Groovy

Programming Language Survey[Groovy]

Presented by Vinoth Kumar C
CB.EN.U4.CYS21085
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 23, 2023

AMRITA VISHWA VIDYAPEETHAM

Outline

About Groovy

- Groovy is a programming language that was created in 2003 by James Strachan.
- Powerful, object oriented, and dynamic language (has features similar to other popular dynamic languages like Python, Ruby and Smalltalk)
- Strachan designed Groovy that was easy to use and that integrated well with Java, which was and still is a widely used programming language.
- Since its release, Groovy has gained popularity among developers who work with Java, as it allows them to write code that is more concise and expressive than traditional Java code. Groovy can be used to write a wide range of applications, including web applications, scripting tools, and test automation frameworks.
- Integrates smoothly with any Java program and libraries.
- Very suitable for scripting though not limited to scripting.

A bit of history

- August 2003 - James Strachan first talked about the development of Groovy.
- Between 2004 - 2006 - Several versions were released.
- January 2007 - After the JCP standardization process began, the version numbering changed and a version called "1.0" was released on.
- December 2007 - Groovy 1.1 Final was released and immediately rebranded as Groovy 1.5 as a reflection of the many changes made.
- November 2008 - Spring Source acquired the Groovy and Grails company (G20ne).
- August 2009 - VMWare acquired SpringSource
- July 2012 - Groovy 2.0 was released. Added static compilation and a static type checker to Groovy.
- April 2013 - Groovy and Grails formed part of Pivotal (JV b/w EMC and VMware) product portfolio.
- April 2015 - Pivotal ceased sponsoring Groovy and Grails from April 2015. Groovy submitted to become a project at The Apache Software Foundation

Features of Groovy

Groovy is a dynamic, object-oriented programming language that is designed to be highly compatible with Java, while also offering a number of features and syntax that make it more concise and expressive than Java. Here are some key features of Groovy:

- **Dynamic Typing:**
  Groovy supports dynamic typing, which means that you don't have to declare the type of a variable before using it. This makes it easier and quicker to write code, and can help reduce the amount of boilerplate code that you have to write.
- **Closures:**
  Groovy supports closures, which are functions that can be passed around as variables. This makes it easy to write code that is more expressive and flexible.
- **Integration with Java:**
  Groovy is designed to be highly compatible with Java, which means that you can use existing Java libraries and frameworks in your Groovy code. You can also call Groovy code from Java code and vice versa.

Features of Groovy

- **Meta programming:**
  Groovy supports meta programming, which means that you can modify the behavior of classes and objects at runtime. This makes it possible to write code that is more flexible and adaptable to changing requirements.
- **Concise Syntax:**
  Groovy has a concise and expressive syntax that makes it easier and quicker to write code than Java. For example, Groovy supports optional semicolons and parentheses, and has a simplified syntax for defining classes and methods.
- **Domain-Specific Languages (DSLs):**
  Groovy is often used to create DSLs, which are small, specialized languages that are tailored to specific tasks or domains. Groovy's concise syntax and support for metaprogramming make it well-suited for creating DSLs.

Overall, Groovy is a versatile and powerful language that offers a number of features and syntax that make it easier and more expressive to write code than Java.

Style Guide

- Do not use semicolon for terminating statement / end of line
- Not using return statement
- Do not use both def and type while declaring variables (def String city = "Blore")use either def or type (for scripting def is more suitable for most cases)
- Do not use explicitly public for classes and methods as Groovy makes them public by default
- Use Groovy properties instead of private backing field and java style getter, setter
- User string interpolation instead of manually formatting the string
- Try to use Groovy Collection classes (range, list, map) before Java collection classes

Paradigm followed

- Groovy is a multi-paradigm programming language that supports a variety of programming paradigms, including object-oriented programming, functional programming, and procedural programming.
- At its core, Groovy is an object-oriented language, and its syntax and features are heavily influenced by Java. However, Groovy also incorporates features from other programming paradigms, such as closures, which are a key feature of functional programming.
- Groovy also supports metaprogramming, which is a programming paradigm that allows developers to modify the behavior of classes and objects at runtime. This is a feature that is not typically found in other programming languages, and it allows developers to create more flexible and adaptable code.
- Overall, Groovy is a flexible and adaptable language that allows developers to use a variety of programming paradigms, depending on the needs of their project.

First Program

- Groovy programs do not need a main entry point, as required in Java.
- No need to explicitly specify System.out.print]n - only println is sufficient.

Program 1

```
println("Hello Groovy")
```

- Method calls can omit the parentheses if there is at least one parameter and there is no ambiguity - so the program could be further simplified to

Program 2

```
println "Hello Groovy"
```

Program

Program 3

```
int a,b,c
a = 5
b = 20
c = a + b
println( "Sum of two number is :" + c) )
```

This prints the output " 25 ".

