

Конспекты лекций по курсу

«Введение в информатику и системы программирования», 1 семестр

С.А. Немнюгин, направление «Прикладные математика и физика»

Лекция 4

Компьютерные сети и Интернет

Введение в HTML

HTML (HyperText Markup Language) – язык гипертекстовой разметки.

Всемирная паутина (WWW – WorldWide Web) состоит из HTML-документов – текстовых документов, размеченных с помощью языка HTML.

HTML-документ – это обычный текстовый документ, в который внесена разметка. HTML-документ состоит из элементов, которые выделяются разметкой. Каждый элемент имеет имя, и может иметь *атрибуты*, которые определяют его свойства (например, цвет или размер)

Элементы HTML бывают двух типов:

1. простые элементы, которые существуют сами по себе;
2. элементы-контейнеры, которые могут включать размечаемый текст или другие элементы (как простые, так и контейнеры).

Разметка осуществляется с помощью *ярлыков*. Ярлык заключен в угловые скобки (<, >), и не виден при просмотре HTML-документа. Непосредственно за открывающей угловой скобкой (без пробелов!) следует имя ярлыка, а за именем через пробел – атрибуты.

Пример определения ярлыка с именем BODY и 8-ю атрибутами (onLoad, onUnload, BACKGROUND, BGCOLOR, TEXT, LINK, VLINK и ALINK):

```
<BODY onLoad="parent.moveDot (UNIT) "
onUnload="parent.lessonLog (LESSON, UNIT, ' STOP ' ) ; "
BACKGROUND=" "
BGCOLOR="#d2bc93"
TEXT="#000000"
LINK="#006464"
VLINK="#800080"
ALINK="#ff0000"
>
```

Простой элемент размечается единственным ярлыком, имя которого совпадает с именем элемента.

Пример:

Изображение в HTML-документе определяется элементом IMG и размечается с помощью ярлыка

Элемент-контейнер размечается двумя ярлыками – открывающим и закрывающим. Между этими ярлыками помещается содержимое контейнера (например, текст абзаца).

Имя открывающего ярлыка элемента-контейнера совпадает с именем этого элемента, а имя закрывающего ярлыка начинается с косой черты (/), за которой следует имя элемента.

Пример:

Фрагмент текста, набранный моноширинным шрифтом – это элемент-контейнер с именем TT:

```
<TT>Чтоб вам столько работать, сколько хотите получать! </TT>
```

Структура HTML-документа

HTML-документ – это контейнер с именем HTML. Он начинается с ярлыка <HTML> и заканчивается ярлыком </HTML>:

```
<HTML>
.....
</HTML>
```

В этом контейнере лежат два других элемента-контейнера HEAD и BODY:

```
<HTML>
<HEAD>
Заголовок – предназначен для размещения служебной информации о
документе
</HEAD>
<BODY>
Говорите, говорите, я всегда зеваю, когда мне интересно.
</BODY>
</HTML>
```

Минимальный документ:

```
<HTML>
<HEAD>
  <TITLE>Простейший HTML-документ </TITLE>
</HEAD>
<BODY>
Говорите, говорите, я всегда зеваю, когда мне интересно.
</BODY>
</HTML>
```

Заглавие HTML-документа

Заглавие HTML-документа используется поисковыми системами. Оно появляется на верхней рамке окна браузера, в истории путешествий по Интернету, в закладках и др. Многие из этих мест имеют ограниченную длину (как правило, не более 60 символов), поэтому заглавие (элемент TITLE) должно быть *коротким* и *информационно емким*. Заглавие должно содержать как русский, так и английский текст.

Кодировка в HTML-документе

Указание кодировки требуется для корректного отображения кириллических (русских) текстов. С помощью служебного, невидимого элемента <META>, помещенного в контейнер HEAD, указать браузеру, какую кодировку следует использовать:

```

<HTML>
<HEAD>
<TITLE>This is my first document - Это мой первый документ</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html, charset=windows-
1251">
</HEAD>
<BODY>
Говорите, говорите, я всегда зеваю, когда мне интересно!
</BODY>
</HTML>

```

Абзац в HTML-документе

Абзац – это элемент логической структуры текста. Для него есть соответствующий элемент HTML. Этот элемент – контейнер, поскольку он содержит текст абзаца; следовательно, этот элемент должен иметь открывающий и закрывающий ярлыки.

```

<HTML>
<HEAD>
<TITLE>This is my first document - Это мой первый документ</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html, charset=windows-
1251">
</HEAD>
<BODY>
Первый абзац!
<P>
Второй абзац!
</P>
</BODY>
</HTML>

