

# 6 Основные типы данных HTML

В этом разделе спецификации описываются основные типы данных, которые могут быть содержимым элементов или значением атрибутов.

Вводную информацию о чтении HTML DTD см. раздел [руководство по SGML](#).

## 6.1 Информация о регистре

Каждое определение атрибута включает информацию об учете регистра его значениями. Информация о регистре представляется следующими ключами:

**CS**

Значение учитывает регистр (агенты пользователя по-разному интерпретируют "a" и "A").

**CI**

Значение не учитывает регистр (агенты пользователя не различают "a" и "A").

**CN**

Значение не зависит от регистра, например, потому что это число или символ из набора символов документа.

**CA**

Само определение элемента или атрибута дает информацию о регистре.

**CT**

Подробнее об учете регистра см. в определении типа.

Если значением атрибута является список, ключи применяются к каждому значению в списке, если не указано обратное.

## 6.2 Основные типы SGML

В [определении типа документа](#) определяется синтаксис содержимого элемента HTML и значений атрибутов с использованием меток SGML (PCDATA, CDATA, NAME, ID и т.д.). Полные определения см. в [\[ISO8879\]](#). Вот обобщенная информация о ключах:

- **CDATA** - это последовательность символов из набора символов документа, она может включать character entities. Агенты пользователей должны интерпретировать значения атрибутов следующим образом:
  - Заменять character entities на символы,
  - Игнорировать перевод строки,
  - Заменять каждый возврат каретки или табуляцию на один пробел.

Агенты пользователей могут игнорировать пробелы в начале и в конце значений атрибута CDATA (например, " myval " интерпретируется как "myval"). Авторы не должны объявлять значения атрибутов с пробелами в начале или в конце.

На некоторые атрибуты HTML 4.0 со значениями атрибутов CDATA спецификация налагает дополнительные ограничения на множество допустимых значений атрибутов, не выраженные в DTD.

Хотя элементы [STYLE](#) и [SCRIPT](#) используют CDATA для своей модели данных, для этих элементов агенты пользователей должны обрабатывать CDATA иначе. Разметка и entities должны считаться текстом и передаваться в приложение как есть. Первое вхождение последовательности символов "</" (открывающий разделитель конечного тэга) считается концом содержимого элемента. В допустимых документах это конечный тэг элемента.

- Метки **ID** и **NAME** должны начинаться с буквы ([A-Za-z]), за которой может следовать любое число букв, цифр ([0-9]), символов переноса ("-"), символов подчеркивания ("\_"), двоеточий (":") и точек (".").
- **IDREF** и **IDREFS** - это ссылки на метки ID, определенные другими атрибутами. IDREF - одиночная метка, а IDREFS -разделенный пробелами список меток.
- Метки **NUMBER** должны содержать, по крайней мере, одну цифру ([0-9]).

## 6.3 Текстовые строки

Ряд атрибутов ([%Text](#); в DTD) принимают текст, который предназначен для чтения людьми. Вводную информацию об атрибутах можно посмотреть в [обсуждении атрибутов](#).

## 6.4 URI

В этой спецификации термин URI используется, как определено в [\[URI\]](#) (и в [\[RFC1630\]](#)). Помните, что URI включают URL (как определено в [\[RFC1738\]](#) и [\[RFC1808\]](#)).

Относительные URI разрешаются до полных URI с использованием основного URI. [\[RFC1808\]](#), раздел 3, где определен нормативный алгоритм этого процесса. Подробнее об основных URI см. в разделе [основные URI](#) в главе о [ссылках](#).

URI представляются в DTD комбинацией символов [%URI](#):

URI вообще [учитывают регистр](#). Могут быть URI, или части URI, в которых регистр не имеет значения (например, имена машин), но идентификация их может быть непроста. Пользователи должны всегда считать, что URI учитывают регистр (чтобы не ошибиться). Информацию о [символах, не входящих в набор ASCII, в значениях атрибутов URI](#) см. в приложении.

## 6.5 Цвета

Значение атрибута типа "color" ([%Color](#);) относится к определениям цветов, как указано в [\[SRGB\]](#). Значение цвета может быть шестнадцатеричным числом (которому предшествует знак диеза) или одним из следующих шестнадцати названий цветов. Названия цветов [учитывают регистр](#).

