

Гипермедийные среды и технологии

Лабораторная работа 1

Введение в SGML, гипертекстовые возможности офисных редакторов

Аннотация

Целью данной работы является практическое знакомство с двумя основными реализациями стандарта SGML: HTML и XML. В ходе работы студент приобретет (восполнит ранее утраченные) знания, позволяющие при помощи базовых средств создать документы представленных форматов, сформировать локальные гиперссылки на необходимые объекты внутри документов, а также усвоить практические способы создания документов формата OOXML, в которых необходимо применение стилей и гиперссылок.

Для выполнения лабораторной работы необходимо:

- рабочая станция с установленной ОС Windows XP и более новой (возможен вариант использования терминального доступа),
- офисный пакет для создания и обработки файлов формата OOXML: Microsoft Office 2007 SP1 и более новой (возможен вариант использования открытых свободных решений, таких как OpenOffice, LibreOffice и иных),
- веб-браузер,
- текстовый редактор с возможностью изменения кодировки файла.

Терминология

- SGML - Standard Generalized Markup Language, метаязык, на котором можно определять язык разметки для документов, стандартизован ISO 8879:1986.
- XML - eXtensible Markup Language, расширяемый язык разметки. язык с простым формальным синтаксисом, удобный для создания и обработки документов программам и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете.
- HTML - HyperText Markup Language, язык гипертекстовой разметки, стандартный язык разметки документов во Всемирной паутине.
- OOXML - Office Open XML - серия форматов файлов для хранения электронных документов пакетов офисных приложений — в частности, Microsoft Office.
- Гиперссылка - часть гипертекстового документа, ссылающаяся на другой элемент в самом документе, на другой объект, расположенный на локальном диске или в компьютерной сети, либо на элементы этого объекта.

Теория

SGML

GML (Generalized Markup Language) разработан в недрах корпорации IBM. Его наследник SGML (Standard Generalized Markup Language) принят в 1986 году в качестве международного стандарта для определения независимых от устройств ввода/вывода,

независимых от вычислительной среды методов представления текстов в электронной форме. Более точно, SGML - это метаязык, то есть средство формального описания языка, в данном случае, языка разметки.

Исторически слово разметка использовалось для описаний аннотаций или других обозначений внутри текста, предназначенных для указаний составителю или "верстальщику" того, как именно конкретное место должно быть напечатано или сверстано. Примеры включают подчеркивание волнистой чертой, обозначающее курсив, специальные значки для пропуска фраз или их печати конкретным шрифтом, и так далее. Когда форматирование и печать текстов стали автоматизированными, этот термин стал охватывать все виды специальных кодов разметки, вставляемых в электронные тексты для управления форматированием, печатью или другой обработкой.

Обобщая, разметку, или кодировку, определяют, как любое средство сделать явным интерпретацию текста. На банальном уровне все напечатанные тексты кодированы в этом смысле: знаки препинания, использование заглавных букв, расположение букв по странице, даже интервалы между словами можно считать в какой-то степени разметкой, функция которой - помочь человеку, читающему текст, определить, где кончается одно слово и начинается другое, или как идентифицировать особенности структуры, такие как заголовки, или простые синтаксические единицы вроде подчиненных предложений. Кодирование текста для компьютерной обработки, в принципе, как расшифровка манускрипта с пергамента, - процесс производства явным неявного или подразумеваемого, процесс указания пользователю того, как должно интерпретироваться содержимое текста.

Под языком разметки понимают набор соглашений о разметке, применяемых для кодирования текстов. Язык разметки должен специфицировать, какая разметка допустима, какая разметка обязательна, как отличить разметку от текста и что разметка значит. SGML предоставляет решения для первых трех задач, отдельная документация обычно необходима для последней.

Структура HTML

Язык HTML имеет довольно большое количество тегов – языковых элементов, при помощи которых форматируется текст, изображения и иная мультимедийная информация. Рассмотрим некоторые элементы языка.

