

Gradle

简介

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。

下载地址

<https://services.gradle.org/distributions/>

配置环境变量

```
GRADLE_HOME=D:\软件\gradle\gradle-6.0.1
path=%GRADLE_HOME%\bin
```

Gradle 也可以编程

idea Tools-->Groovy Console .. 前提是有Gradle项目

变量声明

案例演示

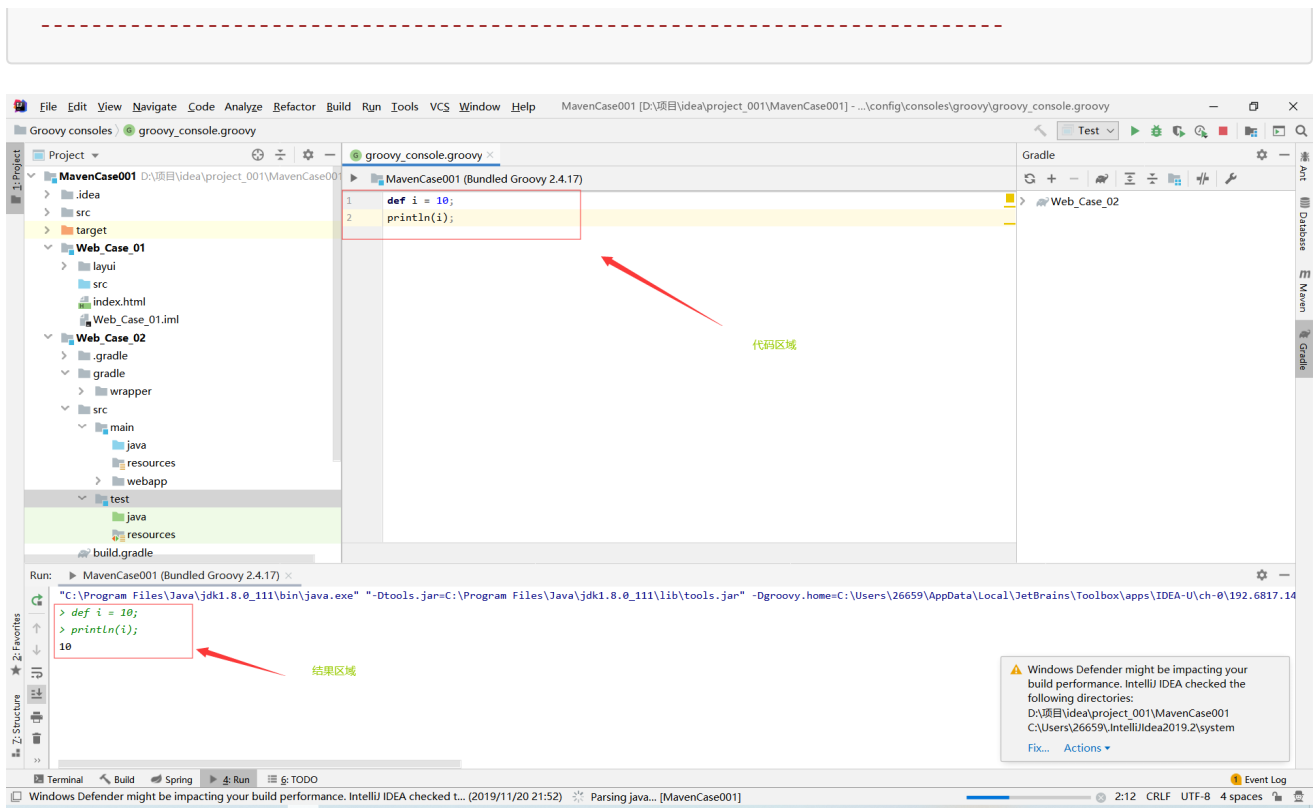
```
//声明变量
def i = 10;
//打印变量
println(i);

-----

//声明list集合
def myList = ["张三", "李四"];
//向list添加数据
myList << "王五";
//获取list中王五的数据
println(myList.get(2));

-----

//声明map集合
def myMap = ["1": "张三", "2": "李四", "3": "王五"];
//向map添加数据 变量.键 = 值 /键值就添加进去了
myMap."6" = "唐唐唐";
//获取map中的唐唐唐数据
myMap.get("6");
```



闭包创建

无参闭包

```
//闭包就是一段代码块
def b1 = {
    println("你好，闭包！");
}
//声明方法 参数类型为 闭包类型 当中直接调用传入的闭包
def fun(Closure closure){
    closure();
}
//调用fun方法，传入闭包
fun(b1);
//结果 你好，闭包！
```

