26.10.2021 МДК0301 Тема Введение в HTML

**Тема лекции Редакторы HTML – документа. Работа с текстом. ГиперСсылки HTML. Списки HTML.**

**Переходим к более подробному рассмотрения HTML.**

**Редакторы HTML – документа.**

HTML-документы могут быть созданы с помощью любого текстового редактора, например с помощью Блокнота фирмы Microsoft, и т.д.

Весь документ, созданный на языке html, состоит из тэгов. Когда браузер загружает web-страницу и встречает в тексте тэг, он выполняет действие, связанное с этим тэгом. Если html-файл, загружен в текстовом редакторе, то на экране отображается текст документа. Если же файл загружен в браузер, то на экране отобразится отформатированная web-страница.

**Работа с текстом.**

**Форматирование текста HTML - документа**

Текст – это единственный объект Web – страницы, который не требует специального определения, т.е. произвольные символы интерпретируются по умолчанию как текстовые данные.

:

**Форматирование гипертекста**

Язык HTML поддерживает как логический (logical), так и физический (physical) стили форматирования содержимого документа. Использование логического форматирования указывает на назначение данного фрагмента текста, а при физическом форматировании досконально задается его внешний вид. По возможности стоит использовать логические стили, так как они позволяют браузеру выбрать наиболее подходящий документу вид. Использование логических стилей также поможет читателю разобраться в структуре документа. Физический стиль используется в основном программами, конвертирующими текстовые файлы, содержащие физическое форматирование, в HTML, так как логическое форматирование документа невозможно выполнить автоматически.

**Логическое форматирование**

Хотя язык HTML включает теги для достижения различных шрифтовых эффектов (полужирный шрифт, курсив, подчеркнутый шрифт), не все браузеры их поддерживают. Однако все браузеры поддерживают тот или иной способ выделения текста. Поэтому использование логического форматирования текста в любом случае приведет к выделению программой просмотра различных частей текста и выявит структуру документа.

Говоря о логической разметке текста, можно выделить две основные части:

* выделение заголовков в документе;
* логическое выделение *элементов текста*.

Название документа, задаваемое с помощью тега <TITLE>, не выводится на экран как часть документа. Чтобы отобразить название используется один из тегов заголовка. Заголовки в типичном документе разделяются по уровням. Язык HTML позволяет задать шесть уровней заголовков: H1 (заголовок первого уровня), Н2, Н3, H4, Н5 и H6. Заголовок первого уровня имеет обычно больший размер и насыщенность по сравнению с заголовком второго уровня.

Если вы посмотрите на эту главу, то "Логическое форматирование" - заголовок третьего уровня, "Форматирование гипертекста" - второго, а "Основы языка HTML" - первого. На практике заголовки четвертого и далее уровней встречаются лишь в очень больших документах.

**Пример**

<HTML>

<HEAD>

<TITLE> Заголовки </TITLE>

</HEAD>

<BODY BGCOLOR=white>

<H1> Заголовок 1 уровня</H1>

<H2> Заголовок 2 уровня</H2>

<H3> Заголовок 3 уровня</H3>

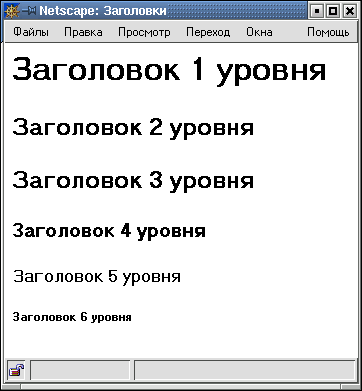
<H4> Заголовок 4 уровня</H4>

<H5> Заголовок 5 уровня</H5>

<H6> Заголовок 6 уровня</H6>

</BODY>

</HTML<



Помните, что если вы забудете поставить закрывающий тег заголовка, вид страницы будет искажен: любой из тегов заголовка автоматически вставляет пустую строку до и после себя.

