

# erlPress\_core

Generate PDFs from your Erlang applications!

User applications based on the Adobe Portable Document Format, PDF, compose text, images, 2-D geometric objects, and presentation instructions for distribution across diverse output devices. The goal is faithful and consistent display of human-readable content. Presentation instructions are encoded in the page description language Adobe PostScript.

*erlPress\_core.01* functions transform human-readable content into Erlang data structures, generate Postscript object code, and format the code into properly structured PDF files. Based on *PDF Reference: third edition, Version 1.4*, *erlPress\_lib.01* provides functions to:

- Display Adobe Type I fonts
- Position and copyfit text
- Justify, kern, and rotate text
- Specify and position images and graphic elements
- Support checklists

*erlPress\_lib.01* revises and extends the Hugh Watkins fork of the Erlang Erlguten library originally developed by Joe Armstrong. See: <https://github.com/hwatkins/erlguten>

## Roadmap:

- Extend and debug Markdown parsing
- Footnotes
- Tables
- Articles and beads
- Test! Test! Test!
- TrueType and OpenType fonts
- Impostion and layout functions
- PDF-to-Erlang functions
- Applications for diverse print formats

*erlPress\_core.01* provides a foundation for development of Erlang applications that support creative generation of print formats ranging from business cards to technical books. We immodestly imagine the evolution of a comprehensive Human Communication Platform, HCP, parallel to OTP.

Join in!

Help make it so.

# Best of erlPress\_core.01

## Justified Text

This is normal text set in 14/21 Times Roman. It includes *emphasized terms*, set in Times-Italic. The TeX hyphenation algorithm is implemented. The term {`person`, "Joe"} is an Erlang term. The variable `X`, was immediately followed by a comma. The justification algorithm does proper *kerning*. AWAY is correctly kerned! Erlang terms {`like`, `this`} are typeset in *courier*.

## Boxed Text

This is normal text set in 10/15 Helvetica. It includes emphasized terms, set in Helvetica-Oblique. The TeX hyphenation algorithm is implemented. The term {`person`, "Joe"} is an Erlang term. The variable `x`, was immediately followed by a comma. The justification algorithm does proper kerning. AWAY is correctly kerned! Erlang terms {`like`, `this`} are typeset in *courier*.

## Poetry (Preformatted Text)

Two roads diverged in a yellow wood,  
And sorry I could not travel both  
And be one traveler, long I stood  
And looked down one as far as I could  
To where it bent in the undergrowth;  
- Robert Frost

# Centered Text

This is normal text set in 14/21 Times Roman.

It includes *emphasized terms*, set in Times-Italic. The TeX hyphenation algorithm is implemented. The term {person, "Joe" } is an Erlang term.

The variable X, was immediately followed by a comma. The justification algorithm does proper *kerning*. AWAY is correctly kerned! Erlang terms {like, this} are typeset in *courier*.

# Right Justified Text

This is normal text set in 14/21 Times Roman. It includes *emphasized terms*, set in Times-Italic. The TeX hyphenation algorithm is implemented. The term {person, "Joe" } is an Erlang term. The variable X, was immediately followed by a comma. The justification algorithm does proper *kerning*. AWAY is correctly kerned! Erlang terms {like, this} are typeset in *courier*.

# Kerning

The blahonga font will fall back to Times-Roman  
This is a check of ( ) \ escape chars  
This is a test of Kerning: AWAY again  
This is a text without Kerning: AWAY again

# Fonts

## Times-Roman 18 pt

*Times-Italic 18 pt*

# Times-Bold 18 pt

*Times-BoldItalic 18 pt*

Courier 18 pt

*Courier-Oblique 18 pt*

## Courier-Bold 18 pt

## **Courier-BoldOblique 18 pt**

## Helvetica 18 pt

## *Helvetica-Oblique 18 pt*

# **Helvetica-Bold 18 pt**

## ***Helvetica-BoldOblique 18 pt***

ΑΒΧΔΕΦΓΗΙΩΚΛΜΝΟΠΘΡΣΤΥζΩΞΨΖ

αβχδεφγηιφκλμνοπθρστυῶωξψζ

0123456789

$\sim ! \equiv \# \exists \% \perp \& * () - \_ = + [ \{ \} ] | ; : \exists , < . > / ?$

