

Лекция 1. INTERNET & WEB

Содержание

Лекция 1. INTERNET & WEB. Язык гипертекстовой разметки HTML. **Модели данных.** (8 сентября)

Лекция 2. Введение в расширенный язык разметки XML. DTD и XML-схема. (15 сентября)

Лекция 3. Описание типов, элементов и атрибутов. (22 сентября)

Лекция 4. Пример моделирования данных. Архитектура ISO/OSI (29 сентября)

Практика 1. *Моделирование документов* (17-22 сентября ?)

Практика 2. *Схема родственных отношений* (24-29 сентября ?)

Практика 3. *Выбор предметной области и ее моделирование.*

Практика 4. *XML- схема предметной области.*

В чем различие между Веб и Интернет?

Из определения в [Wikipedia](#): **“Internet – это глобальная система связанных компьютерных сетей обменивающихся данными посредством пересылки пакетов в рамках стандартизованного Интернет протокола.”**

Таким образом, определяется TCP/IP стандартами.

Веб (всемирное информационное пространство) определяется в стандарте W3C's «Архитектура всемирного информационного пространства» ([Architecture of the World Wide Web, Volume I](#)) следующим образом: **“World Wide Web (WWW, или просто Web) – это информационное пространство в котором представляющие интерес ресурсы идентифицируются глобальными идентификаторами называемыми Uniform Resource Identifiers (URI).”**

Сеть **Интернет** и информационные ресурсы (**Web**) в ней являются общественным явлением — их развитие во многом напоминает развитие цивилизации. И как любая цивилизация в своем развитии они пережили четыре ЭРЫ:

- "СТАНОВЛЕНИЯ" - 1945-1970 гг.;
- "ПЕРВОБЫТНУЮ" - 1970-1990 гг. (эра электронной почты и передачи файлов);
- "РАЗВИТУЮ" — 1991-2000 гг. (эра гипертекста - сервиса) и
- "НОВЕЙШУЮ" - 2001-2010 гг. и далее.

ЭРА СТАНОВЛЕНИЯ

1945—1960 Теоретические работы по интерактивному взаимодействию человека с машиной, появление первых интерактивных устройств и вычислительных машин, на которых реализован режим разделения времени.

1945. В статье Vannevar Bush в Atlantic Monthly описывает фото-электро-механический прибор названный Memex для расширения памяти, осуществляющий связи и следование им между документами на микрофише.

1960. J.C.R.Licklider публикует статью «Симбиоз-человек-компьютер»

1961—1970 Разработка технологических принципов коммутации пакетов. Ввод в действие ARPANet.

1962 Дж. Ликлайдер обосновывает возможность создания глобальной распределенной сети.

Леонард Кляйнрок, Пол Баран, Дональд Дэйвис и Роджер Скэнтлбери работая независимо друг от друга, в 60-е годы предложили идею создания сети, передача информации в которой осуществляется пакетами.

ЭРА СТАНОВЛЕНИЯ

1962. Douglas Englebart публикует статью «Увеличение интеллекта человека. Концептуальная рамка.»

1965. Ted Nelson придумывает термин «гипертекст» в докладе «Структура файла для сложных, изменяющихся и неопределенных данных»

1968. Douglas Engelbart демонстрирует Online System (NLS)

On December 9, 1968, Douglas C. Engelbart and the group of 17 researchers working with him in the Augmentation Research Center at Stanford Research Institute in Menlo Park, CA, presented a 90-minute live public demonstration of the online system, NLS, they had been working on since 1962. The public presentation was a session of the Fall Joint Computer Conference held at the Convention Center in San Francisco, and it was attended by about 1,000 computer professionals. This was the public debut of the computer mouse. But the mouse was only one of many innovations demonstrated that day, including hypertext, object addressing and dynamic file linking, as well as shared-screen collaboration involving two persons at different sites communicating over a network with audio and video interface.

1969. Агентство проектов расширенных исследований уполномочивает ARPANET продолжать исследования по работам в сетях.

ЭРА ПЕРВОБЫТНАЯ

1971—1980 Число узлов ARPANet возросло до нескольких десятков, проложены специальные кабельные линии, соединяющие некоторые узлы. Начинает функционировать электронная почта. О результатах работ ученые докладывают на международных научных конференциях.