```

Логическая структура HTML-документа

HTML допускает использование заголовков шести уровней (часть, раздел, глава, пункт, подпункт, подподпункт). Для разметки заголовков используются элементы-контейнеры:

```

<H1> Заголовок части </H1>
<H2> Заголовок раздела </H2>
<H3> Заголовок главы </H3>
<H4> Заголовок пункта </H4>
<H5> Заголовок подпункта </H5>
<H6> Заголовок подподпункта </H6>

```

Логическая структура документа может также содержать *выделенный текст* (текст, значение которого более важно для автора):

```
<EM>выделенный текст</EM>
```

и "сильно выделенный текст":

```
<STRONG>сильно выделенный текст</STRONG>
```

а также и другие структурные элементы: списки, цитаты, определения и др.

```

<HTML>
<HEAD>
  <TITLE>This is my first document - Это мой первый
документ</TITLE>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html,
charset=windows-1251">
</HEAD>
<BODY>
<EM>За обыском начинается допрос,</EM><STRONG> который тоже является
не самой</STRONG>
</BODY>
</HTML>

```

Простейшая оформительская разметка HTML-документа

Выделение фрагментов текста

Простейшая оформительская разметка позволяет задать **полужирный** или *наклонный* шрифт (но браузер может проигнорировать одновременное задание *полужирности и наклонности* шрифта):

```

<B>полужирный</B>
<I>наклонный</I>
<B><I>полужирный и наклонный</I></B>
<I><B>наклонный и полужирный</B></I>

```

Кроме наклонного и полужирного начертания символов можно определить индексы:

```

<SUP>верхний</SUP>индекс
<SUB>нижний</SUB>индекс

```

Шрифтовое оформление

Элемент-контейнер, позволяющий задать размер цвет шрифта независимо от логической или оформительской разметки:

```
<FONT>...</FONT>
```

Элемент имеет несколько атрибутов. Важнейшие из них: SIZE и COLOR, задающие, соответственно, размер и цвет шрифта.

Размер шрифта

Атрибут SIZE может иметь значения двух типов – абсолютного и относительного. Абсолютное значение атрибута – это число от 1 до 7:

```

<FONT SIZE=" 1 " > - шрифт минимального размера
<FONT SIZE=" 2 " > - шрифт немного меньше стандартного
<FONT SIZE=" 3 " > - шрифт стандартного размера
<FONT SIZE=" 4 " > - шрифт увеличенного размера
<FONT SIZE=" 5 " > - шрифт БОЛЬШОГО размера
<FONT SIZE=" 6 " > - шрифт очень большого размера
<FONT SIZE=" 7 " > - шрифт ГИГАНТСКОГО размера

```

Относительное значение атрибута – это также число от 1 до 7, но со знаком + или –.

```

<FONT SIZE="+1"> - шрифт чуть больше текущего размера
<FONT SIZE="+2"> - шрифт еще немного больше
<FONT SIZE="+3"> - шрифт еще более крупного размера
<FONT SIZE="+4"> - шрифт ГИГАНТСКОГО размера
<FONT SIZE="+5"> - дальше увеличивать шрифт некуда
(если размер шрифта основного текста - "базового шрифта" - равен 3)
<FONT SIZE="+6"> - дальше увеличивать шрифт некуда
<FONT SIZE="+7"> - дальше увеличивать шрифт некуда

```

Цвет шрифта

Цвет шрифта задается значением атрибута COLOR элемента FONT:

```

<FONT COLOR="teal">
ИЛИ
<FONT COLOR="#008080">

```

Разметка связей

Разметка связей выполняется с помощью ярлыка (элемента-контейнера) с именем "A", который следует внести в первый документ:

```

<HTML>
<HEAD>
<TITLE>The first document</TITLE>
</HEAD>
<BODY>
<H1>Это документ #1</H1>
Далее идет
<A HREF="2.html">связь со вторым документом</A>.
</BODY>

```

При щелчке левой кнопкой мыши в момент, когда курсор имеет форму руки, связь активизируется и текущий документ («первый») заменяется тем, на который указывает связь («второй»).

Для внешних связей следует указывать полный URL, включающий и протокол:

```

<HTML>
<HEAD>
<TITLE>The second document</TITLE>
</HEAD>
<BODY TEXT="#008080">
<H1>Это документ #2</H1>
Далее идет
<A HREF="1.html">связь с первым документом</A>.
<P>
А второй абзац указывает на
<A HREF="http://www.microsoft.com">Microsoft</A> ;- )
</BODY>
</HTML>

```

Как создать HTML-документ

Три способа создания HTML-документа:

1. набрать текст в Microsoft Word и сохранить его как HTML;
2. с помощью специализированных редакторов HTML;
3. с помощью обычных текстовых редакторов (например, "Блокнот" или <F4> в FAR Manager).