Теги заголовков поддерживают атрибут ALIGN, действие которого аналогично действию такого же атрибута тега выделения абзаца.

Элементы логического выделения фрагментов текста, а также возможное оформление каждого из них приведены в таблице.

|  |  |  |
| --- | --- | --- |
| **Теги** | **Применение** | **Результат** |
| <CITE> </CITE> | <CITE> Используется для выделения цитат или названий книг и статей </CITE> | Используется для выделения цитат или названий книг и статей |
| <CODE> </CODE> | <CODE> Применяется для вывода небольшого куска программного кода </CODE> | Применяется для вывода небольшого куска программного кода |
| <EM> </EM> | <EM> Используется для выделения важных фрагментов текста </EM> | Используется для выделения важных фрагментов текста |
| <KBD> </KBD> | <KBD> Выделяет текст, вводимый пользователем с клавиатуры </KBD> | Выделяет текст, вводимый пользователем с клавиатуры |
| <*SAMP*> </*SAMP*> | <*SAMP*> Используется для выделения текста примера </*SAMP*> | Используется для выделения текста примера |
| <STRONG> </STRONG> | <STRONG> Используется для выделения очень важных фрагментов текста </STRONG> | **Используется для выделения очень важных фрагментов текста** |
| <VAR> </VAR> | <VAR> Используется для отметки имен переменных <VAR> | Используется для отметки имен переменных |
| <STRIKE> </STRIKE> | <STRIKE> Используется для отметки удаленного текста </STRIKE> | Используется для отметки удаленного текста |

Некоторые элементы логической разметки текста дают одинаковый результат. Зачем же тогда нужно такое их разнообразие? Ответ на этот вопрос содержится в названии этой группы тегов. Они предназначены для расстановки логических ударений, выделения логических частей и подчеркивания сути высказываний. Их использование весьма актуально, поскольку, вероятно, в ближайшем будущем станет возможен, например, поиск цитат в Web-пространстве, а, может быть, следующее поколение браузеров научится читать документы вслух. Программы, умеющие распознавать логические ударения, заменят монотонные речевые процессоры сегодняшнего дня.

Для выделения длинных цитат из основного текста в HTML существует тег <BLOCKQUOTE>. Этот элемент является контейнером и может содержать любые форматирующие теги.

Современные браузеры реагируют на элемент <BLOCKQUOTE> смещением текста цитаты вправо. Некоторые текстовые программы просмотра обозначают цитату символами >, располагающимися в крайнем левом столбце экрана. Так как сегодня большинство браузеров являются графическими программами, элемент <BLOCKQUOTE> позволяет авторам внести в текст некоторое визуальное разнообразие.

**Физическое форматирование**

Одним из отличий HTML-документа от документа, подготовленного на печатной машинке, является возможность форматирования текста. Язык HTML позволяет автору документа выбрать понравившийся ему шрифт, подходящий размер букв, их цвет и начертание. За все эти параметры отображения текста отвечают теги физического форматирования. Они действуют на все символы, стоящие между открывающим и закрывающим тегами.

|  |  |  |
| --- | --- | --- |
| **Теги** | **Применение** | **Результат** |
| <B> </B> | <B> Полужирный </B> | Полужирный |
| <I> </I> | <I> Курсив </I> | Курсив |
| <U> </U> | <U> Подчеркнутый </U> | Подчеркнутый |
| <TT> </TT> | <TT> Пишущая машинка </TT> | Пишушая машинка |
| <S> </S> | <S> Зачеркнутый </S> | Зачеркнутый |
| <BIG> </BIG> | <BIG> Большой </BIG> | Большой |
| <SMALL> </SMALL> | <SMALL> Маленький </SMALL> | Маленький |
| <SUP> </SUP> | Верхний - x <SUP> индекс </SUP> | Верхний - xиндекс |
| <SUB> </SUB> | Нижний - x <SUB> индекс </SUB> | Нижний - xиндекс |

