

12. A bit of Smalltalk



Roadmap

- > The origins of Smalltalk
- > What is Smalltalk?
- > Syntax in a nutshell
- > Seaside — web development with Smalltalk

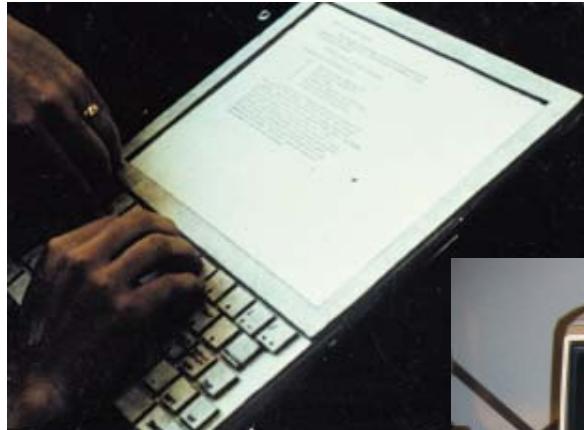


Roadmap

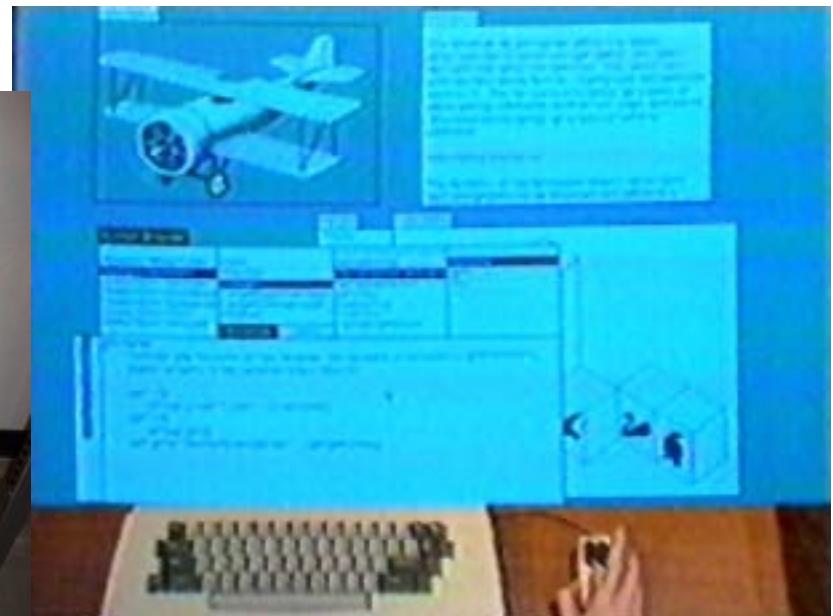
- > **The origins of Smalltalk**
- > What is Smalltalk?
- > Syntax in a nutshell
- > Seaside — web development with Smalltalk



The origins of Smalltalk



Alan Kay's Dynabook project (1968)



Alto — Xerox PARC (1973)

