

# **Описание формата лекций INTUIT.ru**

Версия 1.30  
(последнее изменения: 18 мая 2012г.)

## Содержание

Рабочая среда.....	3
Предварительный просмотр лекций.....	3
Общая информация.....	3
Структура файла lecture.xml.....	4
Структура лекции. Главы.....	5
Абзацы.....	5
Листинги.....	6
Списки.....	6
Цитаты.....	7
Вставки.....	7
Таблицы.....	7
Разметка текстовых областей.....	8
Верхние и нижние индексы, акценты, конструкции языка программирования.....	8
Ссылки.....	10
Ключевые слова.....	12
Сноски.....	13
Изображения.....	13
Страницы.....	14
Вопросы/задания в тексте лекции.....	14
Структура файла теста.....	15
Специальные символы (сущности).....	17
Оформления списка литературы.....	17

## Принцип редактирования XML-документов

Для редактирования лекций можно использовать любой специализированный XML-редактор, способный проверять соответствие документа DTD (проводить валидацию), например [Bonfire Studio](#).

Необходимость использования специализированного редактора вызвана жесткими требованиями к отсутствию в нем синтаксических ошибок (например, опечаток в названии тегов, или ошибках в их вложенности), а также соответствию документа представлениям программистов. Для выполнения данной проверки требуется проводить валидацию документа после окончания верстки. После нажатия F8 ([Bonfire Studio](#)) редактор либо сообщит, что документ успешно прошел проверку, либо укажет на ошибку. Чтобы снизить вероятность появления ошибок, необходимо действовать строго в соответствии со спецификацией; при появлении сомнений в правильности выбранного решения рекомендуется сразу после попытки его применить выполнить валидацию документа. При возникновении необходимости в добавлении новых возможностей верстки, либо при невозможности составить валидный документ без потери информации, необходимо связаться по адресу <[webmaster@intuit.ru](mailto:webmaster@intuit.ru)>.

В теле документа возможно использование специальных символов, определенных для формата xhtml 1.0. Списки обозначений символов доступны здесь:

[http://www.w3.org/TR/xhtml1/dtds.html#a\\_dtd\\_Special\\_characters](http://www.w3.org/TR/xhtml1/dtds.html#a_dtd_Special_characters)

[http://www.w3.org/TR/xhtml1/dtds.html#a\\_dtd\\_Symbols](http://www.w3.org/TR/xhtml1/dtds.html#a_dtd_Symbols)

[http://www.w3.org/TR/xhtml1/dtds.html#a\\_dtd\\_Latin-1\\_characters](http://www.w3.org/TR/xhtml1/dtds.html#a_dtd_Latin-1_characters)

## Рабочая среда

Для работы рекомендуется создать на жестком диске компьютера папку (например, C:\XML), в которую должны быть распакованы файлы [рабочей среды](#), необходимые для валидации (проверки на правильность) и предварительного просмотра результатов верстки. При получении задания на верстку лекций из очередного курса, создайте в данной папке папку с названием курса, а внутри нее – папки для каждой обрабатываемой лекции (по ее номеру), а также папка для тестов ко всему курсу, например, «!Тесты». Например, путь к папке лекции может выглядеть так: C:\XML\Основы PHP\1\ . В ней должен быть файл lecture.xml с текстом лекции и файлы картинок, встречающихся в тексте лекции. Пример пути к папке с тестами для конкретного курса – C:\XML\Основы PHP\!Тесты . Тесты к каждой лекции необходимо разложить в отдельные файлы с именем, соответствующим номеру лекции. Например, для тестов к первой лекции это будет файл C:\XML\Основы PHP\!Тесты\1.xml .

При такой структуре директорий все программы найдут файлы, необходимые им для валидации любого файла и его предварительного просмотра.

## Предварительный просмотр лекций

Для просмотра сверстанного материала требуется открыть xml-файл с лекцией в окне браузера Internet Explorer либо закладке «browse» [Bonfire Studio](#). При этом будут подключены xslt-стили, входящие в состав рабочей среды и обеспечивающие отображение информации. При предварительном просмотре возможно незначительное расхождение с видом лекции на сайте.

