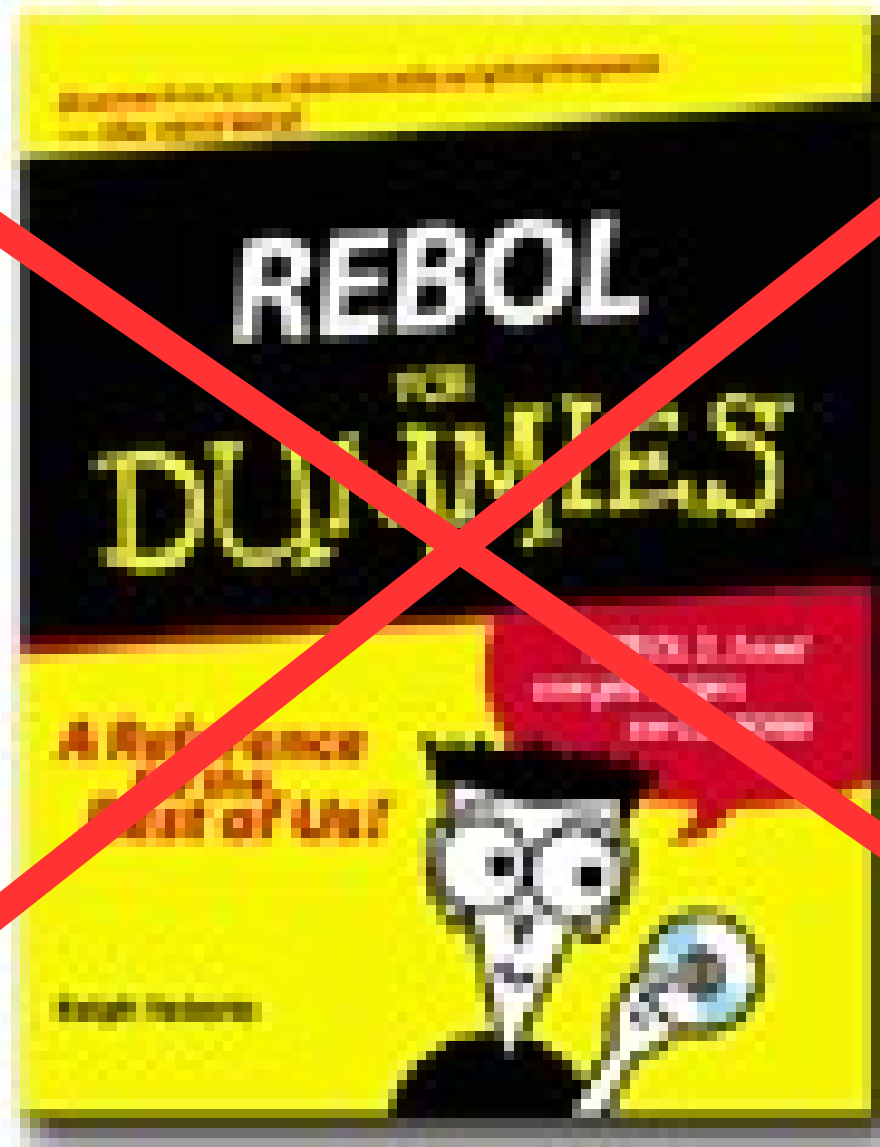


What can Perl learn from Rebol ?