gagne.homedns.org/~tgagne/contrib/EarlyHistoryST.html

Object-oriented language genealogy

1950

1960

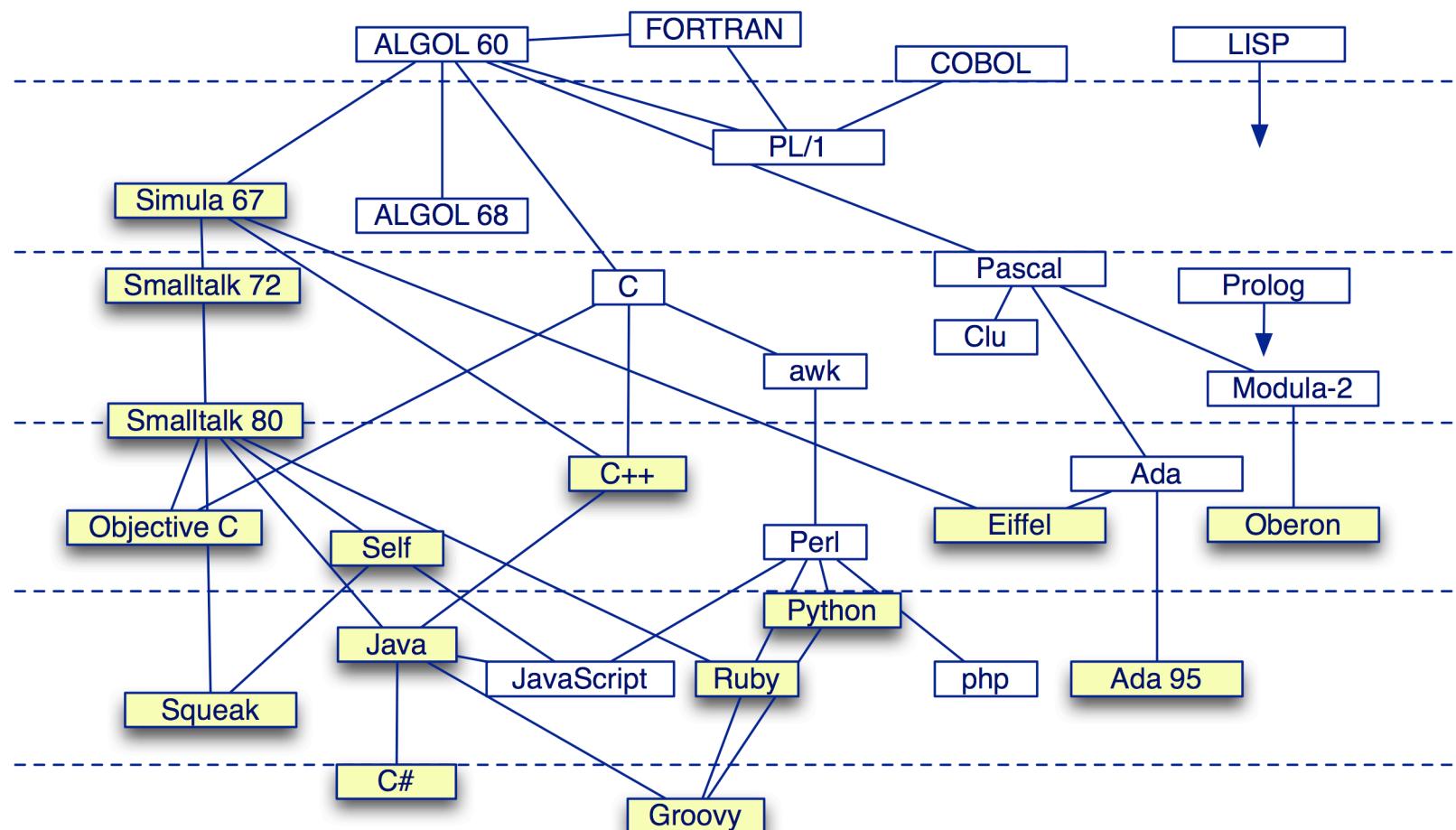
1970

1980

1990

2000

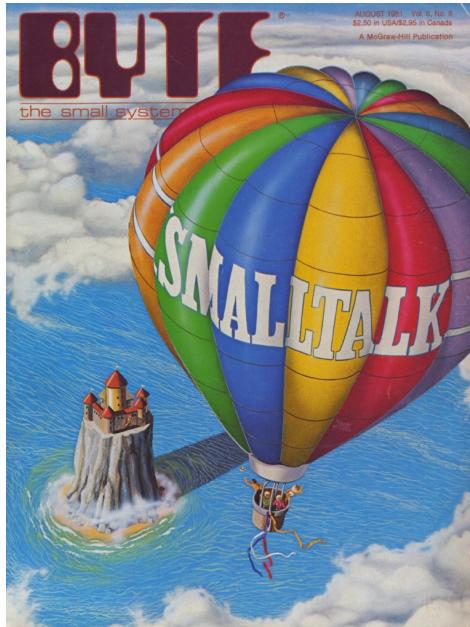
2010



Smalltalk vs. C++ vs. Java

	<i>Smalltalk</i>	<i>C++</i>	<i>Java</i>
<i>Object model</i>	Pure	Hybrid	Hybrid
<i>Garbage collection</i>	Automatic	Manual	Automatic
<i>Inheritance</i>	Single	Multiple	Single
<i>Types</i>	Dynamic	Static	Static
<i>Reflection</i>	Fully reflective	Introspection	Introspection
<i>Concurrency</i>	Semaphores	Some libraries	Monitors
<i>Modules</i>	Categories, namespaces	Namespaces	Packages

Smalltalk-80 and Pharo



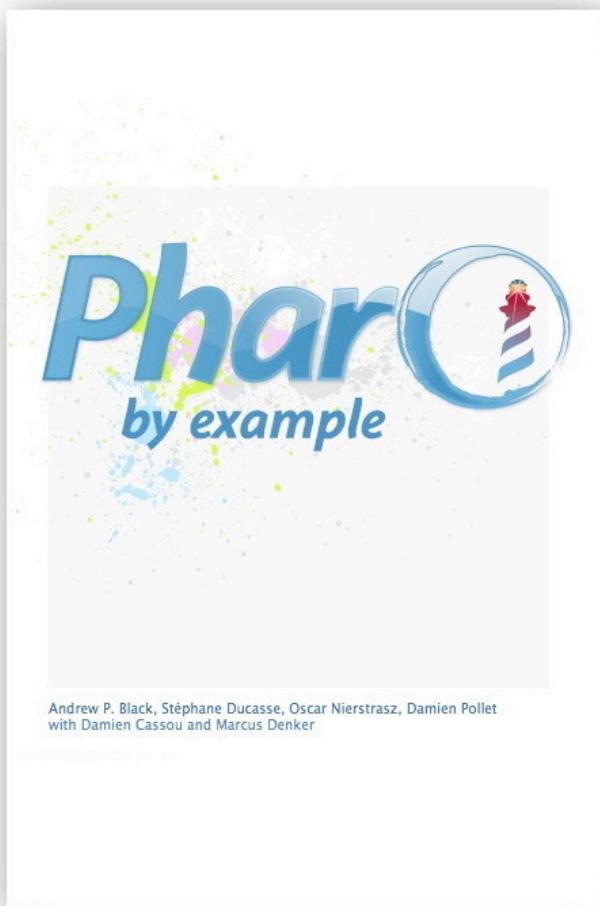
- Everything is an object
- Everything is there, all the time
- First windowing system with mouse
- First graphical IDE

What are Squeak and Pharo?

- > Squeak is a modern, open-source, highly portable, fast, full-featured Smalltalk implementation
 - Based on original Smalltalk-80 code
- > Pharo is a lean and clean fork of Squeak
 - www.pharo-project.org



Pharo by Example



<http://pharobyexample.org/>

- Free download
- Open-Source
- Print-on-demand

Don't panic!

New Smalltalkers often think they need to understand all the details of a thing before they can use it.

Try to answer the question

“How does this work?”

with

“I don’t care”.

—Alan Knight. Smalltalk Guru

Roadmap

- > The origins of Smalltalk
- > **What is Smalltalk?**
- > Syntax in a nutshell
- > Seaside — web development with Smalltalk



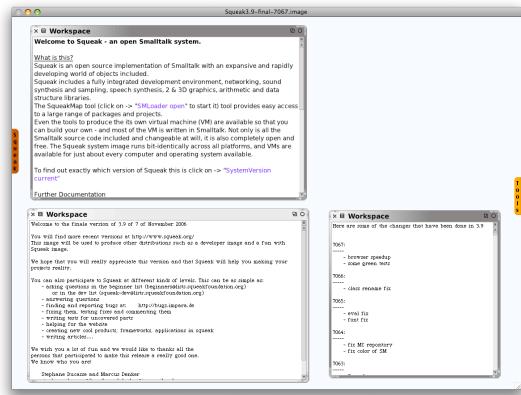
Two rules to remember

Everything is an object

**Everything happens by
sending messages**

What is Smalltalk?