1971. Ray Tomlinson создает email программу для отправки сообщений в распределенной сети.

1972. Tomlinson расширяет программу для пользователей ARPANET и использует знак @ как часть адреса.

1973. К сети были подключены первые иностранные организации из Великобритании и Норвегии. Между ними была установлена связь по электронной почте и, тем самым сеть стала международной.

1973. Роберт Меткалф разработал стандарт Ethernet для локальных сетей. В 1970-ых годах сеть в основном использовалась для пересылки электронной почты. В это время появились первые списки почтовой рассылки, новостные группы и доски объявлений. Но сеть ещё не могла легко взаимодействовать с другими сетями, построенными на других технических стандартах.

ЭРА ПЕРВОБЫТНАЯ

1974. Van Cerf и Bob Kan публикуют статью «Протокол для пакетного сетевого соединения», специфицирующую в деталях Протокол управления передачей (TCP).

1978. Часть протокола TCP публикуется отдельно как интернет протокол (IP).

К концу 1970-ых годов начали бурно развиваться протоколы передачи данных, которые были стандартизированы в 1982-1983 годах.

1981—1990 Принят стандарт протоколов TCP/IP. Министерство обороны США строит собственную сеть на основе принципов ARPANet. Происходит разделение сети на ARPANet и MILNet. Число хостов доходит до 100 000.

ЭРА ПЕРВОБЫТНАЯ

1983 Сеть ARPANet перешла с протокола TCP на использование протоколов TCP/IP, которые и в настоящее время применяется для объединения (или <наслоения>) сетей. Термин <Интернет> закрепился за сетью ARPANet и она передается под конторь NSF.

1984 Разработана эталонная модель взаимодействия открытых систем — **модель ISO/OSI**. Разработана система доменных имён DNS.

1988 Изобретён протокол IRC44, благодаря чему в Интернете стало возможно общение в реальном времени (чат).

1988 Появление первого вируса-<червя>, поражающего почту.

1989 В Европе, в стенах Европейского совета по ядерным исследованиям CERN45 родилась концепция Всемирной паутины WWW. Предложил её британский учёный **Тим Бернерс-Ли**. Он же в течение двух лет разработал протокол HTTP, **язык HTML**.

1990 Сеть ARPANet полностью прекратила своё существование, уступив место NSFNet. В том же году было зафиксировано первое подключение к Интернету по телефонной линии — так называемый <дозвон> (Dial-up access).

ЭРА РАЗВИТИЯ

1

Дец. 1991. Работа Tim Bernes-Lee по глобальному информационному пространству (Web) принята только в качестве стендового доклада на конференции Hypertext'91 (США).

10

Дец. 1992. Первый (вне Европы) Web – сервер установлен в США.

Jan. 1993. Увеличилось число браузеров; Midas, Erwise, Viola, Samba.

Mar. 1993. NSCA реализует первую альфа версию Mosaic для X-Windows

50

Apr. 1993. CERN согласился сделать использование web-протокола и кодов бесплатным.

1993. Становление российских научно-образовательных сетей.

1994. Mark Andreessen с коллегами из NSCA и основали Mosaic Communication Corp., которая позже стала Netscape.

1994. Традиционные системы дозвона начали предоставлять доступ к сети Интернет.

ЭРА РАЗВИТИЯ

В октябре **1994** **Tim Berners-Lee** основал World Wide Web Consortium (W3C) в Massachusetts Institute of Technology, Laboratory for Computer Science [MIT/LCS] в сотрудничестве с CERN, где был создан WEB.

1995. NSFNet вернулась к роли исследовательской сети, маршрутизацией всего трафика Интернета теперь занимались сетевые провайдеры, а не суперкомпьютеры Национального научного фонда.

1995. Всемирная паутина стала основным поставщиком информации в Интернете, обогнав по трафику протокол пересылки файлов FTP49. Образован Консорциум всемирной паутины (W3C50). Всемирная паутина преобразила Интернет и создала его современный облик.

