Introduction to Groovy

Part I – Compare & Contrast with Java

WHERE LESS REALLY IS MORE…

Overview

About Groovy

Getting Started

Groovy Scripts & Classes

POGO v POJO

Groovy Strings

Dynamic Typing

A Taste of Dynamic Behavior

Some Gotchas

What’s Next

About Groovy

A Brief History…

2003

- James Strachan announces the birth of Groovy as a project
- Java language look-and-feel
- Dynamic features modeled from Ruby, Python
- Guillame LaForge starts exploring Groovy

2004

- Groovy submitted to JCP (JSR 241)

2006

- James Strachan moves on from the Groovy project
- Guillame LaForge takes over

About Groovy

A Brief History…

2007

- Groovy 1.0 released after several betas
- G2One company is formed by Graeme Rocher (creator of Grails)
- LaForge is made Groovy Project Lead
- Groovy 1.0 and 1.1 released, then rebranded to 1.5 to reflect rapid changes

2008

- SpringSource acquires G2One

2009

- VMware acquires SpringSource

About Groovy

A Brief History…

2012

- Slow JCP process is abandoned and LaForge lists JSR 241 as dormant
- Groovy 2.0 released

2013

- VMware spins off SpringSource to Pivotal Software, Inc.

2014

- Groovy v2.3 released
- Current version is 2.3.6
  - Includes traits, Java 8 support, and more

About Groovy

So what *exactly* is Groovy?

- Groovy is a dynamic language based on Java
- Groovy objects *are* java.lang.Objects
- Groovy objects implement groovy.lang.GroovyObject interface
- Groovy GDK defines Groovy-specific types
- Groovy also dynamically *enhances* artifacts in the JDK

Getting Started

For this talk, we'll get Groovy manually. We'll discuss including Groovy as a build dependency in Part II

1. Download zip distribution from groovy.codehaus.org/Download
2. Unzip to install it
3. Set GROOVY_HOME
4. Set JAVA_HOME
5. Begin the Groovy love

Groovy Scripts and Classes

```
// file: Hello.groovy
// Groovy code can be in a class like Java
class Hello {
  static void main(args) {
    println “Hello World!”
  }
}
```

```
// file: SayHello.script
// Groovy code can be in a script
println “Hello World!”
```

POGO v POJO

Let’s start with a Java POJO and make it into a Groovy POGO…

Groovy Strings

Let’s explore the four ways we can make String literals…

Dynamic Typing

Duck Typing…

If it walks like a duck and quacks like a duck, let’s treat it like a duck…

Dynamic Behaviors

With great power, comes great responsibility…

Some Gotchas

You can almost rename a .java source file to .groovy, but not quite…

In Groovy:

- a == b means a.equals(b)
- By default, no way to declare package-private visibility
- do/while not supported
- Floating point literals are BigDecimals, not doubles
- Array literal declaration won’t work: int[] x = {1,2,3}. Use int[] x = [1,2,3]
- Groovy compiler will not check throws clause on method as all exceptions treated the same way by the compiler

What’s Next

At our next meeting we will have Part II of this Groovy introduction.

We will explore the following topics:

- Groovy operators
- Closures
- Range types
- Collections API enhancements
- Other JDK enhancements
- Builders
- More…

Resources & Links

Some Groovy history

- http://radio-weblogs.com/0112098/2003/08/29.html
- http://glaforge.appspot.com/article/groovy-s-birthday-and-news

Groovy

- http://groovy.codehaus.org
- http://groovy.codehaus.org/Differences+from+Java