

 eBook Gratuit

APPRENEZ

pharo

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#pharo

Table des matières

À propos	1
Chapitre 1: Démarrer avec le pharo	2
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Chapitre 2: Bord de mer	3
Introduction.....	3
Exemples.....	3
Droppable.....	3
Chapitre 3: Créer une classe	5
Introduction.....	5
Exemples.....	5
Ajout d'une classe dans un navigateur de système.....	5
Code de création de classe triviale.....	6
Classe anonyme.....	7
Crédits	8

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [pharo](#)

It is an unofficial and free pharo ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official pharo.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec le pharo

Remarques

Pharo fournit un excellent moyen de [lutter contre la complexité](#) du développement de logiciels. Avec son retour rapide avec la compilation instantanée de la méthode en cours d'édition, ses objets depuis le compilateur jusqu'à la pile d'exécution et les outils conçus pour être personnalisables par le développeur individuel, il prend en charge la programmation. Il dispose d'excellents outils de refactoring et facilite la création de langages spécifiques à un domaine. Il valorise l'innovation et l'efficacité par rapport à l'efficacité d'exécution.

Exemples

Installation ou configuration

Installation facile

Allez sur <http://pharo.org/download> et sélectionnez un téléchargement adapté et lancez-le.

Détails

Il existe de nombreuses manières d'installer Pharo. Pharo lui-même consiste en une vm et une image. De plus, il a besoin de ses sources et plugins, et a quelques dépendances:

- Il s'agit d'un environnement multi-plateforme, fonctionnant sous OS-X (et iOS), Windows et plusieurs variantes d'Unix (notamment Ubuntu et Android).
- Il s'exécute sur une machine virtuelle pouvant fonctionner sur plusieurs architectures de processeur (Intel, ARM). La machine virtuelle est partagée avec Squeak, Cuis et Newspeak. Avec Pharo 5, une nouvelle vm beaucoup plus rapide a été introduite en utilisant un format d'image et un FFI différents.
- Il existe des variantes 32 bits et 64 bits.
- En plus de l'image standard, PharoLauncher s'intègre à notre infrastructure CI et prend en charge le téléchargement et l'exécution de toutes sortes d'images, notamment celles en bord de mer, magritte et original préconfigurées, les anciennes versions et les dernières versions de développement de Pharo.

Lire [Démarrer avec le pharo en ligne](https://riptutorial.com/fr/pharo/topic/8799/demarrer-avec-le-pharo): <https://riptutorial.com/fr/pharo/topic/8799/demarrer-avec-le-pharo>

Chapitre 2: Bord de mer

Introduction

Seaside est un cadre web pour Pharo et autres petites entreprises. Il est idéal pour les applications complexes avec un modèle de domaine riche.

Exemples

Draggable

Un composant Seaside (sous-classe de `WAComponent`) doit remplacer `#renderContentOn:`. C'est une classe smalltalk qui peut utiliser tous les moyens normaux de structurer une application. Ici, il délègue à trois méthodes différentes.

```
JQDraggableFunctionalTest>>renderContentOn: html
  self renderInstructionsOn: html.
  self renderInventoryOn: html.
  self renderSelectedOn: html
```

En tant que paramètre, il obtient un objet toile HTML qui comprend les messages relatifs à la construction du code HTML et du javascript. Il utilise une interface fluide, où `#with:` est le dernier message envoyé au contexte de canevas actuel.

```
JQDraggableFunctionalTest>>renderInventoryOn: html
html div class: 'inventory ui-corner-all'; with: [
  self colors do: [ :each |
    html div
      class: each;
      passenger: each;
      script: (html jQuery new draggable
        revert: true) ] ]

JQDraggableFunctionalTest>>renderSelectedOn: html
html div
  class: 'selected ui-corner-all';
  script: (html jQuery new draggable
    onDrop: (html jQuery this load
      callbackDraggable: [ :event |
        self selected add: (event at: #draggable) ]);
    html: [ :r | self renderSelectedItemsOn: r ]));
  with: [ self renderSelectedItemsOn: html ]

JQDraggableFunctionalTest>>renderSelectedItemsOn: html
self selected do: [ :each |
  html div
    class: each;
    passenger: each;
    script: (html jQuery new draggable
      onStop: (html jQuery this effect
        percent: 1; puff;
```

```
onComplete: (html jQuery this parent load html: [ :r |  
  self selected remove: each ifAbsent: [ ].  
  self renderSelectedItemOn: r ]))) ]
```

Lire Bord de mer en ligne: <https://riptutorial.com/fr/pharo/topic/8805/bord-de-mer>

Chapitre 3: Créer une classe

Introduction

Créer une classe dans Pharo est aussi simple que d'envoyer une `subclass:` message à un autre objet de classe. Mais la plupart des classes sont créées et modifiées dans un navigateur de système (actuellement Nautilus).

Exemples

Ajout d'une classe dans un navigateur de système

La manière la plus courante d'ajouter (et d'éditer des classes) depuis le navigateur du système Dans le navigateur du système Nautilus, ne sélectionnez rien ou sélectionnez un package (première colonne) ou une classe (deuxième colonne). En fonction de la sélection, l'éditeur de code affichera un modèle de classe légèrement différent:

Sélection	Modèle
Aucun	Modèle de classe vide
Paquet	Un modèle de classe avec un nom de package prédéfini (basé sur le package sélectionné)
Classe	Une définition réelle de la classe sélectionnée

L'image suivante illustre la fenêtre Nautilus avec un package sélectionné:

Navigation pane showing a project structure and a class list.

Buttons: Scoped, Variables

Type: Pkg1|^Pkg2|Pk.*Core\$

- BlueInk-Extras
- BlueInk-Tests
- CodeExport
- CodeImport**
- CodeImportCommandLi
- Collections-Abstract
- Collections-Arithmetic
- Collections-Atomic
- Collections-DoubleLinke
- Collections-DoubleLinke
- Collections-Native
- Collections-Sequenceab
- Collections-Stack

- ChunkFileFormatParser
- ChunkReadStream
- CodeChunk
- BehaviorOwnedChunk
- ClassCommentChunk
- ClassOrganizationChunk
- MethodChunk
- DoltChunk
- StyleChunk
- CodeImporter
- FileCompilerRequestor

Buttons: Hier., Class

```
Object subclass: #NameOfSubclass
  instanceVariableNames: ''
  classVariableNames: ''
  package: 'CodeImport'
```

. Si vous ne possédez pas de superclasse particulière dans votre esprit, il est conseillé de sous-classe de `Object` mais cela amène une confusion tautologique dans l'exemple précédent. Disons que vous voulez créer une classe *PriorityStack* avec la classe *Stack* en tant que super-classe, vous devez alors évaluer:

```
Stack subclass: #PriorityStack
```

Classe anonyme

Vous pouvez créer des classes sans noms qui ne sont pas installés dans le système en envoyant `newAnonymousSubclass` à une classe.

Par exemple

```
anonymousSet := Set newAnonymousSubclass
```

assignera une sous-classe `anonymousSet` variable `Set` à `anonymousSet`. Ensuite, vous pouvez compiler des méthodes dans cette classe et les instancier ou les échanger avec une classe réelle.

Utile pour tester les ressources de pour le proxy

Lire **Créer une classe en ligne**: <https://riptutorial.com/fr/pharo/topic/8834/creer-une-classe>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec le pharo	Community , Stephan Eggermont
2	Bord de mer	Stephan Eggermont
3	Créer une classe	Uko