gradle的groovy语法案例详解

**groovy的语法与python的语法有些类似**

**1.groovy的三个特性**

1）不需要分号

2）不需要写get，set方法，对象里面默认会有get与set方法

3）方法的最后一个值默认为返回值，不需要写return

示例：

1. public class ProjectVersion{
2. private int major *//此处省略了结尾的分号*
3. private int minor
5. public ProjectVersion(int major , int minor){
6. this.major = major
7. this.minor = minor
8. }
10. int getMajor() {
11. *//此处省略了return*
12. major
13. }
15. void setMajor(int major) {
16. this.major = major
17. }
18. *//此处省略了minor的get与set方法*
20. }
22. *//此处省略了方法的括号*
23. print 111

**2.groovy高效特性**

1）可选的类型定义

2）assert断言

3）字符串的定义写法

4）集合api

5）闭包

示例：

1. 1.可选的类型定义
3. def version = 1
5. *//2 assert*
7. assert version == 1
9. *//3 括号是可选的*
10. println version
12. *//4 字符串*
13. def s1 = 'cyjz'
14. def s2 = "gradle version is ${version}"
15. def s3 = '''my
16. name is
17. cyjz'''
19. println s1
20. println s2
21. println s3
23. 特性5 集合api
24. *//list*
25. def buildTool = ['ant','maven']
26. buildTool << 'gradle'
27. assert buildTool.getClass() == ArrayList
28. println buildTool.size()
29. *//map*
30. def buildYears = ['ant':2000,'maven':2004]
31. println buildYears.ant
32. println buildYears['maven']
34. println buildYears.getClass()
35. *//linked hash map*
36. *//6.闭包*
37. def c1 = {
38. v ->
39. print v
41. }
42. def c2 = {
43. print 'hello'
44. }
46. def method1(Closure closure){
47. closure('param')
48. }
50. def method2(Closure closure){
51. closure()
52. }
53. method1(c1)
54. method2(c2)

**3.gradle构建脚本中的写法**

默认都有个project实例

示例：

1. *//构建脚本中默认都是有个Project实例的*
2. apply plugin:'java'
4. version = '0.1'
6. repositories{
7. mavenCentral()
8. }
10. dependencies{
11. compile 'commons-codec:commons-codec:1.6'
12. }