Java / Database 認證日 2013
甲骨文大學Oracle University

淺談 Groovy 與 Gradle

林信良
http://openhome.cc

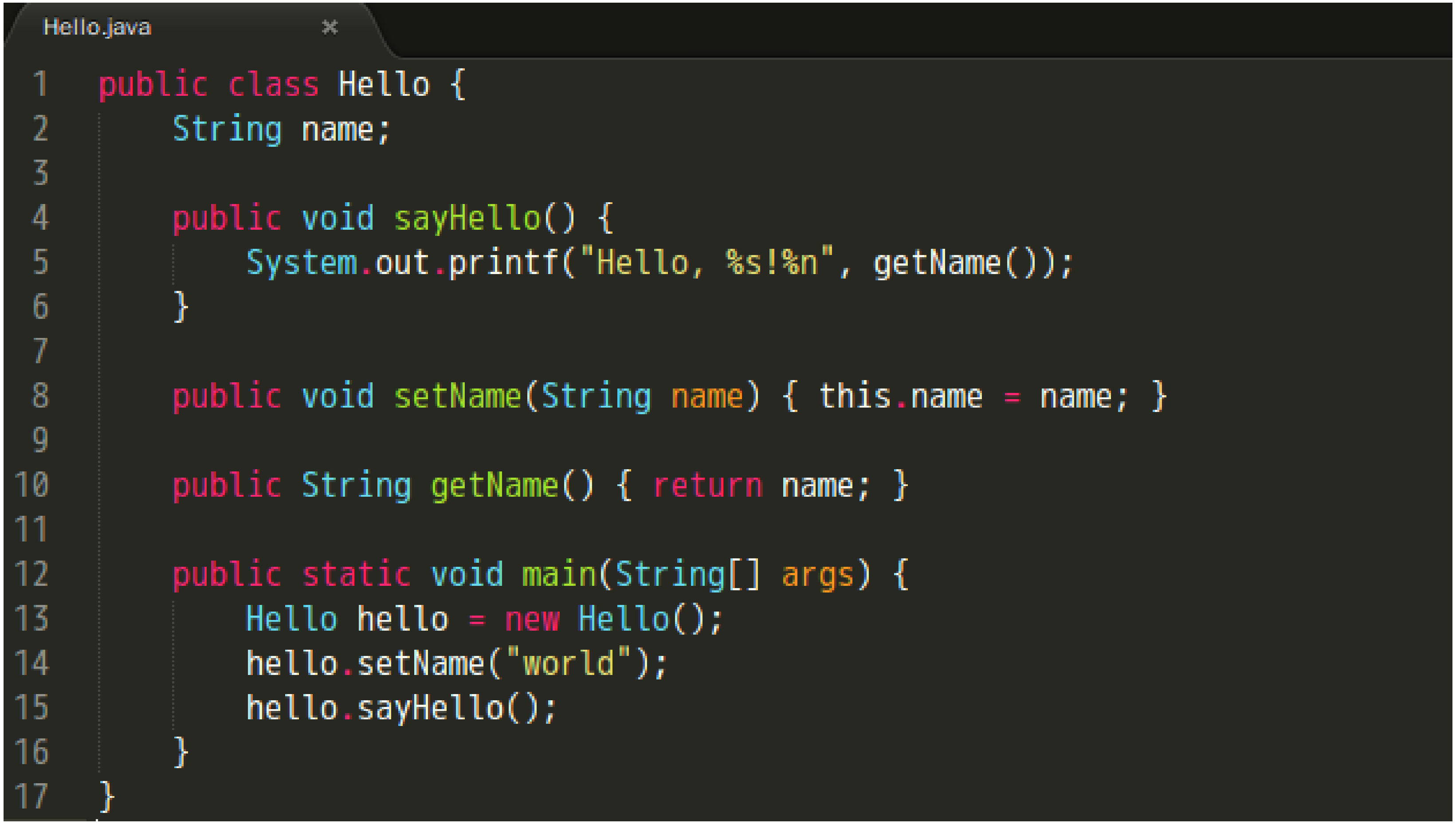
議程

- 從 Java 到 Groovy
- 使用 Groovy 建立 DSL
- 自動化工具 Gradle
- Gradle 與 IDE

從 Java 到 Groovy

Hello, world!

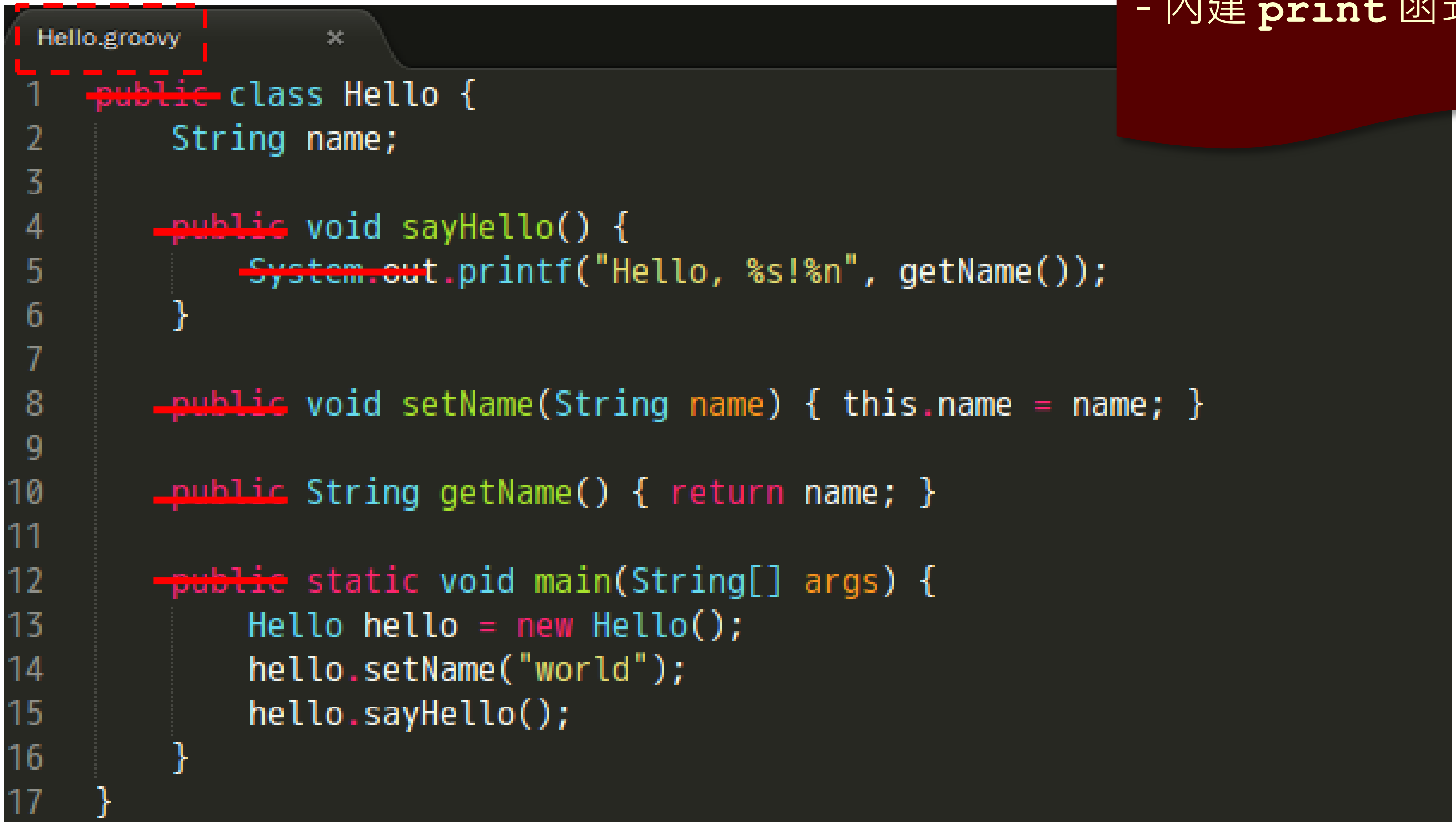
- 這是 Java ...



```java
public class Hello {
    String name;

    public void sayHello() {
        System.out.printf("Hello, %s!%n", getName());
    }

    public void setName(String name) { this.name = name; }

    public String getName() { return name; }

    public static void main(String[] args) {
        Hello hello = new Hello();
        hello.setName("world");
        hello.sayHello();
    }
}
```

- 這是 Groovy ...

- 預設 **public**
- 內建 **print** 函式

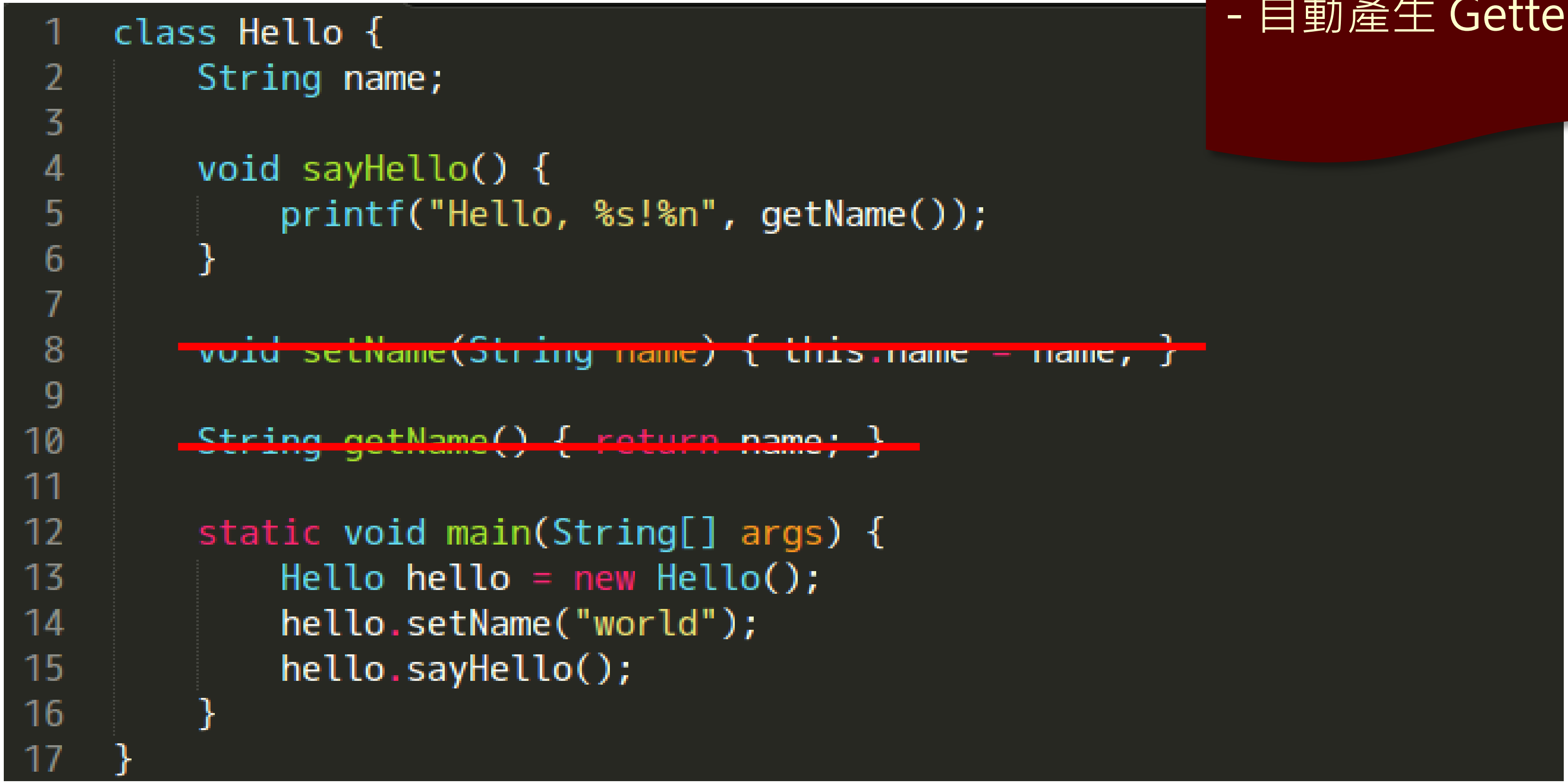


Hello.groovy

```
~~public~~ class Hello {
    String name;

    ~~public~~ void sayHello() {
        ~~System.out~~.printf("Hello, %s!%n", getName());
    }

    ~~public~~ void setName(String name) { this.name = name; }

    ~~public~~ String getName() { return name; }

    ~~public~~ static void main(String[] args) {
        Hello hello = new Hello();
        hello.setName("world");
        hello.sayHello();
    }
}
```

- 短一點 ...

