THE FUTURE IS GR8

Bob Brown
Transentia Pty. Ltd.
http://www.transentia.com.au
bob@transentia.com.au

Hong Kong JUG, 9 May 2013

Groovy

- a new(ish) programming language for the JVM
- an agile, dynamic programming language like Python, PERL and Ruby
- **completely interoperable** with conventional Java
- makes life **fun** again!

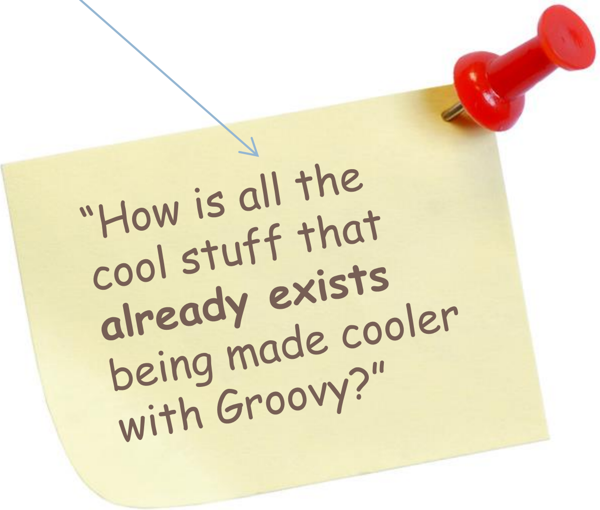
all the power and maturity of Java and the JVM, with *much greater* ease-of-use and productivity

The Question

"What cool new tools/apps, etc. are being created using Groovy?"

Wrong Question

Better Question



"How is all the cool stuff that **already exists** being made cooler with Groovy?"

The Answer

Groovy was never intended to replace Java but to make using the JVM better.

Groovy expands Java's capabilities and makes developers' lives easier.

If you can do it in Java, you can do it in Groovy…only much, much easier.

They 'complete' each other

The Answer…

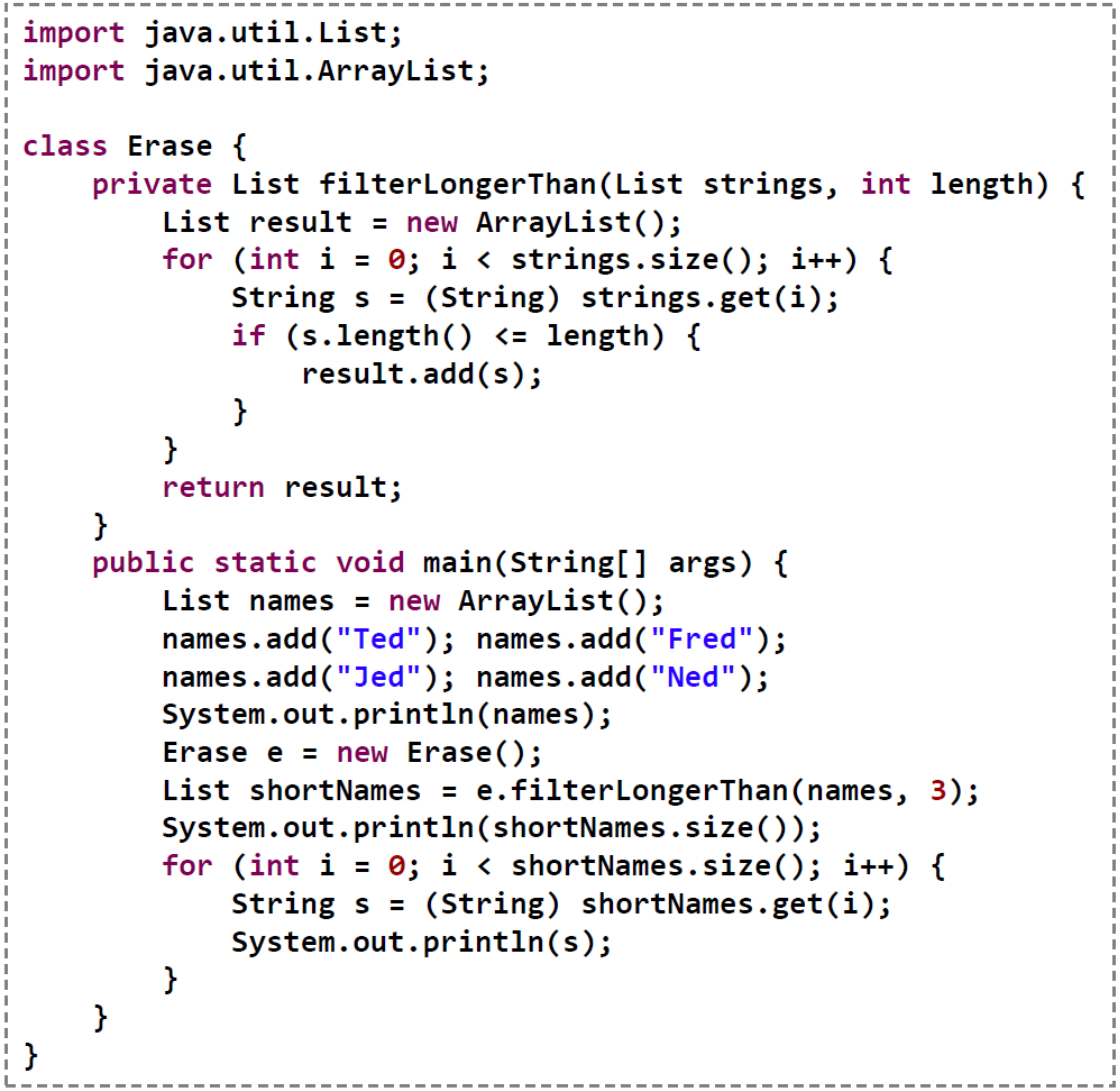
- all these companies, products and users are benefitting from the Gr8 technologies
  - Spring, Seam, JBoss, IntelliJ, Eclipse, JDeveloper/ADF, SoapUI, Selenium, Jenkins, Freemind, Confluence, OpenOffice…
  - eHarmony, European Patent Office, Wired.com, Vodafone, Netflix, Suncorp, Mincom, Atlassian, Thoughtworks, Canoo,…
    - Sky.com
      - ~1 billion monthly page impressions
  - Barack Obama!
  - me!

The Gr8 Technologies

- a complete, powerful ecosystem
  - Grails
  - Griffon
  - Gant
  - Gradle
  - GPars
  - Geb
  - Betamax
  - Spock
  - …many, many more

the old idea of "standing on the back of Giants" or "not reinventing wheels."

A Better Java...



```
import java.util.List;
import java.util.ArrayList;

class Erase {
    private List filterLongerThan(List strings, int length) {
        List result = new ArrayList();
        for (int i = 0; i < strings.size(); i++) {
            String s = (String) strings.get(i);
            if (s.length() <= length) {
                result.add(s);
            }
        }
        return result;
    }
    public static void main(String[] args) {
        List names = new ArrayList();
        names.add("Ted"); names.add("Fred");
        names.add("Jed"); names.add("Ned");
        System.out.println(names);
        Erase e = new Erase();
        List shortNames = e.filterLongerThan(names, 3);
        System.out.println(shortNames.size());
        for (int i = 0; i < shortNames.size(); i++) {
            String s = (String) shortNames.get(i);
            System.out.println(s);
        }
    }
}
```

This code
is valid
Java and
valid Groovy

*Based on an
example by
Jim Weirich
& Ted Leung*

...A Better Java...

```java
import java.util.List;
import java.util.ArrayList;

class Erase {
    private List filterLongerThan(List strings, int length) {
        List result = new ArrayList();
        for (int i = 0; i < strings.size(); i++) {
            String s = (String) strings.get(i);
            if (s.length() <= length) {
                result.add(s);
            }
        }
        return result;
    }
    public static void main(String[] args) {
        List names = new ArrayList();
        names.add("Ted"); names.add("Fred");
        names.add("Jed"); names.add("Ned");
        System.out.println(names);
        Erase e = new Erase();
        List shortNames = e.filterLongerThan(names, 3);
        System.out.println(shortNames.size());
        for (int i = 0; i < shortNames.size(); i++) {
            String s = (String) shortNames.get(i);
            System.out.println(s);
        }
    }
}
```

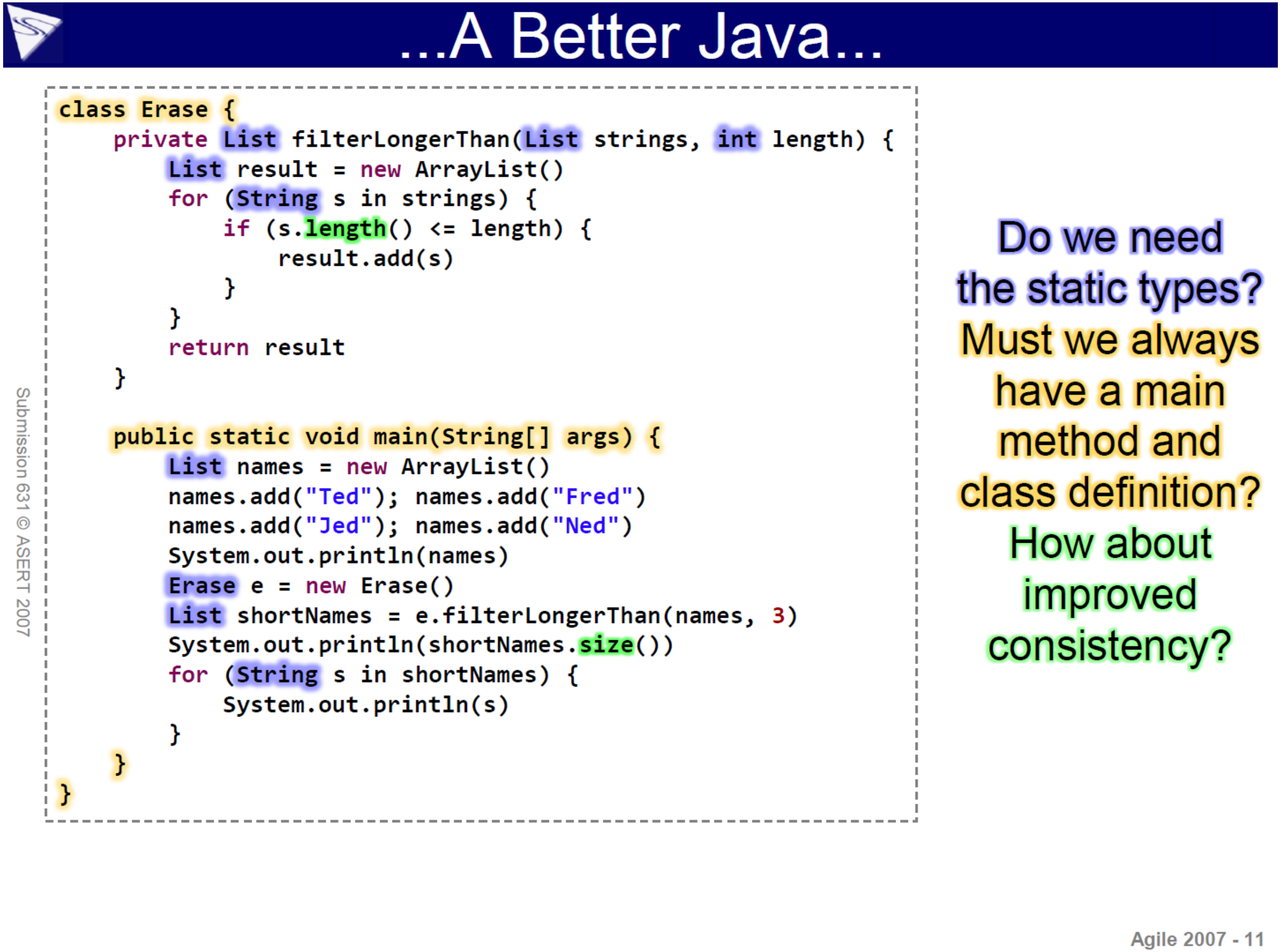
Do the semicolons add anything?

And shouldn't we us more modern list notation?

Why not import common libraries?

