



17 de noviembre de 2009

## Groovy Goodness: ejecutar secuencias de comandos Groovy en Java con GroovyClassLoader

Con el `GroovyClassLoader` podemos cargar scripts Groovy y ejecutarlos en código Java. Primero debemos crear un nuevo `GroovyClassLoader` luego analizar un script Groovy. El script puede estar en un archivo, cadena o flujo de entrada. Una vez que se analiza el script, tenemos un `Classy` podemos hacer una nueva instancia de esta clase. Transmitimos la instancia a una `GroovyObject` instancia, por lo que podemos usar el `invokeMethod()` método para invocar métodos en nuestro script Groovy.

```
0 package com.mrhaki.blog;
1
2 import groovy.lang.GroovyClassLoader;
3 import groovy.lang.GroovyObject;
4
5 import java.io.File;
6 import java.io.IOException;
7
8 public class GroovyRun {
9     public static void main(final String[] args) throws Illegal
10         // Create GroovyClassLoader.
11         final GroovyClassLoader classLoader = new GroovyClass
12
13         // Create a String with Groovy code.
14         final StringBuilder groovyScript = new StringBuilder(
15         groovyScript.append("class Sample {");
16         groovyScript.append("    String sayIt(name) { \"Groovy
17         groovyScript.append("}");
18
19         // Load string as Groovy script class.
20         Class groovy = classLoader.parseClass(groovyScript.to
21         GroovyObject groovyObj = (GroovyObject) groovy.newIns
22         String output = groovyObj.invokeMethod("sayIt", new O
23         assert "Groovy says: Cool mrhaki!".equals(output);
24
25         // Load Groovy script file.
26         groovy = classLoader.parseClass(new File("SampleScrip
27         groovyObj = (GroovyObject) groovy.newInstance();
```

```
28 |         output = groovyObj.invokeMethod("scriptSays", new Obj
29 |         assert "Hello mrhaki, from Groovy. Hello mrhaki, from
30 |     }
31 | }
```

```
0 | // File: SampleScript.groovy
1 | class SampleScript {
2 |     String scriptSays(name, num) {
3 |         "Hello $name, from Groovy. " * num
4 |     }
5 | }
```

[Tweet](#)

Publicado por Hubert Klein Ikink

3 Comentarios

[Groovy](#) [Groovy programación](#) [Groovy: Bondad](#) [GroovyBuena: Intermedia](#) [GroovyBuena: Java](#)

**Comments****Community**
**Recommend** 1

 **Tweet**
**Share**
**Sort by Best** ▾

Join the discussion...

**Gu Ru** • a year ago

IS:

```
assert "Hello mrhaki, from Groovy. Hello mrhaki,
from Groovy".equals(output);
```

SHOULD BE:

```
assert "Hello mrhaki, from Groovy. Hello mrhaki,
from Groovy. ".equals(output);
```

| • Reply • Share ›

**Hubert Klein Ikkink** Mod → **Gu Ru**

• a year ago

Hi, thank you for finding this error. I've fixed it in the blog post.

| • Reply • Share ›

**Johann** • 10 years ago

There's also the *Eval* class with nice methods, such as *Eval.x*.

Very helpful, too.

| • Reply • Share ›

**ALSO ON MESSAGES FROM MRHAKI****Awesome AsciiDoctor: Prevent Transformation**

1 comment • 2 years ago



**Abis Mal** — True enough! The email and other links were duly

**Micronaut Mastery: Decode JSON Using**

1 comment • a year ago



**jardako** — Java is so much uglier than Groovy

**Groovy Goodness: Intersect Collections****Micronaut Mastery: Binding Request**

## Artículos Relacionados

---

 Bondad maravillosa: Convierta los métodos en cierres (puntaje de coincidencia 53%)
 

---

 Grassroots Groovy: the Spring Route (part 1) (Puntuación equivalente al 38%)
 

---

 Componente Wicket para Java Deployment Toolkit para ejecutar applets (puntaje de coincidencia 36%)
 

---

 Grassroots Groovy: Parse XML con XmlSlurper de Java (Puntuación equivalente al 36%)
 

---

 Groovy Goodness: BaseScript con Abstract Run Script Method (puntaje de coincidencia 33%)
 