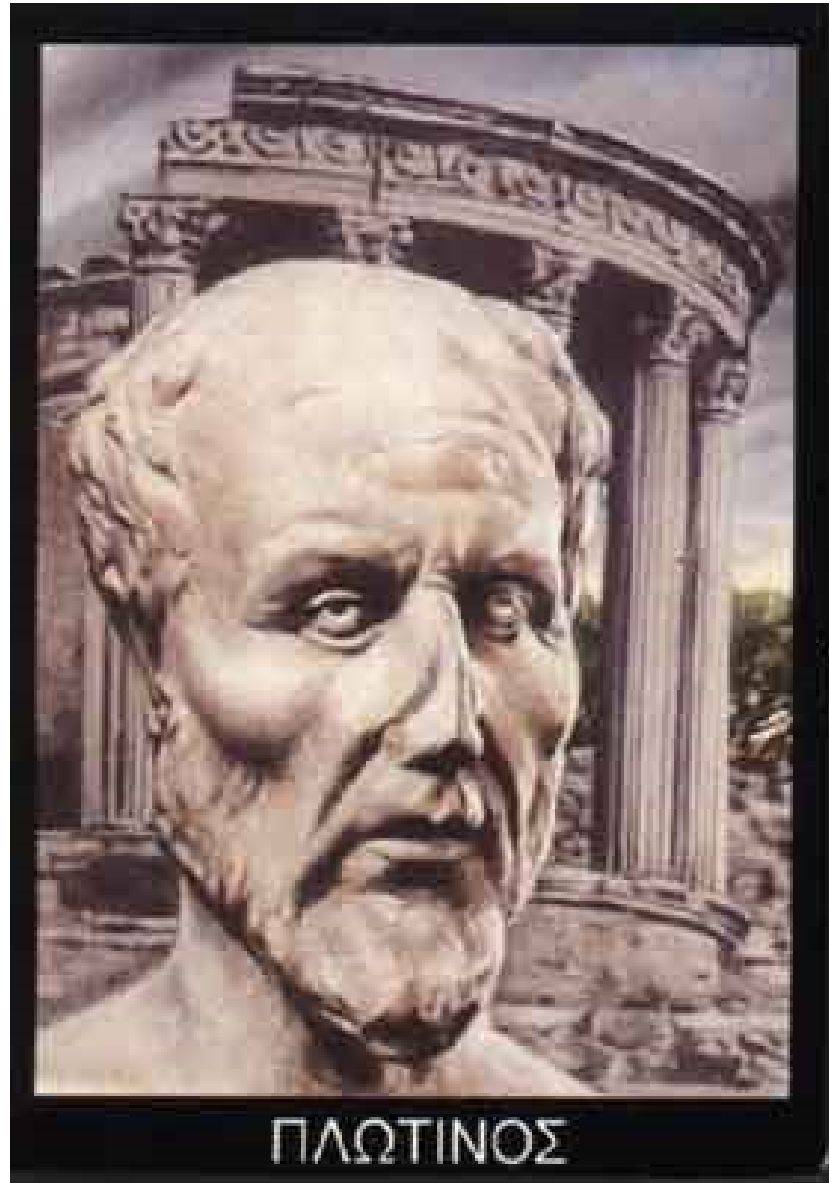
REBOL



No Rebol Course



some hobby philosophy

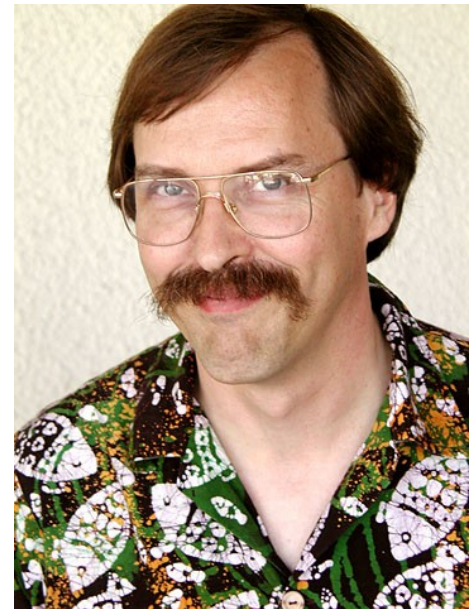


Perl Philosophy

TIMTOWTDI

simple and possible

context sensitive





Philosophy

TIMTOWTDI

simple and possible

context sensitive



Quote

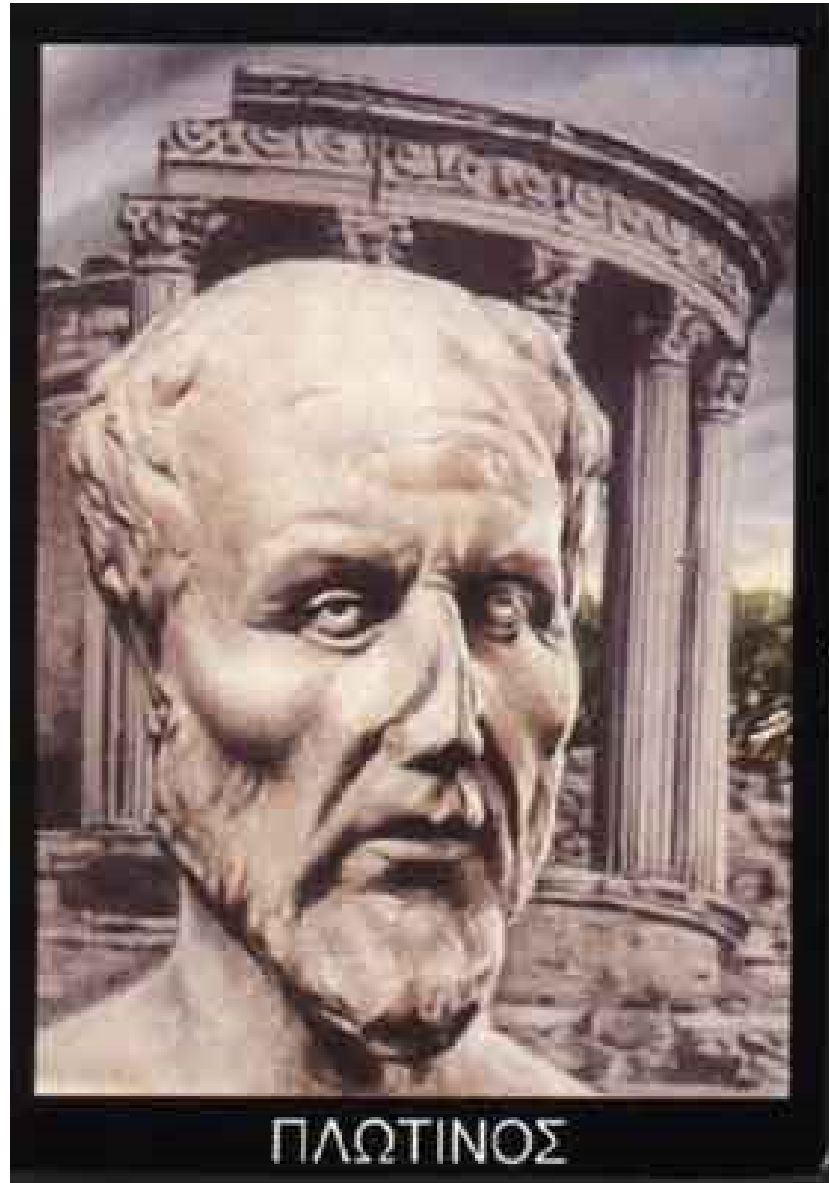


Take a dash of PERL, mix with the cross platform compatibility of a Java, and make it extremely easy for beginners to start coding, and you get the general idea.

A Hobby Philosopher



„Real“ Philosopher



Carl Sassenrath



Carl Sassenrath



- Apple, HP and Commodore

Carl Sassenrath



- Apple, HP and Commodore
- Pantaray, American Multimedia, Videostream

Carl Sassenrath



- Apple, HP and Commodore
- Pantaray, American Multimedia, Videostream
- REBOL Technologies

Carl Sassenrath



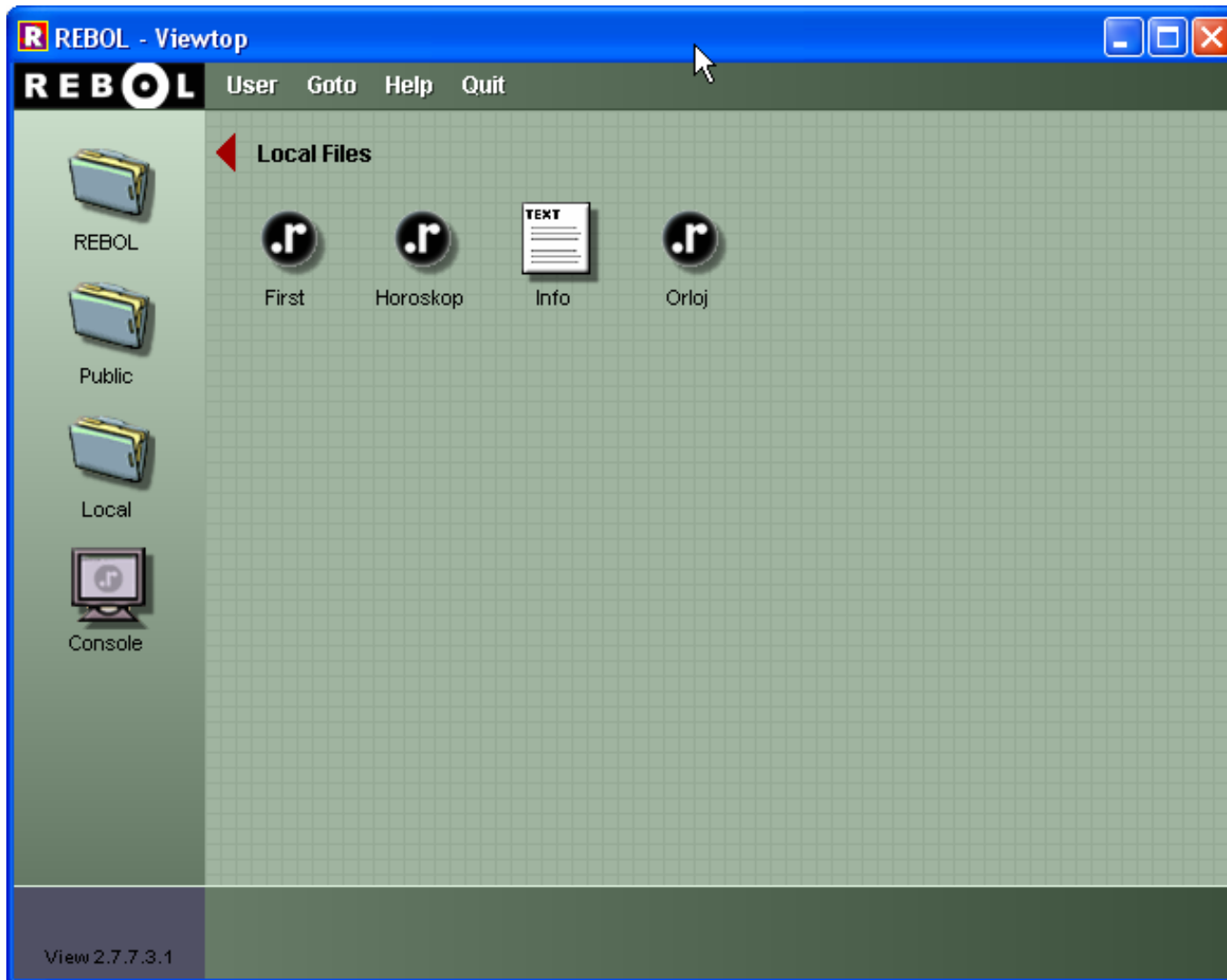
- Amiga OS, IOS, VisCorp ED
- CDTV
- Amiga Logo, Next Gen. Forth, Rebol