...A Better Java...

```
class Erase {
    private List filterLongerThan(List strings, int length) {
        List result = new ArrayList()
        for (String s in strings) {
            if (s.length() <= length) {
                result.add(s)
            }
        }
        return result
    }

    public static void main(String[] args) {
        List names = new ArrayList()
        names.add("Ted"); names.add("Fred")
        names.add("Jed"); names.add("Ned")
        System.out.println(names)
        Erase e = new Erase()
        List shortNames = e.filterLongerThan(names, 3)
        System.out.println(shortNames.size())
        for (String s in shortNames) {
            System.out.println(s)
        }
    }
}
```



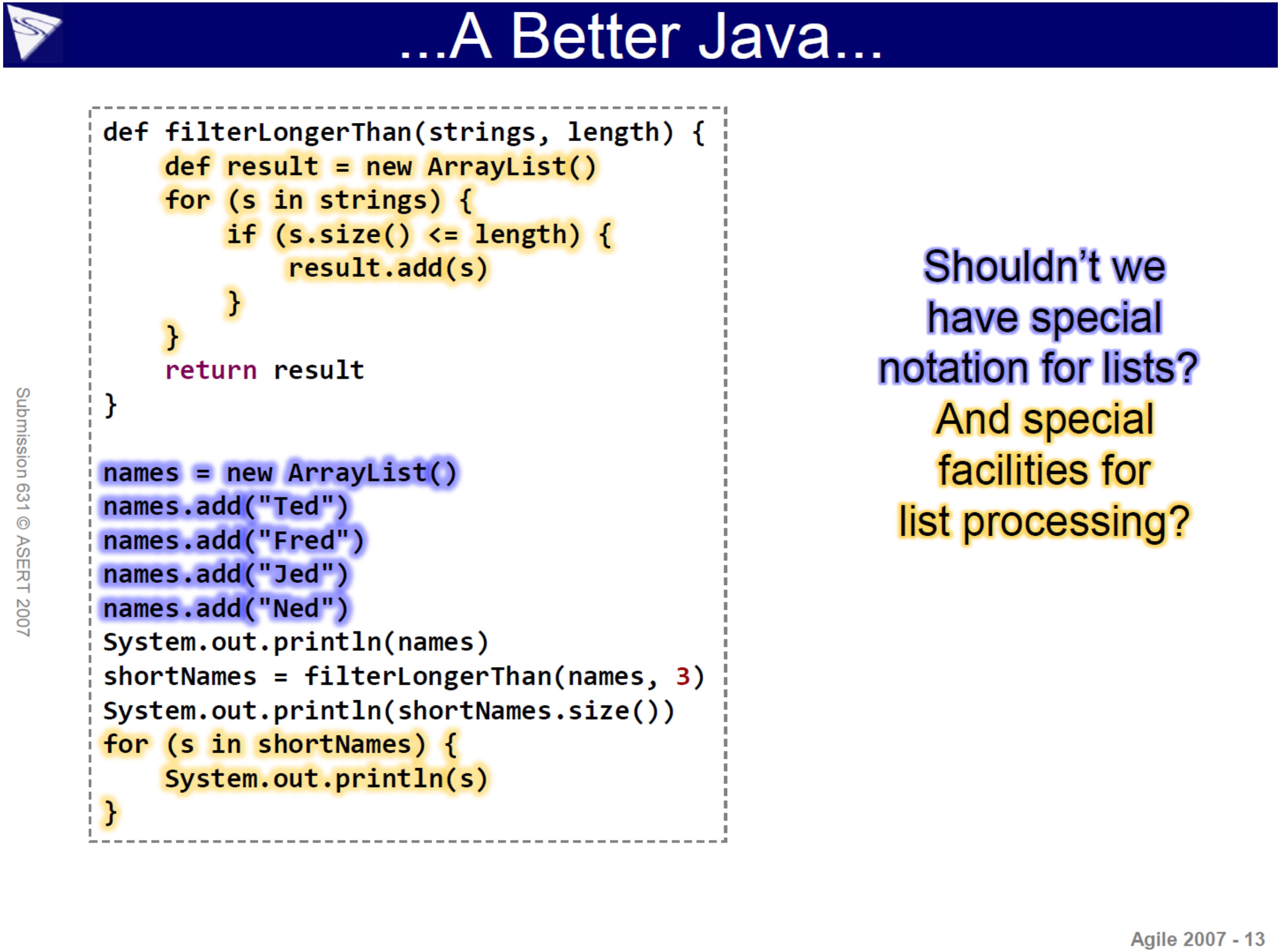
...A Better Java...

```
class Erase {
    private List filterLongerThan(List strings, int length) {
        List result = new ArrayList()
        for (String s in strings) {
            if (s.length() <= length) {
                result.add(s)
            }
        }
        return result
    }

    public static void main(String[] args) {
        List names = new ArrayList()
        names.add("Ted"); names.add("Fred")
        names.add("Jed"); names.add("Ned")
        System.out.println(names)
        Erase e = new Erase()
        List shortNames = e.filterLongerThan(names, 3)
        System.out.println(shortNames.size())
        for (String s in shortNames) {
            System.out.println(s)
        }
    }
}
```

Do we need the static types? Must we always have a main method and class definition? How about improved consistency?

...A Better Java...

```
def filterLongerThan(strings, length) {
    def result = new ArrayList()
    for (s in strings) {
        if (s.size() <= length) {
            result.add(s)
        }
    }
    return result
}

names = new ArrayList()
names.add("Ted")
names.add("Fred")
names.add("Jed")
names.add("Ned")
System.out.println(names)
shortNames = filterLongerThan(names, 3)
System.out.println(shortNames.size())
for (s in shortNames) {
    System.out.println(s)
}
```



...A Better Java...

```
def filterLongerThan(strings, length) {
    def result = new ArrayList()
    for (s in strings) {
        if (s.size() <= length) {
            result.add(s)
        }
    }
    return result
}

names = new ArrayList()
names.add("Ted")
names.add("Fred")
names.add("Jed")
names.add("Ned")
System.out.println(names)
shortNames = filterLongerThan(names, 3)
System.out.println(shortNames.size())
for (s in shortNames) {
    System.out.println(s)
}
```

Shouldn't we have special notation for lists? And special facilities for list processing?

Submission 631 © ASERT 2007

Agile 2007 - 13

...A Better Java...

```
def filterLongerThan(strings, length) {
    return strings.findAll{ it.size() <= length }
}

names = ["Ted", "Fred", "Jed", "Ned"]
System.out.println(names)
shortNames = filterLongerThan(names, 3)
System.out.println(shortNames.size())
shortNames.each{ System.out.println(s) }
```

...A Better Java...

```
def filterLongerThan(strings, length) {
    return strings.findAll{ it.size() <= length }
}

names = ["Ted", "Fred", "Jed", "Ned"]
System.out.println(names)
shortNames = filterLongerThan(names, 3)
System.out.println(shortNames.size())
shortNames.each{ System.out.println(s) }
```

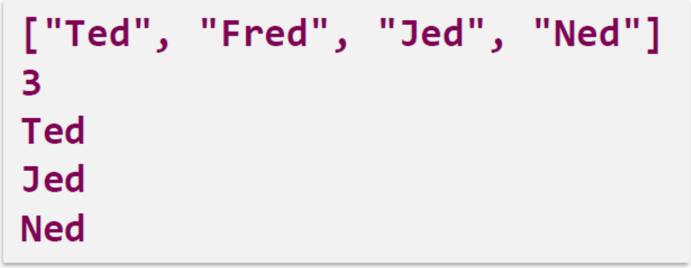
Is the method now needed?

Easier ways to use common methods?

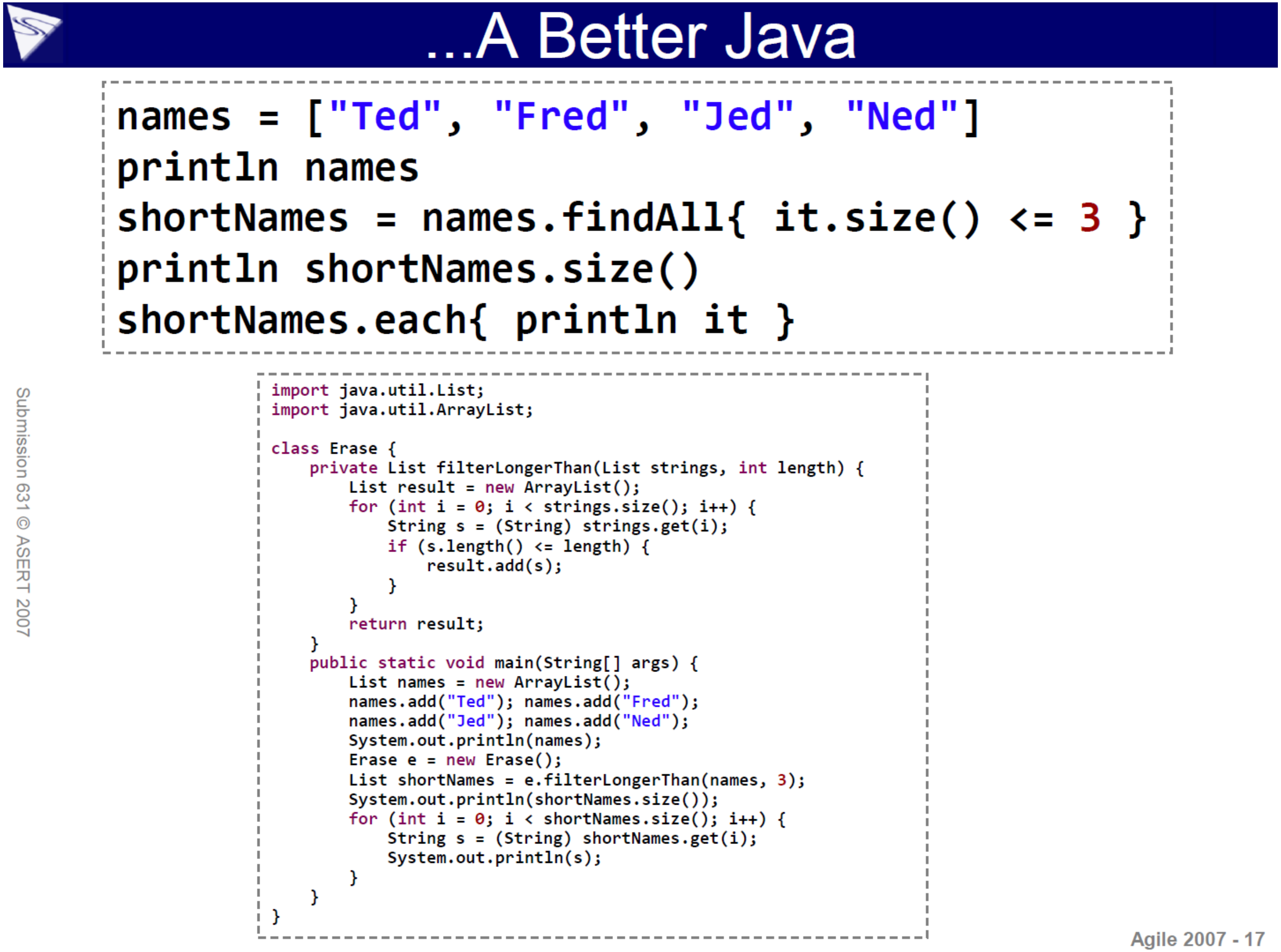
Are brackets required here?

...A Better Java...

```
names = ["Ted", "Fred", "Jed", "Ned"]
println names
shortNames = names.findAll{ it.size() <= 3 }
println shortNames.size()
shortNames.each{ println it }
```



```
["Ted", "Fred", "Jed", "Ned"]
3
Ted
Jed
Ned
```



...A Better Java

```
names = ["Ted", "Fred", "Jed", "Ned"]
println names
shortNames = names.findAll{ it.size() <= 3 }
println shortNames.size()
shortNames.each{ println it }
```

```
import java.util.List;
import java.util.ArrayList;

class Erase {
    private List filterLongerThan(List strings, int length) {
        List result = new ArrayList();
        for (int i = 0; i < strings.size(); i++) {
            String s = (String) strings.get(i);
            if (s.length() <= length) {
                result.add(s);
            }
        }
        return result;
    }
    public static void main(String[] args) {
        List names = new ArrayList();
        names.add("Ted"); names.add("Fred");
        names.add("Jed"); names.add("Ned");
        System.out.println(names);
        Erase e = new Erase();
        List shortNames = e.filterLongerThan(names, 3);
        System.out.println(shortNames.size());
        for (int i = 0; i < shortNames.size(); i++) {
            String s = (String) shortNames.get(i);
            System.out.println(s);
        }
    }
}
```

Many thanks to *maestro* Paul! http://www.asert.com.au

Aims

- put the FUN back into work!
- simplify developers lives
  - convention-over-configuration
  - become more ‘agile’
- make better tools
  - scripting
  - builders and slurpers
- make building tools easier
  - Domain-Specific Languages

Scripting

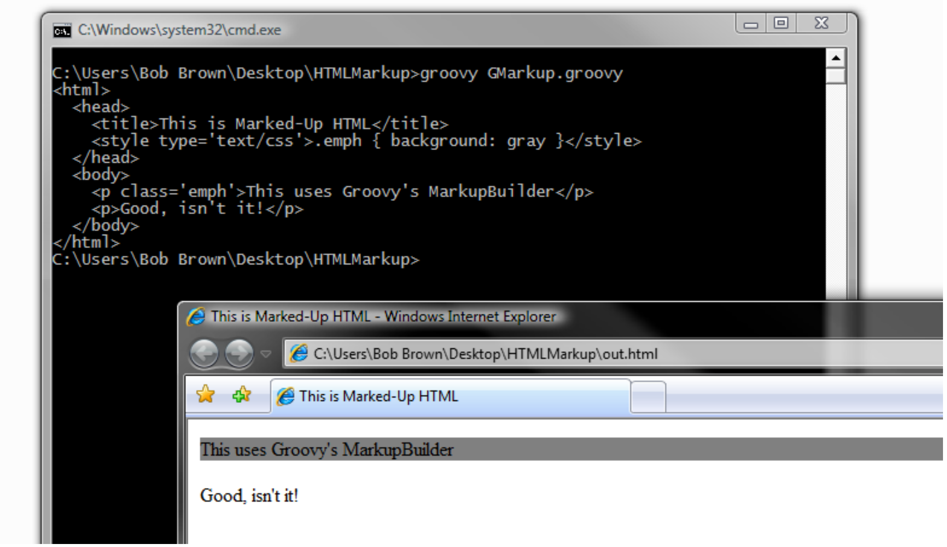
- no more need for shell scripts, PERL, etc.

```
final DIR = /C:\Users\Bob Brown\Desktop/

datPagesScanner = new AntBuilder().fileScanner {
    fileset(dir: DIR, includes: '*.dat')
}

new File("${DIR}/copy.txt").withWriter { file ->
  datPagesScanner.each { datFile ->
    datFile.eachLine { line ->
    if (line =~ /^[AEOIUaeiou].*/)
      file.writeLine(line)
    }
  }
}
```

