A clean, innovative, open-source Smalltalk

http://www.pharo-project.org
Roadmap

Short intro

User community

Look at syntax and runtime

Coding Session

Future
In a nutshell

Pharo = language + IDE + update mechanism
Pure object-oriented programming language
Dynamically typed and trait-based
Open and flexible environment
Platform for Seaside and Aida/Web web frameworks
Pharo?

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

Stable
Bugs fixed fast
But innovative
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...
Smalltalk with OO-Database

Pharo is the IDE
Companies

- netstyle.ch
- cmsbox.com
- Pinesoft
- Smallworks
- Agilitic.be
- Inceptive.be
- GemStone
- SW GmbH
- 2Denker
- ...

...
Universities

Annecy       Bruxelles
Lugano
Bern
Douai
Lille
Santiago
Getting started

Model

Tools

Syntax
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
Syntax in a nutshell
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);
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
\[2 + 3 \text{ squared}\]
$2 + 3$ squared
$> 2 + 9$
2 + 3 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

<table>
<thead>
<tr>
<th>p</th>
<th>pen</th>
</tr>
</thead>
<tbody>
<tr>
<td>p := 100@100.</td>
<td></td>
</tr>
<tr>
<td>pen := Pen new.</td>
<td></td>
</tr>
<tr>
<td>pen up.</td>
<td></td>
</tr>
<tr>
<td>pen goto: p; down; goto: p+p</td>
<td></td>
</tr>
</tbody>
</table>

Temporary variables

Statement

Cascade
Block Closures: aka Function

\[ \text{fct}(x) = x \times x + x \]

\[ |\text{fct}| \]
\[ \text{fct} := [:x \mid x \times x + x]. \]
Function Application

\[ fct (2) = 6 \]
\[ fct (20) = 420 \]

\[ fct \text{ value: 2} \]
\[ > 6 \]
\[ fct \text{ value: 20} \]
\[ > 420 \]

\[ [:x | x * x + x] \text{ value: 2} \]
\[ >6 \]

\[ [:x | x * x + x] \text{ value: 20} \]
\[ >420 \]
#(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]
\[ \text{to: 100 do:} \]
\[ [ :i \mid \text{Transcript show: i \ ; space}] \]
I to: 100 by: 3 do:
   [ :i | Transcript show: i ; space]
\textbf{to: 100 by: 3 do:}

\[ :i \mid \text{Transcript show: } i \ ; \text{ space} \]
A typical method in Point

\[ \leq \text{aPoint} \]

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

\[ ^x \leq \text{aPoint} x \text{ and: } [y \leq \text{aPoint} y] \]

\((2@3) \leq (5@6)\)  \(true\)
Creating classes

Send a message to a class (!)

Number subclass: #Complex

instanceVariableNames: 'real imaginary'

... 

category: 'ComplexNumbers'
Past....

Started with Squeak Smalltalk

+ *Major* Cleanups (MVC, eToys)
+ New UI Look / TrueType
+ Tools
+ *Block Closures*
+ *Lots of bugfixes and small improvements*
+ Preferences clean up
+ *MIT license clean*
Update ~1200 Bug-reports closed

496 Updates (1.0)
61 (1.1 unstable)

Release Candidate: October 2009 ;}
Future
...cleaner

started to run SmallLint... and fix
...smaller
deployment image is 7 MB
...faster
What we dream about

A flexible infrastructure to be able to reinvent itself

Better tools

next generation refactoring

Better infrastructure

New compilers
First class packages
Minimal core

Integration Server....

So that people can invent their future
Everybody can help

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

- FIX/ENHANCEMENT
  - In PharoInbox or Changesets

- Discussed on Mailing-list

- BUG Tracker
  - Described

- BUG

- Other version

- Integrated

- Rejected
Pharo Sprints

May 2008 Bern
July 2009 Bern
October 2009 Lille
November 2009 Buenos Aires

more in the future...
Pharo is a modern open-source development environment for the classic Smalltalk-80 programming language. Despite being the first purely object-oriented language and environment, Smalltalk is in many ways still far ahead of its successors in promoting a vision of an environment where everything is an object, and anything can change at run-time.

Pharo by Example, intended for both students and developers, will guide you gently through the Pharo language and environment by means of a series of examples and exercises.

This book is made available under the Creative Commons Attribution-ShareAlike 3.0 license. You can either download the PDF for free, or you can buy a softcover copy from lulu.com.

Additional material is available from Pharo web site at www.pharo-project.org.

Andrew P. Black, Stéphane Ducasse, Oscar Nierstrasz, Damien Pollet
with Damien Cassou and Marcus Denker

2nd Volume
in preparation
Thanks

Hans Beck
Alexandre Bergel
Cedric Beler
Torsten Bergmann
Matthias Berth
Ralph Boland
Noury Bouraqadi
Brian Brown
Gwenaël Casaccio
Damien Cassou
Nicolas Cellier
Gary Chambers
Miguel Coba
Gabriel Cotelli
Carlos Crosetti
Cyrille Delaunay
Simon Denier
Marcus Denker
Ramiro Diaz Trepot
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
François Stephany
Serge Stinckwich
Mathieu Suen
Lawrence Trutter
Andrew Tween
martin von loewis
Juan Vuletich
Steven Wirts
Hernan Wilkinson
Join Us!

Creating good energy, software quality, learning and having fun

http://pharo-project.org