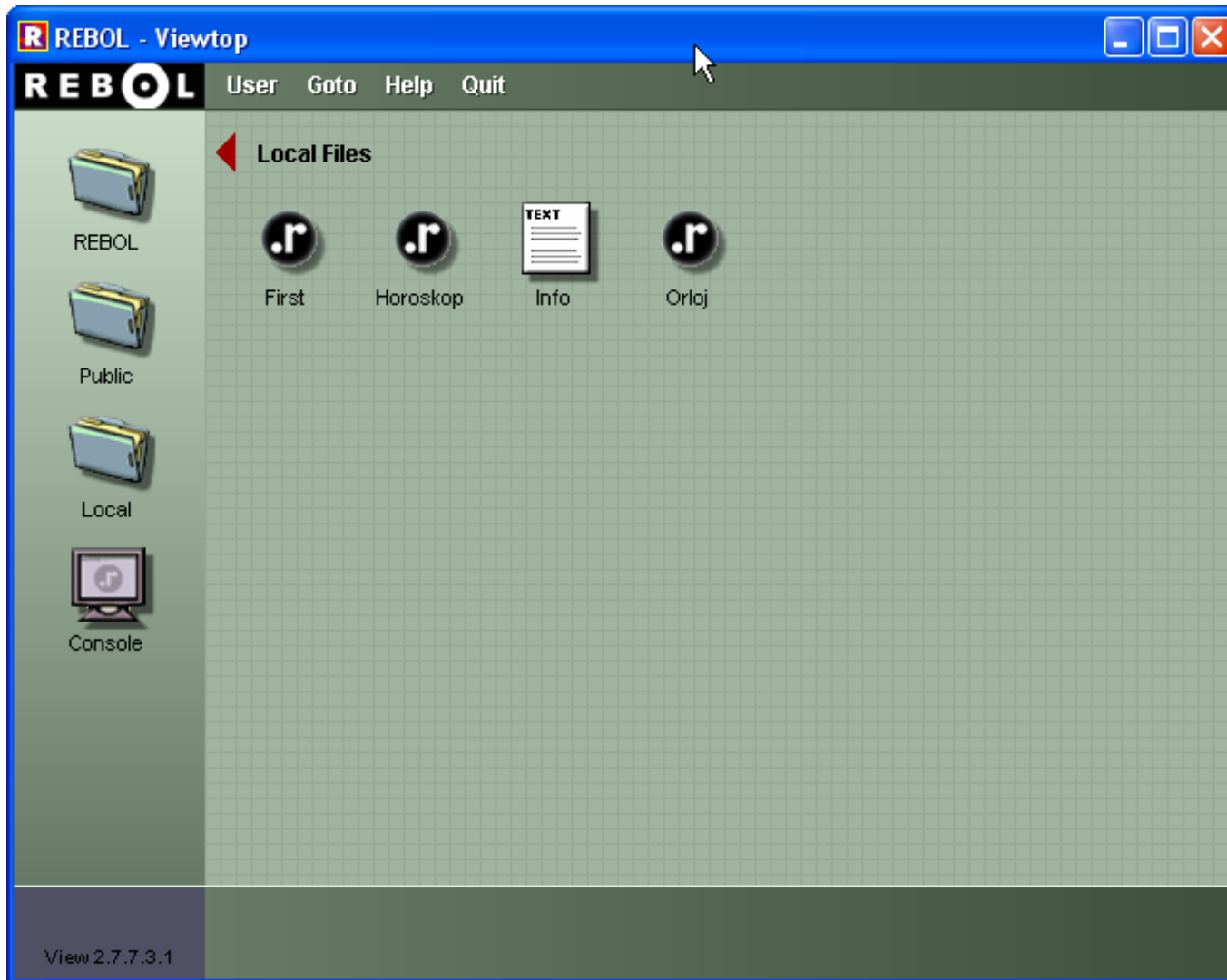
- Distribution



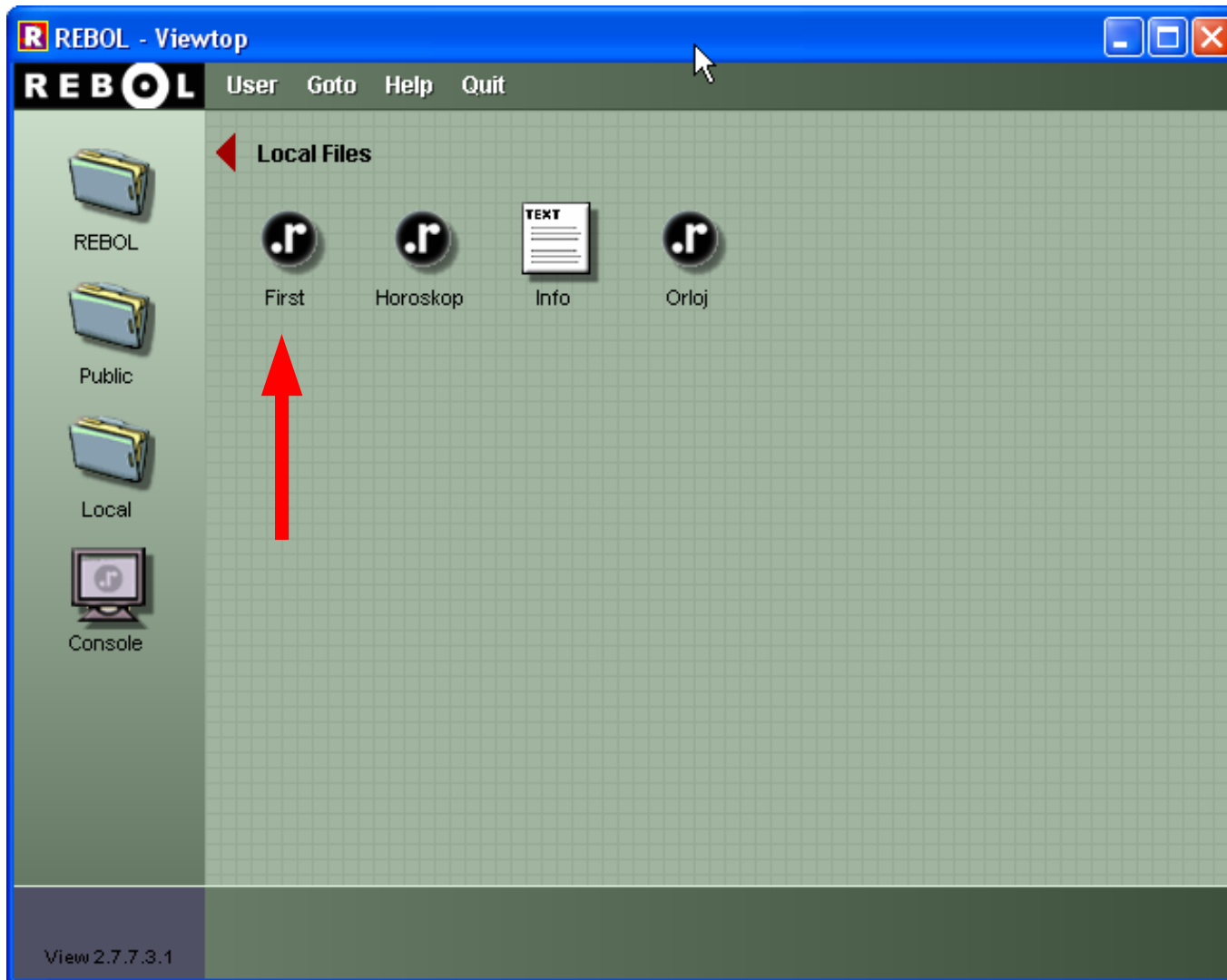
REBOL - Browser



Left Click : Start Rebol App



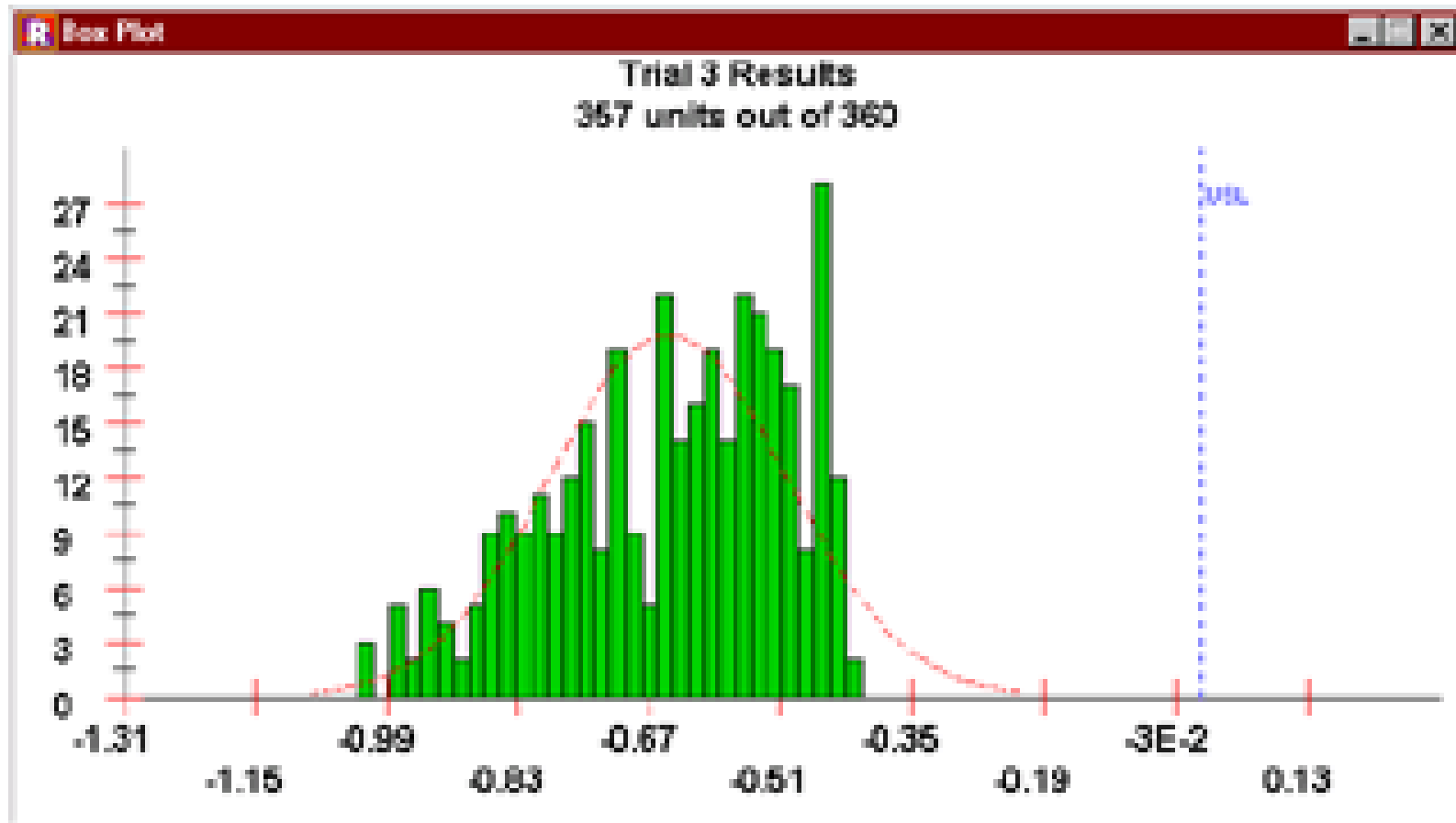
Left Click : Start Rebol App



Rebol App



Rebol App



Rebol App

The screenshot displays the Rebol App interface. At the top, a window titled "Conference - Contra" is open. The main area shows a chat log with the following messages:

Sender	Message	Time
Bob	Just to let you know, we are offering 20% commission per agent on all universal life insurance policies sold this month.	14-Dec-2001 14:26:56
Fred	Bob, are there any limits on the insurability or is it open acceptability?	14-Dec-2001 14:27:19
Bob	We're not sure. There is a meeting scheduled to discuss it. Take a look at the shared calendar and the collaborative agenda has been posted here.	14-Dec-2001 14:27:33
Vanessa	Over here in Europe, a lot of customers are looking for a participating policy that is eligible for the payment of dividends. Any idea when that will be offered?	14-Dec-2001 14:27:57
Bob	We are working on that package, and expect to be offering it to our primary agents and branch offices next week. Watch for a flash message notification on it.	14-Dec-2001 14:28:13
CEO	Insuratron also plans to offer a new marketing program to promote our renewable term insurance. I've updated the general terms spreadsheet in all of your products folders.	14-Dec-2001 14:28:25
