



# I have a dream...

S. Ducasse [stephane.ducasse@free.fr](mailto:stephane.ducasse@free.fr)



yo tengo un sueño ... :)

# S. Ducasse

LISP .... | 1996: discover Smalltalk

| 1998: discover Squeak

~2000: harvester for 3.xx, 3.6, 3.7, started to clean Squeak

Responsible for Squeak3.9

2008: Pharo :)

**Book author:** Squeak (fr), Squeak by example (eng, fr), Pharo by example (eng)

Translated Etoy book and video in french

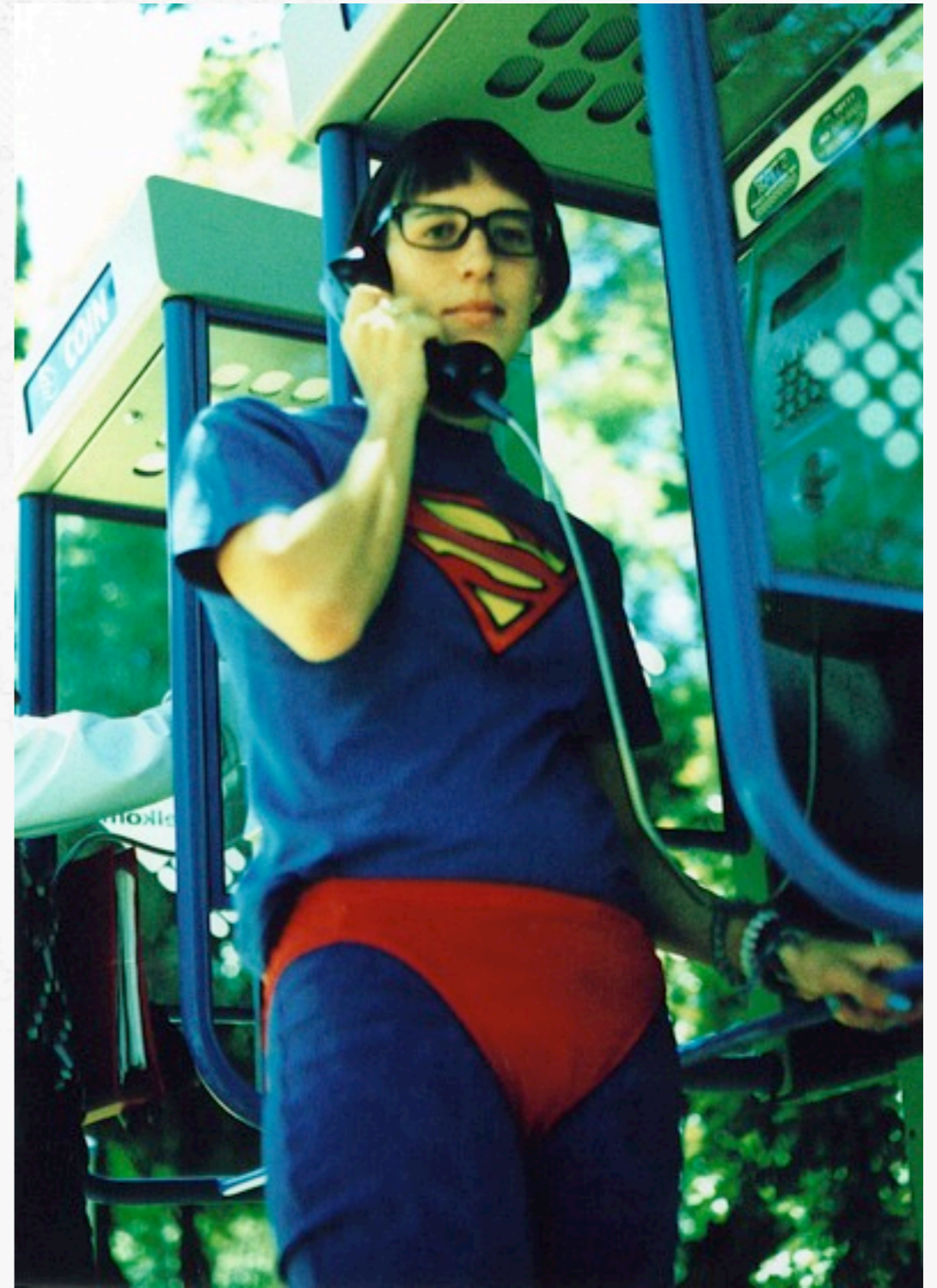


I dreamed about ...

The image shows a complex, futuristic interior, likely a space station or a high-tech vehicle. The scene is dominated by a large, curved structure with a ribbed, metallic appearance, possibly a tunnel or a large container. The lighting is a mix of vibrant blue and bright yellow, creating a high-tech, industrial atmosphere. In the foreground, there are various mechanical components, including pipes, wires, and a large white spherical object. The overall impression is one of advanced technology and a futuristic environment.

A vehicle to *reinvent* the future

Let's stop to think that we are *that* cool!






Python, Ruby, Groovy, Lua...

Erlang...

Let's stop to live in a museum!