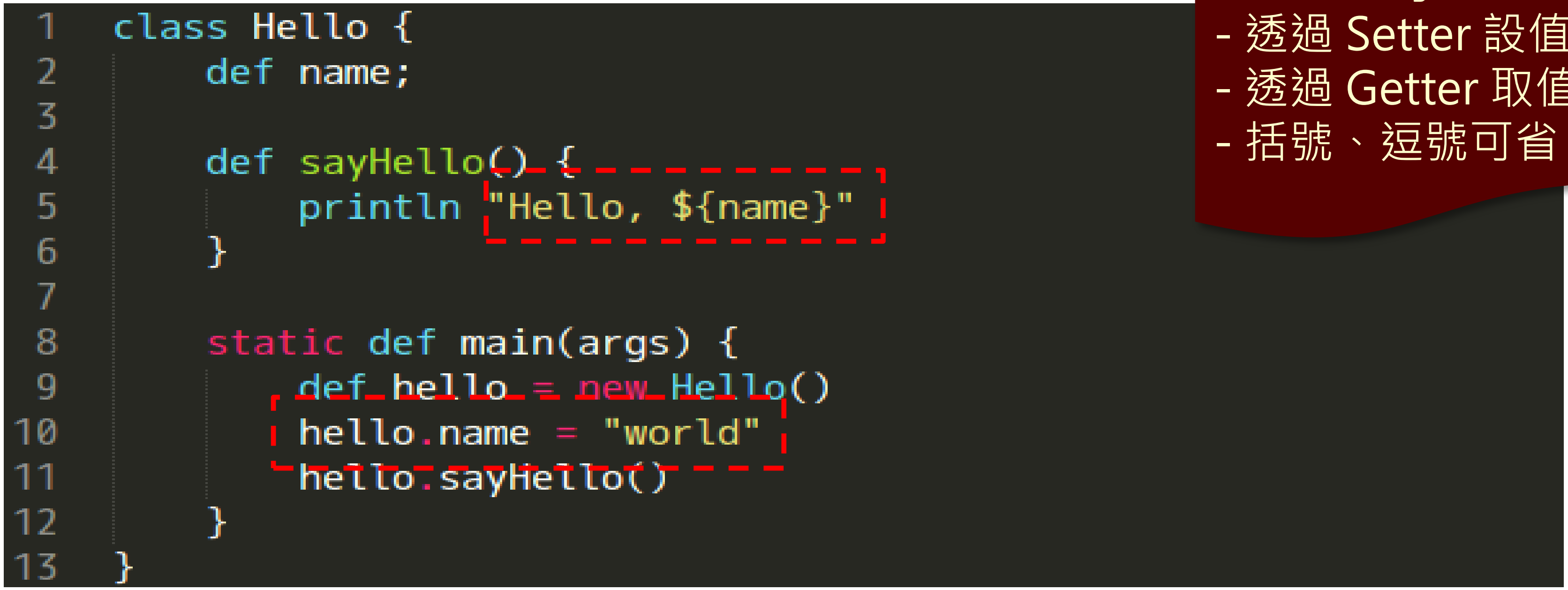
- 自動產生 Setter
- 自動產生 Getter



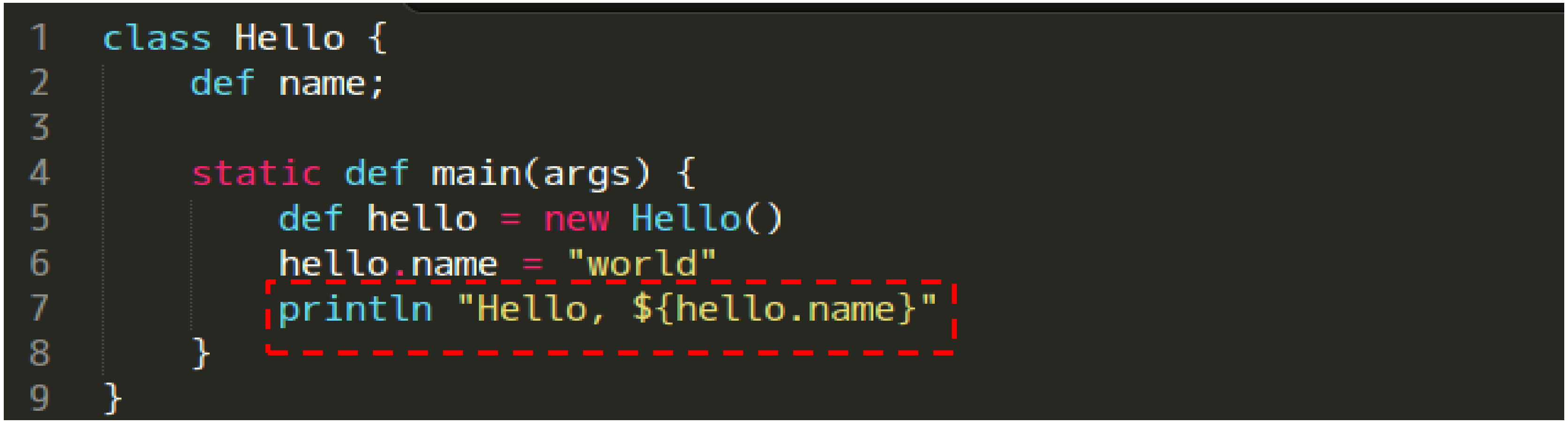
```
class Hello {
    String name;

    void sayHello() {
        printf("Hello, %s!%n", getName());
    }

    ~~void setName(String name) { this.name = name; }~~

    ~~String getName() { return name; }~~

    static void main(String[] args) {
        Hello hello = new Hello();
        hello.setName("world");
        hello.sayHello();
    }
}
```

- 省一點 ...

- Gstring 直譯
- 透過 Setter 設值
- 透過 Getter 取值
- 括號、逗號可省

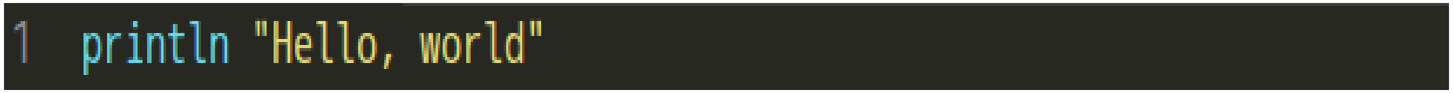


```
class Hello {
    def name;

    def sayHello() {
        println "Hello, ${name}"
    }

    static def main(args) {
        def hello = new Hello()
        hello.name = "world"
        hello.sayHello()
    }
}
```

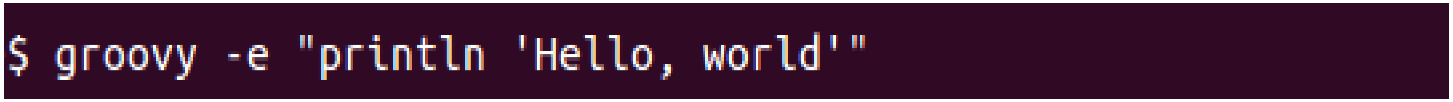


```
class Hello {
    def name;

    static def main(args) {
        def hello = new Hello()
        hello.name = "world"
        println "Hello, ${hello.name}"
    }
}
```

還要更少 ...

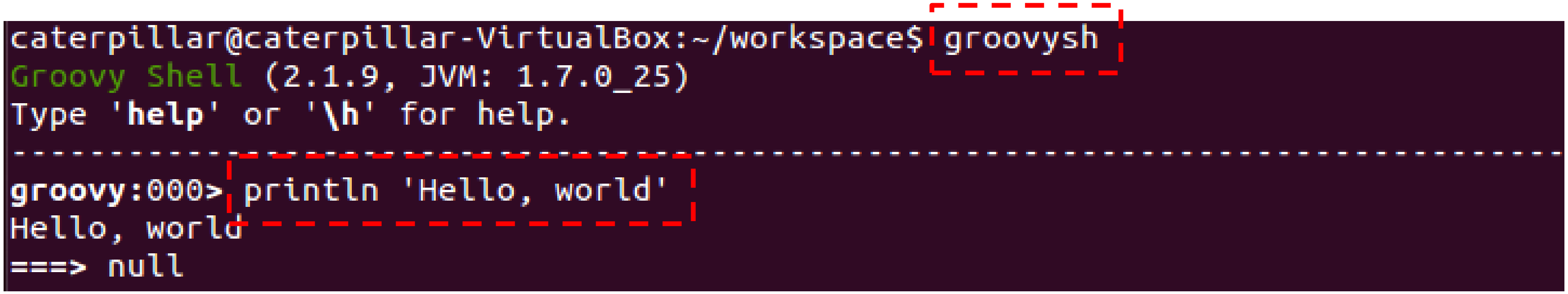


```
println "Hello, world"
```



```
$ groovy -e "println 'Hello, world'"
```

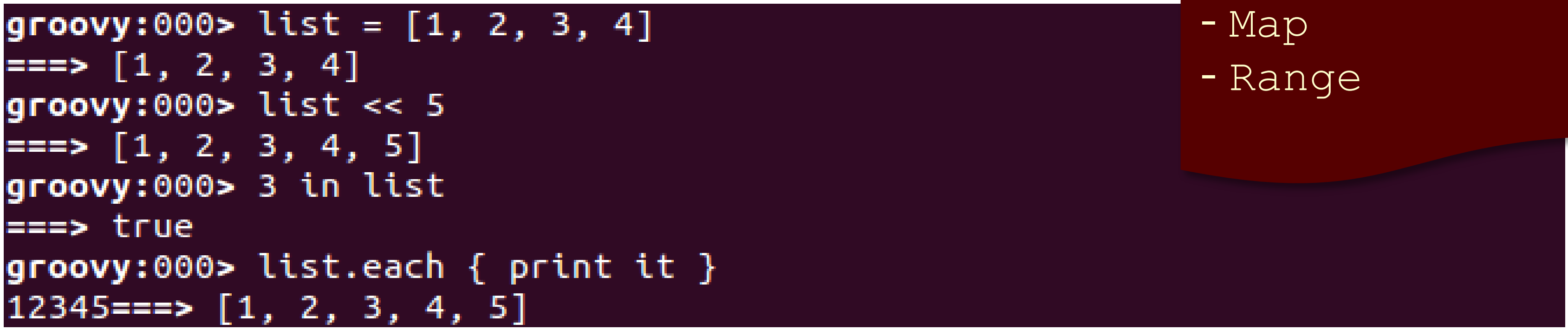
太少了？ ...



