**Глава 1 Введение в HTML**

Итак, почему же HTML так широко распространен и используется для публикации информации в сети Интернет? Своей популярностью HTML во многом обязан удобству навигации между документами и простоте создания самих HTML‑документов. Так, нажимая кнопку мыши над участком текста или изображением, можно открывать документы, расположенные в различных частях света на компьютерах с различными операционными системами.

Сеть из связанных между собой гипертекстовых документов является прозрачной для пользователя. Поэтому при работе с ней вас совершенно не заботят операции, которые осуществляются сотнями устройств глобальной компьютерной сети только для того, чтобы доставить на ваш компьютер содержимое HTML‑документа.

**1.1. Краткая история HTML**

Началось все для HTML (а вместе с ним и для WWW) в конце 1980‑х годов, когда у ученых из Европейской лаборатории элементарных частиц (CERN) возникла необходимость обмениваться множеством различных документов при помощи быстро развивающейся сети Интернет. Тогда необходимо было придумать способ публикации документов в сети таким образом, чтобы к любому нужному документу можно было легко получить доступ. К тому же нужно было, чтобы документы отображались на всех компьютерах одинаковым образом.

Решение поставленной задачи было найдено сотрудником CERN Бернерс‑Ли. В 1989 году Тим Бернерс‑Ли создал новый язык форматирования документов. В его основу был положен другой ранее созданный язык – SGML, который предусматривал установку связей между документами с помощью гиперссылок. Новый язык разметки был назван HTML (HyperText Markup Language). Этот же человек реализовал и первую программу для просмотра HTML‑документов – браузер.

Затея с гипертекстом очень быстро прижилась. Вскоре в Интернете выросла большая сеть гипертекстовых документов, которую постепенно стали называть World Wide Web. К тому же в 1993 году команда программистов, руководителем которой являлся основатель компании Netscape Марк Андриессен, разработала первый браузер, имеющий полноценный графический интерфейс и позволяющий работать с мышью, – Mosaic. Это не могло не добавить популярности быстро развивающейся Сети.

Со временем Интернет стал востребован не только учеными: к нему пришли и рядовые пользователи, причем их число неуклонно возрастало. Для многих было очевидно, что HTML, не предполагающий никакой динамики в создаваемых с его помощью документах, достаточно скучен и невзрачен. Это дало толчок развитию технологий CSS, введению поддержки апплетов Java, а после и сценариев (первым языком был JavaScript).

Нельзя не отметить, что с образованием и ростом Сети создателей HTML не на шутку взволновала «чистота» своего детища. Существовали небезосновательные опасения, что производители браузеров, которые скоро должны были прийти на рынок, будут «баловать» своих пользователей фирменными нововведениями, которые, естественно, будут поддерживаться только фирменными браузерами. В 1994 году была создана организация, взявшая на себя разработку единых стандартов развития WWW – World Wide Web Consortium (W3C). Эта организация и занялась подготовкой стандартов HTML (начиная с HTML 2.0). Правда, несмотря на наличие W3C, нововведения в HTML начинали поддерживаться производителями браузеров гораздо раньше, чем эта организация их стандартизировала (так, например, обстояло дело с фреймами, с внедрением сценариев в HTML‑документы, с объектной моделью документов и т. д.).

Организация W3C существует и сейчас. Она занимается теми же вопросами стандартизации, однако уже давно не HTML (последняя спецификация HTML 4.01 была принята 24 декабря 1999 года). Развитие языка HTML сегодня практически завершено, а основные усилия W3C сконцентрированы на работе над более новыми технологиями, например расширяемым языком разметки XML, языком преобразований XSLT и др. (если интересно, можно заглянуть на [www.w3.org](http://www.w3.org/) – официальный сайт этой организации).

Язык HTML в том виде, в котором он существует сейчас, обладает большим потенциалом представления информации. Причем он рассчитан не только на пользователей персональных компьютеров. Документы HTML можно просматривать на очень большом количестве различных по своим возможностям устройств: от черно‑белого экрана мобильного телефона до телетайпа или терминала. Кроме того, в последние версии HTML включен ряд возможностей, облегчающих представление документов неграфическими средствами (например, речевыми синтезаторами), позволяющими пользоваться услугами WWW людям с физическими недостатками.

**1.2. Базовые понятия HTML**

После маленького исторического экскурса можно наконец‑то перейти к практической части. Перед подробным рассмотрением HTML нужно ознакомиться с основными понятиями, используемыми в книге, а также с базовыми элементами языка HTML.