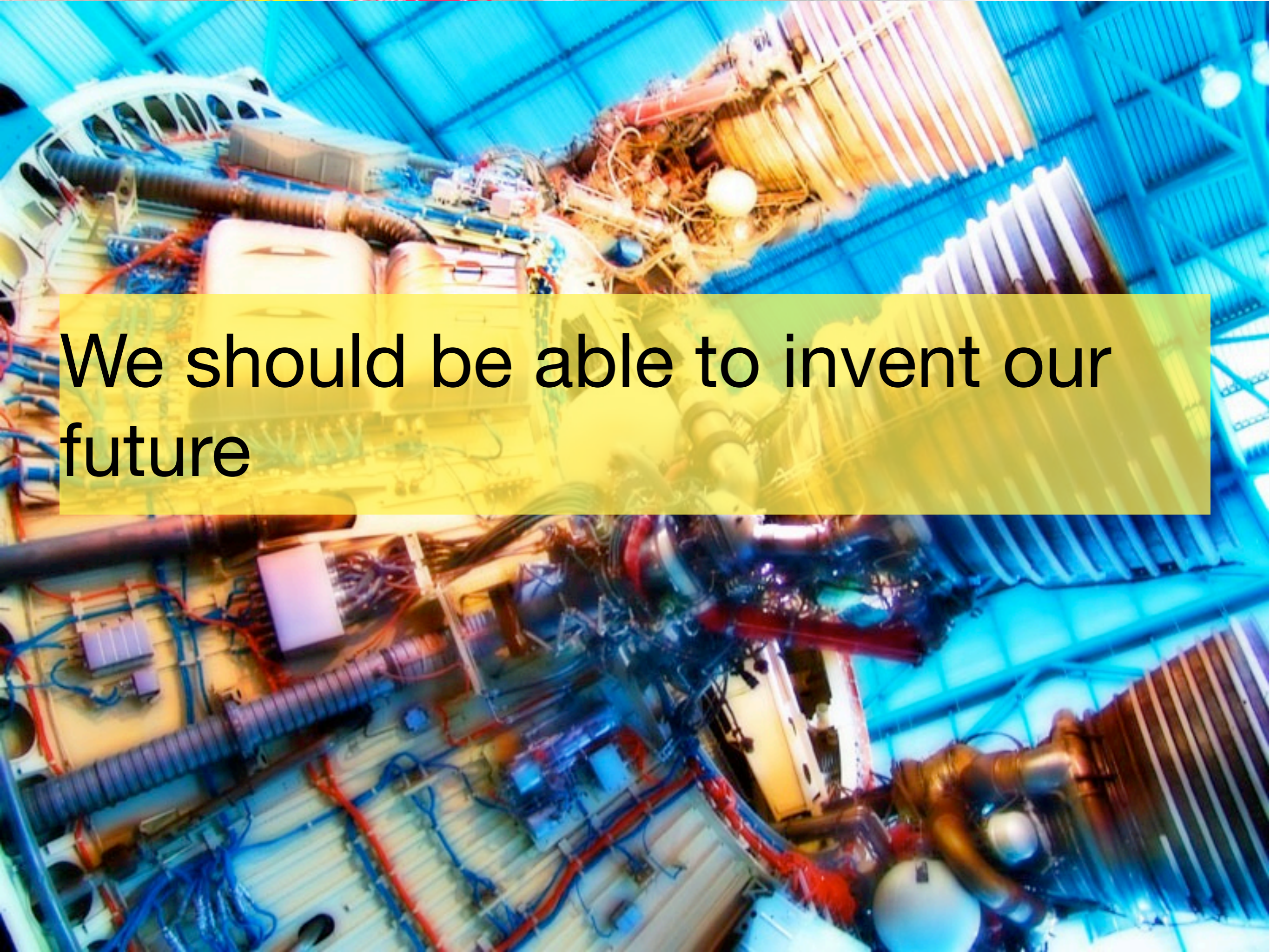




When do we *really* build on top of giant  
shoulders?

Thanks Dan and Alan. Now...

**I want *more!***



**We should be able to invent our future**

The background features a light gray, textured surface with a faint, stylized map of the African continent. Overlaid on this are several clusters of colorful splatters in shades of yellow, green, blue, pink, and purple, primarily concentrated in the upper-left quadrant.

**Where is Smalltalk innovation?**

# Smalltalk as the language to define other languages

Advanced and stronger Mops

We deserve more than doesNotUnderstand:

New ways of defining languages

Domain specific language support

New generation of meta compilers

# Software Engineering

Tests: which ones should I fix first?

Continuous test running

Smoother refactorings and code manipulation

Continuous integration / Extreme integration

Better Lint Rules / Domain specific lint rules

Bug tracking



# Other

Security?

Micro modularity

Advanced runtime

Scripting syntax!

# VM/Languages

Multiple cores?

Open VMs?

New generation JITs?

Immutable



# They show us the way

ObjectPeople (Dave Thomas and others)

Resilient (128K) Smalltalk on hardware

S#: Dave Simmons 90ms boot fully embeddable in  
C Smalltalk

Dave Thomas Sapphire's vision (ESUG@douai)

TTY + Image as a cache



# There is some hope

Seaside

Reflectivity, NewCompiler, AOSTA?

Monticello, O'Meta, PetitParser, Helvetia,  
Glamour, Mondrian, Traits...

Newspeak, Pepsi/Coke...

# When do we make sure that we could invent Smalltalk 3000?

first class instance variables, modules?, immutability bits?, visibility?,  
image as cache?



**We need a new state of mind**



**We are trying...**



*A clean, innovative, open-source Smalltalk*

<http://www.pharo-project.org>



# Pharo?

A *progressive*, open-source Smalltalk platform  
for *professional* use.

Stable

Bugs fixed fast

***But innovative***



# Pharo?

A **flexible** environment to support the  
**innovation** in/of Smalltalk.

Stable

Bugs fixed fast

**But innovative**





OB Package Browser: AutomaticMethodCategorizerApplier

C... (class search), i... (implementor search), #C... (class ref search), #... (code search)

AutomaticMethodCategorizerApplier

- AutomaticMethodCategorizerApplier
- AutomaticMethodCategorizerApplier
- AutomaticMethodCategorizerApplier
- AutomaticMethodCategorizerApplier
- AutomaticMethodCategorizerApplier

Initialization

initializeUsing:onCharacterNotifier

registerForNotifications

pkgs hier. groups instance ? class

DF 3/20/2006 21:15 - AutomaticMethodCategorizerApplier - initialization - 4

```

registerForNotifications

changeNotifier
  notify: self
    ofSystemChangesOfItem: AbstractEvent methodKind
    change: AddedEvent changeKind using: #methodAdditionNotifiedWith:;
  notify: self
    ofSystemChangesOfItem: AbstractEvent methodKind
    change: ModifiedEvent changeKind using: #methodModificationNotifiedWith:;

```

World

- Class Browser
- Workspace
- Test Runner
- Monticello Browser
- Tools
- Windows
- Debug
- System
- Save
- Save As...
- Save and quit
- Quit

Search the Seaside

# seaside The fast lane to HTTP

## About

- [Screenshots](#)
- [Success Stories](#)
- [Examples](#)
- [What others think](#)
- [Hosting](#)
- [Support](#)
- [Trivia](#)
- [Merchandise](#)

[more](#)

## Documentation

- [FAQ](#)
- [Tutorials](#)
- [Migration](#)
- [Videos](#)

[more](#)

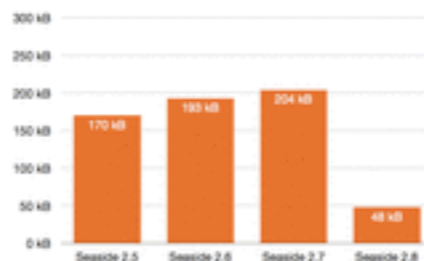
## Community

- [Weblogs](#)
- [Mailing Lists](#)
- [Development](#)
- [Contribute](#)
- [Merchandise](#)
- [Extensions](#)
- [Projects](#)

[more](#)

## Seaside 2.8

### Memory Requirements



## News

[\[Smalltalk Tidbits, Industry Rants\] Industry Misinterpretations 144: Children of Smalltalk](#) 12 July 2009

You'll have to listen to the entire podcast to understand where the title comes from :) This week ...

[London Geek Night](#) 10 July 2009

I have some pictures to share for people that missed the London Geek Nights with the title Seaside: ...

[Web Velocity released](#) 29 June 2009

Big news from Cincom on the Seaside mailing list today: they've made an evaluation version of their ...

[London](#) 23 June 2009

I'll be in London (UK) on Wednesday June 24 for 10 days or so. If anyone around there wants to meet ...

[Seaside 2.8.4 and Pharo](#) 5 June 2009

We just announced Seaside 2.8.4, the fourth maintenance release for Seaside 2.8. Again this release ...

[more](#)



Seaside is a free and [Open Source™](#) web application framework distributed under the [MIT License](#).