## Общая информация

Каждая обработанная лекция должна быть предоставлена в виде одной директории, имя которой совпадает с номером лекции. Данная директория содержит следующие созданные верстальщиком файлы:

- ✱ lecture.xml

- ✱ все файлы изображений, встречающихся в лекции (в форматах GIF, JPEG, PNG)

☞ прочие файлы, на которые есть ссылки.

Обработанные материалы и вопросы отправляются по электронной почте на адрес [job@intuit.ru](mailto:job@intuit.ru).

Важно!

1. Оформляйте конструкции языка программирования, переменные, формулы внутри строк текста, цифровые выражения с помощью тега code.

1) выделение ключевых слов.

2. Кавычки должны быть заменены на &quot;;

3. Перед сдачей работы обязательно проводите валидацию набранных XML-документов. Для этого можно использовать один из XML-редакторов.

## Структура файла lecture.xml

В ходе работы над лекцией необходимо решить следующие задачи:

2) структурирование лекции: разбиение на главы и параграфы; выделение списков, таблиц, листингов, цитат, сносок и рисунков;

3) оформление текста: расстановка внутри параграфов верхних и нижних индексов, акцентов, ссылок, переменных и операторов (внутри параграфов);

В начале лекции может быть эпиграф, который оформляется как *цитата*.

Файл должен соответствовать следующему шаблону:

```
<?xml version="1.0" encoding="windows-1251" standalone="no"?>
<!DOCTYPE lecture SYSTEM "../..//lecture.dtd">
<?xml-stylesheet type="text/xsl"
                    href="../..//lecture_preview.xsl" ?>
<lecture>
<title>Название лекции</title>
<annotation>Описание лекции</annotation>

<section>
  <title>заголовок</title>
  ... тело лекции ...
</section>

<tasks />
</lecture>
```

В теге **lecture** возможен не обязательный атрибут **lecture\_name**, при показе лекции, вместо слова 'Лекция' будет показываться значение этого атрибута. Пример:

```
<?xml version="1.0" encoding="windows-1251" standalone="no"?>
<!DOCTYPE lecture SYSTEM "../..//lecture.dtd">
<?xml-stylesheet type="text/xsl"
                    href="../..//lecture_preview.xsl" ?>
<lecture lecture_name="Задание">
<title>Название задания</title>

...

</lecture>
```

Шаблон лекции можно найти также в файле “lecture-default.xml”. Файл состоит из нескольких частей:

- 1) информационная часть (название лекции, аннотация) – заполняется по шаблону;
- 2) текст лекции (находится внутри тега <section>).

Верстка лекций служит исключительно для создания логической структуры лекции. Запрещается использование тегов логической разметки для создания визуальных эффектов, расходящихся с назначением тега (например, использование тега <em> при желании отобразить не являющийся *акцентом* текст жирным шрифтом). Это не имеет смысла, т.к. **отображение тегов может быть изменено программистами в любой момент** (например, отображение текста внутри тегов <em> может измениться на курсивное). Но если очень нужно, можно пользоваться конструкцией <em type="italic">ваш наклонный текст</em>

## Структура лекции. Главы

Лекция состоит из глав. Каждая глава заключается внутрь тега <section>; название главы помечается тегом <title>, который должен следовать сразу после открывающего тега <section>. Главы могут содержать в произвольном порядке:

- другие главы (подразделы);
- абзацы;
- листинги;
- списки;
- цитаты;
- таблицы.

Максимальный уровень вложенности глав – 5. Заголовки глав-подразделов будут выводиться шрифтом уменьшенного размера.

Пример:

```
<section>
  <title>Программирование</title>
  <section>
    <title>Программирование для Unix</title>
    ...
  </section>
  <section>
    <title>Программирование для Web</title>
    <section>
      <title>Perl</title>
      <p>текст о Perl – параграф 1</p>
      <p>текст о Perl – параграф 2</p>
      ...
    </section>
    ...
  </section>
  ...
</section>
```

## Абзацы

Текст каждого абзаца заключается в теги <p>:

```
<p>текст о Perl – параграф 1</p>
```

## Листинги

Листингом называется блок программного кода на каком-либо языке программирования или разметки, формула, математическое выражение и т. д. Листинг может быть многострочный, либо находящийся на отдельной строке. Фрагменты кода или формул, находящиеся внутри текста (например, внутри абзаца), листингом не считаются (см. ниже). Для выделения листинга существует тег `<example>`:

```
<example [number="1.1"] [title="название"] [visible="yes|no]
        [type="example|listing"]>
...
</example>
```

Если задан атрибут `number`, на данный листинг будут разрешены ссылки из других лекций курса. Номер отображается сразу после листинга. В зависимости от значения атрибута `type`, листинг помечается словом «Листинг» (`type="listing"`) или «Пример» (`type="example"`). Если задан атрибут `title`, его значение отображается там же. Текст листинга записывается «как есть», т.е. в теле листинга учитываются все пробелы и переводы строк.

