прописные буквы A, B, C, D...; **TYPE=a** – строчные буквы a,b,c,d...; **TYPE=I** – римские цифры I, II, III, IV...; **TYPE=i** – строчные римские цифры i,ii,iii,iv,v....

Еще один вид списков – списки определений **DL** – состоит из пар элементов: определяемого **<DT>** и определения **<DD>**.

```
<!--пример #9: списки определений-->
<HTML><HEAD>
<TITLE>Использование списков</TITLE></HEAD>
<BODY> <CENTER><H2> Толковый словарь</H2></CENTER><HR>
<DL>
<LI><Big><B> Список терминов</B></Big></LI><HR>
<DT><B>"Anchor"</B>
```

```
<DD><I>То, что образует гипертекстовую ссылку</I>
    <DT><B>"Lamer"</B>
<DD><I> Юзер-идиот</I>
    <DT><B>"Cookies "</B>
<DD><I>Технология, позволяющая сохранять индивидуальную ин-
формацию о пользователе сети</I>    </DL>
</BODY></HTML>
```

Для создания таблиц используются следующие теги **HTML**: **<TABLE>** и **</TABLE>** – охватывают таблицу. Для того чтобы была видна сетка, разделяющая строки и столбцы, используется атрибут **BORDER** (например: **<TABLE BORDER=1>**). Текст в тегах **<CAPTION>** и **</CAPTION>** выводится в виде заголовка. В тегах **<TH>** и **</TH>** помещаются заголовки столбцов или строк. Теги **<TR>** и **</TR>** определяют каждую строку таблицы. Теги **<TD>** и **</TD>** определяют текст каждой ячейки таблицы.

<!--пример # 10: простая таблица-->

```
<HTML><HEAD>
<TITLE>Использование таблиц</TITLE>
</HEAD><BODY>
<TABLE BORDER=
<CAPTION ALIGN=top>Лучшие нападающие года</CAPTION>
    <TR>
        <TH>Имя</TH>
        <TH>Команда</TH>
        <TH>Очки</TH>
    </TR>
    <TR>
        <TD>Small    </TD>
        <TD> Барселона</TD>
        <TD>5</TD>
    </TR>
    <TR>
        <TD>Superman</TD>
        <TD> Dinamo</TD>
        <TD>10</TD>
    </TR>
</TABLE></BODY></HTML>
```

Чтобы ячейка занимала две строки вместо одной, можно заменить тег на следующий: **<TD ROWSPAN=2>** **</TD>**. Аналогично два столбца можно объединить с помощью тега **<TH COLSPAN=2>** или **<TD COLSPAN=2>**. Изменить цвет в таблице можно с помощью аргумента **BGCOLOR**, как в следующем примере:

<!--пример # 11: изменение цвета-->

```
<HTML><HEAD>
<TITLE> Таблицы </TITLE></HEAD>
<BODY bgcolor="white">
<CENTER><FONT size=6>Примеры таблиц</font></center>
<HR color="blue">
```



```

<TABLE border=2 cellspacing=2>
<TR><TD bgcolor="red">Таблица 2
  <TD bgcolor="red">Ячейка 2-2
<TR><TD bgcolor="red">Ячейка 3-2
  <TD bgcolor="red">Ячейка 4-2
</TABLE>
<TR><TD bgcolor="yellow">Ячейка 3-1
  <TD bgcolor="yellow">Ячейка 4-1
</TABLE></BODY></HTML>

```

Ссылки

HTML позволяет связать текст или картинку с другими гипертекстовыми документами с помощью тегов **<A>** и **<LINK>**. Текст, как правило, выделяется цветом или оформляется подчеркиванием. Чтобы сформировать ссылку, следует набрать **<A, введите HREF= "filename">**, ввести текст ссылки, закрыть тег ****. В следующем примере слово **Minsk** ссылается на документ **MinskAnapa.html**, образуя гипертекстовую ссылку:

```
<A HREF="MinskAnapa.html">Minsk</A>
```

Если документ, формирующий ссылку, находится в другой директории, то подобная ссылка называется относительной. Например:

```
<A HREF="MinskAnapa/MinskMoscow.html">Minsk</A>
```

