

# Язык HTML. История развития, возможности.



# История развития HTML

Годом рождения данного языка принято считать 1986, когда Международная организация по стандартизации (ISO) был принят стандарт, названный SGML (Standard Generalized Markup Language). Данный язык описывали как язык для логического оформления текста, тогда не подразумевали описание внешнего вида документа. SGML даже не представлял собой готовую систему для разметки текста, он не предполагал какого-либо списка структурных элементов, которые применялись бы в определённых ситуациях. Данный язык подразумевал лишь описание синтаксиса написания основных элементов разметки текста (которые сейчас называют тегами). Это основные причины того, что язык SGML не получил значимого распространения.

Интерес к данному языку проявили в 1991 году, когда Европейский институт физики частиц ощутил потребность в передаче гипертекстовой информации по сети Интернет. В том году на основе SGML был создан новый язык - HTML (Hyper Text Markup Language). В HTML версии 1.2 присутствовало около 40 тегов, но, как и в случае с SGML, подразумевалось только логическая разметка текста. Только несколько тегов подразумевали физическую разметку веб-страниц (эти теги не рекомендовались к использованию).

# История развития HTML

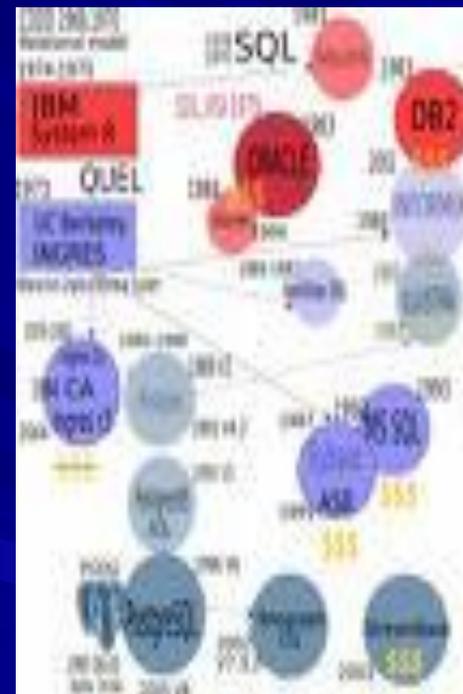
В 1994 году консорциум W3C начал работу над HTML 2.0, окончательный стандарт которой был принят в 1995 году, когда уже вовсю шла разработка HTML 3.0. В HTML 2.0 появились формы - средства отправки информации от пользователя на сервер. Самым большим прорывом явился HTML 3.0. Он включил в себя такие нововведения, как теги для создания таблиц, вставки рисунков, примечаний, разметку математических формул и др. Наиболее важным нововведением явились CSS (таблицы стилей), которые позволяли задавать единый вид всему сайту. Они имели свою структуру, свой синтаксис и применялись вместе с HTML.

В 1995 году появился браузер Netscape Navigator, разработчиком которого была корпорация Netscape Communications. Эта корпорация с целью привлечь пользователей стала вводить новые теги, которые были предназначены на улучшение внешнего вида документов. Тем самым были разрушены изначальные принципы языка.

# История развития HTML

В 1996 году в свет вышел браузер Internet Explorer 2.0 (разработчик: корпорация Microsoft), но он не имел большой популярности ввиду того, что не поддерживал многих тегов, введённых Netscape. Их поддержка была введена в Internet Explorer 3.0, после выхода которого рынок браузеров был поделён пополам между Microsoft и Netscape.

В скором времени был создан HTML 3.2, который был ориентирован на Internet Explorer. Данный стандарт долгое время оставался единственным для веб-разработчиков. В 2004 году в свет вышел HTML 4.01, обеспечивающий достаточно большую кросс-браузерность. Данный стандарт и CSS сейчас применяются при создании большинства сайтов.