Fred	That's great. Our agents expect that could boost our sales by 200% next year.	14-Dec-2001 14:28:41
Bob	Yes, we figured that. Look at the instant-plot figures that the team worked out last week. Enter your own net sales values to see your profits plotted.	14-Dec-2001 14:28:57

At the bottom of the window, there is a control bar with the following elements:

- A vertical "REBOL" logo on the left side.
- Buttons for "Send", "Clear", and "Color".
- A "Branch Agents" button.
- A text input field containing the text: "Click the publish button to share the plot, and I'll go over it with you now".

Rebol App

REBOL Technology

Background
Internet Evolution
Collaborative Market
REBOL IOS
IOS is a Platform
Flora of Apps
Technology
IOS Advantages
Products
Customers
Business Model
Top User Base
Revenue
Competitor
Management
Funding Stages
Conclusion

- IOS is based on an advanced distributed computing language model that was built from the start for Internet communications.

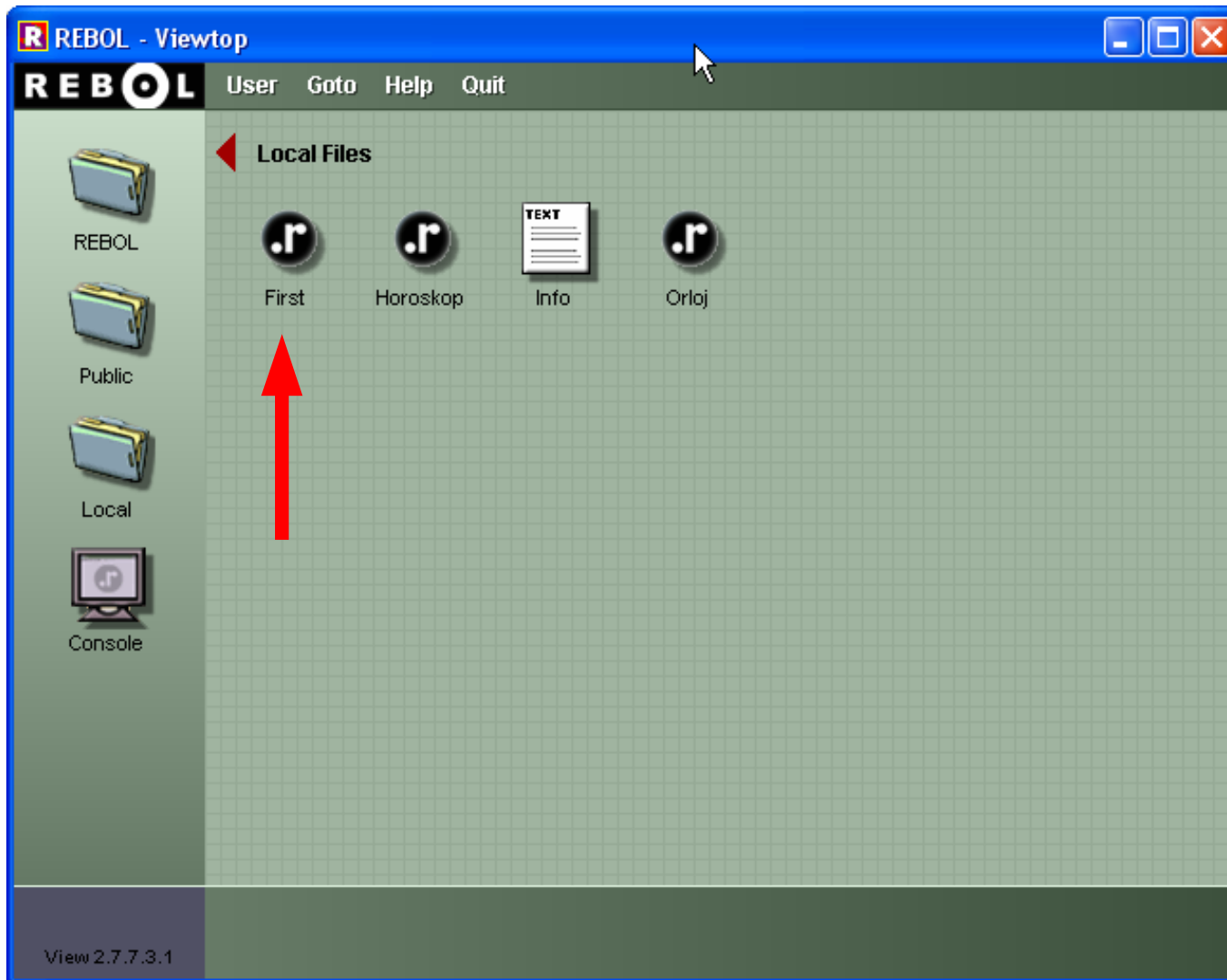
Express	Web Sites	Web Admin	Logging
	App Development	Management	Reliability
	Development	Administration	Configuration

Wire	Server/Client	Mobile
	Local	Virtualization (Cloud)
	Peer-to-Peer	Peer-to-Peer

Connect	Database Access	SQL, API, etc.	API IOS
	Enterprise Apps	Internal Systems	External Apps

Core	Network Protocols	Client Server	Mass Functions
	Web Services	Composites	Interfaces
	Systems	Ports	Resource Models
	Web Integration	Networks	Other Functions
	Network Manager		