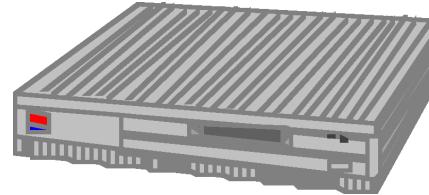
Image



Changes

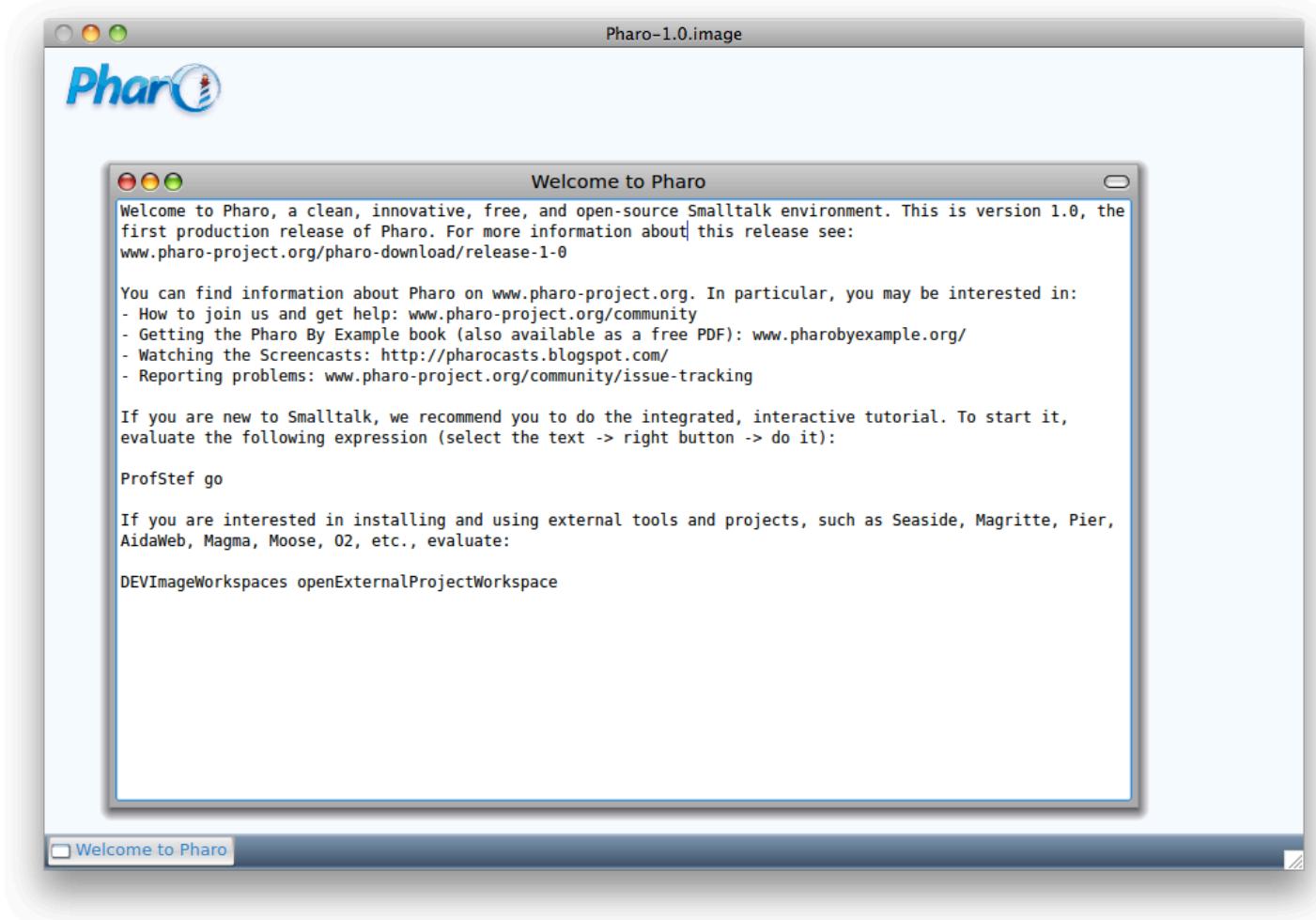


Virtual machine

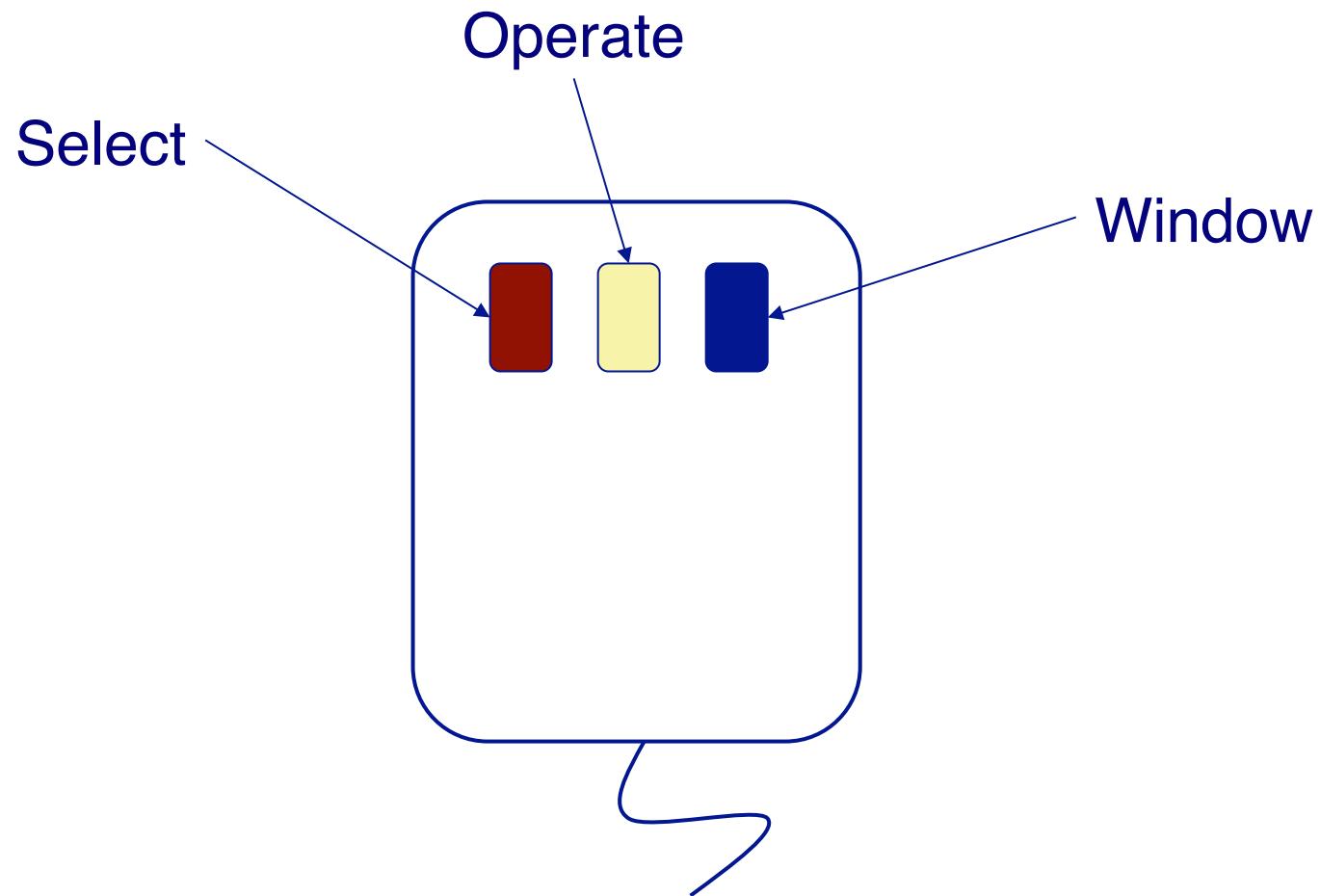


Sources

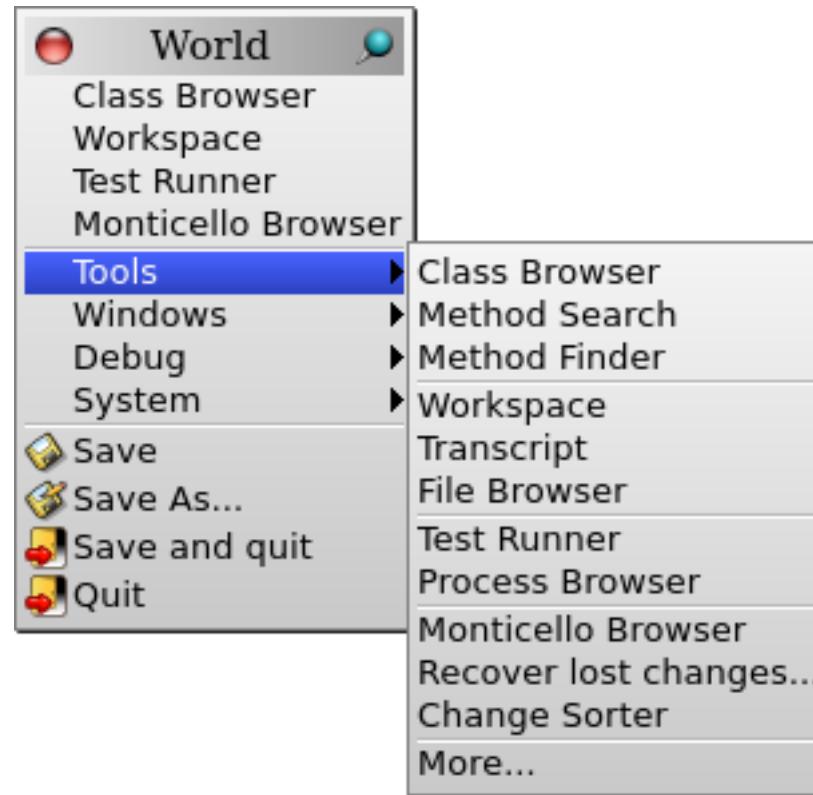
Demo: Running Pahro



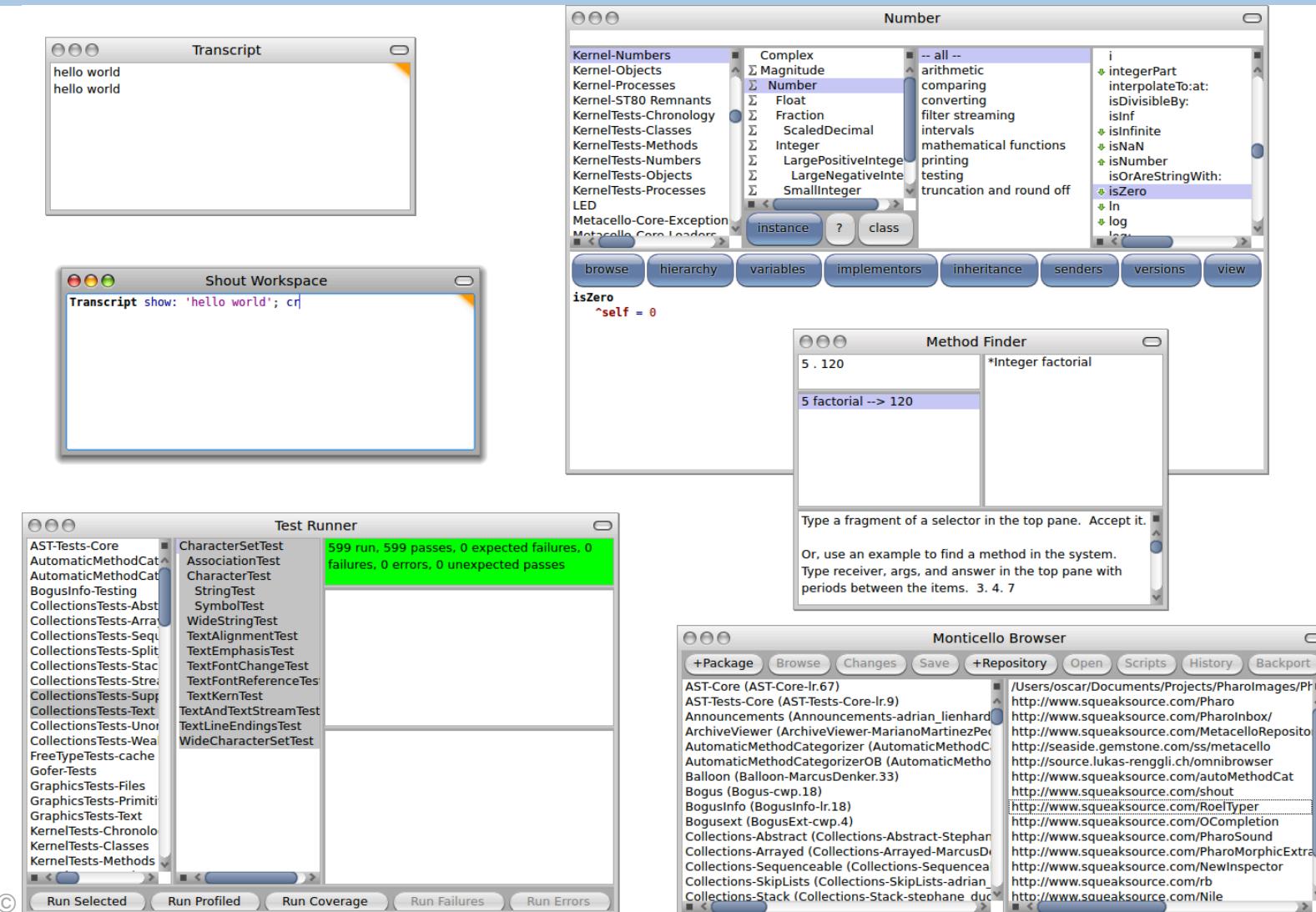
Mouse Semantics



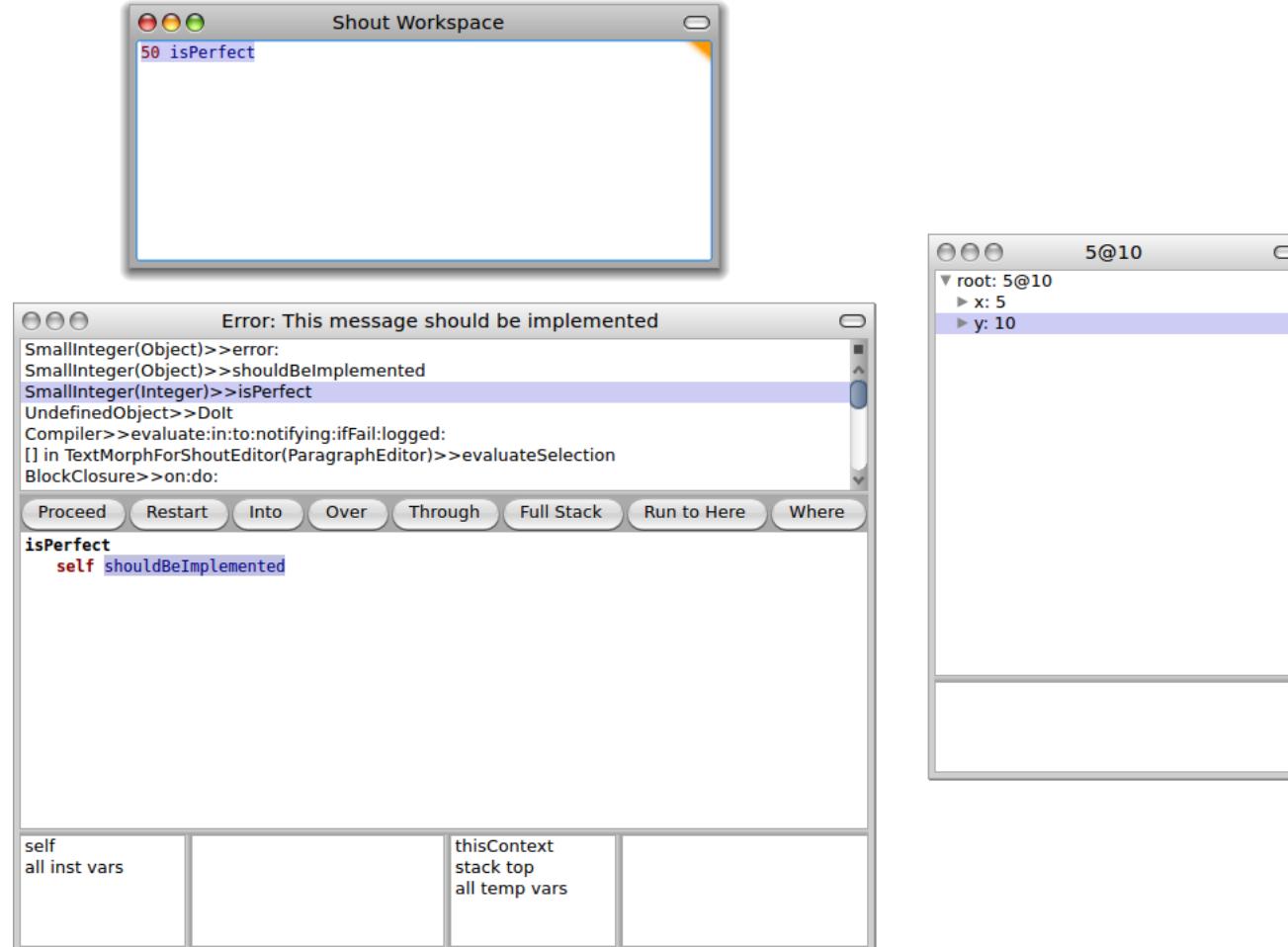
World Menu



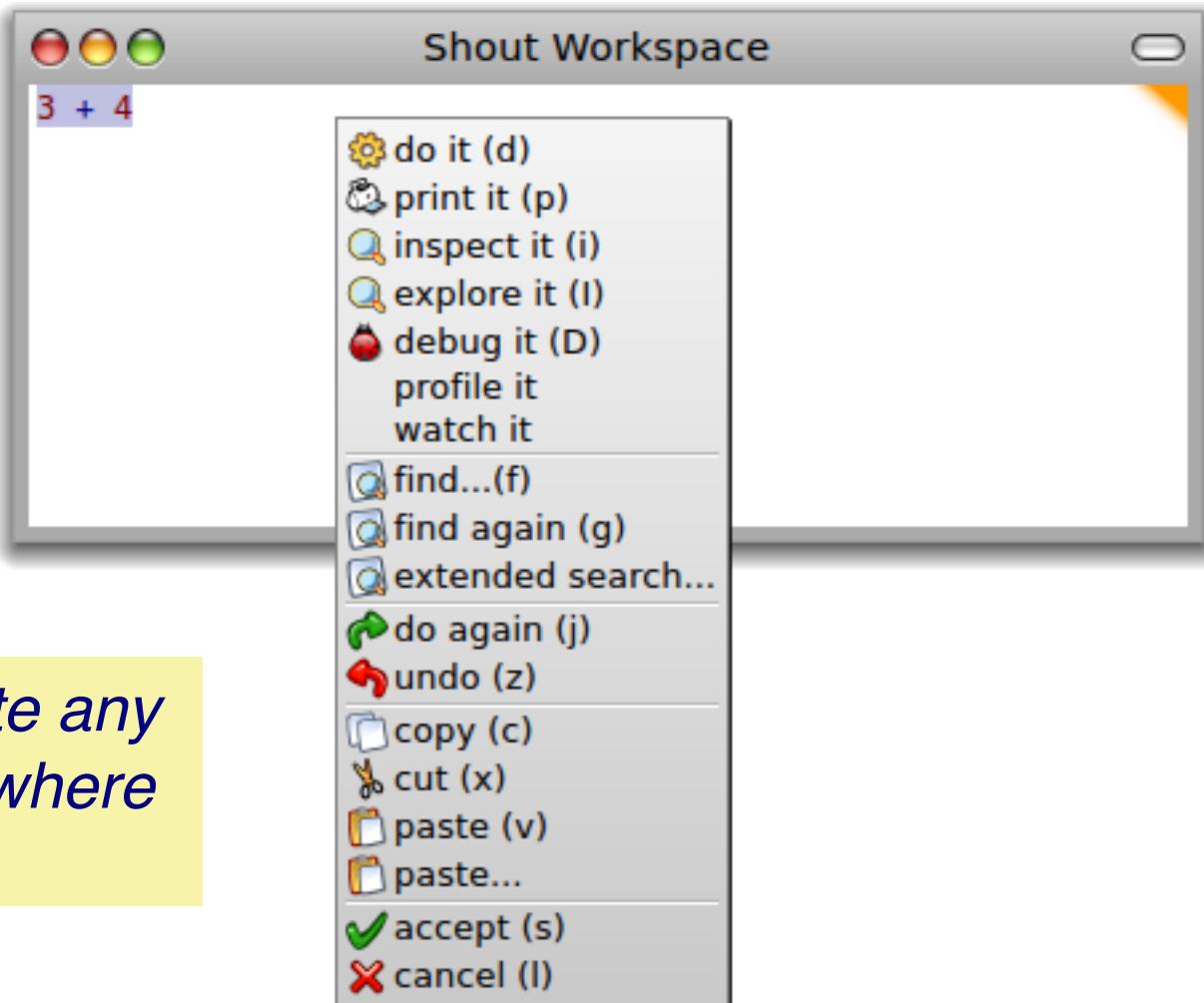
Standard development tools



Debuggers, Inspectors, Explorers



Do it, Print it, ...