# История развития HTML

Однако HTML сегодня уже совсем не тот, что вначале. Становится все более очевидным, что язык этот в его теперешнем состоянии практически исчерпал перспективы развития, и что добавление новых тегов вряд ли выведет его на принципиально иной уровень. Это особенно понятно тем, кто занимается практическим дизайном: из-за того, что HTML с самого начала не ориентировался на описание внешнего вида документа, он не в состоянии удовлетворительно выполнить эту задачу даже сейчас, при наличии множества визуально-ориентированных тегов. Прямым следствием этого является огромное количество расхождений в интерпретации тегов браузерами. Как бы строго вы ни следовали стандарту, HTML-файл придется обязательно тестировать по меньшей мере в двух браузерах - NetscapeNavigator и InternetExplorer, и чаще всего такое тестирование не обходится без неприятных сюрпризов: отступы, пробелы, размеры элементов оформления и логика их размещения на странице даже для простейших тегов различаются довольно сильно.

# Возможности HTML

Язык HTML позволяет размечать электронный документ, который отображается на экране с полиграфическим уровнем оформления; результирующий документ может содержать самые разнообразные метки, иллюстрации, аудио- и видеофрагменты и так далее. В состав языка вошли развитые средства для создания различных уровней заголовков, шрифтовых выделений, различные списки, таблицы и многое другое.

Вторым важным моментом, повлиявшим на судьбу HTML, стало то, что в качестве основы был выбран обычный текстовый файл. Выбор был сделан под влиянием следующих факторов: такой файл можно создать в любом текстовом редакторе на любой аппаратной платформе в среде какой угодно операционной системы; к моменту разработки HTML существовал американский стандарт для разработки сетевых информационных систем — Z39.50, в котором в качестве единицы хранения указывался простой текстовый файл в кодировке LATIN1, что соответствует US ASCII.

# Возможности HTML

Таким образом, гипертекстовая база данных в концепции **WWW** — это набор текстовых файлов, размеченных на языке HTML, который определяет форму представления информации (разметка) и структуру связей между этими файлами и другими информационными ресурсами (гипертекстовые ссылки). Гипертекстовые ссылки, устанавливающие связи между текстовыми документами, постепенно стали объединять самые различные информационные ресурсы, в том числе звук и видео; в результате возникло новое понятие — гипермедиа.

Такой подход предполагает наличие еще одного компонента технологии — интерпретатора языка. В World Wide Web функции интерпретатора разделены между Web-сервером гипертекстовой базы данных и интерфейсом пользователя. Сервер, кроме доступа к документам и обработки гипертекстовых ссылок, обеспечивает предпроцессорную обработку документов, в то время как интерфейс пользователя осуществляет интерпретацию конструкций языка, связанных с представлением информации.

# Возможности HTML

Первая версия языка (HTML 1.0) была направлена на представление языка как такового, где описание его возможностей носило скорее рекомендательный характер. Вторая версия языка (HTML 2.0) фиксировала практику использования его конструкций. Версия ++ (HTML++) представляла новые возможности, расширяя набор тегов HTML в сторону отображения научной информации и таблиц, а также улучшения стиля компоновки изображений и текста. Версия 3.2 смогла упорядочить все нововведения и согласовать их с существующей практикой. HTML 3.2 позволяет реализовать использование таблиц, выполнение кодов языка Java, обтекание графики текстом, а также отображение верхних и нижних индексов.

Сейчас World Wide Web Consortium (W3C) — международная организация, которая занимается подготовкой и распространением документации на описание новых версий HTML — уже опубликовала материалы спецификации HTML 4.01. Кроме возможностей разметки текста, включения мультимедиа и формирования гипертекстовых связей, уже существовавших в предыдущих версиях HTML, в версию 4.01 включены дополнительные средства работы с мультимедиа, языки программирования, таблицы стилей, упрощенные средства печати изображений и документов.

# Возможности HTML

Для управления сценариями просмотра страниц Website (гипертекстовой базы данных, выполненной в технологии World Wide Web) можно использовать языки программирования этих сценариев, например, JavaScript, Java и VBScript.

Усложнение HTML и появление языков программирования привело к тому, что разработка Web-узлов стала делом высокопрофессиональным, требующим специализации по направлениям деятельности и постоянного изучения новых Web-технологий. Но возможности Internet позволяют пользователям, владеющим основами HTML, создавать и размещать собственные Web-узлы без больших затрат. Именно на таких пользователей и рассчитан предлагаемый курс.