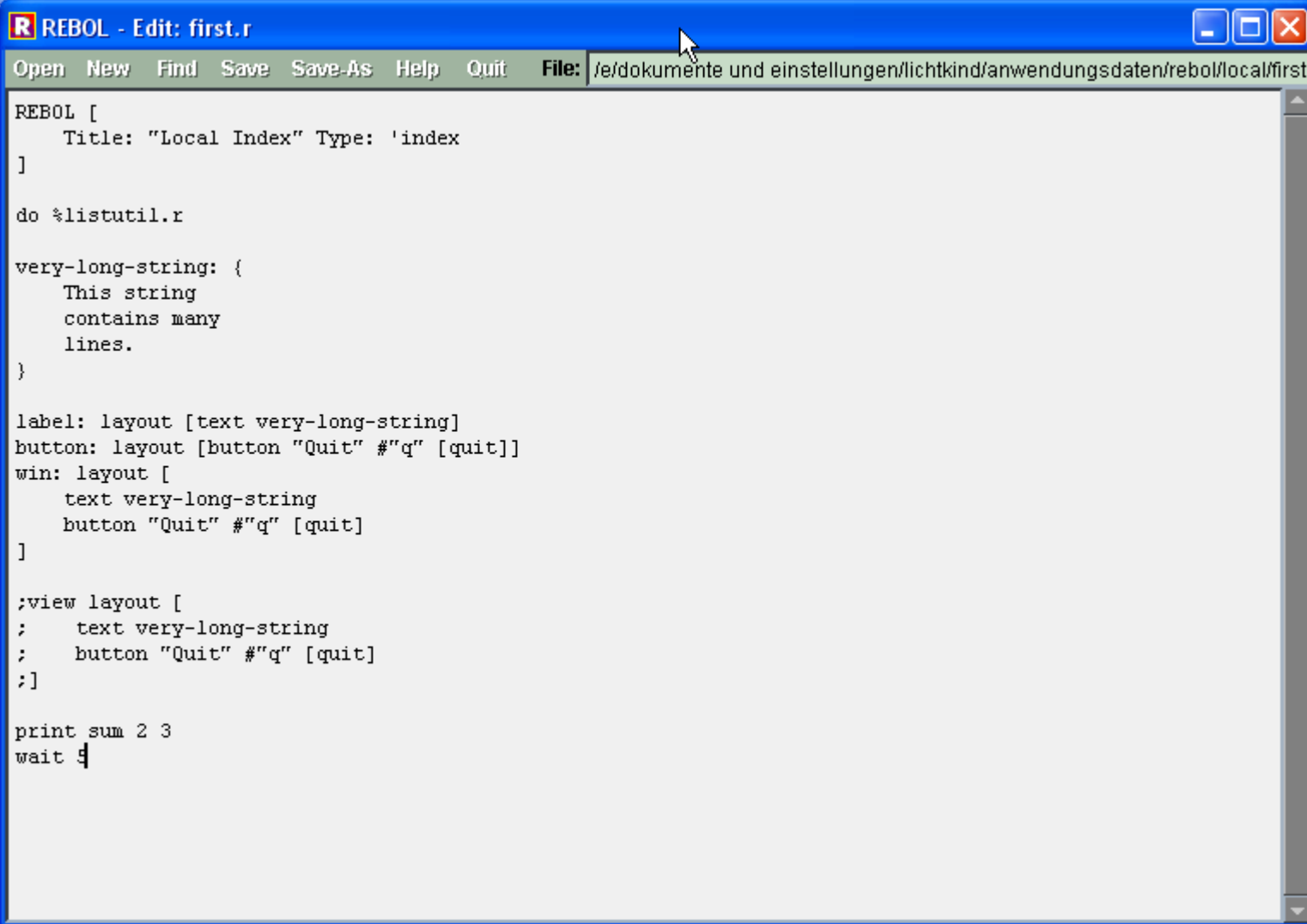
REBOL - Browser



Right Click : Get Meta Data



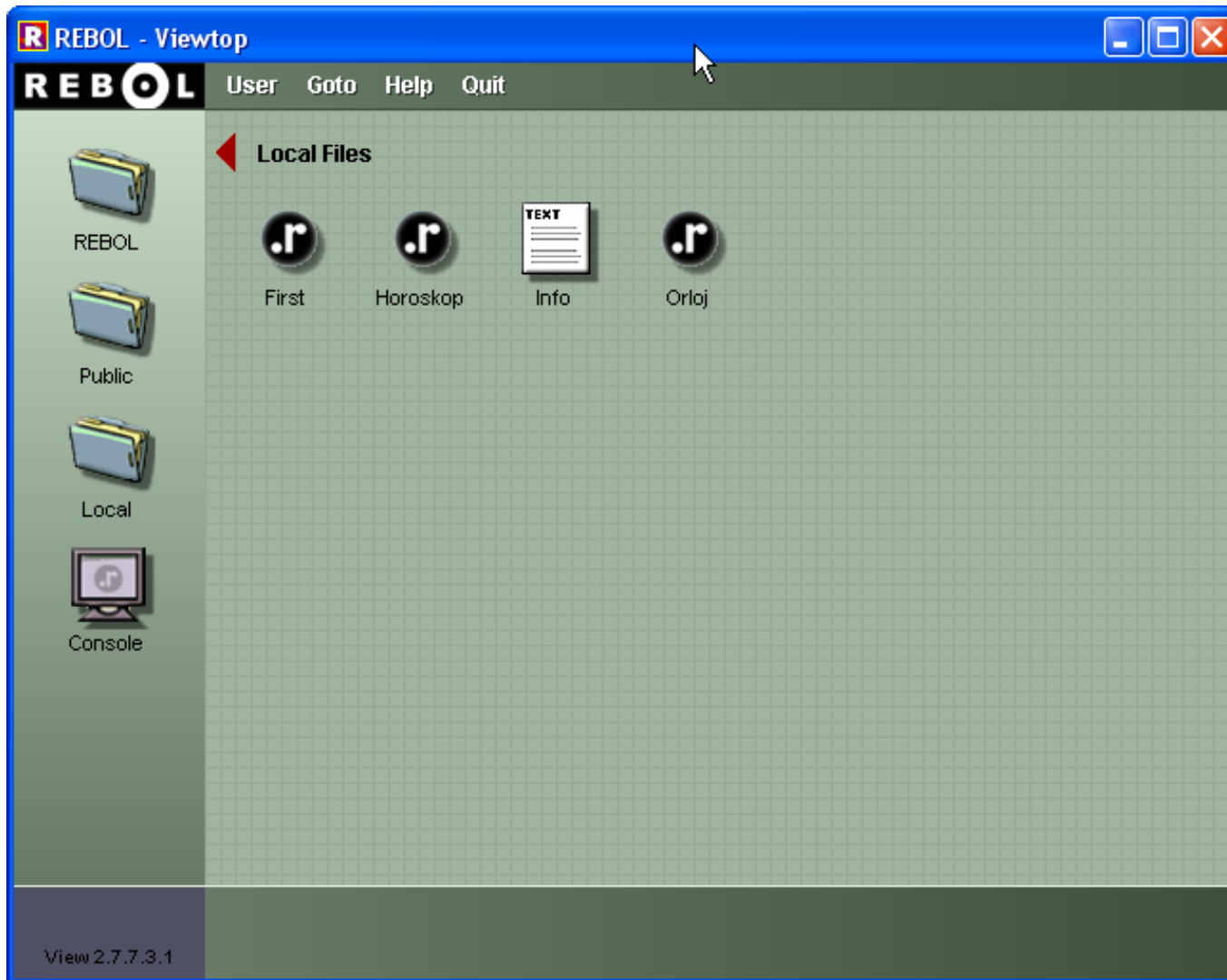
Rebol Editor



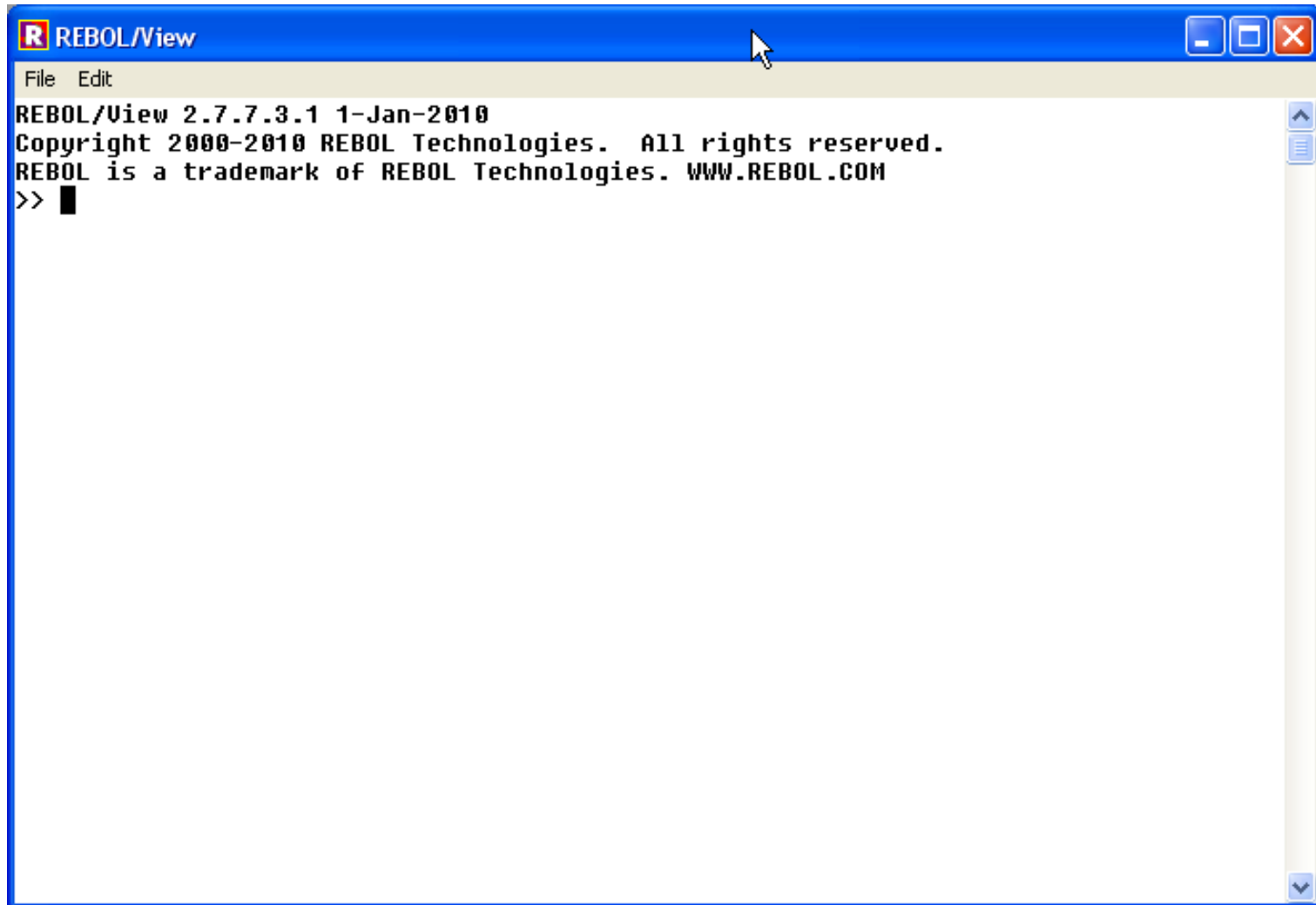
The image shows a screenshot of the Rebol Editor window. The title bar reads "REBOL - Edit: first.r". The menu bar includes "Open", "New", "Find", "Save", "Save-As", "Help", "Quit", and "File". The file path is "/e/dokumente und einstellungen/lichtkind/anwendungsdaten/rebol/local/first". The main text area contains the following Rebol code:

```
REBOL [  
  Title: "Local Index" Type: 'index  
]  
  
do %listutil.r  
  
very-long-string: {  
  This string  
  contains many  
  lines.  
}  
  
label: layout [text very-long-string]  
button: layout [button "Quit" #"q" [quit]]  
win: layout [  
  text very-long-string  
  button "Quit" #"q" [quit]  
]  
  
;view layout [  
;  text very-long-string  
;  button "Quit" #"q" [quit]  
;]  
  
print sum 2 3  
wait 4
```


REBOL - Browser



Rebol Shell

A screenshot of the REBOL/View shell window. The window has a blue title bar with the text "REBOL/View" and a mouse cursor. Below the title bar is a menu bar with "File" and "Edit". The main content area displays the following text:

```
REBOL/View 2.7.7.3.1 1-Jan-2010  
Copyright 2000-2010 REBOL Technologies. All rights reserved.  
REBOL is a trademark of REBOL Technologies. WWW.REBOL.COM  