You can evaluate any expression anywhere in Smalltalk.

Roadmap

- > The origins of Smalltalk
- > What is Smalltalk?
- > **Syntax in a nutshell**
- > Seaside — web development with Smalltalk



Three kinds of messages

> *Unary messages*

```
5 factorial  
Transcript cr
```

> *Binary messages*

```
3 + 4
```

> *Keyword messages*

```
3 raisedTo: 10 modulo: 5
```

```
Transcript show: 'hello world'
```

Precedence

First unary, then binary, then keyword:

```
2 raisedTo: 1 + 3 factorial
```

128

Same as:

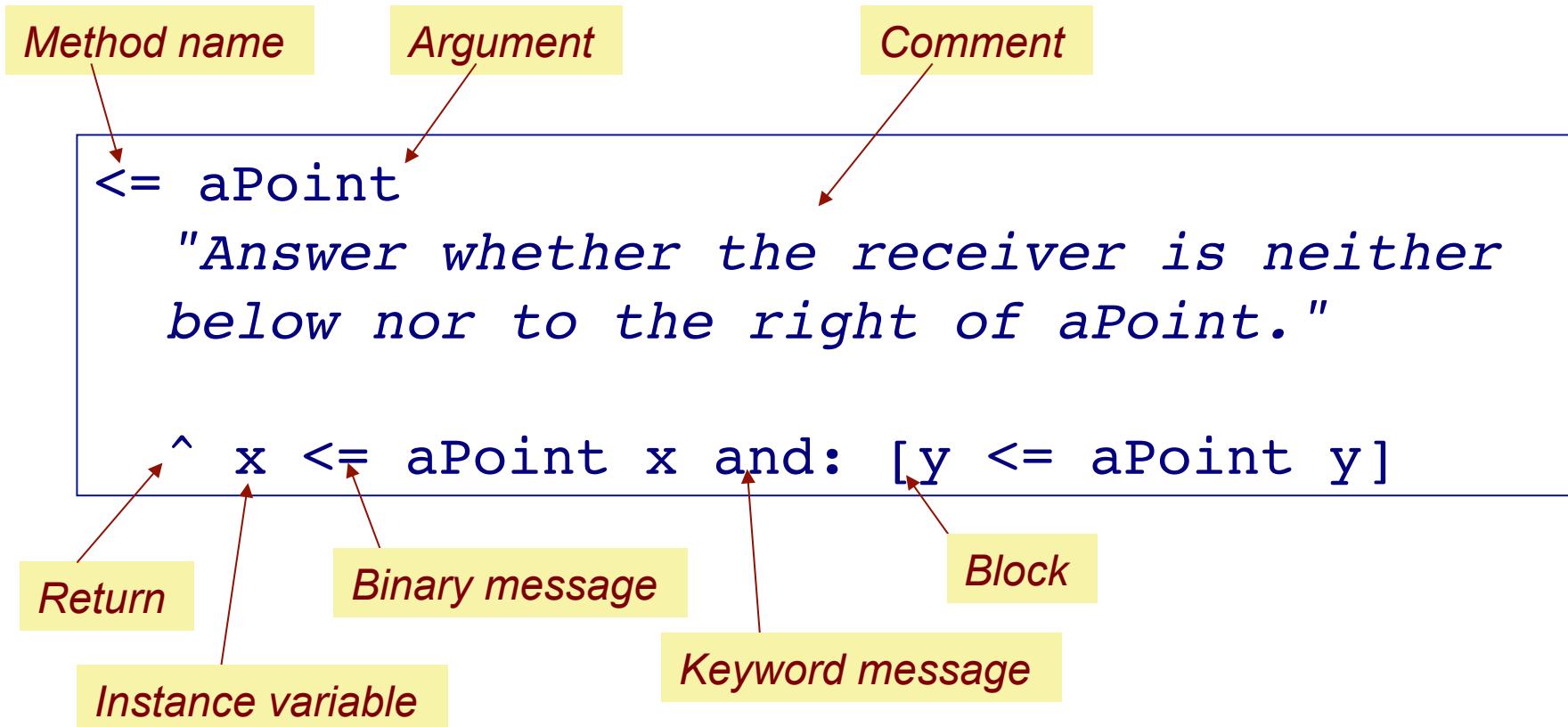
```
2 raisedTo: (1 + (3 factorial))
```

Use parentheses to force order:

```
1 + 2 * 3  
1 + (2 * 3)
```

9 (!)
7

A typical method in the class Point



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

`true`

Statements and cascades

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

Temporary variables

Statement

Assignment

Cascade

The diagram illustrates a segment of Smalltalk code with various linguistic features highlighted:

- Temporary variables**: Points to the parameter declaration `| p pen |`.
- Statement**: Points to the final line of the code block, which includes multiple method invocations: `pen goto: p; down; goto: p+p`.
- Assignment**: Points to the assignment statement `p := 100@100.`.
- Cascade**: Points to the final line of the code block, specifically highlighting the cascade of method invocations: `pen goto: p; down; goto: p+p`.

