

7 Глобальная структура документа в формате HTML

7.1 Введение в структуру документа HTML

Документ в формате HTML 4.0 состоит из трех частей:

1. строки, содержащей [информацию о версии HTML](#),
2. раздела заголовков (определяемого элементом [HEAD](#)),
3. тела, которое включает собственно содержимое документа. Тело может вводиться элементом [BODY](#) или элементом [FRAMESET](#).

Перед каждым элементом или после каждого элемента может находиться пустое пространство (пробелы, переход на новую строку, табуляции и комментарии). Разделы 2 и 3 должны отделяться элементом [HTML](#).

Вот пример простого документа HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Мой первый документ HTML</TITLE>
  </HEAD>
  <BODY>
    <P>Всем привет!
  </BODY>
</HTML>
```

7.2 Информация о версии HTML

В документе HTML должна быть объявлена используемая в нем версия языка HTML.

Объявление типа документа указывает определение типа документа (DTD), используемое в этом документе (см. [ISO8879](#)).

HTML 4.0 определяет три DTD, так что авторы должны включать в свои документы одно из следующих объявлений типов. Разница между DTD заключается в поддерживаемых ими элементах.

- [HTML 4.0 Strict DTD \(строгое определение\)](#) включает все элементы и атрибуты, не являющиеся [нежелательными](#) и не использующиеся в документах с кадрами. Для документов, использующих это DTD, используйте такое объявление типа документа:

```
• <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
• "http://www.w3.org/TR/REC-html40/strict.dtd">
```

- [HTML 4.0 Transitional DTD \(переходное определение\)](#) включает все, что включено в строгое DTD, а также нежелательные элементы и атрибуты (большинство из которых относится к визуальному представлению). Для документов, использующих это DTD, используйте такое объявление типа документа:

```
• <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
• "http://www.w3.org/TR/REC-html40/loose.dtd">
```

- [HTML 4.0 Frameset DTD \(определение для кадров\)](#) включает все, что включено в переходное DTD, а также кадры. Для документов, использующих это DTD, используйте такое объявление типа:

```
• <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
• "http://www.w3.org/TR/REC-html40/frameset.dtd">
```

URI в каждом объявлении типа документа позволяет агентам пользователей загрузить DTD и все необходимые [entity sets](#). Следующие URI указывают на DTD и наборы комбинаций для HTML 4.0, поддерживаемого W3C:

- "<http://www.w3.org/TR/REC-html40/strict.dtd>" - строгое DTD по умолчанию
- "<http://www.w3.org/TR/REC-html40/loose.dtd>" - переходное DTD
- "<http://www.w3.org/TR/REC-html40/frameset.dtd>" -- DTD для документов, исполь-

зующих кадры

- ["http://www.w3.org/TR/REC-html40/HTMLlat1.ent"](http://www.w3.org/TR/REC-html40/HTMLlat1.ent) - комбинации Latin-1
- ["http://www.w3.org/TR/REC-html40/HTMLsymbol.ent"](http://www.w3.org/TR/REC-html40/HTMLsymbol.ent) - комбинации символов
- ["http://www.w3.org/TR/REC-html40/HTMLspecial.ent"](http://www.w3.org/TR/REC-html40/HTMLspecial.ent) - специальные комбинации

Связь между общими идентификаторами и файлами можно указать с использованием файла каталога, за которым следует формат, рекомендуемый Открытым консорциумом SGML (см. [\[SGMLOPEN\]](#)). [Примерный файл каталога для HTML 4.0](#) включен в начало раздела о справочной информации об SGML для HTML. Две последние буквы декларации указывают язык DTD. Для HTML это всегда английский ("EN").

7.3 Элемент HTML

```
<!ENTITY % html.content "HEAD, BODY">

<!ELEMENT HTML O O (%html.content;) -- корневой элемент документа -->
<!ATTLIST HTML
  %i18n; -- lang, dir --
>
```

Начальный тег: **необязателен**, Конечный тег: **необязателен**

Определения атрибутов

версия = [CDATA\[CN\]](#)

Нежелателен. Значение этого атрибута указывает версию HTML DTD, которой подчиняется этот документ. Этот атрибут является нежелательным, поскольку он является избыточным при наличии [информации о версии](#), указываемой в объявлении типа документа.

Атрибуты, определяемые в любом другом месте

- [lang](#) ([информация о языке](#)), [dir](#) ([направление текста](#))

После объявления типа документа остальная часть документа HTML содержится в элементе [HTML](#). Таким образом, типичный документ HTML имеет такую структуру:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
...Здесь идут заголовок, тело и т.д...
</HTML>
```

