Erlguten

Welcome to ErlGuten. ErlGuten is a system for typesetting. The aim of ErlGuten is to produce high quality PDF from a layout language or from a program. The ErlGuten distribution includes a programming API, so that programs can produce PDF - and a typesetting system for typesetting documents written in XML.

The document that you are reading was authored in XML and typeset by ErlGuten. The name ErlGuten is chosen because the program is written in Erlang - the Guten part is a reference to Gutenberg the father of printing.

ErlGuten is a system for high quality typesetting, so we take a great deal of care when formatting text, a large number of optimizations are performed which improve the quality of the printed text. Many of these optimizations are only found in expensive professional type-setting programs. We believe that WYSIWYG programs have destroyed the fine art of typesetting - ErlGuten is a modest attempt to improve the situation.

We have chosen XML as the input language for it's wide appeal - not because of any intrinsic advantages that XML might have. XML provides only a thin abstraction layer over the typesetting system - so the adventurous can use the programming interface to ErlGuten - to directly produce typographic quality PDF in real-time. We expect this facility to be useful for the dynamic generation of documents from web-servers.

In ErlGuten we take the view that the highest level of abstraction is the layout of a document - we are very concerned that the user can specify the exact position of text on the printed page. At the next level of abstraction we are concerned with the typefaces that are used to format different regions of the document - our system makes use of the 14 in-built PDF fonts - for specialist work type 1 postscript fonts can be used and embedded in the resulting output. The ErlGuten distribution includes a number of public domain Type 1 postscript fonts that can be freely included in any document produced by ErlGuten.

Here, paragraphs are composed of 11/13 NewCenturySchlbk-Roman set in a 30 pica measure. Emphasized text is set in NewCentrySchlbk-Italic. The TeX hyphenation algorithm is also implemented. I have also some cursive text and an example of an Erlang term.

The term `{person,"Joe"}` is an Erlang term. The variable `X`, was immediately followed by a comma. The justification algorithm does proper kerning, which is more than Microsoft Word can do. AWAY again is correctly kerned! Erlang terms `{like,this}` are typeset in courier.

Note, if punctuation is used, it must be attached to the preceding tagged block, like this, but never like this, which is incorrect.

This is a small footnote set in 6/8 Times Italic. Note that the order of lines in the output has nothing to do with the order on the page.