The scientific editor TEXMACS

Warning: This file contains dynamic content which will be lost in PDF format.
Please use TEXMACS to view the original file

Miguel de Benito

Plan

- Intro.
- Quick demo: math, tables, drawings, scripting.
- Plugins and sessions.
- Collaboration.
- Extending TEXMACS.

Plan

- Intro.
- Quick demo: math, tables, drawings, scripting.
- Plugins and sessions.
- Collaboration.
- Extending TEXMACS.

Plan

- Intro.
- Quick demo: math, tables, drawings, scripting.
- Plugins and sessions.
- Collaboration.
- Extending TeXmacs.

Plan

- Intro.
- Quick demo: math, tables, drawings, scripting.
- Plugins and sessions.
- Collaboration.
- Extending TEXMACS.

Plan

- Intro.
- Quick demo: math, tables, drawings, scripting.
- Plugins and sessions.
- Collaboration.
- Extending TEXMACS.

- What it is
- What it isn't

- **What it is**
  - Truly WYSIWYG scientific editing and typesetting platform. **Structured** editor.

- **What it isn't**

- **What it is**
  - Truly WYSIWYG scientific editing and typesetting platform. **Structured** editor.
  - Open source, GNU project. All major platforms.

- **What it isn't**

- **What it is**
  - Truly WYSIWYG scientific editing and typesetting platform. **Structured** editor.
  - Open source, GNU project. All major platforms.
  - Fully extensible using SCHEME.

- **What it isn't**

- **What it is**
  - Truly WYSIWYG scientific editing and typesetting platform. **Structured** editor.
  - Open source, GNU project. All major platforms.
  - Fully extensible using SCHEME.
  - Mainly C++ and SCHEME with mature codebase.

- **What it isn't**

- **What it is**
  - Truly WYSIWYG scientific editing and typesetting platform. **Structured** editor.
  - Open source, GNU project. All major platforms.
  - Fully extensible using SCHEME.
  - Mainly C++ and SCHEME with mature codebase.
  - Small team (4-8 active). Lead developer: Joris van der Hoeven, CNRS.

- **What it isn't**

- **What it is**
  - Truly WYSIWYG scientific editing and typesetting platform. **Structured** editor.
  - Open source, GNU project. All major platforms.
  - Fully extensible using SCHEME.
  - Mainly C++ and SCHEME with mature codebase.
  - Small team (4-8 active). Lead developer: Joris van der Hoeven, CNRS.

- **What it isn't**
  - TEX. Nor EMACS.

- **What it is**
  - Truly WYSIWYG scientific editing and typesetting platform. **Structured** editor.
  - Open source, GNU project. All major platforms.
  - Fully extensible using SCHEME.
  - Mainly C++ and SCHEME with mature codebase.
  - Small team (4-8 active). Lead developer: Joris van der Hoeven, CNRS.

- **What it isn't**
  - TEX. Nor EMACS.
  - A frontend to LATEX.

- **What it is**
  - Truly WYSIWYG scientific editing and typesetting platform. **Structured** editor.
  - Open source, GNU project. All major platforms.
  - Fully extensible using SCHEME.
  - Mainly C++ and SCHEME with mature codebase.
  - Small team (4-8 active). Lead developer: Joris van der Hoeven, CNRS.

- **What it isn't**
  - TEX. Nor EMACS.
  - A frontend to LATEX.
  - A programming language.

- **What it is**
  - Truly WYSIWYG scientific editing and typesetting platform. **Structured** editor.
  - Open source, GNU project. All major platforms.
  - Fully extensible using SCHEME.
  - Mainly C++ and SCHEME with mature codebase.
  - Small team (4-8 active). Lead developer: Joris van der Hoeven, CNRS.

- **What it isn't**
  - TEX. Nor EMACS.
  - A frontend to LATEX.
  - A programming language.
  - Your kitchen robot.