Элементы физического форматирования могут быть вложенными друг в друга, хотя конечный результат зависит от браузера. При этом нужно внимательно следить, чтобы один контейнер находился целиком в другом контейнере, например,

<B><U>жирный и подчеркнутый текст</U></B>

Кроме вышеперечисленных тегов в документе может использоваться тег <FONT>, позволяющий непосредственно задать размер и цвет шрифта. Элемент FONT представляет собой контейнер, т. е. требует как открывающего, так и закрывающего тегов, и сам может использоваться внутри любого другого текстового контейнера.

После стартового тега обязательно указание атрибутов, без которых элемент не оказывает никакого влияния на текст, помещенный в контейнер.

Атрибут FACE позволяет указать тип шрифта, которым программа просмотра выведет ваш текст (если таковым располагает пользователь). Если нужного шрифта нет, программа проигнорирует запрос и будет использовать шрифт, установленный по умолчанию.

Этот атрибут позволяет указать как один, так и несколько шрифтов (через запятую). Весь список будет просмотрен слева направо и первый из имеющихся на машине пользователя будет использован для вывода документа.

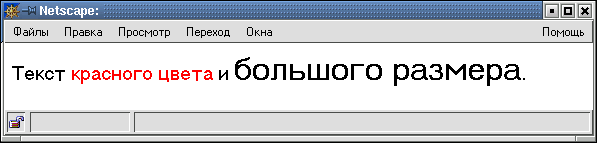
Атрибут SIZE служит для указания размера шрифта в условных единицах от 1 до 7. Считается, что размер "нормального" шрифта соответствует числу 3. Размер может быть как *абсолютной величиной* ( SIZE=5 ), так и относительной ( SIZE=+2 ). Во втором примере текущий размер шрифта увеличивается на 2.

Атрибут COLOR устанавливает цвет шрифта, который может быть задан как в формате RGB, так и указанием имени.

**Пример**

Текст <FONT COLOR=red> красного цвета </FONT> и

<FONT SIZE=7> большого размера </FONT>



**Предварительно отформатированный текст**

Любой фрагмент текста, расположенный между тегами <PRE> и </PRE>, разбивается на строки и абзацы в точном соответствии с тем, как вы его вводили. Обычно при выводе на экран предформатированного (preformatted) текста используется моноширинный шрифт (так называется шрифт печатной машинки). Этот тег можно использовать, например, для печати стихов, в которых взаимное расположение строк задал сам автор, или для изображения "картинки", состоящей из символов. Другим важным применением тега <PRE> является вывод на экран текстов компьютерных программ (на языках Java, C++ и др.), в которых традиционно используют отступы, дополнительные пробелы и пустые строки для выделения структуры исходного кода.

Текст внутри контейнера <PRE> подчиняется действиям тегов <P> и <BR> и поддерживает теги заголовков.

К содержимому контейнера <PRE> могут применяться любые элементы физического и логического форматирования, но некоторые моноширинные шрифты не содержат наборов символов полужирного и курсивного начертания.

**Пример**

Рассмотрим фрагмент HTML-документа, содержащий предварительно отформатированный текст:

<PRE>

Здесь можно ставить <B>сколько</B> угодно

<EM>пробелов</EM>,

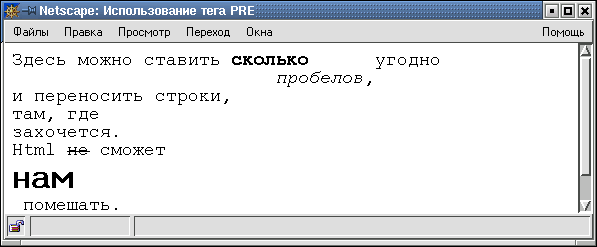
и переносить строки,

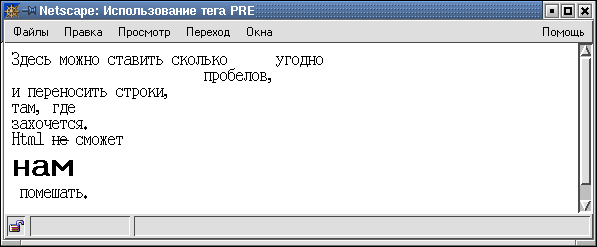
там, где

захочется. <P>Html <S>не</S> сможет <H1>нам</H1> помешать.