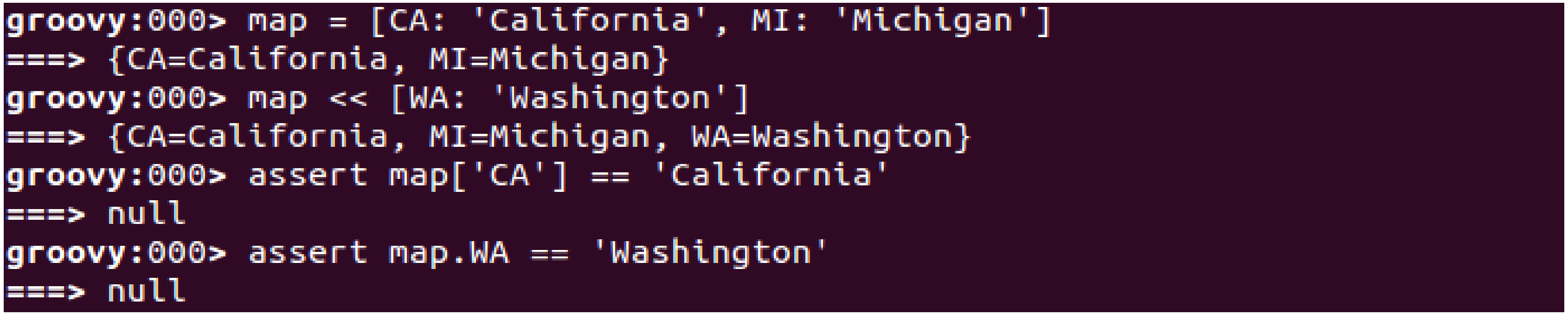
```
caterpillar@caterpillar-VirtualBox:~/workspace$ groovysh
Groovy Shell (2.1.9, JVM: 1.7.0_25)
Type 'help' or '\h' for help.
-------------------------------------------------------------------------------
groovy:000> println 'Hello, world'
Hello, world
===> null
```

- 多一點 ...

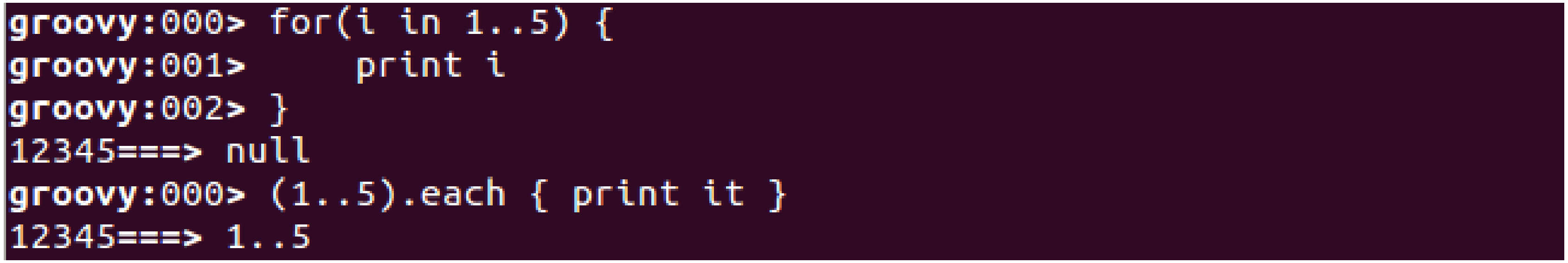
- List
- Map
- Range



```
groovy:000> list = [1, 2, 3, 4]
===> [1, 2, 3, 4]
groovy:000> list << 5
===> [1, 2, 3, 4, 5]
groovy:000> 3 in list
===> true
groovy:000> list.each { print it }
12345===> [1, 2, 3, 4, 5]
```

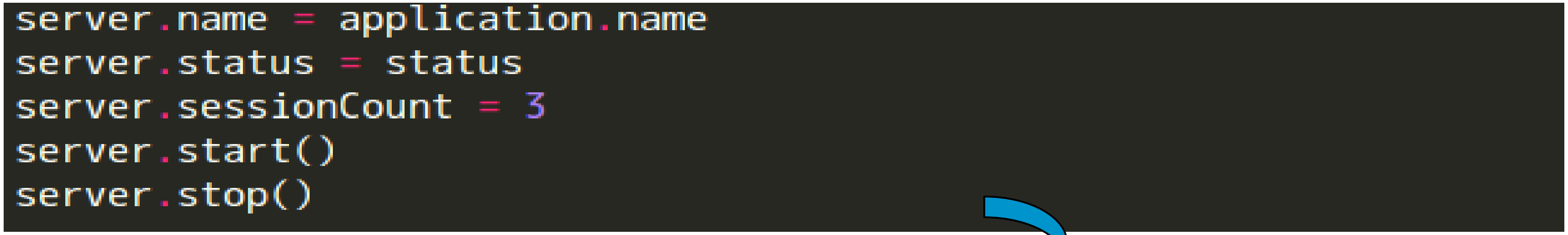


```
groovy:000> map = [CA: 'California', MI: 'Michigan']
===> {CA=California, MI=Michigan}
groovy:000> map << [WA: 'Washington']
===> {CA=California, MI=Michigan, WA=Washington}
groovy:000> assert map['CA'] == 'California'
===> null
groovy:000> assert map.WA == 'Washington'
===> null
```

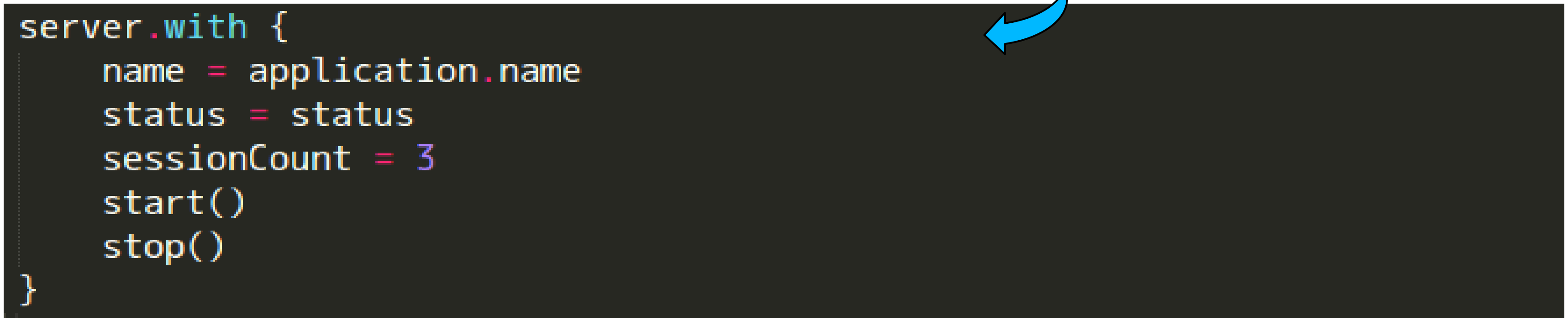


```
groovy:000> for(i in 1..5) {
groovy:001>     print i
groovy:002> }
12345===> null
groovy:000> (1..5).each { print it }
12345===> 1..5
```

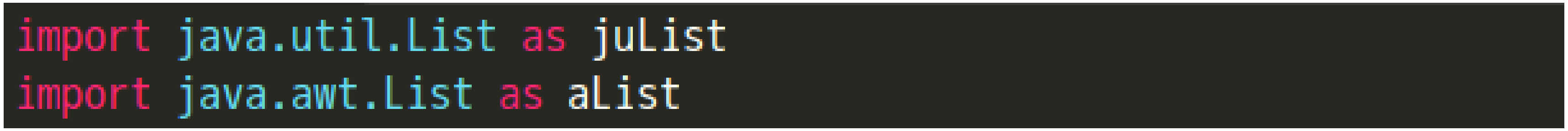
- 方便一點 …



```
server.name = application.name
server.status = status
server.sessionCount = 3
server.start()
server.stop()
```

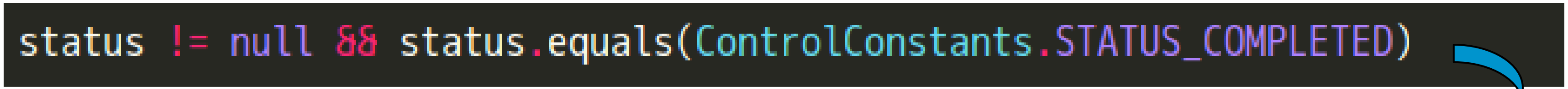


```
server.with {
    name = application.name
    status = status
    sessionCount = 3
    start()
    stop()
}
```

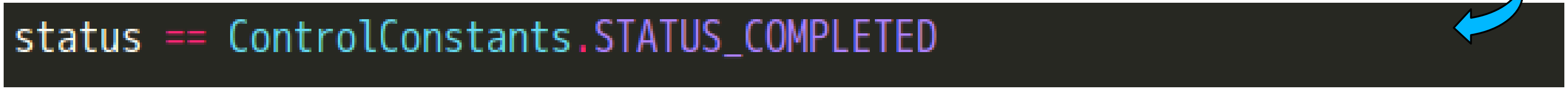


```
import java.util.List as juList
import java.awt.List as aList
```

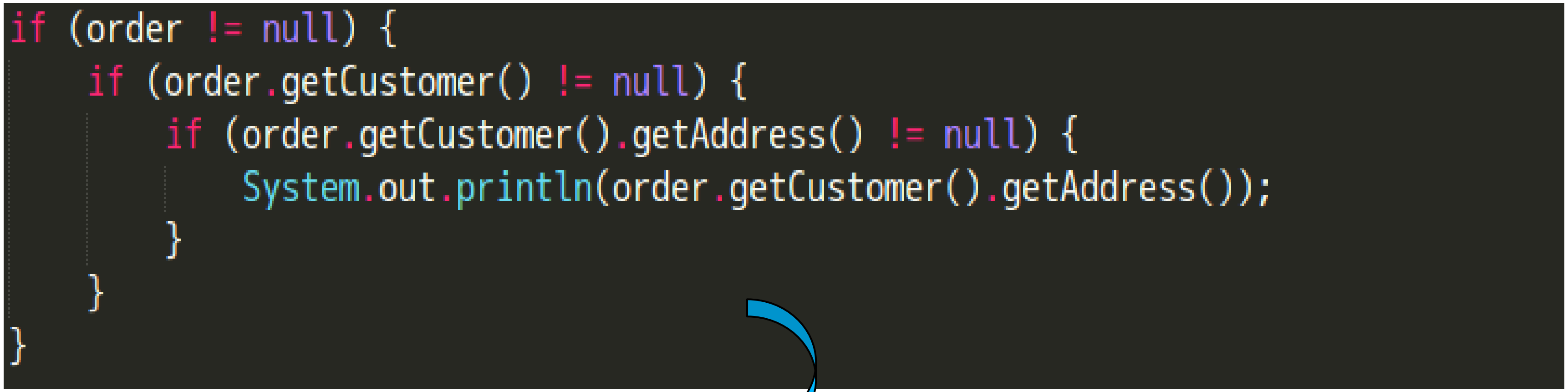
- 安全一點 ...