- **Beautiful math**

- **Beautiful math**

$$|\mathrm{e}^{tA_e}| \leqslant \mathrm{e}^{-t/\varepsilon} \sum_{n=0}^{\infty} \left(\frac{t}{\varepsilon}\right)^n \frac{1}{n!} \gamma^{n\varepsilon+1} = \gamma \exp\left\{\frac{t}{\varepsilon}(\gamma^{\varepsilon} - 1)\right\}.$$

- **Beautiful math**

$$|\mathrm{e}^{tA_\varepsilon}| \leqslant \mathrm{e}^{-t/\varepsilon} \sum_{n=0}^{\infty} \left(\frac{t}{\varepsilon}\right)^n \frac{1}{n!}\, \gamma^{n\varepsilon+1} = \gamma \exp\left\{\frac{t}{\varepsilon}\left(\gamma^\varepsilon - 1\right)\right\}.$$

- **Fast input**

  LaTeX input emulation **but(!)** intuitive shortcuts, (structured) variants.

- **Beautiful math**

$$|\mathrm{e}^{tA_\varepsilon}| \leqslant \mathrm{e}^{-t/\varepsilon} \sum_{n=0}^{\infty} \left(\frac{t}{\varepsilon}\right)^n \frac{1}{n!} \gamma^{n\varepsilon+1} = \gamma \exp\left\{\frac{t}{\varepsilon}(\gamma^\varepsilon - 1)\right\}.$$

- **Fast input**

  LaTeX input emulation **but(!)** intuitive shortcuts, (structured) variants.

- **Semantic editing**

  Validation, manipulation, conversion, interfacing.

- **Tables are easy**

- **Tables are easy**

a	b	c
c	d	

- **Tables are easy**

a	b	c
c	d	

- **And powerful**

- **Tables are easy**

a	b	c
c	d	

- **And powerful**

Tomaten, 1Kg	5
Bananen, 2Kg	6
Kekse, 1Pk	2
	=b1+b2+b3

Table 1. A shopping list.

$\sin(4\,x^2)$	$\cos(4\,y^2)$
$=\text{diff}(a1, x)$	$=\text{diff}(b1, y)$

Table 2. More computations.

- **Tables are easy**

a	b	c
c	d	

- **And powerful**

Tomaten, 1Kg	5
Bananen, 2Kg	6
Kekse, 1Pk	2
	13

Table 1. A shopping list.

$\sin(4\,x^2)$	$\cos(4\,y^2)$
$=\mathrm{diff}(a1, x)$	$=\mathrm{diff}(b1, y)$

Table 2. More computations.

- **Tables are easy**

a	b	c
c	d	

- **And powerful**

Tomaten, 1Kg	5
Bananen, 2Kg	6
Kekse, 1Pk	2
	13

Table 1. A shopping list.

$\sin(4\,x^2)$	$\cos(4\,y^2)$
$8\,x\cos(4\,x^2)$	$=\mathrm{diff}(b1, y)$

Table 2. More computations.