1996. Подписано соглашение по созданию точки взаимного обмена IP-трафиком (Internet eXchange) в Москве (M9-IX).

К **1997.** году в Интернете насчитывалось уже около 10 миллионов компьютеров, было зарегистрировано более миллиона доменных имён. Интернет стал очень популярным средством для обмена информацией.

1999 Создание **XML** Information Set Requirements, XML Schema Requirements, Hypertext Transfer Protocol -- HTTP/1.1

ЭРА НОВЕЙШАЯ

2001—Новейшая история. Semantic Web.

Распределенные информационно-вычислительные ресурсы, GRID-технологии, интерактивные и мультимедийные технологии (телефония, видеоконференции и др.) в сети.

2004 Создание открытых стандартов XML Schema, RDF, RDF Schema, OWL, Web Services Architecture

2007 Создание протокола по открытой архитектуре сервисов и языка описания веб-сервисов (SOAP 1.2 Part 3: One-Way MEP, Web Services Description Language (WSDL) Version 2.0)

2008 Создание стандарта языка запросов SPARQL Query Language for RDF, Прототипа Базы знаний для наук о жизни

2009 OWL 2 Web Ontology Language, SKOS Simple Knowledge Organization System, Semantic Web Applications in Neuromedicine (SWAN) Ontology

2010 OWL 2 RL in RIF, XQuery 1.0 and XPath 2.0 Data Model

Язык гипертекстовой разметки HTML

Стандарты

2010-11-23

XHTML™ Basic 1.1 - Second Edition

XHTML Basic 1.1 defines a profile of XHTML designed for small devices.

2010-11-23

XHTML™ 1.1 - Module-based XHTML - Second Edition

XHTML 1.1 redefines XHTML 1.0 based on a modular design, with support for Ruby Annotation.

2010-11-23

XHTML-Print - Second Edition

XHTML Print defines a profile of XHTML designed for printing from mobile devices to low-cost printers.

2008-10-14

RDFa in XHTML: Syntax and Processing

RDFa in XHTML allows to express structured data through markup attributes, based on an RDF modelling.

2003-01-09

Document Object Model (DOM) Level 2 HTML Specification

DOM Level 2 HTML defines a programming interface to access and update the content and structure of HTML and XHTML documents.

2002-08-01

XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)

XHTML 1.0 is a reformulation of HTML 4 as an XML language.

1999-12-24

HTML 4.01 Specification

HTML 4.01 defines the semantics and structure of HTML documents.

Структурированный текст

Данные представляют собой иерархию элементов. Каждый элемент имеет имя, атрибуты и несет некую информацию. Большинство элементов представлены в документе в виде начальной метки, указывающей имя и атрибуты. Далее следует собственно содержание элемента. И наконец, заканчивает все это конечная метка. Например,

```
<HTML>
  <TITLE>    Простой блок данных    </TITLE>
  <H1>      Пример структуры    </H1>
  _____
  <P>      Обычный параграф
  <UL>
    <LI>Первая запись, включающая
      <A NAME="URI">          текст          </A>
    <LI>Вторая запись
  </UL>
</HTML>
```

Пример 1

Структурированный текст

Метки

Каждый элемент начинается с метки, меткой же и заканчивается каждый непустой элемент. Начальные метки выделяются символами `<` и `>`, а конечные - символами `</` и `>`.

Имена

Имя элемента следует в метке сразу за символом открытия `<`. Имя начинается с буквы, за которой могут следовать еще 33 буквы, цифры, пробела или дефиса. В именах игнорируется разница между прописными и строчными буквами.

Атрибуты

Начальная метка позволяет вставить между именем и символом `>` пробелы и атрибуты. Атрибут состоит из имени, символа равенства и значения. Слева и справа от символа равенства можно оставлять пробелы.

Значение атрибута указывается в виде строки, заключенной в одинарные или двойные кавычки.

Разделители строк

Разделитель строки рассматривается как элемент разметки (и соответственно игнорируется), если он помещен в элемент в начале или конце поля данных. Это позволяет Вам писать либо

```
<pre>some example text</pre>
```

либо

```
<pre>  
    some example text  
</pre>
```

Оба примера будут обработаны одинаковым образом. Кроме того, если строка документа не является пустой, но все равно не содержит информацию для анализатора, она будет игнорироваться полностью.