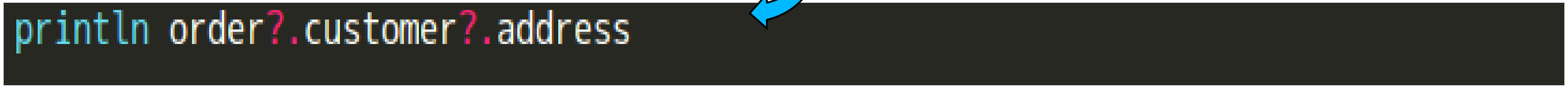
```
status != null && status.equals(ControlConstants.STATUS_COMPLETED)
```



```
status == ControlConstants.STATUS_COMPLETED
```



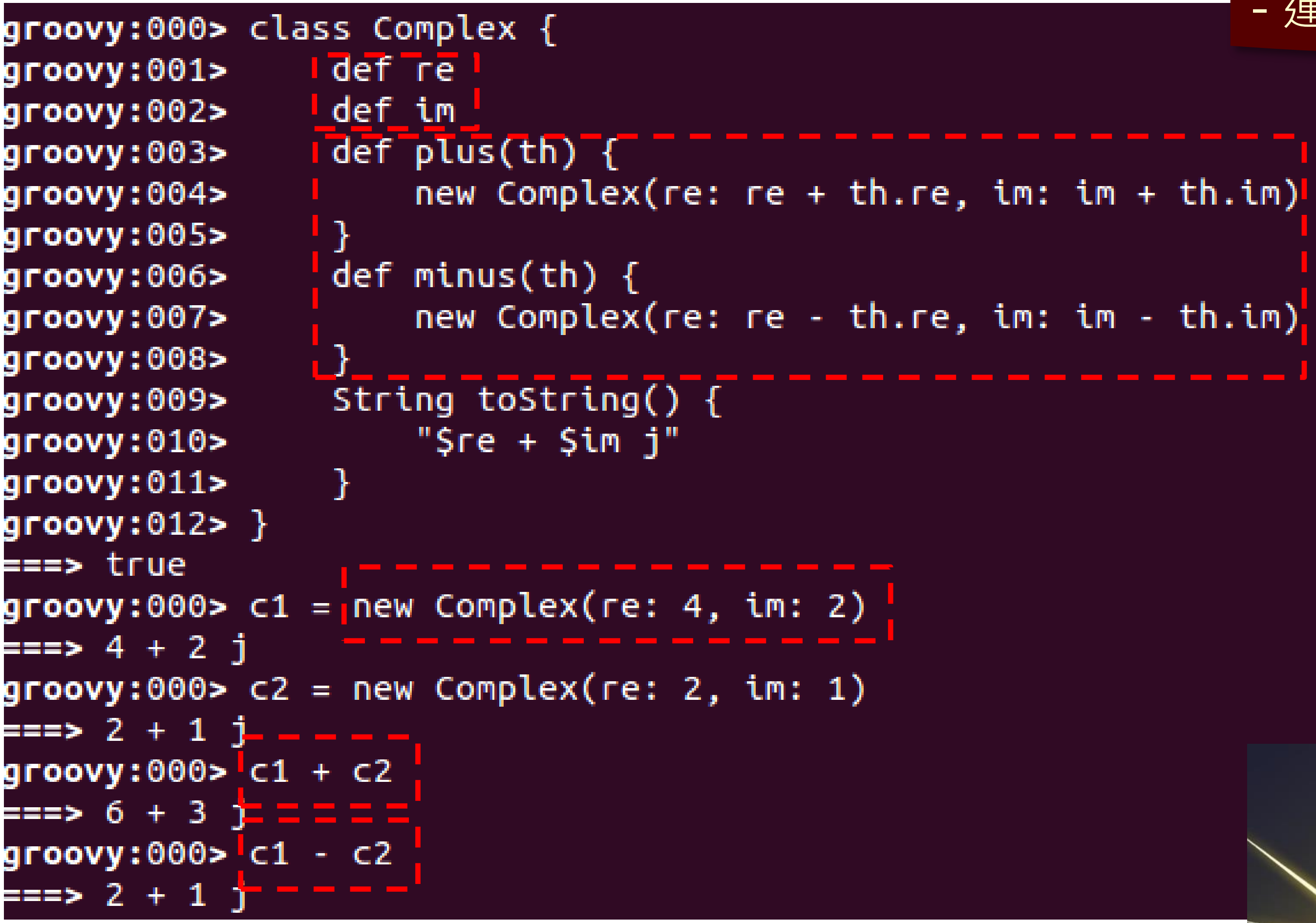
```
if (order != null) {
    if (order.getCustomer() != null) {
        if (order.getCustomer().getAddress() != null) {
            System.out.println(order.getCustomer().getAddress());
        }
    }
}
```



```
println order?.customer?.address
```

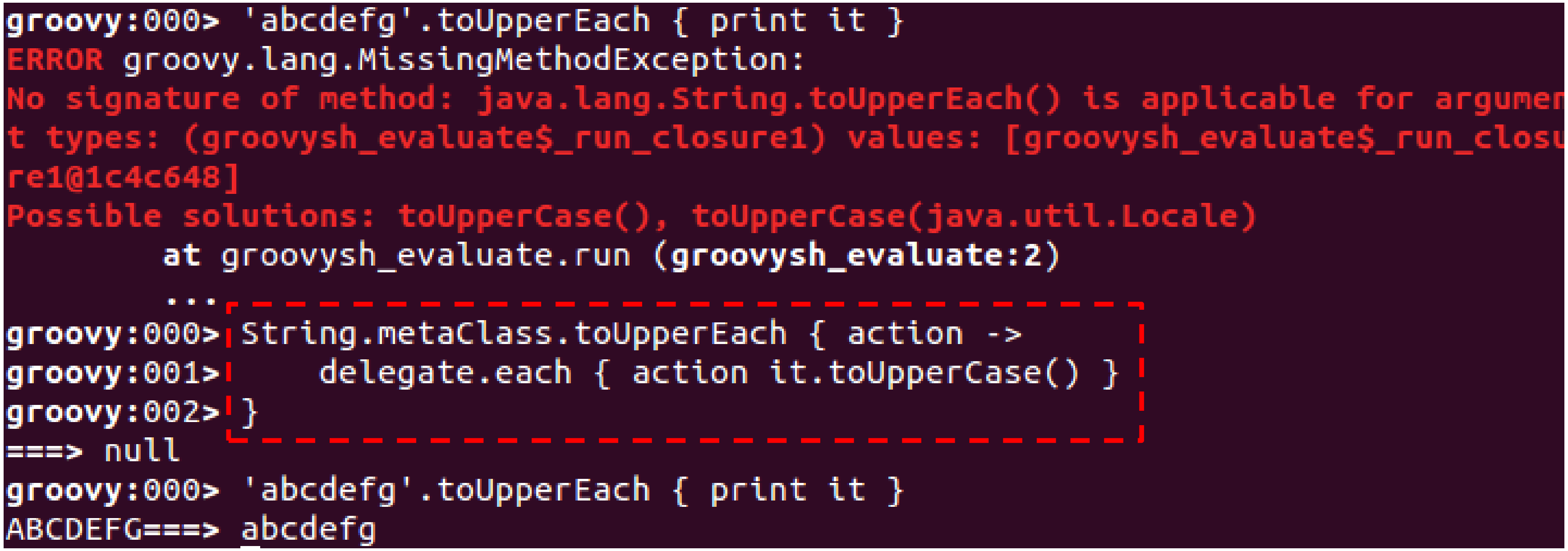
- 強大一點 ...

- 建構式
- 運算子重載



```
groovy:000> class Complex {
groovy:001>     def re
groovy:002>     def im
groovy:003>     def plus(th) {
groovy:004>         new Complex(re: re + th.re, im: im + th.im)
groovy:005>     }
groovy:006>     def minus(th) {
groovy:007>         new Complex(re: re - th.re, im: im - th.im)
groovy:008>     }
groovy:009>     String toString() {
groovy:010>         "$re + $im j"
groovy:011>     }
groovy:012> }
===> true
groovy:000> c1 = new Complex(re: 4, im: 2)
===> 4 + 2 j
groovy:000> c2 = new Complex(re: 2, im: 1)
===> 2 + 1 j
groovy:000> c1 + c2
===> 6 + 3 j
groovy:000> c1 - c2
===> 2 + 1 j
```

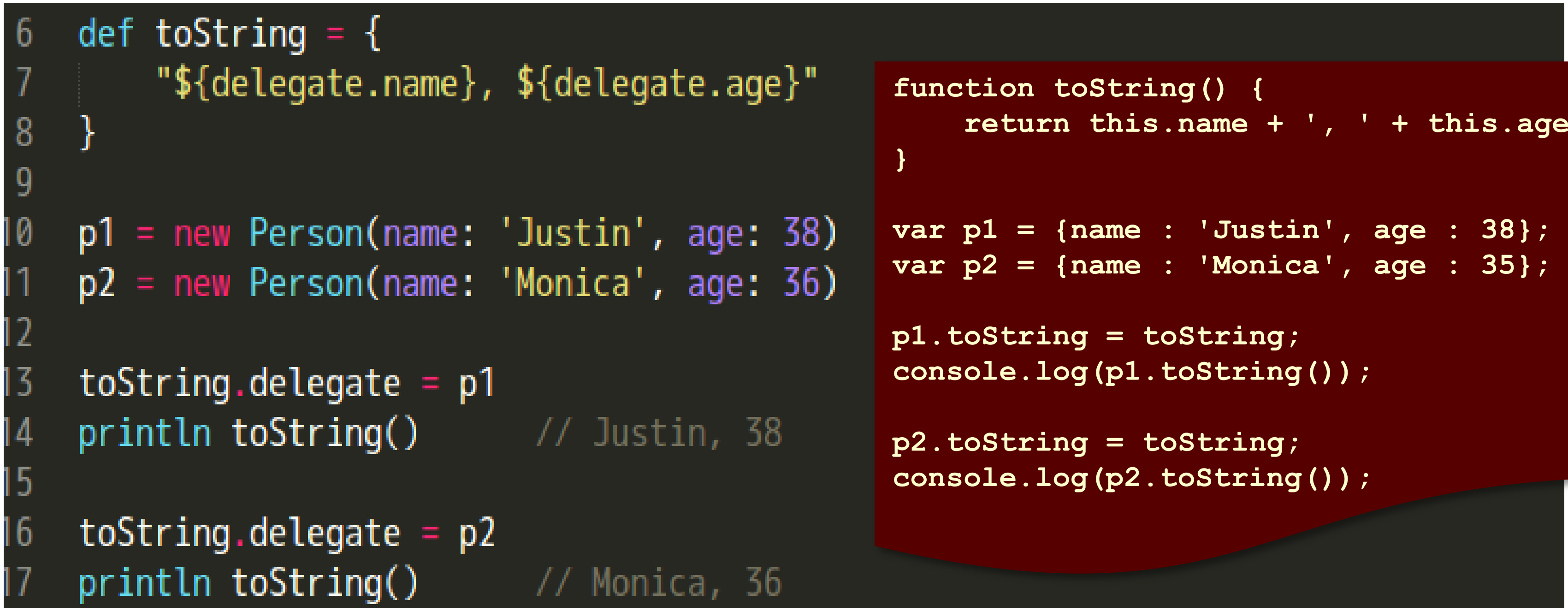
- 更強大一點 ...



```
groovy:000> 'abcdefg'.toUpperEach { print it }
ERROR groovy.lang.MissingMethodException:
No signature of method: java.lang.String.toUpperEach() is applicable for argumen
t types: (groovysh_evaluate$_run_closure1) values: [groovysh_evaluate$_run_closu
re1@1c4c648]
Possible solutions: toUpperCase(), toUpperCase(java.util.Locale)
        at groovysh_evaluate.run (groovysh_evaluate:2)
        ...
groovy:000> String.metaClass.toUpperEach { action ->
groovy:001>     delegate.each { action it.toUpperCase() }
groovy:002> }
===> null
groovy:000> 'abcdefg'.toUpperEach { print it }
ABCDEFG===> abcdefg
```

```
String.prototype.toUpperEach = function(action) {
    for(var i = 0; i < this.length; i++) {
        action(this.charAt(i).toUpperCase());
    }
};

'abcdef'.toUpperEach(function(ch) {
    console.log(ch);
});
```