Seaside is available on the following Smalltalk platforms:

- [Pharo Smalltalk \(download\)](#)
- [Cincom Smalltalk](#)
- [Dolphin Smalltalk](#)
- [GemStone Smalltalk](#)
- [GNU Smalltalk](#)
- [VA Smalltalk](#)

[more](#)



EUROPEAN SMALLTALK USER GROUP

|||

www.esug.org

# pier

Pier is a content management system that is light, flexible and free. It is light as in double click and go ([download it](#)). It is flexible as in make it be what you want ([learn more](#)). It is free as in freedom ([MIT license](#)).

Pier 1.2

11 July 2009

Pier 1.2 is now available.

What's new: \* Mature value links (see Pier syntax for more details) \* ...

July Pier sprint

11 July 2009

Yesterday, Lukas and me spent an intense afternoon working on Pier. We focused on closing the open ...

Pier sprint

5 April 2009

Pier-session.jpg Yesterday, Lukas and me sprinted intensively through Pier. It was quite exciting....

[Read more blog entries ...](#)

# iPhone





**GEMSTONE**

**Seaside under GLASS**

GemStone • Linux • Apache • Seaside • Smalltalk

**Smalltalk with OO-Database**

**Pharo is the IDE**

# Companies

netstyle.ch

cmsbox.com

Pinesoft

Smallworks

Agilitic.be

Inceptive.be

10Pines

GemStone

SWEng

Denker2Denker

...

# Universities

Annecy

Lugano

Bern

Douai

Lille

Santiago

Bruxelles

Limoge

...

# A Simple and Pure Model


Everything is an object instance of a class

Public methods

Protected attributes

Single inheritance





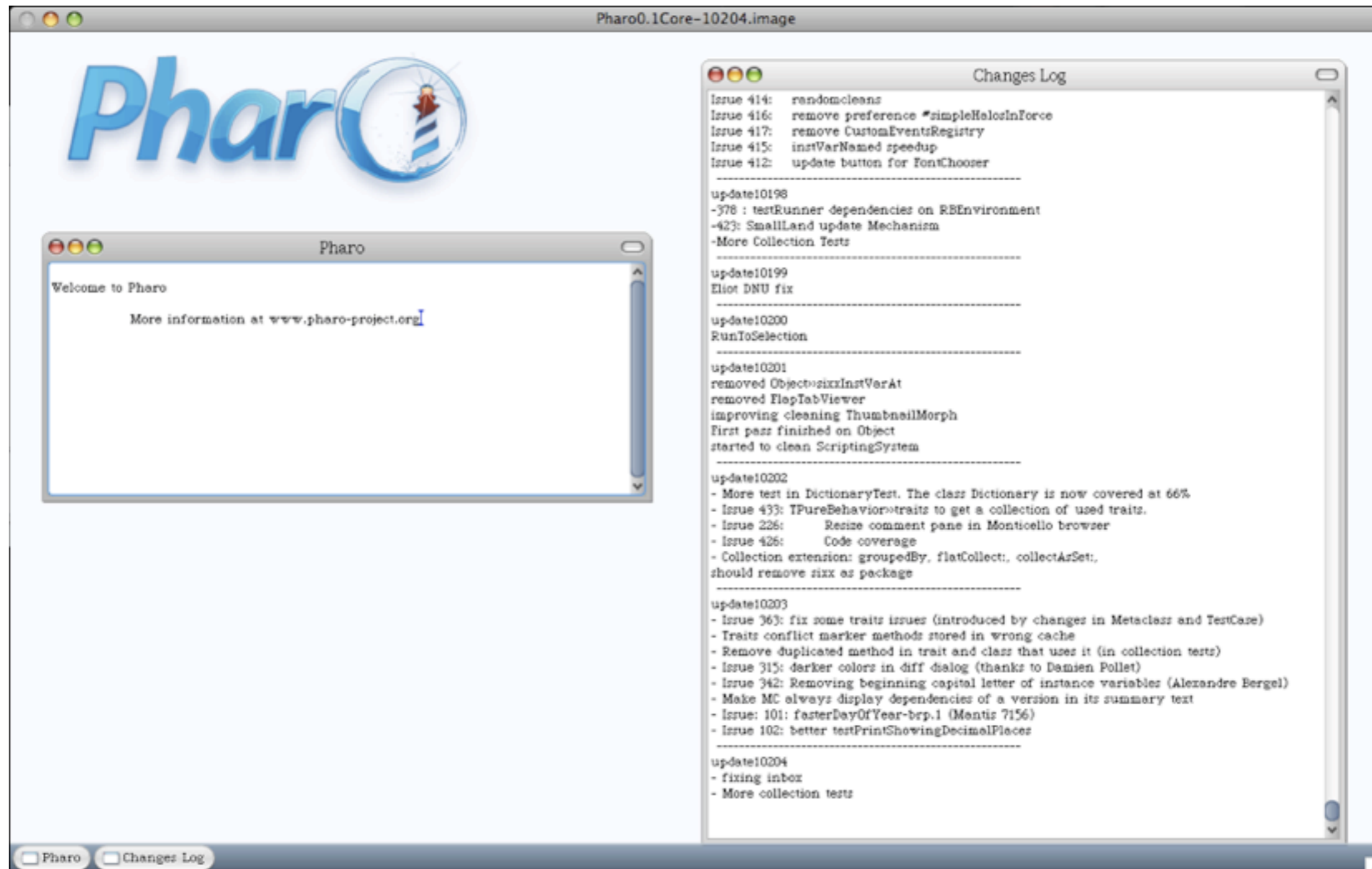
# Everything happens by sending messages to objects

1000 factorial / 999 factorial

(Smalltalk isCool) ifTrue: ['Yeahh']

#(1 -2 3) collect: [ :each | each abs ]