- **Tables are easy**

a	b	c
c	d	

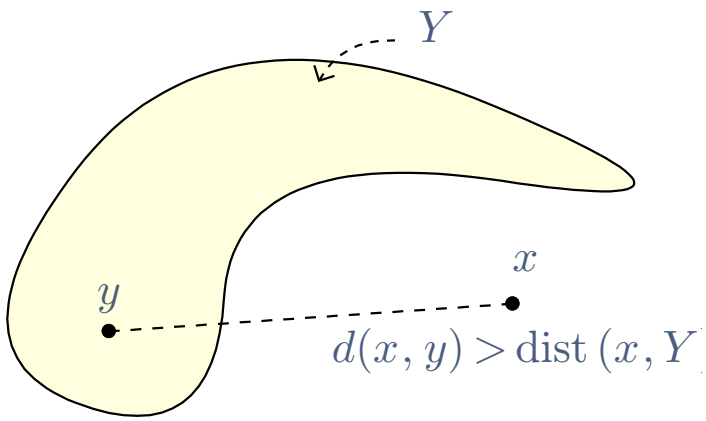
- **And powerful**

Tomaten, 1Kg	5
Bananen, 2Kg	6
Kekse, 1Pk	2
	13

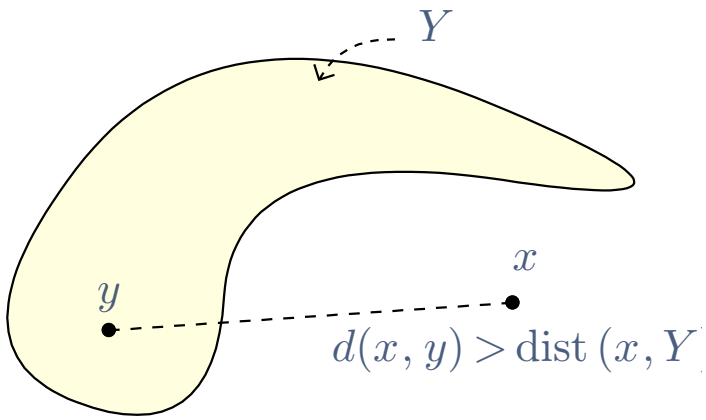
Table 1. A shopping list.

$\sin(4\,x^2)$	$\cos(4\,y^2)$
$8\,x\cos\left(4\,x^2\right)$	$-8\,y\sin\left(4\,y^2\right)$

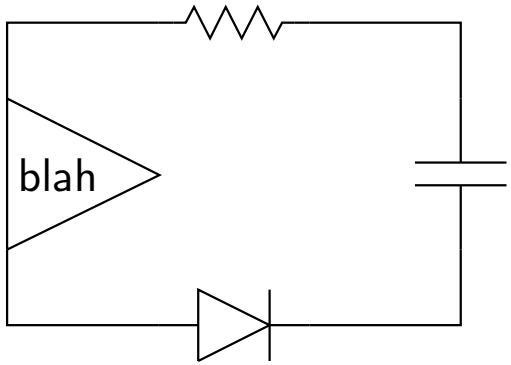
Table 2. More computations.



Simple vector graphics.



Simple vector graphics.



User-defined graphical macros.

- **Scripting**

- **Scripting**

  An example with MAXIMA:

  Let $p(x) = x^2 - 9$ and $q(x) = x^2 + 6\,x + 9$. Integrate:

$$\int \frac{p(x)}{q(x)}\,\mathrm{d}x = \mathrm{integrate}(p(x)\,/\,q(x), x) + C.$$

- **Scripting**

  An example with MAXIMA:

  Let $p(x) = x^2 - 9$ and $q(x) = x^2 + 6\,x + 9$. Integrate:

$$\int \frac{p(x)}{q(x)}\,\mathrm{d}x = x - 6\log\left(x + 3\right) + C.$$

- **Scripting**

  An example with MAXIMA:

  Let $p(x) = x^2 - 9$ and $q(x) = x^2 + 6\,x + 9$. Integrate:

$$\int \frac{p(x)}{q(x)}\,\mathrm{d}x = x - 6\log\left(x + 3\right) + C.$$

- **Embedded sessions** (later)

- **Scripting**

  An example with MAXIMA:

  Let $p(x) = x^2 - 9$ and $q(x) = x^2 + 6\,x + 9$. Integrate:

$$\int \frac{p(x)}{q(x)}\,\mathrm{d}x = x - 6\log\left(x+3\right) + C.$$

- **Embedded sessions** (later)

- **Many plugins**

  ASYMPTOTE, AXIOM, CADABRA, COQ, EUKLEIDES, GHOSTSCRIPT, GIAC, GNUPLOT, GTYBALT, MACAULAY2, MATLAB, MAXIMA, OCTAVE, PARI, PYTHON, QCL, R, REDUCE, SAGE, SCILAB, TEXGRAPH, XFIG, YACAS and more...

- **Native converters**

  PDF, XML, HTML+MathML, LaTeX.

- **Native converters**

  PDF, XML, HTML+MathML, LaTeX.

- **Styles with macro language**

  Functional and powerful! (more later)

- **Native converters**

  PDF, XML, HTML+MathML, LaTeX.

- **Styles with macro language**

  Functional and powerful! (more later)

- **Bibliography**

  BibTeX support, custom styles, support for internal databases.

- **Native converters**

  PDF, XML, HTML+MathML, LaTeX.

- **Styles with macro language**

  Functional and powerful! (more later)

- **Bibliography**

  BibTeX support, custom styles, support for internal databases.

- **Beamer presentations**

  Live demos, live fixes!

There's much more!