- 強大、強大再強大 ... XD



```
def toString = {
    "${delegate.name}, ${delegate.age}"
}

p1 = new Person(name: 'Justin', age: 38)
p2 = new Person(name: 'Monica', age: 36)

toString.delegate = p1
println toString()       // Justin, 38

toString.delegate = p2
println toString()       // Monica, 36
```

```
function toString() {
    return this.name + ', ' + this.age;
}

var p1 = {name : 'Justin', age : 38};
var p2 = {name : 'Monica', age : 35};

p1.toString = toString;
console.log(p1.toString());

p2.toString = toString;
console.log(p2.toString());
```

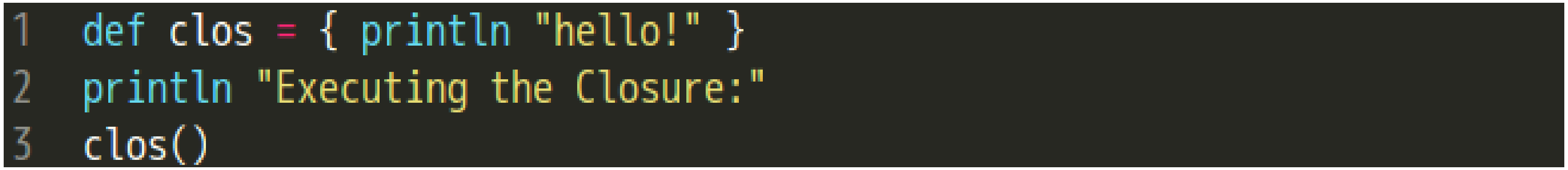
使用 Groovy 建立 DSL

```
<memo>
   <to>Nirav Assar</to>
  <from>Barack Obama</from>
  <body>How are things? We are doing well
  <idea>The economy is key</idea>
  <request>Please vote for me</
</memo>
```

```
MemoDsl.make {
   to "Nirav Assar"
   from "Barack Obama"
   body "How are things? We are doing well. Take care"
   idea "The economy is key"
   request "Please vote for me"
   xml
}
```

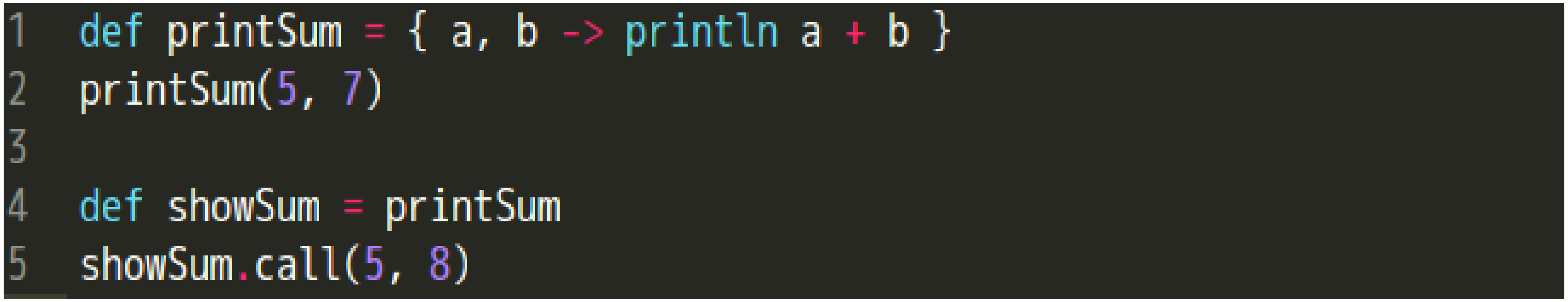
Groovy 的 Closure

- Closure？



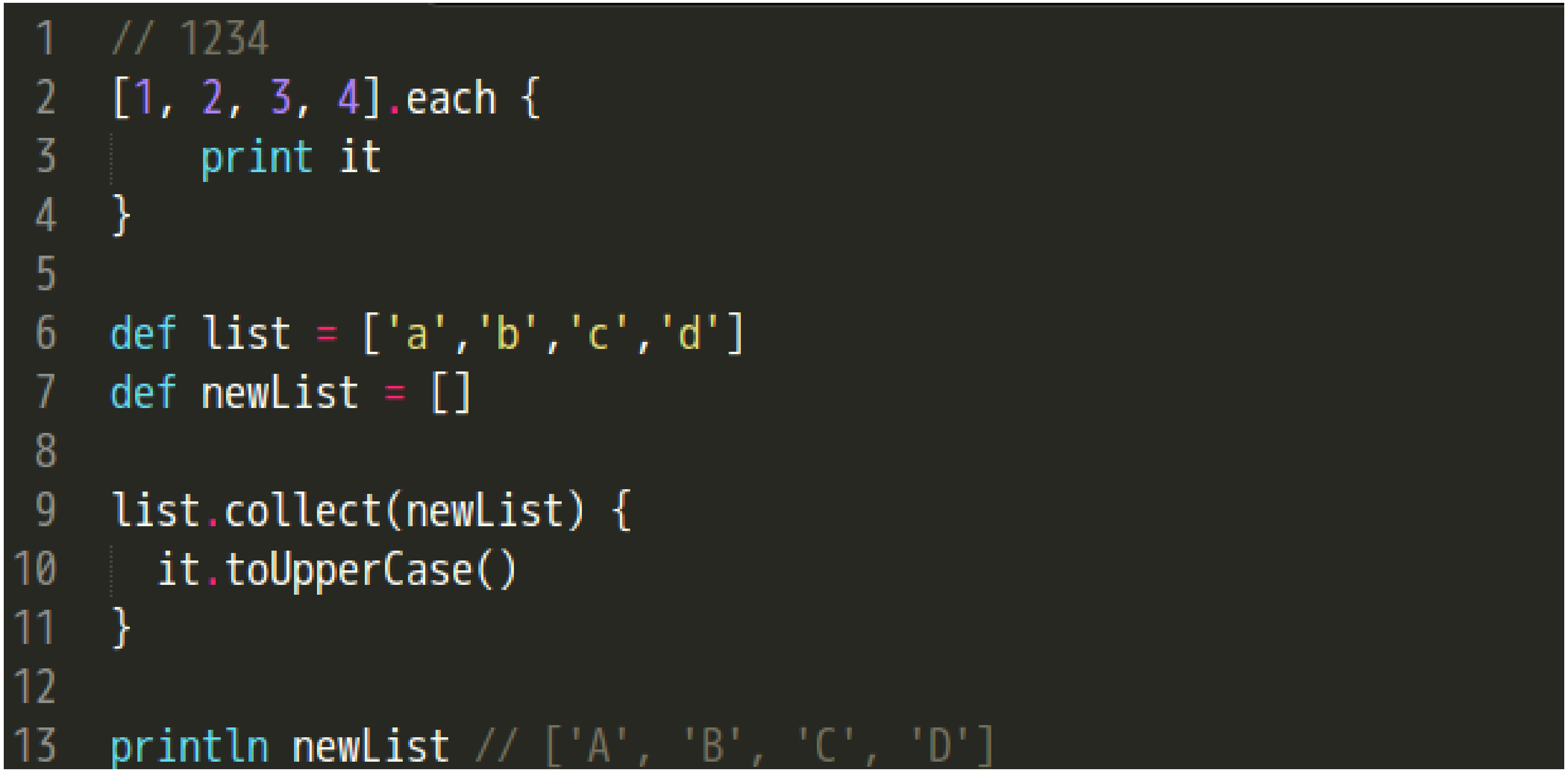
```groovy
def clos = { println "hello!" }
println "Executing the Closure:"
clos()
```

- Lambda？匿名函式？一級函式？



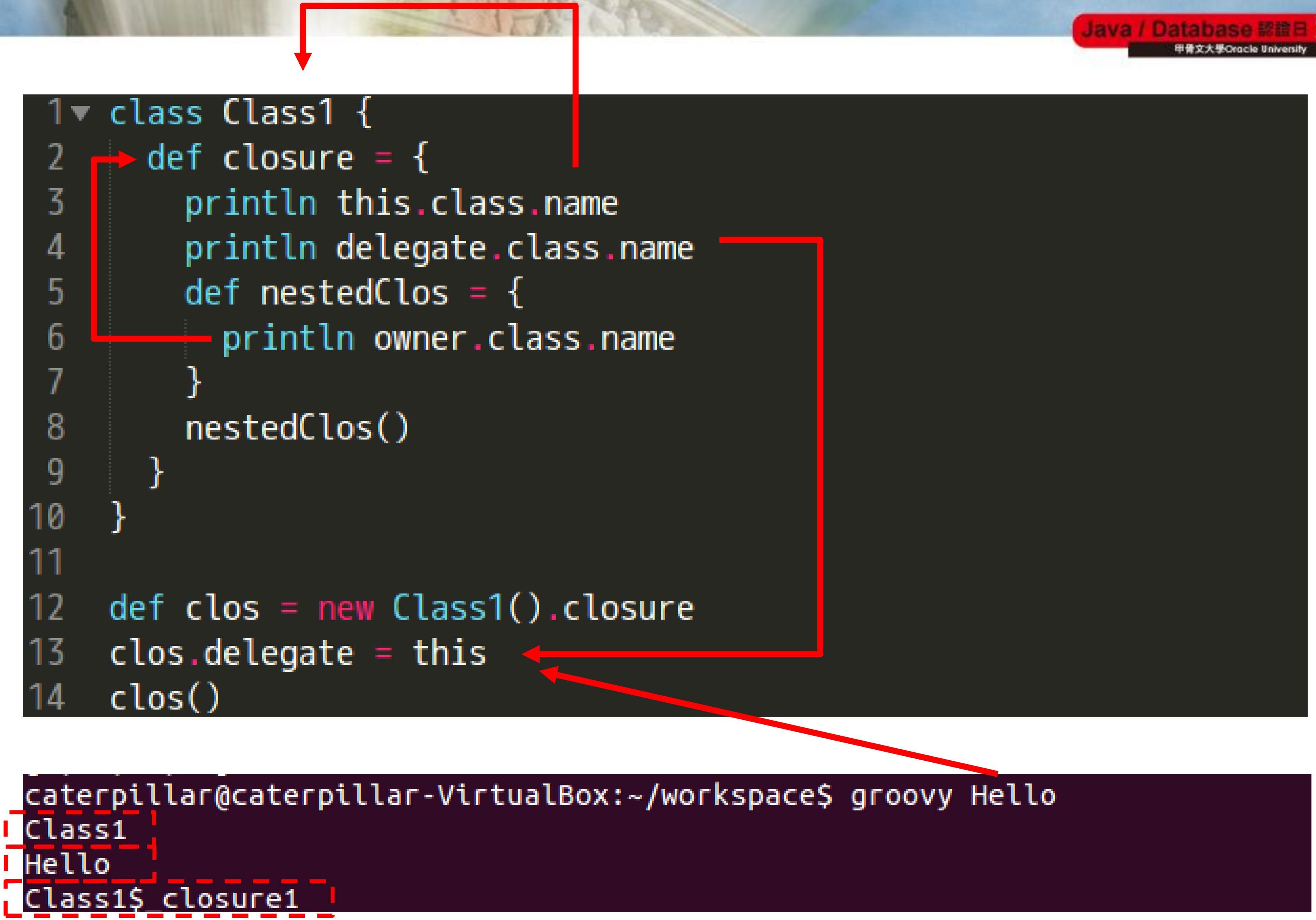
```groovy
def printSum = { a, b -> println a + b }
printSum(5, 7)

def showSum = printSum
showSum.call(5, 8)
```

