

Кратко о *re*Структурированном тексте

<http://docutils.sourceforge.net/docs/user/rst/quickref.html>

Взято с этого ресурса (англ.).

Полное описание можете найти по ссылке [reStructuredText](#).

Ссылки подобные этой "[\(подробнее\)](#)" указывают на HTML версию полной [reStructuredText спецификации](#).

Это относительные ссылки, если они не работают используйте "[Quick reStructuredText](#)" документ.

Разделы

[\(подробнее\)](#)

Исходный текст	Результат
<pre>===== Заголовок ===== Подзаголовок ----- Заголовки подчеркиваются (или оббиваются сверху и снизу) с помощью небуквенных и нецифровых 7-битных ASCII символов. Рекомендуется использовать: "``= - ` : ' " ~ ^ _ * + # < >``". Оббивка должна быть не короче текста заголовка.</pre>	<p>Заголовок</p> <p>Подзаголовок</p> <p>Заголовки подчеркиваются (или оббиваются сверху и снизу) с помощью небуквенных и нецифровых 7-битных ASCII символов. Рекомендуется использовать: "``= - ` : ' " ~ ^ _ * + # < >``". Оббивка должна быть не короче текста заголовка.</p>

Параграфы

[\(подробнее\)](#)

Исходный текст	Результат
<p>Это параграф.</p> <p>Строки параграфов начинаются от левой границы и разделяются параграфы друг от друга пустой строкой.</p>	<p>Это параграф.</p> <p>Строки параграфов начинаются от левой границы и разделяются параграфы друг от друга пустой строкой.</p>

Ненумерованные списки

[\(подробнее\)](#)

Исходный текст	Результат
<p>Ненумерованные списки:</p> <ul style="list-style-type: none">- Это элемент списка 1- Это элемент списка 2 <p>- Используются: "-", "*" или "+". Далее через пробел идет сам текст.</p> <p>Важно, чтобы чистая строка присутствовала перед первым элементом списка и после последнего. Между элементами - не обязательно.</p>	<p>Ненумерованные списки:</p> <ul style="list-style-type: none">• Это элемент списка 1• Это элемент списка 2• Используются: "-", "*" или "+". Далее через пробел идет сам текст. <p>Важно, чтобы чистая строка присутствовала перед первым элементом списка и после последнего. Между элементами - не обязательно.</p>

Нумерованные списки

(подробнее)

Исходный текст	Результат
Нумерованный список: 3. Это первый элемент списка 4. Это второй элемент списка 5. Номера – арабские цифры, одиночные буквы или римские числа 6. Элементы списка должны быть последовательно пронумерованны #. Это автоматическая нумерация	Нумерованный список: 3. Это первый элемент списка 4. Это второй элемент списка 5. Номера - арабские цифры, одиночные буквы или римские числа 6. Элементы списка должны быть последовательно пронумерованны 7. Это автоматическая нумерация

Определения

(подробнее)

Исходный текст	Результат
Определения: Что Списки определений связывают термин с его поределением. Как Термин – это однострочковая фраза, а определение – это одно или несколько параграфов, текст ссылающийся на другой термин. Пустые строки между терминами и определениями не требуются.	Определения: Что Списки определений связывают термин с его поределением. Как Термин - это однострочковая фраза, а определение - это одно или несколько параграфов, текст ссылающийся на другой термин. Пустые строки между терминами и определениями не требуются.

Списки полей

([подробнее](#))

Исходный текст	Результат
<pre>:Авторы: Tony J. (Tibs) Ibbs, David Goodger (и кто-нибудь еще) :Версия: 1.0 от 08/08/2001 :Посвящается: Моему отцу.</pre>	<pre>Авторы: Tony J. (Tibs) Ibbs, David Goodger (и кто-нибудь еще) Версия: 1.0 от 08/08/2001 Посвящается: Мему отцу.</pre>

Символьные блоки

([details](#))

Исходный текст	Результат
<pre>Параграф, содержащий только два двоеточия обозначает, что следующий с отступом или промаркерванный текст является символьным блоком. :: Пробел, перевод строки, чистые строки и все типы разметок (такие как *эта* или \эта) представляют символьные блоки. Параграф, включающий только '::', в результате не будет отображаться. Двоеточия ``::`` можно добавлять в самый конец параграфа. ``::`` не будут отображаться, если им предшествует пробел.</pre>	<pre>Параграф, содержащий только два двоеточия обозначает, что следующий с отступом или промаркерванный текст является символьным блоком. Пробел, перевод строки, чистые строки и все типы разметок (такие как *эта* или \эта) представляют символьные блоки. Параграф, включающий только '::', в результате не будет отображаться. Двоеточия :: можно добавлять в самый конец параграфа. :: не будут отображаться, если им предшествует пробел. И :: будут отображаться как одно двоеточие, если они идут сразу за текстом, как вот эти:</pre>