>> █
```

The window also features standard window control buttons (minimize, maximize, close) in the top right corner and a vertical scrollbar on the right side.

Carl Sassenrath



- Amiga OS, IOS, VisCorp ED
- CDTV
- Amiga Logo, Next Gen. Forth, Rebol

Carl Sassenrath



- Amiga OS, IOS, VisCorp ED
- CDTV
- Amiga Logo, Next Gen. Forth, **Rebol**

REBOL



REBOL



- Lisp, Forth, Logo and Self

REBOL



- Lisp, Forth, Logo and Self
- Lightweight Distributed Computing



- Lisp, Forth, Logo and Self
- Lightweight Distributed Computing
- programming-in-the-small (PITS)

PITS:



~ 800kB binary
1.8 MB for distribution

Libs
Tools
Samples
Docs
Links

PITS:



Remote Editor: 53
Daytime Server Deamon: 61
bmp2png Converter
TCP Port Scanner: 98
Simple Clock App: 106
Minimal Email Client: 130
Simple Picture Viewer: 132

PITS:



Remote Editor: 53

Daytime Server Deamon: 61

bmp2png Converter

TCP Port Scanner: 98

Simple Clock App: 106

Minimal Email Client: 130

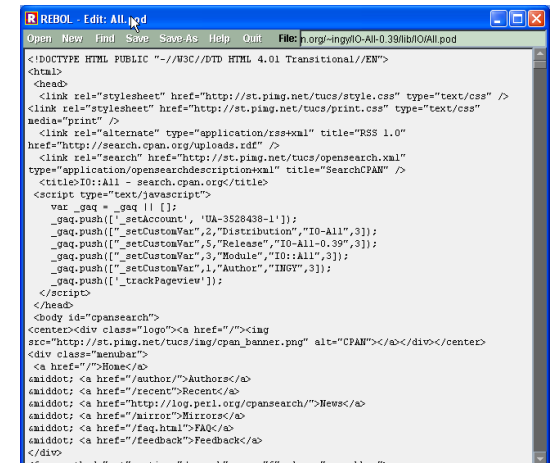
Simple Picture Viewer: 132



PITS:



view layout [
f: field btn"Edit"[editor to-url f/text]
]





- Lisp, Forth, Logo and Self
- Lightweight Distributed Computing
- programming-in-the-small (PITS)

REBOL



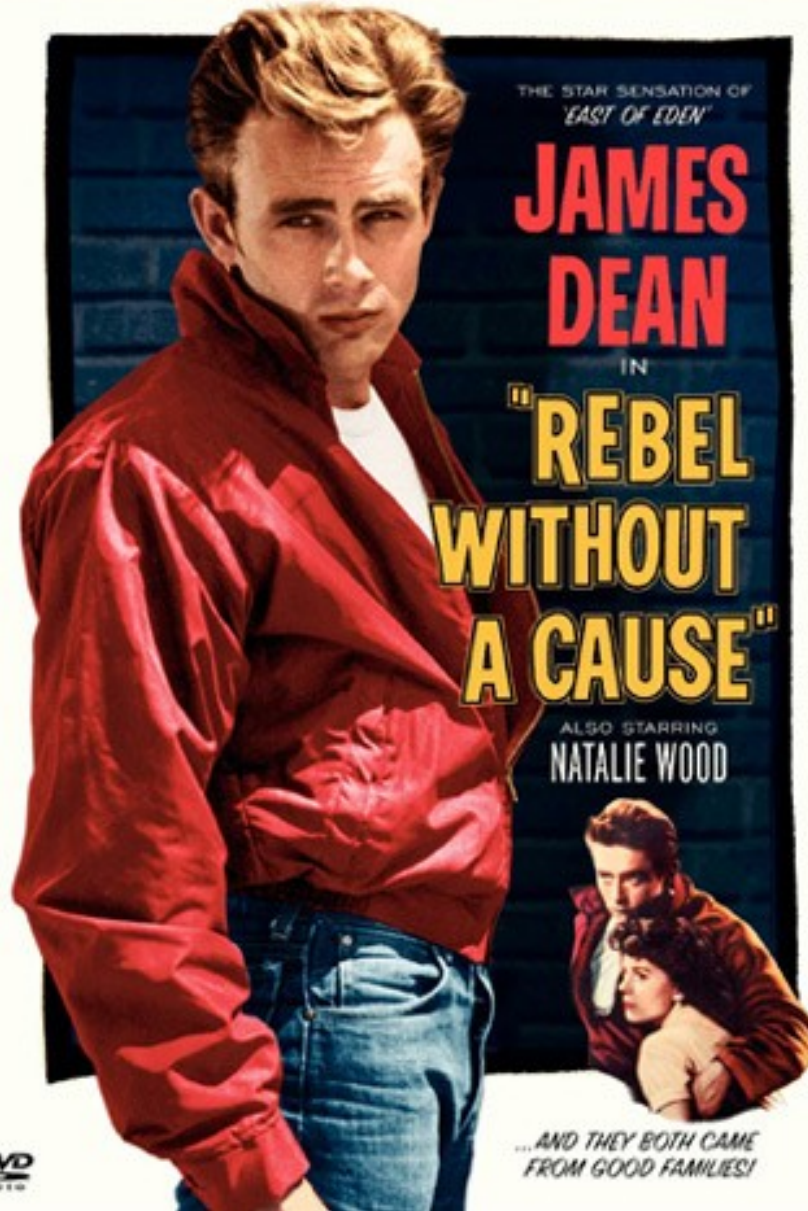
- Lisp, Forth, Logo and Self
- Lightweight Distributed Computing
- fighting software complexity