Literals and constants

Strings & Characters	'hello'	\$a
Numbers	1	3.14159
Symbols	#yada	yada
Arrays	#(1 2 3)	
Pseudo-variables	self	super
Constants	true	false

Variables

- > *Local variables* are delimited by `| var|`
Block variables by `:var|`

```
OrderedCollection>>collect: aBlock
    "Evaluate aBlock with each of my elements as the argument."
    | newCollection |
    newCollection := self species new: self size.
    firstIndex to: lastIndex do:
        [ :index |
            newCollection addLast: (aBlock value: (array at: index))].
    ^ newCollection
```

```
(OrderedCollection with: 10 with: 5) collect: [:each| each factorial ]
```

an OrderedCollection(3628800 120)

Control Structures

- > *Every control structure is realized by message sends*

```
max: aNumber
  ^ self < aNumber
    ifTrue: [aNumber]
    ifFalse: [self]
```

```
4 timesRepeat: [Beeper beep]
```

Creating objects

- > *Class methods*

```
OrderedCollection new  
Array with: 1 with: 2
```

- > *Factory methods*

```
1@2  
1/2
```

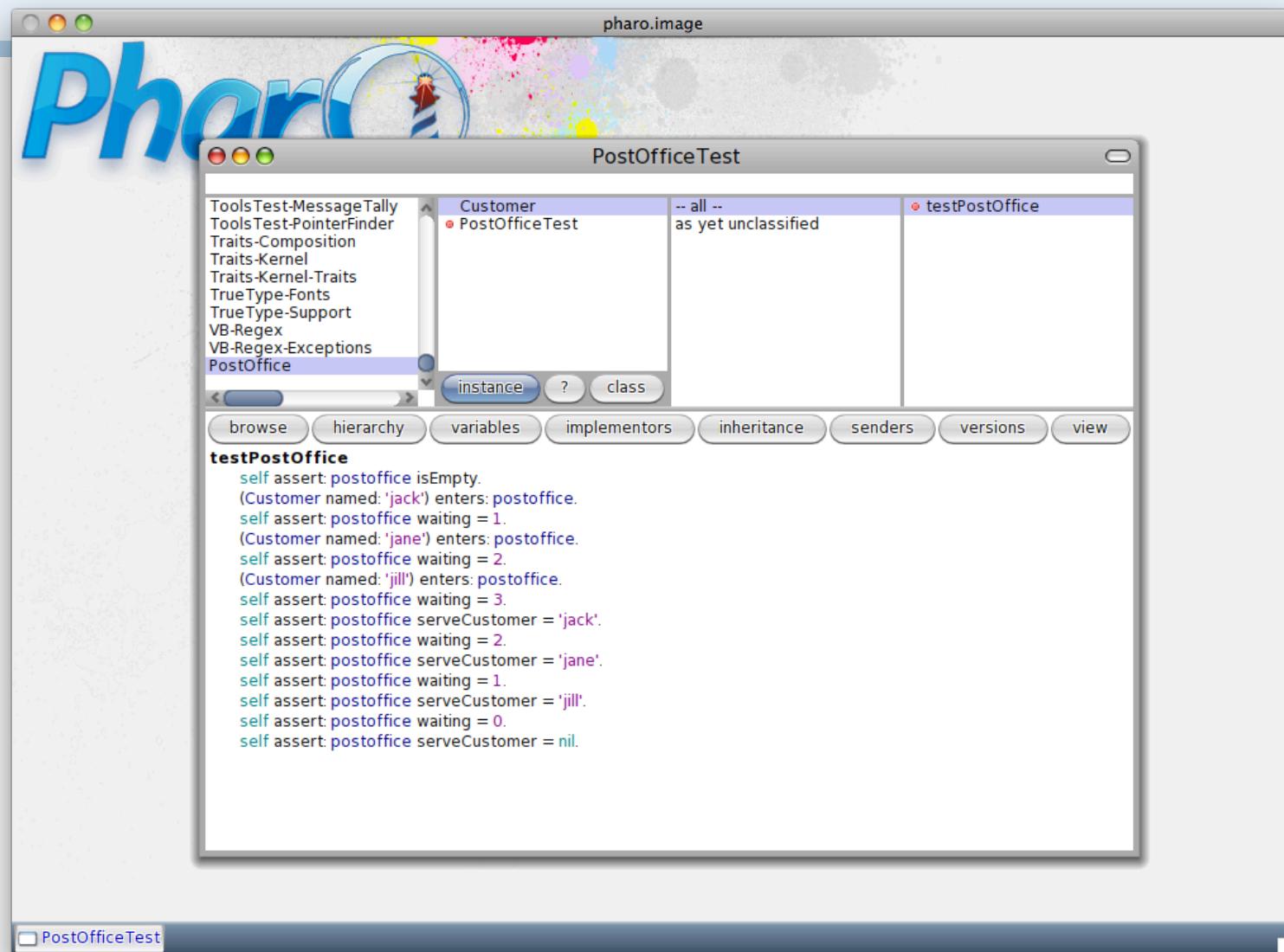
```
a Point  
a Fraction
```

Creating classes

- > Send a message to a class (!)

