5. Представление документа в формате HTML

В этой главе обсуждается, как документы в формате HTML представляются на компьютере и в Интернет.

Раздел набор символов документа относится к вопросу об абстрактных *символах*, которые могут входить в состав документа в формате HTML. В число этих символов входят латинская буква "A", кириллическая буква "I", китайский иероглиф "вода" и т.д.

Раздел кодировки символов относится к вопросу о том, как эти символы могут быть *представлены* в файле или во время передачи по Интернет. Поскольку некоторые кодировки могут прямо не представлять все символы, которые автор захочет включить в документ, HTML предлагает другие механизмы, называемые ссылками на символы, для ссылки на любой символ.

Поскольку в человеческих языках имеется огромное количество символов и множество способов их представления, следует позаботиться о том, чтобы эти документы могли понимать агенты пользователей во всем мире.

5.1 Набор символов документа

Для обеспечения возможность взаимодействия сетей SGML требует от каждого приложения (включая HTML) указания *набора символов документа.* Документ включает:

- *Репертуар*: Набор абстрактных символов,, таких как латинская буква "A", кириллическая буква "I", китайский иероглиф "вода" и т.д.
- *Коды*: Набор целочисленных ссылок на символы репертуара.

Каждый документ SGML (включая каждый документ HTML) - это последовательность символов из репертуара. Компьютерные системы идентифицируют каждый символ по его коду; например, в наборе символов ASCII коды 65, 66 и 67 означают символы 'A', 'B' и 'C' соответственно.

Набора символов ASCII недостаточно для такой глобальной информационной системы, как Web, поэтому HTML использует более полный набор символов, называемый *Универсальным набором символов (Universal Character Set - UCS)*, и определенный в [ISO10646]. Этот стандарт определяет репертуар тысяч символов, используемых во всем мире.

Набор символов, определенный в [ISO10646] - это посимвольный эквивалент Unicode 2.0 ([UNICODE]). Оба эти стандарта время от времени обновляются, пополняются новыми символами, об изменениях следует узнавать на соответствующих серверах Web. В этой спецификации ISO/IEC-10646 или Unicode означают этот самый набор символов. Однако в спецификации HTML Unicode также упоминается при обсуждении других вопросов, таких как алгоритм двунаправленного текста.

Набора символов документа, однако, недостаточно, чтобы агенты пользователей могли корректно интерпретировать документы HTML при типичном обмене - закодированные как последовательность байт в файле или во время передачи по сети. Агенты пользователя должны также знать кодировки символов, которые использовались для преобразования потока символов документа в поток байт.

5.2 Кодировки символов

Кодировки символов в этой спецификации имеют другие названия в других спецификациях (что может вызвать некоторую путаницу). Однако это понятие в Интернет означает примерно одно и то же. Одно и то же имя -- "charset - набор символов" - используется в заголовках протоколов, атрибутах и параметрах, ссылающихся на символы и использующих одни и те же значения из [IANA] реестра (полный список в разделе [CHARSETS]).

Параметр "charset" идентифицирует кодировку символов, которая является способом преобразования последовательности байт в последовательность символов. Это преобразование естественно вписывается в схему деятельности Web: серверы отправляют документы HTML агентам пользователей в виде потока байт; агенты пользователей

интерпретируют их как последовательность символов. Способы преобразования могут меняться от простого соответствия один к одному до сложных схем или алгоритмов переключения.
Простой техники кодировки "один байт - один символ" недостаточно для текстовых строк с таким широким репертуаром символов, как [ISO10646]. Кроме кодировок всего набора символов (например, UCS-4), имеются некоторые другие кодировки частей [ISO10646].

5.2.1 Выбор кодировки

Средства разработки (например, текстовые редакторы) могут кодировать документы HTML в кодировках по своему выбору, и этот выбор существенно зависит от соглашений, используемых системным программным обеспечением. Эти средства могут использовать любую удобную кодировку, включающую большинство символов в документе, при условии, что кодировка корректно помечена. Некоторые символы, не включенные в эту кодировку, можно представить с помощью ссылок на символы. Это всегда относится к набору символов документа, а не к кодировке символов.
Серверы и прокси могут изменять кодировку символов (что называется *транскодированием*) на лету для выполнения запросов агентов пользователей (см. раздел 14.2 [RFC2068], заголовок запроса HTTP "Accept-Charset"). Серверы и прокси не должны обслуживать документ в кодировке, включающей весь набор символов документа.
Широко используемые в Web кодировки - ISO-8859-1 (также называется "Latin-1"; используется для большинства западноевропейских языков), ISO-8859-5 (с поддержкой кириллицы), SHIFT_JIS (японская кодировка), EUC-JP (еще одна японская кодировка) и UTF-8 (вариант кодировки ISO 10646, использующий разное число байт для разных символов). Названия кодировок символов не учитывают регистр, так что, например, "SHIFT_JIS", "Shift_JIS" и "shift_jis" эквивалентны. Эта спецификация не определяет, какие кодировки символов должен поддерживать агент пользователя.
Соответствующие агенты пользователей должны корректно отображать в Unicode все символы в любых кодировках, которые они могут распознавать.