---

## Más publicaciones Groovy: Bondad

---

Bondad maravillosa: utilizar variables expandidas en la consulta SQL GString

---

Bondad maravillosa: preordenar y posordenar el recorrido del árbol

---

Bondad maravillosa: tuplas con hasta 9 artículos

---

Groovy Goodness: Creación de un método adicional que admite argumentos con nombre usando la anotación @NamedVariant

---

Bondad maravillosa: colecciones no modificables

---

Groovy Goodness: Agregar un constructor de mapas con anotaciones

---

Groovy Goodness: Implemente la interfaz y los métodos abstractos automáticamente

---

Maravillosa bondad: Personalizando la salida JSON

---

Bondad maravillosa: eliminar el último elemento de la lista con el método RemoveLast (y los métodos Pop / Push reimplementados)

---

Bondad maravillosa: obtener todos los valores iniciales y finales de forma recursiva

---

Bondad maravillosa: valores truncados y redondos BigDecimal

---

Groovy Goodness: entrecruza colecciones con un comparador personalizado

---

Bondad maravillosa: Creación fácil de objetos con método Tap

---

Bondad maravillosa: ¿dónde está mi clase?

---

Bondad maravillosa: calcule los valores de hash MD5 y SHA

---

Bondad maravillosa: codificación segura de URL y nombre de archivo Base64

---

Groovy Goodness: Mejoras de Java 8 Stream

---

Bondad maravillosa: uso opcional en contexto condicional

---

Bondad maravillosa: uso de valores de cadena en rangos

---

Bondad maravillosa: utilice el rango por método para establecer pasos entre números

---

Groovy Goodness: Descargar Grab Dependencies en IntelliJ IDEA

---

Bondad maravillosa: asegúrese de que los objetos que se pueden cerrar se cierren con el método de cierre

---

Bondad maravillosa: redireccionar los métodos de impresión en los scripts

---

Bondad maravillosa: uso del operador de llamadas ()

---

Groovy Goodness: Creando una matriz JSON raíz con JsonBuilder

---

Groovy Goodness: Uncapitalize Strings

---

Groovy Goodness Notebook se actualiza

---

Bondad maravillosa: cierre de identidad

---

Bondad maravillosa: sueño interrumpido

---

Bondad maravillosa: acceso directo al campo en clases (super)

---

Groovy Goodness: Personalice anotaciones de registro AST

---

Bondad maravillosa: convertir un mapa o lista como una cadena en un mapa o lista

---

Bondad maravillosa: representar el mapa como una cadena

---

Bondad maravillosa: formato IntelliJ IDEA de cadenas de cierre

---

Bondad maravillosa: haga que la clase se pueda clonar con @AutoClone

---

Groovy Goodness: Creación de archivos y directorios con buen DSL usando FileTreeBuilder

---

Bondad maravillosa: uso de tuplas

---

Groovy Goodness Notebook se actualiza

---

---

Groovy Goodness: Personalización del compilador Groovy

---

Groovy Goodness: Personalice la salida de Groovydoc con Gradle

---

Bondad maravillosa: definición de campos constantes accesibles al público

---

Bondad maravillosa: Convierta los parámetros del método en argumentos de mapa nombrados con IntelliJ IDEA

---

Bondad maravillosa: convierta el mapa en clase con IntelliJ IDEA

---

Bondad maravillosa: IntelliJ IDEA Intenciones para valores de cadena

---

Bondad maravillosa: excluir dependencias transitivas con uva

---

Groovy Goodness: vea más información sobre la descarga con Grape

---

Groovy Goodness: Cambiar directorio para guardar dependencias Uva

---

Bondad maravillosa: sobrecarga del operador en reversa

---

Bondad maravillosa: el método de inspección devuelve valores de objeto con un formato agradable

---

Bondad maravillosa: eliminar elementos de una colección

---

Bondad maravillosa: comparta datos en un entorno concurrente con variables de flujo de datos

---

Groovy Goodness: Use Closures como Java Lambda Expressions

---

Bondad maravillosa: nuevos métodos para ordenar y eliminar duplicados de la colección

---

Maravillosa bondad: Combina elementos Iterables con índice

---

Bondad maravillosa: intercambiando elementos en una lista

---

Bondad maravillosa: usar el constructor como puntero de método

---

Bondad maravillosa: Acceda a la API XML-RPC

---

Bondad maravillosa: obtener los índices de una colección

---

Bondad maravillosa: Pop y elementos de inserción en una lista

---

Groovy Goodness: Obteniendo todo menos el último elemento en una colección con el método Init