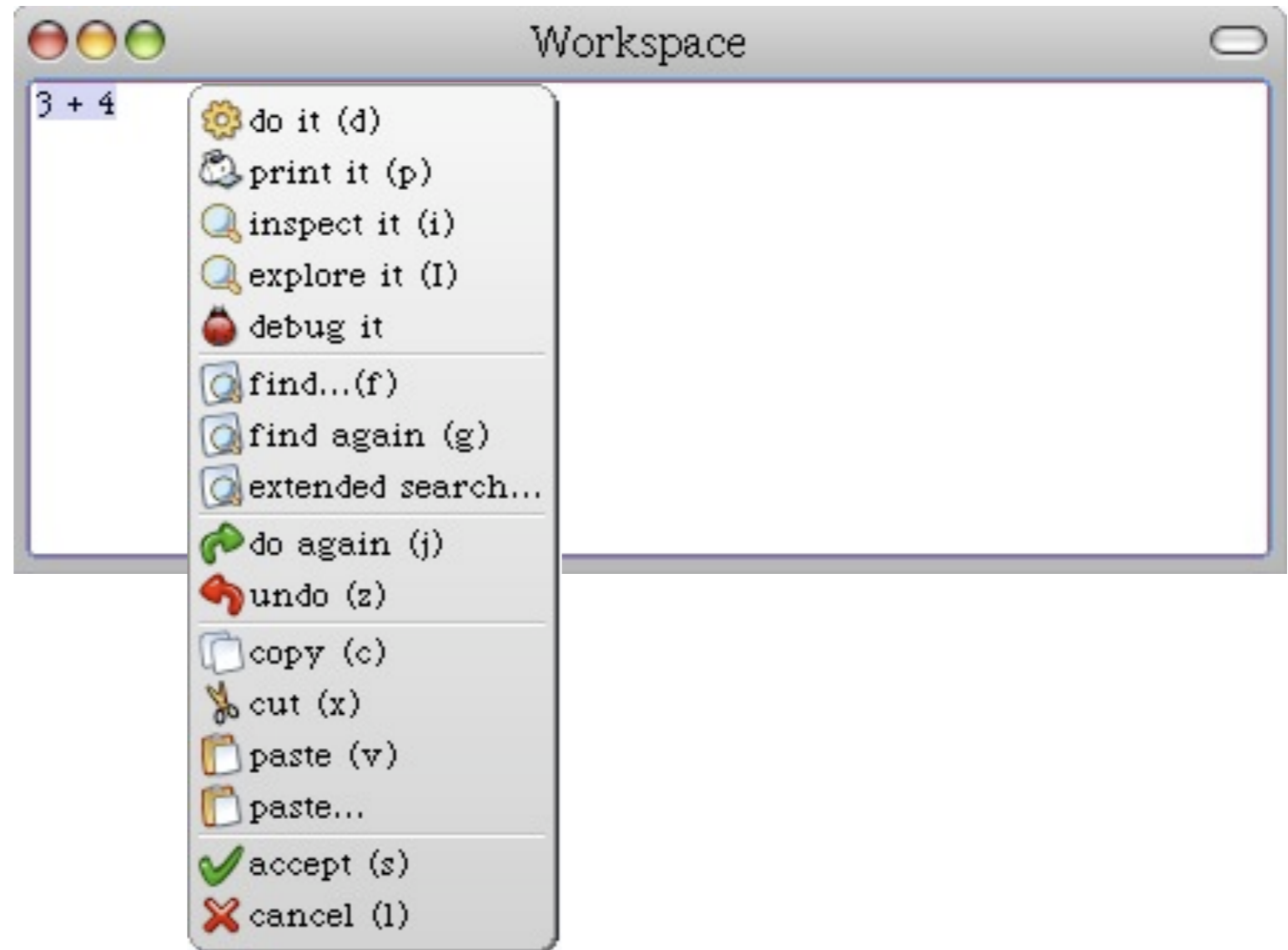
# Running Pharo



# Do it, print it

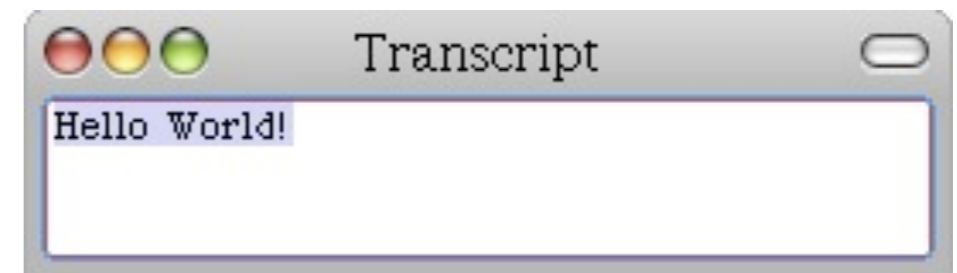
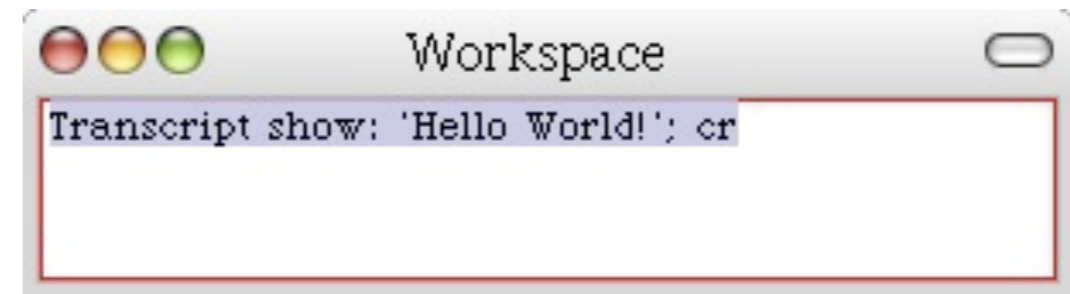
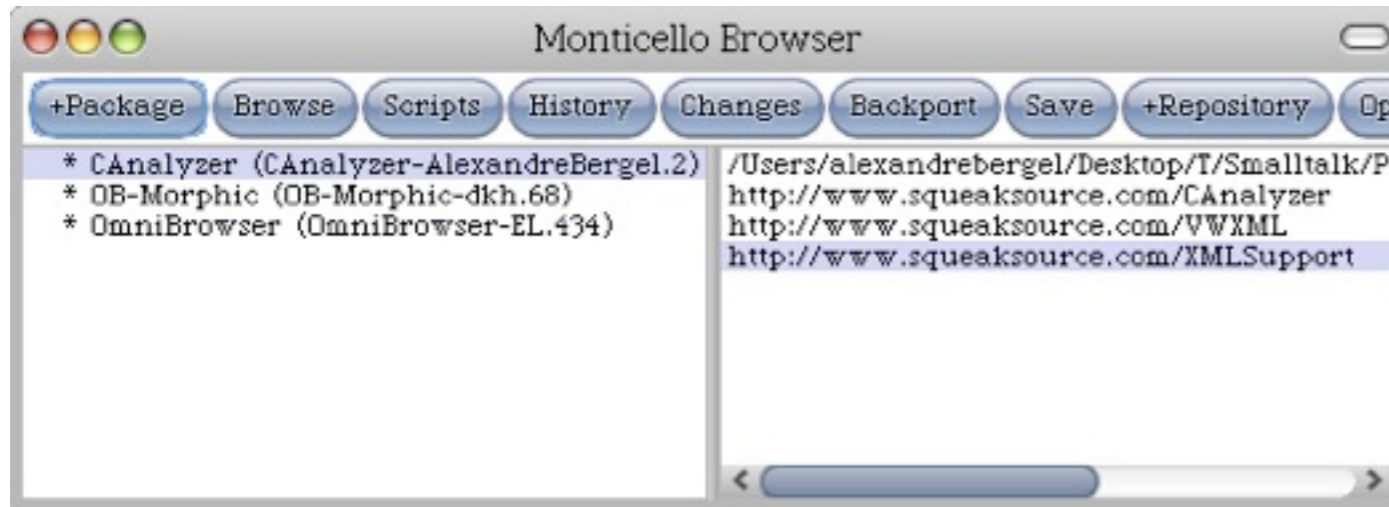
---