- Closure 應用



```
// 1234
[1, 2, 3, 4].each {
    print it
}

def list = ['a','b','c','d']
def newList = []

list.collect(newList) {
  it.toUpperCase()
}

println newList // ['A', 'B', 'C', 'D']
```

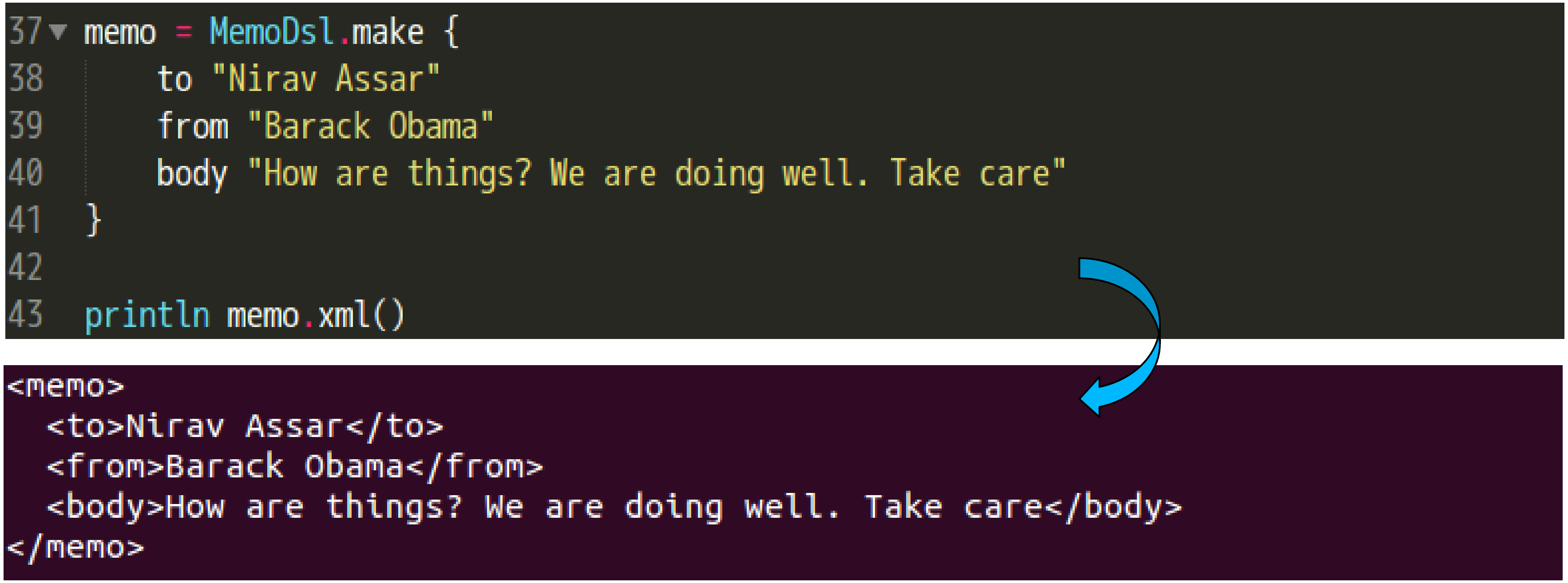
- **this** - 像 Java 的 this，參考至定義 Closure 的類別之實例。
- **owner** - 包圍 Closure 之物件，也許是 this 參考之物件，也許是包圍 Closure 之 Closure。
- **delegate** - 預設是 owner，可以變更指定。



```groovy
class Class1 {
  def closure = {
    println this.class.name
    println delegate.class.name
    def nestedClos = {
      println owner.class.name
    }
    nestedClos()
  }
}

def clos = new Class1().closure
clos.delegate = this
clos()
```

```
caterpillar@caterpillar-VirtualBox:~/workspace$ groovy Hello
Class1
Hello
Class1$_closure1
```

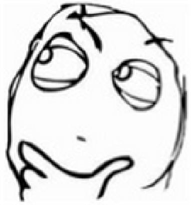
- DSL – Domain Specific Language

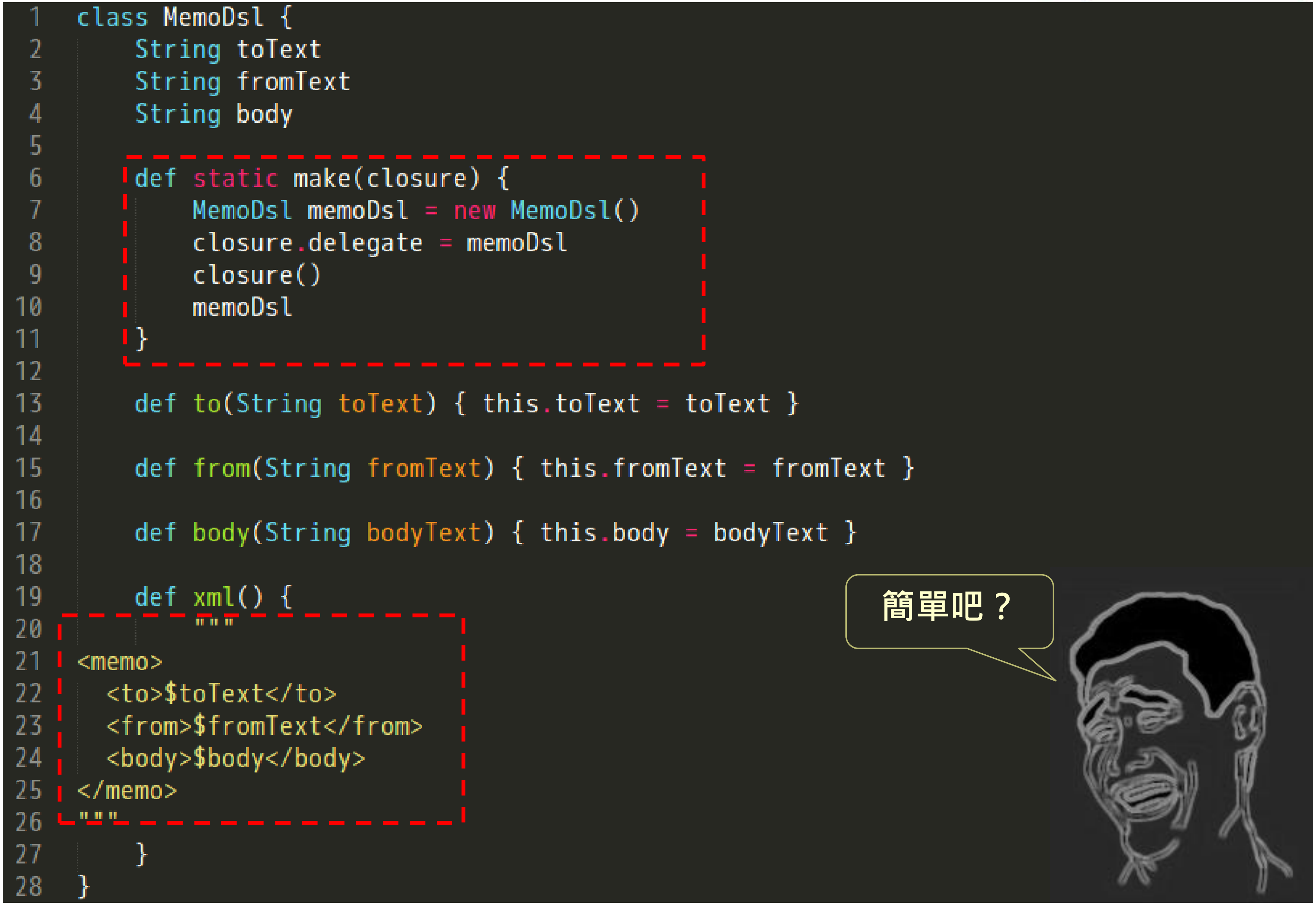


```
memo = MemoDsl.make {
    to "Nirav Assar"
    from "Barack Obama"
    body "How are things? We are doing well. Take care"
}

println memo.xml()
```

```
<memo>
  <to>Nirav Assar</to>
  <from>Barack Obama</from>
  <body>How are things? We are doing well. Take care</body>
</memo>
```

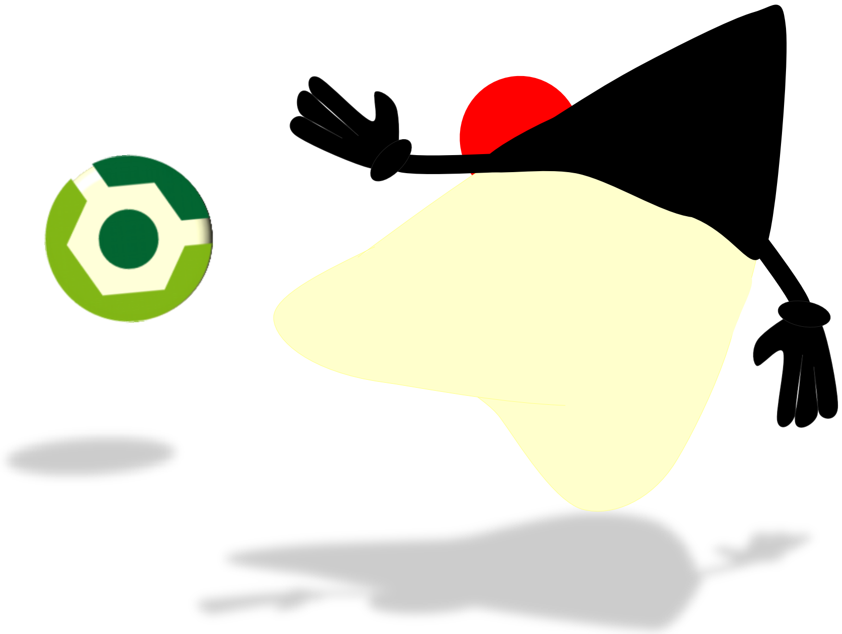
- 構思 ...傳入 Closure、會有 make、to、from、body、xml 等方法 ...





```
class MemoDsl {
    String toText
    String fromText
    String body

    def static make(closure) {
        MemoDsl memoDsl = new MemoDsl()
        closure.delegate = memoDsl
        closure()
        memoDsl
    }

    def to(String toText) { this.toText = toText }

    def from(String fromText) { this.fromText = fromText }

    def body(String bodyText) { this.body = bodyText }

    def xml() {
        """
<memo>
  <to>$toText</to>
  <from>$fromText</from>
  <body>$body</body>
</memo>
"""
    }
}
```