---

Bondad maravillosa: tomar o soltar los últimos artículos de una colección

---

Bondad maravillosa: el cierre como clase

---

Bondad maravillosa: uso de diseños con MarkupTemplateEngine

---

Bondad maravillosa: plantillas anidadas con MarkupTemplateEngine

---

Bondad maravillosa: usar clase de plantilla personalizada con MarkupTemplateEngine

---

Groovy Goodness: Relájate ... Groovy analizará a tu malvado JSON

---

Groovy Goodness: Implementando rasgos en tiempo de ejecución

---

Bondad maravillosa: rasgos de encadenamiento

---

Bondad maravillosa: la propiedad Comprobar configuración se establece en ConfigObject

---

Groovy Goodness: Extend ConfigSlurper with Custom Environments Sections

---

Groovy Goodness: Use Sortable Annotation to Make Classes Comparable

---

Groovy Goodness: BaseScript with Abstract Run Script Method

---

Groovy Goodness: @Builder Definition with Extra Type Checks

---

Groovy Goodness: Using Builder to Create Fluent API for Other Classes

---

Groovy Goodness: Use Builder AST Transformation for Fluent API

---

Groovy Goodness: Extra Methods for NIO Path

---

Groovy Goodness: More Efficient Tail Recursion With TailRecursive Annotation

---

Groovy Goodness: Customize ToString Creation

---

Groovy Goodness: Define Compilation Customizers With Builder Syntax

---

Groovy Goodness: Restricting Script Syntax With SecureASTCustomizer

---

Groovy Goodness: Closure as Writable

---

Groovy Goodness: GString as Writable

---

Groovy Goodness: Converting Byte Array to Hex String

---

Groovy Goodness: Using Package Scoped Methods, Fields and Classes

---

Groovy Goodness Notebook Updated with New Groovy 2.2 Features

---

Groovy Goodness: Enhancements for Iterable Implementations

---

Groovy Goodness: Set Delegating Class for Script

---

Groovy Goodness: Set Script Class with BaseScript Annotation

---

Groovy Goodness: Cache Methods Invocations with Memoize Annotation

---

Groovy Goodness: Implicit Closure Coercion

---

Groovy Goodness: Create CopyWith Method with Immutable Annotation

---

Groovy Goodness: Choose Which Delegated Methods to Use

---

Groovy Goodness: Create New Date or Calendar from Existing and Set Property Value