7.4 Заголовок документа

7.4.1 Элемент HEAD

```
<!-- %head.misc;, определенный ранее как "SCRIPT|STYLE|META|LINK|OBJECT" -->
<!ENTITY % head.content "TITLE & BASE?">

<!ELEMENT HEAD O O (%head.content;) +(%head.misc;) - заголовок документа -->
<!ATTLIST HEAD
  %i18n; -- lang, dir --
  profile %URI; #IMPLIED -- каталог метаинформации --
>
```

Начальный тег: **не обязателен**, Конечный тег: **не обязательно**

Определения атрибутов

profile = [uri](#) [CT]

Этот атрибут указывает местоположение одного или нескольких профилей метаданных, отделяемых пробелами. Для расширения в будущем агенты пользователей должны предполагать, что значение является списком, хотя в данной спецификации значащим считается только первый URI. [Профили](#) обсуждаются ниже в разделе о [метаданных](#).

Атрибуты, определяемые в любом другом месте

- [lang](#) ([информация о языке](#)), [dir](#) ([направление текста](#))

Элемент [HEAD](#) содержит информацию о текущем документе, такую как заголовок, ключевые слова, которые могут использоваться поисковыми машинами, и другие данные, которые не считаются содержимым документа. Агенты пользователей обычно не используют

при генерации элементы из раздела [HEAD](#). Однако они могут предоставлять пользователям информацию из раздела [HEAD](#) с помощью своих собственных механизмов.

7.4.2 Элемент `TITLE`

```
<!-- Элемент TITLE не считается частью текста.
      Он должен отображаться, например, в качестве заголовка страницы
      или окна. В документе должен быть ровно один заголовок.
-->
<!ELEMENT TITLE -- (#PCDATA) -(%head.misc;) - заголовок документа -->
<!ATTLIST TITLE %i18n>
```

*Начальный тег: **обязателен**, Конечный тег: **обязателен***

Атрибуты, определяемые в любом другом месте

- [lang](#) ([информация о языке](#)), [dir](#) ([направление текста](#))

Каждый документ HTML **должен** иметь элемент [TITLE](#) в разделе [HEAD](#).

Авторы должны использовать элемент [TITLE](#) для идентификации содержимого документа. Поскольку пользователи часто обращаются к документам за пределами контекста, авторам следует обеспечивать заголовки в широком контексте. Таким образом, вместо заголовков типа "Введение", ничего не говорящих о контексте, авторам следует использовать заголовки типа "Введение в средневековое пчеловодство".

Из соображений доступности агенты пользователей всегда должны делать содержимое элемента [TITLE](#) доступным пользователям (включая элементы [TITLE](#) в кадрах). Механизм этого зависит от агента пользователя (например, в виде заголовка или произносимый).

Заголовки могут включать [комбинации символов](#) (для символов со знаком ударения, специальных символов и т.д.), но не могут содержать другой разметки. Вот образец заголовка документа:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
      "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
<HEAD>
<TITLE>Исследование динамики популяции</TITLE>
... другие элементы заголовка...
</HEAD>
<BODY>
... тело документа...
</BODY>
</HTML>
```

7.4.3 Атрибут `title`

Определения атрибутов

`title = text [CS]` Этот атрибут предлагает информацию об элементе, для которого он установлен.

В отличие от элемента [TITLE](#), который предоставляет информацию обо всем документе и может присутствовать в тексте только один раз, атрибут [title](#) может сопровождать любое число элементов. Узнать, поддерживает ли элемент этот атрибут, можно в определении элемента.

Значения атрибута [title](#) могут использоваться агентами пользователя в генерации изображения по-разному. Например, визуальные браузеры часто отображают заголовок как подсказку (краткое сообщение, которое появляется, если Вы указываете на объект). Аудиоагенты пользователя могут проговаривать информацию заголовка. Например, установка этого атрибута для ссылки позволяет агентам пользователей (визуальным и не визуальным) сообщить пользователям о природе связанного ресурса:

```
...текст...
Вот фотография
<A href="http://someplace.com/neatstuff.gif" title="Me scuba diving">
  как я нырял в прошлом году
</A>
```

...еще текст...

Атрибут `title` играет дополнительную роль при использовании с элементом `LINK` для назначения [внешней таблицы стилей](#). Подробнее см. в разделе о [ссылках и таблицах стилей](#).

Примечание. Для улучшения качества синтеза речи в случае плохой обработки стандартными механизмами, будущие версии HTML могут включать атрибут для кодирования фонематической информации.

7.4.4 Метаданные

Во время написания этой спецификации проводились работы по предоставлению авторам возможности задания большего количества машинно-читаемой информации о документах HTML и других сетевых ресурсах. Для метаданных W3C разработан Язык описания ресурсов (см. [\[RDF\]](#)).

HTML позволяет авторам указывать метаданные -- информацию о документе вместо содержимого документа - множеством способов.

Например, чтобы указать автора документа, можно использовать элемент `META` следующим образом:

```
<META name="Author" content="Дэйв Рэггетт">
```

Элемент `META` задает свойство (здесь "Author (Автор)") и назначает ему значение (здесь - "Дэйв Рэггетт").

Данная спецификация не определяет набор допустимых свойств метаданных. Значение свойства и набор допустимых значений этого свойства должны определяться в относительном словаре, называемом [профилем](#). Например, профиль, разработанный для помощи в индексировании документов для поисковых машин, может определять такие свойства как "author", "copyright", "keywords" и т.д.

Указание метаданных

В общем случае задание метаданных состоит из двух шагов:

1. Объявление свойства и его значения. Это можно сделать двумя способами:
 1. Из документа с помощью элемента `META`.
 2. Не из документа с помощью ссылки на метаданные через элемент `LINK` (см. раздел о [типах ссылок](#)).
2. Сославшись на [профиль](#), в котором определяются свойства и их допустимые значения. Для назначения профиля используйте атрибут `profile` элемента `HEAD`. Помните, что поскольку профиль определяется для элемента `HEAD`, этот профиль применяется ко всем элементам `META` и `LINK` в заголовке документа.

Агенты пользователей не обязательно должны поддерживать механизмы метаданных. В этой спецификации не определяется интерпретация метаданных.

Элемент META

```
<!ELEMENT META - O EMPTY -- общая метаинформация -->
<!ATTLIST META
  %i18n; -- lang, dir, для использования с содержимым --
  http-equiv NAME #IMPLIED -- имя заголовка ответа HTTP --
  name NAME #IMPLIED -- имя метаинформации --
  content CDATA #REQUIRED -- связанная информация --
  scheme CDATA #IMPLIED -- выбор формы содержимого --
  >
```

Начальный тег: **обязателен**, Конечный тег: **запрещен**

Определения атрибутов

Для следующих атрибутов допустимые значения и интерпретация зависят от [профиля](#):
name = `name` [\[CS\]](#)

Этот атрибут определяет имя свойства. В данной спецификации не перечисляются допустимые значения этого атрибута.

content = [CDATA](#) [CS]

Этот атрибут определяет значение свойства. В данной спецификации не перечисляются допустимые значения этого атрибута.

scheme = [CDATA](#) [CS]

Этот атрибут дает имя схеме, используемой для интерпретации значения свойства (подробнее см. раздел о [профилях](#)).

http-equiv = [name](#) [CI]

Этот атрибут может использоваться вместо атрибута name. Серверы HTTP используют этот атрибут для сбора информации для заголовков сообщений ответов HTTP.

Атрибуты, определяемые в любом другом месте

- [lang](#) ([информация о языке](#)), [dir](#) ([направление текста](#))

Элемент [META](#) может использоваться для идентификации свойств документа (например, автора, срок истечения, список ключевых слов и т.д.) и назначения им значений. В этой спецификации не определяется нормативный набор значений.

Каждый элемент [META](#) задает пару свойство/значение. Атрибут name определяет свойство, а атрибут [content](#) - значение.

Например, в следующем объявлении устанавливается значение свойства Author:

```
<META name="Author" content="Дэйв Перретт">
```

Атрибут [lang](#) может использоваться с элементом [META](#) для указания языка значения атрибута [content](#). Это позволяет синтезаторам речи использовать правила произношения для разных языков.

В этом примере имя автора объявляется на французском языке:

```
<META name="Author" lang="fr" content="Arnaud Le Hors">
```

Примечание. Элемент [META](#) - это общий механизм задания метаданных. Однако некоторые элементы и атрибуты HTML уже обрабатывают некоторые части метаданных и могут использоваться авторами вместо элементов [META](#) для указания этих частей: элементы [TITLE](#), [ADDRESS](#), [INS](#) и [DEL](#), атрибут [title](#) и атрибут [cite](#).

Примечание. Если свойство, заданное с помощью элемента [META](#), принимает значение [URI](#), некоторые авторы предпочитают указывать метаданные с помощью элемента [LINK](#). Таким образом, следующее объявление:

```
<META name="DC.identifier"
      content="ftp://ds.internic.net/rfc/rfc1866.txt">
```

можно также записать следующим образом:

```
<LINK rel="DC.identifier"
       type="text/plain"
       href="ftp://ds.internic.net/rfc/rfc1866.txt">
```

[META](#) и заголовки HTTP

Атрибут [http-equiv](#) может использоваться вместо атрибута name; он особенно важен, если документы загружаются по протоколу передачи гипертекста (HTTP). Серверы HTTP могут использовать имя свойства, указанное в атрибуте [http-equiv](#) для создания заголовка в ответе HTTP в стиле [RFC822](#). Подробнее о допустимых заголовках HTTP см. в спецификации HTTP ([RFC2068](#)).

В следующем примере объявление [META](#):

```
<META http-equiv="Expires" content="Tue, 20 Aug 1996 14:25:27 GMT">
```

вернет следующий заголовок HTTP:

```
Expires: Tue, 20 Aug 1996 14:25:27 GMT
```

Это может использоваться кэш-памятью для определения того, когда следует загрузить

новую копию связанного документа.

Некоторые агенты пользователей поддерживают использование элемента [META](#) для обновления текущей страницы по истечении указанного числа секунд с возможностью замены на другой URI.

```
<META http-equiv="refresh" content="3,http://www.acme.com/intro.html">
```

content - это число, указывающее задержку в секундах, за которым следует URI, который нужно загрузить по прошествии этого времени. Этот механизм широко используется для создания кратковременных заставок. Однако поскольку некоторые агенты пользователей не поддерживают этот механизм, авторам следует включить в заставку возможность перейти на следующую страницу (чтобы они на "зависли" на заставке).

[META](#) и поисковые машины

Основной способ использования элемента [META](#) - задание ключевых слов, которые поисковые машины могут использовать для улучшения результатов поиска. Если информация о документе представлена в нескольких элементах [META](#) в зависимости от языка, поисковые машины могут фильтровать атрибут [lang](#) и отображать результаты поиска с использованием выбранного пользователем языка. Например,

```
<-- Для говорящих на американском английском -->
<META name="keywords" lang="en-us"
      content="vacation, Greece, sunshine">
<-- Для говорящих на британском английском -->
<META name="keywords" lang="en"
      content="holiday, Greece, sunshine">
<-- для русскоязычных пользователей -->
<META name="keywords" lang="fr"
      content="отпуск, Греция, солнце">
```

Эффективность поисковых машин можно также повысить с использованием элемента [LINK](#) для задания ссылок на переводы документа на другие языки, ссылки на версии документа для другого носителя (например, PDF), и, если документ является частью набора, ссылки на соответствующую начальную точку для просмотра набора.

Дальнейшая информация находится в разделе [Как помочь поисковым машинам проиндексировать Ваш Web-сервер](#).

[META](#) и PICS

Platform for Internet Content Selection (Платформа для выбора содержимого Интернет) (PICS, specified in [PICS](#)) - это инфраструктура для связывания меток (метаданных) с содержимым Интернет. Созданная для помощи родителям и учителям в управлении доступом детей к Интернет, она также упрощает другое использование меток, включая управление подписью кодов, секретностью и правами интеллектуальной собственности.

Этот пример иллюстрирует использование объявления [META](#) для включения метки PICS 1.1:

```
<HEAD>
<META http-equiv="PICS-Label" content='
(PICS-1.1 "http://www.gcf.org/v2.5"
  labels on "1994.11.05T08:15-0500"
  until "1995.12.31T23:59-0000"
  for "http://w3.org/PICS/Overview.html"
  ratings (suds 0.5 density 0 color/hue 1))
'>
<TITLE>... название документа ...</TITLE>
</HEAD>
```

[META](#) и информация по умолчанию

Элемент [META](#) может использоваться для указания информации по умолчанию для документа в следующих случаях:

- [Язык сценариев по умолчанию](#).
- [Язык таблиц стилей](#) по умолчанию.

- [Кодировка символов документа.](#)

В следующем примере для документа указывается [кодировка символов](#) ISO-8859-5

```
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">
```

Профили метаданных

Атрибут [profile](#) элемента [HEAD](#) указывает местоположение профиля метаданных. Значением атрибута [profile](#) является URI. Агенты пользователей могут использовать этот URI двояко:

- Как глобальное уникальное имя. Агенты пользователя могут распознавать имя (не загружая в действительности профиль) и выполнять некоторые действия на базе известных соглашений для этого профиля. Например, поисковые машины могут обеспечивать интерфейс для поиска в каталогах документов HTML, где все эти документы используют один и тот же профиль для представления записей каталога.
- Как ссылку. Агенты пользователей могут разменовывать URI и выполнять некоторые действия на базе определений из профиля (например, авторизовать использование профиля в текущем документе HTML). В этой спецификации не определяются форматы профилей.

В этом примере используется гипотетический профиль, определяющий полезные свойства для индексирования документов. Для свойств, определяемых этим профилем - включая "author", "copyright", "keywords" и "date" -- значения устанавливаются с помощью последовательных объявлений [META](#).