Например, элемент

```
<pre>
```

```
    <!-- this line is ignored, including the linebreak character -->  
    first line
```

T hird line<!-- the following linebreak is content: -->

```
fourth line<!-- this one's ignored because it's the last piece of  
content: -->
```

```
</pre>
```

содержит лишь строки

```
    first line  
    third line  
    fourth line
```

Пример 2

Пробелы и табуляторы

Символы пробелов должны воспроизводиться как горизонтальные пробелы. В языке HTML несколько рядом стоящих символов пробела должны воспроизводиться как один пробел.

Порядок воспроизведения символа горизонтальной табуляции (HT) не определен, и поэтому символ HT не должен использоваться в документах, за исключением тех случаев, когда он является частью элемента PRE.

Ни пробелы, ни табуляторы нельзя использовать для того, чтобы сделать документ в формате SGML более красивым и легко читаемым.

Список инструментария для разметки

Разделитель	Его значение
<!--	комментарий
&#	обращение к символу по числовому значению
&	обращение к объекту
</	метка завершения
<!	декларация элемента разметки
]]>	конец выделенной части документа (приводит к ошибке)
<	начальная метка

Общие свойства документа

HEAD

Элемент **HEAD** содержит всю информацию о документе в целом. Однако он не содержит какого-либо текста. Последний является лишь частью документа и должен находиться в элементе **BODY**.

В элементе заголовка **HEAD** можно использовать лишь строго заданный набор элементов.

Нижеприведенные элементы определяют общие свойства документа. Они должны появляться в элементе **HEAD**. Порядок элементов значения не имеет.

TITLE Название элемента.

ISINDEX Элемент, посылаемый серверу вместе с документом, предназначенным для информации к поиску.

LINK Элемент, определяющий связь этого документа с другими. В документе может присутствовать несколько элементов **LINK**. См. описание связей между документами и описание элемента Анкер.

BASE Запись, сделанная на языке URL при фиксации данного документа. .

Форматирование текста

В элементе **BODY** документа встречаются элементы из приведенного ниже списка. Они выстроены в том порядке, в каком должны подаваться на устройство вывода.

Заголовки (Headings)

Язык поддерживает заголовки разделов различных уровней.

Анкеры (Anchors)

Части текста, которые формируют начало и/или конец связей в гипертексте, называются, анкерами и формируются меткой **A**.

Метки параграфов (Paragraph marks)

Элемент **P** указывает на границу между параграфами.

Стиль адреса (Address style)

Этот элемент указывает, в каком стиле предстает перед клиентом элемент **ADDRESS**.

Выделенный блок текста. (Blockquote style).

Списки, словари и т.д.

Преформатированный текст (Preformatted text)

Части текста, предварительно отформатированные с использованием шрифта фиксированной ширины.

Выделение символов (Character highlighting)

Элементы форматирования, не вызывающие разбиения на параграфы.

Графика

IMG Метка **IMG** может использоваться для включения в текст графических изображений.

Форматирование текста

BODY

В противоположность элементу HEAD элемент BODY содержит всю ту информацию, из которой собственно и состоит рассматриваемый документ.

Анкер (элемент A)

Анкер - это некий текст, который указывает на начало и/или конец связи в гипертексте. Текст между открывающей и закрывающей метками определяет начало связи или указываемое ею место (или и то, и другое вместе). Метка анкера может иметь следующие атрибуты:

Форматирование текста

HREF Необязательный.

Если атрибут HREF установлен, то анкер является точно выверенный текстом - началом соединения. Если читатель выбрал этот текст, то ему (ей) будет представлен другой элемент, чей сетевой адрес определяется значением HREF атрибута. Формат сетевого адреса определяется в другом месте. Такой подход позволяет с помощью формы HREF="#индикатор" сослаться на другой анкер в том же самом документе. Если же анкер относится к другому документу, атрибут является относительным именем, именем относительно данного документа (либо он указывает базовый адрес, если таковой имеется).

NAME Необязательный.