---

Groovy Goodness: Loop Through Date and Calendar Ranges

---

Groovy Goodness: Remove Part of String With Regular Expression Pattern

---

Groovy Goodness: Running Scripts from a JAR Archive

---

Groovy Goodness: Check if a String Only Contains Whitespaces

---

Groovy Goodness: Replace Characters in a String with CollectReplacements

---

Groovy Goodness Notebook is Published

---

Groovy Goodness: Customize Log Variable Name with Log AST Annotations

---

Groovy Goodness: @DelegatesTo For Type Checking DSL

---

Groovy Goodness: Combining Annotations with AnnotationCollector

---

Groovy Goodness: Apply Mixin to Object Instances

---

Groovy Goodness: Adding Extra Methods Using Extension Modules

---

Groovy Goodness: Append Values to Appendable Objects

---

Groovy Goodness: Calculating Directory Size

---

Groovy Goodness: Pretty Print XML

---

Groovy Goodness: Create a List with Default Values

---

Groovy Goodness: Getting the First and Last Element of an Iterable

---

Groovy Goodness: Using Groovy for Git Hooks

---

Groovy Goodness: Using Project Coin Features Also With Older Java Versions

---

Groovy Goodness: Return Closure From Another Closure or Method

---

Groovy Goodness: Drop or Take Elements with Condition

---

Groovy Goodness: Using Implicit call() Method

---

Groovy Goodness: Boolean Implications

---

Groovy Goodness: Partial Matches

---

Groovy Goodness: Multiple Overloaded Operator Methods for Nice API

---

Groovy Goodness: Revisited Getting the Sum of Items in a Collection

---

Groovy Goodness: Collate a List into Sub-lists

---

Groovy Goodness: Solve Naming Conflicts with Builders

---

Groovy Goodness: Magic Package to Add Custom MetaClass

---

Groovy Goodness: Create Simple Builders with Closures

---

Groovy Goodness: Create Our Own Script Class

---

Groovy Goodness: Parse Date.toString() Value

---

Groovy Goodness: Find Non-Null Results After Transformation in a Collection

---

Groovy Goodness: Customize Groovy Console Visual Output

---

Groovy Goodness: Format Dates with TimeZone

---

Groovy Goodness: Default Groovy Script File Extensions

---

Groovy Goodness: Run Remote Scripts via URL

---

Groovy Goodness: Use inject Method on a Map

---

Groovy Goodness: Access ResultSetMetaData with Groovy SQL

---

Groovy Goodness: Using Named (Ordinal) Parameters with Groovy SQL

---

Groovy Goodness: Use Connection Parameters to Get Text From URL

---

Groovy Goodness: Streaming JSON with StreamingJsonBuilder

---

Groovy Goodness: Transform Items into a Collection with collectMany

---

Groovy Goodness: Collect on Nested Collections

---

Groovy Goodness: GroupBy with Multiple Closures

---

Groovy Goodness: Sort or Remove Duplicates without Changing the Original Collection

---

Groovy Goodness: Take and Drop Items from a List

---

Groovy Goodness: Find Elements Matching Groovy Truth

---

Groovy Goodness: Transform Collection to a Map with collectEntries

---

Groovy Goodness: Add AST Transformations Transparently to Scripts

---

Groovy Goodness: Add Imports Transparently to Scripts with ImportCustomizer

---

Groovy Goodness: Transforming Reader Input to Writer Output

---

Groovy Goodness: Apply Read and Write Locking

---

Groovy Goodness: Command Chain Expressions for Fluid DSLs

---

Groovy Goodness: Change XML Structure

---

Groovy Goodness: Add Java-style Listener Support with ListenerList Annotation

---

Groovy Goodness: Canonical Annotation to Create Mutable Class

---

Groovy Goodness: Paging Support in Groovy SQL

---

Groovy Goodness: Cache Closure Results with Memoization

---

Groovy Goodness: Recursion with Closure Trampoline Capability

---

Groovy Goodness: New Dollar Slashy Strings

---

Groovy Goodness: Chain Closures Together with Closure Composition

---

Groovy Goodness: See if Sets are Equal

---

Groovy Goodness: Add Items to a List at Specified Position

---

Groovy Goodness: Convert Collection to Set with Only Unique Elements

---

Groovy Goodness: Get Unique Characters in a String

---

Groovy Goodness: Generate equals and hashCode Methods with EqualsAndHashCode Annotation

---

Groovy Goodness: Tuple Constructor Creation

---

Groovy Goodness: Change Scope Script Variable with Field Annotation

---

Groovy Goodness: Group and Count Elements in Collection or Map

---

Groovy Goodness: Count Occurrences in a Collection or Map

---

Groovy Goodness: Easy toString Creation for Our Classes

---

Groovy Goodness: Check if Maps are Equal

---

Groovy Goodness: Inject Logging Using Annotations

---

Groovy Goodness: Build JSON with JsonBuilder and Pretty Print JSON Text

---

Groovy Goodness: Parse JSON with JsonSlurper

---

Groovy Goodness: See if List and Object Array are Equal

---

Groovy Goodness: Create GMetrics Report with XSLT

---

Groovy Goodness: Create CodeNarc Reports with XSLT

---

Groovy Goodness: Transform String into Enum

---

Groovy Goodness: Determine Min and Max Entries in a Map

---

Groovy Goodness: Use Map in Switch Statement

---

Groovy Goodness: Convert Date to Calendar

---

Groovy Goodness: Find First Non-Null Result From a Closure

---

Groovy Goodness: Invoke Anonymous Closure

---

Groovy Goodness: String Continuation

---

Groovy Goodness: Use GroovyWS to Access SOAP Web Services (Part 3)

---

Groovy Goodness: Use GroovyWS to Access SOAP Web Services (Part 2)