```
<HEAD profile="http://www.acme.com/profiles/core">
<TITLE>How to complete Memorandum cover sheets</TITLE>
<META name="author" content="John Doe">
<META name="copyright" content="&copy; 1997 Acme Corp.">
<META name="keywords" content="corporate, guidelines, cataloging">
<META name="date" content="1994-11-06T08:49:37+00:00">
</HEAD>
```

Во время написания этой спецификации распространенной стала практика использования форматов дат, описанных в [\[RFC2068\]](#), раздел 3.3. Поскольку обработка этих форматов относительно сложна, мы рекомендуем авторам использовать формат даты [\[ISO8601\]](#). Подробнее см. разделы об элементах [INS](#) и [DEL](#).

Атрибут [scheme](#) позволяет авторам предоставлять агентам пользователей дополнительный контекст для корректной интерпретации метаданных. Иногда такая дополнительная информация может иметь важное значение, например, если метаданные указаны в другом формате. Например, автор может указать дату в формате "10-9-97" (неоднозначно); означает ли это 9 октября 1997 г. или 10 сентября 1997 г.? Значение атрибута [scheme](#) "Month-Date-Year" устранил неоднозначность.

В других случаях атрибут [scheme](#) может предоставлять агентам пользователей полезную, но не столь важную информацию.

Например, следующее объявление [scheme](#) поможет агентам пользователей определить, что значение свойства "identifier" - номер кода ISBN:

```
<META scheme="ISBN" name="identifier" content="0-8230-2355-9">
```

Значения атрибута [scheme](#) зависят от свойства name и связанного [профиля](#).

***Примечание.** Примером профиля является Dublin Core ([\[DCORE\]](#)). Этот профиль определяет набор рекомендуемых свойств для электронных библиографических описаний и предназначен для обеспечения интероперабельности в несопоставимых моделях.*

7.5 Тело документа

7.5.1 Элемент BODY

```
<!ELEMENT BODY O O (%block;|SCRIPT)+ +(INS|DEL) - тело документа -->
<!ATTLIST BODY
  %attrs;                -- %coreattrs, %i18n, %events --
```

```
onload      %Script; #IMPLIED -- документ загружен --
onunload    %Script; #IMPLIED -- документ удален --
>
```

Начальный тег: не обязателен, Конечный тег: не обязателен

Определения атрибутов

background = [uri](#)[CI]

Нежелателен. Значение этого атрибута - URI, указывающий на изображение. Это изображение является фоном (для визуальных браузеров).

text = [color](#)[CI]

Нежелателен. Этот атрибут устанавливает цвет текста (для визуальных браузеров).

link = [color](#) [CI]

Нежелателен. Этот атрибут устанавливает цвет текста гипертекстовых ссылок, по которым Вы не переходили (для визуальных браузеров).

vlink = [color](#) [CI]

Нежелателен. Этот атрибут устанавливает цвет текста ссылок, по которым Вы переходили (для визуальных браузеров).

alink = [color](#) [CI]

Нежелателен. Этот атрибут устанавливает цвет текста ссылок, когда они выбраны пользователем.

Атрибуты, определяемые в любом другом месте

- [id](#), [class](#) ([идентификаторы в пределах документа](#))
- [lang](#) ([информация о языке](#)), [dir](#) ([направление текста](#))
- [title](#) ([заголовок элемента](#))
- [style](#) ([встроенная информация о стиле](#))
- [bgcolor](#) ([цвет фона](#))
- [onload](#), [onunload](#) ([внутренние события](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([внутренние события](#))

В теле документа располагается содержание документа. Содержимое может представляться агентом пользователя несколькими способами. Например, для визуальных браузеров Вы можете считать тело документа полотном, на котором отображаются: текст, изображения, цвета, рисунки и т.д. Для аудиоагентов пользователей оно может произноситься. Поскольку предпочтительным способом представления документов теперь являются [таблицы стилей](#), атрибуты представления в теге `BODY` являются [нежелательными](#).

ПРИМЕР НЕЖЕЛАТЕЛЬНОГО ИСПОЛЬЗОВАНИЯ:

В следующем фрагменте кода HTML показано использование [нежелательного](#) атрибута. Он устанавливает белый цвет фона, черный текста и красный гиперссылок изначально, функции при активизации ссылок и коричневый для ссылок, по которым Вы переходили.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
<HTML>
<HEAD>
<TITLE>Динамика популяции</TITLE>
</HEAD>
<BODY bgcolor="white" text="black"
link="red" alink="fuchsia" vlink="maroon">
... тело документа...
</BODY>
</HTML>
```

Используя [таблицы стилей](#), того же эффекта можно достичь следующим образом:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
```