# Контейнеры и теги HTML

Язык HTML позволяет определять структуру электронного документа с полиграфическим уровнем оформления; результирующий документ может содержать самые разнообразные элементы: иллюстрации, аудио- и видеофрагменты и так далее. Язык включает в свой состав развитые средства для специфицирования нескольких уровней заголовков, шрифтовых выделений, различных групп объектов, например, словари, каталоги или меню для размещения иллюстраций и других фрагментов, а также множество других возможностей.



# Контейнеры и теги HTML

Теговая модель описания документа позволяет представить его в виде совокупности элементов, каждый из которых окружен тегом (управляющим HTML-кодом).

Документ HTML представляет собой не что иное, как обычный ASCII-файл, с добавленными в него управляющими HTML-кодами (тегами). Каждый тег начинается с левой угловой скобки "<" и заканчивается правой угловой скобкой ">". Между скобками записывается идентификатор тега и, если необходимо, дополнительные аргументы. Некоторые теги используются вместе с закрывающей парой, которая сигнализирует об окончании области действия соответствующего назначения. Закрывающая пара после левой угловой скобки содержит символ "/" (например, для тега наклонного шрифта <I> закрывающая пара представляет собой </I>, для тега заголовка <TITLE> закрывающей парой будет </TITLE>).

Иногда тег с закрывающей парой называют контейнером в том смысле, что внутри него можно поместить содержательную информацию определенного характера. Структура HTML-документа позволяет использовать вложенные друг в друга контейнеры.

Такой подход предполагает наличие еще одной компоненты технологии – интерпретатора языка. В WWW функции интерпретатора в основном выполняются клиентом-браузером.

# Группы тегов HTML

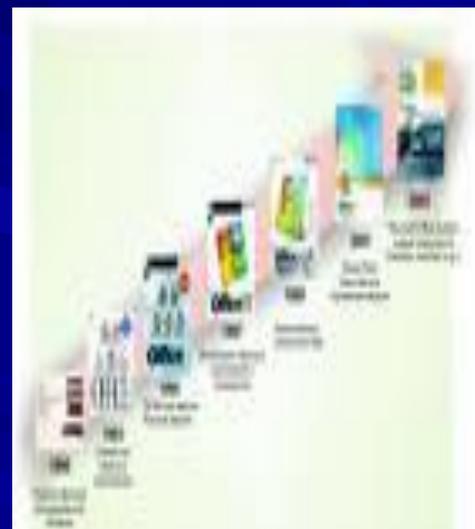
Все теги HTML по их назначению и области действия можно разделить на следующие основные группы:

- структура документа;
- оформление блоков гипертекста (параграфы, списки, таблицы, картинки);
- гипертекстовые ссылки и закладки;
- формы для организации диалога;
- вызов программ.

# Структура HTML-документа

HTML-документ представляет собой иерархию элементов документа, заключенных в "теговые скобки". В секцию головы всегда помещается заглавие документа (не путать с именем файла), которое при просмотре HTML-страницы появляется в заголовке окна браузера. Заглавие показывает общий смысл HTML-страницы. Для определения заглавия HTML-страницы применяется теговый контейнер `<TITLE>`, например:

- `<TITLE>Моя любимая страница</TITLE>`
- `<HTML>`
- `<HEAD>`
- *Голова документа*
- `</HEAD>`
- `<BODY>`
- *Тело, Содержание документа*
- `</BODY> </HTML>`



# Структура HTML-документа

Кроме классической структуры простого гипертекстового документа, современные браузеры поддерживают *кадры* - не перекрывающиеся области окна браузера, в которых возможно одновременно отображать несколько документов.