### Названия цветов и значения RGB

 Black = "#000000"	 Green = "#008000"
 Silver = "#C0C0C0"	 Lime = "#00FF00"
 Gray = "#808080"	 Olive = "#808000"
 White = "#FFFFFF"	 Yellow = "#FFFF00"
 Maroon = "#800000"	 Navy = "#000080"
 Red = "#FF0000"	 Blue = "#0000FF"
 Purple = "#800080"	 Teal = "#008080"
 Fuchsia = "#FF00FF"	 Aqua = "#00FFFF"

То есть, значения "#800080" и "Purple" оба означают пурпурный цвет.

### 6.5.1 Замечания об использовании цветов

Хотя цвета могут существенно добавлять информации в документ и повышать удобство чтения, при использовании цветов имейте в виду следующие основные принципы:

- Использование элементов и атрибутов HTML для указания цвета [нежелательно](#). Вместо этого следует использовать [таблицы стилей](#).
- Не используйте комбинации цветов, вызывающие проблемы у пользователей.
- Если Вы используете изображение в качестве фона или устанавливаете цвет фона, не забудьте становить и цвета текста.
- Цвета, указанные в элементах [BODY](#) и [FONT](#) и в [bgcolor](#) в таблицах выглядят по-разному на разных платформах (на рабочих станциях, Mac, Windows и на панелях LCD и CRT), поэтому не рассчитывайте на определенный эффект. В будущем поддержка цветовой модели [\[SRGB\]](#) вместе с цветовыми профилями ICC должна устранить эти проблемы.
- При возможности принимайте общие соглашения.

## 6.6 Длины

HTML определяет три типа значений длины для атрибутов:

1. **Пиксели:** Значение ([%Pixels](#); в DTD) - это целое, представляющее число пикселей

(на экране, на бумаге). Таким образом, значение "50" означает пятьдесят пикселей. Нормативную информацию об определении пиксела см. в [\[CSS1\]](#).

2. **Длина:** Значение ([%Length](#); в DTD) может быть %Pixel; или доля вертикального или горизонтального расстояния в процентах. Таким образом, значение "50%" означает половину доступного пространства.
3. **МультиДлина:** Значение ([%MultiLength](#); в DTD) может быть %Length; или *относительной длиной*. Относительная длина имеет форму "i\*", где "i" - целое число. При распределении пространства между элементами, конкурирующими за это пространства, агенты пользователя сначала отводят место для длин, определенных в пикселях и процентах, а затем делят оставшееся место между относительными длинами. Каждая относительная длина получает часть доступного пространства, пропорциональную целому числу, предшествующему "\*". Значение "\*" эквивалентно "1\*". Таким образом, если имеется 60 пикселей пространства после того, как агент пользователя распределит пространство для длин, определенных в пикселях и процентах, а конкурирующими относительными длинами являются 1\*, 2\* и 3\*, 1\* получит 10 пикселей, 2\* - 20 пикселей, а 3\* - 30 пикселей. Значения длин [не учитывают регистр](#).

## 6.7 Типы содержимого (типы MIME)

*Примечание.* "Тип носителя" (определенный в [\[RFC2045\]](#) и [\[RFC2046\]](#)) указывает природу связанного ресурса. Эта спецификация использует термин "тип содержимого" вместо "типа носителя" в соответствии с его использованием. Более того, "тип носителя" может означать [носитель](#), на котором агент пользователя генерирует документ.

Этот тип представлен в DTD с помощью [%ContentType](#):

Типы содержимого [учитывают регистр](#). Примеры типов содержимого включают "text/html", "image/png", "image/gif", "video/mpeg", "audio/basic", "text/tcl", "text/javascript" и "text/vbscript". Текущий список зарегистрированных типов MIME см. в [\[MIMETYPES\]](#).

*Примечание.* Тип содержимого "text/css", хотя он и не зарегистрирован в IANA, должен использоваться, если связываемым элементом является таблица стилей [\[CSS1\]](#).

## 6.8 Коды языков

Значения атрибутов, типом которых является код языка ([%LanguageCode](#) в DTD), относятся к коду языка, как указано в [\[RFC1766\]](#), раздел 2. Информацию об указании кодов языков в HTML см. в разделе [коды языков](#). В кодах языков пробелы недопустимы.

Коды языков [учитывают регистр](#).

## 6.9 Кодировки символов

Атрибуты "charset" ([%Charset](#) в DTD) относятся к кодировкам символов, как описано в разделе [кодировки символов](#). Значениями должны быть строки (например, "euc-jp") из реестра IANA (полный список см. в [\[CHARSETS\]](#)). Имена кодировок символов [учитывают регистр](#). Агенты пользователей для определения кодировки символов внешнего ресурса должны выполнять шаги, описанные в разделе [указание кодировок символов](#).

