



EBook Gratis

APRENDIZAJE pharo

Free unaffiliated eBook created from
Stack Overflow contributors.

#pharo

Tabla de contenido

Acerca de	1
Capítulo 1: Empezando con pharo	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Capítulo 2: Creando una clase	3
Introducción.....	3
Examples.....	3
Agregar una clase en un navegador de sistema.....	3
Código de creación de clase trivial.....	4
Clase anonima.....	5
Capítulo 3: Playa	6
Introducción.....	6
Examples.....	6
Dropable.....	6
Creditos	8

Acerca de

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

Capítulo 1: Empezando con pharo

Observaciones

Pharo proporciona una excelente manera de [combatir la complejidad](#) en el desarrollo de software. Con su rápida retroalimentación con la compilación instantánea del método editado actualmente, sus objetos desde el compilador hasta la pila en tiempo de ejecución, y las herramientas diseñadas para ser personalizadas por el desarrollador individual, admite la programación para salir de un lío nuevamente. Cuenta con excelentes herramientas de refactorización y facilita la creación de lenguajes específicos de dominio. Se valora la innovación y la eficacia en el tiempo de ejecución.

Examples

Instalación o configuración

Instalación fácil

Vaya a <http://pharo.org/download> y seleccione una descarga apropiada y ejecútela.

Detalles

Hay muchas maneras diferentes de instalar Pharo. Pharo en sí consiste en una vm y una imagen. Además necesita sus fuentes y complementos, y tiene algunas dependencias:

- Es un entorno multiplataforma, que se ejecuta en OS-X (y iOS), Windows y varias variantes de Unix (por ejemplo, Ubuntu y Android).
- Se ejecuta en una máquina virtual que puede ejecutarse en varias arquitecturas de procesador (Intel, ARM). La máquina virtual se comparte con Squeak, Cuis y Newspeak. Con Pharo 5, se ha introducido una nueva y mucho más vm utilizando un formato de imagen y FFI diferentes.
- Hay variantes de 32 bits y 64 bits.
- Además de la imagen estándar, está el PharoLauncher que se integra con nuestra infraestructura de CI y admite la descarga y ejecución de todo tipo de imágenes, como las de playa, magritte y alce preconfiguradas, versiones anteriores y las últimas versiones de desarrollo de Pharo.

Lea Empezando con pharo en línea: <https://riptutorial.com/es/pharo/topic/8799/empezando-con-pharo>

Capítulo 2: Creando una clase

Introducción

Crear una clase en Pharo es tan simple como enviar una `subclass:` mensaje a otro objeto de clase. Pero la mayoría de las clases se crean y modifican en un navegador del sistema (actualmente Nautilus).

Examples

Agregar una clase en un navegador de sistema

La forma más común de agregar (y editar clases) desde el navegador del sistema En el navegador del sistema Nautilus no tiene nada seleccionado o seleccione un paquete (primera columna) o una clase (segunda columna). Dependiendo de la selección, el editor de código mostrará una plantilla de clase ligeramente diferente:

Selección	Modelo
Ninguna	Plantilla de clase vacía
Paquete	Una plantilla de clase con nombre de paquete precargado (basado en el paquete seleccionado)
Clase	Una definición real de la clase seleccionada

La siguiente imagen muestra la ventana de Nautilus con un paquete seleccionado:

Variables

- 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

Hier.

Class

?

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

. Si no tiene ninguna superclase en particular en su mente, se recomienda hacer una subclase de `Object` pero esto trae una confusión tautológica en el ejemplo anterior. Digamos que desea crear una clase *PriorityStack* con la clase *Stack* como una superclase, entonces necesita evaluar:

```
Stack subclass: #PriorityStack
```

Clase anonima

Puede crear clases sin nombres que no estén instalados en el sistema enviando `newAnonymousSubclass` a una clase.

Por ejemplo

```
anonymousSet := Set newAnonymousSubclass
```

asignará una subclase anónima de Establecer a la variable `anonymousSet`. Luego, puede compilar métodos en esta clase e instanciarlos, o intercambiarlos con una clase real.

Útil para recursos de prueba de proxying

Lea Creando una clase en línea: <https://riptutorial.com/es/pharo/topic/8834/creando-una-clase>

Capítulo 3: Playa

Introducción

Seaside es un framework web para Pharo y otros pequeños. Es ideal para aplicaciones complejas con un modelo de dominio rico.

Examples

Dropable

Un componente Seaside (subclase de `WACComponent`) debe reemplazar `#renderContentOn:`. Es una clase pequeña que puede usar todas las formas normales de estructurar una aplicación. Aquí se delega a tres métodos diferentes.

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

Como parámetro, obtiene un objeto de lienzo html que entiende los mensajes relevantes para crear el html y el javascript. Utiliza una interfaz fluida, donde `#with:` es el último mensaje enviado al contexto del lienzo actual.

```
JQDropableFunctionalTest>>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) ] ]

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

JQDropableFunctionalTest>>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 ]))) ]
```

Lea Playa en línea: <https://riptutorial.com/es/pharo/topic/8805/playa>

Creditos

S. No	Capítulos	Contributors
1	Empezando con pharo	Community , Stephan Eggermont
2	Creando una clase	Uko
3	Playa	Stephan Eggermont