- **Native converters**

  PDF, XML, HTML+MathML, LaTeX.

- **Styles with macro language**

  Functional and powerful! (more later)

- **Bibliography**

  BibTeX support, custom styles, support for internal databases.

- **Beamer presentations**

  Live demos, live fixes!

- **Spell checking**

  Uses standard aSpell.

There's much more!

- **Native converters**

  PDF, XML, HTML+MathML, LaTeX.

- **Styles with macro language**

  Functional and powerful! (more later)

- **Bibliography**

  BibTeX support, custom styles, support for internal databases.

- **Beamer presentations**

  Live demos, live fixes!

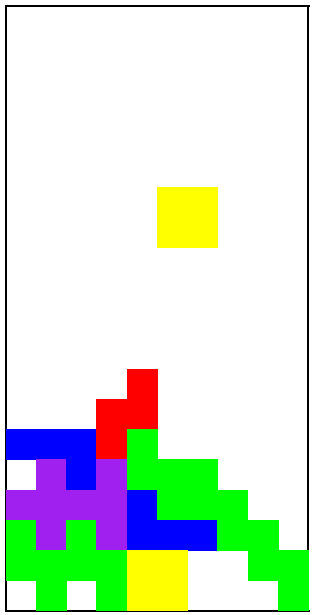
- **Spell checking**

  Uses standard aSpell.

-

Bonus: Tables abused


```
>> (load (url->string (url-append (url-head (buffer-master)) "t-mockup.scm")))
>> (start-game)
```