Пример:

```
<example number="1.1" title="Простейшая программа на C"
        type=" listing">

int main()
{
    printf("Hello, world!\n");
}
</example>
```

Данный текст может быть отображен в виде:

```
int main()
{
    printf("Hello, world!\n");
}
```

### **Листинг 1.1. Простейшая программа на C**

Атрибут `visible` отвечает за отображение листинга на странице лекции. Если `visible="no"`, листинг на странице не отображается, выводится только его название и ссылка, позволяющая открыть текст листинга в отдельном окне.

## Списки

Для обозначения упорядоченных и неупорядоченных списков используется контейнер `list`:

```
<list [type="none|number"]>
<li>...</li>
<li>...</li>
...
</list>
```

Внутри контейнера находятся только теги `<li>...</li>`, в которые заключается для каждая позиция списка.

В зависимости от значения атрибута `type` список является упорядоченным (`type="number"`) или неупорядоченным (`type="none"`, необязательно). Позиции упорядоченных списков нумеруются, в неупорядоченных списках каждая позиция помечается пулкой (точкой).

Пример 1:

```
<list type="number">
<li>perl</li>
<li>java</li>
</list>
```

Данный текст может быть отображен в виде:

```
1. perl
2. java
```

Пример 2:

```
<list>
<li>perl</li>
<li>java</li>
</list>
```

Данный текст может быть отображен в виде:

```
└─ perl
└─ java
```

## Цитаты

Блок текста отмечается как цитата при помощи тега `<quote>`. Если цитата содержит несколько абзацев текста, все абзацы выделяются в 1 тег `<quote>`. Примеры:

```
<quote>Было бы ошибкой думать...</quote>
```

```
<quote>
<p>Первый абзац цитаты</p>
<p>Второй абзац цитаты</p>
</quote>
```

Цитаты также можно использовать для отображения эпиграфа.

```
<quote author="Стендаль, &quot;Жизнь Анри Брюлара&quot;;">
Мсье Дюпки [учитель математики] произносил напыщенные фразы об
этом предмете, но ни разу не сказал этих простых слов: это
разделение труда, которое, как и всякое другое разделение труда
производит чудеса и позволяет уму сконцентрировать все свои силы
только на одной стороне объектов, только на одном из их качеств.
</quote>
```

## Вставки

Блок текста отмечается как *вставка* при помощи тега `<inset>`. У вставки может быть необязательный атрибут **icon** (предполагается картинка в левом верхнем углу вставки для акцентирования внимания). Значение атрибута **icon** – урл картинки. Примеры:

```
<inset>Было бы ошибкой думать...</inset>
```

```
<inset icon="impotant.gif">
<p>Первый абзац вставки</p>
<p>Второй абзац вставки</p>
</quote>
```

## Таблицы

Как и в HTML, таблица заключается в тег `<table>`:

```
<table [number="1.1"]>
[<caption>Название таблицы</caption>]
<tr>
<th [colspan="1"] [rowspan="1"]>Шапка таблицы</th>
<th [colspan="1"] [rowspan="1"]>Шапка таблицы</th>
</tr>
<tr>
<td [colspan="1"] [rowspan="1"]>Содержимое ячеек</td>
<td [colspan="1"] [rowspan="1"]>Содержимое ячеек</td>
</tr>
```

&lt;/table&gt;

Если задан атрибут `number`, на таблицу можно будет ссылаться по этому номеру из других лекций. Кроме того, рядом с названием таблицы будет отображен ее номер. Название таблицы размещается в необязательном теге `<caption>`.

Каждая строка таблицы размещается между тегами `<tr>`. Ячейки таблицы представлены тегами `<th>` (ячейка помечается как содержащая заголовок колонки/строки) и `<td>` (обычная ячейка). Ячейка может быть растянута на несколько ячеек по вертикали (при помощи атрибута `rowspan`) или по горизонтали (атрибут `colspan`).