REBOL



- Lisp, Forth, Logo and Self
- Lightweight Distributed Computing
- fighting software complexity

TWO-DISC SPECIAL EDITION



THE STAR SENSATION OF
'EAST OF EDEN'

**JAMES
DEAN**

IN

**"REBEL
WITHOUT
A CAUSE"**

ALSO STARRING
NATALIE WOOD

*...AND THEY BOTH CAME
FROM GOOD FAMILIES!*

DVD
VIDEO

Silent Rage: Rebel with a cause




CHAMELEON
R E C O R D S
1101 WEST 11TH AVENUE, HAWTHORNE, CA 90250
1-818-671-8800 FAX: 1-818-671-8800

Silent
RAGE

SILENT RAGE F.C.
P.O. BOX 441
HAWTHORNE, CA 90250
(213) 536-0060

PHOTO: BARRY LEVINE

Relative Expression-Based Object Language



Relative Expression-Based Object Language

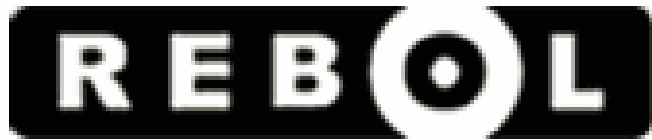
eval that like REBOL does:

back to front



Relative Expression-Based Object Language

the obvious, don't argue with that



Relative Expression-Based **Object** Language

sounds like OOP ?

REBOL



Relative Expression-Based **Object** Language

- no OOP like Ruby or Perl 6

REBOL



Relative Expression-Based **Object** Language

- no OOP like Ruby or Perl 6
- Object means here just „thing“

REBOL



Relative Expression-Based Object Language

That brings no meaning anyway :)

There are same principles it's based on



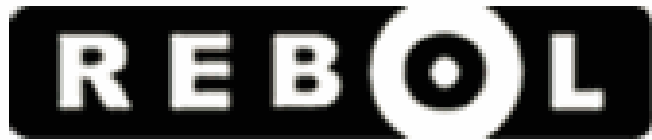
Relative **Expression**-Based Object Language

Sounds like a functional language ?



Relative **Expression**-Based Object Language

(not pure) functional (since 2.0)



Relative Expression-Based Object Language

REBOL



Relative Expression-Based Object Language

- That's the main message



Relative Expression-Based Object Language

- That's the main message
- That's how Rebol syntax works



Relative Expression-Based Object Language

- That's the main message
- That's how Rebol syntax works
- In Perl we call that context sensitive

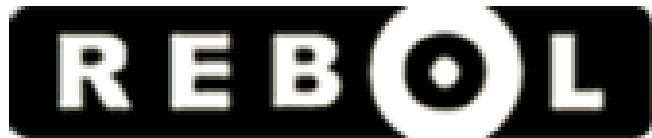
REBOL



Relative Expression-Based Object Language

```
read http://www.perl.org/learn.html
```

```
read %info.txt
```



Rebol

Perl 5

```
read http://www.perl.org/learn.html  
use LWP::Simple;  
get("http://www.perl.org/learn.html");
```

```
text: read %info.txt  
use File::Slurp;  
my $text = slurp( 'info.txt' );
```



Rebol

Perl 6

```
read http://www.perl.org/learn.html
    use HTTP::Client;
HTTP::Client.new.get("http://www.perl.org/learn.html");

text: read %info.txt
    my $text = slurp( 'info.txt' );
```



welcome to the strange land



shock therapy



REBOL, forget all you know

no variables

no hashes

no precedence table ($2 + 3 * 2 == 10$)

no builtins

no Subroutinen

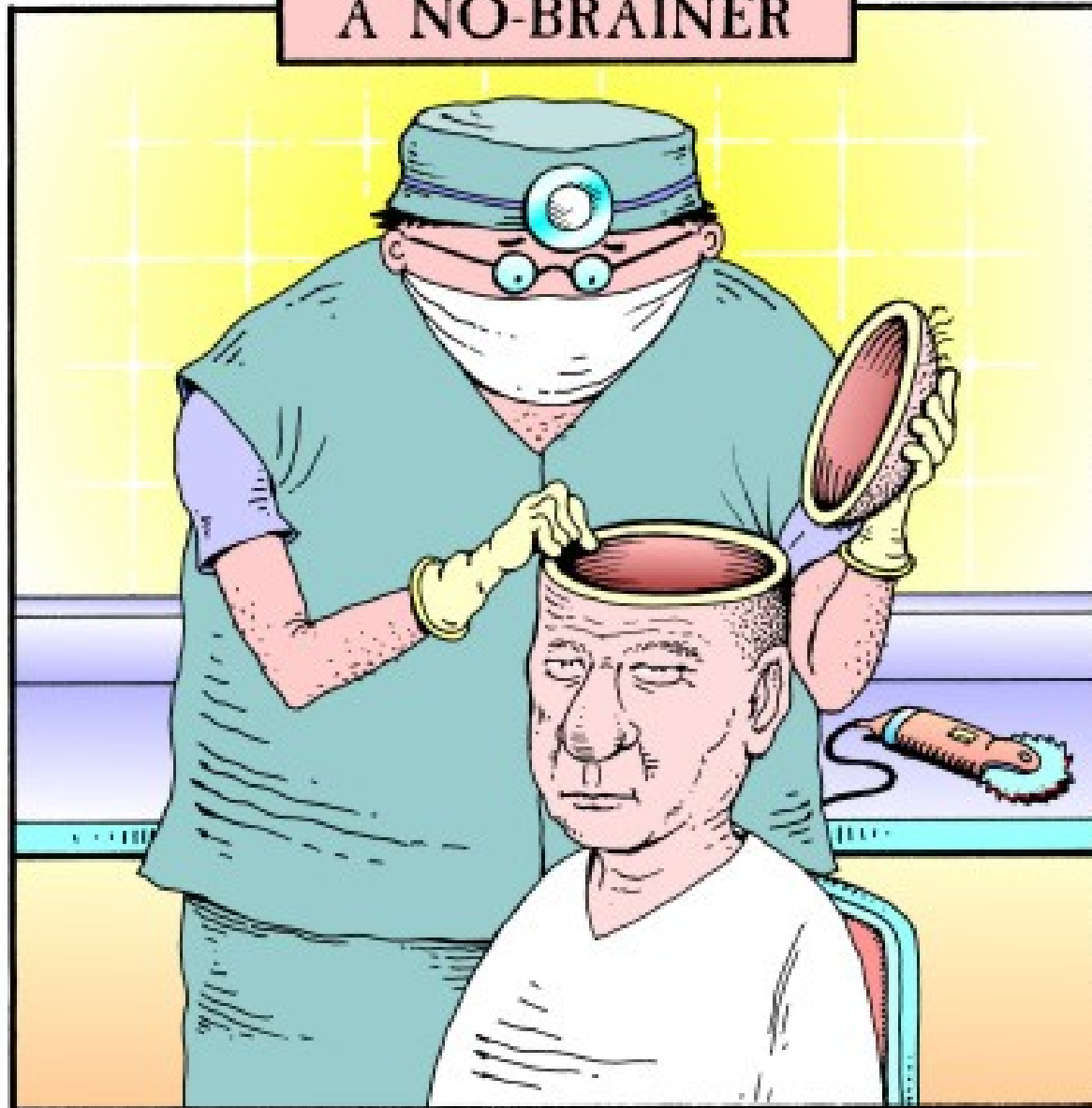
sometimes scopes

no name spaces (befor Rebol 3)

no Regex (parse)

All is gone?

