基本注解：

* @interface
* 注解可以没有主体的方法的形式和可选的默认值来定义成员

**字符串注解**

@interface Simple {

String str1() default "HelloWorld";

}

**枚举注解**

enum DayOfWeek { mon, tue, wed, thu, fri, sat, sun }

@interface Scheduled {

DayOfWeek dayOfWeek()

}

**类注解**

@interface Simple {}

@Simple

class User {

String username

int age

}

def user = new User(username: "Joe",age:1);

println(user.age);

println(user.username);

**注解成员值**

使用注解时，需要设置没有默认值的成员值。

@interface Example {

int status()

}

@Example(status = 1)

**关闭注解参数**

可以使用闭包作为注解值。这样，注解可以与各种表达式一起使用。

@interface OnlyIf {

Class value()

}

@OnlyIf({ number<=6 })

void Version6() {

result << 'Number greater than 6'

}

@OnlyIf({ number>=6 })

void Version7() {

result << 'Number greater than 6'

}

**元注解**

一个方法可以有多个注解，此时可以定义一个元注解，将多个注解集中在一起,用AnnotationCollector来定义注释的集合：

import groovy.transform.AnnotationCollector

@Procedure

@Master

@AnnotationCollector

@MasterProcedure

class MyMasterProcedure {}