---

Groovy Goodness: Use GroovyWS to Access SOAP Web Services

---

Groovy Goodness: Convert Date to java.sql.Timestamp

---

Groovy Goodness: Store Closures in Script Binding

---

Groovy Goodness: Resolve Module Location with Grape

---

Groovy Goodness: Process Map Entries in Reverse

---

Groovy Goodness: Subtracting Map Entries

---

Groovy Goodness: Simple File Renaming

---

Groovy Goodness: Groovy Truth for Simple Type Arrays

---

Groovy Goodness: Intersect Maps

---

Groovy Goodness: Use a Class by Proxy

---

Groovy Goodness: Create Indexed Property Getter and Setter Methods

---

Groovy Goodness: Map with Default Values

---

Groovy Goodness: Get to Know More About a GString

---

Groovy Goodness: Clear Time Portion of a Date

---

Groovy Goodness: Synchronized Annotation for Synchronizing Methods

---

Groovy Goodness: Strip Leading Spaces from Lines with Margin

---

Groovy Goodness: Text Translation

---

Groovy Goodness: Represent a List as String

---

Groovy Goodness: Getting Head And Tail From an Array of Objects

---

Groovy Goodness: Inherit Constructors From Parent Classes

---

Groovy Goodness: Create New Date from Old Date with Updates



---

Groovy Goodness: Use the set Method to Define a Date or Calendar Value

---

Groovy Goodness: Setting Date and Calendar Values with Subscript Operators

---

Groovy Goodness: Strip Leading Spaces from Lines

---

Groovy Goodness: Expand or Unexpand Space or Tab Delimited Text

---

Groovy Goodness: Capitalize Strings

---

Groovy Goodness: Get Hints About Missing Methods or Properties

---

Groovy Goodness: Create Class Instance Dynamically

---

Groovy Goodness: Use GrabResolver for Custom Repositories

---

Groovy Goodness: Use Collect with Initial Collection Value

---

Groovy Goodness: Multiple Assignments Revisited

---

Groovy Goodness: Using the Subscript Operator on Iterators

---

Groovy Goodness: Traversing a Directory

---

Groovy Goodness: Working on Files or Directories (or Both) with FileType

---

Groovy Goodness: Refactor XML Creation with StreamingMarkupBuilder

---

Groovy Goodness: Sum the Values in a Object Array

---

Groovy Goodness: Tokenize a String

---

Groovy Goodness: See if String is a Number

---

Groovy Goodness: Adding Maps to Map

---

Groovy Goodness: Sorting a Map

---

Groovy Goodness: New Ways to Curry

---

Groovy Goodness: Configuring Grape to Use Classloader

---

Groovy Goodness: Use Keywords as Method Names

---

Groovy Goodness: Use Strings as Subscript Operator

---

Groovy Goodness: Use Ranges as Subscript Operators

---

Groovy Goodness: Intersect Collections

---

Groovy Goodness: Invoke Methods Dynamically

---

Groovy Goodness: Create Stubs for Testing

---

Groovy Goodness: Round and Truncate Decimal Values

---

Groovy Goodness: Integer Divisions

---

Groovy Goodness: Override getProperty Method with Dynamic Groovy

---

Groovy Goodness: Create Elements in Collection Definition

---

Groovy Goodness: Apply Method to All Elements in a List

---

Groovy Goodness: Is Object in Class Hierarchy

---

Groovy Goodness: Make a Number Absolute

---

Groovy Goodness: Add a ShutdownHook

---

Groovy Goodness: Turn a List into a Map

---

Groovy Goodness: Check if String is a Number

---

Groovy Goodness: Normalize and Denormalize Linefeeds and Carriage Returns

---

Groovy Goodness: Removing Duplicate Elements in a Collection

---

Groovy Goodness: Recursively Collect Elements

---

Groovy Goodness: What Character Are You?

---

Groovy Goodness: Powerful Assert

---

Groovy Goodness: Apply Closure to All Element in a Collection with collect

---

Groovy Goodness: Using the OrderBy Comparator

---

Groovy Goodness: Getting All Subsequences of a List

---

Groovy Goodness: Looping Through Each Permutation in a Collection

---

Groovy Goodness: We Can Handle the Truth

---

Groovy Goodness: Implementing MetaClass Methods with Same Name but Different Arguments