```

<HEAD>
<TITLE>Динамика популяции</TITLE>
<STYLE type="text/css">
BODY { background: white; color: black}
A:link { color: red }
A:visited { color: maroon }
A:active { color: fuchsia }
</STYLE>
</HEAD>
<BODY>
... тело документа...
</BODY>
</HTML>

```

Использование внешних (связываемых) таблиц стилей обеспечивает гибкость при изменении представления без пересмотра источника документа HTML:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
<HEAD>
<TITLE>Динамика популяции</TITLE>
<LINK rel="stylesheet" type="text/css" href="smartstyle.css">
</HEAD>
<BODY>
... тело документа...
</BODY>
</HTML>

```

Кадры и тела документов HTML. В документах, содержащих кадры, элемент `BODY` заменяется элементом `FRAMESET`. Подробнее см. в разделе о [кадрах](#).

7.5.2 Идентификаторы элементов: атрибуты `id` and `class`

Определения атрибутов

`id = name` [CS]

Этот атрибут назначает элементу имя. Имя в пределах документа должно быть уникальным.

`class = cdata-list` [CS]

Этот атрибут назначает элементу имя класса или набор имен классов. Одно и то же имя или имена классов могут быть назначены любому числу элементов. Несколько имен классов должны быть разделены пробелами.

Атрибут `id` назначает элементу уникальный идентификатор (который может проверяться синтаксическим анализатором SGML). Например, следующие абзацы распознаются по значениям их атрибутов `id`:

```

<P id="myparagraph"> Этот абзац имеет уникальное имя.</P>
<P id="yourparagraph"> Этот абзац тоже имеет уникальное имя.</P>

```

Атрибут `id` имеет в HTML несколько ролей:

- Способ выбора [таблицы стиля](#).
- Назначение цели ([якорь](#)) для гипертекстовых ссылок.
- Средство ссылки на определенный элемент [сценария](#).
- Имя объявленного объекта [ОБЪЕКТ](#).
- В целях обработки агентами пользователей (например, для полей идентификации при извлечении данных из страниц HTML в базу данных, преобразовании из HTML в другие форматы и т.д.).

Атрибут `class`, с другой стороны, назначает одно или несколько имен классов элементу; при этом элемент может называться принадлежащим к этим классам. Имя класса может использоваться несколькими экземплярами элемента. Атрибут `class` имеет в HTML несколько ролей:

- Способ выбора [таблицы стиля](#) (когда автор назначает информацию о стиле набору

элементов).

- Для общей обработки агентами пользователей.

Далее элемент `example` `SPAN` используется вместе с атрибутами `id` и `class` для пометки сообщений документа. Сообщения отображаются в английской и русской версиях.

```
<!-- английские сообщения -->
<P><SPAN id="msg1" class="info" lang="en">Variable declared twice</SPAN>
<P><SPAN id="msg2" class="warning" lang="en">Undeclared variable</SPAN>
<P><SPAN id="msg3" class="error" lang="en">Bad syntax for variable name</SPAN>
<!-- русские сообщений -->
<P><SPAN id="msg1" class="info" lang="fr">Переменная объявлена дважды</SPAN>
<P><SPAN id="msg2" class="warning" lang="fr">Переменная не объявлена</SPAN>
<P><SPAN id="msg3" class="error" lang="fr">Синтаксическая ошибка в имени
переменной</SPAN>
```

Следующие правила стиля CSS сообщат агентам пользователя о необходимости отображения информации зеленым цветом, предупреждений - желтым, а сообщений об ошибках - красным:

```
SPAN.info { color: green }
SPAN.warning { color: yellow }
SPAN.error { color: red }
```

Помните, что русской `"msg1"` и английское `"msg1"` не могут отображаться в одном документе, поскольку они используют одно и то же значение атрибута `id`. Авторы могут извлечь дальнейшую пользу, используя атрибут `id` для усовершенствования представления отдельных сообщений, указания их в качестве целей (якорей) и т.д. Почти каждому элементу HTML может быть назначен идентификатор и информация о классе.

Предположим, мы пишем документ о языке программирования. Этот документ должен включать ряд отформатированных примеров. Для форматирования примеров мы используем элемент `PRE`. Мы также назначаем цвет фона (зеленый) всем экземплярам элемента `PRE`, принадлежащим классу `"example"`.

```
<HEAD>
<TITLE>... название документа...</TITLE>
<STYLE type="text/css">
PRE.example { background : green }
</STYLE>
</HEAD>
<BODY>
<PRE class="example" id="example-1">
...код примера...
</PRE>
</BODY>
```

Установив атрибут `id` для этого примера, мы можем (1) создать на него гиперссылку и (2) использовать информацию о стиле, отличную от определенной в таблице, для одного экземпляра информации о стиле.