You can evaluate  
any expression  
anywhere  
in Pharo

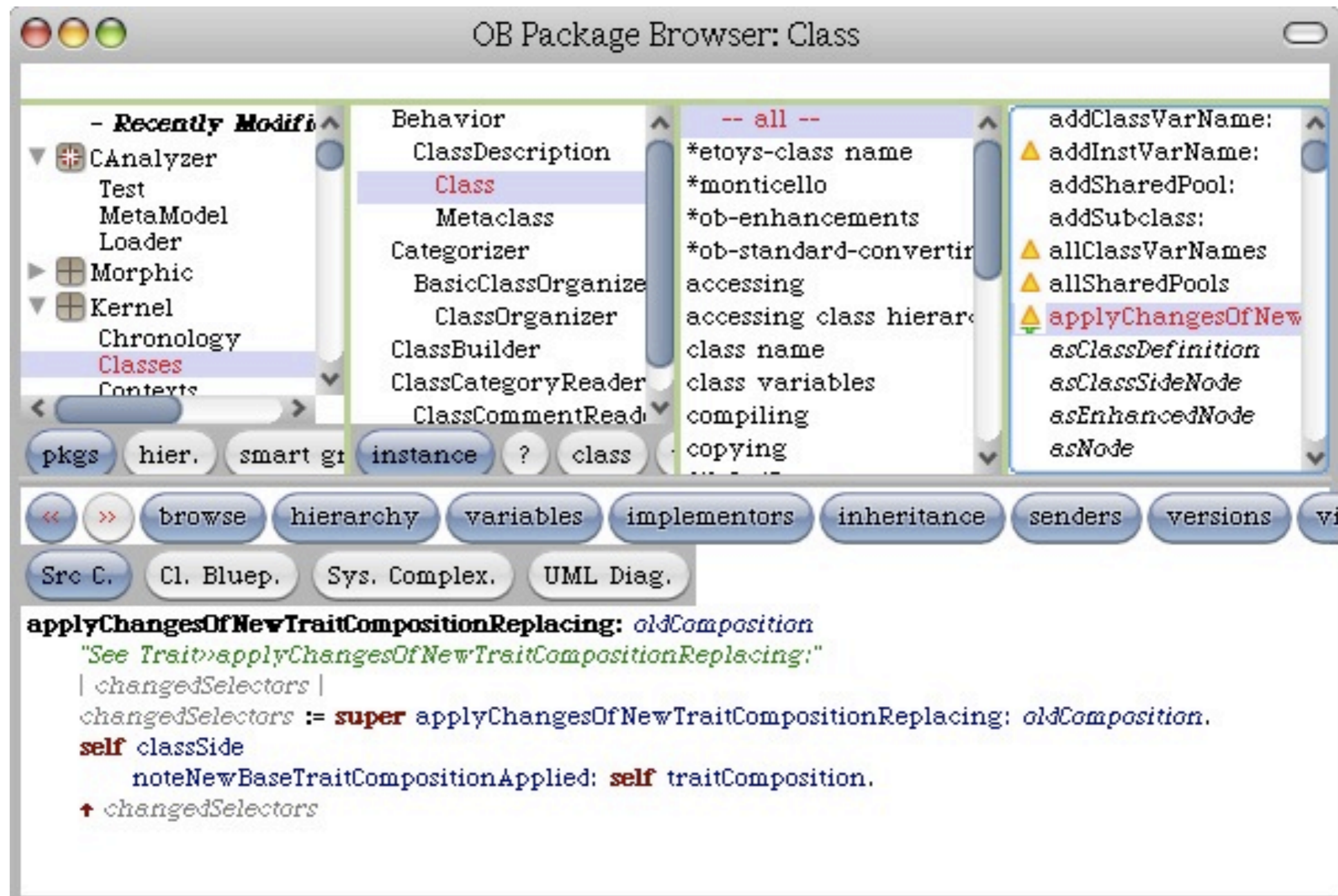


# Standard development tools

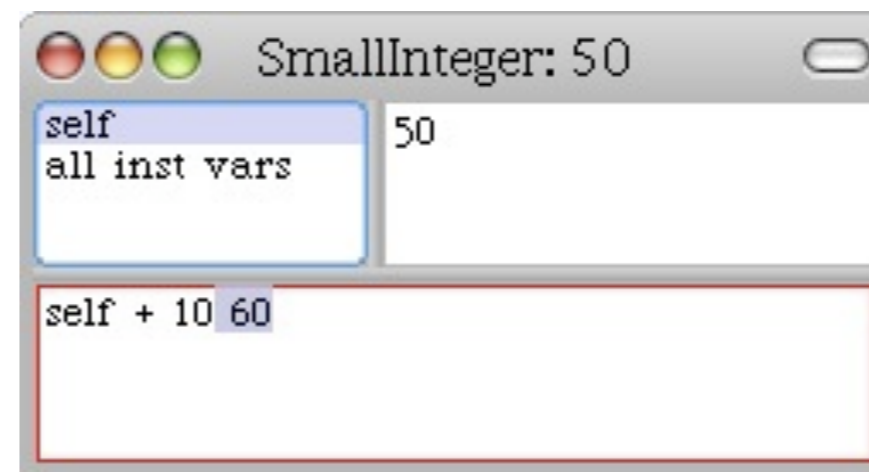
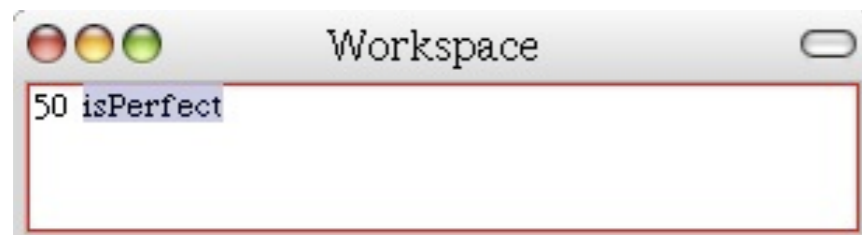
---



# Standard development tools



# Debugger, explorer, inspector



# 3 kinds of messages

---

Unary messages

```
5 factorial  
Transcript cr
```

Binary messages

```
3 + 4
```

Keywords messages

```
3 raisedTo: 10 modulo: 5  
Transcript show: 'hello world'
```

# From Java to Smalltalk

```
postman.send(mail,recipient);
```



# Removing

```
postman.send(mail, recipient);
```

The background features a light gray world map with a subtle texture. Overlaid on the map are numerous colorful splatters in shades of yellow, blue, pink, and purple, primarily concentrated in the upper-left quadrant.