## 6.10 Отдельные символы

Определенные атрибуты вызывают отдельный символ из [набора символов документа](#). Эти атрибуты имеют тип [%Character](#) в DTD.

Отдельные символы можно указать с помощью [ссылок на символы](#) (например, "&amp;").

## 6.11 Дата и время

[\[ISO8601\]](#) позволяет много вариантов представления даты и времени. Текущая спецификация использует один из форматов, описанных в профиле [\[DATETIME\]](#) для определения допустимых строк дата/время ([%Datetime](#) в DTD).

Это следующий формат:

ГГГГ-ММ-ДДТчч:мм:ссУЧП
------------------------

где:

ГГГГ	=	год из четырех цифр
ММ	=	месяц из двух цифр (01=январь и т.д.)
ДД	=	день из двух цифр (01 - 31)
чч	=	две цифры часов (00 - 23) (до/пп НЕ допускается)
мм	=	две цифры минут (00 - 59)
сс	=	две цифры секунд (00 - 59)
УЧП	=	указатель часового пояса

Указатели часового пояса:

**Z** означает UTC (Общее скоординированное время). "Z" должно быть в верхнем регистре. **+чч:мм** указывает, что местное время отстоит на чч часов и мм минут от UTC вперед.

**-чч:мм** указывает, что местное время отстает на чч часов и мм минут от UTC.

Указанные компоненты должны присутствовать в точности, с точно такой же пунктуацией. Помните, что буква "T" отображается в строке литерально (она должна быть в верхнем регистре), для указания начала времени, как описано в [\[ISO8601\]](#)

Если генерирующее приложение не знает времени с точностью до секунды, для секунд может использоваться значение "00" (при необходимости также для минут и для часов).

*Примечание. [\[DATETIME\]](#) не касается добавочных секунд.*

## 6.12 Типы ссылок

Авторы могут использовать следующие распознаваемые типы ссылок, перечисленные здесь вместе с условными интерпретациями. В DTD [%LinkTypes](#) означает список типов ссылок, разделенных пробелами. Символы пробелов в типах ссылок не допускаются.

Эти типы ссылок не учитывают регистр, т.е. "Alternate" означает то же, что и "alternate".

Агенты пользователей, поисковые машины и т.д. могут интерпретировать эти типы ссылок несколькими способами. Например, агенты пользователя могут предоставлять доступ к связанным документам с помощью навигационной панели.

### Alternate

Обозначает альтернативные версии документа, в котором находится ссылка. Вместе с атрибутом [lang](#) означает переведенную версию документа. Вместе с атрибутом [media](#) означает версию, созданную для другого носителя.

### Stylesheet

Обозначает внешнюю таблицу стилей. Подробнее см. раздел о [внешних таблицах стилей](#). Используется вместе с типом ссылки "Alternate" для таблиц стилей, выбираемых пользователем.

### Start

Обозначает первый документ в наборе. Этот тип ссылки сообщает поисковым машинам о том, какой документ автор считает началом набора.

### Next

Обозначает следующий документ в линейной последовательности документов. Агенты пользователей могут предварительно загружать документ "next" для сокращения времени загрузки.

### Prev

Обозначает предыдущий документ в упорядоченной серии документов. Некоторые агенты пользователей также поддерживают синоним "Previous".

### Contents

Обозначает документ, служащий содержанием. Некоторые агенты пользователей также поддерживают синоним *ToC* (из "Table of Contents").

### Index

Обозначает документ, являющийся указателем текущего документа.

### Glossary

Обозначает документ - глоссарий терминов, относящихся к текущему документу.

### Copyright

Обозначает замечание об авторском праве для текущего документа.

## Chapter

Обозначает документ, являющийся главой в наборе документов.

## Section

Обозначает документ, являющийся разделом в наборе документов.

## Subsection

Обозначает документ, являющийся подразделом в наборе документов.

## Appendix

Обозначает документ, являющийся приложением в наборе документов.

## Help

Обозначает документ, содержащий справку (более подробная информация, ссылки на другие информационные ресурсы и т.д.)

## Bookmark

Обозначает закладку. Закладка - это ссылка на ключевую точку в расширенном документе. Атрибут [title](#) может использоваться, например, для пометки закладки. Помните, что в каждом документе можно определить несколько закладок.

Авторы могут определить дополнительные типы ссылок, не описанные в этой спецификации. При этом они должны использовать [профиль](#) для указания соглашений, используемых для определения типов ссылок. Подробнее см. атрибут [profile](#) элемента [HEAD](#).