Пример:

```
<table number="1.1">
  <caption>Тестовая таблица</caption>
  <tr>
    <th>Заголовок 1</th>
    <th>Заголовок 2</th>
    <th>Заголовок 3</th>
  </tr>
  <tr>
    <td rowspan="2">высокая ячейка</td>
    <td colspan="2">широкая ячейка</td>
  </tr>
  <tr>
    <td>просто ячейка 1</td>
    <td>просто ячейка 2</td>
  </tr>
</table>
```

Данный пример может быть отображен следующим образом:

1.1. Тестовая таблица		
Заголовок 1	Заголовок 2	Заголовок 3
высокая ячейка	широкая ячейка	
	просто ячейка 1	просто ячейка 2

## Разметка текстовых областей

Любой блок текста (параграф, элемент списка, цитата, ячейка таблицы) может содержать следующие строчные элементы (т.е. элементы, которые могут располагаться внутри строки текста, не разрывая ее):

- верхние и нижние индексы;
- верхнее подчеркивание (математическое обозначение «вектора»);
- акценты (выделенные особо слова и фразы);
- конструкции языка программирования или разметки (например, имена переменных или функций);
- ссылки на веб-ресурсы, листинги, изображения, таблицы или литературные источники;
- ключевые слова;
- сноски.

## Верхние и нижние индексы, акценты, конструкции языка программирования

### Набор математических формул в LaTeX'e

Производится с помощью тега **math**.

Ссылки на литературу и примеры набора формул можно найти здесь –



<http://lib.mexmat.ru/forum/viewtopic.php?p=901>

Пример:

```
<math>\int\limits_a^b f(x) dx \oint\limits_a^b f(x) dx </math>
```

$$\int_a^b f(x) dx \oint_a^b f(x) dx$$

```
<math>\tilde{\alpha}, \tilde{\beta}, \dots, \tilde{\omega}</math>
```

$\tilde{\alpha}, \tilde{\beta}, \dots, \tilde{\omega}$

```
<math>\widehat{ABC}</math>
```

$\widehat{ABC}$

```
<math>\sqrt[n]{\pi} + \sqrt{\pi+2}</math>
```

$$\sqrt[n]{\pi} + \sqrt{\pi+2}$$

```
<math mode="block">\mathbf{X} =
```

```
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
```

```
</math>
```

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Для нумерации формул используется атрибут tag, пример

```
<math tag="3" mode="block">2+2=4</math>
```

Будет выглядеть так

$$2 + 2 = 4$$

(3)

Для ликвидации TeX'овской нумерации формул, необходимы дополнительное изменение LaTeX кода, например, конструкцию

```
\begin{...}..формула..\end{...}
```

заменить на

```
\begin{...*}..формула..\end{...*}
```

**Важно!** При переводе TeX'овских формул в XML, необходимо произвести замены символов:

1. & переходит в &amp;
2. < переходит в &lt;
3. > переходит в &gt;

Для отладки формул LaTeX пользуйтесь скриптом

<http://www.intuit.ru/cgi-bin/tex2png.cgi>

Замену символов также можно осуществлять с помощью скрипта.

ID курса надо узнать дополнительно. Если формулы не отображаются, необходимо связаться с администрацией. При ошибках, сообщайте

а) где ошибка (номер лекции, в каком месте): б) код (в TeX'е, но лучше в XML, с тегом `math`, например, `<math>a+b^c</math>`).

Иногда необходима дополнительная настройка TeX'а для конкретного курса.

### Как набирать текст лекций, набранные в LaTeX'е

1. Формулы (строчные, inline) в LaTeX'е обрамляются символами `$`, `$..` LaTeX формула..`</math>`. В XML их следует набирать как `<math>.. LaTeX формула..</math>`.
2. Формулы (выделенные, block) в LaTeX'е обрамляются символами `\[` и `\]`, `\[.. LaTeX формула..\]`  
Ее следует набирать `<math mode="block">.. LaTeX формула.. </math>`
3. Иногда необходимо набирать `<math mode="text">.. LaTeX формула.. </math>` или `<math mode="text">$.. LaTeX формула.. $</math>`. Часто так надо набирать LaTeX формулы вида `\begin{...}..формула..\end{...}`