# Removing unnecessary

## **postman send mail recipient**

**But without losing  
information**

**postman send mail **to** recipient**

**postman send: mail to: recipient**

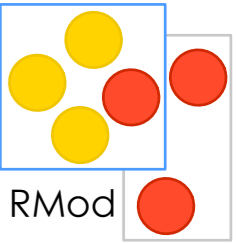
**postman.send(mail,recipient);**

# ***Precedence***

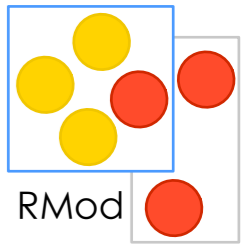
(Msg) > Unary > Binary > Keywords

from left to right

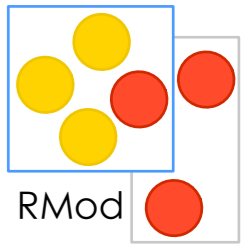
No mathematical precedence



$2 + 3$  squared



$2 + 3 \text{ squared}$   
 $> 2 + 9$

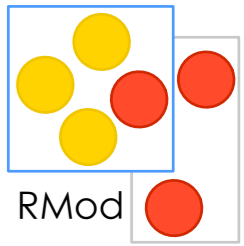


$2 + 3 \text{ squared}$

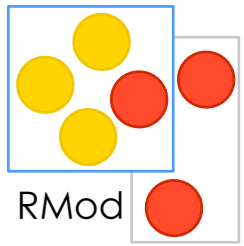
$> 2 + 9$

$> 11$

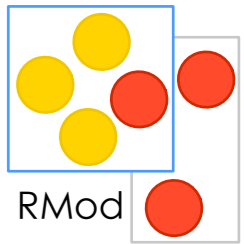




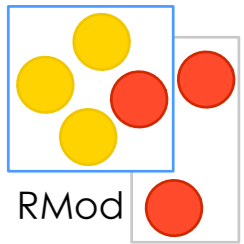
Color gray - Color white = Color black



Color gray - Color white = Color black



Color gray - Color white = Color black  
> aColor = Color black



**Color gray - Color white = Color black**

**> aColor = Color black**

**> true**

# Statement and cascades

---

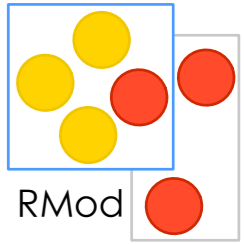
Temporary variables

Statement

```
| p pen |  
p := 100@100.  
pen := Pen new.  
pen up.  
pen goto: p; down; goto: p+p
```

Cascade

# Block Closures: aka Function

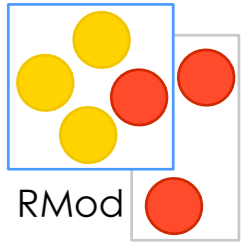


$$\text{fct}(x) = x * x + x$$

|fct|

fct:= **[ :x | x \* x + x ]**.

# Function Application



fct (2) = 6

fct (20) = 420

fct value: 2

> 6

fct value: 20

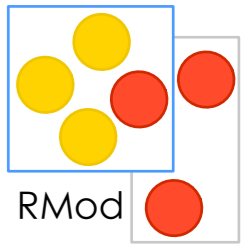
> 420

**[ :x | x \* x + x ] value: 2**

**>6**

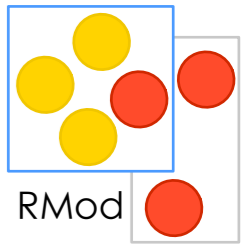
**[ :x | x \* x + x ] value: 20**

**>420**

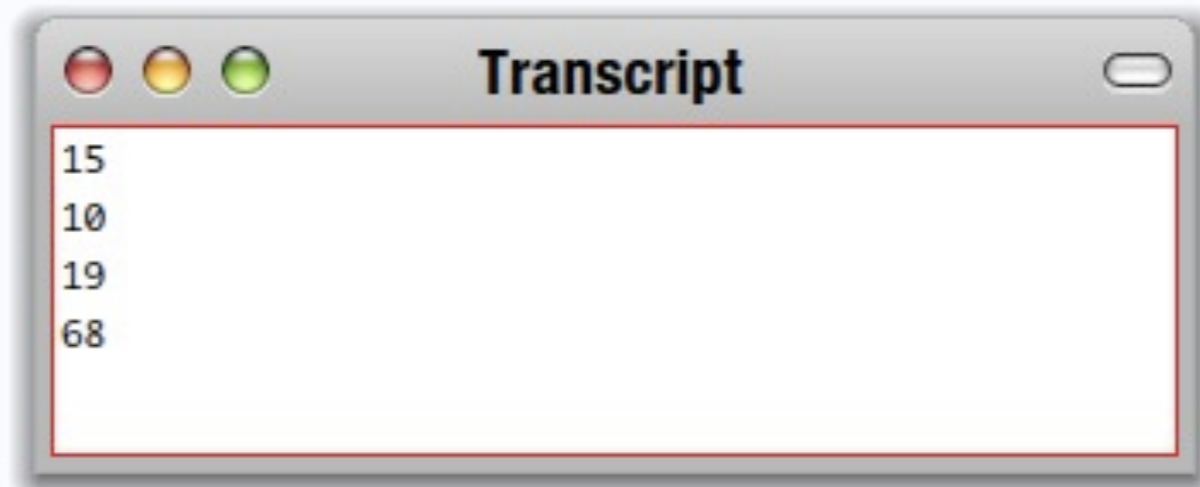


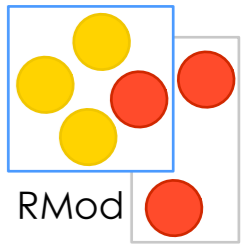
```
#(15 10 19 68) do:  
[:i | Transcript show: i ; cr ]
```





```
#(15 10 19 68) do:  
[ :i | Transcript show:i ; cr ]
```