Замечания об определенных кодировках

Когда текст HTML передается в UTF-16 (charset=UTF-16), текстовые данные должны передаваться в сетевом порядке байт ("big-endian", байт высшего порядка - первый) в соответствии с [ISO10646], раздел 6.3 и [UNICODE], положение C3, страница 3-1.
Более того, чтобы повысить вероятность правильной интерпретации, рекомендуется передавать документы UTF-16, всегда начиная с символа НЕРАЗДЕЛЯЮЩИЙ ПРОБЕЛ НУЛЕВОЙ ШИРИНЫ (шестнадцатеричный код FEFF, также называется Меткой порядка байтов (Byte Order Mark - BOM)), который при обращении байт становится шестнадцатеричным FFFE, никогда не назначаемым символом. Таким образом, агент пользователя, получивший шестнадцатеричный код FFFE в качестве первых байтов текста будет знать, что в остальном тексте байты нужно обратить.
Не следует использовать формат трансформации UTF-1[ISO10646] (зарегистрированный IANA как ISO-10646-UTF-1). Информацию об ISO 8859-8 и двунаправленном алгоритме см. в разделе двунаправленности и кодировки символов.

5.2.2 Указание кодировки символов

Как сервер определяет, какая кодировка символов применяется в документе? Некоторые серверы проверяют первые несколько байт документа или сверяются с базой данных известных файлов и кодировок. Многие современные серверы Web предоставляют администраторам больше возможностей управления конфигурацией набора символов, чем старые серверы. Администраторы серверов Web должны при возможности использовать следующие механизмы для отправки параметра "charset", но должны позаботиться о том, чтобы не установить для документов ошибочное значение параметра "charset".
Как агент пользователя узнает, какая использовалась кодировка символов? Эту информацию предоставляет сервер. Лучшим способом проинформировать агента пользователя о кодировке символов документа - использовать параметр "charset" в поле

заголовка "Content-Type" протокола HTTP ([RFC2068], разделы 3.4 и 14.18) Например, следующий заголовок HTTP объявляет, что используется кодировка EUC-JP:

```
Content-Type: text/html; charset=EUC-JP
```

Определение text/html см. в разделе соответствие.
Протокол HTTP ([RFC2068], раздел 3.7.1) считает ISO-8859-1 кодировкой символов по умолчанию, если параметр "charset" в поле заголовка "Content-Type" отсутствует. На практике эта рекомендация бесполезна, поскольку некоторые серверы не позволяют отправлять параметр "charset", а некоторые могут не быть сконфигурированы для отправки этого параметр. Поэтому агенты пользователей не должны предполагать никакого значения параметра "charset".
Для указания ограничений сервера или конфигурации документы HTML могут включать явную информацию о кодировке символов документа; для предоставления такой информации агентам пользователя может использоваться элемент `META`.
Например, чтобы указать, что кодировкой символов в текущем документе является "EUC-JP", включите следующее объявление `META`:

```
<META http-equiv="Content-Type" content="text/html; charset=EUC-JP">
```

Объявление `META` должно использоваться, только если кодировка символов упорядочена так, что символы ASCII стоят на своем месте (по крайней мере, при разборе элемента `META`). Объявления `META` должны быть в тексте как можно раньше в элементе `HEAD`.
В случаях, когда ни протокол HTTP, ни элемент `META` не предоставляют информации о кодировке документа, HTML предоставляет атрибут `charset` для некоторых элементов.
Объединив все эти механизмы, автор может существенно повысить шансы на то, что, когда пользователь загружает ресурс, агент пользователя распознает кодировку символов.
Подводя итоги, соответствующие агенты пользователей при определении кодировки символов документа (от высшего приоритета к низшему) должны руководствоваться следующими источниками в соответствии с приоритетом:

1. Параметр "charset" протокола HTTP в поле "Content-Type".
2. Объявление `META`, в котором для "http-equiv" установлено "Content-Type" и установлено значение для "charset".
3. Атрибут `charset` устанавливается на элемент, обозначающий внешний ресурс.

Кроме этого списка приоритетов, агент пользователя может использовать эвристические установки и установки пользователя. Например, многие агенты пользователей используют эвристику для распознавания различных кодировок, используемых для японского языка.
Агенты пользователей обычно имеют определяемую пользователем локальную кодировку по умолчанию, которую они используют, если нет указаний кодировки.
Агенты пользователей могут обеспечивать механизм, позволяющий пользователям изменять некорректную информацию о наборе символов. Однако если агент пользователя предлагает такой механизм, он должен предлагать его только для просмотра, а не для изменения, во избежание создания Web-страниц с некорректным параметром "charset".
Примечание. *Если в каком-то приложении нужно использовать символы, не входящие в кодировку [ISO10646], этим символам должна быть назначена персональная зона во избежание конфликтов с настоящей или будущими версиями стандарта. Однако это не рекомендуется из соображений переносимости.*

5.3 Ссылки на символы

Данная кодировка символов может не содержать все символы из набора символов документа. Для таких кодировок или для таких конфигураций оборудования и программного обеспечения, не позволяющих пользователям вводить определенные символы, авторы могут использовать ссылки на символы SGML. Ссылки на символы - это независимый от кодировки механизм ввода любых символов.
Ссылки на символы в HTML могут принимать две формы:

- Числовые ссылки на символы (десятичные или шестнадцатеричные).
- Ссылки на комбинации символов.