### Индексы

Назначение	Элемент
верхние индексы	<code>sup</code>
нижние индексы	<code>sub</code>
Векторы	<code>vector</code>
Акценты (выделения текста)	<code>em</code>
Акценты (над символами в формулах)	<code>accent</code>
конструкции языка программирования, переменные, формулы внутри строк текста	<code>code</code>

Пример:

`<p>В этом параграфе приводится пример расстановки <sup>верхних индексов</sup>, <sub>нижних индексов</sub>, <em>акцентов</em> и выводится название оператора <code>print</code>. А переменную  $X_{22}$  с чертой наверху стоит выводить так: <code><vector>X_{22}</vector></code>.</p>`

`<p>В формулах для задания акцента над символом используется тег <code><accent name="diff2">x</accent>=x</code>.</p>`

Список возможных значений атрибута **name** тега **accent** можно найти по адресу <http://www.intuit.ru/img/accent/>

### Ссылки

В тексте могут быть размещены ссылки на веб-ресурсы, курсы и лекции на сайте INTUIT.ru, листинги, изображения, таблицы или литературные источники. Для формирования ссылок применяется тег `<ref>`:

```
<ref src="URL or number"
      type="url|example|literature|image|table">...</ref>
```

или

```
<ref src="URL or number"
      type="url|example|literature|image|table" />
```

Значение атрибута type определяет тип сущности, на которую размещается ссылка, в атрибуте src указывается URL (для type="url") или значение поля number данной сущности (для всех других случаев). Если тег не пустой, ссылка применяется к его содержимому, иначе текст для ссылки формируется автоматически. По возможности следует применять пустые теги.

Если дается ссылка на источник литературы, номер источника необходимо взять в описании курса.

Примеры:

Добро пожаловать на сайт `<ref src="http://www.intuit.ru" type="url" />`

Посмотрите `<ref src="1.1" type="example">пример кода 1.1</ref>`

Для ссылок на лекции и курсы сайта INTUIT.ru значение атрибута type равно lecture и course, соответственно. Значение атрибута src равно

1. **номеру\_лекции\_или\_синониму** для лекции текущего курса;
2. **синоним\_курса::номеру\_лекции\_или\_синониму** для лекции другого курса;
3. **синоним\_курса::номеру\_лекции\_или\_синониму/урл\_страницы** для страницы лекции;
4. **синоним\_курса** для курса

`<p>Именованные ссылки:`

`<list>`

`<li><ref src="2" type="lecture">на 2-ю лекцию в текущем курсе</ref></li>`

`<li><ref src="htmlintro::2" type="lecture">на 2-ю лекцию курса <em>htmlintro</em></ref></li>`

`<li><ref src="htmlintro::2/2.html" type="lecture">на 2-ю лекцию курса <em>htmlintro</em>, 2-ю страницу </ref></li>`

`<li><ref src="htmlintro" type="course">на курс <em>htmlintro</em></ref></li>`

`</list>`

`Ссылки без текста:`

`<list>`

`<li>на 2-ю лекцию в текущем курсе: <ref src="2" type="lecture" /></li>`

`<li>на 2-ю лекцию курса <em>htmlintro</em>: <ref src="htmlintro::2" type="lecture" /></li>`

`<li>на 2-ю лекцию курса <em>htmlintro</em>, 2-ю страницу: <ref src="htmlintro::2/2.html" type="lecture" /></li>`

`<li>на курс <em>htmlintro</em>: <ref src="htmlintro" type="course" /></li>`

`</list></p>`

Отображаться этот код будет приблизительно так:

Именованные ссылки:

- ☞ [на 2-ю лекцию в текущем курсе](#)
- ☞ [на 2-ю лекцию курса \*\*htmlintro\*\*](#)
- ☞ [на 2-ю лекцию курса \*\*htmlintro\*\*, 2-ю страницу](#)
- ☞ [на курс \*\*htmlintro\*\*](#)

Ссылки без текста:

- ☞ на 2-ю лекцию в текущем курсе: ["Основные стандартные модули Python"](#)

- ☞ на 2-ю лекцию курса **htmlintro**: ["Структура HTML-документа и элементы разметки заголовка документа"](#)
- ☞ на 2-ю лекцию курса **htmlintro**, 2-ю страницу: ["Структура HTML-документа и элементы разметки заголовка документа"](#)
- ☞ на курс **htmlintro**: ["Введение в HTML"](#)