<PRE>

В окне браузера количество пробелов и разрывы строк будут сохранены, к ним добавятся разрывы строк, вызванные использованием тегов <P>, <BR> и <H1>. Первый из приведенных ниже рисунков соответствует заданию в качестве моноширинного в настройках браузера шрифта Courier (Adobe), а для второго использовался шрифт Fixed (Sony).





**Гиперссылки(Ссылки)**

Несмотря на то, что в состав *HTML*-документа входят самые различные компоненты, можно сказать, что **гипертекстовые ссылки** - основа *WWW*. Если бы *Web*-страницы не ссылались друг на друга, содержимое *Web* превратилось бы в обычный набор файлов, не связанных между собой.

Для создания гипертекстовой ссылки используется пара тегов <A>... </A>. Фрагмент текста, изображение или любой другой *объект*, расположенный между этими тегами, отображается в окне браузера как гипертекстовая *ссылка*. Активация такого объекта приводит к загрузке в окно браузера нового документа или к отображению другой части текущей *Web*-страницы. Гипертекстовая *ссылка* формируется с помощью выражения.

<A HREF="URL\_ресурса">фрагмент документа</A>

HREF здесь является обязательным атрибутом, *значение* которого и есть URL-адрес запрашиваемого ресурса. Кавычки в задании значения атрибута HREF не обязательны.

**Текстовые указатели**, т. е. фрагменты текста, являющиеся ссылками, не отличаются разнообразием внешнего вида. Обычно такой *указатель* представляет собой *слово* или слова, подчеркнутые *прямой* линией. Цвет указателя может регулироваться автором и *установками программы* просмотра.

**Пример**

<A HREF="example.html">Ссылка</A>

**Графические указатели**, т. е. изображения-ссылки, выделяются рамкой того же цвета, что и текстовые указатели.

**Пример**

<A HREF="example.html"><IMG SRC="*pict*.gif"></A>

**Абсолютный и относительный URL**

В разделе, посвященном компьютерным сетям и принципам поиска информации в Интернет, было введено понятие URL (универсального локатора ресурса), однозначно определяющего расположение объекта в глобальной сети. В общем случае URL имеет следующий вид (части, заключенные в квадратные скобки, не обязательны и могут быть опущены):

протокол://адрес\_узла[:порт]/путь/файл[#метка]

В таблице ниже приведены назначения компонент URL.

|  |  |
| --- | --- |
| Компонента | Назначение |
| протокол | Обозначение одного из протоколов, используемых для обращения к ресурсу, возможные значения: http, ftp, file и др. |
| адрес\_узла | *Доменное имя* или IP-адрес компьютера в сети Интернет |
| порт | Порт, по которому клиент обращается к серверу для установления соединения; указывается только в случае обращения к нестандартному порту |
| путь | Путь к требуемому ресурсу |
| файл | Имя файла, содержащего HTML-документ или другой ресурс |
| метка | Позиция в документе, начиная с которой он отображается в окне браузера |

URL, заданный в таком виде, называется **абсолютным URL**, так как он полностью описывает расположение ресурса в глобальной сети.