Ссылки на символы в комментариях не имеют значения; они являются только данными комментариев.

Примечание. *HTML обеспечивает другие способы представления символов, в частности, встроенные изображения.*

Примечание. *В SGML можно в некоторых случаях не использовать заключительный символ ";" после ссылки на символы (например, в символе переноса строки или непосредственно перед тэгом). В других обстоятельствах их нельзя удалять (например, в середине слова). Мы предлагаем использовать ";" всегда во избежание проблем с агентами пользователей, для которых этот символ обязателен.*

5.3.1 Числовые ссылки на символы

Числовые ссылки на символы указывают код символа в наборе символов документа. Числовые ссылки на символы могут также принимать две формы:

- Синтаксис "&#*D*;", где *D* - десятичное число, указывает символ Unicode с десятичным номером *D*.
- Синтаксис "&#x*H*;" или "&#X*H*;", где *H* - шестнадцатеричное число, указывает на символ Unicode с шестнадцатеричным номером *H*. Шестнадцатеричные числовые ссылки учитывают регистр.

Вот некоторые примеры числовых ссылок на символы:

- å (десятичное) представляет букву "a" с кружком сверху (используемую, например, в норвежском языке).
- å (шестнадцатеричное) представляет тот же символ.
- å (шестнадцатеричное) представляет тот же символ.
- И (десятичное) представляет кириллическую заглавную букву "I".
- 水 (шестнадцатеричное) представляет китайский иероглиф "вода".

Примечание. *Хотя шестнадцатеричное представление не определено в [ISO8879], оно ожидается в новой версии, как описано в [WEBSGML]. Это соглашение особенно полезно, поскольку стандарты символов обычно используют шестнадцатеричные представления.*

5.3.2 Комбинации ссылок на символы

Чтобы дать авторам более инициативный способ использования символов, HTML предлагает набор *character entity references*. Комбинации ссылок на символы используют символические имена, так что авторам не придется запоминать коды. Например, комбинация å обозначает символ "a" нижнего регистра с кружком сверху; "å" легче запомнить, чем å.

HTML 4.0 не определяет character entity reference для каждого символа. Например, для кириллической буквы "I" нет character entity reference. См. полный список ссылок на символы, определенные в HTML 4.0.

Комбинации ссылок на символы учитывают регистр. Так, Å указывает на другой символ (A с кружком верхнего регистра), а не на å (a с кружком нижнего регистра).

Четыре ссылки нужно упомянуть специально, поскольку они часто используются для указания специальных символов:

- "<" представляет знак <.
- ">" представляет знак >.
- "&" представляет символ &.
- "" представляет знак ".

Авторы, которые хотят поместить в текст символ "<", должны использовать ссылку "<" (десятичный код ASCII 60) во избежание возможной путаницы с началом тэга (открывающий разделитель начального тэга). Точно так же следует использовать ">" (десятичный код ASCII 62) вместо ">", чтобы избежать проблем со старыми версиями агентов пользователей, некорректно принимающих их за окончание тэга (закрывающий разделитель тэга).

Авторам следует использовать "&" (десятичный код ASCII 38) вместо "&" во избежание путаницы со ссылками на символы (открывающий разделитель entity reference). Авторам также следует использовать "&" в значениях атрибутов, поскольку ссылки на символы внутри значений атрибута CDATA разрешены. Некоторые авторы используют character entity reference """ для кодирования экземпляров двойных кавычек ("), поскольку этот символ может использоваться для разделения значений атрибутов.

5.4 Неотображаемые символы

Возможно, агент пользователя не сможет отобразить все символы в документе, например, из-за отсутствия соответствующего шрифта или если символ имеет значение, которое не может быть выражено во внутренней кодировке агента пользователя и т.д. Поскольку в этом случае есть несколько вариантов, этот документ не предписывает определенной тактики. В зависимости от применения непечатные символы могут также обрабатываться дополнительной системой отображения, а не самим приложением. В случае более сложного поведения, например, настроенного для определенного сценария или языка, рекомендуем следующее поведение для агентов пользователей:

1. Примите явно видимый, но незаметный механизм для предупреждения пользователя об отсутствующих ресурсах.
2. Если отсутствующие символы представляются в другом числовом представлении, используйте шестнадцатеричную (не десятичную) форму, поскольку эта форма используется в стандартах наборов символов.