Исходный текст	Результат
<p>И ``:`` будут отображаться как одно двоеточие, если они идут сразу за текстом, как вот эти::</p> <p> Это очень удобно использовать.</p> <p>Символьные блоки заканчиваются, когда происходит возвращение по отступу предыдущего параграфа. Подразумевается что-то вроде этого::</p> <p> Мы начали тут продолжили тут закончили здесь.</p> <p>Промаркерованные строки тоже могут образовывать символьные блоки:</p> <p>> Метки, используемые в электронной почте и > в Haskell программировании.</p>	<p>Это очень удобно использовать.</p> <p>Символьные блоки заканчиваются, когда происходит возвращение по отступу предыдущего параграфа. Подразумевается что-то вроде этого::</p> <p> Мы начали тут продолжили тут закончили здесь.</p> <p>Промаркерованные строки тоже могут образовывать символьные блоки:</p> <p>> Метки, используемые в электронной почте и > в Haskell программировании.</p>

Строковые блоки

[\(подробнее\)](#)

Исходный текст	Результат
<p> Строковые блоки еспользуются для форматирования адресов, стихов и списков.</p> <p> Каждая новая строка начинается с вертикальной черты (" "). Строки прерываются и продолжаютс с нужным отступом.</p> <p> Длинные строки можно прерывать и продолжать с тем же отступом, что и строка с вертикально чертой</p>	<p>Строковые блоки еспользуются для форматирования адресов, стихов и списков.</p> <p>Каждая новая строка начинается с вертикальной черты (" "). Строки прерываются и и продолжаютс с нужным отступом. Длинные строки можно прерывать и продолжать с тем же отступом, что и строка с вертикально чертой</p>

Таблицы

(подробнее)

Есть два способа построения таблиц в реСтруктурированном тексте. Сеточные таблицы самодостаточны, но сложны в построении. Простые таблицы легки в построении, но есть ограничения (нет объединения ячеек, и т.д.).

Исходный текст	Результат																		
Сеточная таблица: <pre>+-----+-----+-----+ Заголовок 1 Заголовок 2 Заголовок 3 +=====+=====+=====+ строка 1 столбец 2 столбец 3 +-----+-----+-----+ строка 2 объединение столбцов +-----+-----+-----+ строка 3 объединение - Ячейки +-----+-----+-----+ строка 4 строка - содержат +-----+-----+-----+ строка 4 - блоки. +-----+-----+-----+</pre>	Сеточная таблица: <table border="1"><thead><tr><th>Заголовок 1</th><th>Заголовок 2</th><th>Заголовок 3</th></tr></thead><tbody><tr><td>строка 1</td><td>столбец 2</td><td>столбец 3</td></tr><tr><td>строка 2</td><td colspan="2">объединение столбцов</td></tr><tr><td>строка 3</td><td>объединение</td><td rowspan="2">• Ячейки • содержат • блоки.</td></tr><tr><td>строка 4</td><td>строка</td></tr></tbody></table>	Заголовок 1	Заголовок 2	Заголовок 3	строка 1	столбец 2	столбец 3	строка 2	объединение столбцов		строка 3	объединение	• Ячейки • содержат • блоки.	строка 4	строка				
Заголовок 1	Заголовок 2	Заголовок 3																	
строка 1	столбец 2	столбец 3																	
строка 2	объединение столбцов																		
строка 3	объединение	• Ячейки • содержат • блоки.																	
строка 4	строка																		
Простая таблица: <pre>===== ===== ===== Вход Выход ----- ----- ----- А В А or В ===== ===== ===== Ложь Ложь Ложь Истина Ложь Истина Ложь Истина Истина Истина Истина Истина ===== ===== =====</pre>	Простая таблица: <table border="1"><thead><tr><th colspan="2">Вход</th><th>Выход</th></tr><tr><th>А</th><th>В</th><th>А or В</th></tr></thead><tbody><tr><td>Ложь</td><td>Ложь</td><td>Ложь</td></tr><tr><td>Истина</td><td>Ложь</td><td>Истина</td></tr><tr><td>Ложь</td><td>Истина</td><td>Истина</td></tr><tr><td>Истина</td><td>Истина</td><td>Истина</td></tr></tbody></table>	Вход		Выход	А	В	А or В	Ложь	Ложь	Ложь	Истина	Ложь	Истина	Ложь	Истина	Истина	Истина	Истина	Истина
Вход		Выход																	
А	В	А or В																	
Ложь	Ложь	Ложь																	
Истина	Ложь	Истина																	
Ложь	Истина	Истина																	
Истина	Истина	Истина																	

