

Groovy and Grails seminar

Groovy: Java's dynamic friend

Dierk König • Graeme Rocher

dierk.koenig@canoo.com

graeme.rocher@gmail.com

Welcome!



Dierk König

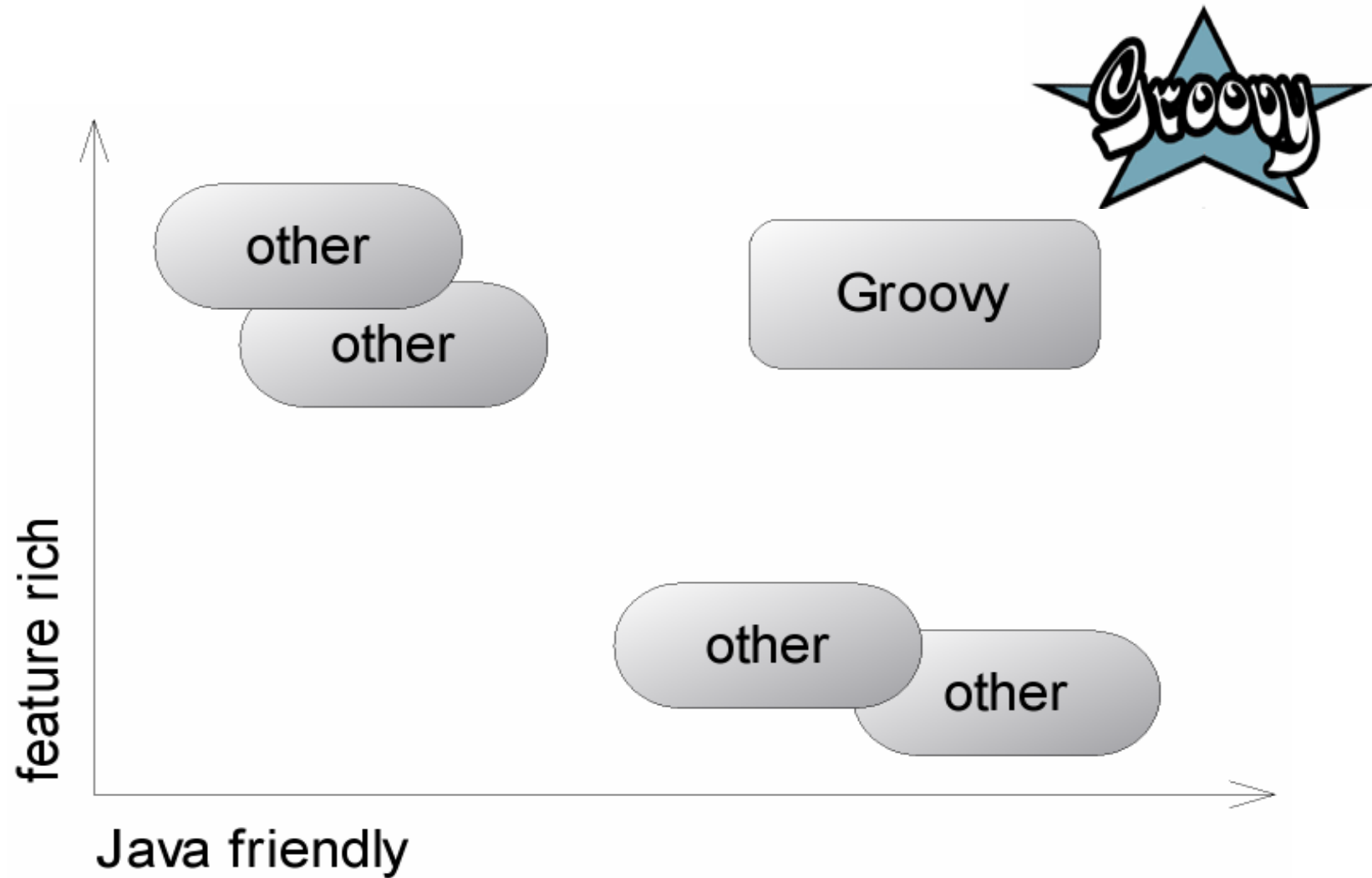
- Canoo Engineering AG, Basel (CH)
- Developer, tutor, and coach
- Groovy and Grails committer
- Author "Groovy in Action"