自動化工具 Gradle



建構工具元老

- 一組 XML 標籤
  - 專案（Project）
  - 目標（Target）
  - 任務（Task）

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="helloworld" default="build">
    <property name="src.dir" value="src" />
    <property name="lib.dir" value="libs" />
    <property name="resource.dir" value="resources" />
    <property name="build.dir" value="build" />
    <property name="dist.dir" value="dist" />
    <!-- 刪除 build.dir 與 dist.dir -->
    <target name="clean">
        <delete dir="${build.dir}" />
        <delete dir="${dist.dir}" />
    </target>
    <!-- 建立 build.dir 與 dist.dir 與複製相關設定檔 -->
    <target name="prepare" depends="clean">
        <mkdir dir="${build.dir}" />
        <mkdir dir="${dist.dir}" />
        <mkdir dir="${build.dir}/libs" />
        <mkdir dir="${build.dir}/all" />
    </target>

...
```

- Ant 的經驗累積
  - 自動建構的標準化
  - 相依管理的需求（以 Ant Ivy 補足）

maven 專案管理工具

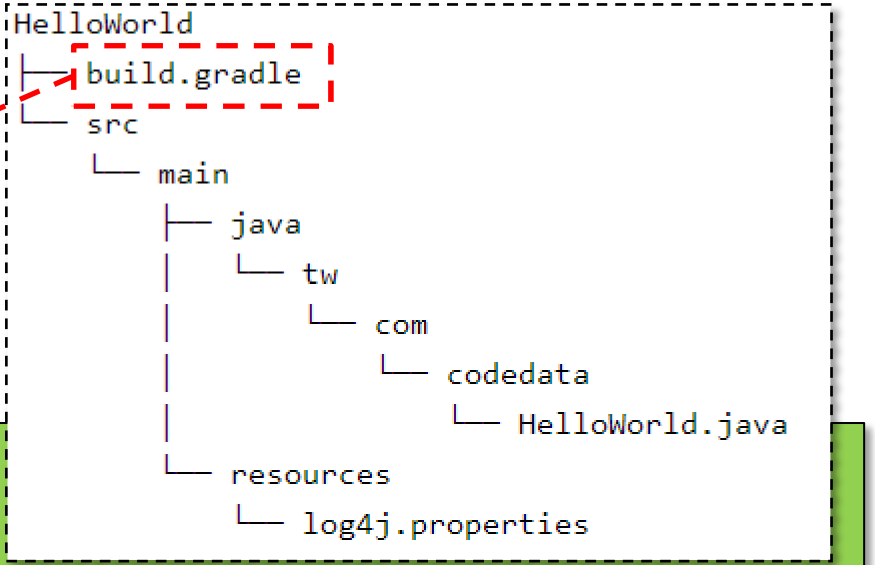
- 一組 XML 標籤
  - Convention Over Configuration
  - Archetype plugin
  - 相依性宣告
- Maven 的經驗累積
  - 相依性管理的濫用
  - 修改預設行為的難度

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.
0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>tw.com.codedata</groupId>
  <artifactId>helloworld</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>helloworld</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

企業自動化工具

- 保持 Ant 的自由度、低於 Maven 學習門檻
  - 有著 Lifecycle 的精神
  - 相依性管理機制的便利
  - DSL 的可讀性與容易撰寫
  - 引用 plugin 獲得需要的編譯功能
- 許多開放原始碼專案與工具採用 Gradle
  - Hibernate 從 Maven 改用 Gradle ...
  - Spring 只提供 Maven 或 Gradle 兩種下載方式
  - Android Studio 使用 Gradle 建構專案

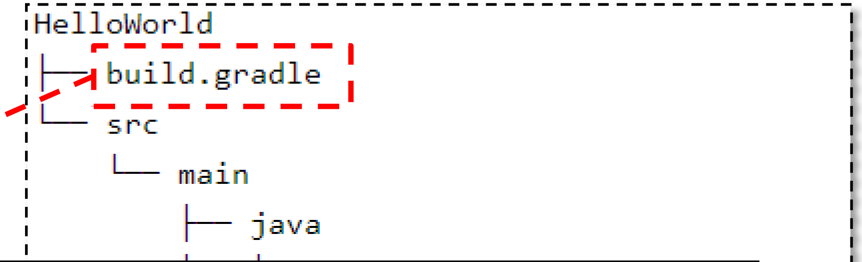
第一個 Gradle 專案



```
HelloWorld
├── build.gradle
└── src
    └── main
        ├── java
        │   └── tw
        │       └── com
        │           └── codedata
        │               └── HelloWorld.java
        └── resources
            └── log4j.properties
```

```
apply plugin: 'java'

apply plugin:'application'

mainClassName = "tw.com.codedata.HelloWorld"

repositories {
    mavenCentral()
}

dependencies {
    compile group: 'commons-logging', name: 'commons-logging', version: '1.1.1'
    compile group: 'log4j', name: 'log4j', version: '1.2.16'
}
```

第一個 Gradle 專案



```
HelloWorld
├── build.gradle
└── src
    └── main
        ├── java
```

```
appl

appl

main

repo

}

dependencies {
    compile group: 'commons-logging', name: 'commons-logging', version: '1.1.1'
    compile group: 'log4j', name: 'log4j', version: '1.2.16'
}
```

```
$ gradle run
:compileJava
:processResources
:classes
:run
 INFO [main] (HelloWorld.java:11) - Hello World

BUILD SUCCESSFUL

Total time: 6.027 secs
```

Every configuration file becomes a programming language…

James Gosling

Hello World

- build.gradle

```
task hello {
     doLast {
          println 'Hello world!'
     }
}
```

Closure

Closure

- 它是一個 Groovy ...

```
$ gradle -q hello
Hello world!
```

```
task upper << {
    String someString = 'mY_nAmE'
    println "Original: " + someString
    println "Upper case: " + someString.toUpperCase()
}
```

```
task count << {
    4.times { print "$it " }
}
```

```
4.times { counter ->
    task "task$counter" << {
        println "I'm task number $counter"
    }
}
```

```
$ gradle -q task1
I'm task number 1
```

- 我不只是設定檔 ... XD

```
task checksum << {
    fileList('../antLoadfileResources').each {File file ->
        ant.checksum(file: file, property: "cs_$file.name")
        println "$file.name Checksum: ${ant.properties["cs_$file.name"]}"
    }
}

task loadfile << {
    fileList('../antLoadfileResources').each {File file ->
        ant.loadfile(srcFile: file, property: file.name)
        println "I'm fond of $file.name"
    }
}

File[] fileList(String dir) {
    file(dir).listFiles({file -> file.isFile() } as FileFilter).sort()
}
```

- 不用 Groovy 太多知識，先知道二件事 …
  - Groovy DSL 大量運用 Closure 機制
  - Gradle 的實作大量使用 Delegation 模式

Gradle 與 IDE

Gradle 建立基本專案目錄？

- build.gradle 可視為 Groovy 原始碼檔案

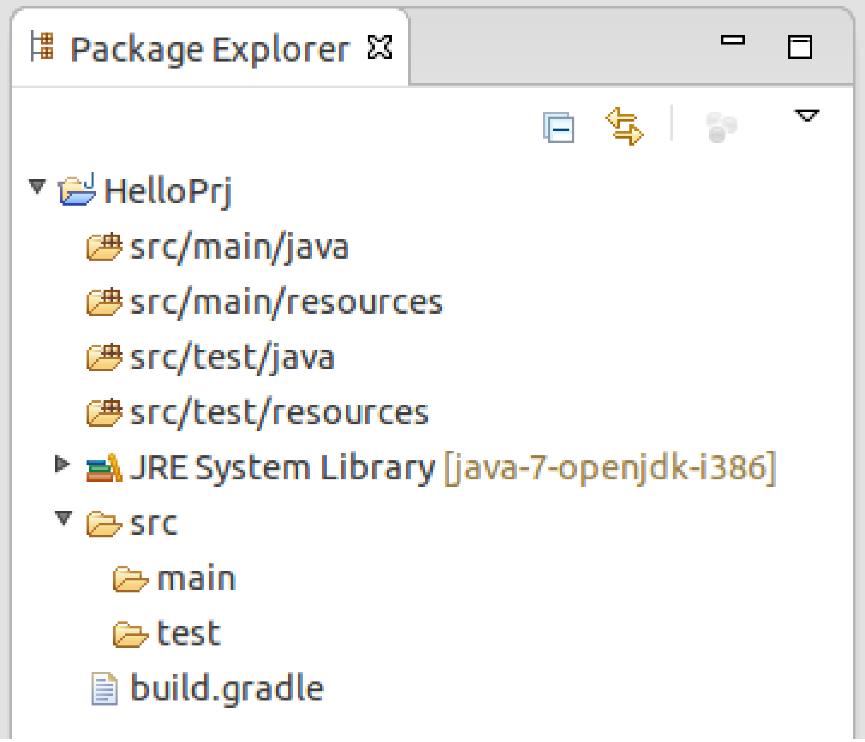
```
apply plugin: 'java'
task "create-dirs" << {
    sourceSets*.java.srcDirs*.each { it.mkdirs() }
    sourceSets*.resources.srcDirs*.each { it.mkdirs() }
}
```

- 使用 gradle-templates
  - https://github.com/townsfolk/gradle-templates
  - `gradle createJavaProject`
  - `gradle createWebappProject`
  - `gradle createScalaProject`
  - `...`

Eclipse

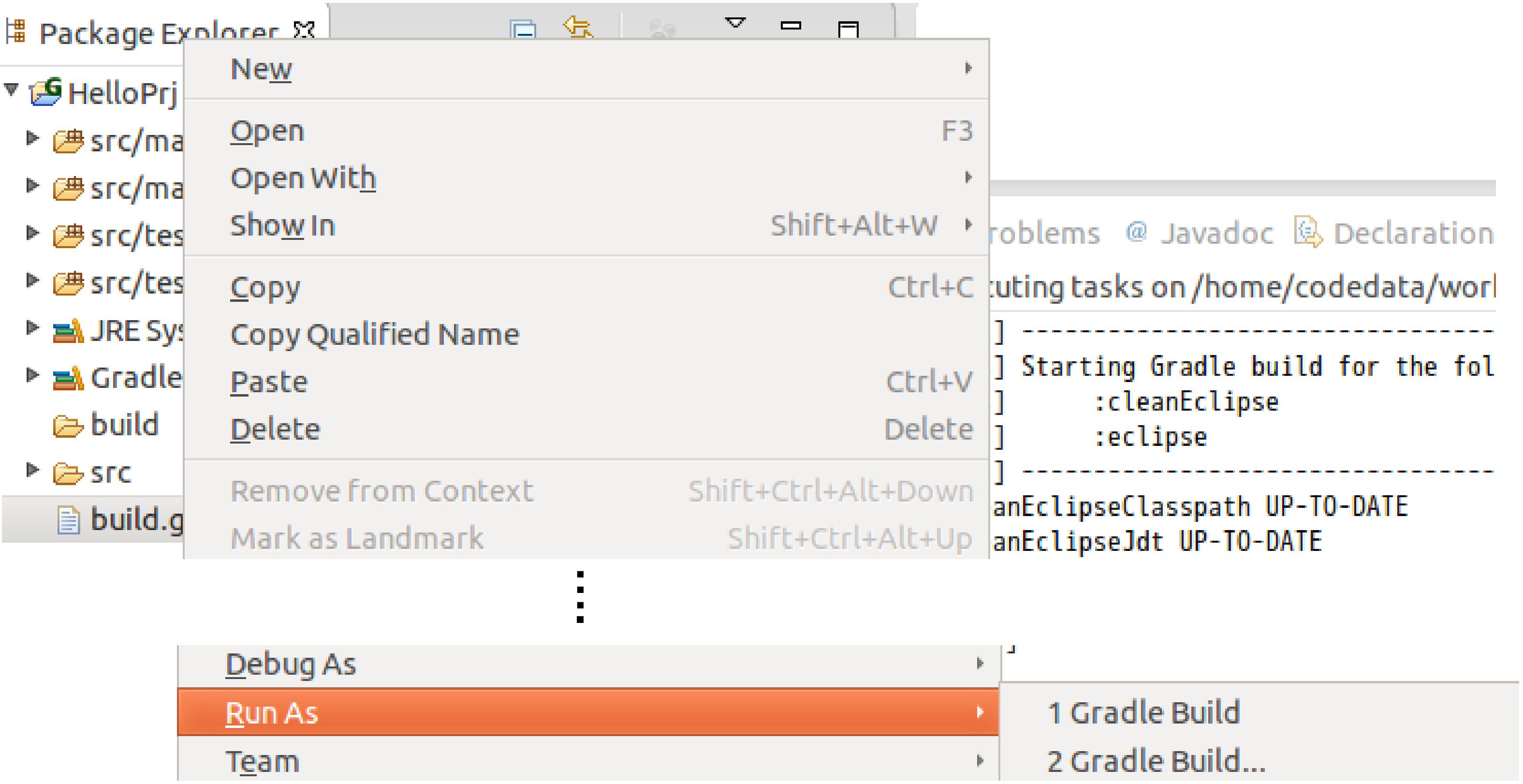
- build.gradle

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



