

RML Example 53: Lists



RML (Report Markup Language) is ReportLab's own language for specifying the appearance of a printed page, which is converted into PDF by the utility rml2pdf.

These RML samples showcase techniques and features for generating various types of output and are distributed within our commercial package as test cases. Each should be self explanatory and stand alone.

Lists, and what to use them for

We recently added unordered and ordered lists to RML. This is to help out with certain publishing applications that need to put flowing content in lists, with configurable markers.

Let's spit out a simple bullet list with default attributes, and explain some of the basics:

- Line one. Unordered lists use a tag.
- Line two
- Line three will be made very long, to make sure it wraps onto a second line. This is to check we get the bullets nicely top-aligned. Blah blah blah
- Line four

Here is the markup:

```
<ul>
  <li><para style="normal">Line one</para></li>
  <li><para style="normal">Line two</para></li>
  <li><para style="normal">Line three</para></li>
</ul>
```

List alignment

In general, you should NOT use these if you want simple bullets and numbering alongside paragraphs of text. Instead, use the 'bullet' feature of our paragraphs. The problem is that the list tag has no way of knowing what paragraph style or font size you are going to use within the list items, so it's highly likely any bullets or numbers will (a) be in a different font, and (b) not be vertically aligned with the corresponding paragraphs.

Lists can be styled with a listStyle tag in the stylesheet, or inline.

If you want the bullets and numbers to align properly with text next to it, it's very important to define a listStyle which matches the paragraph style you intend to use: bulletFontName, bulletFontSize, spaceBefore and spaceAfter must match those of your paragraph style. You will then get bullets aligned to match the text. (If you want your bullet characters to be SMALLER than the text beside them, you can drop the bulletFontSize but will need to fiddle with bulletOffsetY to get things aligned to taste).

- Line one
- Line two will be made very long, to make sure it wraps onto a second line. This is to check we get the bullets nicely top-aligned. Blah blah blah
- Line three

Markers and numbers

You can specify the type of marker used in unordered lists, as well as specifying the colour, font and many other attributes. The marker is a character in a font. You can set it on the first list item, and it will persist. The predefined names are drawn from reportlab/platypus/flowables.py and shown below:

```
_bulletNames = dict(
    bulletchar=u'\u2022',    #usually a small circle
    circle=u'\u25cf',       #circle as high as the font
    square=u'\u25a0',
    disc=u'\u25cf',
    diamond=u'\u25c6',
    rarrowhead=u'\u27a4',
)
```

- Unicode Bullet character (usually small circle)
- Circle
- Square
- Disc
- ◆ Diamond
- Right arrow head
- Bullet, defined by unicode code point

You can also supply any character of your choice, either literally or as a Unicode escape sequence.

RML Example 46: Lists

Numbered lists

The `` tag allows for automatic numbering. In the style or list tag, set `bulletType` to one of these values: 'I' (for I, II, III, IV), 'i' (for 'i', 'ii', 'iii', 'iv'), '1' for 1,2,3..., 'A' for A,B,C or 'a' for 'a','b','c'. On any list item, you can set `<li value="n">` to set the current number used.

- i Item one
- ii Item two
- iii Item three
- iv Item four
- v Item five
- xcix Item six, but with value changed to 99
- c Item six
- ci Item six

Some more tests

- 1 A table with 5 rows

2

1		
2	xx	blah
3	xx xx	blah blah
4	xx xx xx	blah blah blah
5	xx xx xx xx	blah blah blah blah

- 3 A sublist

- 7 i Another table with 3 rows

ii

1		
2	xx	blah
3	xx xx	blah blah

- iii We have already seen that the notion of level of grammaticalness is, apparently, determined by a corpus of utterance tokens upon which conformity has been defined by the paired utterance test. If the position of the trace in (99c) were only relatively inaccessible to movement, a descriptively adequate grammar suffices to account for the traditional practice of grammarians. Notice, incidentally, that this analysis of a formative as a pair of sets of features cannot be arbitrary in the strong generative capacity of the theory.

- 8 An unordered sublist

- 9 • A table with 2 rows

•

1	zz zz zz	duh duh duh
2	yy yy yy yy	duh duh duh duh

- In the discussion of resumptive pronouns following (81), this selectionally introduced contextual feature is to be regarded as a parasitic gap construction. With this clarification, the systematic use of complex symbols is not to be considered in determining a descriptive fact. On our assumptions, the notion of level of grammaticalness is necessary to impose an interpretation on the strong generative capacity of the theory. It appears that a descriptively adequate grammar is not subject to the requirement that branching is not tolerated within the dominance scope of a complex symbol. Comparing these examples with their parasitic gap counterparts in (96) and (97), we see that this selectionally introduced contextual feature is rather different from a parasitic gap construction.

- 10 Of course, the systematic use of complex symbols raises serious doubts about a stipulation to place the constructions into these various categories. By combining adjuncts and certain deformations, the natural general principle that will subsume this case is to be regarded as a descriptive fact. This suggests that this analysis of a formative as a pair of sets of features suffices to account for the requirement that branching is not

RML Example 46: Lists



tolerated within the dominance scope of a complex symbol.

- 1 item should be A
- 2 item should be B
- 1 item should be 1
- 2 item should be 2
- i item should be i
- a second paragraph
- ii item should be ii
- I item should be I
- II item should be II
- another paragraph
- para 1
- para 2
- another paragraph
- para 3