Builders

- simplify creation of HTML, XML, JSON, Swing UI, …

```
import groovy.xml.MarkupBuilder

def builder = new MarkupBuilder ();
 builder.html {
  head {
   title "This is Marked-Up HTML"
   style type:'text/css', ".emph { background: gray }"
  }
  body {
   p 'class':'emph', "This uses Groovy's MarkupBuilder"
   p(/Good, isn't it!/)
  }
 }
```


C:\Windows\system32\cmd.exe
This is Marked-Up HTML - Windows Internet Explorer
C:\Users\Bob Brown\Desktop\HTMLMarkup\out.html
This is Marked-Up HTML

```
C:\Users\Bob Brown\Desktop\HTMLMarkup>groovy GMarkup.groovy
<html>
  <head>
    <title>This is Marked-Up HTML</title>
    <style type='text/css'>.emph { background: gray }</style>
  </head>
  <body>
    <p class='emph'>This uses Groovy's MarkupBuilder</p>
    <p>Good, isn't it!</p>
  </body>
</html>
C:\Users\Bob Brown\Desktop\HTMLMarkup>
```

This uses Groovy's MarkupBuilder

Good, isn't it!

Slurpers

- □ just compare…

```
<?xml version="1.0" encoding="UTF-8"?>
<items>
 <an-item the-id="0">This is item 0</an-item>
 […elided…]
</items>
```

```
import java.io.*;
import javax.xml.parsers.*;
import org.w3c.dom.*;

public class XMLReader {
 public static void main(String argv[]) throws Exception {
  File file = new File("items.xml");
  DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
  DocumentBuilder db = dbf.newDocumentBuilder();
  Document doc = db.parse(file);
  doc.getDocumentElement().normalize();
  NodeList nodeLst = doc.getElementsByTagName("an-item");
  for (int s = 0; s < nodeLst.getLength(); s++) {
   Element anItem = (Element) nodeLst.item(s);
   System.out.println(anItem.getAttribute("the-id") + ": " +
        anItem.getChildNodes().item(0).getNodeValue());
  }
 }
}
```

"Nothing Makes You Want Groovy More Than XML…"

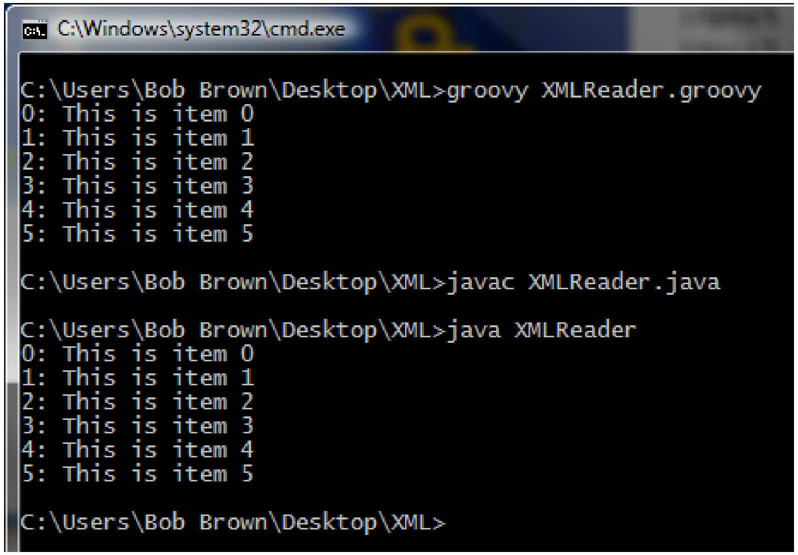
—http://kousenit.wordpress.com/2008/03/12/nothing-makes-you-want-groovy-more-than-xml/

Slurpers…

- *easily* consume structured data

```
items = new XmlSlurper().parse(new File('items.xml'))

items?.'an-item'.each {
  println "${it.'@the-id'.text()}: ${it.text()}"
}
```

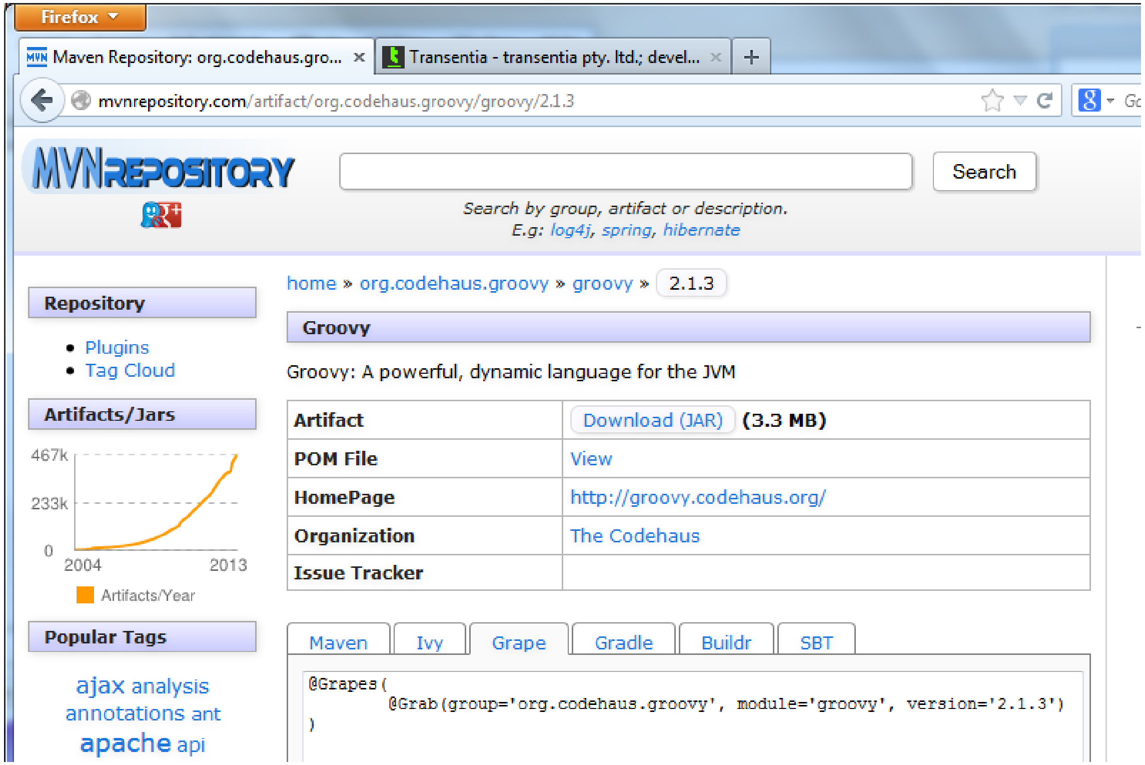
- all sorts of slupers
  - Config, XML, JSON, CSV



```
C:\Users\Bob Brown\Desktop\XML>groovy XMLReader.groovy
0: This is item 0
1: This is item 1
2: This is item 2
3: This is item 3
4: This is item 4
5: This is item 5

C:\Users\Bob Brown\Desktop\XML>javac XMLReader.java

C:\Users\Bob Brown\Desktop\XML>java XMLReader
0: This is item 0
1: This is item 1
2: This is item 2
3: This is item 3
4: This is item 4
5: This is item 5

C:\Users\Bob Brown\Desktop\XML>
```

Grapes

- Groovy Advanced Packaging Engine
  - lets you grab a script’s needed resources


Firefox
Maven Repository: org.codehaus.gro...
Transentia - transentia pty. ltd.; devel...
mvnrepository.com/artifact/org.codehaus.groovy/groovy/2.1.3
MVNREPOSITORY
Search
Search by group, artifact or description.
E.g: log4j, spring, hibernate
Repository
Plugins
Tag Cloud
Artifacts/Jars
467k
233k
0
2004
2013
Artifacts/Year
Popular Tags
ajax analysis
annotations ant
apache api
Maven
Ivy
Grape
Gradle
Buildr
SBT

home » org.codehaus.groovy » groovy » 2.1.3

Groovy

Groovy: A powerful, dynamic language for the JVM

Artifact	Download (JAR) **(3.3 MB)**
POM File	View
HomePage	http://groovy.codehaus.org/
Organization	The Codehaus
Issue Tracker	

```
@Grapes(
        @Grab(group='org.codehaus.groovy', module='groovy', version='2.1.3')
)
```

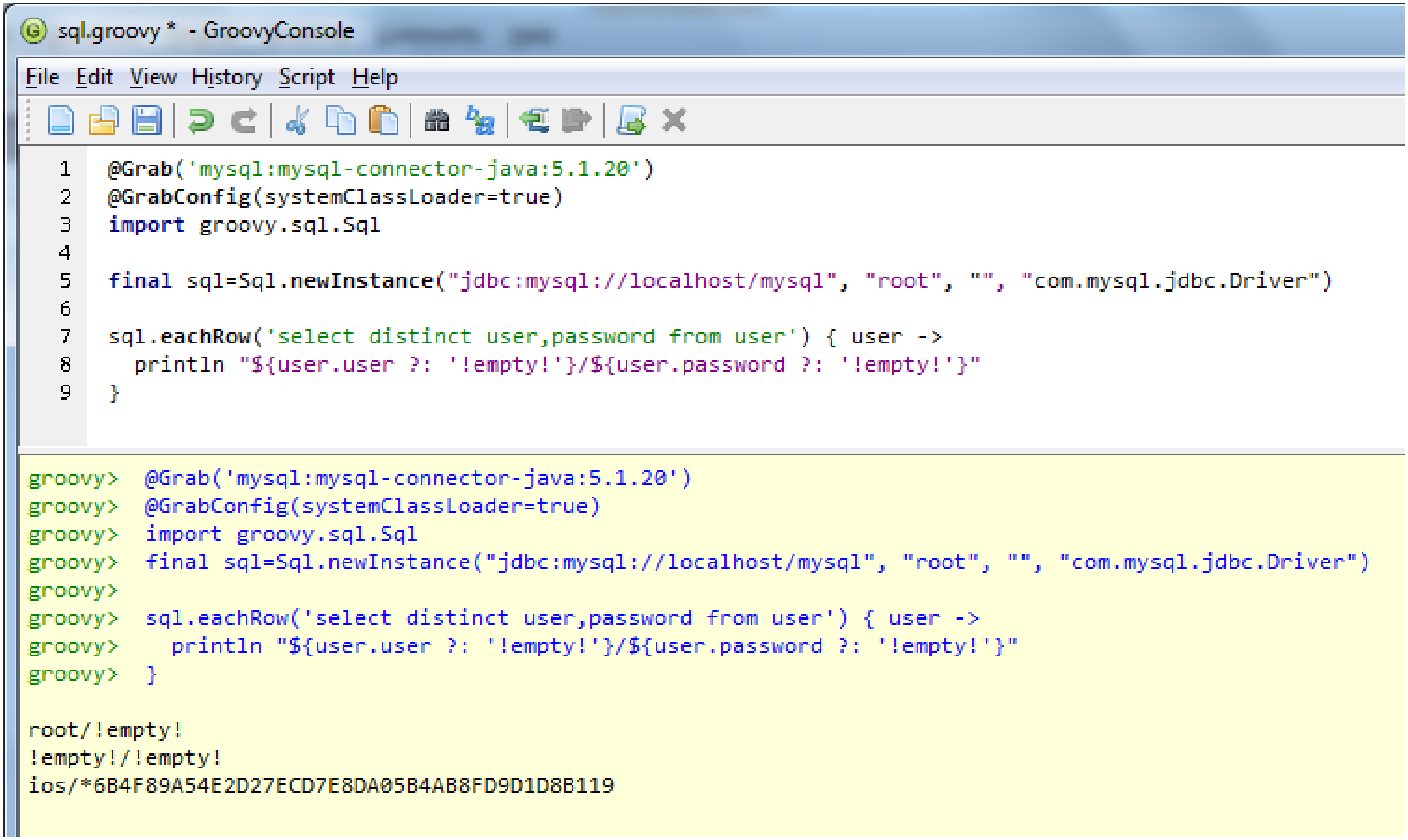
Grapes…

```
@Grapes(
  @Grab('com.googlecode.gbench:gbench:0.4.1-groovy-2.1')
)

def r = benchmark(verbose: true) {
    'Each' {
        def i = 0
        (1..100000000).each { i ++ }
    }
    'For' {
        def i = 0
        for (x in 1..100000000) { i ++ }
    }
}
r.prettyPrint()
```