Для формирования кадров используется контейнер набора кадров FRAMESET и тега FRAME. Альтернативное содержимое задается с помощью контейнера NOFRAMES:

- <HTML>
- <HEAD>
- Голова документа
- </HEAD>
- <Frameset ...>
- <FRAME ...> Связывание кадров
- <FRAME ...> с документами
- </FRAMESET>
- <NOFRAMES>
- Часть документа только для старых браузеров
- </NOFRAMES>
- </HTML>



# Структура HTML-документа

- Начальный тег FRAMESET определяет разбиение окна по строкам и столбцам, например:
- `<FRAMESET ROWS="60%,*" COLS="60%,20%,*">`.
- Тег кадра FRAME устанавливает какой документ (атрибут SRC) первоначально выводится в кадре. Атрибут Name определяет имя целевого кадра для указания его в гиперссылках.
- Одна из распространенных компоновок кадров содержит левый узкий кадр для оглавления в виде гиперссылок и правый кадр для показа документов по этим ссылкам. Другая компоновка предлагает разбиение на верхний и нижний кадры.
- Посредством вложения контейнеров FRAMESET можно реализовать более сложные компоновки кадров, например, добавить верхний и нижний колонтитулы.



# Оформление блоков гипертекста

Для придания большей выразительности HTML-страницам применяются следующие теги и их закрывающие пары:

- для использования жирного (**bold**) написания `<B>` и `</B>`
- для использования наклонного (*italic*) написания `<I>` и `</I>`
- для использования режима (underline) подчеркивания `<U>` и `</U>`

В текст HTML-страницы могут быть добавлены заголовки и подзаголовки (с уровнем вложения до шести). Для этого применяются теги `<H1>`, `<H2>`, `<H3>`, `<H4>`, `<H5>`, `<H6>` с соответствующими закрывающими тегами.

Для форматирования строк и абзацев используют следующие теги:

`<P> ... </P>` - оформление отдельного абзаца,  
`<BR>` - принудительный перевод строки,  
`<HR>` - горизонтальная черта (разделитель),  
`<PRE> ... </PRE>` - передача текста без дополнительного форматирования.

# Оформление блоков гипертекста

Это так называемые способы физического выделения, они не зависят от реализации браузера, а есть логические стили (<EM>, <BIG> и др.), форматирующие текст по-разному в различных браузерах.

Для оформления перечней в гипертексте определены три типа списков: неупорядоченные <UL>, упорядоченные <OL> и списки определений <DL>. Пример оформления неупорядоченного списка:

- <UL>
- <LI> Первый элемент
- <LI> Второй элемент
- <LI> Последний элемент
- </UL>
- Списки определений служат для создания глоссариев и индексов:
- <DL>
- <DT>Первый термин
- <DD>Определение первого термина
- <DT>Второй термин
- <DD>Определение второго термина
- </DL>

Совместно с механизмом ссылок перечни позволяют строить простые меню.

# Описание таблиц

Таблица представляет собой расположенную на HTML-странице прямоугольную сетку строк и столбцов, в которую можно вводить разнообразную информацию, включая текст, числа, связи и даже изображения. Общий формат описания таблицы следующий:

```
<TABLE> Контейнер таблицы
<CAPTION> Контейнер заголовка таблицы
    Заголовок таблицы
</CAPTION>
<TR> Контейнер строки таблицы
<TH>Заголовок 1-го столбца</TH> Контейнеры заголовков столбцов
<TH>Заголовок 2-го столбца</TH>
    ...
</TR>
<TR>
<TD>Первая ячейка</TD> Контейнеры ячеек
<TD>Вторая ячейка</TD>
    ...
</TR>
... </TABLE>
```

Все описание таблицы размещается в контейнере TABLE. Чтобы показать в таблице рамку применяют атрибут BORDER тега <TABLE>. Дополнительно задается заголовок таблицы - контейнер CAPTION:

# Описание таблиц

`<CAPTIONALIGN=выравнивание>Заголовок таблицы</CAPTION>`

где выравнивание принимает значения: TOP - вверху, BOTTOM - внизу.

Таблица определяется по строкам с помощью контейнера TR, в который вложены контейнеры ячеек TD, разделенные по столбцам.

При этом в контейнере ячейки можно помещать не только текст, но и метки для форматирования текста (например, `<B>` или `<I>`), списки и даже изображения, а также вложенные таблицы.

```
<TABLE>
  <TR>
    <TD>One
    <TD><IMGSRC="picture.jpg" ALT=" ">
  <TD><FONTSIZE=5>W<FONTSIZE=3>Three
  </TR>
</TABLE>
```

Обратите внимание, что в таблицах допускается смешивать текст и графику.

