
link: null
title: 珠峰架构师成长计划
description: null
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=35 sentences=26, words=225

1. Groovy#

- Groovy是用于Java虚拟机的一种敏捷的动态语言，它是一种成熟的面向对象编程语言，既可以用于面向对象编程，又可以用作纯粹脚本语言。使用该种语言不必编写过多的代码，同时又具有闭包和动态语言中的其它特性
- Groovy完全 兼容Java语法

2. Groovy语法

2.1 字符串

- 可选类型定义
- 字符串可以用单引号，双引号，和三个单引号的表达方式

```
def name='zhufeng'
println 'hello $name'
println "hello $name"
println '''hello
$name'''
```

2.2 定义class

- 编译器给属性自动添加getter/setter方法
- 属性可以直接用点号来获取

```
class Person{
    String name
    int age
    Person(name,age){
        this.name = name
        this.age = age
    }
}
Person p = new Person('zhufeng',10)
println p.name
p.setAge(11)
println p.age
```

2.3 assert

- 自带assert语句
- 调用方法时括号是可选的

```
def version =1
assert version == 1
println(version)
```

2.4 集合API

2.4.1 List

```
def range = 0..4
println range.class
assert range instanceof List
```

2.4.2 ArrayList

```
def coll = ["Groovy", "Java", "Ruby"]
assert coll instanceof Collection
assert coll instanceof ArrayList
assert coll.size() == 3
assert coll.getClass() == ArrayList
```

添加元素

```
coll.add("Python")
coll << "Smalltalk"
coll[5] = "Perl"
```

查找元素

```
assert coll[1] == "Java"
```

```
def maps = [name:'zhufeng',age:10]
maps.home = 'beijing'
assert maps.size() == 3
assert maps.getClass() == LinkedHashMap
```

2.5 循环

```
for(i = 0; i < 5; i++){
    println i
}
```

```
for(i in 0..5){
    println i
}
```

2.6 函数

2.6.1 定义一个函数

最后一行的为返回值 不需要用return

```
def stage() {  
}
```

2.6.2 参数类型

```
String function(arg1,args2){  
}  
def nonReturnTypeFunc(){  
    last_line  
}  
  
String getString(){  
    return "I am a string"  
}
```

2.7 闭包

- 闭包是一种数据类型，它代表了一段可执行的代码
- 闭包是一个类型为 groovy.lang.Closure的代码块
- 闭包可以赋值给变量，作为参数传递给方法，并且像普通方法一样来调用
- 闭包可以访问上下文的变量，函数不可以

```
def xxx = {parameters -> code}  
def xxx = {无参数，纯code} 这种case不需要->符号  
,
```

2.7.1 闭包格式

- 闭包[closureParameters->] 是可选的，参数的类型也是可选的
- 如果我们不指定参数的类型，会由编译器自动推断。如果闭包只有一个参数，这个参数可以省略，我们可以直接使用it来访问该参数

```
{  
    [closureParameters -> ]  
    statements  
}
```

```
def it1 = { it -> println it }  
def it2 = { name -> println name }  
def it3 = { String x, int y -> println "${x} 's value is ${y}" }  
  
it1('aaa')  
it2('bbb')  
it3('ccc',10)
```

2.7.2 闭包返回值

- 闭包总会返回一个值，返回值是闭包的最后一条语句的值(如果没有显式的return语句)

```
def name2 = 'zhufeng'  
def greeting = {  
    'hello ' + name2  
}  
  
println greeting()
```

2.7.3 闭包作为方法参数

```
Integer increment(Closure closure, Integer count){  
    closure() + count  
}  
  
assert increment({ 10 }, 2) == 12
```

3. Groovy实例

```
def stage(name,closure){  
    name+closure()  
}  
  
def r = stage('Preparation'){  
    5  
}  
println r
```