## Ключевые слова

В начале лекции указан список ключевых слов (терминов), относящихся к данной лекции. Каждое вхождение ключевого слова в текст необходимо отметить тегом `<keyword>`:

```
<keyword name="ключевое слово в именительном падеже или
аббревиатура" [type="ref|def"]>ключевое слово в нужном
падеже</keyword>
```

Значение атрибута `name` должно совпадать с ключевым словом в именительном падеже (если это не аббревиатура и не имя собственное, то строчными буквами) и служит для однозначной идентификации ключевого слова. Т.е. для каждого упоминания одного и того же ключевого слова независимо от его словоформы значение этого атрибута должно быть одинаковым.

В тексте может встретиться определение ключевого слова, либо простое его упоминание. В первом случае необходимо установить атрибут `type="def"`, во втором – `type="ref"` (значение по умолчанию). Это позволит, например, составить список вхождений термина в текст, а место, где дается его определение, отметить особо.

Пример:

```
<keyword name="профиль" type="def">Профилем</keyword> будем
называть набор опций, описывающих необязательные возможности.
...
```

Соответствие `<keyword name="профиль">профилю</keyword>` означает соответствие стандарту POSIX и поддержку заданных возможностей.

Поскольку определение ключевого слова может быть задано в «расплывчатой» форме, встает проблема выделения блока текста, содержащего определение. Для выделения текста определения служит тег `keyword-context`:

```
<keyword-context name="профиль"><keyword name="профиль"
type="def">Профилем</keyword> будем называть набор опций,
описывающих необязательные возможности.</keyword-context>
```

## Перевод слова

Для записи перевода слова используется тег `<tword>`.

Используются следующие атрибуты

<code>from</code>	язык слова, en, ru и т.д.
<code>to</code>	язык перевода, en, ru и т.д.
<code>word</code>	слово для перевода
<code>translation</code>	перевод
<code>transcription</code>	транскрипция
<code>forms</code>	формы слова
<code>sound</code>	звучание слова

Пример

```
<tword
  from="en"
  to="ru"
  word="book"
  translation="книга"
```

```
transcription="buk"
forms="books"
sound="tw_books.mp3"
/>
```

## Сноски

Сноска обозначается в тексте при помощи тега <footnote>:

```
<footnote>
  <term>слово</term>
  <def>Текст сноски</def>
</footnote>
```

При отображении сноски содержимое тега <term> может быть помечено как ссылка, при нажатии на которую всплывает окно с содержимым тега <def>; возможны и другие варианты отображения.

Пример:

```
Перед тем как верстать лекцию, прочтите, пожалуйста,
<footnote><term>спецификацию</term>
<def>Спецификацию можно найти по адресу
<ref src="http://www.intuit.ru/job/" type="url" />
</def>
</footnote>.
```

Возможный вариант отображения (при нажатии на ссылку в сноске всплывает окно с разъяснением сноски):

Перед тем как верстать лекцию, прочтите, пожалуйста, спецификацию<sup>1</sup>.

## Изображения

Изображения вставляются в текст при помощи тега <image>:

```
<image src="name.gif" title="Название" [number="1.1"] />
```

Приведенный пример может отобразиться так:

[здесь выводится изображение из файла name.gif]

**Рис. 1.1. Название**

Для того чтобы предварительный просмотр лекции с изображениями работал правильно, файлы с изображением должны лежать в одной папке с файлом лекции (lecture.xml).

Для отображения изображения в тексте (а не в отдельном параграфе) используется атрибут **type** со значением **inline**. По умолчанию, **type=block**.

```
После этого, нажмите на кнопку <image src="knopka.gif"
title="Кнопка" type="inline" />
```

**Важно!** В именах файлов изображений не используйте русские буквы!

Если необходима специальная разметка в подписи к рисунку, то можно использовать следующую конструкцию

```
<image src="name.gif" title="" number="1.1">Название
<em>рисунка</em></image>
```

Если некий текст необходимо оформить, как рисунок, то текст пишем в теге image, а атрибут src делаем пустым. Пример:

```
<image src="" title="Подпись к рисунку" number="1.1">Текст,
который оформляется, как рисунок</image>
```

Атрибут **align** позволяет управлять выравниванием расположения вынесенной картинки, допустимые значения:

1. **left** – налево, по умолчанию

2. **center** – центрировать
3. **right** – на право

#### Пример

```
<image src="name.gif" align="center" title="Название"
number="1.1" />
```

Для отображения рисунков большого размера используется атрибут **full-size**.