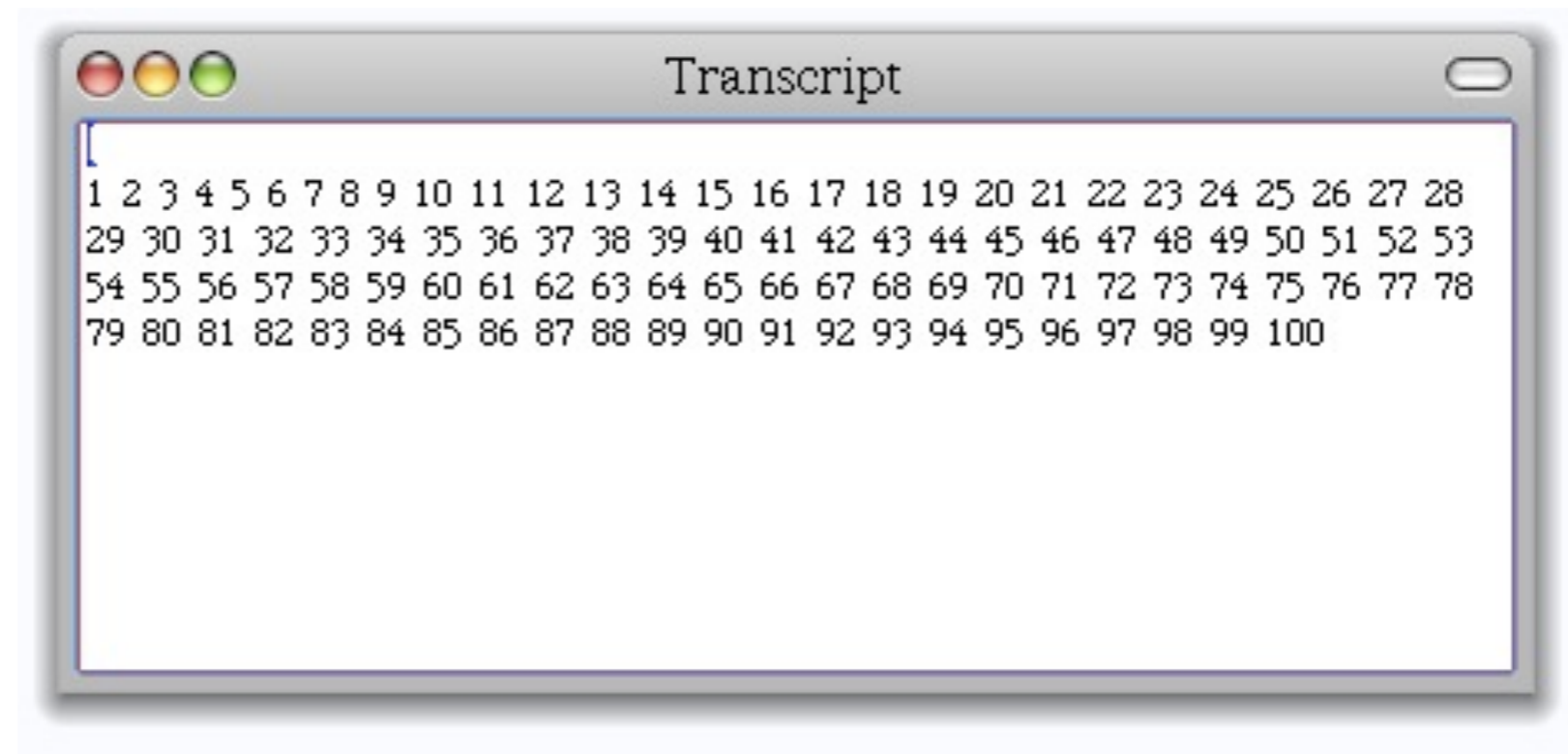




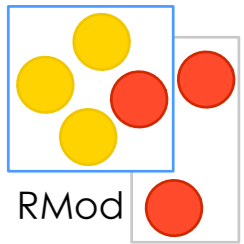
**| to: 100 do:**

**[ :i | Transcript show: i ; space]**

```
| to: 100 do:  
  [:i | Transcript show: i ; space]
```

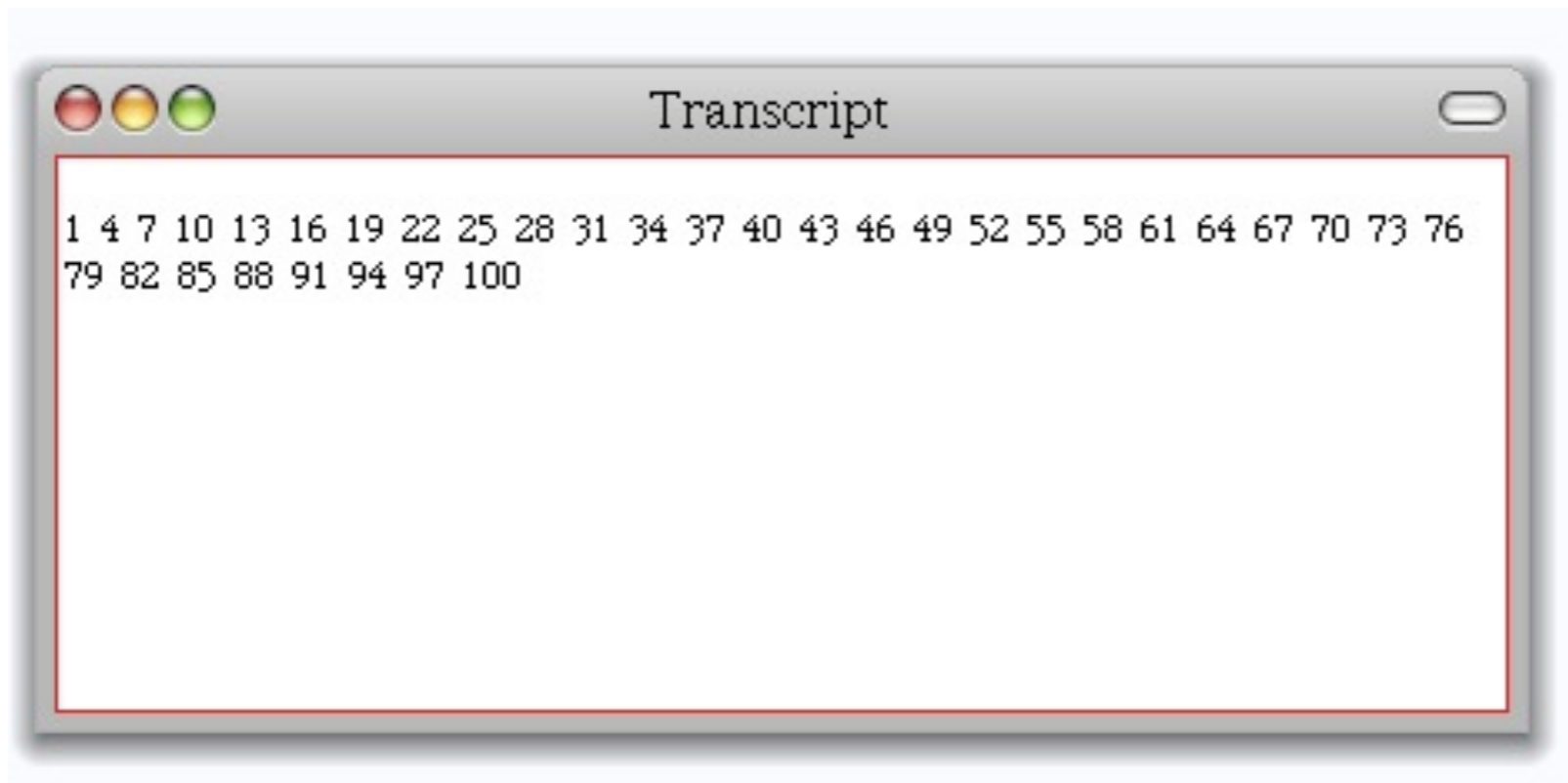
A screenshot of a window titled "Transcript" showing the output of the Racket code. The window has a standard macOS-style title bar with red, yellow, and green window control buttons on the left and a close button on the right. The text inside the window is:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28  
29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53  
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78  
79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
```



```
| to: 100 by: 3 do:  
  [ :i | Transcript show: i ; space]
```

```
l to: 100 by: 3 do:  
  [ :i | Transcript show: i ; space]
```

A screenshot of a window titled "Transcript" with a red border. The window contains two lines of text: "1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76" on the first line and "79 82 85 88 91 94 97 100" on the second line.

```
Transcript  
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76  
79 82 85 88 91 94 97 100
```

# A typical method in Point

---

Method name

Argument

Comment

```
<= aPoint
```

```
"Answer whether the receiver is neither  
below nor to the right of aPoint."
```

```
^ x <= aPoint x and: [y <= aPoint y]
```

Return

Instance variable

Binary message

Keyword message

Block

```
(2@3) <= (5@6)
```

```
true
```

# Creating classes

---

Send a message to a class (!)