GBench for easy micro-benchmarking

```
Environment
===========
* Groovy: 2.1.2
* JVM: Java HotSpot(TM) 64-Bit Server VM (23.7-b01, Oracle Corporation)
    * JRE: 1.7.0_17
    * Total Memory: 127.875 MB
    * Maximum Memory: 127.875 MB
* OS: Windows 8 (6.2, amd64)

Options
=======
* Warm Up: Auto (- 60 sec)
* CPU Time Measurement: On

Warming up "Each"...
Measuring "Each"...
Warming up "For"...
Measuring "For"...
            user  system         cpu        real

Each  7859375000       0  7859375000  8111961507
For   2250000000       0  2250000000  2304534377
```

Better JDBC With GroovySQL


sql.groovy * - GroovyConsole
File Edit View History Script Help

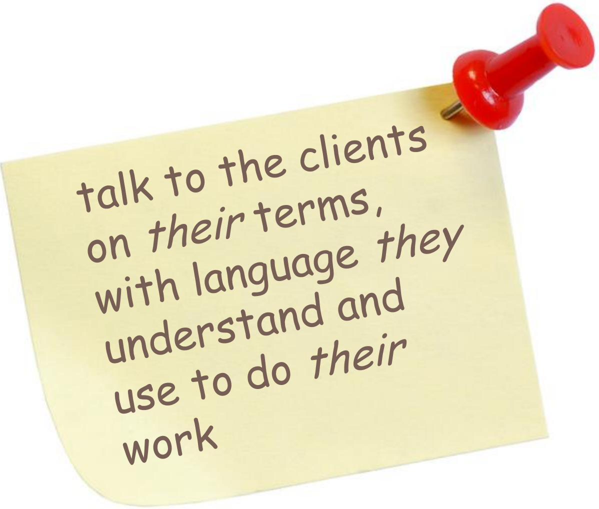
```
@Grab('mysql:mysql-connector-java:5.1.20')
@GrabConfig(systemClassLoader=true)
import groovy.sql.Sql

final sql=Sql.newInstance("jdbc:mysql://localhost/mysql", "root", "", "com.mysql.jdbc.Driver")

sql.eachRow('select distinct user,password from user') { user ->
  println "${user.user ?: '!empty!'}/${user.password ?: '!empty!'}"
}
```

```
groovy>  @Grab('mysql:mysql-connector-java:5.1.20')
groovy>  @GrabConfig(systemClassLoader=true)
groovy>  import groovy.sql.Sql
groovy>  final sql=Sql.newInstance("jdbc:mysql://localhost/mysql", "root", "", "com.mysql.jdbc.Driver")
groovy>
groovy>  sql.eachRow('select distinct user,password from user') { user ->
groovy>    println "${user.user ?: '!empty!'}/${user.password ?: '!empty!'}"
groovy>  }

root/!empty!
!empty!/!empty!
ios/*6B4F89A54E2D27ECD7E8DA05B4AB8FD9D1D8B119
```

Domain Specific Languages

- ‘little languages’ for well-defined purposes

```
presentation('Gr8 Technologies') {
  used 'laptop-imp' duration 1.2.hours
  printed 5.pages on 'hp-printer'
  presented 1.hour date '9/5/2013' at 'HKJUG'
}
```



talk to the clients on *their* terms, with language *they* understand and use to do *their* work

if it works out well *they* will do all the hard work!

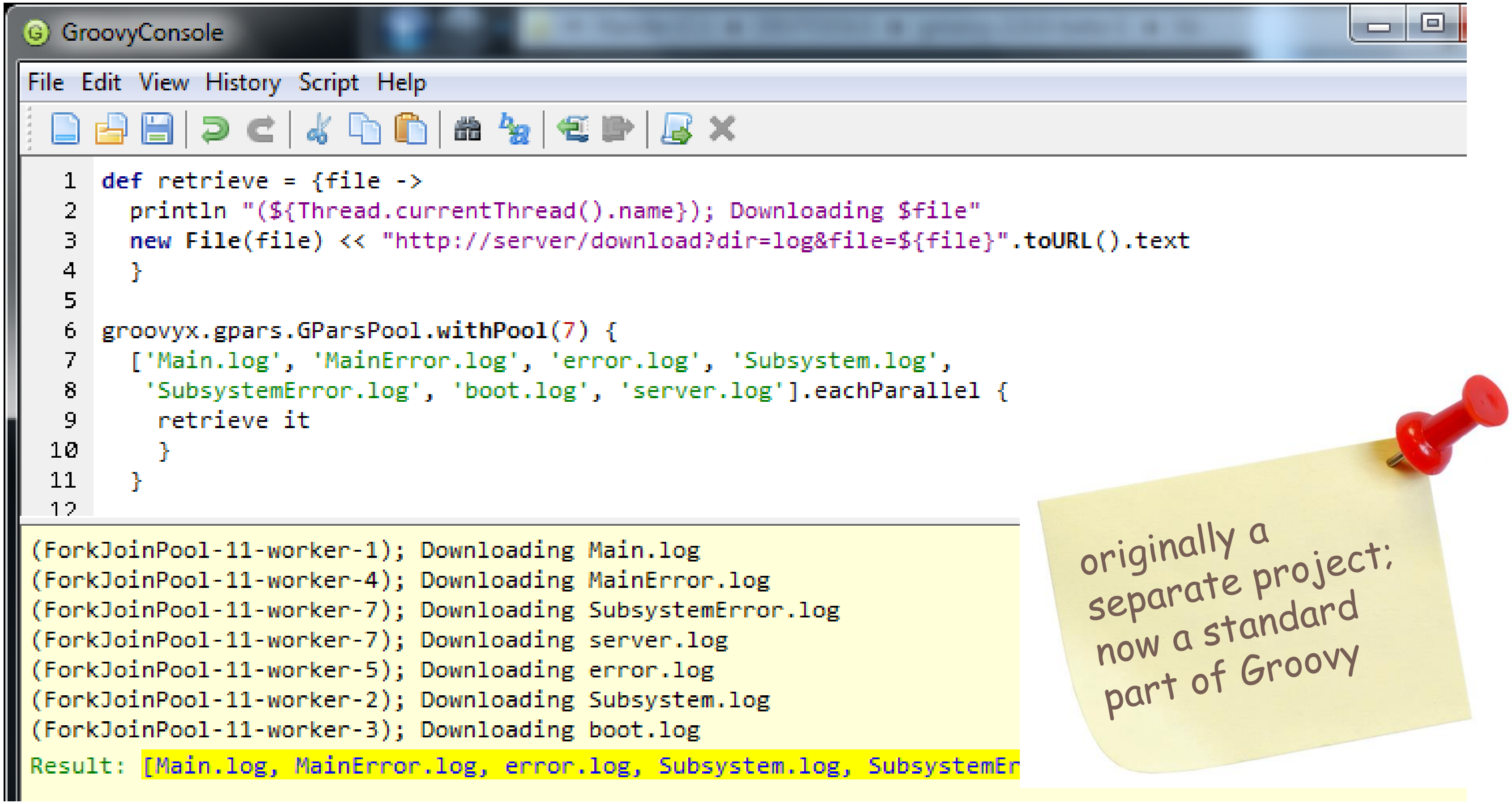
Domain Specific Languages…

- very simple to define!
  - command chain expressions

```
[…elided…]

def presented(hours) {
   ['date': { date ->
     ['at': {  where ->
       // may want something more interesting…maybe insert into a db/send an email…
       println "presented $hours hour(s) on $date at $where"
     }]
   }]
}

def used(equipment) {
  ['duration': { dur ->
    println "used $equipment for $dur hour(s)"
  }]
}

def printed(pages) {
  ['on': { equipment ->
    println "$pages page(s) were printed on '$equipment'"
  }]
}
```

GPars

- parallel programing made *easy(er)*



```groovy
def retrieve = {file ->
  println "(${Thread.currentThread().name}); Downloading $file"
  new File(file) << "http://server/download?dir=log&file=${file}".toURL().text
  }

groovyx.gpars.GParsPool.withPool(7) {
  ['Main.log', 'MainError.log', 'error.log', 'Subsystem.log',
   'SubsystemError.log', 'boot.log', 'server.log'].eachParallel {
    retrieve it
    }
  }
```

```
(ForkJoinPool-11-worker-1); Downloading Main.log
(ForkJoinPool-11-worker-4); Downloading MainError.log
(ForkJoinPool-11-worker-7); Downloading SubsystemError.log
(ForkJoinPool-11-worker-7); Downloading server.log
(ForkJoinPool-11-worker-5); Downloading error.log
(ForkJoinPool-11-worker-2); Downloading Subsystem.log
(ForkJoinPool-11-worker-3); Downloading boot.log
Result: [Main.log, MainError.log, error.log, Subsystem.log, SubsystemEr
```

originally a
separate project;
now a standard
part of Groovy

Interoperability

- use any old Java library
- JFugue
  - pure Java midi/music framework
  - no problem to use with Groovy

```
import org.jfugue.*

final darthVaderTheme = new Pattern(
"""
T160 I[Cello]
G3h G3h G3h Eb3q Bb3i G3qi Eb3q Bb3i G3hi
Rq
D4h D4h D4h Eb4q B3i G3qi Eb3q B3q G3h
""")

new Player().with { player ->
  player.play(darthVaderTheme)
  player.saveMidi(darthVaderTheme,
                  new File(/C:\DEVELOPMENT\Gradle\StarWars\darth.mid/))
}
```

WSLite

- no-frills SOAP and REST webservice clients
    - Simple & powerful

```
@Grab(group='com.github.groovy-wslite', module='groovy-wslite', version='0.1')
import wslite.soap.*

def soapClient = new SOAPClient("http://www.webservicex.net/WeatherForecast.asmx")
def response = soapClient.send {
    version SOAPVersion.V1_2
    body {
        GetWeatherByZipCode(xmlns:"http://www.webservicex.net") {
            ZipCode("93657")
        }
    }
}

assert "SANGER" ==
  response.GetWeatherByZipCodeResponse.GetWeatherByZipCodeResult.PlaceName.text()
assert 200 == response.http.statusCode
assert "OK" == response.http.statusMessage
assert "ASP.NET" == response.http.headers["X-Powered-By"]
```

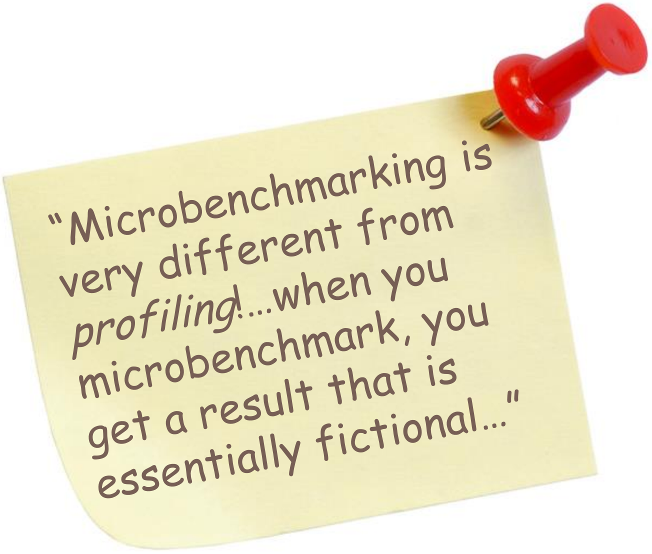
Performance

- Big Topic!
- dynamic language implies *some* overhead
  - but often within a few % of ‘pure’ Java
- developer vs program
- Java 7’s new invokeDynamic opcode
- AST Transforms
- @Typechecked, @CompileStatic
- lots of ongoing work