Элемент `<!DOCTYPE>` предназначен для указания типа текущего документа — DTD (document type definition, описание типа документа). Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, поскольку HTML существует в нескольких версиях, кроме того, имеется XHTML (EXtensible HyperText Markup Language, расширенный язык разметки гипертекста), похожий на HTML, но различающийся с ним по синтаксису. Чтобы браузер «не путался» и понимал, согласно какому стандарту отображать веб-страницу и необходимо в первой строке кода задавать `<!DOCTYPE>`.

Существует несколько видов `<!DOCTYPE>`, они различаются в зависимости от версии языка, на которого ориентированы. В таблице приведены основные типы документов с их описанием.

DOCTYPE	Описание
HTML 4.01	
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"></code>	Строгий синтаксис HTML.
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>	Переходный синтаксис HTML.
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"></code>	В HTML-документе применяются фреймы.

Тег <html> является контейнером, который заключает в себе все содержимое веб-страницы, включая теги <head> и <body>. Открывающий и закрывающий теги <html> в документе необязательны, но хороший стиль диктует непременно их использование. Как правило, тег <html> идет в документе вторым, после определения типа документа (Document Type Definition, DTD), устанавливаемого через элемент <!DOCTYPE>. Закрывающий тег <html> должен всегда стоять в документе последним.

Тег <head> предназначен для хранения других элементов, цель которых — помочь браузеру в работе с данными. Также внутри контейнера <head> находятся метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных.

Содержимое тега <head> не отображается напрямую на веб-странице, за исключением тега <title> устанавливающего заголовок окна веб-страницы.

Тег <title> Определяет заголовок документа. Элемент <title> не является частью документа и не показывается напрямую на веб-странице. Допускается использовать только один тег <title> на документ и размещать его в контейнере <head>.

HTML предлагает шесть заголовков разного уровня, которые показывают относительную важность секции, расположенной после заголовка. Так, тег <h1> представляет собой наиболее важный заголовок первого уровня, а тег <h6> служит для обозначения заголовка шестого уровня и является наименее значительным. По умолчанию, заголовок первого уровня отображается самым крупным шрифтом жирного начертания, заголовки последующего уровня по размеру меньше. Теги <h1>, ..., <h6> относятся к блочным элементам, они всегда начинаются с новой строки, а после них другие элементы отображаются на следующей строке. Кроме того, перед заголовком и после него добавляется пустое пространство.

Определяет текстовый абзац. Тег <p> является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой отбивкой. Величиной отбивки можно управлять с помощью стилей. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

**Тег ** Устанавливает жирное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

Тег <i> Устанавливает курсивное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

Тег <meta> определяет метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных. Разрешается использовать более чем один метатег, все они размещаются в контейнере <head>. Как правило, атрибуты любого метатега сводятся к парам «имя=значение», которые определяются ключевыми словами content, name или http-equiv.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

**Тег
** устанавливает перевод строки в том месте, где этот тег находится. В отличие от тега абзаца <p>, использование тега
 не добавляет пустой отступ перед строкой.

Тег <a> является одним из важных элементов HTML с точки зрения гипермедиа и предназначен для создания ссылок. В зависимости от присутствия атрибутов name или href тег <a> устанавливает ссылку или якорь. Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы. При использовании атрибута href наименование локальной ссылки указывается с символом #.

**Тег ** определяет отдельный элемент списка. Внешний тег или устанавливает тип списка — маркированный или нумерованный.

XML

XML (Extensible Markup Language) – это, относительно, новый SGML-производный язык разметки документов, позволяющий структурировать информацию разного типа, используя для этого произвольный набор инструкций.

Простейший XML-документ состоит из двух частей: пролога и корневого элемента. Пролог содержит объявление XML, указывающее на то, что это XML-документ, и содержит номер версии XML. Пролог может также содержать необязательные компоненты:

Объявление типа документа.

Одну или несколько инструкций по обработке.

XML-документ может содержать комментарии, начинающиеся с символов "<!--" и заканчивающиеся символами "-->". Комментарий может содержать любой текст, за исключением символов "--". Тексты комментариев доступны для написанного внутри HTML-страницы кода сценария.