```
Number subclass: #Complex  
instanceVariableNames: 'real imaginary'  
...  
category: 'ComplexNumbers'
```

# 1.0

- + **Major** Cleanups (MVC, eToys)
- + New UI Look / TrueType
- + Tools
- + **Block Closures (thanks Qwak)**
- + **Lots** of bugfixes and small improvements
- + Preferences clean up
- + **MIT license clean**





**Update ~1200 Bug-reports closed**

**470 Updates**

**Release Candidate: October 2009**

The background features a light gray, textured surface with a faint world map. Overlaid on this are various colorful splatters and dots in shades of yellow, green, blue, pink, and purple, primarily concentrated in the upper left quadrant. The word "Future" is centered in a bold, black, sans-serif font.

**Future**



**...cleaner**

**started to run SmallLint... and fix**



**...better**

**I.I has a lot of fixes and improvements**



**...smaller**

**small image is 2 Mb**

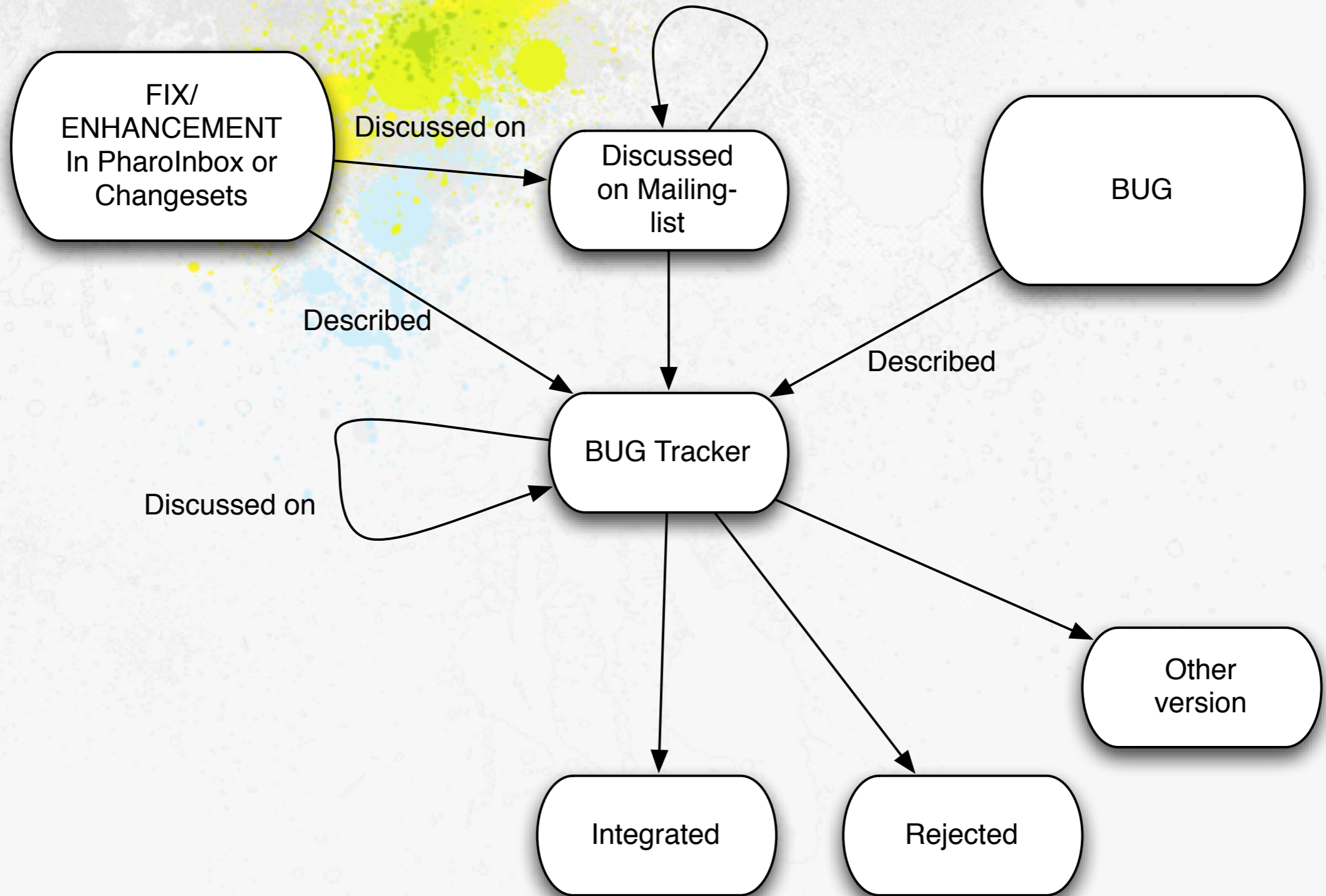
The background features a light gray world map with a thin, dotted outline. Overlaid on the map are several clusters of colorful splatters in shades of yellow, blue, pink, and purple, primarily concentrated in the upper-left quadrant. The overall aesthetic is clean and modern.

**...faster**

# Everybody can help

Reporting bugs  
Confirming bugs  
Writing tests  
Writing examples  
Writing comments  
Simple contributing fixes  
Deep discussion...

# Process





# Pharo Sprints

May 2008 Bern

July 2009 Bern

October 2009 Lille

November 2009 Buenos Aires

Free access

# Books

Pharo by example  
vol. 1 is out

Pharo by example  
vol. 2 on preparation

# Thanks

Hans Beck  
Alexandre Bergel  
Cedric Beler  
Torsten Bergmann  
Matthias Berth  
Ralph Boland  
Noury Bouraqadi  
Brian Brown  
Gwenael Casaccio  
Damien Cassou  
Nicolas Cellier  
Gary Chambers  
Miguel Coba  
Gabriel Cotelli  
Carlos Crosetti  
Cyrille Delaunay  
Simon Denier  
Marcus Denker  
Ramiro Diaz Trepas  
Stephane Ducasse  
Morales Durand Hernan  
Stephan Eggermont  
Luc Fabresse

Matthew Fulmer  
Hilaire Fernandes  
Julian Fitzell  
Tudor Girba  
Sean Glazier  
Norbert Hartl  
Dale Henrichs  
Reinout Heeck  
Eric Hochmeister  
Keith Hodges  
Henrik Sperre Johansen  
Pavel Krivanek  
Adrian Kuhn  
Adrian Lienhard  
Andreas Leidig  
Mariano Martinez Peck  
Dave Mason  
John McIntosh  
Johnaton Meichtry  
Eliot Miranda  
Hernan Morales Durand  
Philipp Marshall  
Jannick Menanteau  
Yann Monclair  
Oscar Nierstrasz

David J Pennell  
Joseph Pelrine  
Alain Plantec  
Damien Pollet  
Lukas Renggli  
Jorge Ressia  
Mike Roberts  
Robert Rothwell  
David Rotlisberger  
Michael Rueger  
Bill Schwab  
Niko Schwarz  
Igor Stasenko  
Francois Stephany  
Serge Stinckwich  
Mathieu Suen  
Lawrence Trutter  
Andrew Tween  
martin von loewis  
Andres Valloud  
Juan Vuletich  
Steven Wirts  
Hernan Wilkinson



# Join Us!

Creating good energy, software quality,  
learning and having fun

**<http://pharo-project.org>**