Ссылки можно формировать на основе URL, используя синтаксис: **protocol: //hostport/path**. Например:

```

<A HREF="http://www.w3.org/TR/REC-html40">Документ HTML</A>
<!--пример # 12: создание ссылок на html-файлы-->
<HTML><HEAD>
<TITLE>Ссылки на домашней странице</TITLE></HEAD>
<BODY>
<H1>Внутренние ссылки на части документа</H1></CENTER>
<FONT SIZE=+1>
<HR NOSHADE>
<H2>Вы можете:</H2>
<UL>
<LI>Посмотреть <A HREF="Pr11.htm">Простейший пример</A>
<LI>Посмотреть <A HREF="Pr12.htm">Второй пример</A>
<LI>Посмотреть <A HREF="Pr13.htm">разбиение текста</A>
<LI>Узнать <A HREF="Pr14.htm">О линиях</A>
</UL>
<HR NOSHADE>
Если вас интересует подробная информация, пишите по адресу
  <A HREF="mailto:Rom@Bsu.by">Rom@Bsu.by</A>
</FONT></BODY></HTML>

```

Якоря на Web-странице

Для того чтобы организовать ссылки на разделы документа, находящегося в одном файле, используются якоря (**anchors**). При этом создается ссылка на якорь: **Текст гиперссылки**.

Сам якорь с указанным именем помещается в то место документа, в которое осуществляется переход: ** Текст **.

```
<!--пример # 13: ссылки на якоря-->
<HTML> <HEAD>
<TITLE>Якоря на домашней странице</TITLE>
</HEAD>
<BODY>
<!-- Создание ссылок на якоря -->
<UL><LI><A href="#section1">Введение</A>
<LI><A href="#section2">Обзор</A>
<LI><A href="#section3">Подробное рассмотрение</A></UL><P>
<HR>
...тело документа...
<HR>
<H2><A name="section1">Введение</A></H2><HR>
...section 1...
<HR><!-- Установка якорей -->
<H2><A name="section2">Обзор</A></H2><HR>
...section 2...
<HR>
<H3><A name="section3">Подробное рассмотрение</A></H3><HR>
...section 3...
<HR>
<A HREF="mailto:Romanchik@Bsu.by">
<ADDRESS>Романчик - e-mail:rom@bsu.by</ADDRESS></A>
<P></BODY> </HTML>
```

Такой же эффект можно получить, используя заглавия вместо якоря:

```
<!--пример # 14: заглавия вместо якоря-->
<HTML><HEAD>
<TITLE>Ссылки на заголовки</TITLE>
</HEAD>
<BODY>
<H1>Table of Contents</H1><P>
<A href="#section1">Introduction</A><BR>
<A href="#section2">Some background</A><BR>
<A href="#section3">The first part</A><BR>
...the rest of the table of contents...
<H2 id="section1">Introduction</H2> <HR>
...section 1...<HR>
<H2 id="section2">Some background</H2><HR>
...section 2...<HR>
<H3 id="section3">The first part</H3><HR>
...section 3...<HR>
...Продолжение документа...
</BODY></HTML>
```

Кроме элемента **A**, который помещается в теле HTML-документа **<BODY>**, для организации ссылок используется элемент **LINK**, который помещается в заголовке документа **<HEAD>**.

Установить цвет ссылки можно с помощью атрибута **LINK** тега **<BODY>**, цвет посещенной ссылки – с помощью атрибута **VLINK**, цвет активной ссылки – с помощью атрибута **ALINK**. Например:

```
<BODY TEXT=LIME LINK=RED VLINK=SILVER ALINK=BLUE>
```

Чтобы установить ссылку с помощью изображения, надо вместо текста ссылки поставить HTML-код для вывода изображения:

```
<A HREF="sample.htm"> <IMG SRC="image.gif"> </A>
```

Изображения на странице

Любая Web-страница должна иметь графические изображения. Рисунки можно изготовить с помощью графических редакторов.

Как поместить изображение на домашнюю страницу? Изображение можно вставить в любое место страницы с помощью тега ****, вставляемого между тегами **<BODY>** и **</BODY>**. Например, изображение **FRACT.GIF** загружается из текущего каталога: ****.

Загрузка изображения из каталога уровнем выше:

```
<IMG SRC=../FRACT.GIF>
```

Загрузка изображения с другого диска:

```
<IMG SRC=FILE:///D:\FRACT.GIF>
```

Считается хорошим тоном вместе с изображением использовать альтернативный текст, который выводится, пока изображение загружается. Например:

```
<IMG SRC=FRACT.GIF ALT="Фрактал">
```

В последних версиях HTML атрибут **alt** обязателен для тега **img**.