Примечание. Атрибут `id` использует одно пространство имен с атрибутом `name`, если он используется для имен якорей. Подробнее см. в разделе о [якорях с id](#).

7.5.3 Элементы уровня блока и встроенные элементы

Некоторые элементы HTML, которые могут присутствовать внутри тега `BODY`, называются элементами "уровня блока", в то время как другие - "встроенными" (также называемыми элементами "уровня текста"). Разница между ними описана ниже:

Модель содержимого

Обычно элементы уровня блока могут содержать встроенные элементы и другие элементы уровня блока. Обычно встроенные элементы могут содержать только данные и другие встроенные элементы. Этому структурному различию свойственна идея о том, что элементы блока создают "большие" структуры, чем встроенные элементы.

Форматирование

По умолчанию элементы уровня блока форматируются иначе, чем встроенные элементы. Обычно элементы уровня блока начинаются с новой строки, а встроенные элементы - нет. Информацию о пробелах, переносах строки и форматировании блоков см. в разделе о [тексте](#).

Направление

По техническим причинам, затрагивающим алгоритм двунаправленного текста [\[UNICODE\]](#), элементы уровня блока и встроенные элементы различаются способами наследования информации о направлении. Подробнее см. в разделе о [наследовании направления текста](#).

[Таблицы стилей](#) обеспечивают средства задания отображения произвольных элементов, включая и то, генерируется ли элемент как блочный или встроенный. В некоторых случаях, например, в случае встроенного стиля для элементов списка, это может быть полезным, но вообще говоря, авторам следует избегать такого переопределения интерпретации элементов языка HTML.

Изменение традиционных выражений представления для элементов уровня блока и встроенных элементов влияет на алгоритм двунаправленного текста. Подробнее см. раздел о [влиянии таблиц стилей на двунаправленность](#).

7.5.4 Группировка элементов: элементы DIV и SPAN

```
<!ELEMENT DIV - - (%flow;)*           -- общий контейнер языка/стиля -->
<!ATTLIST DIV
  %attrs;                               -- %coreattrs, %i18n, %events --
  >
<!ELEMENT SPAN - - (%inline;)*        -- общий контейнер языка/стиля -->
<!ATTLIST SPAN
  %attrs;                               -- %coreattrs, %i18n, %events --
  >
```

*Начальный тег: **обязателен**, Конечный тег: **обязателен***

Определения атрибутов

href = [uri](#) [\[CT\]](#)

Этот атрибут указывает ресурс, предоставляющий дополнительную информацию о содержимом элемента [DIV](#) или [SPAN](#).

Атрибуты, определяемые в любом другом месте

- [id](#), [class](#) ([идентификаторы в пределах документа](#))
- [lang](#) ([информация о языке](#)), [dir](#) ([направление текста](#))
- [title](#) ([заголовок элемента](#))
- [style](#) ([встроенная информация о стиле information](#))
- [align](#) ([выравнивание](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([внутренние события](#))

Элементы [DIV](#) и [SPAN](#) вместе с атрибутами [id](#) и [class](#) обеспечивают общий механизм добавления в документы структуры. Эти элементы определяют встраиваемую информацию ([SPAN](#)) или информацию уровня блока ([DIV](#)), но не налагают никаких других выражений для представления контекста. Таким образом, авторы могут использовать эти элементы с [таблицами стилей](#), атрибутами [lang](#) и т.д..

Предположим, Вы хотите сгенерировать документ в формате HTML на основе базы данных информации о клиентах. Поскольку HTML не включает элементов для идентификации таких объектов как "клиент", "номер телефона", "адрес электронной почты" и т.д., мы используем элементы [DIV](#) и [SPAN](#) для достижения нужных эффектов структуры и представления. Для структурирования информации мы могли использовать элемент [TABLE](#) следующим образом:

```
<!-- Пример данных из базы данных о клиентах: -->
```

```

<!-- Имя: Stephane Boyera, Тел.: (212) 555-1212, Email: sb@foo.org -->

<DIV id="client-boyera" class="client">
<P><SPAN class="client-title">Информация о клиенте:</SPAN>
<TABLE class="client-data">
<TR><TH>Фамилия:<TD>Boyera</TR>
<TR><TH>Имя:<TD>Stephane</TR>
<TR><TH>Тел:<TD>(212) 555-1212</TR>
<TR><TH>Email:<TD>sb@foo.org</TR>
</TABLE>
</DIV>

<DIV id="client-lafon" class="client">
<P><SPAN class="client-title">Информация о клиенте:</SPAN>
<TABLE class="client-data">
<TR><TH>Фамилия:<TD>Lafon</TR>
<TR><TH>Имя:<TD>Yves</TR>
<TR><TH>Тел:<TD>(617) 555-1212</TR>
<TR><TH>Email:<TD>yves@coucou.com</TR>
</TABLE>
</DIV>

```

Затем мы легко сможем добавить объявление таблицы стиля для настройки представления этих записей.

