

Gradle 入门

Gradle是一个基于Apache Ant和Apache Maven概念的项目自动化构建开源工具。它使用一种基于Groovy的特定领域语言(DSL)来声明项目设置，目前也增加了基于Kotlin语言的kotlin-based DSL，抛弃了基于XML的各种繁琐配置。

面向Java应用为主。当前其支持的语言限于Java、[Groovy](#)、[Kotlin](#)和[Scala](#)，计划未来将支持更多的语言。

Gradle 安装 (Windows)

安装 Gradle 前，需要本机安装了 Java8 环境才能支持

Gradle 安装地址：<https://services.gradle.org/distributions/>

Groovy 语言

Groovy 语法

```
// 输出语句:不需要带分号结尾,不需要添加大括号
println "hello groovy"

// 定义变量
def i = 18 // 定义一个 int 类型的变量 i
println i // 输出 18

def name = "zs" // 定义一个 String 类型的变量 name
println name // 输出 zs

// 定义 list 集合
def list = ['a', 'b']

// 向 list 集合中添加元素
list << 'c'

// 获取添加的元素
println list.get(2) // 通过 list 集合的下标获取,打印结果为 c

// 定义 map 集合
def map = ['name': 'zs', 'age': 20]

// 向 map 集合中添加元素 (sex 不存在直接添加 key,然后直接为其赋值)
map.sex = '男'

// 获取 map 中刚刚添加的元素
println map.get("sex") // 通过 map 集合的 key 获取 value,打印结果为 男
```

Groovy 闭包

```
// groovy 中的闭包
// 什么是闭包，闭包就是一段代码块，在 gradle 中，我们需要把闭包当作参数来使用
// 定义一个闭包
def b1 = {
    println "hello b1"
}

// 定义个方法，方法里面需要闭包的参数类型
def run(Closure closure) {
    closure()
}

// 调用 run 方法
run(b1) // 打印结果 hello b1

// 定义一段带参数的闭包
def b2 = {
    name ->
        println "hello ${name}"
}

def run2(Closure closure, String name) {
    closure(name)
}

run2(b2, "小明") // 打印结果 hello 小明
```

Gradle 配置文件介绍

```
group `com.itheima` // 这行内容是项目的目录结构
version `1.0-SNAPSHOT` // 表示项目的版本号

apply plugin: `java` // 基于 java 环境运行

sourceCompatibility = 1.8 // java 环境的版本号

// 指定所使用的仓库
// mavenCentral() 表示使用中央仓库，此刻项目中所需要的 jar 包都会默认从中央仓库下载到本地指定目录
repositories {
    mavenCentral()
}

// gradle 工程所有 jar 包的坐标都在 dependencies 属性内
// 每一个 jar 包的坐标都有三个基本元素组成
// group name version
// testCompile 表示该 jar 包在测试的时候起作用，该属性为 jar 包的作用域
// 我们在 gradle 里面添加坐标的时候都要带上 jar 包的作用域
dependencies {
    testCompile group: `junit`, name: `junit`, version: `4.12`
}
```