

Groovy



Java companion

<http://groovy.codehaus.org/>

Presentation source code: <https://github.com/zsokolowski/Groovy-TJUG>

About me:



Zbigniew Sokołowski

Java/Groovy developer



THE TECHNOLOGY
PERFORMANCE COMPANY

Contact info:

Email: zbigniew.sokolowski@data.pl



- dynamic language
- inspired by languages like Python, Ruby and Smalltalk
- modern programming features: functional programming (lambda), closures



- supports Domain-Specific Languages - Gradle
- increases developer productivity by reducing scaffolding code
 - simplifies testing



- seamlessly integrates with all existing Java classes and libraries
- compiles straight to Java bytecode so you can use it anywhere you can use Java

Integration with Java

Integration is not about producing valid bytecode

- Call to/from Java
- Inheritance mix
- Cross compile
- Class substitution
- Cross IDE support

Groovy stays in Java model!

Related well known projects



Gradle



project automation tool that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy based DSL

GParS

- concurrency
- parallelism
- actors like in Scala
- asynchronous programming



Related well known projects(2)



Grails  **GRAILS**

open source web application framework. It is intended to be a high-productivity framework by following the "coding by convention" paradigm.

Spock

Spock is a testing and specification framework for Java and Groovy applications.

Griffon



A Grails-like Rich Application Platform

Dynamic language



- Error at runtime
@TypeChecked
- Dynamic and Flow Typing – def keyword
 - IDE can support dev

Groovy features not in Java

- GroovyBeans
 - Working with XML - GPath
- Another kind of strings GStrings
 - Regular expressions

Groovy features not in Java

- Closures
- Native collections/ranges
 - GDK
 - AST

Regular expressions



Method Description and Java equivalent

- ~ Creates a pattern (creates a compiled Java Pattern object)
- =~ Creates a matcher (creates a Java Matcher object)
- ==~ Evaluates the string (effectively calls Java's match() on the Pattern)

"Zosia 1" =~ /(\w+) (\d+)/

Ranges



Range is an object, you can pass a range around and call its methods:

- `each`
- `contains == isCase`
- `grep` to filter
- `switch` – you can use range switch statements

Collections



Lists

```
def list = [1, 2, 3]
```

Maps

```
def map = [a:1, b:2, c:3]
```

GDK



- Processes
 - Thread
 - I/O
 - Templating
- Groovlets/GSP

Closures



What is a Closure?

A Groovy Closure is like a "code block" or a method pointer. It is a piece of code that is defined and then executed at a later point.

```
def list = ['a','b','c','d']  
def upperCase = list.collect { it.toUpperCase() }  
assert list instanceof ArrayList  
assert upperCase == ["A", "B", "C", "D"]
```


AST



Abstract Syntax Tree - The purpose of AST Transformations is to let developers hook into the compilation process to be able to modify the AST before it is turned into bytecode that will be run by the JVM.

@Singleton

@Delegate

@Immutable

@Mixin

@PackageScope

@CompileStatic

Others

Dynamic language



- Meta-Object-Protocol (MOP)
 - MetaClass:
 - [MetaClassImpl](#)
 - [ExpandoMetaClass](#), which can expand state and behavior,
 - [ProxyMetaClass](#), which can decorate a meta class with interception capabilities,
 - [Others](#), like testing purposes

```
println 'Hello' // Groovy
```

```
InvokerHelper.invokeMethod(this, "println",  
{"Hello"}); // Java
```

Performance



- provides the ability to statically type check and statically compile your code for robustness and performance
- offers full support for the JDK 7 “invoke dynamic” bytecode instruction (Groovy 2.1)

Dynamic language



For Groovy < 2.0
Groovy++

Groovy 2.0

`@TypeChecked` annotation – checks at compile time

`@CompileStatic` – static compilation

Extensions

Dynamic language: Groovy++

Groovy++ is a statically typed extension to the Groovy programming language. Groovy++ provides high performance and compile time type checking for compiled code.

- Beyond all the goodies of standard Groovy, this project adds a lot of functionality
 - compile time checking of code
 - as fast as Java performance of compiled code
 - easy mixing of statically and dynamically typed code
 - very powerful type inference
 - tail recursion
 - traits (interfaces with default implementation)

Embedding Groovy

Groovy has been designed to be very lightweight and easy to embed into any Java application system.

- Evaluate scripts or expressions using the Groovy shell
- Dynamically loading and running Groovy code inside Java – GroovyClassLoader
- The GroovyScriptEngine
- Embedding a Groovy Console in a Java Application