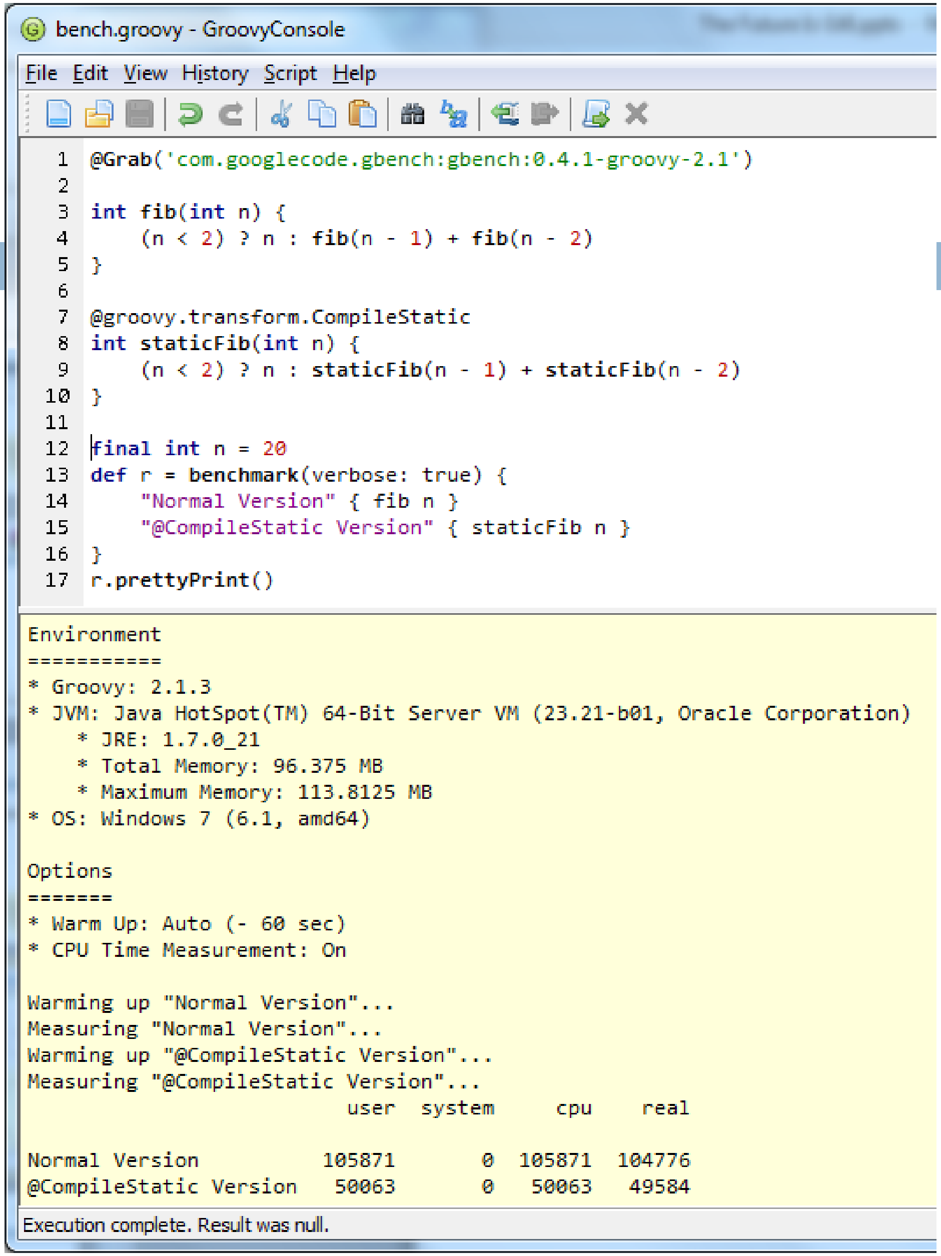
“ …use Groovy to improve **your** performance not the computer’s…”

Performance…

- @CompileStatic
  - > 2 * speedup



"Microbenchmarking is very different from *profiling*!…when you microbenchmark, you get a result that is essentially fictional…"



bench.groovy - GroovyConsole
File Edit View History Script Help

```groovy
@Grab('com.googlecode.gbench:gbench:0.4.1-groovy-2.1')

int fib(int n) {
    (n < 2) ? n : fib(n - 1) + fib(n - 2)
}

@groovy.transform.CompileStatic
int staticFib(int n) {
    (n < 2) ? n : staticFib(n - 1) + staticFib(n - 2)
}

final int n = 20
def r = benchmark(verbose: true) {
    "Normal Version" { fib n }
    "@CompileStatic Version" { staticFib n }
}
r.prettyPrint()
```

```
Environment
===========
* Groovy: 2.1.3
* JVM: Java HotSpot(TM) 64-Bit Server VM (23.21-b01, Oracle Corporation)
    * JRE: 1.7.0_21
    * Total Memory: 96.375 MB
    * Maximum Memory: 113.8125 MB
* OS: Windows 7 (6.1, amd64)

Options
=======
* Warm Up: Auto (- 60 sec)
* CPU Time Measurement: On

Warming up "Normal Version"...
Measuring "Normal Version"...
Warming up "@CompileStatic Version"...
Measuring "@CompileStatic Version"...
                          user  system     cpu    real

Normal Version          105871       0  105871  104776
@CompileStatic Version   50063       0   50063   49584
```

Execution complete. Result was null.

http://www.sergeydolgopolov.me/2012/07/groovy-20-has-been-released-testing-new.html
http://code.google.com/p/caliper/wiki/JavaMicrobenchmarks

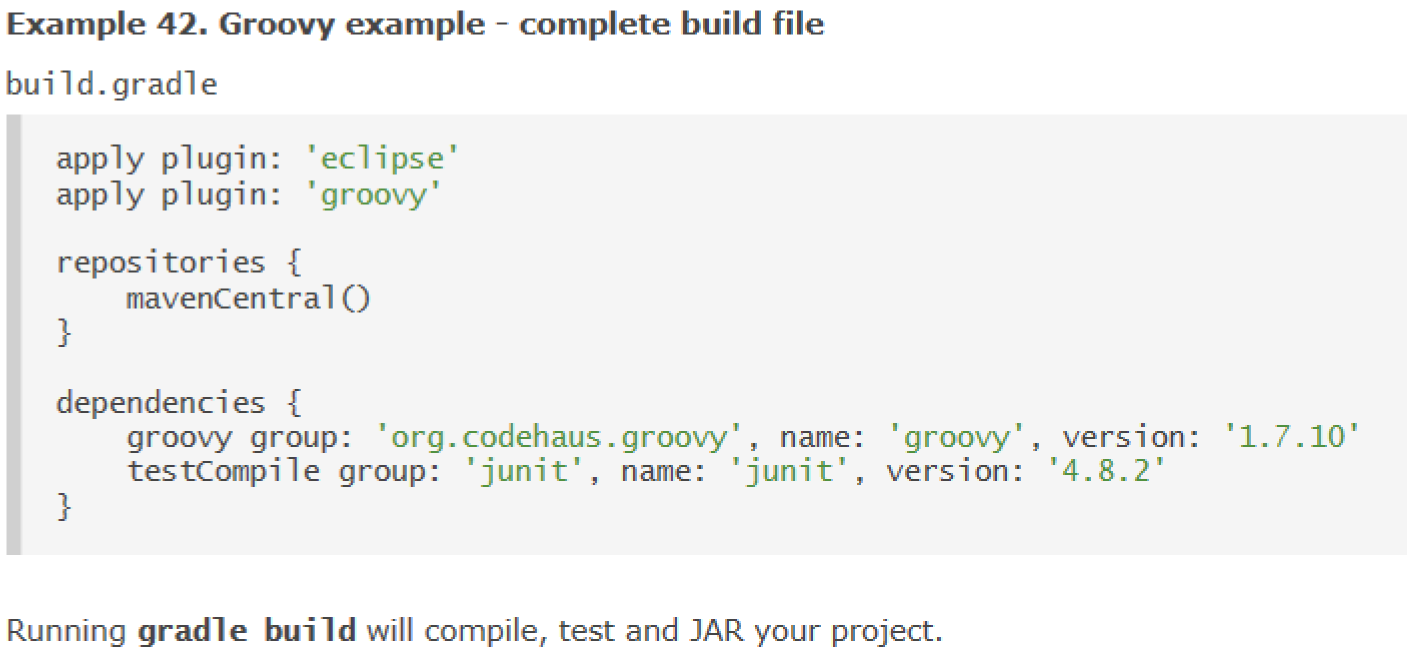
Gant

- scripting Ant tasks using Groovy
  - no XML!

```
includeTargets << gant.targets.Clean
cleanPattern << ['**/*.class', '**/*~', '**/*.bak', '**/*.OLD']
cleanDirectory << 'build'

taskdef (name: 'groovyc', classname: 'org.codehaus.groovy.ant.Groovyc')

ant.path(id: 'runtimeClasspath') {
  pathelement(location: 'build')
  pathelement(location: 'C:/DEVTOOLS/gant-1.8.1/lib/groovy-all-1.6.5.jar')
}

target(name: 'default') {
  ant.mkdir(dir: 'build')
  groovyc (srcdir: 'src', destdir: 'build', verbose: false)
  java(classname: 'HelloWorld', fork:true, dir: 'build',
       classpathref: 'runtimeClasspath') {
    arg(line: 'FRED')
  }
}
```

Gradle

- more convention, less configuration
  - no more “{classpath, xml, maven} hell”



Example 42. Groovy example - complete build file

build.gradle

```
apply plugin: 'eclipse'
apply plugin: 'groovy'

repositories {
    mavenCentral()
}

dependencies {
    groovy group: 'org.codehaus.groovy', name: 'groovy', version: '1.7.10'
    testCompile group: 'junit', name: 'junit', version: '4.8.2'
}
```

Running **`gradle build`** will compile, test and JAR your project.

http://prezi.com/ypobbs_ikvtj/gradle-a-better-way-to-build/

Vert.X

- “Effortless asynchronous application development for the modern web and enterprise”
  - in the style of Node.js…
  - …but polyglot
    - Groovy/Java, JavaScript, CoffeeScript, Ruby, Python
  - …and on the JVM

```
vertx.createHttpServer().requestHandler { req ->
    def file = req.uri == "/" ? "index.html" : req.uri
    req.response.sendFile "webroot/$file"
}.listen(8080)
```

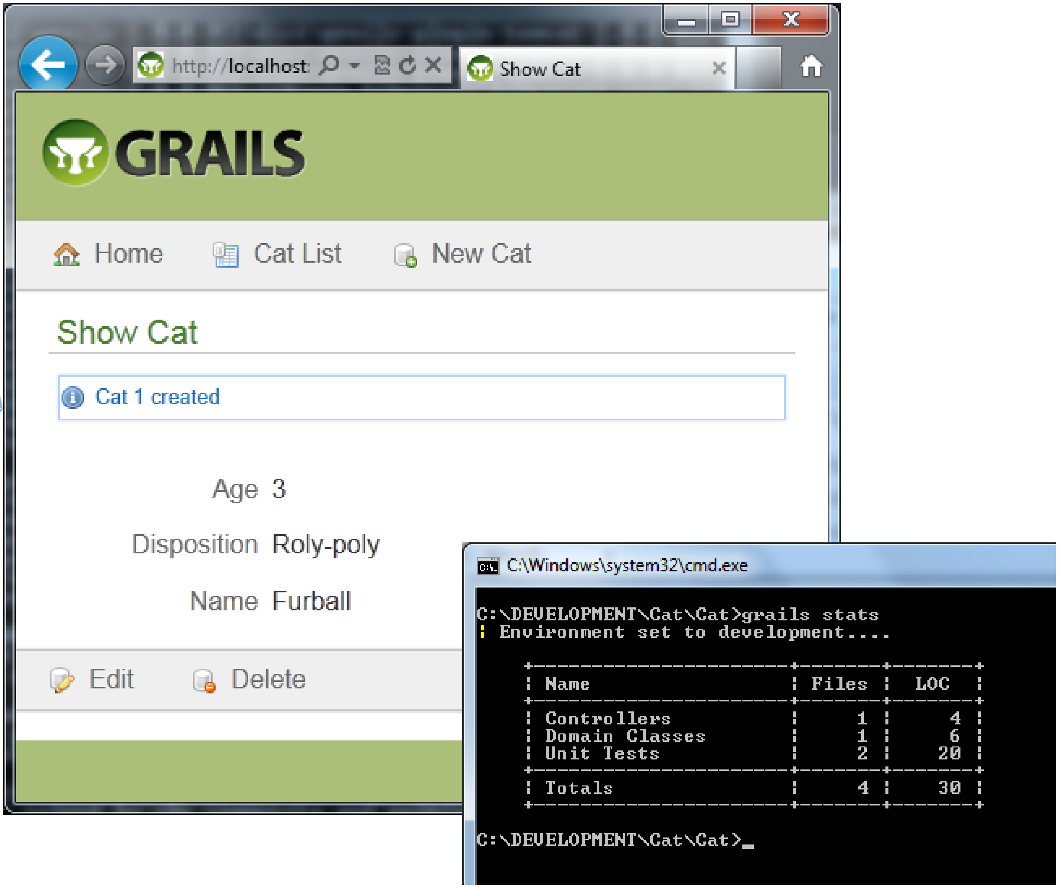
Grails

"Grails is a dynamic web application framework built on Java and Groovy, leveraging best of breed APIs including Spring, Hibernate and SiteMesh. Grails brings to Java and Groovy developers the joys of convention-based rapid development while allowing them to leverage their existing knowledge and capitalize on the proven and performant APIs Java developers have been using for years."

Grails...

