



# pyspread for presentations under Linux

Is it usable and efficient?

*Martin Manns 2015*

# Spreadsheets are like ducks

They can store data - but not as good as a data base.

They can do calculations - but not as good as scientific toolkits.

They can visualize data - but not as good as presentation software.

Spreadsheets are good for getting **good-enough results fast.**



*Image: Mike Baird,  
Morro Bay, USA*

# Why pyspread for presentations?

---

3

- A grid allows keeping a clear structure for all slides.
- Pyspread can display vector images, bitmaps and charts.
- No automagically altered font sizes
- Easy export of multiple grid sheets as PDF pages.

**But pyspread only makes sense for presentations if it is easier, faster or gets better results than existing solutions.**

# Competition



LibreOffice / OpenOffice.org Impress



Calligra Stage / KPresenter



Inkscape & JessyInk



Inkscape & Sozi

... and of course a well-known Windows program that is running under wine.

*Solutions without graphical slide builders that compile text or code into slides (e.g. LaTeX, Bruce or MagicPoint) are omitted here.*

# Slide templates: Creation and application

5

- 1) On table 0, create a template layout (~15 rows, ~10 columns).
- 2) Load the following script as a macro:

```
def rowcol_from_template(target_tab, template_tab=0):
    """Adjusts row heights and column widths to match the template

    Parameters
    -----
    target_tab: Integer
        \tTable to be adjusted
    template_tab: Integer, defaults to 0
        \tTemplate table
    """

    for row, tab in S.row_heights.keys():
        # Delete all row heights in target table
        if tab == target_tab:
            S.row_heights.pop((row, tab))

        if tab == template_tab:
            S.row_heights[(row, target_tab)] = \
                S.row_heights[(row, tab)]

    for col, tab in S.col_widths.keys():
        # Delete all column widths in target table
        if tab == target_tab:
            S.col_widths.pop((col, tab))

        if tab == template_tab:
            S.col_widths[(col, target_tab)] = \
                S.col_widths[(col, tab)]

    return "Table {tab} adjusted.".format(tab=target_tab)
```

```
def cell_attributes_from_template(target_tab, template_tab=0):
    """Adjusts cell parameters to match the template

    Parameters
    -----
    target_tab: Integer
        \tTable to be adjusted
    template_tab: Integer, defaults to 0
        \tTemplate table
    """

    new_cell_attributes = []
    for attr in S.cell_attributes:
        if attr[1] == template_tab:
            new_attr = (attr[0], target_tab, attr[2])
            new_cell_attributes.append(new_attr)
    S.cell_attributes.extend(new_cell_attributes)

    return "Table {tab} adjusted.".format(tab=target_tab)

# Shortcuts
rc = rowcol_from_template
ca = cell_attributes_from_template
```

- 3) In each table, execute ca(Z) and rc(Z) to apply template.

# Slide content - the fastest way in pyspread

6

Enter text into a cell followed by  
<Ctrl> + <Enter>.

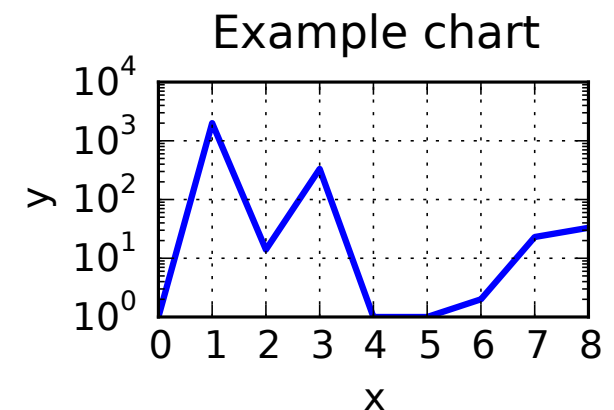
Copy bitmaps from the clipboard.



Import SVG graphics via the import macro.

Create charts via the chart dialog.

Merge cells that are too small.



# Presentation export

---

Export a PDF via the export button.

In the PDF export dialog start from sheet 1 and end with your last slide.

View the result with a PDF viewer of your choice.

# Light and shadow

- Slide layout easy and fast
  - High quality PDF export
  - Slide layout follows template
  - Integration of high quality charts
- 
- Cell merging required in most cases
  - In-cell formatting only via markup
  - No spell checker
  - No animation or video support



*Images: Kreuzschnabel (top),  
Tom Bayly, England (bottom)*



# pyspread for presentations

---

9

This presentation has been created with pyspread v0.4.3.

Pyspread is usable. No crashes so far.

It is faster to create long presentations than JessyInk or Sozi.



**Pyspread is free.  
Just try it out.**