Если изображение не найдено, браузеры выводят на место изображения стандартную пиктограмму. Изображения можно выравнивать с помощью атрибута **ALIGN** по левому, правому краю, самому высокому элементу в строке, по середине: **ALIGN=LEFT|RIGHT|TOP|TEXTTOP|MIDDLE|ABSMIDDLE|BOTTOM|BASELINE**

В примере текст будет выводиться справа от изображения:

```
<IMG SRC=fract.gif ALT="Фрактал" ALIGN=LEFT>
```

Размеры изображения можно изменять с помощью атрибутов **HEIGHT** и **WIDTH**. При этом меняются не размеры изображения, а только вид на экране. Например:

```
<IMG SRC=tigers.gif ALT="ТИГРЫ" ALIGN=LEFT WIDTH=240  
HEIGHT=260>
```

Фон

Цвет фона устанавливается атрибутом **BGCOLOR** тега **<BODY>**. Например:

```
<BODY BGCOLOR="RED">
```

Фоновые изображения задаются атрибутом **BACKGROUND**. Например:

```
<BODY BACKGROUND="bg.jpg">
```

Фоновые рисунки похожи на обои, наклеиваемые из небольших периодических рисунков. Обычно цвет фона светлый, рисунок легкий.

Формат GIF позволяет задать один из 256 цветов, который при отображении браузером будет игнорироваться, вместо него устанавливается цвет фона, так что изображение будет прозрачным. Прозрачные изображения можно создать в формате GIF графическим редактором.

Наложение изображений

Ключевое слово **LOWSRC** позволяет сначала загрузить файл с низким разрешением, затем больший файл с высоким разрешением:

```
<IMG SRC="highcar.gif" LOWSRC="lowcar.gif">
```

В примере сначала загружается файл **lowcar.gif**, а затем **highcar.gif**.

Анимация

Анимационный GIF является обыкновенным графическим файлом. Дело в том, что подобный файл состоит из нескольких изображений, которые через браузер последовательно выводятся на экран. Чтобы создать анимационную картинку, необходимо сначала создать картинки, из которых будет состоять результирующий файл. Эти картинки можно сделать, например, в Adobe Photoshop.

Для внедрения в документ таких объектов, как видеоклипы, музыкальные файлы, flash-анимация могут использоваться теги **<OBJECT>** и **<EMBED>**.

Например, подгрузить картинку в документ можно следующим образом:

```
<object data="fract.jpg" type="image/jpeg"> текст  
фрактала</object>
```

Атрибут **data** указывает местоположение данных для объекта, атрибут **type** указывает тип содержимого **image** – изображение, **jpeg** – тип изображения. Текст, заключенный в тег **<OBJECT></OBJECT>**, отобразится, если браузер не сможет отобразить объект-картинку.

Чтобы подгрузить видеоклип, запишем аналогично:

```
<object data="film.mpeg" type="application/mpeg"> фильм  
</object>
```

Тэг **<EMBED>** поддерживается на данный момент почти всеми браузерами и используется, как и тег **<OBJECT>**, для внедрения объектов в документ. Рассмотрим внедрение музыкального файла в документ:

```
<embed src="music.mid" width="40" height="30"  
autostart="false" loop="true"  
play_loop="5" hidden="false">  
</embed>
```

В теге могут использоваться атрибуты: **src** (путь к объекту), **align** (выравнивание), **width** (ширина), **height** (высота), **hspace** (расстояние до текста или других объектов по горизонтали), **vspace** (расстояние до текста или других объектов по вертикали), **hidden="true" (false)** – отображать указанный объект, **type=" image/jpeg"** – тип подгружаемого объекта, **pluginspage="http://..."** – путь к плагину (plugin – небольшая программка, выполняющая какие-либо дополнительные функции. Например, чтобы проиграть flash-анимацию, браузеру нужен плагин, к которому мы указываем путь на случай отсутствия в браузере).