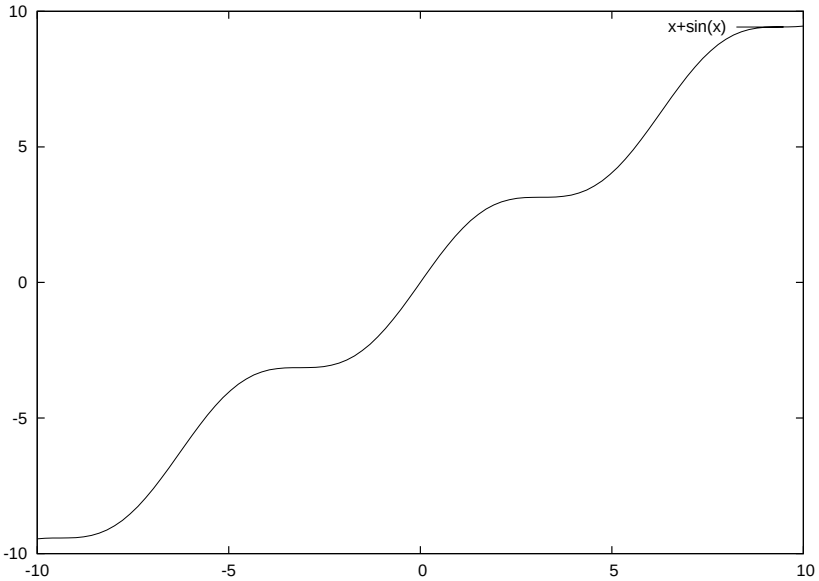


Sessions

```
GNUplot] plot [-10:10][-10:10] x+sin(x)
GNUplot]
```

Sessions

```
GNUplot] plot [-10:10][-10:10] x+sin(x)
```




```
GNUplot]
```

Easy graphs

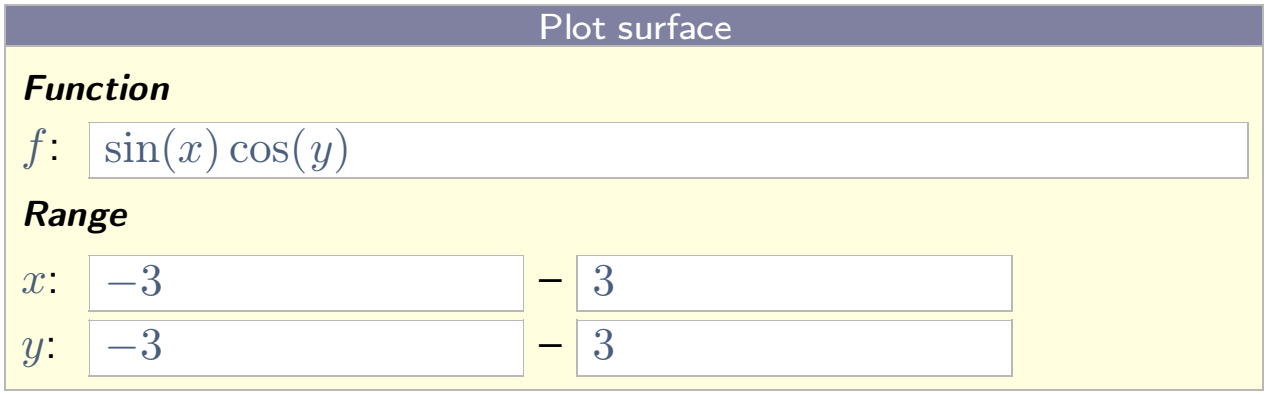


Figure. A simple surface plot.

A SCILAB session:

```
--> A = [0, 1; 0, 0]; B = [1 ; 1]; C = [1, 1];
--> S1 = syslin ('c', A, B, C)
--> x= -6.28:0.1:6.28; y= sin(x); plot (x, y);
-->
```

A SCILAB session:

```
--> A = [0, 1; 0, 0]; B = [1 ; 1]; C = [1, 1];

--> S1 = syslin ('c', A, B, C)
--> x= -6.28:0.1:6.28; y= sin(x); plot (x, y);
-->
```

A SCILAB session:

```
--> A = [0, 1; 0, 0]; B = [1 ; 1]; C = [1, 1];


--> S1 = syslin ('c', A, B, C)
```

S1 =

$$\begin{cases} \dot{X}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} X(t) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} U(t) \\ Y(t) = \begin{pmatrix} 1 & 1 \end{pmatrix} X(t) \end{cases}$$

```
--> x= -6.28:0.1:6.28; y= sin(x); plot (x, y);
-->
```

A SCILAB session:

```
--> A = [0, 1; 0, 0]; B = [1 ; 1]; C = [1, 1];


--> S1 = syslin ('c', A, B, C)
```

S1 =

$$\begin{cases} \dot{X}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} X(t) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} U(t) \\ Y(t) = \begin{pmatrix} 1 & 1 \end{pmatrix} X(t) \end{cases}$$

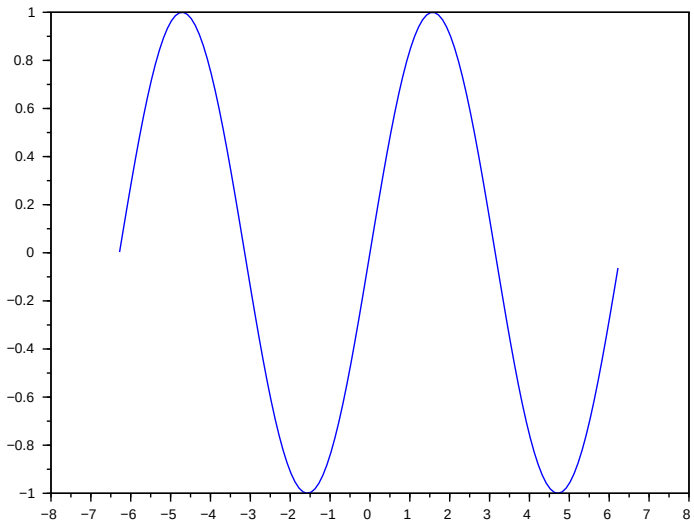
```
--> x= -6.28:0.1:6.28; y= sin(x); plot (x, y);


-->
```