**Пример**

При обращении к конкретному ресурсу порт и позиция в документе часто не указываются:

http://www.*ctc*.msiu.ru/*education*/book/index1.html

**Пример**

Рассмотрим ссылку на **домашнюю страницу** (Home Page) конкретного пользователя. Предположим, что на сервере **www.msiu.ru** зарегистрирован человек с пользовательским именем **(login) ivanov**. Домашней страницей называют файл **index.html**, находящийся в директории с именем **public\_html**, которая, в свою очередь, должна располагаться в *домашней директории* пользователя. Тогда ссылка на домашнюю страницу этого человека может быть задана в виде

<A HREF="http://www.msiu.ru/~ivanov">Текст ссылки</A>

Обратите внимание, что имя самого файла при такой записи не указывается. Если потребуется сослаться на какой-либо другой документ данного пользователя, расположенный в директории **public\_html**, например, **photo.html**, то ссылка примет вид:

<A HREF="http://www.msiu.ru/~ivanov/photo.html">Текст</A>

**Относительный URL** описывает положение ресурса, на который указывает ссылка, относительно URL текущего документа.

**Пример**

При задании относительных ссылок указывается путь по файловому дереву до того места, где находится требуемый HTML-ресурс. Вот ссылка на документ **image.html**, размещенный в текущем каталоге: <A HREF="image.html">Ссылка</A>.

Если файл **pict.html** лежит в родительском по отношению к текущему документу каталоге, то следует использовать запись

<A HREF="../*pict*.html">Текст ссылки</a>

Если же требуемый документ находится в поддиректории Picture, то используется запись, аналогичная следующей

<A HREF="Picture/*pict*.html">Текст ссылки</A>

**Гиперссылки в пределах одного документа**

Другая форма тега <A> предназначена для присваивания имени некоторому фрагменту документа HTML:

<A NAME="имя">фрагмент документа</A>

Тег <A> часто называют тегом **якоря** (anchor). Если якорь применяется для добавления метки (имени) к фрагменту документа, то его называют **именованным якорем**.

Для того чтобы сослаться на фрагмент, которому присвоено имя, используется следующая форма тега <A>:

<A HREF="URL\_ресурса#имя">Текст ссылки</A>

Если нужно сослаться на фрагмент текущего документа, то URL-ресурса можно опустить:

<A HREF="#имя">Текст ссылки</A>

**Пример**

Пусть в документе с именем **book.html** заголовку главы 2 присвоено имя **chapter2**:

<A NAME="chapter2">Глава 2</A>

Тогда ссылка на эту главу, расположенная в этом же документе, будет иметь вид:

<A HREF="#chapter2">Глава 2</A>

Для задания ссылки на эту же метку файла **book.html** из другого файла нужно написать:

<A HREF="book.html#chapter2">Глава 2</A>

**Ссылка на почтовый ящик**

Для того чтобы создать ссылку на почтовый ящик, напишите:

<A HREF=*mailto*:адрес\_e-mail>текст ссылки2</A>

Здесь вместо **адрес\_e-mail** нужно поставить адрес почтового ящика. После того, как посетитель активизирует эту ссылку, у него на экране появится специальное окно, позволяющее послать письмо по указанному адресу.

**Пример**

<A HREF=*mailto*:sidorov@mail.msiu.ru>Почта автору</A>

**Списки**

**Создание списков в HTML – документе**

Списки отличаются от обычного текста тем, что пользователю не надо думать о нумерации его пунктов - эту задачу программа берет на себя. Если список укорачивается или дополняется новыми пунктами, нумерация корректируется автоматически. Средствами HTML можно создать любые списки

В настоящее время стандарты HTML поддерживают теги для списков: нумерованных (упорядоченных), маркированных (неупорядоченных) , списков определений. Списки и элементы списков являются блочными элементами. Это означает, что перед ними и после них автоматически добавляются пустые строки.

Язык HTML допускает вложенность любых видов списков. Для этого размещают одну пару тегов (стартовый и завершающий) внутри другой. Следует помнить о том, что все имеющиеся списки должны завершаться закрывающим тегом.