Рассмотрим пример вставки flash-анимации в документ:

```
<EMBED src="someMovie.swf" quality=high bgcolor=#FFFFFF
WIDTH=500 HEIGHT=400 swLiveConnect=true NAME=yourMovie
TYPE="application/x-shockwave-flash"
PLUGINSOURCE="http://www.macromedia.com/Shockwave/download/
index.cgi?Pl_Prod_Version=ShockwaveFlash">
</EMBED>
```

Для тега **<OBJECT>** четыре параметра (**HEIGHT**, **WIDTH**, **CLASSID** и **CODEBASE**) являются атрибутами и записываются внутри тега **<OBJECT>**, а все остальные записываются отдельно в виде тегов **<PARAM>**. Например:

```
<OBJECT CLASSID="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
WIDTH="100"
HEIGHT="100"CODEBASE="http://active.macromedia.com/flashS/
cabs/swflash.cab#version=5,0,0,0">
<PARAM NAME="MOVIE"VALUE="movienam .swf">
<PARAM NAME="PLAY"VALUE="true">
<PARAM NAME="LOOP"VALUE="true">
<PARAM NAME="QUALITY"VALUE="high">
</OBJECT>
```

Фреймы

Технология фреймирования в HTML позволяет просматривать в одном окне обозревателя несколько гипертекстовых документов. Однако в последние годы фреймы практически не используются.

Один фрейм отображает только один гипертекстовый документ. Создание фрейма осуществляется с помощью тегов **<FRAMESET>** и **</FRAMESET>**. Тег **<BODY>** в этом случае не используется. Если браузер не поддерживает фреймы, то между тегом **<NOFRAMES>** и тегом **</NOFRAMES>** заносится текст, который распознает браузер.

Тег **<FRAME SRC="Name1">** позволяет описать первый фрейм, т.е. присвоить имя гипертекстовому документу. Второй фрейм описывается тегом **<FRAME SRC="Name2" NAME="Main">**.

Тег **<FRAMESET COLS=n>** позволяет определить количество фреймов и задать размер фреймов в процентах от размера окна обозревателя или зафиксировать эти размеры в пикселях.

Тег **<FRAME>**, имеющий самое большое количество атрибутов, позволяет настроить свойства фрейма. Ниже описываются эти атрибуты:

NAME= – имя фрейма.

MARGINWIDTH= – горизонтальный отступ (от 1 до 6) между фреймом и его границей.

MARGINHEIGHT= – вертикальный отступ (от 1 до 6) между фреймом и его границей.

SCROLLING=YES|NO|AUTO – позволяет создать/не создавать полосы прокрутки.

SCROLLING=AUTO – позволяет отображать полосы прокрутки в зависимости от свойств обозревателя.

NORESIZE – фиксированный размер фрейма.

SRC= – задать гипертекстовый документ для этого фрейма.

TARGET=Name – открыть ссылку во фрейме с именем **Name**.

```
<!--пример # 15: фреймы-->
<HTML><HEAD>
<TITLE> Пример фреймов </TITLE></HEAD>
<FRAMESET rows="20%,60%,20%">
<FRAME src="fr1.htm" noresize scrolling="no">
<FRAMESET cols="22%,78%">
<FRAME src="fr2.htm">
<FRAME src="fr3.htm" scrolling="yes" marginwidth="10"
marginheight="75">
</frameset>
<FRAME src="fr4.htm">
</frameset>
<NOFRAMES>
<CENTER><FONT size=6>Фреймы</font></center>
<HR color="blue">
Этот браузер не может воспроизводить фреймы
</noframes>
</frameset></HTML>
```

Формы HTML

Пользователь заполняет форму и передает информацию из нее для обработки программе, работающей на сервере. Эта программа может быть написана по технологии CGI, ASP или Servlet/JSP.

Теги **<FORM>...</FORM>** используются для обозначения документа как формы. Внутри элемента **<FORM>** определяется последовательность элементов **<INPUT>**, которые представляют поля для ввода информации.

<INPUT TYPE=TEXT> помещает в форму текстовое поле данных.

Если **<INPUT>** используется с атрибутом **TYPE=TEXT**, устанавливаемым по умолчанию, то могут быть использованы еще три атрибута. Атрибут **MAXLENGTH** устанавливает максимальное число вводимых символов. Атрибут **SIZE** определяет размер видимой на экране области, занимаемой текущим полем. Атрибут **VALUE** устанавливает начальное значение поля.