Если рисунок (например, 11\_03.jpg) имеет размер более 620 пикселей по высоте или ширине, то необходимо:

1. Уменьшить рисунок, так чтобы его размеры не превышали 620 по высоте и ширине, например, 11\_03sm.jpg
2. После чего в тексте лекции пишем следующий код

```
<image src="11_03sm.jpg" full-size="11_03.jpg" .../>
```

#### Страницы

При необходимости текст лекции может быть разбит на страницы при помощи тега `<page-break/>`. Тег рекомендуется помещать перед началом очередной главы, но в случае, если страницы при этом получаются слишком громоздкими, от этого правила разрешено отступать. В случае глав небольшого размера несколько глав могут быть размещены на одной странице. Критериями, говорящими о необходимости разбивки лекции на страницы являются:

1. Большой размер файла лекции. Рекомендуемый размер страницы – 10..15 Кб.
2. Чрезмерная длина лекции (визуальная). Рекомендуемая длина страницы – 3..4 экрана.

#### Вопросы/задания в тексте лекции.

```
<inline-test>
<exercise code="код">
<question>Вопрос
    ...
</question>
<answer type="single|multiple">
    <variant correct="yes|no" code="_код_1">
        ...</variant>
    <variant correct="yes|no" code="_код_2">
        ...</variant>
    ...
</answer>
</exercise>
</inline-test>
```

Атрибуты **exercise**

**code**                      уникальный в лекции

Атрибуты **variant**

**code**                      уникальный в упражнении, как правило цифры, которые нумеруют варианты ответа

## Структура файла теста

Тест представляет собой набор **задач**, относящихся к одной лекции. Для каждой задачи существует несколько ее **вариантов** (вопросов). Например, если в тесте 3 задачи, это означает, что для прохождения теста пользователь должен будет ответить на 3 вопроса. Конкретные вопросы, предложенные пользователю, будут выбираться из набора вариантов: первый вопрос – это один из вариантов к первой задаче, второй – один из вариантов ко второй, и т. д. У каждого варианта задачи существует набор **ответов**, каждый из которых помечается при верстке как правильный или неправильный. В зависимости от варианта, при ответе на вопрос пользователь может выбрать только один из ответов (режим **single**), или ему будет предоставлена возможность выбрать несколько ответов (режим **multiple**). Режим также устанавливается верстальщиком на основании простого правила: если правильный ответ строго один – это single, если правильных ответов нет, или их больше 1 – это multiple.

Также у каждого варианта задачи предусмотрен параметр time. Это время в секундах, которое дается на ответ. По-умолчанию time равен 180.

В ходе работы над тестом необходимо решить следующие задачи:

- 1) структурирование теста: разбиение на задачи, варианты и ответы; установка режима у вариантов задач, флагов «правильный» у ответов;
- 2) оформление текста: расстановка внутри вопросов и ответов верхних и нижних индексов, акцентов, ссылок, переменных и операторов, примеров кода, таблиц, рисунков и т. д.

При решении задачи (2) следует руководствоваться набором тегов, описанным выше для файла lecture.xml.