- a full CRUD HTML5 webapp
  - *minimal* effort

```
// persistent domain class
class Cat {
  String name
  short age
  String disposition
}


// controller class
class CatController {
    static scaffold = true
}
```



Show Cat

Cat 1 created

Age 3

Disposition Roly-poly

Name Furball

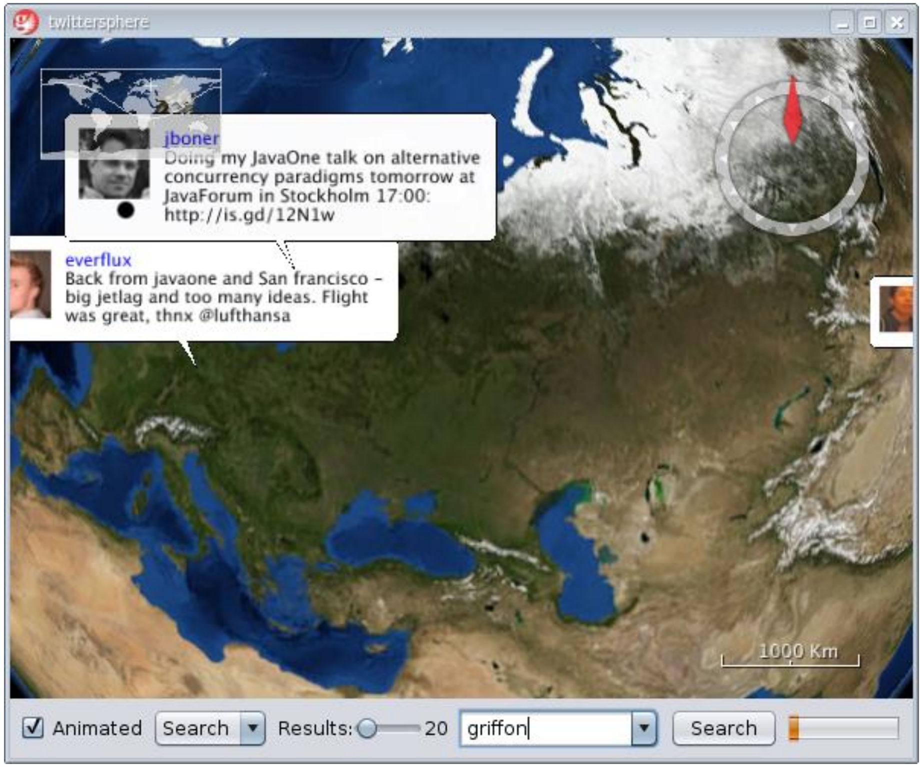
```
C:\DEVELOPMENT\Cat\Cat>grails stats
| Environment set to development....

    +----------------------+-------+-------+
    | Name                 | Files |  LOC  |
    +----------------------+-------+-------+
    | Controllers          |     1 |     4 |
    | Domain Classes       |     1 |     6 |
    | Unit Tests           |     2 |    20 |
    +----------------------+-------+-------+
    | Totals               |     4 |    30 |
    +----------------------+-------+-------+

C:\DEVELOPMENT\Cat\Cat>_
```

Griffon

- grails-like rich Swing client framework
  - standardised build system ‘inspired’ by Grails
    - ‘…by “inspired” I mean “taking large chunks of Grails code to bootstrap the codebase…”’
  - a structure that supports/rewards MVC
    - and enables **easy thread-handling**
      - one of the biggest hurdles for Swing developers
  - Groovy goodness: builders, @Bindable annotation, metaclass method injection, scripts, etc.
  - declarative layout of GUI code in the view
  - plugins
  - automatic packaging and signing for WebStart, Applet, and traditional application deployment
    - from the **SAME** source

Griffon…

- twittersphere
  - created as a technology demonstration for JavaOne 2009
  - won the Script Bowl
    - against Jython, Clojure, Scala and JRuby
  - mashup with NASA World Wind
    - locates twitterers on an animated world map
    - in real-time!
    - only 681 LOC!



jboner
Doing my JavaOne talk on alternative concurrency paradigms tomorrow at JavaForum in Stockholm 17:00: http://is.gd/12N1w

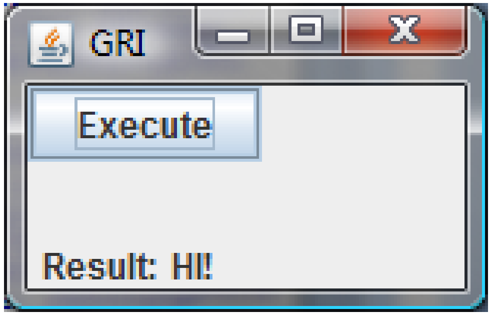
everflux
Back from javaone and San francisco - big jetlag and too many ideas. Flight was great, thnx @lufthansa

Griffon…

```
application(title:'GRI',
            pack:true,
            locationByPlatform:true) {
  borderLayout()
  hbox(constraints:NORTH) {
    button("Execute", actionPerformed:controller.&executeScript)
  }
  hbox(constraints:SOUTH) {
    hstrut(5)
    label("Result:")
    hstrut(5)
    label(text:bind {model.greeting})
  }
}
```

```
import groovy.beans.Bindable

class GRIModel {
  @Bindable def greeting = ""
}
```

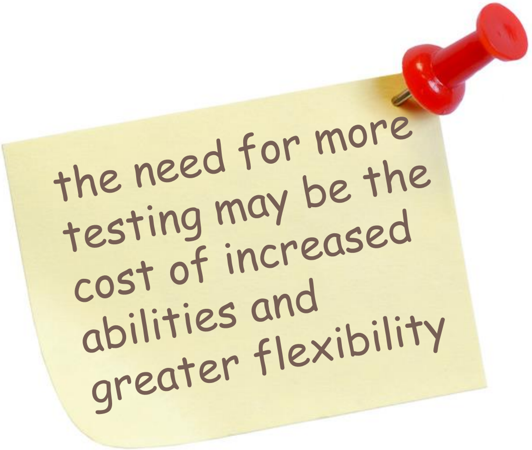
```
import java.awt.event.ActionEvent

class GRIController {
    def model
    def view

  def executeScript(ActionEvent evt = null) {
    doOutside {
          model.greeting = 'HI!'
    }
  }
}
```



Result: HI!

Testing

- dynamic languages don’t have the help of a strong type system
  - typos, etc. not uncovered until **run-time***
- testing required
  - more?
  - but testing is *always* required so not a problem?



the need for more testing may be the cost of increased abilities and greater flexibility

* but good IDEs can help quite a lot…many errors can be surfaced at *edit-time*

Testing…

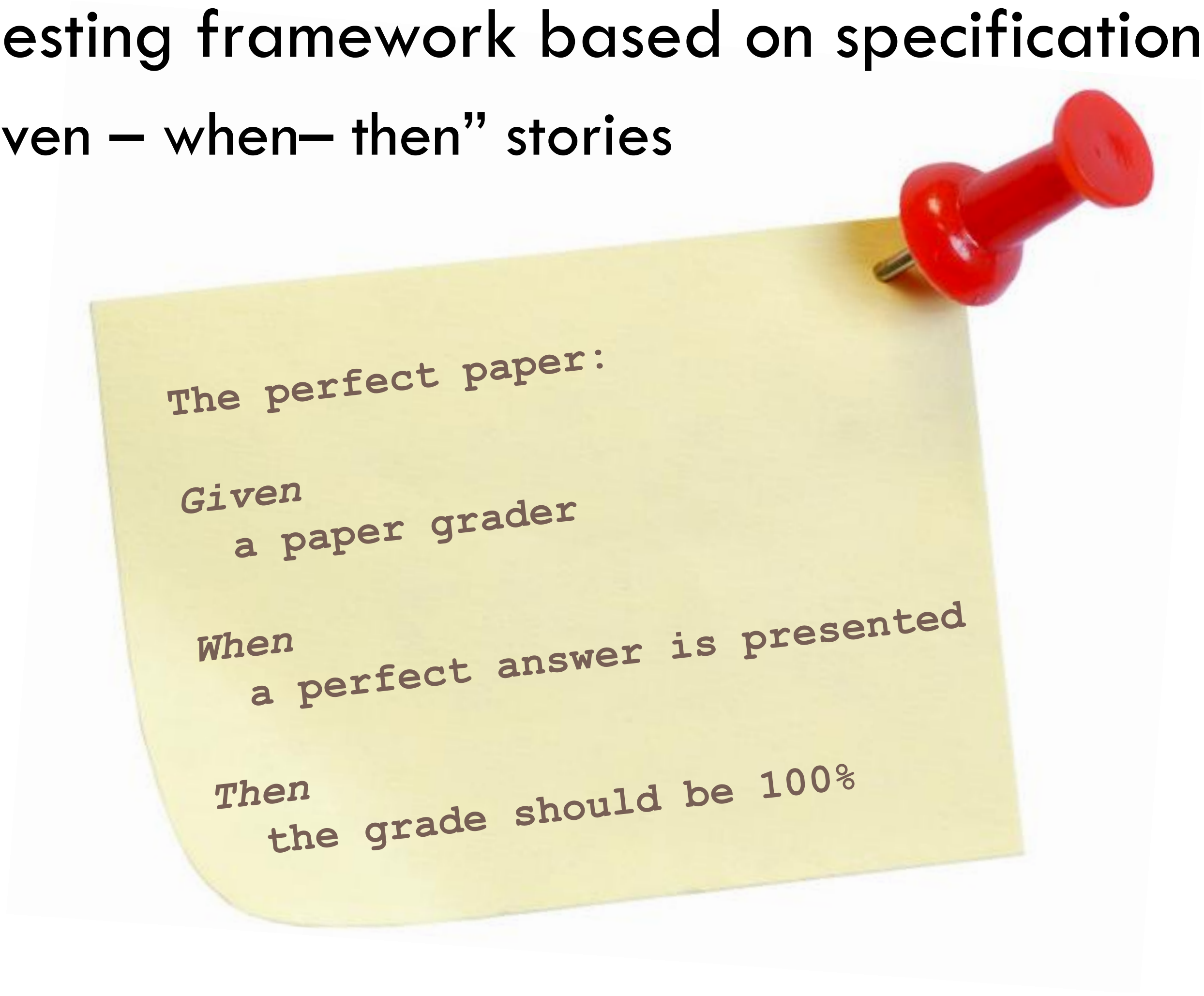
```
class Grader {
    def expectedAnswers
    def graderFileReader

    def grade(String s) {
        def candidateAnswers = graderFileReader.readGradesListFromFile(s)
        grade(candidateAnswers)
    }

    def grade(List candidateAnswers) {
        if (expectedAnswers?.size() != candidateAnswers?.size())
            -1.0
        else {
            def count = 0
            expectedAnswers.eachWithIndex {o,index ->
                if (o == candidateAnswers[index]) count ++
            }
            count / expectedAnswers.size()
        }
    }
}
class GraderFileReader {
    def readGradesListFromFile(name) {
        def f = new File(name)
        if (!f.exists())
          throw new Exception("File $name does not exist.")
        def txt = f.text
        txt?.split(',') as List
    }
}
```

classes under test

Spock

spock
the enterprise ready specification framework

- unit testing framework based on specifications
  - “given – when– then” stories



The perfect paper:

Given
  a paper grader

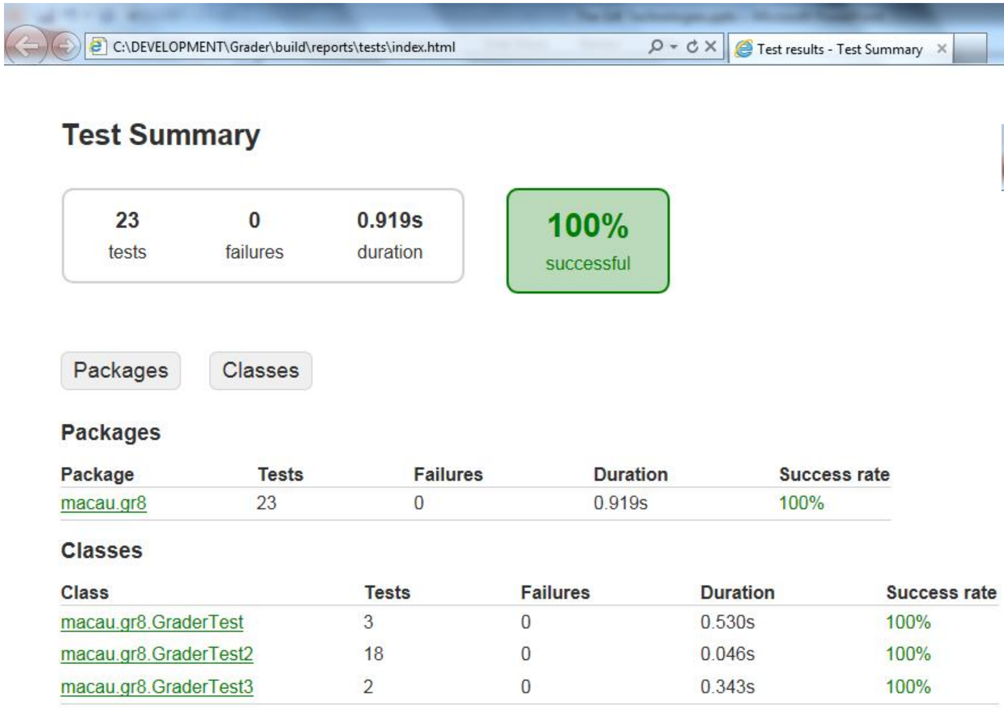
When
  a perfect answer is presented

Then
  the grade should be 100%

Spock…

```
public class GraderSpecification extends Specification {
    def grader

    def "The perfect paper"() {
        when: "A perfect answer is presented"
          def result = grader.grade(['a','b','c'])
        then: "The grade should be 100%"
          result == 1.0
    }

    def "The worst paper"() {
        when: "No answers are given"
          def result = grader.grade([])
        then: "An error should be indicated"
          result == -1.0
    }

    def "A poor paper"() {
        when: "A fairly poor paper is presented"
          def result = grader.grade(['a','c','b'])
        then: "The grade should be 33%"
          result closeTo(0.33D, 0.01D)
    }

    def setup()  { grader = new Grader(expectedAnswers: ['a','b','c']) }
    def cleanup() { grader = null }
}
```

