

Создание тестов без редактора. Часть II

Аннотация

В предыдущей статье "Создание тестов без редактора" мы познакомились с возможностью написания тестов для VeralTest на языке XML без использования редактора тестов. Мы рассмотрели простой тест из двух вопросов с единственным и множественным выбором ответа. Надеюсь, что у вас уже получается создавать простые тесты, используя сведения первой статьи.

Здесь будет уместно упомянуть о том, что формат тестов для системы VeralTest, основанный на стандарте XML называется Xml Test Format (XTF). Этот формат разработан VeralSoft специально для VeralTest. Формат является полностью открытым – полное описание формата должно появиться на сайте VeralSoft.com в ближайшее время (может и уже появилось, пока готовится эта статья)

В данной статье я расскажу о возможностях форматирования текста вопросов и ответов, предлагаемые форматом XTF.

Введение

Во многом форматирование текста в XTF схоже с языком HTML, так что если вы умеете создавать HTML странички, то материал этой статьи будет для вас знаком. Не забывайте только, что мы используем стандарт XML, а это значит, что в отличие от HTML нужно обязательно закрывать теги и, поскольку XML чувствителен к регистру, имена всех элементов писать в нижнем регистре клавиатуры. Например, тег перехода на новую строку в HTML может быть записан как
, так и
 - и это будут абсолютно правильные варианты. Используя язык XML, тот же тег может быть записан только так:
. Обратите внимание на символ "/". В прошлой статье я уже говорил, что в языке XML все без исключения элементы должны иметь открывающий и закрывающий теги. В нашем примере br – это элемент который не должен содержать внутри себя ничего, но по правилам XML закрывающий тег все равно должен быть, т.е., правильной будет такая запись элемента br:
</br>. Но это слишком длинно, поэтому для элементов, которые не имеют содержимого, XML разрешает сокращенную запись: вместо <имя_элемента></имя_элемента> можно использовать <имя_элемента />.

Оформление текста

Текст вопросов, ответов и сообщений допускает применение различных шрифтов, цветов и стилей.

Изменение типа, размера и цвета шрифта.

Используемый шрифт, размер символов и цвет текста задаются элементом . Этот элемент имеет следующие атрибуты:

- face – задает имя используемого шрифта, например face="Arial";
- size задает размер шрифта. Размер может быть от 1 до 7. Например: size="7";
- color – задает цвет текста. Допускаются как мнемонические названия цветов: black,

silver, red, purple и т.д, так и цвета записанные в виде кодов цветовой модели RGB, например #000000, #C0C0C0 и т.д; Пример: color= "blue"

Примеры:

Допустим нам нужно записать текст вопроса шрифтом Times New Roman размера 2. Тогда, используя элемент font, этот пример следует записать так:

`Текст вопроса`.

Пусть, теперь мы хотим выделить текст вопроса красным цветом:

`Текст вопроса`

Естественно, что можно применять элемент не ко всему тексту, а к какой то его части, например:

Этот текст не имеет форматирования ``Этот текст записан шрифтом Arial размером 7````Этот текст записан шрифтом Times New Roman, размером 5, цвет: голубой``

При выполнении теста в окне браузера наш текст будет отображаться так:



Элементы допускают вложенность, т.е., мы можем использовать один элемент внутри другого:

Этот текст не имеет форматирования. `` Применили шрифт: Arial.
``Увеличили размер. ``Здесь напишем красным `` ``
``

В результате получим:

Этот текст не имеет форматирования. Применили шрифт: Arial. Увеличили размер. **Здесь напишем красным**

Форматирование текста с помощью стилей

В XHTML предусмотрены следующие элементы для изменения стиля текста:

- `` - Полужирный шрифт;
- `<i>` - Курсив;
- `<u>` - Подчеркнутый текст;
- `<sup>` - Надстрочный текст;
- `<sub>` - Подстрочный текст.

Рассмотрим пример форматирования текста при помощи всех этих элементов:

Выделение текста `` полужирным шрифтом ``, `<i>` а теперь курсивом `</i>`.
`<u>` Подчеркнем текст `</u>`. Применение `^{` надстрочного написания `}`. Применение `_{` подстрочного написания `}`.