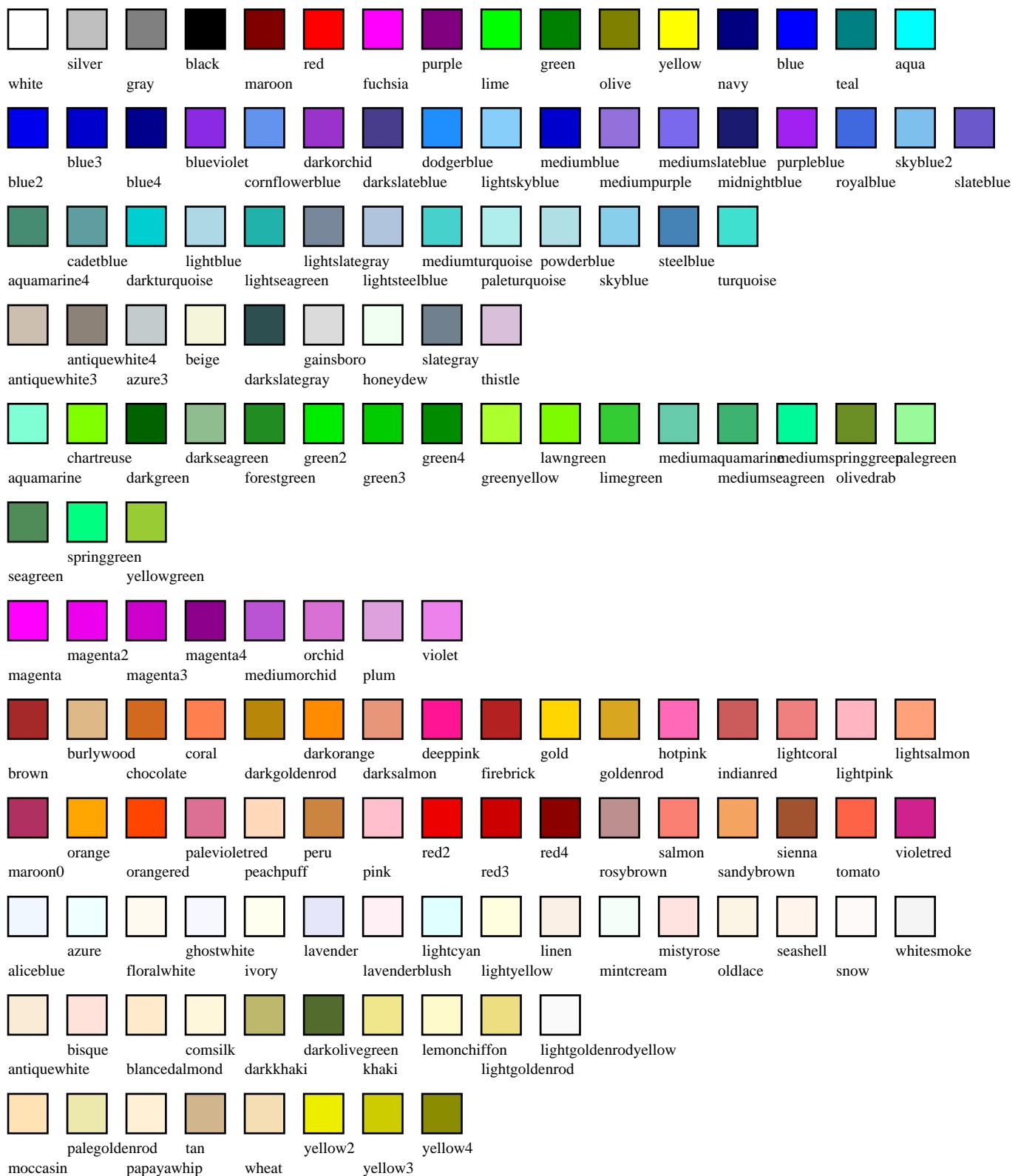
A decorative horizontal border featuring a repeating pattern of black geometric shapes on a white background. The pattern includes various star-like and cross-like symbols.