Если этот атрибут указан, то он позволяет данному анкеру быть местом в документе, на которое ссылается какой-либо анкер. Значение атрибута является идентификатором анкера. Идентификатор анкера - это произвольная строка текста, которая тем не менее уникальна в пределах рассматриваемого HTML документа. Другие документы тоже могут создавать ссылки именно на этот анкер, помещая его идентификатор в поле адреса документа после символа #.

Форматирование текста

URN Необязательный.

Если этот атрибут указан, то это определяет универсальный номер ресурса для данного документа.

TITLE Необязательный.

Данный атрибут является чисто информационным. Если атрибут присутствует в анкере, его значение должно совпадать со значением элемента **TITLE** в документе, чей адрес указан в атрибуте **HREF**.

METHODS Необязательный.

Значение этого атрибута - строка. Она должна представлять собой список через запятую методов HTTP, которые программа общего пользования в состоянии поддерживать. См. замечание 3.

Форматирование текста

Пример использования анкеров

See `CERN`'s information for more details.

A `serious` crime is one which is associated with imprisonment.

The Organization may refuse employment to anyone convicted of a `serious` crime.

Пример 3

Форматирование текста

Заголовки (Headings)

Обрабатывается до шести уровней заголовков. Элемент заголовка несет в себе все изменения шрифтов, разбиение на параграфы до и после, пробелы, необходимые, например, для обработки заголовка. Язык HTML не требует применения иных средств для выделения символов или разбивки текста на параграфы.

Заголовок **H1** относится к самому верхнему уровню и рекомендуется в качестве начального для узла в гипертексте. Предполагается, что текст первого заголовка будет соответствовать запросам клиента, уже производящего анализ, связанный с этим узлом информации. Это отличает заголовок (heading) от названия (title), которое должно характеризовать данный узел в более широком плане.

Элементы заголовка: <H1>, <H2>, <H3>, <H4>, <H5>, <H6>.

Форматирование текста

Заголовки (Headings)

Было бы отклонением от правил при переходе от заголовка к заголовку пропускать какой-либо уровень, например, ставить после элемента H1 сразу элемент H3. Хотя такая практика и не запрещена, но нежелательна и может привести к странным результатам при написании других реализаций языка HTML.

Пример

```
<H1>This is a heading</H1>  
<P>Here is some text  
<H2>Second level heading</H2>  
<P>Here is some more text
```

Пример 4

Форматирование текста

Заголовки (Headings)

Типовые методы обработки

H1 Толстый, очень крупный шрифт, текст центрирован. Между заголовком и последующим текстом вставляется одна или две пустых строки. При выводе на принтер заголовки печатаются на новой странице.

H2 Толстый крупный шрифт. Без отступа. До и после заголовка помещаются одна или две пустых строки.

H3 Наклонный большой шрифт. До и после заголовка помещаются одна или две пустые строки. С небольшим отступом.

H4 Толстый нормальный шрифт. Отступ больше, чем в H3. До и после заголовка помещается пустая строка.

H5 Наклонный нормальный шрифт. Отступ как у заголовка H4. Пустая строка ставится перед заголовком, но не после.

H6 Толстый шрифт. Отступ такой же, как у обычного текста и больше, чем у H5. Перед заголовком ставится пустая строка.

Форматирование текста

IMG: Встроенные изображения

Элемент IMG позволяет вставлять информацию из другого документа. Последний обычно является иконкой, маленькой картинкой и т.д. Элемент IMG не предназначен для вставки дополнительного HTML текста.

Элемент IMG является пустым (не имеет заключительной метки) и имеет два атрибута:

SRC Значением этого атрибута является URL документа, который должен быть вставлен в гипертекст. Синтаксис этого атрибута такой же, как и у атрибута HREF для метки A. Атрибут SRC является обязательным.

ALIGN Это атрибут, принимая значения TOP, MIDDLE или BOTTOM, определяет, верхняя, средняя или нижняя часть изображения должна быть поставлена вровень с текстом. В тексте анкеров допускается применение элементов IMG.