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

Demo: Defining classes and methods



Roadmap

- > The origins of Smalltalk
- > What is Smalltalk?
- > Syntax in a nutshell
- > **Seaside — web development with Smalltalk**

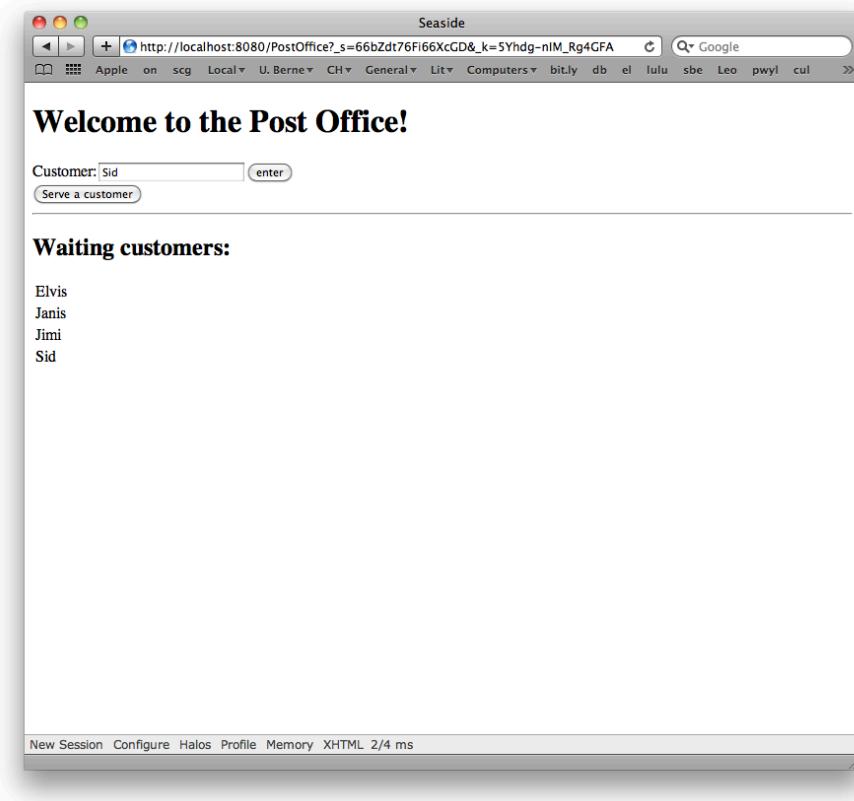
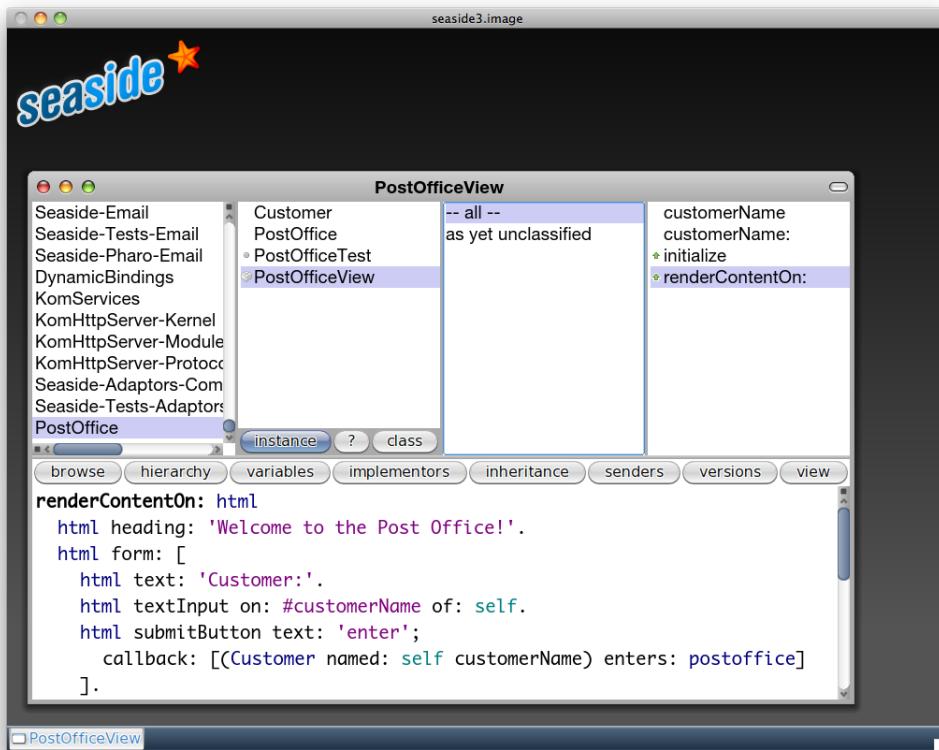


Seaside – a Smalltalk web development platform

The image displays four screenshots of web applications built with the Seaside framework:

- seaside.st Home:** The main homepage featuring the Seaside logo, a star icon, and a banner stating "The framework for developing sophisticated web applications in Smalltalk". It includes sections for About, Community, News, and Documentation.
- SqueakSource:** A project management and code repository site. It shows a navigation bar with Home, Projects, Tags, Members, Groups, and Help. The main content area discusses the service's purpose and features like RSS feeds and authentication.
- CMS BOX – CONTENT MANAGEMENT SYSTEM:** A demo page for CMSBOX. It features a banner with "NEW & IMPROVED", "APPLICATION VERSIONS", and "cmsbox". Below it, there's a section titled "Cmsbox – systematic content management" with a screenshot of the CMS interface showing a search bar and a list of items.
- Dabble DB – Online Database:** A demo page for Dabble DB. It shows a database table titled "Unseen View" with columns for "New entry", "Sale", "Category", "Address", "Close Date", "Probability", "Value", and "Count". The table lists various entries such as "Catering", "Restaurant", "Kiosk", and "Shop".

Demo: PostOffice in Seaside



What you should know!

- ☞ *What are the key differences between Smalltalk, C++ and Java?*
- ☞ *What is at the root of the Smalltalk class hierarchy?*
- ☞ *What kinds of messages can one send to objects?*
- ☞ *What is a cascade?*
- ☞ *Why does $1+2 / 3 = 1$ in Smalltalk?*
- ☞ *How are control structures realized?*
- ☞ *How is a new class created?*
- ☞ *What are categories for?*
- ☞ *What are Factory methods? When are they useful?*

Can you answer these questions?

- ☞ *Which is faster, a program written in Smalltalk, C++ or Java?*
- ☞ *Which is faster to develop & debug, a program written in Smalltalk, C++ or Java?*
- ☞ *How are Booleans implemented?*
- ☞ *Is a comment an Object? How would you check this?*
- ☞ *What is the equivalent of a static method in Smalltalk?*
- ☞ *How do you make methods private in Smalltalk?*
- ☞ *What is the difference between = and ==?*
- ☞ *If classes are objects too, what classes are they instances of?*

License

<http://creativecommons.org/licenses/by-sa/2.5/>



You are free:

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions:



Attribution. You must attribute the work in the manner specified by the author or licensor.



Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.