XML-документ можно набрать в любом текстовом редакторе, сохранив документ как текстовый файл с расширением .xml. В дальнейшем такой документ будет открываться двойным щелчком в Internet Explorer.

Чтобы иметь возможность использовать в XML-документе символы кириллицы (как в символьных данных, так и в именах элементов), вы можете указать кодировку в прологе:

<?xml version="1.0" encoding="windows-1251"?>

OOXML

OOXML или Office Open XML – это стандарт, описывающий структуру файла в формате DOCX, XLSX и пр. то есть наиболее используемых офисных форматов документов. Файлы такого формата можно создавать при помощи таких офисных пакетов как Microsoft (MS) Office, Open Office, LibreOffice и пр. Фактически данные файлы представляют собой архив из множества XML с определенными DTD и мультимедиа файлов, которые располагаются внутри документа. Приложения для работы с файлами позволяют создавать сложные гипермедийные документы, с изображениями, векторными схемами, а также ссылками. Как правило, интерфейс данных приложений является интуитивно понятным и позволяет оперировать элементами при помощи контекстного меню, а также панелей (вкладок). В частности, для создания ссылки необходимо выделить текст, который в последствие станет ссылкой, нажать правой кнопкой мыши и выбрать пункт «Гиперссылка».

Особым способом форматирования документов в текстовом процессоре, работающем с файлом формата OOXML является применение стилей. Стил – это набор правил для форматирования текста. Стили можно свободно создавать и редактировать (MS Word: Вкладка «Главная», раздел «Стили»), таким образом генерируя целую серию правил для форматирования текста определенного назначения. Также существуют специальные стили для создания заголовков, которые в последствии могут стать элементами автособираемого оглавления (MS Word: вкладка «Ссылки», пункт «Автособираемое оглавление»), то есть ссылками в привычном понимании.

Задание 1

При помощи текстового редактора, сформировать правильный (отображаемый в браузере) HTML документ, состоящий из заголовка, оглавления и списка с 1 уровнем вложенности на тему:

- a. 5 животных на выбор, их описание + список, чем они питаются.
- b. 5 языков программирования, их описание + список на какие языки данный язык повлиял.
- c. 5 типов данных, их описание + список к каким другим типам можно явно привести
- d. 5 радиоактивных изотопов, их описание + список на какие хм. Элементы они распадаются.

Оглавление должно быть в виде гиперссылок на элементы списка, которые в свою очередь должны являться подзаголовками. В списке второго уровня должно быть, как минимум, 2 ссылки на элементы списка первого уровня.

Пример 1

Пример представления

Животные

[Лев](#)

[Зебра](#)

...

- **Лев** – животное из семейства больших кошек. Обитает на южном полюсе, иногда заплывает на Камчатку.
 - Пингвины
 - Трюфели
 - [Зебры](#)
- **Зебра** – бескилевая нелетающая птица. Обитает в водах бермудского треугольника.
 - Кашалоты
 - [Львы](#)
 - Дятлы

Задание 2

Сформировать XML документ, представляющий структуру из задания 1, без оглавления. Наименование и описание элемента должны являться свойствами, сами элементы должны иметь идентификаторы, ссылки на элементы первого уровня должны быть свойствами.

Пример 2

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE animals>
<animals>
  <animal name="Лев" id="1" desc="животное из семейства больших кошек.
Обитает на южном полюсе, иногда заплывает на камчатку.">
    <food>Пингвины</food>
    <food>Трюфели</food>
    <food id="2">Зебры</food>
  </animal>
  <animal name="Зебра" id="2" desc="бескилевая нелетающая птица. Обитает
в водах бермудского треугольника.">
    <food>Кашалоты</food>
    <food id="1">Львы</food>
    <food>Дятлы</food>
  </animal>
</animals>
```

Задание 3

1. Сформировать документ в формате OOXML, представляющий структуру из задания 1. Должно быть произведено ручное форматирование и должно присутствовать оглавление из вручную сформированных гиперссылок.
2. Показать.
3. Произвести стилизацию документа, наименования элементов должны быть заголовками 2 уровня. Вставить автособираемое оглавление.