A decorative horizontal border featuring a variety of black geometric and floral motifs arranged in a repeating sequence.

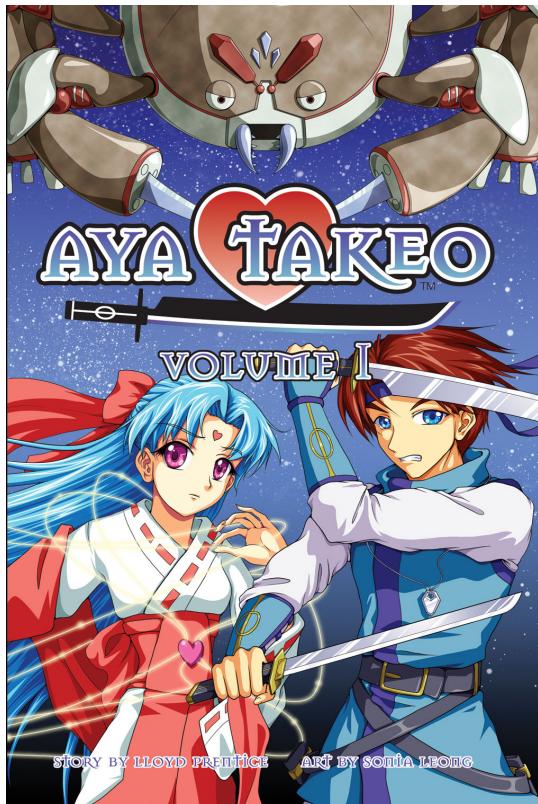
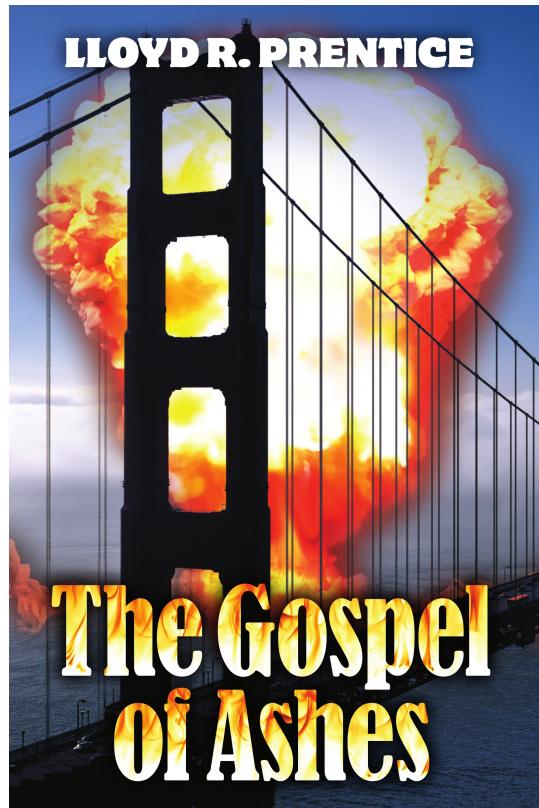
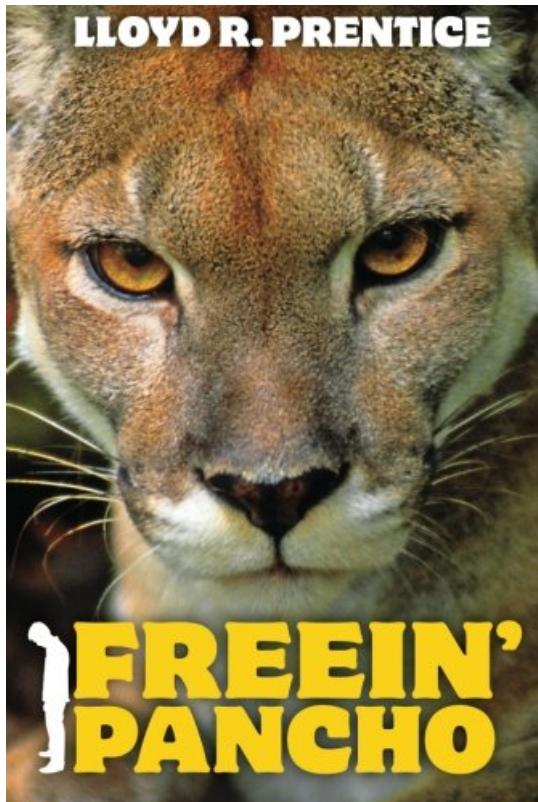
A horizontal row of six icons. From left to right: a pencil, a quill pen, a fountain pen, a checkmark, another checkmark, and an 'X' mark.