<INPUT TYPE=CHECKBOX> позволяет определить флажок для протокола передачи. Тип элемента ввода **CHECKBOX** позволяет получить ответы пользователя типа ДА/НЕТ. Элемент **INPUT** при установке атрибута **TYPE=CHECKBOX** использует также атрибуты **NAME="имя"** **VALUE="значение"**. Элемент **<INPUT TYPE=RADIO>** позволяет определить кнопку переключения и используется, если надо выбрать одно из нескольких значений.

```
<!--пример # 16: простая форма и элементы checkbox и radio-->
<HTML><HEAD>
<TITLE>Простая форма, checkbox и radio </TITLE>
</HEAD><BODY>
<FORM><H2>Простая форма</H2>
```

```

<P>My street:<INPUT NAME="street"><BR>
City: <INPUT NAME="city" SIZE="20" MAXLENGTH="20"
VALUE="Minsk"> <BR>
Zip: <INPUT NAME="zip" SIZE="5" MAXLENGTH="5"
VALUE="99999"><BR>
</FORM> <HR>
<P><H2>Ваша любимая команда</H2>
<FORM><!--Выбор одной или нескольких команд-->
<INPUT TYPE="checkbox" NAME="team" VALUE="шахтеры">
шахтеры<BR>
<INPUT TYPE="checkbox" NAME="team" VALUE="ковбои"> ковбои
<BR> <INPUT TYPE="checkbox" NAME="team" VALUE="викинги">
викинги<BR>
</FORM> <HR>
<P><H2>Какая из команд самая любимая?</H2>
<FORM><!--Выбор только одной из нескольких команд-->
<INPUT TYPE="radio" NAME="team" VALUE="шахтеры">
шахтеры <BR>
<INPUT TYPE="radio" NAME="team" VALUE="ковбои"> ковбои
<BR>
<INPUT TYPE="radio" NAME="team" VALUE="викинги"> викинги
<BR>
</FORM><HR>
</BODY></HTML>

```

Элемент ввода **SELECT** позволяет использовать при вводе списки с прокруткой и выпадающее меню. Для определения списка пунктов используется элемент **<OPTION>** и необязательные атрибуты **MULTIPLE**, **NAME**, **SIZE**.

Атрибут **SELECTED** устанавливает значение элемента для первоначального выбора. Атрибут **VALUE** указывает на значение, возвращаемое формой после выбора данного пункта.

```

<!--пример # 17: формы. Элемент SELECT-->
<HTML>
<FORM><SELECT NAME="flower">
<OPTION>chocolate
<OPTION>vanila
<OPTION VALUE="Banana">Banana
<OPTION SELECTED>cherry
</SELECT> </FORM>
</HTML>

```

Элемент **<INPUT TYPE=RESET>** используется для создания кнопки Reset, по которой можно щелкнуть мышкой и вернуться к начальным значениям полей.

Элемент **<INPUT TYPE=SUBMIT>** используется для создания кнопки, по которой можно щелкнуть и отправить введенные данные в виде сообщения по указанному адресу. Дополнительный атрибут **NAME** устанавливает название кнопки **submit**. Атрибут **VALUE** хранит значение переменной поля формы.

[illegible]

```

<INPUT type="image" src="knopka1.gif">
<H3> Кнопка очистки формы </h3>
<INPUT type="reset" value="Очистка">
<H3> Файл </h3>
<INPUT type="file" name="photo" accept="image/*">
<HR color="blue">
<H2>Элемент SELECT </h2>
<SELECT multiple>
<OPTION value=a>Первый
<OPTION value=b>Второй
<OPTION value=c>Третий
<OPTION value=d>Четвертый
</select>
<HR color="blue">
<H2>Элемент TEXTAREA
<TEXTAREA rows=5 cols=30>
Область для ввода текста
</textarea></h2>
<HR color="blue">
<H2>Кнопка с надписью и рисунком</h2>
<BUTTON name="submit" value="submit" type="submit">
Надпись<IMG src="gif1.gif" alt="Рисунок"></button>
<HR color="blue">
<H2>Группа полей</h2>
<FIELDSET>
<LEGEND>Заголовок группы</legend>
Имя: <INPUT name="imya2" type="text">
Фамилия: <INPUT name="familiya2" type="text"><BR>
Телефон: <INPUT name="telefon2" type="text"><BR>
Текст подсказки
</fieldset>
<HR color="blue">
</BODY></HTML>

```

Метатеги

Любой метатэг размещается в заголовке HTML-документа между тегами **<HEAD>** и **</HEAD>** и состоит из следующих атрибутов:

```

<META HTTP-EQUIV="имя" CONTENT="содержимое">
<META NAME="имя" CONTENT="содержимое">