---

Groovy Goodness: Override and Use Old Implementation with Dynamic Methods

---

Groovy Goodness: Adding or Overriding Methods with ExpandoMetaClass

---

Groovy Goodness: Get Properties from ReleaseInfo

---

Groovy Goodness: Getting Groovy with DOM

---

Groovy Goodness: Splitting with Closures

---

Groovy Goodness: Getting Information About Objects

---

Groovy Goodness: Groovy Mystic Expressions

---

Groovy Goodness: Add Static Methods with MetaClass to Classes

---

Groovy Goodness: Info About Closure Parameters

---

Groovy Goodness: Getting the Sum of Items in a Collection

---

Groovy Goodness: Formatted Strings with sprintf

---

Groovy Goodness: Delete Non-empty Directory Recursively with File

---

Groovy Goodness: Type Conversion Using GPathResult When Reading XML

---

Groovy Goodness: Semicolons are Optional

---

Groovy Goodness: Counting the Items in a List

---

Groovy Goodness: Add Methods to Interfaces with ExpandoMetaClass

---

Groovy Goodness: Know More About a Class

---

Groovy Goodness: Using the ServletCategory

---

Groovy Goodness: Implicit Method doCall in Closure

---

Groovy Goodness: Automatic Return Value Casting

---

Groovy Goodness: ServerSocket Scripts

---

Groovy Goodness: Use a Category and Get a Return Value

---

Groovy Goodness: A Bit of metaClass DSL

---

Groovy Goodness: Closure Arguments

---

Groovy Goodness: Intercept Methods with Interceptors

---

Groovy Goodness: Compiled Groovy Script is a Java Class

---

Groovy Goodness: Simple Evaluation of Groovy Expressions in Java

---

Groovy Goodness: Setting a Closure's Delegate

---

Groovy Goodness: Create Dynamic Methods

---

Groovy Goodness: Finding Files with FileNameFinder

---

Groovy Goodness: Are You In? (Membership Operator)