Дальнейшее обсуждение типов ссылок см. в разделе [ссылки в документах HTML](#).

## 6.13 Дескрипторы носителей

Ниже приведен список распознаваемых дескрипторов носителей ( [%MediaDesc](#) в DTD).

**Screen** - Предназначен для экранов компьютеров, не разделенных на страницы.

**Tty** - Предназначен для носителя с фиксированной сеткой символов, таких как телетайпы, терминалы или переносные устройства с ограниченными возможностями отображения.

**tv** - Предназначен для устройств типа телевизора (низкое разрешение, цвета, ограниченные возможности прокрутки).

**projection** Предназначен для проекторов.

**handheld** Предназначен для карманных устройств (небольшой экран, монохромный, растровая графика, ограниченный диапазон).

**print** Предназначен для страничных, непрозрачных материалов и документов, просматриваемых на экране в режиме предварительного просмотра печати.

**braille** Предназначен для тактильных устройств с алфавитом Брайля.

**aural** Предназначен для синтезаторов речи.

**all** Для всех устройств.

В будущих версиях HTML могут быть введены новые значения и разрешены параметризованные значения. Для упрощения введения этих расширений соответствующие спецификации агенты пользователя должны иметь возможность анализировать значение атрибута [media](#) следующим образом:

1. Значение - это разделенный запятыми список элементов. Например,

```
2.      media="screen, 3d-glasses, print and resolution > 90dpi"
```

отображается в :

```
"screen"  
"3d-glasses"  
"print and resolution > 90dpi"
```

3. Каждый элемент усекается перед первым символом, не являющимся буквой кодировки US ASCII [a-zA-Z] (десятичные коды Unicode 65-90, 97-122), цифрой [0-9] (шестнадцатеричные коды Unicode 30-39) или знаком переноса (45). В данном примере получается:

```
4.      "screen"  
5.      "3d-glasses"
```

7. Затем [с учетом регистра](#) проводится сверка с набором определенных выше типов дескрипторов. Агенты пользователей могут игнорировать несовпадающие элементы. В данном примере останутся только элементы screen и print.

*Примечание. Таблицы стилей могут включать вариации в зависимости от носителя (например, конструкция CSS @media). В таких случаях имеет смысл использовать "media=all".*

## 6.14 Данные сценария

Данные сценария ([%Script](#); в [DTD](#)) могут быть содержимым элемента [SCRIPT](#) и значением [атрибутов внутренних событий](#). Агенты пользователей не должны оценивать данные сценариев в разметке HTML, а должны передавать эти данные ядру сценариев.

Учет регистра в данных сценариев зависит от языка сценариев.

Помните, что данные сценариев, являющиеся содержимым элемента, не могут содержать [ссылки на символы](#), но данные сценария, являющиеся значением атрибута, могут. В приложении приводится информация об [указании данных, отличных от данных HTML](#).

## 6.15 Данные таблиц стилей

Данные таблиц стилей ([%StyleSheet](#); в [DTD](#)) могут быть содержимым элемента [STYLE](#) и значением атрибута [style](#). Агенты пользователей не должны оценивать данные стилей в разметке HTML.

Учет регистра данных стиля зависит от языка таблиц стилей.

Помните, что данные таблиц стилей, являющиеся содержимым элемента, не могут включать [ссылки на символы](#), но данные таблиц стилей, являющиеся значением атрибута, могут включать их. В приложении приводится дальнейшая информация об [указании данных, отличных от данных HTML](#).

## 6.16 Целевые имена кадров

За исключением приведенных ниже зарезервированных имен, целевые имена кадров ([%FrameTarget](#); в DTD) должны начинаться с алфавитных символов (a-zA-Z). Агенты пользователей должны игнорировать все остальные имена.

Следующие target names зарезервированы и имеют специальные значения.

**\_blank** Агенты пользователей должны загружать документ в новое окно без имени.

**\_self** Агенты пользователей должны загружать документ в тот же кадр, в котором находится ссылающийся на него документ.

**\_parent** Агенты пользователей должны загружать документ в непосредственный родительский кадр этого кадра во [FRAMESET](#). Это значение эквивалентно `_self`, если текущий кадр не имеет родительского кадра.

**\_top** Агенты пользователей должны загружать документ в полное окно (закрывая все остальные кадры). Это значение эквивалентно `_self`, если у текущего кадра нет родительского кадра.