А вот как это будет выглядеть:

Выделение текста **полужирным шрифтом**, а теперь курсивом. Подчеркнем текст.
Применение ^{надстрочного написания}. Применение _{подстрочного написания}

Элементы стиля могут быть вложены друг в друга, что позволяет комбинировать стили при оформлении текста:

Текст без стиля. `` Полужирный. `<i>` Полужирный курсив `</i>`

Текст без стиля. **Полужирный. Полужирный курсив**

Абзацы и разрывы строк

Для удобства восприятия большого текста его целесообразно разделять на абзацы. Для обозначения абзаца в XHTML, так же как и в HTML применяется элемент `<p>`.

Пример:

`<p>` Длинный текст лучше разделять на абзацы. Это был первый абзац. `</p>`

`<p>` Второй абзац нашего длинного текста. Видно, что абзацы отделяются друг от друга пустой строкой. `</p>`

`<p>` А это третий абзац. `</p>`

Длинный текст лучше разделять на абзацы. Это был первый абзац.

Второй абзац нашего длинного текста. Видно, что абзацы отделяются друг от друга пустой строкой.

А это третий абзац.

При помощи атрибута align можно задавать выравнивание текста в абзаце. Возможно выравнивание по левому краю: align="left" (это выравнивание по-умолчанию), по центру: align="center", и по правому краю: align="right".

Рассмотрим различные типы выравнивания на примере:

`<p align="left">Этот текст будет выровнен по левому краю.</p>`

`<p>Этот текст тоже будет выровнен по левому краю, хотя мы и не указали атрибут align.</p>`

`<p align="center">Этот текст будет выровнен по центру.</p>`

`<p align="right">Этот текст будет выровнен по правому краю.</p>`

Этот текст будет выровнен по левому краю.

Этот текст тоже будет выровнен по левому краю, хотя мы и не указали атрибут align.

Этот текст будет выровнен по центру.

Этот текст будет выровнен по правому краю.

Бывают ситуации, когда необходимо сделать разрыв в строке и начать вывод текста с новой строчки, не делая двойного интервала между строками. Здесь поможет элемент разрыва строки `
`. Обратите внимание – элемент `
` всегда пуст.

Пример:

Сейчас мы сделаем разрыв строки. Вот здесь. `
` Теперь вывод текста начат с новой строки.

Сейчас мы сделаем разрыв строки. Вот здесь.
Теперь вывод текста начат с новой строки.

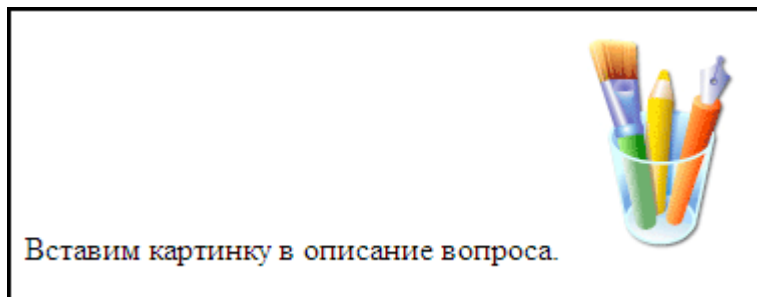
Вставка изображений

В текст вопроса, описание ответа или сообщение можно вставлять изображения в форматах GIF и JPEG. Для вставки изображения, так же как и в HTML, используется элемент ``.

Имя файла картинки задается атрибутом src.

Пример:

Вставим картинку в описание вопроса. ``

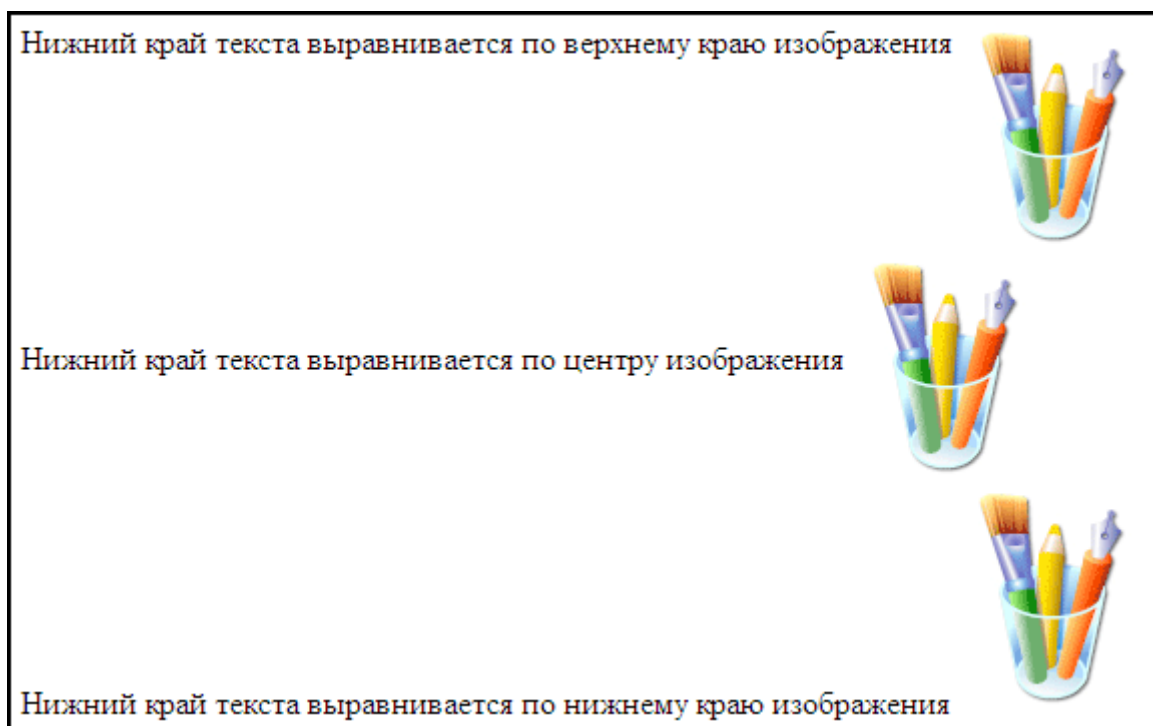


Выравнивание изображения относительно текста задается параметром атрибутом align. Рассмотрим наиболее популярные варианты выравнивания:

Нижний край текста выравнивается по верхнему краю изображения `
`

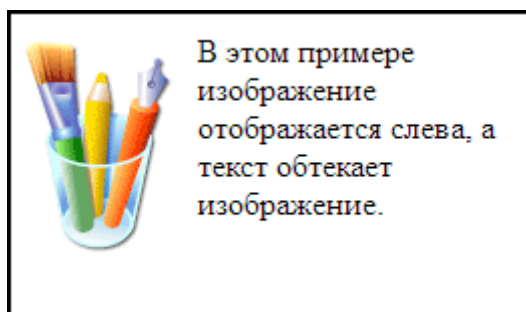
Нижний край текста выравнивается по центру изображения `
`

Нижний край текста выравнивается по нижнему краю изображения ``

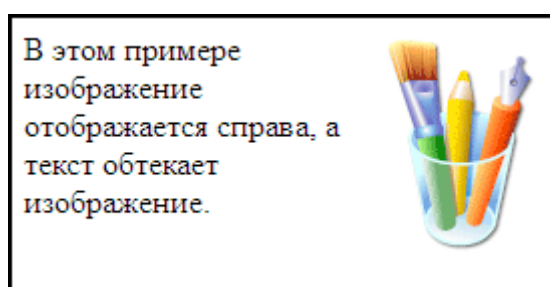


Во всех рассмотренных примерах изображение привязано к тексту и перемещается вместе с текстом. Однако можно разместить изображение, привязав его к левой или правой части экрана, а текст будет обтекать его:

`` В этом примере изображение отображается слева, а текст обтекает изображение.



`` В этом примере изображение отображается справа, а текст обтекает изображение.



Не забывайте, что изображения, которые вы включаете в текст вопросов, ответов и сообщений, должны быть упакованы в файл вместе с content.xml (как это сделать, я рассказывал в предыдущей статье). По этой же причине ссылка на изображение, которая указывается в атрибуте src, должна содержать только имя файла без указания какого-либо пути.

Заключение

Надеюсь, что воспользовавшись материалом этой статьи, вы сможете сделать ваш тест более выразительным и удобным, ну а в следующей статье мы поговорим о других возможностях, которые предоставляет формат XHTML.

Соколов А.Ф.

VeralSoft.com