3.1 Введение в SGML

SGML - это система определения языков разметки. Авторы *размечают* свои документы, представляя информацию о структуре, представлении и семантике в одном документе. HTML является одним из примеров языка разметки. Вот пример документа на языке 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 на языке HTML</TITLE>
   </HEAD>
   <BODY>
      <P>Всем привет!
   </BODY>
</HTML>
```

Документ HTML состоит из раздела заголовка (здесь - между тэгами <HEAD> и </HEAD>) и тела (здесь - между заголовками <BODY> и </BODY>). Заголовок документа отображается в заголовке (вместе с другой информацией о документе), а содержимое документа находится в теле. В этом примере тело документа состоит только из одного абзаца, помеченного <P>.

Каждый язык разметки, определенный в SGML, называется *приложением SGML*. Приложение SGML характеризуется:

1. Объявлением SGML. *SGML Объявление* указывает, какие символы и разделители могут отображаться в приложении.
2. *Определение типа документа (DTD)* (DTD). DTD определяет синтаксис конструкций разметки. DTD может включать дополнительные определения, например, ссылки на комбинации символов.
3. Спецификация, описывающая семантику, используемую в разметке. Эта спецификация также налагает синтаксические ограничения, которые невозможно выразить при помощи DTD.
4. Экземпляры документа содержат данные (содержимое) и разметку. Каждый экземпляр содержит ссылку на DTD, которое должно использоваться для интерпретации.

Спецификация HTML 4.0 включает объявление SGML , три определения типа документов (описание их см. в разделе информация о версии HTML) и список ссылок на символы.

3.2 Конструкции SGML, используемые в HTML

В следующих разделах описаны конструкции SGML, используемые в HTML.

В приложении перечислены некоторые функции SGML, которые не поддерживаются средствами HTML и агентами пользователей, и использования которых следует избегать.

3.2.1 Элементы

Определение типа документа SGML объявляет *типы элементов*, представляющие структуры или желательное поведение. HTML включает типы элементов, представляющие абзацы, гипертекстовые ссылки, списки, таблицы, изображения и т.д.

Каждое *объявление типа элемента* обычно включает три части: начальный тэг, содержимое и конечный тэг.

Имя элемента отображается в *начальном тэге* (пишется <имя-элемента>) и в *конечном тэге* (пишется </имя-элемента>); не забывайте про слеш перед именем элемента в конечном тэге. Например, начальные и конечные тэги элемента UL определяют список:

```
<UL>
<LI><P>...элемент списка 1...
<LI><P>...элемент списка 2...
</UL>
```

Некоторые типы элементов HTML позволяют авторам опускать конечные тэги (например, типы элементов P and LI). Несколько типов элементов также позволяют опускать началь-

ные тэги; например, HEAD и BODY. HTML DTD указывает для каждого типа элемента, являются ли начальный и конечный тэги обязательными.
Некоторые типы элементов HTML не имеют содержимого. Например, элемент перехода на следующую строку BR не имеет содержимого; его роль - прерывание строки текста. Такие *пустые* элементы никогда не имеют конечных тэгов. Определение типа документа и текст спецификации указывают, является ли тип элемента пустым (не имеет содержимого) или, если он может иметь содержимое, что является допустимым содержимым.
Имена элементов всегда учитывают регистр.
Информацию о правилах, управляющих элементами, (например, что они могут быть вложенными соответствующим образом, конечный тэг закрывает все опущенные начальные тэги вплоть до соответствующего ему начального тэга (раздел 7.5.1) и т.д.) см. в стандарте SGML.
Например, следующий абзац:

```
<P>Это первый абзац.</P>
...элемент блока...
```

можно перезаписать без конечного тэга:

```
<P>Это первый абзац.
...элемент блока...
```

поскольку начальный тэг <P> закрывается следующим элементом блока. Точно так же, если абзац включен в элемент блока, например:

```
<DIV>
<P>Это абзац.
</DIV>
```

конечный тэг включающего элемента блока (здесь - </DIV>) служит также конечным тэгом открытого начального тэга <P>.
Элементы - это не тэги. *Некоторые люди называют элементы тэгами (например, "тэг P "). Помните, что элемент - это одно, а тэг (начала или конца, неважно) - другое. Например, элемент HEAD всегда присутствует, даже если начальный и конечный тэги HEAD отсутствуют.*
Все типы элементов, объявленные в спецификации, перечислены в указателе элементов.

3.2.2 Атрибуты

С элементами могут быть связаны свойства, называемые *атрибутами*, которые могут иметь значения (стандартные или устанавливаемые авторами или сценариями). Пары атрибут/значение помещаются перед закрывающей скобкой ">" начального тэга элемента. В начальном тэге элемента может быть любое число (допустимых) пар атрибут/значение, разделенных пробелами. Они могут указываться в любом порядке.
В данном примере для элемента H1 установлен атрибут id:

```
<H1 id="section1">
Это определенный заголовок, спасибо атрибуту id
</H1>
```

По умолчанию в SGML необходимо, чтобы все значения атрибутов были разделены с помощью двойных (десятичный код ASCII 34) или одинарных кавычек (десятичный код ASCII 39). Одинарные кавычки могут включаться в значение атрибута, если значение отделяется двойными кавычками, и наоборот. Авторы могут также использовать цифровые ссылки на символы для представления двойных (") и одинарных кавычек ('). Для двойных кавычек авторы могут использовать комбинацию ссылок на символы ".
В определенных случаях авторы могут указывать значение атрибута без кавычек. Значение атрибута может включать только буквы (a-z и A-Z), цифры (0-9), знаки переноса (десятичный код ASCII 45) и точки (десятичный код ASCII 46). Рекомендуется всегда использовать кавычки. Имена атрибутов всегда учитывают регистр. Значения атрибутов

обычно учитывают регистр. Определение каждого атрибута в списке атрибутов указывается, учитывает ли значение регистр. Список всех атрибутов, определенных в этой спецификации, приводится в указателе атрибутов.

3.2.3 Ссылки на символы

Ссылки на символы - это числовые или символьные имена символов, которые могут быть включены в документ HTML. Они удобны для обращения к редко используемым символам или к символам, которые трудно или невозможно вводить в средствах разработки документов. Вы увидите ссылки на символы в этом документе; они начинаются со знака "&" и заканчиваются точкой с запятой (;). Вот некоторые примеры:

- "<" представляет знак <.
- ">" представляет знак >.
- """ представляет знак ".
- "å" (десятичное число) представляет букву "a" с кружком сверху.
- "И" (десятичное число) представляет кириллическую букву "I".
- "水" (шестнадцатеричное число) представляет китайский знак воды.

Ссылки на символы в HTML подробно обсуждаются дальше в этом разделе под заголовком набор символов документа HTML. В спецификации также содержится список ссылок на символы, которые могут использоваться в документах в формате HTML 4.0.

3.2.4 Комментарии

Комментарии в HTML имеют следующий синтаксис:

```
<!-- это комментарий -->
<!-- это тоже комментарий,
    он занимает несколько строк -->
```

Проблемы между открывающим разделителем разметки ("<!") и открывающим разделителем комментария ("--") недопустимы, но их можно использовать между закрывающим разделителем комментария ("--") и закрывающим разделителем разметки (">"). Распространенной ошибкой является включение строки символов переноса ("---") в комментарий. Следует избегать использования в комментариях двух или более символов переноса.
Информация в комментариях не имеет специального значения (например, ссылки на символы не интерпретируются).

3.3 Как читать HTML DTD

Каждое объявление элемента и атрибута в этой спецификации сопровождается фрагментом его определения типа документа. Мы решили включить фрагменты DTD в спецификацию вместо того, чтобы искать более доступные, но более длинные и менее точные средства описания свойств элементов. При помощи следующего учебника читатели, не знакомые с SGML, смогут научиться читать DTD и понимать технические подробности спецификации HTML.

3.3.1 Комментарии DTD

В DTD комментарии могут занимать несколько строк. В DTD комментарии отделяются парой меток "--", например,

```
<!ELEMENT PARAM - O EMPTY       -- значение именованного свойства -->
```

Здесь комментарий "значение именованного свойства" объясняет использование типа элемента PARAM. Комментарии в DTD носят информативный характер.

3.3.2 Определения комбинаций параметров

HTML DTD начинается с ряда определений комбинаций параметров. *Определение комбинации параметров* определяет макрос, на который можно ссылаться в любом месте DTD. Эти макросы не отображаются в документах HTML, они отображаются только в DTD. Другие типы макросов, называемые ссылками на символы, могут использоваться в тексте документа в формате HTML или в значениях атрибутов. Когда на комбинацию параметров ссылаются в DTD по имени, она разворачивается в строку.
Определение комбинации параметров начинается с ключевого слова <!ENTITY %, за кото-

рым следует имя entity, строка в кавычках, в которую разворачивается entity и наконец, закрывающий >. Экземпляры комбинаций параметров в DTD начинаются со знака "%", затем идет имя комбинации и заканчивается необязательным символом ";".
В следующем примере определяется, в какую строку будет разворачиваться "%fontstyle;".

```
<!ENTITY % fontstyle "TT | I | B | BIG | SMALL">
```

Строка, в которую разворачивается комбинация параметров, может содержать другие имена комбинаций параметров. Эти имена разворачиваются рекурсивно. В следующем примере "%inline;" комбинация параметров включает комбинации "%fontstyle;", "%phrase;", "%special;" и "%formctrl;".