---

Groovy Goodness: Add Methods Dynamically to Classes with ExpandoMetaClass

---

Groovy Goodness: Running a Unit Test

---

Groovy Goodness: Testing for Expected Exceptions

---

Groovy Goodness: Running Groovy Scripts in Java with GroovyClassLoader

---

Groovy Goodness: Source of Groovy GDK

---

Groovy Goodness: Raising the Power of Numbers

---

Groovy Goodness: Run Code at a Specified Time

---

Groovy Goodness: Convert String to Boolean

---

Groovy Goodness: Passing Closures to Methods

---

Groovy Goodness: Building a GUI with SwingBuilder

---

Groovy Goodness: Define Your Own Type Conversion

---

Groovy Goodness: withReader and withWriter

---

Groovy Goodness: Using def to Define a Variable

---

Groovy Goodness: Complex Keys in Maps

---

Groovy Goodness: Variable Scope in Scripts

---

Groovy Goodness: Splitting Strings

---

Groovy Goodness: Base64 Encoding

---

Groovy Goodness: Get Value from Map or a Default Value

---

Groovy Goodness: Remove Parts of a String

---

Groovy Goodness: Working with Lines in Strings

---

Groovy Goodness: Access Strings with Subscript Operator

---

Groovy Goodness: Invoke Methods Dynamically

---

Groovy Goodness: Getting a Submap from a Map

---

Groovy Goodness: Immutable Collections

---

Groovy Goodness: Getting and Setting Properties

---

Groovy Goodness: Groovy SQL DataSet

---

Groovy Goodness: Groovy SQL

---

Groovy Goodness: Optional Parenthesis

---

Groovy Goodness: Static Imports

---

Groovy Goodness: Using the replaceAll Methods from String

---

Groovy Goodness: Groovlets as Lightweight Servlets

---

Groovy Goodness: Using the Template Engines

---

Groovy Goodness: BigDecimal

---

Groovy Goodness: Getting the Tail of a List

---

Groovy Goodness: Check If Method or Property is Available

---

Groovy Goodness: Default Imports

---

Groovy Goodness: Grouping Map Elements

---

Groovy Goodness: Using AntBuilder

---

Groovy Goodness: Reading XML

---

Groovy Goodness: Finding Data in Collections

---

Groovy Goodness: Using Groovy on the Command-Line

---

Groovy Goodness: Executing String or List

---

Groovy Goodness: Using ConfigSlurper with Configuration Scripts

---

Groovy Goodness: Reading URL Content

---

Groovy Goodness: Creating XML with MarkupBuilder

---

Groovy Goodness: GroovyCollections Utility Class

---

Groovy Goodness: Join Elements to a String

---

Groovy Goodness: Mixin Functionality to Classes

---

Groovy Goodness: Expando as Dynamic Bean

---

Groovy Goodness: Create a Singleton Class

---

Groovy Goodness: Newify to Create New Instances

---

Groovy Goodness: Use Categories to Add Functionality to Classes

---

Groovy Goodness: Building Object Graphs

---

Groovy Goodness: Observable Map and List

---

Groovy Goodness: Grab That Dependency

---

Groovy Goodness: Matchers for Regular Expressions

---

Groovy Goodness: Using Regular Expression Pattern Class

---

Groovy Goodness: Using GroovyMBeans for Easy JMX

---

Groovy Goodness: MultiMethods or Multiple Dispatch

---

Groovy Goodness: Parsing Commandline Arguments with CliBuilder

---

Groovy Goodness: Padding Strings

---

Groovy Goodness: Working with Arrays

---

Groovy Goodness: Lazy Initialization of Properties

---

Groovy Goodness: Making a Class Immutable

---

Groovy Goodness: Looping in Different Ways

---

Groovy Goodness: Check for Object Equality

---

Groovy Goodness: Named Parameters are Converted to Map

---

Groovy Goodness: Exception Handling

---

Groovy Goodness: Parameters with Default Values

---

Groovy Goodness: Optional return Keyword

---

Groovy Goodness: Using the GroupBy Method

---

Groovy Goodness: the as Keyword

---

Groovy Goodness: Using Lists and Maps As Constructors

---

Groovy Goodness: Add Some Curry for Taste

---

Groovy Goodness: Using the Inject Method

---

Groovy Goodness: the With Method

---

Groovy Goodness: Keep Your Values in Ranges

---

Groovy Goodness: the Spread Operator

---

Groovy Goodness: Date and Time Durations and the TimeCategory

---

Groovy Goodness: Multiple Assignments

---

Groovy Goodness: Working with Dates

---

Groovy Goodness: The Spread-Dot Operator

---

Groovy Goodness: Delegate to Simplify Code

---

Groovy Goodness: Working with Files

---

Groovy Goodness: the Spaceship Operator

[Groovy Goodness: the Grep Method](#)[Groovy Goodness: the Switch Statement](#)[Groovy Goodness: Tell the Groovy Truth](#)[Groovy Goodness: 'String', 'Strings', /Strings/ and GStrings](#)[Groovy Goodness: Turn Methods into Closures](#)[Groovy Goodness: Bound and Constrained Properties in GroovyBeans](#)[Groovy Goodness: GroovyBeans the Simpler JavaBeans](#)[Groovy Goodness: Safe Navigation to Avoid NullPointerException](#)[Groovy Goodness: the Elvis Operator ?:](#)[Groovy Goodness: Use a Map as Interface Implementation](#)[Groovy Goodness: String.multiply\(\)](#)[← Older Post](#)[Home](#)[Newer Post →](#)

#### Links

[Website](#)[Blog](#)[Books](#)[Github](#)[About](#)

#### Link to friends

[Droomkoekjes](#)[JDriven Blog](#)

#### Connect

[Twitter](#)[LinkedIn](#)[Contact](#)

#### Labels

[Apache Camel \(5\)](#)[Asciidoc \(69\)](#)[Asciidoctor \(73\)](#)[Awesome:Asciidoctor \(72\)](#)[Cocoon \(26\)](#)[Git \(5\)](#)[Google App Engine \(4\)](#)[Google Guava \(4\)](#)[Gradle \(152\)](#)[Gradle:Goodness \(136\)](#)[Grails \(203\)](#)[Grails:Goodness \(154\)](#)[Griffon:Goodness \(2\)](#)[Groovy \(578\)](#)[Groovy:Goodness \(376\)](#)[IntelliJ IDEA \(39\)](#)[Java \(53\)](#)[Maven \(31\)](#)[Maven2 \(33\)](#)[Micronaut \(16\)](#)

---

Micronaut:Mastery (16)

---

NetBeans (114)

---

OSX (4)

---

PlantUML (19)

---

PlantUML:Pleasantness (13)

---

Ratpack (63)

---

Ratpacked (63)

---

RESTful webservices (3)

---

Spock (43)

---

Spock:Spotlight (22)

---

Primavera (31)

---

Primavera 3 (1)

---

Primavera 4 (1)

---

Bota De Resorte (13)

---

Nube de primavera (1)

---

Primavera: Dulces (15)

---

Tomcat (8)

---

XSLT (5)

---

Yahoo Tuberías (5)

---

© 2008-2018 [Hubert A. Klein Ikkink](#) | [Tema oscuro](#) | [Tema ligero](#)

 Suscríbase en un lector