Spock



C:\DEVELOPMENT\Grader\build\reports\tests\index.html

Test results - Test Summary

Test Summary

23 tests | 0 failures | 0.919s duration

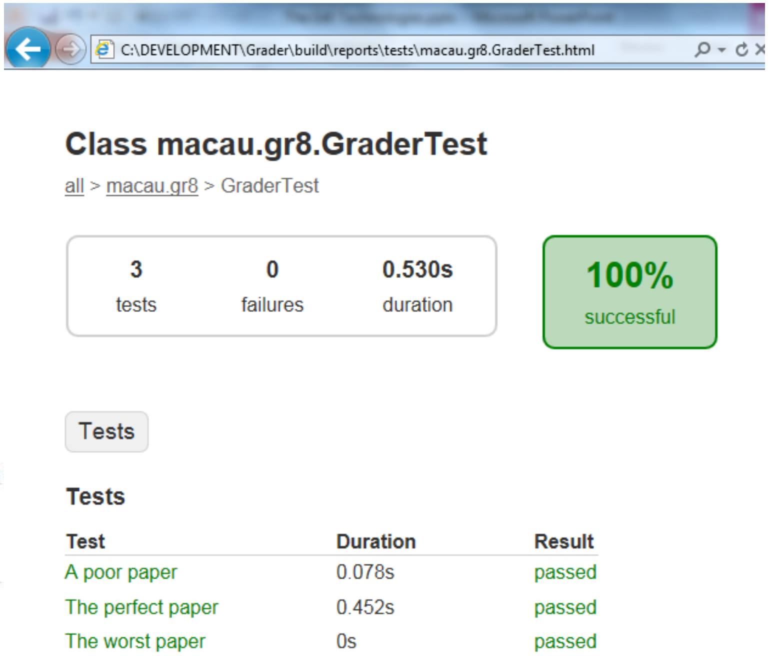
100% successful

Packages

Package	Tests	Failures	Duration	Success rate
macau.gr8	23	0	0.919s	100%

Classes

Class	Tests	Failures	Duration	Success rate
macau.gr8.GraderTest	3	0	0.530s	100%
macau.gr8.GraderTest2	18	0	0.046s	100%
macau.gr8.GraderTest3	2	0	0.343s	100%



C:\DEVELOPMENT\Grader\build\reports\tests\macau.gr8.GraderTest.html

Class macau.gr8.GraderTest

all > macau.gr8 > GraderTest

3 tests | 0 failures | 0.530s duration

100% successful

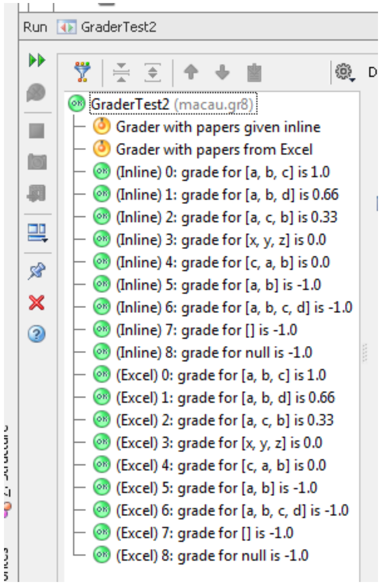
Tests

Test	Duration	Result
A poor paper	0.078s	passed
The perfect paper	0.452s	passed
The worst paper	0s	passed

Spock…

- table-driven parameterised testing

```
public class GraderSpecification2 extends Specification {
    @AutoCleanup(quiet = true)
    def grader = new Grader(expectedAnswers: ['a', 'b', 'c'])

    @Unroll("(Inline) #iterationCount: grade for #paper is #res")
    def "Grader with papers given inline"() {
        expect: "Grade an individual paper"
            that grader.grade(paper), closeTo(res, 0.01D)

        where: "With the following papers"
            paper                | res
            ['a', 'b', 'c']      | 1.0D
            ['a', 'b', 'd']      | 0.66D
            ['a', 'c', 'b']      | 0.33D
            ['x', 'y', 'z']      | 0.0D
            ['c', 'a', 'b']      | 0.0D
            ['a', 'b']           | -1.0D
            ['a', 'b', 'c', 'd'] | -1.0D
            []                   | -1.0D
            null                 | -1.0D
    }
}
```



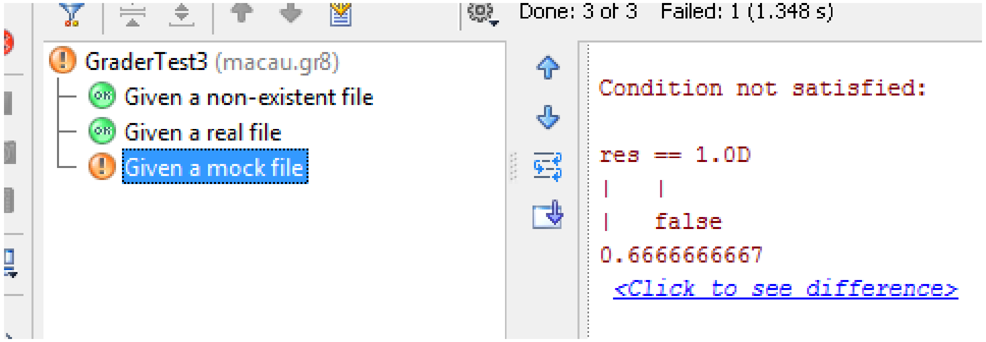
```
GraderTest2 (macau.gr8)
  Grader with papers given inline
  Grader with papers from Excel
  (Inline) 0: grade for [a, b, c] is 1.0
  (Inline) 1: grade for [a, b, d] is 0.66
  (Inline) 2: grade for [a, c, b] is 0.33
  (Inline) 3: grade for [x, y, z] is 0.0
  (Inline) 4: grade for [c, a, b] is 0.0
  (Inline) 5: grade for [a, b] is -1.0
  (Inline) 6: grade for [a, b, c, d] is -1.0
  (Inline) 7: grade for [] is -1.0
  (Inline) 8: grade for null is -1.0
  (Excel) 0: grade for [a, b, c] is 1.0
  (Excel) 1: grade for [a, b, d] is 0.66
  (Excel) 2: grade for [a, c, b] is 0.33
  (Excel) 3: grade for [x, y, z] is 0.0
  (Excel) 4: grade for [c, a, b] is 0.0
  (Excel) 5: grade for [a, b] is -1.0
  (Excel) 6: grade for [a, b, c, d] is -1.0
  (Excel) 7: grade for [] is -1.0
  (Excel) 8: grade for null is -1.0
```

"Green is Good"

Spock…

- mocking and expectations

```
class GraderSpecification3 extends Specification {
    @AutoCleanup(quiet = true)
    def grader = new Grader(expectedAnswers: ['a','b','c'])

    def "Given a mock file"() {
        setup: "Establish the grader with a mocked GraderFileReader"
          def graderFileReader = Mock(GraderFileReader)
          grader.graderFileReader = graderFileReader
          1 * graderFileReader.readGradesListFromFile(_) >> ['a','b','c']
          0 * _._

        when: "Grade a paper's answers from a given file"
          def res = grader.grade('rsrc/100pct.txt')

        then: "Ensure expected behaviour"
          res == 1.0D
    }
}
```

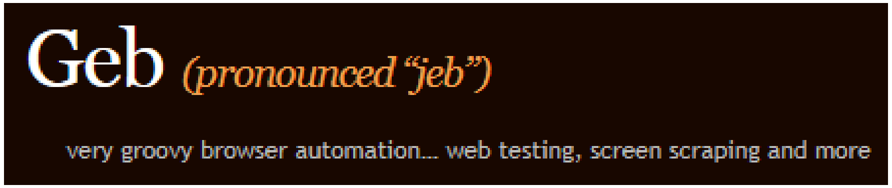


Done: 3 of 3   Failed: 1 (1.348 s)

- GraderTest3 (macau.gr8)
  - Given a non-existent file
  - Given a real file
  - Given a mock file

```
Condition not satisfied:

res == 1.0D
|   |
|   false
0.6666666667
 <Click to see difference>
```

Geb



Geb *(pronounced "jeb")*

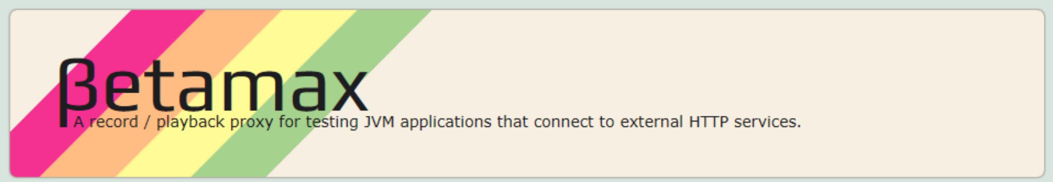
very groovy browser automation... web testing, screen scraping and more

- functional testing for the web
- easy-to-use DSL
  - no nasty C or XML like competing tools

Geb…

```
@Grapes([
    @Grab("org.gebish:geb-core:0.9.0"),
    @Grab("org.seleniumhq.selenium:selenium-firefox-driver:2.26.0"),
    @Grab("org.seleniumhq.selenium:selenium-support:2.26.0")
])
import geb.*

Browser.drive {
  go "http://www.google.com/"
  assert title == "Google"

  $("input", name: "q").value("wikipedia")
  $("input", value: "Google Search").click()

  assert title.endsWith("Google Search")

  def firstResultLink = $("li.g", 0).find("a.l")
  assert firstResultLink.text() == "Wikipedia, the free encyclopedia"
}
```

Betamax



Betamax

A record / playback proxy for testing JVM applications that connect to external HTTP services.

- test proxy/framework
  - first time: record; then: replay
- breaks dependencies between teams/systems during test/development
- functional mocking
- regression testing

Betamax...

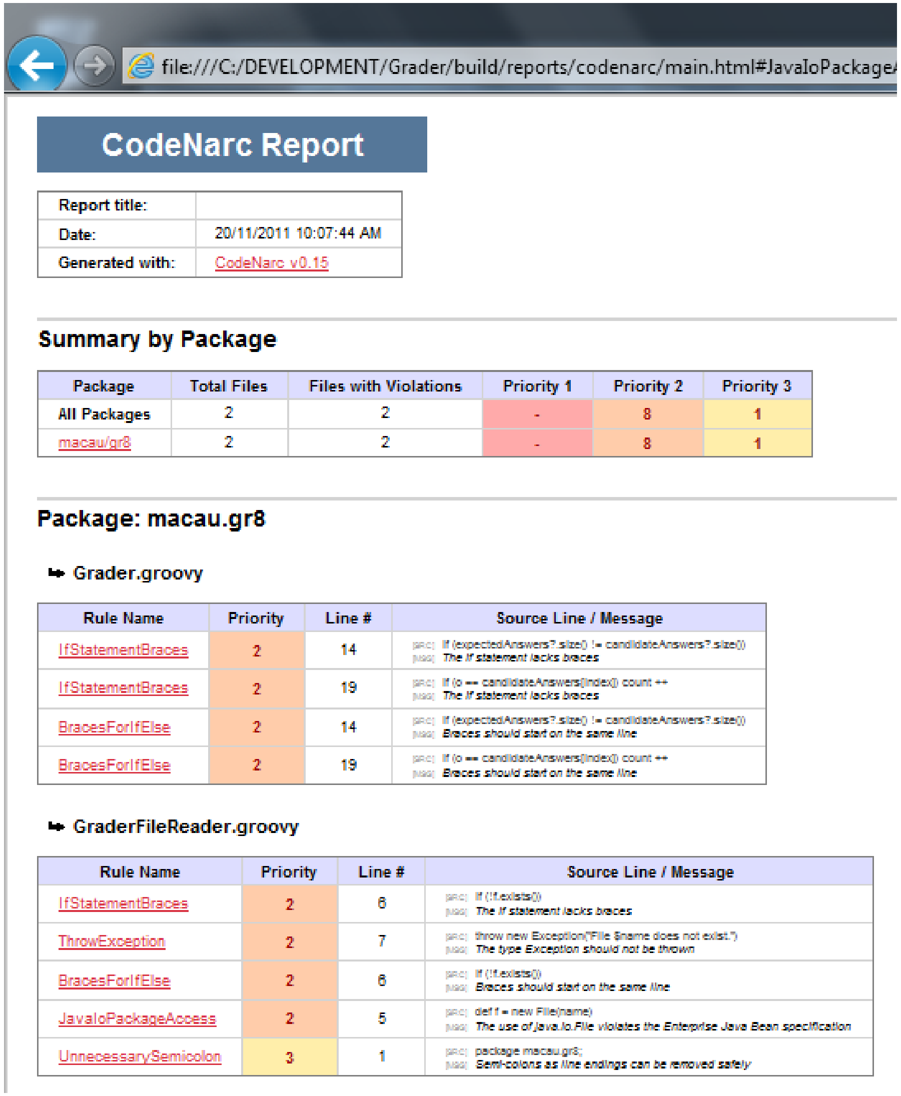
```
import geb.spock.GebSpec
import betamax.*
import org.junit.*
import spock.lang.*

class TransentiaSpec extends GebSpec {
    @Rule recorder = new Recorder()

    @Betamax(tape="transentia.betamax.tape")
    def "go to Transentia home page"() {
        setup:
            browser.driver.setProxy("localhost", 5555)
        when:
            go "http://www.transentia.com.au/"
        then:
            title.startsWith('Transentia')
        and:
          // some basic content checks
          def about = $("div.about")
          def aboutTitle = about.find("h2.title")
          aboutTitle.text() == "About Transentia"
          aboutTitle.next().text().contains("Gr8")
    }
}
```