Для начала следует разобраться с тем, что такое HTML‑документ. HTML‑документ – это обычный текстовый документ, созданный в любом текстовом редакторе, например в Блокноте, и оформленный в соответствии с правилами языка HTML. Для файлов, содержащих HTML‑документы, используется расширение HTML или HTM (например, MyWedPage.html или MyWedPage.htm). Расширение HTM использовалось ранее для корректного отображения имен файлов в формате MS‑DOS. На самом деле неважно, какое из приведенных расширений использовать.

Далее в книге HTML‑документы могут называться просто документами, а также страницами и веб‑страницами. Под сайтами в тексте книги подразумевается несколько документов, объединенных единой системой навигации.

**Элементы**

Элемент – это конструкция языка HTML, содержащая данные. HTML включает в себя различные типы элементов, которые позволяют задавать абзацы, гипертекстовые ссылки, списки, таблицы, изображения и т. д. Конструкция <P>Привет !</P> представляет собой элемент. Обычно элемент можно разделить на три части. Первая часть – <P> – называется открывающим тегом (англ. tag). Далее идет содержание элемента, которое в данном случае состоит из слова Привет !. И наконец, </P> является закрывающим тегом. Как видно, название элемента (P) присутствует и в открывающем, и в закрывающем теге. Регистр символов в названии элемента не имеет значения. Однако в соответствии с соглашениями, принятыми большинством разработчиков, в примерах данной книги названия элементов записаны в верхнем регистре.

Открывающий и закрывающий теги нужны для указания начала и конца элемента. Теги всегда начинаются символом < и заканчиваются символом >. В закрывающем теге перед его именем помещается символ /. Для некоторых типов элементов допускается отсутствие закрывающего тега (например, элемент P, указывающий начало абзаца). Существуют также элементы, не имеющие закрывающего тега, то есть его не просто можно опустить, а он вообще не существует в языке.

**Атрибуты**

Элементы могут содержать параметры, называемые атрибутами. Атрибуты могут иметь определенные значения (по умолчанию или устанавливаемые авторами). Пара атрибут/значение указывается в начальном теге элемента перед символом >, например:

*<BODY bgcolor = «#FF0000»>*

Каждому атрибуту может быть присвоено значение определенного типа. В приведенном примере указание атрибута bgcolor (имеющего тип %Color) в элементе BODY приведет к тому, что цвет фона страницы станет красным. Значения атрибутов заключаются в кавычки, хотя в определенных случаях кавычки необязательны.

В начальном теге элемента может быть указано любое количество допустимых пар атрибут/значение, разделенных пробелами, например:

*<BODY bgcolor = «#FF0000» text = «#0000FF»>*

В приведенном примере устанавливаются красный цвет фона страницы и синий цвет основного текста. При установке значений нескольких атрибутов порядок их записи не имеет значения. Важно отметить, что регистр, в котором записываются названия атрибутов, также не имеет значения. Однако для повышения читабельности HTML‑кода названия атрибутов обычно записываются в нижнем регистре (как в приведенных выше и ниже примерах).

Существует любопытная разновидность атрибутов – булевы атрибуты. Для них возможны только два значения: ИСТИНА и ЛОЖЬ. По умолчанию эти атрибуты имеют значение ЛОЖЬ. Чтобы присвоить им истинное значение, достаточно просто указать имя этого атрибута, не присваивая ему никакого значения. Ниже приведен пример, в котором устанавливаются истинные значения двух атрибутов HTML‑элемента INPUT:

*<INPUT readonly disabled>*

**Вложенные элементы**

Важным моментом HTML является возможность использования вложенных элементов. Элемент, находящийся внутри другого элемента, называется вложенным. Пример использования вложенных элементов для задания начертания шрифта:

*<I>Курсив<B>-Полужирный курсив-</B>Курсив</I>*

При обработке приведенного HTML‑кода получится страница, показанная на рис. 1.1 (как и чем обрабатываются HTML‑документы, будет рассказано далее).



Рис. 1.1. Применение вложенных элементов

На приведенном рисунке видно, как действие внешнего (или родительского) элемента I (задание курсива) дополняет действие внутреннего элемента B (задание полужирного начертания шрифта).

При использовании вложенности следует помнить, что вложенные элементы должны закрываться до того, как будут закрыты внешние элементы. Так, следующий пример является неверным:

*<I>Неправильное закрытие<B> внутреннего элемента</I>*

*до закрытия внешнего</B>*

**Блочные и встроенные элементы**

Различают также блочные и встроенные элементы (иногда их называют элементами уровня блока и элементами уровня текста). Основным отличием блочных элементов является форматирование их браузером как обособленной части документа. Блочные элементы задаются парными тегами и могут содержать вложенные блочные или встроенные элементы и, естественно, текст.

