

| JPoint | 说明 | Pointcut语法（详见1-2 Signature介绍） |
|-----------------------|--------------|---|
| method call | 函数被调用 | call(MethodSignature) |
| method execution | 函数执行内部 | execution(MethodSignature) |
| constructor call | 构造函数被调用 | call(ConstructorSignature) |
| constructor execution | 构造函数执行内部 | execution(ConstructorSignature) |
| field get | 读“变量” | get(FieldSignature) |
| field set | 写“变量” | set(FieldSignature) |
| pre-initialization | 与构造函数有关，很少用到 | preinitialization(ConstructorSignature) |
| initialization | 与构造函数有关，很少用到 | initialization(ConstructorSignature) |
| static initialization | static块初始化 | staticinitialization(TypeSignature) |
| handler | 异常处理 | handler(TypeSignature) 注： 只能与@Before()配合使用，不支持@After、@Around等 |
| advice execution | advice执行 | adviceexecution() |

1-1

| Signature | 语法（间隔一个空格），详见 1-3 Signature语法明细 |
|----------------------|---|
| MethodSignature | @注解 访问权限 返回值类型 类名.函数名(参数) |
| ConstructorSignature | @注解 访问权限 类名.new(参数) |
| FieldSignature | @注解 访问权限 变量类型 类名.成员变量名 |
| TypeSignature | 类名，可以使用的通配符，包括*和..以及+号。其中*号用于匹配除.号之外的任意字符，而..则表示任意子package，+号表示子类。 示例： <ol style="list-style-type: none">staticinitialization(Student)：表示Student类的static block；handler(NullPointerException)：表示catch到NullPointerException，因为是编译时匹配，所以如果抛出的是Exception，是无法相匹配的。 |

1-2

| Signature语法明细 | 解释 |
|--|---|
| @注解 | 完整路径@android.aspectjdemo.MPermisson，如果没有则不写。 |
| 访问权限 | public/private/protect，以及static/final，如果没有则不写。 注： <ol style="list-style-type: none">public final或public static，则方法必须是public final或public static；只写public，则匹配public final或public static。 |
| 返回值类型 | 如果不限定类型，使用通配符*表示。 |
| 类名.函数名 | 可以使用的通配符，包括*和..以及+号。其中*号用于匹配除.号之外的任意字符，而..则表示任意子package，+号表示子类。 注： <ol style="list-style-type: none">ConstructorSignature的函数名只能为new；(.函数名)可以不写，常用于和注解一起使用；不能以..开头。 |
| 变量类型 | 成员变量类型，*代表任意类型。 |
| 类名.成员变量名 | 类名可以使用的通配符，包括*和..以及+号，与类名.函数名类似；成员变量名可以是*，代表任意成员变量。 |
| 参数 | |
| 示例： <ol style="list-style-type: none">(int, char)：表示参数只有两个，并且第一个参数类型是int，第二个参数类型是char；(String, ..)：表示至少有一个参数。并且第一个参数类型是String，后面参数类型和个数不限。在参数匹配中，..代表任意参数个数和类型；(Object ...): 表示不定个数的参数，且类型都是Object，这里的...不是通配符，而是Java中代表不定参数的意思。 | |

1-3

| Advice | 说明 |
|-------------------|---|
| @Before(Pointcut) | 执行JPoint之前。 |
| @After(Pointcut) | 执行JPoint之后。 |
| @AfterReturning | @AfterReturning(pointcut = “xxx”, returning = “retValue”) |
| @AfterThrowing | @AfterThrowing(pointcut = “xxx”, throwing = “throwable”) |
| @Around(Pointcut) | 替代原来的代码，如果要执行原来的代码，需要使用 ProceedingJoinPoint#proceed() 。 注： 不支持和@Before()、@After()等一起使用 |

1-4

| 间接选择JPoint | 说明 |
|--|--|
| within(TypePattern) | TypePattern表示某个包或者类中包含的JPoint，可以使用通配符，包括*和..以及+号。 |
| withincode(Constructor Signature Method Signature) | 表示某个类的构造函数或方法中涉及到的JPoint。 |
| cflow(pointcuts) | 比如cflow(call Animal.fly)：表示调用Animal.fly函数时所包含的JPoint，包括fly的call这个JPoint本身。 |
| cflowbelow(pointcuts) | 比如cflowbelow(call Animal.fly)：表示调用Animal.fly函数时所包含的JPoint，但不包括fly的call这个JPoint本身。 |
| this(Type) | JPoint所在的对象是否满足instanceof Type条件。 注： <ol style="list-style-type: none">不能使用通配符；与within有点类似，但within包含内部类，而this不包含。 |
| target(Type) | 与 this 相对，表示“ Pointcut ”所在的对象是否满足instanceof Type条件。 注： 不能使用通配符。 |
| args(TypeSignature) | Constructor Signature Method Signature的参数类型，比如args(int,..)表示第一个参数是int，后面参数个数和类型不限。 注： 在申明Constructor Signature Method Signature的时候，也是可以指定参数类型的。 |