有参闭包

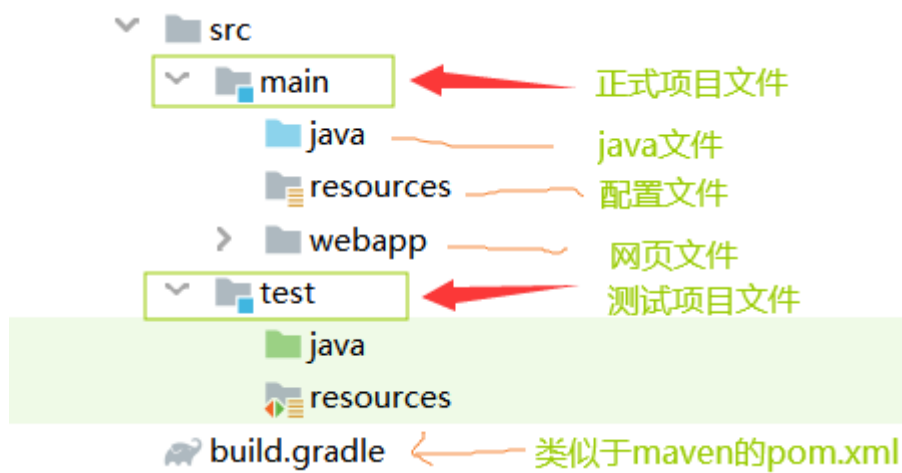
```

def b1 = {
    v -> println("你好鸭!${v}"); // ${v}获取传入v的参数值
}
//声明方法 参数类型为 闭包类型 当中直接调用传入的闭包
def fun(Closure closure, def name){
    closure(name);
}
//调用fun方法, 传入闭包
fun(b1, "张三");
//结果 你好, 闭包!

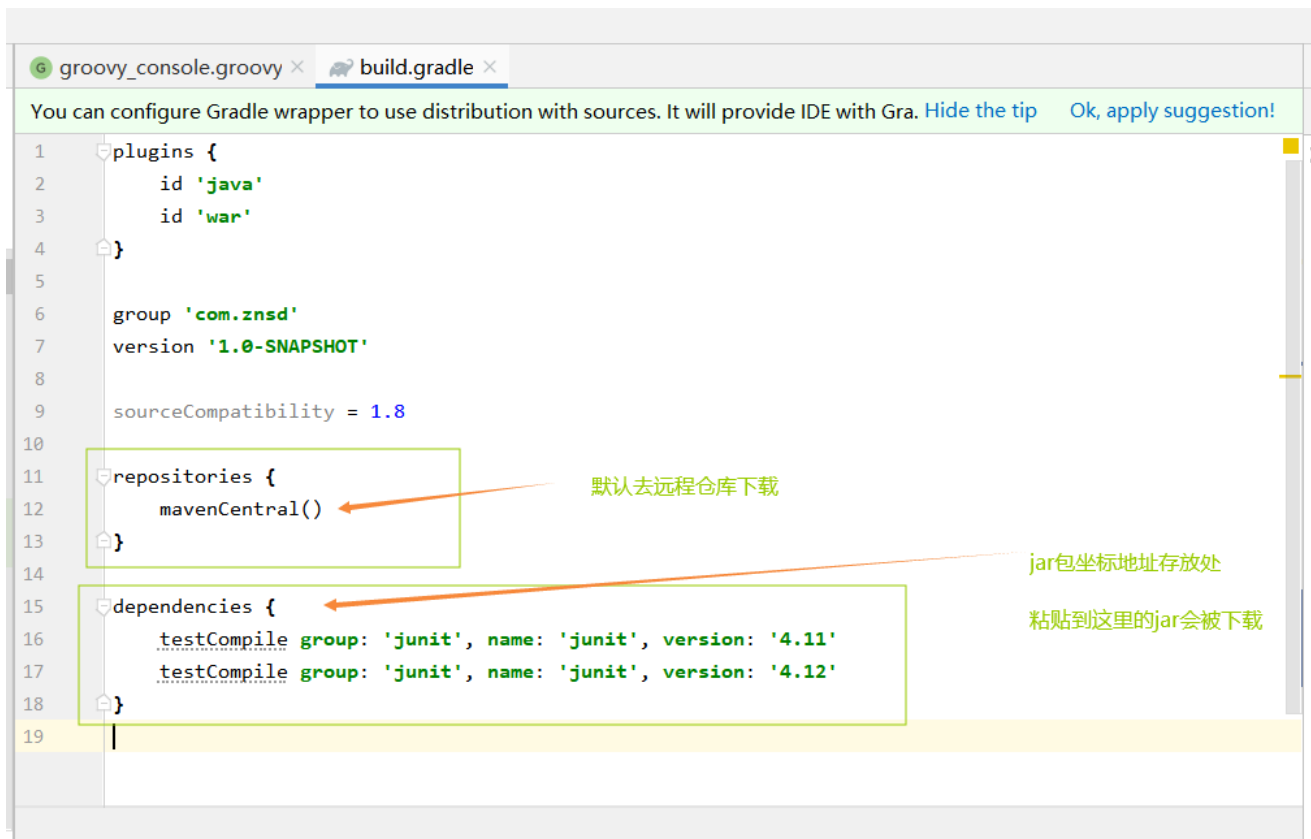
```

Gradle 下载jar包

Gradle目录介绍



Build目录介绍



在maven官网中复制想要jar的坐标

maven官网地址: <https://mvnrepository.com/>

坐标位置粘贴在build.gradle 的dependencies 中 不需要分号

```
dependencies {
    testCompile group: 'junit', name: 'junit', version: '4.11'
    compile group: 'org.springframework', name: 'spring-context', version: '5.2.0.RELEASE'
}
```

当复制后会到maven的远程仓库中下载到本地.

使用maven仓库

当使用gradle的时候，导入坐标时若想使用本地maven仓库内下载好的jar，并且想把jar下载到maven时就需要配置一个环境变量

```
GRADLE_USER_HOME=MAVEN本地仓库位置
```

配置完环境变量后，在xml中需要再添加一个方法

```
repositories {  
    mavenLocal() //新增的方法，当出现jar时首先去本地找若找到则返回否则执行下面的方法  
    mavenCentral()  
}
```

若idea 中gradle的地址未发生改变则手动替换为maven本地仓库地址