A decorative horizontal separator consisting of various stylized floral and geometric symbols, including flowers, crosses, and arrows.

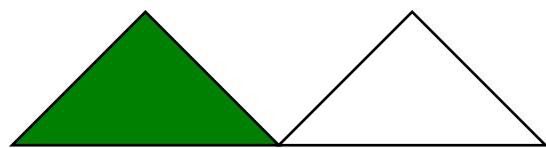
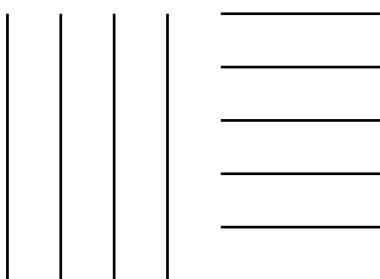
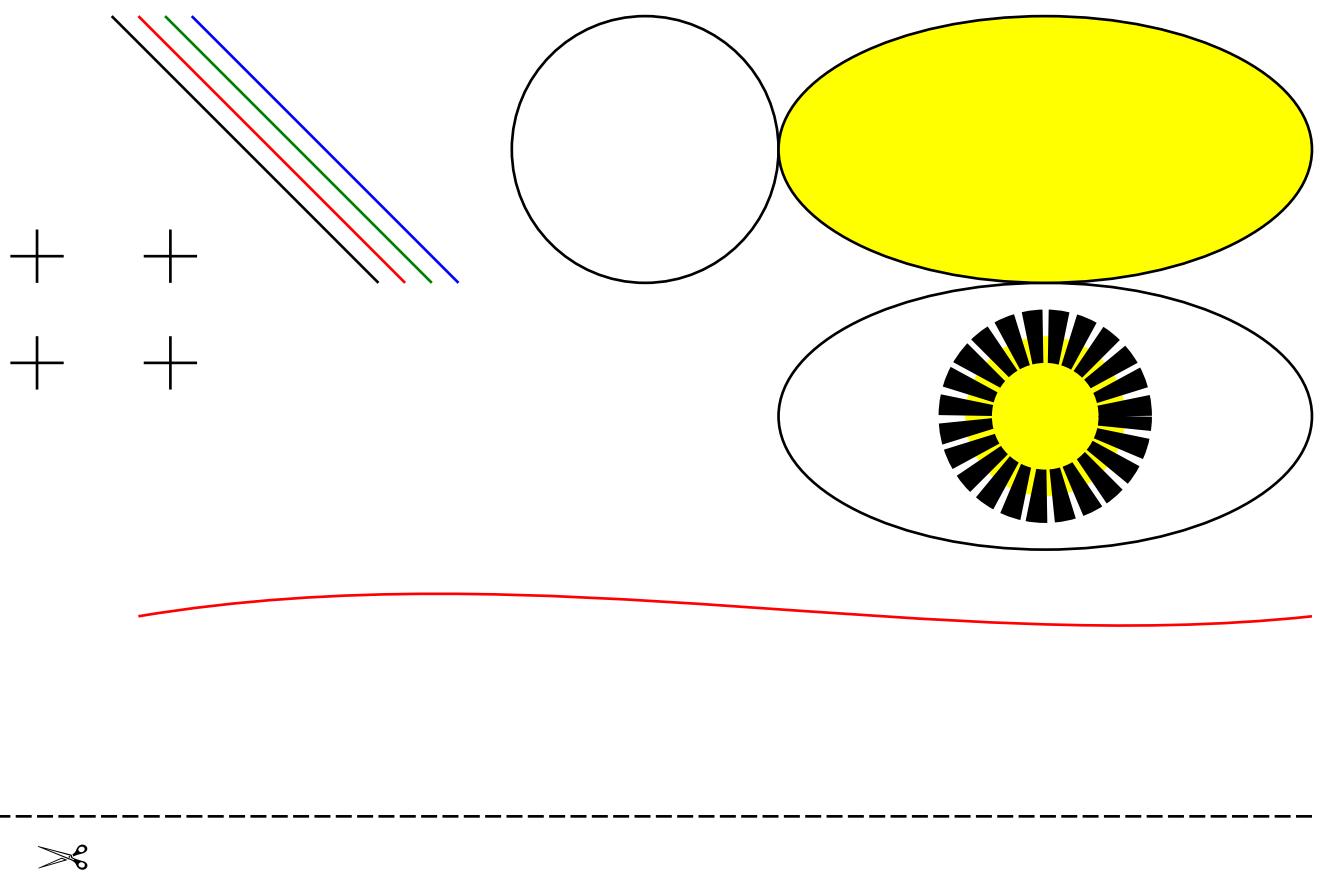
# Colors