Graeme Rocher

- Technical Consultant, AdVal
- Developer, consultant and web expert
- Grails project lead
- Author "Definitive Guide to Grails"

The landscape of JVM languages

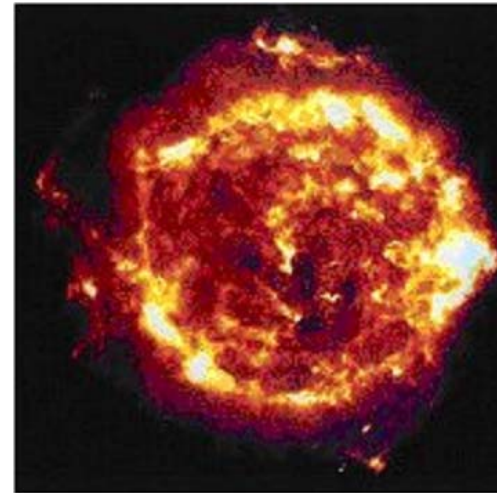


Scripting Patterns



Super glue

- Combine building blocks
- Groovy glues infrastructure and business logic together



Melted core

- Domain models evolve
- Groovy supports smart configs and DSLs

Dynamicity beyond scripting

- **Meta-Objekt-Protocol** changes structure and behaviour at runtime
- Work on JDK classes with richer functionality (**GDK**)
- No more „incomplete library smell“
- Intercepting method calls
 - Tracing and debugging
 - Mocks and stubs
 - Compares to „AOP“ or „Mixin“
- Behaviour becomes injectable, even „non-intrusive“!
- Reuse **pattern** implementations
- Avoid duplication

Groovy goodies



- Fully **object** oriented
no primitive types
- **Operators** can be implemented
- Multimethods
- Literal declaration for **lists** (arrays), **maps**, **ranges** and **regular expressions**
- Powerful operators []
- Interpolated Strings (GString)
- **Closures**
- **GPath**: efficient object navigation
- **GroovyBeans**
- Classifiers for grep and switch
- **Templates**, **builder**, swing, Ant, **markup**, XML, HTML, SQL, XML-RPC, Scriptom, GORM, **Grails**, Unit Tests, Mocks,...

Groovy starter

```
System.out.println("Hello World!");  
println 'Hello, World!'
```

```
def name = 'Graeme'  
println "$name, I get the car."  
String longer = "" "${name}, the  
is in the next row.""
```

```
assert 0.5 == 1/2
```

```
def printSize(obj) {  
    print obj?.size()  
}
```

- optional ; ()
GDK method
- Typ optional
- GString
- java.lang.String
- BigDecimal
.equals
- optional
duck typing
- safe
dereferencing

Regular expressions

```
if ("Hello World!" =~ /Hello/)
if ("Hello World!" == ~ /Hello\b.*/)
~ /Hello\b.*/
```

```
"1.23".replaceAll(/\d+/){ num ->
    num.toInteger() + 1
}
```

-> 2.24

- Find operator
- Match operator
- Pattern operator
- Replace with calculated values

Lists, maps, ranges

```
def empty = []
def full = [1, 2, 'SM']

assert full+full == full*2

assert full[0] == 1
assert full[0..1] == [1, 2]

voll[0..1] = [0, 1, 2, 3]
assert voll ==
    [0, 1, 2, 3, 'SM']
```

```
def empty = [:]
def full = [a: 1, b: 2]

assert full['a'] == 1
assert full.a == 1

full.a = 2
assert full == [a: 2, b: 2]

def inclusive = 'a'..'z'
inclusive.each {...}
def exclusive = 0..<10
for (i in exclusive) {...}
```

Closures

```
3.times { println 'Hi' }

[0, 1, 2].each { number ->
  println number
}

[0, 1, 2].each { println it }

def printit = { println it }
[0, 1, 2].each printit

new File('/x.txt').eachLine {
  println it
}
```

```
def houston(Closure doit) {
  (10..1).each { count ->
    doit(count)
  }
}

houston { println it }
```