```

Метатеги с атрибутом **HTTP-EQUIV** управляют действиями браузеров. В качестве параметра **"имя"** могут быть использованы следующие аргументы:

Expires – Дата устаревания: если указанная дата прошла, то запрос этого документа вызывает повторный сетевой запрос, а не подгрузку документа из кэша. Дата со значением **"0"** заставляет браузер каждый раз при запросе проверять, изменялся ли этот документ. Например:

```

<META HTTP-EQUIV="expires" CONTENT="Sun, 3 April 2005
05:45:15 GMT">

```

Pragma – контроль кэширования. Значением должно быть “no-cache”.

Content-Type – указание типа документа. Может быть расширено указанием браузеру кодировки страницы (charset). Например:

```
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=windows-1251">
```

Content-language – указание языка документа. Комбинация поля **Accept-Language** (посылаемого браузером) с содержимым **Content-language** может быть условием выбора сервером того или иного языка.

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

Refresh – определение задержки в секундах, после которой браузер автоматически обновляет документ. Дополнительная возможность – автоматическая загрузка другого документа.

```
<META HTTP-EQUIV="Refresh" Content="3,
URL=http://www.bsu.iba.by/cgi-bin/news.pl">
```

Window-target – определяет окно текущей страницы; может быть использован для прекращения появления новых окон браузера при применении фреймовых структур.

```
<META HTTP-EQUIV="Window-target" CONTENT="_top">
```

Ext-cache – определяет имя альтернативного кэша

```
<META HTTP-EQUIV="Ext-cache" CONTENT=
"name=/some/path/index.db; instructions=User nstructions">
```

PICS-Label – Platform-Independent Content rating Scheme. Обычно используется для определения рейтинга “взрослости” содержания (**sex**), однако может использоваться для других целей.

Управление индексацией страницы для поисковых роботов осуществляется с использованием атрибута **NAME**.

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

Возможные значения: **ALL, NONE, INDEX, NOINDEX, FOLLOW, NOFOLLOW**.

Description – краткая аннотация содержания документа. Используется поисковыми системами для описания документа.

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

Keywords – используется поисковыми системами для индексирования документа. Обычно здесь указываются синонимы к словам в заголовке **title** или альтернативный заголовок.

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

Document-state – управление индексацией страницы для поисковых роботов. Определяет частоту индексации – или один раз индексировать, или реиндексировать документ регулярно.

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

Возможные значения: **Static, Dynamic**

URL – управление индексацией страницы для поисковых роботов. Определяет частоту индексации: или один раз индексировать, или реиндексировать документ регулярно.

```
<META NAME="URL" CONTENT="absolute_url">
```

Author – обычно имя автора, формат произвольный.

Generator – обычно название и версия редактора, с помощью которого создана эта страница.

Copyright – обычно описание авторских прав на документ.

Resource-type – текущее состояние данного файла. Важен для поисковых систем: если его значение **document**, то поисковая система приступает к индексированию.

Каскадные таблицы стилей

Таблицы стилей (CSS) позволяют отделить содержание документа от его форматирования и отображения. HTML-документы могут содержать внедренные стили непосредственно внутри документа или импортировать стили из связанных таблиц стилей, находящихся в файлах с расширением CSS. Элемент **META** указывает тип документа в виде:

```
<META http-equiv="Content-Style-Type" content="text/css">
```

При использовании внедренных стилей для установки стиля отдельного элемента HTML в этом элементе используется атрибут **style**. В следующем примере устанавливаются цвет и размер шрифта для отдельного параграфа и заголовка:

```
<P style="font-size: 12pt; color: fuchsia">
Размер шрифта 12 пикселей </P>
<H1 style="text-decoration:underline">
Текст с подчеркиванием </H1>
<H2 style="color: red">
Текст заголовка красным цветом</H2>
```

Объявление значений свойств имеет вид **"name: value"**.