```
HelloPrj
  src/main/java
  src/main/resources
  src/test/java
  src/test/resources
  JRE System Library [java-7-openjdk-i386]
  src
    main
    test
  build.gradle
```

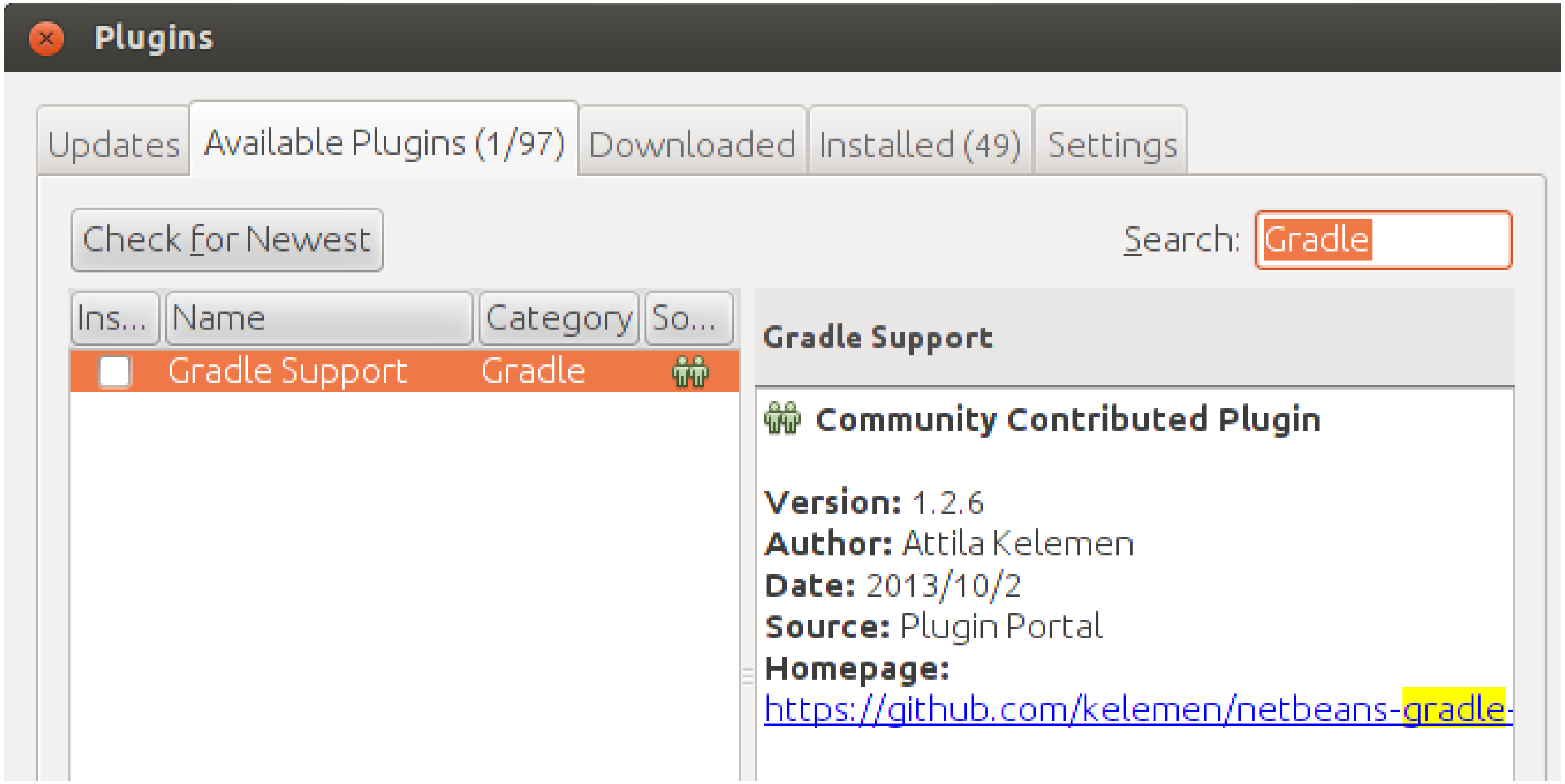
- 安裝 eclipse-integration-gradle
  - https://github.com/spring-projects/eclipse-integration-gradle/



```
uting tasks on /home/codedata/worl
] ---------------------------------
] Starting Gradle build for the fol
]      :cleanEclipse
]      :eclipse
] ---------------------------------
anEclipseClasspath UP-TO-DATE
anEclipseJdt UP-TO-DATE
```

NetBeans

- Gradle Plugins
  - http://plugins.netbeans.org/plugin/44510/gradle-support



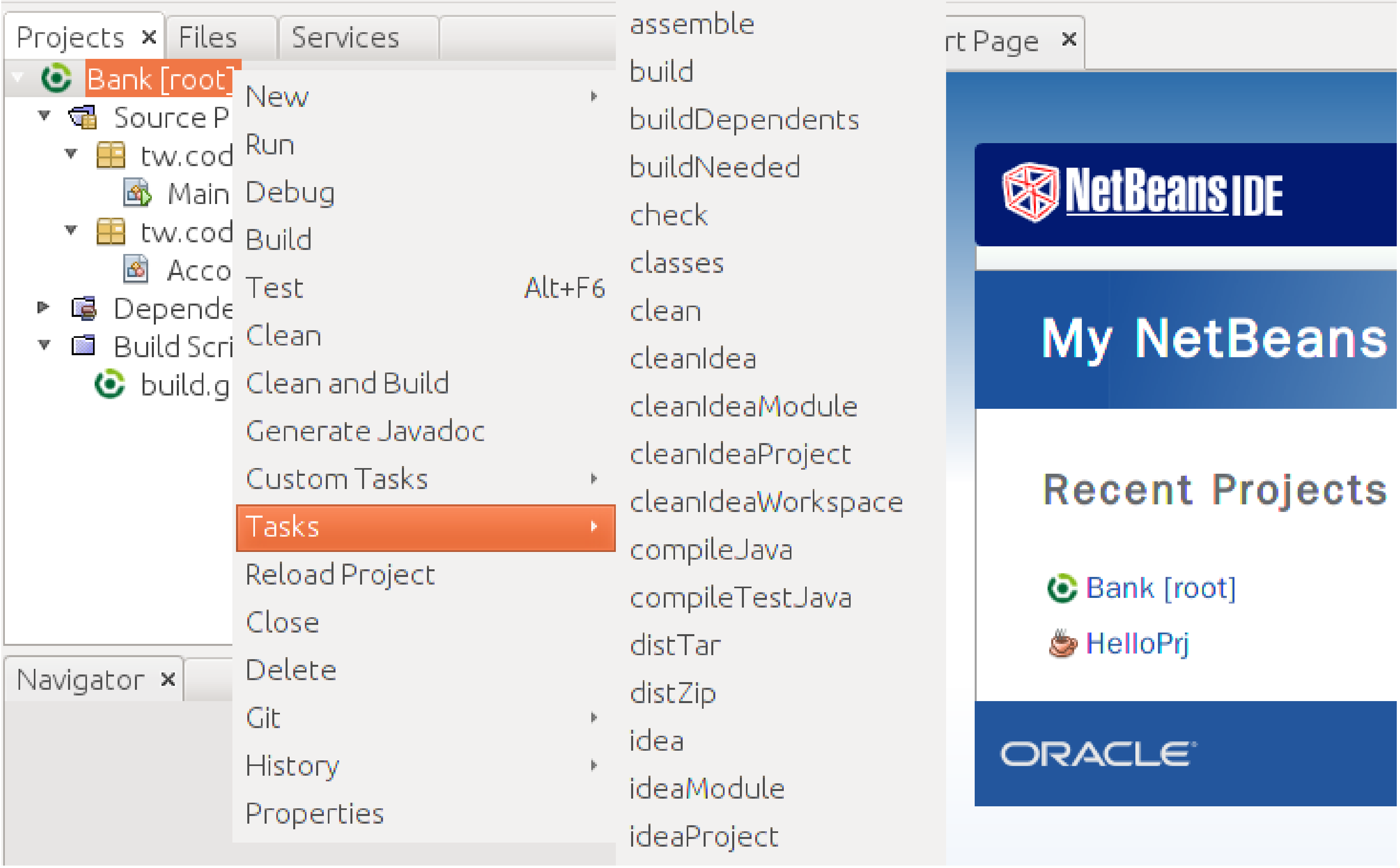
Search: Gradle

Ins...	Name	Category	So...
□	Gradle Support	Gradle	

Gradle Support

Community Contributed Plugin

Version: 1.2.6
Author: Attila Kelemen
Date: 2013/10/2
Source: Plugin Portal
Homepage:
https://github.com/kelemen/netbeans-gradle-


Projects ×
Files
Services
Bank [root]
Source P
tw.cod
Main
tw.cod
Acco
Depende
Build Scri
build.g
Navigator ×
New
Run
Debug
Build
Test Alt+F6
Clean
Clean and Build
Generate Javadoc
Custom Tasks
Tasks
Reload Project
Close
Delete
Git
History
Properties
assemble
build
buildDependents
buildNeeded
check
classes
clean
cleanIdea
cleanIdeaModule
cleanIdeaProject
cleanIdeaWorkspace
compileJava
compileTestJava
distTar
distZip
idea
ideaModule
ideaProject
rt Page ×
NetBeans IDE
My NetBeans
Recent Projects
Bank [root]
HelloPrj
ORACLE

參考資料

- Groovy
  - http://groovy.codehaus.org/Documentation
  - http://www.oracle.com/technetwork/articles/java/groovy-1695411.html
  - http://java.dzone.com/articles/groovy-dsl-simple-example
- Gradle
  - http://www.gradle.org/documentation
  - http://www.codedata.com.tw/tag/gradle/
- Gradle 與 IntelliJ IDEA
  - http://blog.lyhdev.com/2013/11/intellij-idea-community-edition-free.html