Пример

```
Warning: < IMG SRC ="triangle.png">  
Thus must be done by a qualified echnician.  
< A HREF="Go">< IMG SRC =" rtriangle.png "> Press to start</A>
```

Пример 5

Форматирование текста

Списки

Список - это последовательность параграфов, каждому из которых может предшествовать специальная метка или очередной номер. Синтаксис списка:

```
<UL>  
<LI> list element  
<LI> another list element ...  
</UL>
```

Открывающими метками для списка могут быть UL, OL, MENU или DIR. Сразу за открывающей меткой должен следовать первый элемент списка.

Типовая обработка

Здесь мы не определяем порядок воспроизведения списков, однако должны быть четко оговорены размеченный список для случая неупорядоченных списков и очередь нумерованных параграфов для упорядоченного списка. Среди других возможностей для интерактивного показа списков - встроенные панели скроллинга для ускоренного просмотра.

Форматирование текста

Списки

Список элементов, имеющих типовые алгоритмы обработки:

UL

Список многострочных параграфов, обычно разделенных несколькими пробелами и/или размеченный кружками или крупными черными точками.

OL Этот элемент похож на элемент UL, за исключением того, что параграфы нумеруются.

MENU

Список параграфов меньшего размера. Обычно на одну запись приходится лишь одна строка, а ее стиль более компактен, чем в случае элемента UL.

DIR Список элементов, чей размер, как правило, не превышает 20 символов. Элементы могут размещаться в несколько колонок на странице, причем ширина такой колонки обычно 24 символа. Намного лучше, если программа обработки в состоянии оптимизировать ширину колонки в зависимости от ширины составляющих ее элементов.

Форматирование текста

P: Метка параграфа

Пустой элемент P служит разделителем параграфов. Конкретная процедура обработки (отступы, инструкции и т.д.) здесь не оговаривается и может зависеть от наличия иных меток, стилей и т.д.

Метка <P> ставится между двумя частями текста для их разделения.

Нет нужды применять <P> для создания пустого места вокруг заголовка, списка, адреса или выделенных элементов, которые уже сами по себе предполагают наличие разделителей параграфов. Создание пустых мест вокруг перечисленных элементов - обязанность программы обработки. Соседство метки разделителя параграфов и такого элемента, который сам автоматически создает разделители параграфов, может привести к непредсказуемым последствиям. Следует избегать того, чтобы метке разделителя параграфов предшествовал или следовал за ней такой элемент.

Форматирование текста

P: Метка параграфа

Типовая обработка

Обычно метка `<P>` создает небольшой вертикальный пропуск между параграфами (одна строка или полстроки). Этого не происходит (как правило) в тексте элементов ADDRESS и (даже) PRE. В некоторых версиях в обычном тексте метка `<P>` может также создавать небольшой отступ слева в первой строке открываемого ею параграфа.

Примеры использования

```
<h1>What to do</h1>
```

```
This is a one paragraph.<p>This is a second.
```

```
< P >
```

```
This is a third.
```

Примеры ошибок

```
<h1><P>What not to do</h1>
```

```
<p>I found that on my XYZ browser it looked prettier to me if I put some paragraph marks
```

```
<p>
```

```
<ul><p><li>Around lists, and
```

```
<li>After headings.
```

```
</ul>
```

```
<p> None of the paragraph marks in this example should be there.
```

Пример 8

Форматирование текста

PRE: предварительно отформатированный текст

В языке HTML предварительно отформатированный текст отображается в виде текста, использующего шрифт фиксированной ширины.

Необязательный атрибут этого элемента

WIDTH

Этот атрибут задает максимальное количество символов, которые может содержать строка текста. Эта информация позволяет системе обработки выбирать соответствующие данному случаю шрифты и отступ. В случаях, когда атрибут WIDTH не идентифицирован, рекомендуется выбирать ширину 80 символов. В случаях, когда атрибут WIDTH присутствует, рекомендуется иметь фиксированный набор размеров 40, 80 и 132 символа, остальные значения атрибута должны быть промежуточными.

Форматирование текста

PRE: предварительно отформатированный текст

В пределах элемента PRE:

В тексте символы конца строки обрабатываются как информация к переходу на начало следующей строки. Это не относится к строке непосредственно предшествующей метке, или следующей за ней.