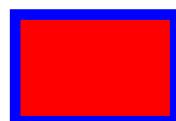
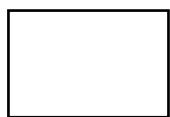
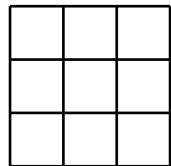
# Images



# Lines and Shapes



# Dots, Boxes, and Grids



# Oh, yes...

## Ordered Lists

This is a test of an **ordered list** with a bold phrase.

1. Display Adobe Type I fonts
2. Position and copyfit text
3. Justify, kern, and rotate text
4. Specify and position images and graphic elements
5. Support checklists

More copy

## Code

### **Listing: cdoc test**

```
run()->
  OutFile = "Test",
  PDF = eg_pdf:new(),
  PanelNameF = {1, 3, code},
  CopyF = ep_sample_text:code_listing(),
  PositionF = {72, 530},
  SizeF = {210, 150},
  PanelMapF1 = ep_panel:create(PanelNameF, PositionF, SizeF),
  PanelMapF2 = ep_panel:reveal(PanelMapF1),
  PanelMapF3 = ep_panel:update_typestyle(cdoc, PanelMapF2),
  {PasteF, _GapF, _PanelMapF4} = ep_copy_block:text(CopyF, PanelMapF3),
  ok = ep_paste_lib:paste_panel(PDF, Job, PanelMapF3),
  ok = ep_paste_lib:paste(PDF, PasteF, [], PanelMapF3),
  ep_job:save_job(PDF, OutFile).
```

# Layout Grid

Paper Stock: letter

Th-th-th-that's all folks!

Profound thanks to...

- Joe Armstrong
- Carl Wright
- Hugh Watkins
- Patrice Bruno
- Frank Muller
- Jesse Gumm

...and all who have made Erlang great.