Для того чтобы стили относились к нескольким элементам документа, например к одному или всем **P**-элементам, **H1**-элементам, гиперссылкам, используется блок **STYLE**. Блок **STYLE** размещается в секции **HEAD** документа. Следующий стиль обводит границы вокруг каждого **H1**-элемента и центрирует его на странице. Кроме этого, устанавливается стиль параграфа и гиперссылки.

```
<!--пример # 19: стиль-->
<HEAD>
<STYLE type="text/css">
H1 {border-width: 1; border: solid; text-align: center}
P { color: blue}
a: hover {color: red; text-decoration: overline}
</STYLE></HEAD>
```

Для гиперссылок устанавливаются следующие значения стилей:

a: hover – стиль меняется при наведении курсора;
a: active – стиль меняется при щелчке левой кнопкой мыши;
a: visited – стиль меняется после посещения;
a: link – непосещенная ссылка.

Данные о стиле размещаются в фигурных скобках.

В следующем примере стиль относится к определенным **H1**-элементам. Чтобы скрыть таблицы стилей от старых программ просмотра, их помещают внутрь HTML-комментария:

```
<!--пример # 20: стиль в комментарии-->
<HEAD><STYLE type="text/css">
<!--маскируем
H1.myclass {border-width: 1; border: solid;text-align: center} -->
</STYLE></HEAD>
<BODY>
<H1 class="myclass">H1 is affected by our style</H1>
<H1> This one is not affected by our style </H1>
</BODY>
```

Два тега – **DIV** и **SPAN**, играющие роль скобок, используются, чтобы применить стили к ограниченному фрагменту документа. В следующем примере элемент **SPAN** используется, чтобы вывести несколько слов параграфа прописными буквами и установить стили для других параграфов.

```
<!--пример # 21: применение тегов DEV и SPAN-->
<HEAD><STYLE type="text/css">
SPAN.sc-ex { font-variant: small-caps }
</STYLE></HEAD><BODY>
<P><SPAN class="sc-ex">The first</SPAN> few words are in
small-caps.</P>
<P>Это<SPAN style="font-style:italic"> курсив</SPAN></p>
<P>Это<SPAN style="text-transform-style:uppercase">верхний
регистр </SPAN> </p>
</BODY>
```

В следующем примере используется **DIV** и атрибут **class**, чтобы установить правила для двух параграфов.

```
<!--пример # 22: стиль для фрагментов-->
<HEAD><STYLE type="text/css">
DIV.Abstract { text-align: justify }
</STYLE></HEAD><BODY>
<DIV class="Abstract">
<P>The Chieftain product range is our market winner for the
coming year. This report sets out how to position Chieftain
against competing products.
<P>Chieftain replaces the Commander range, which will
remain on the price list until further notice.
</DIV>
<P style="position:absolute"; top:125px; left:200px > Про-
стой текст для позиционирования, на который накладывается
изображение </p>
<DIV style="position:absolute"; top:125px; left:200px >
 </DIV>
</BODY>
```

Для третьего параграфа в примере, в котором на текст накладывается изображение, установлено позиционирование: **position: absolute** – точка отсчета: левый угол окна; **top** – вертикальное, **left** – горизонтальное смещение от точки отсчета.

Внешние таблицы стилей позволяют установить стили для нескольких документов, сохранить в файле **.css** и затем изменять их без модификации документа. При этом используются следующие атрибуты элемента **LINK**:

Значение **ref** устанавливается на URL файла стилей. Значение атрибута **type** определяет тип таблицы стилей. Атрибут **rel** устанавливается в таблицу стилей **stylesheet**. Например:

```
<LINK href="mystyle.css" rel="stylesheet" type="text/css">
```

В следующем примере таблица, помещенная в файл **special.css**, устанавливает цвет текста в параграфе зеленым, а границу – красным:

```
P.special {
  color: green;
  border: solid red;
}
```

Эту таблицу стилей можно связать с HTML-документом с помощью элемента **LINK**:

```
<!--пример # 23: установка стиля для HTML-документа-->
```

```
<HTML><HEAD>
```

```
<LINK href="special.css" rel="stylesheet" type="text/css">
```

```
</HEAD><BODY>
```

```
<P class="special">paragraph should have green text.
```

```
</BODY></HTML>
```

В контексте использования Java-технологий можно отметить три возможности использования HTML:

1. Использование тегов **<applet>** **</applet>** для включения java-апплетов в HTML-документ.
2. Использование форм HTML и методов **GET** и **POST** для пересылки запросов и информации из форм серверу для обработки сервлетами.
3. Ответы клиенту, пересылаемые серверу на основании выполнения сервлетов и JSP, также конвертируются в HTML-документ и отображаются на стороне клиента.