Не должна применяться метка <r>. Если она все же обнаружена, то должна интерпретироваться как команда перехода на начало следующей строки.

Могут применяться анкеры и элементы выделения символов.

Не должны применяться элементы, определяющие форматирование параграфов (заголовки, адреса и т.д.).

ASCII символ горизонтальной табуляции (HT) должен интерпретироваться как наименьшее положительное количество пробелов, остающихся до позиции, чей номер кратен 8. Однако применение этого символа не рекомендуется.

Пример использования

```
<PRE WIDTH="80">
```

```
This is an example line
```

```
</PRE> Пример 9
```

Форматирование текста

Название документа

Название документа задается элементом TITLE. В документе элемент TITLE должен быть частью элемента HEAD. Любой документ может иметь не более одного названия. Название должно характеризовать содержание элемента в более широком плане. Название не является всего лишь частью текста в документе. Оно относится ко всему документу в целом. Название не может содержать анкеры, разделители параграфов, выделение символов. Название документа может использоваться при идентификации узла в архивном списке, в качестве метки на окне, показывающем этот узел гипертекста. Название, как правило, не появляется в текстовом поле документа. Названия противопоставляются заголовкам. В идеале, размер названия документа не должен превышать 64 символов.

Примеры приемлемых названий

`<TITLE>Rivest and Neuman. 1989(b)</TITLE>`

Примеры неприемлемых заголовков

Заголовок, имеющий смысл лишь в случае привязки к конкретному контексту

`<TITLE>Introduction</TITLE>`

или слишком длинный заголовок

`<TITLE>Remarks on the Quantum-Gravity effects of "Bean Pone" diversification in Mononucleosis patients in Developing Countries under Economic Conditions Prevalent during the Second half of the Twentieth Century, and Related Papers: a Summary</TITLE>`

[Пример 10](#)

Форматирование текста

Выделение символов

Элементы выделения позволяют форматировать отдельные части текста особым образом, производить выделение и т.д. Метки выделения не приводят к разбиению на параграфы и могут применяться к отдельным кускам текста внутри параграфов. Как и все метки, не поддерживаемые имеющимися версиями анализаторов языка HTML, эти метки будут игнорироваться, однако размеченный ими текст будет обрабатываться как и любой другой.

Все метки выделения имеют соответствующие им завершающие метки, как в этом примере

This is emphasized text.

Форматирование текста

Выделение символов

Физические стили

T Шрифт фиксированной ширины.

B Толстый или еще каким-либо образом выделенный шрифт.

I Наклонный шрифт (или искаженный каким-либо образом, если просто наклон невозможен).

U Подчеркивание.

Форматирование текста

Выделение символов

Логические стили

EM Выделение символов (обычно наклон шрифта).

STRONG Более четкое выделение (обычно применение более жирного шрифта).

CODE Пример кода. Обычно фиксированный шрифт (не путать с элементом PRE).

SAMP Последовательность символов. (*названия команд, примеры.*)

KBD Текст, набираемый пользователем. Этот стиль применяется в описаниях.

VAR Имя переменной.

DFN Пример определения к какому-либо термину. Обычно жирный наклонный шрифт или просто жирный.

CITE Цитата. Обычно наклонный шрифт.

Пример 11 Форматирование

Форматирование текста

Пример использования

This text contains an `emphasized` word.

`Don't assume` that it will be italic!

It was made using the `<CODE>EM</CODE>` element.

A citation is typically italic and was no formal necessary structure:

`<cite>Moby Dick</cite>` is a book title.

Пример 12 Выделение символов

Форматирование текста

Объекты

В языке HTML применяются нижеуказанные имена объектов. Этим именам всегда предшествует амперсанд (&), а за ними следует точка с запятой. Объекты представляют конкретные графические символы, имеющие определенное значение при разметке, или же они не относятся к символам, которые не могут быть напечатаны иным образом.

< символ "меньше чем" <

> символ "больше чем" >

& символ амперсанда &.

" символ двойных кавычек "

[Пример 13](#) Примеры отображения символов

[Пример 1](#)[Пример 14](#) Таблица цветов

[Пример](#)[Пример 15](#) Пример составление таблиц