```
<!ENTITY % inline "#PCDATA | %fontstyle; | %phrase; | %special; | %formctrl;">
```

Вы часто будете встречать в HTML DTD два DTD entities: "%block;" "%inline;". Они используются, если модель содержимого включает элементы уровня блока и встроенные элементы соответственно (определены в разделе глобальная структура документа HTML).

3.3.3 Объявления элементов

HTML DTD состоит из объявлений *типов элементов* и их атрибутов. Объявление начинается с ключевого слова <!ELEMENT и заканчивается символом >. Внутри указываются:

1. Имя элемента.
2. Обязателен ли конечный тэг элемента. Два символа переноса после имени элемента означают, что начальный и конечный тэги являются обязательными. Один символ переноса, за которым следует буква "O", указывает, что конечный тэг можно опустить. Две буквы "O" указывают на то, что можно опустить как начальный, так и конечный тэги.
3. Содержимое элемента, если таковое имеется. Допустимым содержимым элемента называется его *модель содержимого*. Типы элементов, не имеющие содержимого, называются *пустыми элементами*. Модель содержимого для таких типов элементов объявляется при помощи ключевого слова "EMPTY".

В этом примере:

```
    <!ELEMENT UL - - (LI)+>
```

- Объявляется тип элемента **UL**.
- Два знака переноса указывают, что начальный тэг <UL> и конечный тэг </UL> для этого элемента обязательны.
- Модель содержимого для этого типа элемента - "по крайней мере, один элемент LI". Ниже объясняется, как задать модель содержимого.

В этом примере показано объявление пустого типа элемента:

```
    <!ELEMENT IMG - O EMPTY>
```

- Объявляется тип элемента **IMG**.
- Знак '-', за которым следует буква "O", указывает, что конечный тэг можно опустить, но если модель содержимого "EMPTY", это правило усиливается, и конечный тэг **должен** быть опущен.
- Ключевое слово "EMPTY" означает, что экземпляры этого типа не должны иметь содержимого.

Определения модели содержимого

Модель содержимого описывает, что может содержаться в экземпляре типа элемента. Определения модели содержимого могут включать:

- Имена допустимых или запрещенных типов элементов (например, элемент UL содержит экземпляры типа элемента LI, а тип элемента P не может включать другие элементы P).
- Комбинации DTD (например, элемент LABEL включают экземпляры комбинации параметров "%inline;").
- Текст документа (указываемый SGML-конструкцией "#PCDATA"). Текст может

включать ссылки на символы. Помните, что они начинаются с & и заканчиваются точкой с запятой (например, "Hergé's adventures of Tintin" содержит ссылку на комбинацию символов для отображения символа "e со знаком ударения").
Модель содержимого элемента указывается с использованием следующего синтаксиса:
(...)
Разделяет группы.
A | B
Происходит A или B, но не оба.
A , B
Происходят A и B в указанном порядке.
A & B
Происходят A и B в любом порядке.
A?
A происходит ноль или один раз.
A*
A происходит ноль или более раз.
A+
A происходит один или несколько раз.
Вот некоторые примеры HTML DTD:

```
<!ELEMENT UL - - (LI)+>
```

Элемент UL должен содержать один или несколько элементов LI.

```
<!ELEMENT DL    - - (DT|DD)+>
```

Элемент DL должен содержать один или несколько элементов DT или DD в любом порядке.

```
<!ELEMENT OPTION - O (#PCDATA)>
```

Элемент OPTION может содержать только текст и такие entities как & -- это определяется типом данных SGML #PCDATA.
Некоторые типы элементов HTML используют дополнительную функцию SGML для исключения элементов из модели содержимого. Исключенным элементам предшествует символ переноса. Явные исключения имеют приоритет по отношению к допустимым элементам.
В этом примере -(A) означает, что элемент A не может находиться в другом элементе A (то есть ссылки не могут быть вложенными).

```
<!ELEMENT A - - (%inline;)* -(A)>
```

Помните, что тип элемента A является частью DTD комбинации параметров "%inline;", но явно исключается выражением -(A).
Точно так же следующее объявление типа элемента FORM запрещает вложенные формы:

```
<!ELEMENT FORM - - (%block;|SCRIPT)+ -(FORM)>
```

3.3.4 Объявления атрибутов

Объявление атрибутов, которые может иметь элемент, начинается с ключевого слова <!ATTLIST. За ним следует имя элемента с вопросительным знаком, список определений атрибутов и закрывающий символ >. Каждое определение атрибута - это тройка, определяющая:

- Имя атрибута.
- Тип значения атрибута или явный набор допустимых значений. Значения, определенные явным образом при помощи DTD, учитывают регистр. Подробнее о типах значений атрибутов см. в разделе основные типы данных HTML.
- Является ли значение атрибута по умолчанию неявным (ключевое слово

"#IMPLIED"), то есть значение по умолчанию устанавливается агентом пользователя (в некоторых случаях с использованием наследования от родительских элементов); всегда обязательным (ключевое слово "#REQUIRED"); или фиксированным данным значением (ключевое слово "#FIXED"). Некоторые определения атрибутов явным образом указывают значение атрибута по умолчанию.
В этом примере атрибут имя определен для элемента MAP. Атрибут не является обязательным для этого элемента.