A SCILAB session:

```
--> plotout();
-->
```

A SCILAB session:

```
--> plotout();
```

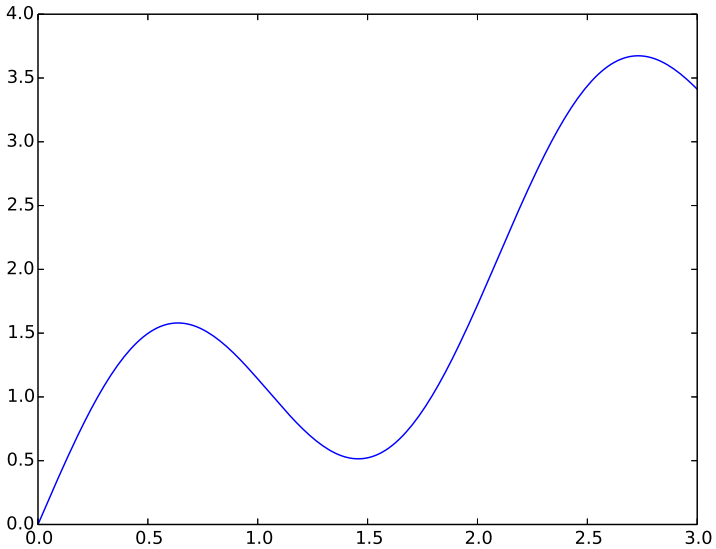



```
-->
```

```
>>> import matplotlib as mpl
    mpl.use('PS')
    import matplotlib.pyplot as pl
    import numpy as np
    x = np.linspace(0,3,200)
    pl.plot(x, x + np.sin(3*x))
    fig = pl.gcf()
>>>
```

```
>>> ps_out(fig)
>>>
```

```
>>> ps_out(fig)
```




```
>>>
```

- **A live figure**

Python

```
pl.plot(x, x + np.sin(pow(x, 5)))
ps_out(fig)
```

- **A live figure**

  Busy...

- **A live figure**

  Busy...

- **A real example**

  Gaussian Mixtures and Expectation Maximization

- **Embedded** computations.

- **Embedded** computations.

- **Remote** computations.

  (not discussed here)

- **Embedded** computations.

- **Remote** computations.

  (not discussed here)

- **Embedded** graphics.

- **Embedded** computations.

- **Remote** computations.

  (not discussed here)

- **Embedded** graphics.

- **Live** documents.

- **Embedded** computations.
- **Remote** computations.

  (not discussed here)
- **Embedded** graphics.
- **Live** documents.
- **Easy** to extend.

- LaTeX

  Conservative conversions.

- **LaTeX**

  Conservative conversions.

- **Versioning**

  Tree diff better than line diff. Also: SVN support (GIT available but not integrated yet).

- **LaTeX**

  Conservative conversions.

- **Versioning**

  Tree diff better than line diff. Also: SVN support (GIT available but not integrated yet).

- **Remote documents**

  TeXmacs server.

- **LaTeX**

  Conservative conversions.

- **Versioning**

  Tree diff better than line diff. Also: SVN support (GIT available but not integrated yet).

- **Remote documents**

  TeXmacs server.

- **Concurrent editing**

  Currently under development.

- **Preferences & shortcuts**

  Through UI and config files.

- **Preferences & shortcuts**

  Through UI and config files.

- **Stylesheet language**

  Macros, control structures, variables, dynamic features.

- **Preferences & shortcuts**

  Through UI and config files.

- **Stylesheet language**

  Macros, control structures, variables, dynamic features.

- **SCHEME**

  Coming up next.

- **Embedded SCHEME**:

  Currently GUILE 1.8. Help needed for 2.0!

- **Embedded SCHEME**:

  Currently GUILE 1.8. Help needed for 2.0!

- **Why?**

  Any sufficiently complicated C or Fortran program contains an ad hoc, informally-specified, bug-ridden, slow implementation of half of Common Lisp. (Greenspun's tenth rule)

- **Embedded SCHEME**:

  Currently GUILE 1.8. Help needed for 2.0!

- **Why?**

  > *Any sufficiently complicated C or Fortran program contains an ad hoc, informally-specified, bug-ridden, slow implementation of half of Common Lisp. (Greenspun's tenth rule)*

- **But... why?!**
  - Code is data is code if fed to the evaluator.
  - Easy implementation of (micro) DSLs: menus, widgets, graphics, converters, preferences, ...
  - But it's ugly!

- **Embedded SCHEME**:

  Currently GUILE 1.8. Help needed for 2.0!