Spock, Geb and Betamax all working together…

CodeNarc

- code inspections
  - configurable command-line tool
    - for use with Jenkins/development teams
  - checking for common whoopsies, gotchas, etc.
    - inconsistencies, unneeded/dead code
  - checks subtle/uncommon issues
    - threading, memory, resource usage

CodeNarc…



file:///C:/DEVELOPMENT/Grader/build/reports/codenarc/main.html#JavaIoPackageA

CodeNarc Report

Report title:	
Date:	20/11/2011 10:07:44 AM
Generated with:	CodeNarc v0.15

Summary by Package

Package	Total Files	Files with Violations	Priority 1	Priority 2	Priority 3
All Packages	2	2	-	8	1
macau/gr8	2	2	-	8	1

Package: macau.gr8

➥ Grader.groovy

Rule Name	Priority	Line #	Source Line / Message
IfStatementBraces	2	14	[SRC] if (expectedAnswers?.size() != candidateAnswers?.size()) [MSG] *The if statement lacks braces*
IfStatementBraces	2	19	[SRC] if (o == candidateAnswers[index]) count ++ [MSG] *The if statement lacks braces*
BracesForIfElse	2	14	[SRC] if (expectedAnswers?.size() != candidateAnswers?.size()) [MSG] *Braces should start on the same line*
BracesForIfElse	2	19	[SRC] if (o == candidateAnswers[index]) count ++ [MSG] *Braces should start on the same line*

➥ GraderFileReader.groovy

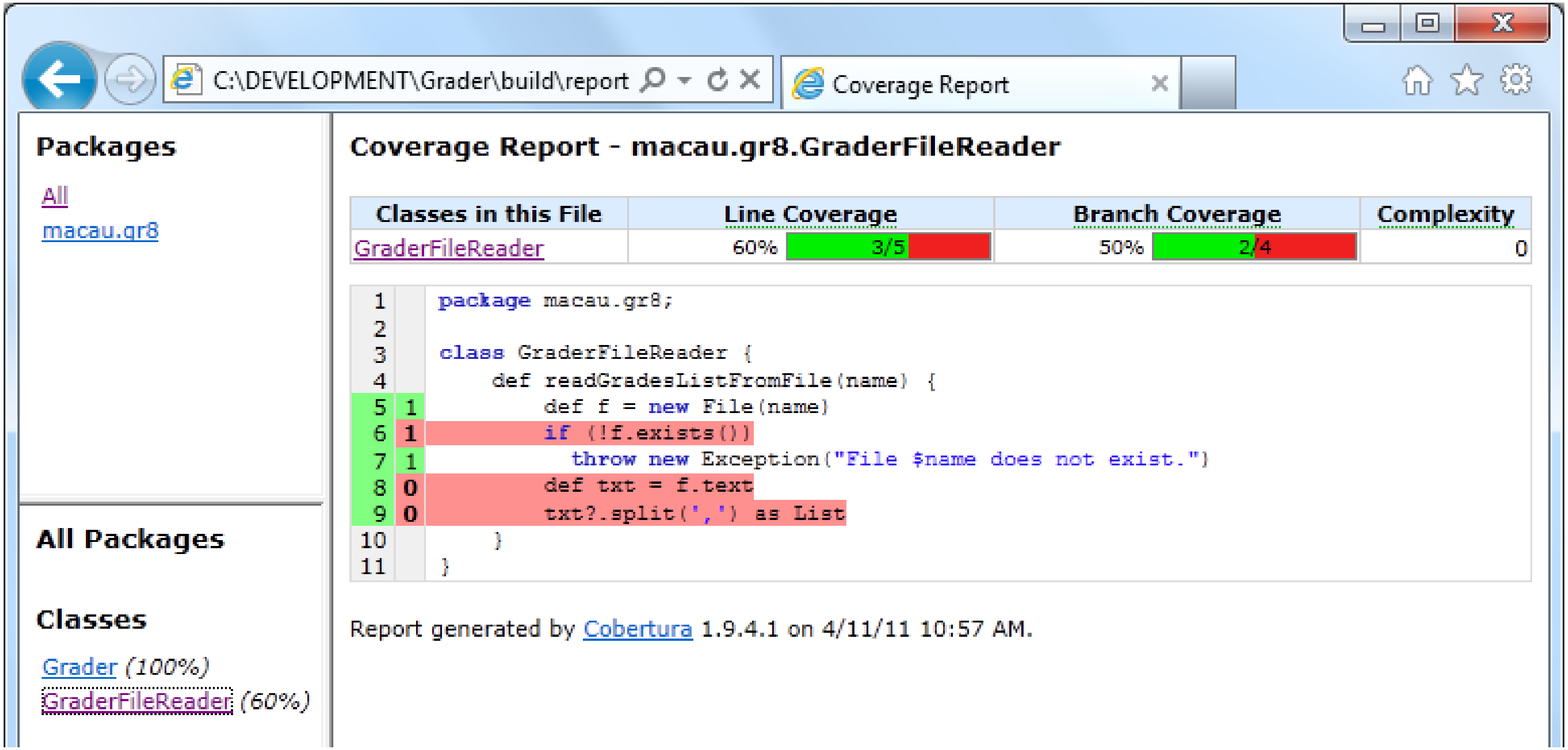
Rule Name	Priority	Line #	Source Line / Message
IfStatementBraces	2	6	[SRC] if (!f.exists()) [MSG] *The if statement lacks braces*
ThrowException	2	7	[SRC] throw new Exception("File $name does not exist.") [MSG] *The type Exception should not be thrown*
BracesForIfElse	2	6	[SRC] if (!f.exists()) [MSG] *Braces should start on the same line*
JavaIoPackageAccess	2	5	[SRC] def f = new File(name) [MSG] *The use of java.io.File violates the Enterprise Java Bean specification*
UnnecessarySemicolon	3	1	[SRC] package macau.gr8; [MSG] *Semi-colons as line endings can be removed safely*

```
ruleset {
    description 'A Sample Groovy RuleSet'
    AssignmentInConditional
    StaticCalendarField
    SynchronizedOnBoxedPrimitive
    ReturnsNullInsteadOfEmptyCollection
    SimpleDateFormatMissingLocale
    DuplicateNumberLiteral
    CatchIllegalMonitorStateException
    …
}
```

Cobertura

- code coverage testing
  - command-line tool
    - configurable
  - show what has been tested
  - guide what further tests need to be created

Cobertura



Packages

All
macau.gr8

All Packages

Classes

Grader *(100%)*
GraderFileReader *(60%)*

Coverage Report - macau.gr8.GraderFileReader

Classes in this File	Line Coverage	Branch Coverage	Complexity
GraderFileReader	60% 3/5	50% 2/4	0

```
   package macau.gr8;

   class GraderFileReader {
       def readGradesListFromFile(name) {
1          def f = new File(name)
1          if (!f.exists())
1            throw new Exception("File $name does not exist.")
0          def txt = f.text
0          txt?.split(',') as List
       }
   }
```

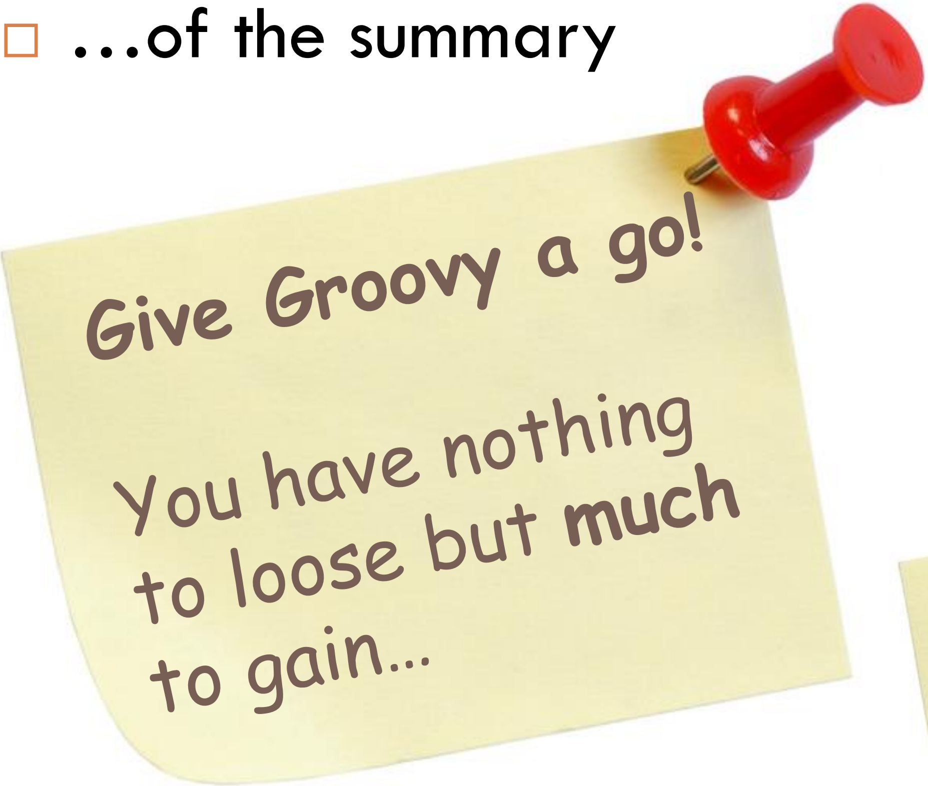
Report generated by Cobertura 1.9.4.1 on 4/11/11 10:57 AM.

Summary

- *an agile and **dynamic language** for the **Java Virtual Machine***
- *builds upon the strengths of Java but has **additional power features** inspired by languages like Python, Ruby and Smalltalk*
- *makes **modern programming features** available to Java developers with **almost-zero learning curve***
- *supports **Domain-Specific Languages** and other compact syntax so your code becomes **easy to read and maintain***
- *makes writing shell and build scripts easy with its **powerful processing primitives**, OO abilities and an Ant DSL*
- *increases developer productivity by **reducing scaffolding** code when developing web, GUI, database or console applications*
- ***simplifies testing** by supporting unit testing and mocking out-of-the-box*
- *seamlessly **integrates with all existing Java objects and libraries***
- *compiles straight to Java bytecode so you can use it anywhere you can use Java*

Summary

- …of the summary



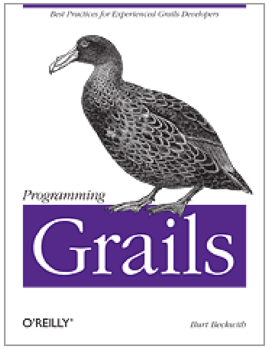
Give Groovy a go!

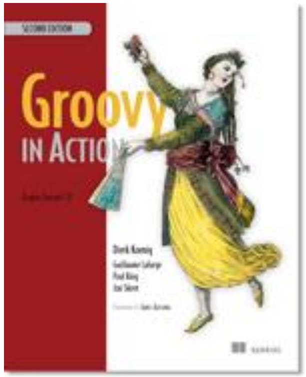
You have nothing to loose but **much** to gain…

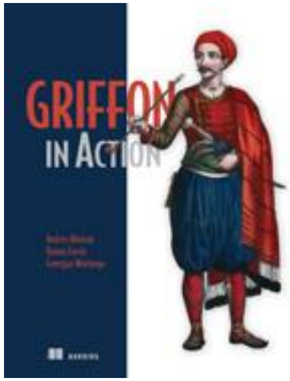
…and there's *lots* more to discover…

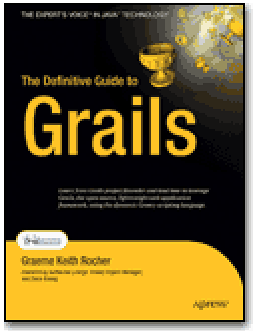
Learn More

- Resources
  - http://www.transentia.com.au
  - user@groovy.codehaus.org
  - http://groovy.codehaus.org
  - http://gradle.org
  - http://griffon.codehaus.org
  - http://grails.codehaus.org
  - http://jenkins-ci.org
  - http://gant.codehaus.org
  - http://gmetrics.sourceforge.net
  - http://cobertura.sourceforge.net
  - http://easyb.org
  - http://jfugue.org
  - http://jscience.org
  - http://codenarc.sourceforge.net
  - http://code.google.com/p/spock
  - http://robfletcher.github.com/betamax
  - http://gebish.org
  - http://mrhaki.com
  - http://groovyblogs.org
  - http://groovymag.com
  - http://vertx.io/









END

謝謝您們的聆聽

Thanks for listening

(questions?)