A NO-BRAINER



You have to know 4 concepts



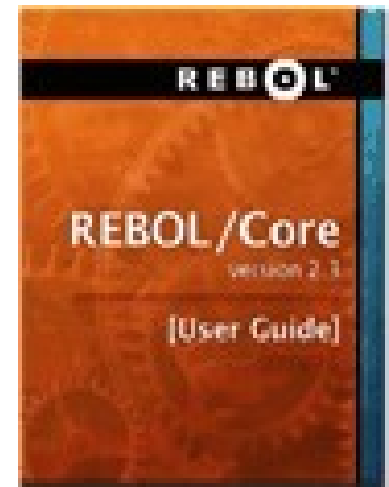
4 Terms

words:

data types:

series:

refinements:



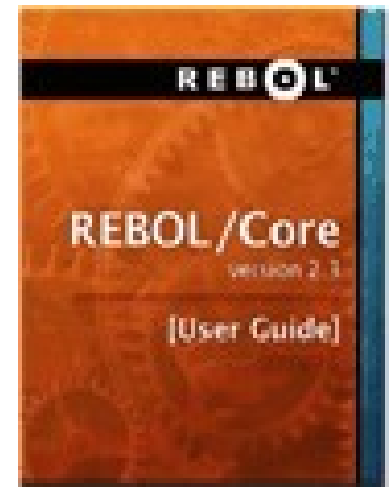
4 Terms

words: typeless data container

data types:

series:

refinements:



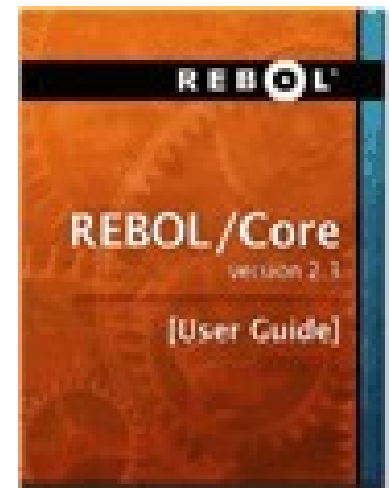
4 Terms

words: variables, routines, buildins

data types:

series:

refinements:



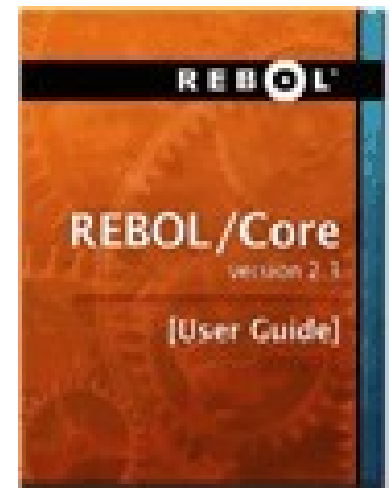
4 Terms

words: variables, routines, buildins

data types: integer, string, char, tuple, pair, block,

series:

refinements:



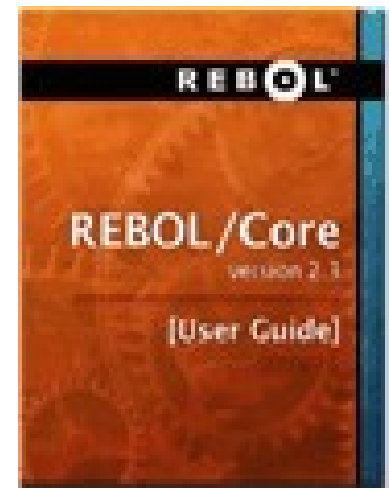
4 Terms

words: variables, routines, buildins

data types: integer, string, char, tuple, pair, block, money, file, url, email, tag, binary, date, time, issue

series:

refinements:



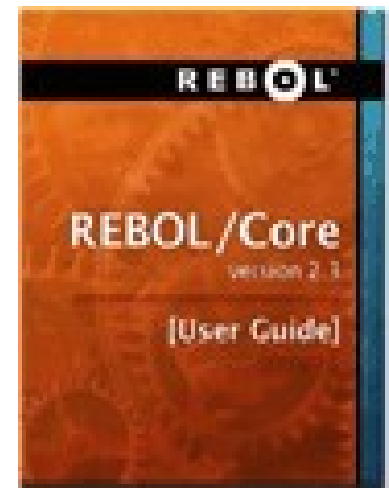
4 Terms

words: variables, routines, buildins

data types: integer, string, char, tuple, pair, block, money, file, url, email, tag, binary, date, time, issue

series:

refinements:



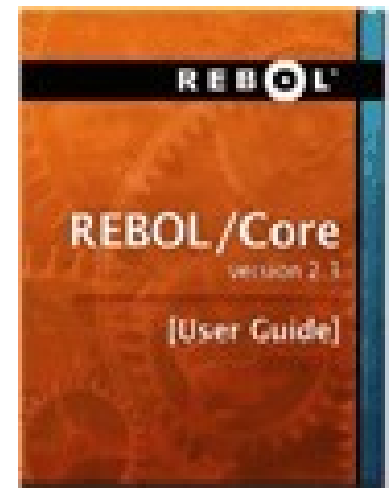
4 Terms

words: variables, routines, buildins

data types: integer, string, char, tuple, pair, block, money, file, url, email, tag, binary, date, time, issue

series: lists, hashes, code

refinements:



4 Terms

words: variables, routines, buildins

data types: integer, string, char, tuple, pair, block, money, file, url, email, tag, binary, date, time, issue

series: lists, hashes, code

refinements: (*@* / %) slices / return value

Perl

Rebol

#	⇒	;
}	⇒	[]
\$var =	⇒	var:
\$var	⇒	:var
eval \$var	⇒	var
\$var[3]	⇒	var/3
\$var{'sub'}	⇒	var/sub
sub()	⇒	sub
sub(\$p,\$m)	⇒	sub p m

Perl

Rebol

`$a++`

`a: a + 1`

Perl

```
$a = 30 * 3.1415 / 180  
sin($a) / cos($a)
```

```
sin($a) / cos($a)
```

```
use Math::Trig;  
tan(deg2rad(30));
```

Rebol

```
tangent 30
```

```
tangent/radians 30
```

Perl 6

Rebol

loop	=>	forever
all, any none @a	=>	all, any []
prompt	=>	ask
slurp	=>	read

no break in switch
Pair als nativer Typ
grammars, heavy usage of MMD



Meta Data

```
REBOL [  
  Title:  
  Date:  
  Version:  
  ...  
]
```



Meta Data

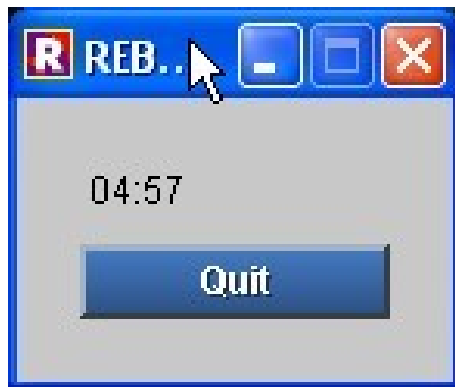
File: Author: Email: Web: Rights: Requires:

Purpose: { }, Comment: { }, Notes: { }

History: [0.1.1 12-Oct-2008 "add to ... path"

Library: [...

DSL for GUI



```
view layout [  
  text :rise_time  
  button "Quit" #"q" [quit]  
]
```

in Perl

```
use Tkx;
```

```
Tkx::button(".b",  
    -text => "Hello, world",  
    -command => sub { Tkx::destroy(".") },  
);
```

```
Tkx::pack(".b");
```

```
Tkx::MainLoop();
```

in Perl

```
use Prima qw(Application Button);
```

```
New Prima::MainWindow(  
    text => "Hello, world",  
    size => [ 200, 200 ],  
)-> insert( Button =>  
    centered => 1,  
    text => "Hello, world",  
    onClick => sub { $::application->close } ,  
);
```

```
run Prima;
```

RebGUI

```
view layout [  
  text :rise_time  
  button "Quit" #"q" [quit]  
]
```





DSL for Regex

```
expr: [term ["+" | "-"] expr | term]
term: [factor ["*" | "/"] term | factor]
factor: [primary "*" factor | primary]
primary: [some digit | "(" expr ")"]
digit: charset "0123456789"
```

```
probe parse "1 + 2 * ( 3 - 2 ) / 4" expr
== true
```



DSL for Regexp

parse read

<http://www.gaisma.com/en/location/halle.html>

[

thru <td class="sunshine"><i>

copy rise_time

to </i>

]

Functional DSL

does

func

function

DSL for OOP

make

Rebol 3

Open Source

Unicode

Debug Hooks

Better GUI Objects

Ideas

Io::All

Perlapi

Modules

Prima