Program 4

```
int a,b,c
a = 20
b = 5
c = a - b
println( "Sum of two number is :" + c) )
```

This prints the output " 15 ".

Application

Groovy is used in a variety of real-world applications, particularly in the development of enterprise applications and web applications. Here are some examples:

- **Web Development:**
  Groovy is often used to develop web applications using popular web frameworks like Grails and Ratpack. These frameworks make it easy to build web applications with Groovy by providing pre-built components and features, such as database access, security, and routing.
- **Test Automation:**
  Groovy is widely used for test automation, particularly with the popular testing framework, Spock. Spock is a powerful and expressive testing framework that uses Groovy syntax to make it easier and more intuitive to write tests.
- **Build Automation:**
  Groovy is often used in build automation tools like Gradle, which is a popular build automation tool for Java applications. Gradle uses a Groovy-based DSL to define the build process, which makes it easier to write and maintain build scripts.

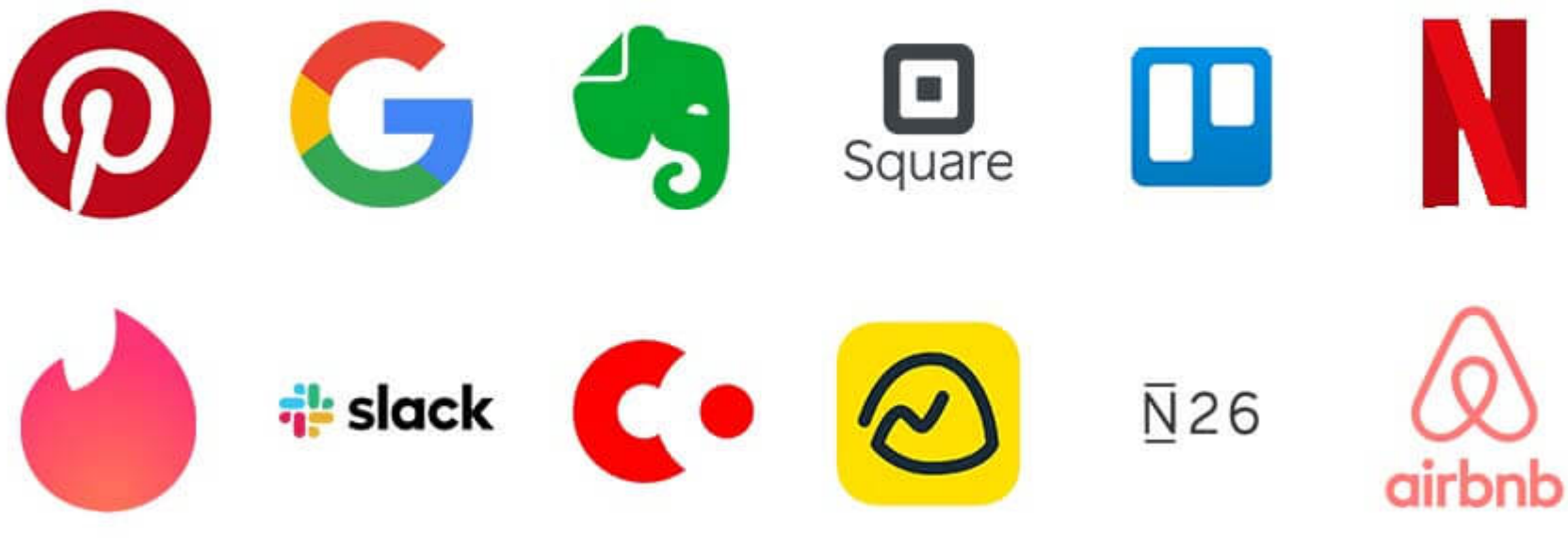
Application

- **Data Analysis:**
  Groovy is also used in data analysis and data processing applications. For example, the Apache Groovy-based tool, Apache NiFi, is used for data ingestion, processing, and transformation.
- **Scientific Computing:**
  Groovy is also used in scientific computing and numerical analysis. The GroovyLab project provides a scripting environment for scientific computing that allows scientists and researchers to use Groovy to perform complex calculations and simulations.

Overall, Groovy is a versatile and powerful language that is used in a wide range of applications, particularly in enterprise and web development, test automation, build automation, data analysis, and scientific computing.

Company using Groovy Language

Groovy is widely used by:



Thankyou!

Done by: Vinoth Kumar C [CB.EN.U4CYS21085]

References

https://groovy-lang.org/, https://dzone.com/articles/an-interview-with-james strachan-creator-of-groovy, Groovy website, Grails website, Apache NiFi website, GroovyLab website, https://groovy-lang.org/documentation.html
https://groovyconsole.appspot.com