**Нумерованные списки**

Нумерованные (упорядоченные) списки используют, когда важен порядок вывода элементов списка. Браузер автоматически вставляет номера элементов по порядку, в исходном HTML-тексте номера не печатаются. Если количество элементов списка изменится (в результате удаления или добавления новых элементов), то нумерация автоматически обновится.

Весь нумерованный список заключается между парой тегов <OL> и </OL>, а каждый элемент списка расположен между тегами <LI> и </LI> (закрывающий тег </LI> может отсутствовать).

**Пример**

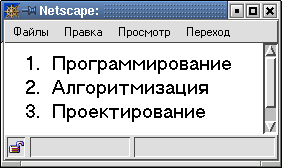
<OL>

<LI>Программирование</LI>

<LI>Алгоритмизация</LI>

<LI>Проектирование</LI>

</OL>



Тег <OL> может имеет атрибуты TYPE и START:

<OL START=n TYPE=вид\_счетчика>

Атрибут TYPE задает вид счетчика, возможные значения которого приведены в таблице, a START - начальное значение.

|  |  |
| --- | --- |
| **Значение** | **Функция** |
| A | большие латинские буквы (A,B,C...) |
| a | маленькие латинские буквы (a,b,c...) |
| I | большие римские цифры (I,II,III...) |
| i | маленькие римские цифры (i,ii,iii...) |
| 1 | арабские цифры (1,2,3...); используется по умолчанию |

**Пример**

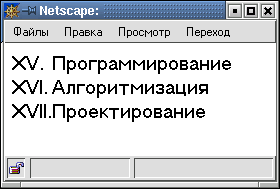
<OL TYPE=I START=15>

<LI>Программирование</LI>

<LI>Алгоритмизация</LI>

<LI>Проектирование</LI>

</OL>



**Маркированные списки**

Маркированный (неупорядоченный) список используется для представления коллекции элементов, порядок вывода которых не важен. При выводе маркированных списков браузер автоматически вставляет специальные значки (маркеры), отмечающие каждый элемент списка.

*Маркированный список* начинается стартовым тегом <UL> и завершается тегом </UL>. Каждый элемент списка начинается с тега <LI> и завершается (необязательным) тегом </LI>.

**Пример**

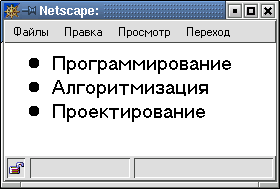
<UL>

<LI>Программирование</LI>

<LI>Алгоритмизация</LI>

<LI>Проектирование</LI>

</UL>



Тег <UL> имеет атрибут TYPE, определяющий внешний вид маркера:

<UL TYPE=тип\_маркера>

Значение атрибута TYPE может быть одним из следующих: *disc* (круг - форма по умолчанию), circle (окружность) или square (квадрат).

**Пример**

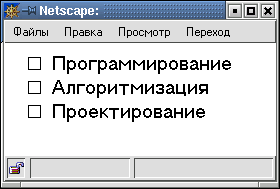
<UL TYPE=square>

<LI>Программирование</LI>

<LI>Алгоритмизация</LI>

<LI>Проектирование</LI>

</ul>



Атрибут TYPE применяется и в теге <LI> для изменения формы маркера перед конкретным элементом списка.

**Пример**

<UL>

<LI TYPE=circle>

Программирование

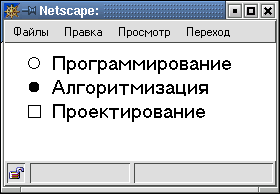
<LI TYPE=disk>

Алгоритмизация

<LI TYPE=square>

Проектирование

</ul>



Тег <LI> обеспечивает вывод маркера и разделение элементов списка. Если хочется использовать нестандартные маркеры, то тег <LI> не указывается. Для выделения элементов списка в этом случае используются какие-либо картинки или символы, а тег <BR> обеспечивает переход к следующему элементу списка.