Встроенные элементы обычно находятся прямо в тексте и могут иметь содержимое или не иметь его. Примерами встроенных элементов могут служить приведенные ранее элементы B и I, а также элементы перевода строки, изображения и т. д. В отличие от блочных элементов, встроенные элементы могут содержать только текст или вложенные встроенные элементы.

**1.3. Просмотр HTML-документа**

Сам по себе HTML‑документ практически нечитабелен для обычного пользователя. Для чего же тогда применяется форматирование документа с использованием HTML? Для просмотра HTML‑документов используются специальные программы – браузеры. При открытии HTML‑документа браузер распознает теги и учитывает их при отображении текста. Если по каким‑то причинам (например, при ошибке в записи тега) тег не распознается браузером, то он игнорируется.

Существует большое количество программ, позволяющих просматривать HTML‑документы. Это такие распространенные приложения, как Internet Explorer, Opera, Firefox, Netscape Navigator.

Ниже приведен пример простого HTML‑документа (назначение используемых элементов будет рассмотрено далее в книге) (пример 1.1).

***Пример 1.1. HTML-документ***

*<HTML>*

*<TITLE>Простой HTML-документ</TITLE>*

*<BODY>*

*<H1>Заголовок</H1>*

*Текст страницы*

*</BODY>*

*</HTML>*

Этот HTML‑документ отображается браузером Internet Explorer так, как показано на рис. 1.2.



Рис. 1.2. Отображение HTML-документа

**1.4. Универсальный идентификатор ресурса URI**

Чтобы полностью понимать, как происходит взаимодействие HTML‑документов, переход между страницами и откуда вообще компьютер пользователя получает данные при работе с сетью, нужно рассмотреть, как и к чему осуществляется доступ при помощи Глобальной сети.

Многие виды ресурсов, размещенных в Интернете, независимо от того, являются ли они HTML‑документами, рисунками или файлами архива, чаще всего представляют собой файлы на жестком диске компьютера (сервера), подключенного к сети. С каждым ресурсом сопоставляется значение, по которому можно однозначно определить его расположение, – универсальный идентификатор ресурса или URI (Universal Resource Identifier). URI широко используются как при самостоятельном доступе пользователя к ресурсу (когда, например, пользователь сам вводит URI в адресной строке браузера), так и при переходе между веб‑страницами. URI также используются в HTML‑документе для указания браузеру, где искать ресурсы (например, рисунки), используемые в самом документе.

***Примечание***

*В литературе также часто применяется обозначение URL. Следует отметить, что URI является более общим понятием, включающим в себя URL: любой URL является универсальным идентификатором ресурса и подчиняется тем же правилам, что и URI.*

Идентификатор ресурса URI состоит из трех частей: из наименования механизма доступа к ресурсу, доменного имени компьютера и пути файла ресурса. Для пояснения сказанного можно рассмотреть пример:

[*http://www.somesite.com/info/examples/ex\_1.html*](http://www.somesite.com/info/examples/ex_1.html)

Здесь можно увидеть, что для доступа к ресурсу, которым в данном случае является HTML‑документ, используется протокол HTTP (Hyper Text Transfer Protocol). Ресурс хранится на компьютере, имеющем доменное имя somesite.com в файле ex\_1.html, расположенном в папке /info/examples.

При помощи URI можно также ссылаться на части HTML‑документов, например:

[*http://www.somesite.com/info/examples/ex\_1.html#description*](http://www.somesite.com/info/examples/ex_1.html#description)

При использовании этого URI можно получить доступ к части HTML‑документа, имеющей имя description (то, как создавать имена для фрагментов HTML‑документов, будет рассмотрено в гл. 5).

URI также позволяют ссылаться на ресурсы в пределах одного компьютера. При этом указывается относительный путь ресурса. Например, чтобы из HTML‑документа, расположенного в папке /info/examples, сослаться на файл /info/files/file1.jpg, достаточно задать URI /files/file1.jpg. В HTML‑документах при помощи подобных ссылок указываются пути рисунков и других объектов, используемых в документах, но непосредственно не хранимых в них.

В общем случае URI считаются нечувствительными к регистру символов. Однако для полной уверенности в правильности интерпретации URI все же обращайте внимание на регистр символов в URI гиперссылок, рисунков и т. д. Это полезно для устранения таких ситуаций, когда, например, при работе сайта на компьютере под Windows все гиперссылки работают, а при помещении сайта на UNIX‑сервер работать отказываются (в UNIX имена файлов чувствительны к регистру).