

# 在开始学习java逆向工程前需要知道什么？

## 术语：

**限定名称(Qualified name)**：用 `.` 分隔包名。例如 `Class.forName(name)` 的函数需要使用限定名称作为参数

举例：

- `java.lang.String`
- `com.example.MyClass.InnerClass`

注：只写最后一个类的名字，就是非限定名称。例如

```
import java.net.URL
URL a = new URL("www.baidu.com");//这里URL就是非限定名称，所以需要import
java.net.URL b = new java.net.URL("www.baidu.com");//和上面作用相同，但不需要import
```

**内部名称(Internal name)**：用 `/` 分隔包名。在 `$` 后接内部类名。内部名称是在 `.class` 类文件内部的类的名称

例如：

- `com/example/MyClass$InnerClass`

**原语 (Primitives)** ：使用单个字符来表示原语（原语即不可再分的类型）

原语	JVM中的描述
<code>long</code>	<code>J</code>
<code>int</code>	<code>I</code>
<code>short</code>	<code>S</code>
<code>byte</code>	<code>B</code>
<code>boolean</code>	<code>Z</code>
<code>float</code>	<code>F</code>
<code>double</code>	<code>D</code>
<code>void</code>	<code>V</code>

**描述符 (Descriptor)**:描述符被用来描述字段和方法类型。几乎和JVM内部类型相同，只是类名被 `L` 与 `;` 包裹。

例如：

- `Ljava/lang/String;`
- `I` (原语同上保持不变)

方法构造器格式举例：

- `double method(int i, String s) = (ILjava/lang/String;)D`

- `void method() = ()V`

每一层数组被 [ 前缀标记:

- `int[] = [I`
- `String[][] = [[Ljava/lang/String;`

**奇葩之处:** `double` 和 `long` 类型的变量占用两个槽位 (无论是在操作栈还是在局部变量表上的)

**方法(Method):** 类(Class)中的函数成员

**字段(Field):**类(Class)中的变量成员

**静态调试(static-analysis):**指的是对Java字节码进行分析

**动态调试(dynamic-analysis):**指得是利用调试器对正在运行中的VM进行跟踪来进行分析