Файл теста должен иметь имя вида X.xml (где X – номер лекции) и должен соответствовать следующему шаблону:

```
<?xml version="1.0" encoding="windows-1251" standalone="no" ?>
<!DOCTYPE tasks SYSTEM
    "../../../tasks.dtd">
<?xml-stylesheet type="text/xsl"
    href="../../../tasks_preview.xsl"?>
<test>

<info>
    <title>Название теста</title>
    <title_en>Название на английском</title_en>
    <text><![CDATA[ Текст-аннотация ]]></text> <!-- макс. 256
    символов -->
    <advice><![CDATA[ Текст-Рекомендации ]]></advice>

    <subjects> <!-- Тематический план -->
        <subject name="Главный раздел 1" position="Порядковый
номер">
            <topic id="Уникальный номер подраздела">Тема
1</topic>
            <topic id="2">Тема 2</topic>
            ...
        </subject>
        <subject name="Преобразование типов" position="1">
            <topic id="3">Тема 1</topic>
            <topic id="4">Тема 2</topic>
```

```

...
</subject>
<subject name="Структура и исполнение программы"
position="2" />
    <topic id="5">Тема 1</topic>
    <topic id="6">Тема 2</topic>
...
</subject>
</subjects>
</info>

<tasks>
<task code="1">
    <variant code="1" answer-type="single" time="180">
        <subjects> <!-- подтемы -->
            <topic id="1" /> <!-- указывается уникальный номер темы
-->
            ...
            <topic id="2" />
        </subjects>
        <question> <![CDATA[Вопрос №1 задачи №1 ]]></question>
        <answers>
            <answer code="1" correct="yes"><![CDATA[Ответ 1
правильный ]]></answer>
            <answer code="2" correct="no"><![CDATA[Ответ 2
неправильный ]]></answer>
        </answers>
    </variant>

    <variant code="2" answer-type="multiple" time="120">
        <subjects>
            <topic id="2" />
            ...
            <topic id="3" />
        </subjects>
        <question> <![CDATA[Вопрос №2 задачи 1 - 2 прав.
ответа ]]></question>
        <answers>
            <answer code="1" correct="yes"><![CDATA[прав. ответ
1 ]]></answer>
            <answer code="2" correct="yes"><![CDATA[прав. ответ
2 ]]></answer>
            ...
        </answers>
    </variant>
</task>
<task code="2">
...
</task>

```



```
...
</tasks>

</test>
```

Конструкция `<![CDATA[текст]]>` позволяет использовать HTML-тэги для форматирования текста.

При верстке нового теста значения атрибутов `code` тегов `task` заполняются последовательно (1, 2, 3...). Значения атрибутов `code` для каждой задачи начинаются с единицы и также заполняются последовательно. То же верно и для ответов.

При изменении существующего теста нумерация должна оставаться такой же как и ранее.

**При удалении ответа (варианта, задачи) номера следующих за ним ответов (вариантов, задач) не должны изменяться!!!**

## Специальные символы (сущности)

Для вывода сущности в документе применяется выражение вида:

```
&entityName;
```

, где `entityName` – название сущности. Вместо приведенного выше выражения в конечном документе выведется символ, соответствующий данной сущности.

К специальным символам относятся:

- 1) символы, имеющие особое значение в языке XML

К данным символам относятся:

Символ	Сущность
<	&lt;
>	&gt;
&	&amp;
“	&quot;
‘	&apos;

Если в тексте содержатся данные символы, есть риск того, что он не пройдет валидацию, т.к. например символ `<` будет считаться открывающим тег – даже если это на самом деле не так. В этом случае его нужно будет заменить на `&lt;`. Другой частый случай – знак кавычек в значении атрибута. Если ввести его «как есть», программа «запутывается» в кавычках и документ также не проходит валидацию. Вместо этого необходимо ввести `&quot;`. Ну и наконец, если потребовалось ввести символ амперсанда (`&`), скажем, в названии «Procter&Gamble», программа, обрабатывающая документ, начинает искать сущность `&Gamble;`, но не находит ее. Вместо этого следует вводить название как «Procter&Gamble».

## Оформления списка литературы

Список литературы записывается в файле формата электронных таблиц .XLS Microsoft Excel. Поля таблицы -

	synony	nu	author			
cid	m	m	s	title	date_line	url

**cid** – ID курса, узнать у администрации или оставить пустым.

**synonym** - имя ссылки `_synonym_`, которое используется в конструкции вида

```
<ref src="_synonym_" type="literature" />
```

## Описание формата лекций INTUIT.ru

Как правило, совпадает с номером **num** (*смотрите файл Liter302.xls для примера*), но в случае, если литература в оригинальном курсе прилагается отдельно к каждой лекции, то для **synonym** следует использовать нумерацию вида <номер лекции>.<номер пункта литературы в списке литературы к лекции> (*смотрите файл Liter308.xls для примера*). Возможно для ссылок использовать имена для литературы, которые использует автор курса, оригинальная нумерация пунктов литературы часто встречается в книгах набранных в LaTeX'е.

**authors** – авторы

**title** – название

**date\_line** – выходные данные

**url** – URL, если есть

Примеры списков литературы смотрите в файлах Liter302.xls и Liter308.xls.