GroovyBeans and GPath

```
class Dir {
    String name
    List  dirs
}

def root =
new Dir (name: '/', dirs: [
    new Dir (name: 'a'),
    new Dir (name: 'b')
])

assert root.dirs[0].name ==
    'a'
```

```
assert root.dirs.name ==
    ['a','b']
assert root.dirs.name*.size() ==
    [1, 1]
```

```
Dir.methods.name
    .grep(~/(g|s)et/)
-> [getName, setName, getDirs,
    setDirs, ...]
```

```
find, findAll, grep, every, any, ...
```

Templates and builder

```
import groovy.text.*
def text = '''
Dear salut${name.toUpperCase()},
nice to meet you in <%print city%>.
See you in <%= month %>.'''

def mail =
new SimpleTemplateEngine().
  createTemplate(text).
  make(
    salut:'Mrs.', name: 'Davis',
    city: 'London', month: 'July'
  )
```

```
import groovy.xml.*
def page = new MarkupBuilder()
page.html {
  head { title 'Hello' }
  body {
    ul {
      for (count in 1..10) {
        li "world $count"
      }
    }
  }
}
```

NodeBuilder, DomBuilder,
SwingBuilder, AntBuilder, ...

Control structures

```
if (1)
if (object)
if (list)

for (item in iterable) { }

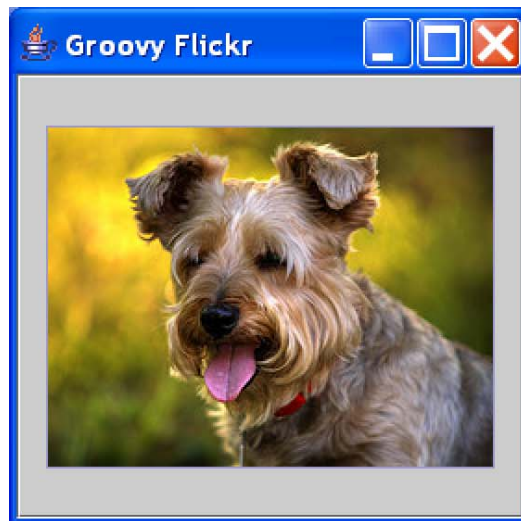
[a:1, b:2].each { key, value ->
    println "$key : $value"
}
```

throw, catch, finally
while, eachWithIndex, eachLine

```
switch (10) {
    case 0 : println 'is 0'; break
    case 0..9 : ...
    case [8,9,11] : ...
    case Float : ...
    case {it%3 == 0}: ...
    case ~/../ : ...
    default :
}
```

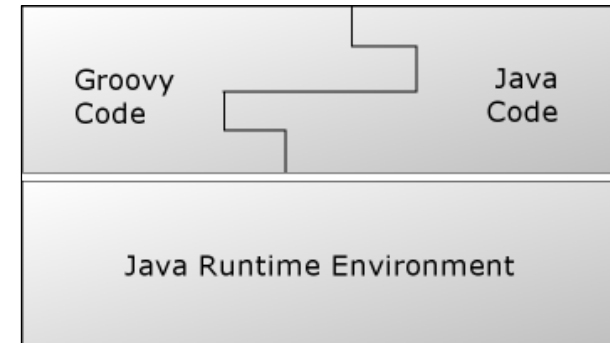
implement isCase()

Demo



Integration options

- Groovy objects are Java objects
- Compile with groovyc
- Evaluation at runtime
 - Eval.x (`users`,
`"x.grep{ it.salary > 100000 }.address.town"`);
 - GroovyShell,
GroovyScriptEngine, GroovyClassLoader
 - Bean Scripting Framework / [JSR-223](#) (Java 6)
 - Groovy [Spring](#) Beans



Groovy users

- Mutual of Omaha (USA, Fortune 500)
- Aspect Capital (London)
- KnowledgePool www.knowledgepool.com
- Pepsi, UUZone, www.pj-smoothies.co.uk
- Oracle clustering OC4J
- OSS: Spring, RIFE, NanoContainer, Drools, SnipSnap, XWiki, eXoPlatform, Turbine Fulcrum, Groovestry, Luxor XUL, Canoo WebTest, ...
- First job offers asking for Groovy

Further information

- <http://groovy.codehaus.org>
- <http://grails.org>
- **Groovy in Action**
Manning, 2006;
Foreword by James Gosling
König with Glover, Laforge, King, Skeet
- **The Definitive Guide to Grails**
Apress, 2007; Graeme Rocher
- More books in pipeline