**Пример**

<UL>

<IMG SRC=arr.gif ALIGN=top>

<nbr>Программирование<BR>

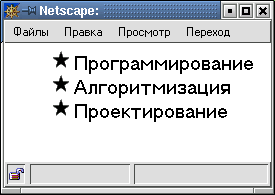
<IMG SRC=arr.gif ALIGN=top>

<nbr>Алгоритмизация<BR>

<IMG SRC=arr.gif ALIGN=top>

<nbr>Проектирование<BR>

</UL>



**Список определений**

Список описаний (список определений) начинается с тега <DL> и завершается тегом </DL>. Данный список служит для создания списков типа "термин" - "описание". Термин автоматически размещается у левой границы страницы, а их определения смещены относительно них вправо. Каждый термин обозначается тегом <DT>, а его описание - тегом <DD>.

<DL>

<DT>Отдел маркетинга

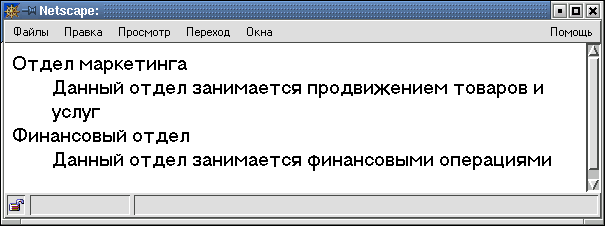
<DD>Данный отдел занимается продвижением товаров и услуг

<DT>Финансовый отдел

<DD>Данный отдел занимается финансовыми операциями

</DL>

В результате браузер покажет следующее:



Отметим, что все вышеперечисленные виды списков имеют атрибут СОМРАСТ, позволяющий выводить списки в более компактном виде.

**Вложенные списки**

Любой список может быть частью другого списка, вложен в другой список. Считается полезным использование сдвигов при подготовки текста исходного HTML-документа, чтобы четко представлять уровни вложенности списков.

<OL>

<LI>Дирекция</LI>

<UL>

<LI>Иванов И.И.</LI>

<LI>Петров К.В.</LI>

</UL>

<LI>Отдел маркетинга</LI>

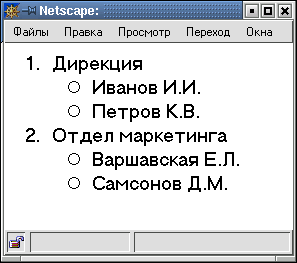
<UL>

<LI>Варшавская Е.Л.</LI>

<LI>Самсонов Д.М.</LI>

</UL>

</OL>



При выводе вложенных маркированных списков браузер автоматически проставляет маркеры перед элементами, находящимися на разных уровнях вложенности.

Для вложенных нумерованных списков браузер, к сожалению, не изменяет тип нумерации. По умолчанию каждый уровень внутри такого списка будет отмечен арабскими цифрами. Для получения списка в другом формате потребуется вручную проставить каждую метку, используя атрибут TYPE.

<OL TYPE=I>

<LI>Дирекция</LI>

<OL>

<LI>Иванов И.И.</LI>

<LI>Петров К.В.</LI>

</OL>

<LI>Отдел маркетинга</LI>

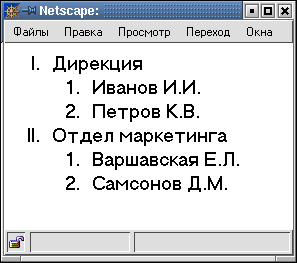
<OL>

<LI>Варшавская Е.Л.</LI>

<LI>Самсонов Д.М.</LI>

</OL>

</OL>



**Вопросы к домашнему заданию:**

1. **Редакторы HTML?(перечислить)**
2. **Какой элемент является основой в HTML? Почему?**
3. **Виды Гиперссылок? (перечислить)**
4. **Для чего нужны Списки?**
5. **Перечислите основные виды Списков.**

Присылать **kuzn117@yandex.ru Срок 27.10.2021**