Переходы

([подробнее](#))

Исходный текст	Результат
<p>Указатель перехода – это горизонтальная линия из 4 и более повторяющихся знаков препинания</p> <p>-----</p> <p>Переход не должен начинать или заканчивать раздел и два перехода не должны стоять сразу друг за другом.</p>	<p>Указатель перехода - это горизонтальная линия из 4 и более повторяющихся знаков препинания</p> <hr/> <p>Переход не должен начинать или заканчивать раздел и два перехода не должны стоять сразу друг за другом.</p>

Сноски

([подробнее](#))

Исходный текст	Результат
<p>Ссылки сносок таковы [5]_. Важно, чтобы сноска могла быть перегруппированна, т.е. помещена в конец страницы.</p> <p>.. [5] Числовая сноска. После ``]`` не должно быть двоеточия.</p>	<p>Ссылки сносок таковы ⁵. Важно, чтобы сноска могла быть перегруппированна, т.е. помещена в конец страницы.</p> <hr/> <p>[5] Числовая сноска. После] не должно быть двоеточия.</p>
<p>Сноски с автоматической нумерацией, тоже возможны, так [#]_ и так [#]_.</p> <p>.. [#] Это первая сноска. .. [#] Это вторая сноска.</p> <p>Им можно указывать 'метку автономера'. Например, [#fourth]_ и [#third]_.</p>	<p>Сноски с автоматической нумерацией, тоже возможны, так ¹ и так ².</p> <p>Им можно указывать 'метку автономера'. Например, ⁴ и ³.</p>

Исходный текст	Результат
<pre>.. [#third] типа третья third_ .. [#fourth] типа четвертая fourth_</pre>	<hr/> <p>[1] Это первая сноска. [2] Это вторая сноска. [3] типа третья third [4] типа четвертая fourth</p>
<p>Авто-символ сносок используйте вот так [*]_ и [†]_.</p> <pre>.. [*] Это первый символ. .. [†] Это второй символ.</pre>	<p>Авто-символ сносок используйте вот так [*] и [†].</p> <hr/> <p>[*] Это первый символ. [†] Это второй символ.</p>

Цитаты

([подробнее](#))

Исходный текст	Результат
<p>Ссылки на цитаты выглядят так [CIT2002]_. Важно чтобы цитаты можно было разместить в конце страницы.</p> <pre>.. [CIT2002] Это цитата (как часто используемая в журналах).</pre> <p>В метках цитат используются буквы, цифры, знак подчеркивания, дефис. Регистр букв не важен.</p> <p>Некая цитата как эта [this]_, может ссылаться сюда же this_.</p>	<p>Ссылки на цитаты выглядят так [CIT2002]. Важно чтобы цитаты можно было разместить в конце страницы.</p> <p>В метках цитат используются буквы, цифры, знак подчеркивания, дефис. Регистр букв не важен.</p> <p>Некая цитата как эта [this], может ссылаться сюда же this.</p> <hr/> <p>[CIT2002] Это цитата (как часто используемая в журналах) [this] здесь.</p>

Исходный текст	Результат
.. [this] здесь.	

Ссылки

([подробнее](#))

Внешние ссылки

Исходный текст	Результат
<p>Внешние ссылки выглядят так ссылка_.</p> <p>.. _ссылка: http://www.if-site.com/</p> <p>Если несколько слов, тогда так `ссылка в несколько слов`_.</p> <p>.. _`ссылка в несколько слов`: http://www.if-site.com/</p>	<p>Внешние ссылки выглядят так ссылка.</p> <p>Если несколько слов, тогда так ссылка в несколько слов.</p>

Внутренние ссылки

Исходный текст	Результат
<p>Внутренние ссылки делаем так_.</p> <p>.. _так:</p> <p>Такой вот пример внутренних ссылок.</p>	<p>Внутренние ссылки делаем так</p> <p>Такой вот пример внутренних ссылок</p>

Неявные ссылки

([подробнее](#))


Заголовки разделов, сноски и цитаты автоматически генерируют ссылки (текст заголовков или сносок/цитат становятся именами для ссылок).

Исходный текст	Результат
Заголовки тоже ссылки ===== Неявная ссылка: `Заголовки тоже ссылки`_.	Заголовки тоже ссылки Неявная ссылка: Заголовки тоже ссылки .

Иллюстрации

([подробнее](#))

Иллюстрации вставляются в документ с помощью механизма директив. Для описания стандартных директив смотрите [reStructuredText Directives](#).

Исходный текст	Результат
Например: .. image:: pic1.gif Вот тут другой пример как вставить картинку рожица .. рожица image:: pic1.gif	Например:  Вот тут другой пример как вставить картинку 