- **Why?**

  > *Any sufficiently complicated C or Fortran program contains an ad hoc, informally-specified, bug-ridden, slow implementation of half of Common Lisp. (Greenspun's tenth rule)*

- **But... why?!**
  - Code is data is code if fed to the evaluator.
  - Easy implementation of (micro) DSLs: menus, widgets, graphics, converters, preferences, ...
  - But it's ugly! Maybe, but "*consistently so*".

- **Embedded SCHEME**:

  Currently GUILE 1.8. Help needed for 2.0!

- **Why?**

  > *Any sufficiently complicated C or Fortran program contains an ad hoc, informally-specified, bug-ridden, slow implementation of half of Common Lisp. (Greenspun's tenth rule)*

- **But... why?!**
  - Code is data is code if fed to the evaluator.
  - Easy implementation of (micro) DSLs: menus, widgets, graphics, converters, preferences, ...
  - But it's ugly! Maybe, but "*consistently so*".
- **Basic aids**

  Syntax highlighting, code browsing, online help, basic auto-completion.

- **Key bindings**

```
>> (kbd-map
     (:mode in-math?)
     ("I var" (insert '(big "int"))))
>>
```

- **Key bindings**

```
>> (kbd-map
     (:mode in-math?)
     ("I var" (insert '(big "int"))))
```

("I tab")

```
>>
```

- **Key bindings**

```
>> (kbd-map
     (:mode in-math?)
     ("I var" (insert '(big "int"))))
```

("I tab")

```
>>
```

- **Widgets**

```
>> (tm-widget (demo-widget)
     (resize ("100px" "200px" "1000px") ("300px" "600px" "3000px")
       (tree-view (lambda (ev t) (if (== ev 1) (tree-select t)))
                  (buffer-tree) (tree 'dummy))))
```

```
>> (show demo-widget)
```

```
>>
```

- **Key bindings**

```
>> (kbd-map
     (:mode in-math?)
     ("I var" (insert '(big "int"))))
```

("I tab")

```
>>
```

- **Widgets**

```
>> (tm-widget (demo-widget)
     (resize ("100px" "200px" "1000px") ("300px" "600px" "3000px")
       (tree-view (lambda (ev t) (if (== ev 1) (tree-select t)))
                  (buffer-tree) (tree 'dummy))))
```

((guile-user))

```
>> (show demo-widget)
```

```
>>
```

```
>> (select (buffer-tree) '(:* screens shown :%1 tit :%1))
>> (with l (select (buffer-tree) '(:* screens shown :%1 tit :%1))
     (with t (car l)
       (tree-set! t (string->tree "Hi there!"))))
```

```
>> (select (buffer-tree) '(:* screens shown :%1 tit :%1))
```

(<tree More <scheme>>)

```
>> (with l (select (buffer-tree) '(:* screens shown :%1 tit :%1))
     (with t (car l)
       (tree-set! t (string->tree "Hi there!"))))
```

Is this **truly** the state of the art?

```
$$
    B \, \, = \, \, \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix} \, , \quad
    \textbf{x} \, \, = \, \, \begin{pmatrix} a \\ b \end{pmatrix} \, , \quad
    \textbf{h} \, \, = \, \, \begin{pmatrix} 1 \\ 3 \\ 4 \\ 4 \end{pmatrix} \, .
$$
Wir erhalten
$$
    B^T B \, \, = \, \,
    \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix}
    \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}
    \, \, = \, \,
    \begin{pmatrix} 4 & 6 \\ 6 & 14 \end{pmatrix} \, ,
$$
ausserdem existiert $(B^T B)^{-1}$, da $\det(B^T B) = 4 \cdot 14 - 6 \cdot 6 = 20 \neq 0$.
Nach kurzer Rechnung mit Hilfe der zu $B$ komplement\"aren Matrix (bzw. der Formel f\"ur das
Inverse einer $2 \times 2$-Matrix) erhalten wir
$$
    (B^T B)^{-1} \, \, = \, \, \frac{1}{10}
    \begin{pmatrix} 7 & -3 \\ -3 & 2 \end{pmatrix} \, .
$$
```

In 2015 ?!?!

Glad to help

mdbenito@texmacs.org

Office L-2020 @ UNA

Many others too

TEXMACS.org

texmacs-users@texmacs.org **texmacs-dev@gnu.org**