Закрывающие теги `</TR>` и `</TD>` можно опускать.

Стандартным выравниваем ячеек таблицы является выравнивание влево ячеек с данными и центрирование ячеек надписей.

# Описание таблиц

Кроме этого, внутри тегов <TD> и <TH> можно осуществлять выравнивание с помощью атрибута ALIGN со значениями: LEFT (влево), RIGHT (вправо) или CENTER (центрировать).

Кроме горизонтального выравнивания в клетках таблиц можно применять и вертикальное выравнивание текста. В этом случае в тегах <TD> и <TH> применяется атрибут VALIGN с выравниванием по верху (TOP), по низу (BOTTOM) и выравниванием по середине (MIDDLE).

Для перекрытия текстом нескольких строк или столбцов внутри тегов <TD> и <TH> используются специальные атрибуты: COLSPAN=m для перекрытия нескольких колонок (где m - число колонок), и ROWSPAN=n для перекрытия нескольких столбцов (где n - число столбцов).

Таблицы служат хорошим инструментом для упорядочения информации на Web-странице.



# Формы для организации диалога

Форма FORM служит для организации диалога по выбору и вводу данных, по нажатию экранных кнопок. Обработка введенных данных осуществляется серверной программой, заданной в атрибуте ACTION. Обмен данными между браузером и сервером производится через интерфейс CGI с помощью методов GET (через переменные среды) и POST (через входной и выходной потоки), задаваемые атрибутом METHOD. В обмене участвуют пары "имя=значение" для диалоговых переменных.

При наличии нескольких форм на HTML-странице необходимо их именовать (атрибут NAME).

Основными элементами формы являются элемент ввода INPUT, многострочная область текста TEXTAREA и меню SELECT



# Формы для организации диалога

<FORM NAME=*форма* ACTION=*сервер\_прогр*METHOD=*метод\_доступа*  
<INPUTTYPE=*тип* NAME=*перем...*> Ввод данных

...

<TEXTAREANAME=*имя*>

*многострочный текст*

</TEXTAREA>

<SELECTNAME=*имя*> Выбор из меню (списка) <OPTION>

Первое <OPTION> Второе <OPTION> Третье </SELECT> </FORM>

Элемент INPUT является наиболее универсальным, реализует наибольшее число диалоговых элементов, тип которых определяется атрибутом TYPE.

Поля ввода имеют типы:

TEXT - простое текстовое поле

PASSWORD - ввод пароля без отображения

FILE - поле ввода или выбора имени файла.

# Формы для организации диалога

Для полей выбора различают типы:

- CHECKBOX - поле выбора с помощью пометки
- RADIO - поле выбора с помощью "радиокнопок".

В формах поддерживаются экранные кнопки следующих типов:

- SUBMIT - кнопка вызова действия ACTION
- RESET - кнопка сброса (восстановления значений по умолчанию)
- IMAGE - пиктографическая кнопка.

Каждый элемент INPUT обязательно имеет имя переменной, задаваемое атрибутом NAME, оно служит для идентификации переменной при обмене данными с обрабатывающей программой.

Для полей ввода можно задавать значения по умолчанию с помощью атрибута VALUE. Этот же атрибут для текстовых кнопок определяет надписи на них. Начальное состояние полей выбора задается атрибутом CHECKED.

Атрибут SIZE определяет размер видимой части поля ввода, а MAXLENGTH - максимальное число символов, которое можно ввести в поле. С помощью атрибута ALIGN обеспечивается выравнивание введенного значения.

# Формы для организации диалога

Специальный тип `HIDDEN` позволяет скрыть элемент диалога, а также может служить для хранения значений рабочих переменных.

Для графических кнопок атрибут `SRC` задает адрес изображения.

Элемент `TEXTAREA` предназначен для ввода строк длинного текста. Каждая область поименована (атрибут `NAME`) и имеет два параметра для определения ее размера: число видимых строк `ROWS` и ширина области `COLS`. Текст внутри контейнера `TEXTAREA` отображается на экране.



Спасибо за внимание!