Поиск информации в Интернете

Информационная потребность. Понятие пертинентного документа

АВТОР создает ДОКУМЕНТ. У нас возникает ИНФОРМАЦИОННАЯ ПОТРЕБНОСТЬ.

В теории информационного поиска вместо слова "подходит" используют термин *ПЕРТИНЕНТНЫЙ ДОКУМЕНТ*, а вместо "не подходит" – *НЕ ПЕРТИНЕНТНЫЙ*. Слово *пертинентный* происходит от английского *pertinent*, что значит *относящийся к делу, подходящий по сути*.

Субъективно понимаемая цель информационного поиска – найти все пертинентные и только пертинентные документы (мы хотим найти "только то, что хотим, и ничего больше"). Эта цель недостижима.

Информационный шум

Оценить пертинентность документа часто можно только в сравнении с другими документами. Для того, чтобы было с чем сравнивать, необходимо некоторое количество непертинентных документов. Эти документы называются – "ШУМ".

Слишком большой шум затрудняет выделение пертинентных документов, слишком малый – не дает уверенности в том, что найдено достаточное количество пертинентных документов.

Практика показывает, что когда количество непертинентных документов лежит в интервале от 10% до 30%, ищущий чувствует себя комфортно, не теряясь в море шума и считая, что количество найденных документов – удовлетворительно.

Информационно-поисковые системы

Когда документов много, используется информационно-поисковая система (ИПС). Для общения с ИПС информационная потребность должна быть выражена средствами, которые эта ИПС "понимает" – должен быть сформулирован *запрос*:

Понятие релевантного документа

Запрос редко может точно выразить информационную потребность. Многие ИПС по разным причинам не могут определить, соответствует ли тот или иной документ запросу. Степень соответствия документа запросу называется *релевантностью*. Релевантный документ может оказаться непертинентным и наоборот.

Словарные ИПС

Основная идея словарной ИПС – создать словарь из слов, встречающихся в документах Интернета, в котором при каждом слове будет храниться список документов, из которых взято данное слово.

Логические операторы

Вместо того, чтобы говорить «Список документов содержащих слово **стол** или документов, содержащих слово **стул**», употребляются сокращенные выражения. Дальнейшее сокращение эти выражения находят в языке запросов словарных ИПС: вместо "Найти список документов содержащих слово **стол** или документов, содержащих слово **стул**, большинству словарных ИПС достаточно написать что-то вроде:

стол ИЛИ стул

Союз ИЛИ в запросе к словарной ИПС выступает в роли **ЛОГИЧЕСКОГО ОПЕРАТОРА**, связывающего множества искомых документов. Словарные ИПС используют три логических оператора: ИЛИ, И и И-НЕ ("но без"); как правило, эти операторы обозначаются одним из следующих способов:

Оператор	Полное обозначение	Сокращенное обозначение	Обозначение при простом поиске (кроме российской ИПС Rambler)
ИЛИ	OR		пробел
И	AND	&	+ (знак "плюс")
И-НЕ	AND NOT	! или &! или !&	- (знак минус)

Эти операторы имеют приоритет (прежде всего выполняется И-НЕ, затем – И, и лишь потом – ИЛИ), поэтому для составления сложных запросов могут использоваться скобки

Как правило, словарные ИПС Интернета предоставляют пользователям два интерфейса – режим *сложного запроса* (**advanced search**), в котором доступны все логические операторы, и режим простого поиска, в котором, как правило, невозможно использование скобок, и, следовательно, можно использовать не все сочетания операторов.

Правое усечение

Рассмотрим гипотетический пример поиска информации о столах. С учетом падежей слова "**стол**" и наших знаний о логических операторах, запрос к словарной ИПС мог бы выглядеть так: стол ИЛИ стола ИЛИ столу ИЛИ столе ИЛИ столом

Западные ИПС, ориентированные на английский язык, предлагают простое решение: вместо слова можно написать его начало, заменив изменяемую часть звездочкой:

стол*

Формально говоря, звездочка заменяет любое количество символов, поэтому говорят, что она обозначает правое усечение.

Стратегия поиска

Дать общий рецепт эффективной стратегии поиска информации в Интернете невозможно. Есть лишь некоторые принципы, позволяющие тратить меньше времени.

Целью использования универсальной ИПС общего назначения может быть поиск специализированной ИПС, посвященной тематике вашего поиска.

Читайте найденные документы в поисках наиболее точных терминов и связей между терминами. Возможно, вы мыслите совершенно не в тех терминах, которые используют авторы искомых документов (вспомним о культурных различиях!).

Используйте несколько ИПС. Если вы регулярно занимаетесь поиском информации по какой-то тематике, отметьте те ИПС, которые для вас наиболее эффективны.