Другой пример см. в разделе об атрибутах [class и id](#).

Визуальные агенты пользователей обычно помещают символ перевода строки перед и после элементов [DIV](#), например:

```

<P>aaaaaaaa<DIV>бббббббббб</DIV><DIV>ввввв<P>ввввв</DIV>

```

что обычно генерируется как:

```

aaaaaaaaaa
бббббббббб
ввввв

ввввв

```

7.5.5 Заголовки: Элементы h1, h2, h3, h4, h5, h6

```

<!ENTITY % heading "h1|h2|h3|h4|h5|h6">
<!--
  Существует шесть уровней заголовков - с h1 (наиболее важный) до h6 (наименее важный) .
-->

<!ELEMENT (%heading;) - - (%inline;)* -- заголовки -->
<!ATTLIST (%heading;)
  %attrs; -- %coreattrs, %i18n, %events --
>

```

*Начальный тег: **обязателен**, Конечный тег: **обязателен***

Атрибуты, определяемые в любом другом месте

- [id](#), [class](#) (идентификаторы в пределах документа)
- [lang](#) (информация о языке), [dir](#) (направление текста)
- [title](#) (заголовок элемента)
- [style](#) (встроенная информация о стиле)
- [align](#) (выравнивание)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (внутренние события)

Заголовок кратко описывает содержание раздела, которому он предшествует. Информация из заголовка может использоваться агентами пользователей, например, для автоматического построения оглавления документа.

В языке HTML существует шесть уровней заголовков: [H1](#) - наиболее важный - и [H6](#) - наименее важный. Визуальные браузеры обычно отображают более важные заголовки более крупным шрифтом.

В следующем примере показано, как использовать элемент [DIV](#) для того, чтобы связать заголовок с последующим разделом документа. Это позволит Вам определить стиль для раздела (цвет фона, шрифт и т.д.) с использованием таблиц стилей.

```
<DIV class="section" id="forest-elephants" >
<H1>Лесные слоны</H1>
<P>В этом разделе мы обсуждаем менее известных лесных слонов.
...продолжение раздела...
<DIV class="subsection" id="forest-habitat" >
<H2>Ариал</H2>
<P>Лесные слоны не живут в деревьях, а среди них.
...продолжение раздела...
</DIV>
</DIV>
```

Эту структуру можно представить с использованием информации о стиле, например:

```
<HEAD>
<TITLE>... название документа ...</TITLE>
<STYLE type="text/css">
DIV.section { text-align: justify; font-size: 12pt}
DIV.subsection { text-indent: 2em }
H1 { font-style: italic; color: green }
H2 { color: green }
</STYLE>
</HEAD>
```

Пронумерованные разделы и ссылки

Язык HTML не генерирует номера разделов из заголовков. Это может выполняться агентами пользователей. Вскоре языки описания таблиц стилей, такие как CSS, предоставят авторам возможность управления генерацией номеров разделов (для удобства в ссылках в печатной документации, например "См. раздел 7.2").

Некоторые люди считают пропуск уровней заголовков дурным тоном. Они принимают порядок заголовков H1 H2 H1, но не принимают порядок H1 H3 H1, поскольку пропущен уровень H2.

7.5.6 Элемент ADDRESS

```
<!ELEMENT ADDRESS - - (%inline;)* -- информация об авторе -->
<!ATTLIST ADDRESS
  %attrs; -- %coreattrs, %i18n, %events --
  >
```

Начальный тег: **обязателен**, Конечный тег: **обязателен**

Атрибуты, определяемые в любом другом месте

- [id](#), [class](#) (идентификаторы в пределах документ)
- [lang](#) (информация о языке), [dir](#) (направление текста)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (внутренние события)

Элемент [ADDRESS](#) может использоваться авторами для указания контактной информации или основной части документа, такой как форма. Этот элемент часто находится в начале или в конце документа.

Например, страница на сервере W3C, относящаяся к языку HTML, может включать следующую контактную информацию:

```
<ADDRESS>
<P><A href="../People/Raggett/">Dave Raggett</A>,
  <A href="../People/Arnaud/">Arnaud Le Hors</A>,
contact persons for the <A href="Activity">W3C HTML Activity</A><BR>
$Date: 1997/12/16 05:38:14 $
```

</ADDRESS>