```
<!ATTLIST MAP
  name        CDATA     #IMPLIED
  >
```

Тип значений, допустимых для этого атрибута, дается как CDATA, тип данных SGML. CDATA - это текст, который может содержать ссылки на символы.
Подробнее о типах данных "CDATA", "NAME", "ID" и др. см. в разделе типы данных HTML.
В следующих примерах показано несколько определений атрибутов:

```
rowspan     NUMBER     1          -- число строк, охватываемых ячейкой --
http-equiv  NAME       #IMPLIED   -- Имя заголовка ответа HTTP --
id          ID         #IMPLIED   -- уникальный ИД в пределах документа --
valign      (top|middle|bottom|baseline) #IMPLIED
```

Для атрибута `rowspan` необходимы значения типа NUMBER. Значение по умолчанию дается явно - "1". Для необязательного атрибута `http-equiv` необходимы значения типа NAME. Для необязательного атрибута `id` необходимы значения типа ID. Необязательный атрибут `valign` ограничен значениями из набора {top, middle, bottom, baseline}.

Комбинации DTD в определениях атрибутов

Определения атрибутов могут также содержать ссылки на комбинации параметров.
В этом примере мы видим, что список определений атрибутов для элемента LINK начинается с комбинации параметров "%attrs;".

```
<!ELEMENT LINK - O EMPTY                -- ссылка, независимая от устройства -->
<!ATTLIST LINK
  %attrs;                               -- %coreattrs, %i18n, %events --
  charset     %Charset;       #IMPLIED  -- кодировка связанного документа --
  href        %URI;           #IMPLIED  -- URI для связанного документа --
  hreflang    %LanguageCode;  #IMPLIED  -- код языка --
  type%ContentType;   #IMPLIED  -- рекомендуемый тип содержимого --
  rel%LinkTypes;     #IMPLIED  -- типы ссылок для перехода вперед --
  rev%LinkTypes;     #IMPLIED  -- типы ссылок для перехода назад --
  media%MediaDesc;     #IMPLIED  -- для генерации на этом устройстве --
  >
```

Начальный тег: ***обязателен****, Конечный тег:* ***запрещен***
Комбинация параметров "%attrs;" определена следующим образом:

```
<!ENTITY % attrs "%coreattrs; %i18n; %events;">
```

Комбинация "%coreattrs;" в определении "%attrs;" разворачивается следующим образом:

```
<!ENTITY % coreattrs
 "idID              #IMPLIED  -- уникальный ИД в пределах документа --
  classCDATA           #IMPLIED  -- список классов, разделенных пробелами --
  style%StyleSheet;    #IMPLIED  -- информация о стиле --
  title%Text;          #IMPLIED  -- рекомендуемый заголовок/распространение --"
  >
```

Комбинация параметров "%attrs;" определена для удобства, поскольку эти атрибуты определены для большинства типов элементов HTML. Таким же образом DTD определяет комбинацию параметров "%URI;" как расширение строки "CDATA".

```
<!ENTITY % URI "CDATA"
    -- Универсальный идентификатор ресурсов, см. [URI]
```

```
    -->
```

Как показано в этом примере, комбинация параметров "%URI;" предоставляет читателям DTD больше информации, чем для типа данных, ожидаемого для этого атрибута. Похожие entities определены для "%Color;", "%Charset;", "%Length;", "%Pixels;" и т.д.

Логические атрибуты

Некоторые атрибуты играют роль логических переменных (например, атрибут `selected` для элемента `OPTION`). Их наличие в начальном тэге элемента подразумевает, что значением атрибута является "истина". Их отсутствие означает "ложь".

Логические атрибуты могут принимать только одно значение: собственно имя атрибута (например, `selected="selected"`).

В этом примере атрибут `selected` определяется как булев.

```
selected     (selected)  #IMPLIED  -- уменьшенный интервал между элементами --
```

Для атрибута устанавливается значение "истина", поскольку он находится в начальном тэге элемента:

```
<OPTION selected="selected">
...contents...
<OPTION>
```

В HTML логические атрибуты могут быть в *минимизированной форме* -- в начальном тэге элемента находится только **значение** атрибута. Таким образом, `selected` можно установить, написав:

```
<OPTION selected>
```

вместо:

```
<OPTION selected="selected">
```

Авторам следует знать, что многие агенты пользователей распознают **только** минимизированную форму логических атрибутов и не распознают полную.