

关键字就是 Java 语言赋予的特定含义的单词：**关键字一律用小写字母标识**，按其用途划分为如下几组。

(1)数据类型：用于数据类型的关键字有 *boolean*、*byte*、*char*、*double*、*false*、*float*、*int*、*long*、*new*、*short*、*true*、*void*、*instanceof*。

(2)语句：用于语句的关键字有 *break*、*case*、*catch*、*continue*、*default*、*do*、*else*、*for*、*if*、*return*、*switch*、*try*、*while*、*finally*、*throw*、*this*、*super*。

(3)用于修饰用于修饰的关键字有 *abstract*、*final*、*native*、*private*、*protected*、*public*、*static*、*synchronized*、*transient*、*volatile*。

(4)用于方法、类、接口、包和异常。用于方法、类、接口、包和异常的关键字有 *class*、*extends*、*implements*、*interface*、*package*、*import*、*throws*。

关键字及含义

|                            |  |
|----------------------------|--|
| <code>abstract</code>      | 表明类或者成员方法具有抽象属性  |
| <code>assert</code>        | 断言，用来进行程序调试  |
| <code>boolean</code>       | 基本数据类型之一，声明布尔类型的关键字  |
| <code>break</code>         | 跳出一个块  |
| <code>byte</code>          | 字节类型   |
| <code>case</code>          | 用在 switch 语句之中，表示其中的一个分支                                   |
| <code>catch</code>         | 用在异常处理中，用来捕捉异常   |
| <code>char</code>          | 字符类型   |
| <code>class</code>         | 声明一个类  |
| <code>const</code>         | 常量   |
| <code>continue</code>      | 继续   |
| <code>default</code><br>实现 | 默认，例如，用在 switch 语句中，表明一个默认分支。Java8 中也作用于声明接口函数的默认实现        |
| <code>do</code>            | 用在 do-while 循环结构中  |
| <code>double</code>        | 双精度浮点数类型   |
| <code>else</code>          | 否则，或者  |
| <code>enum</code>          | 枚举   |
| <code>extends</code>       | 表明一个类型是另一个类型的子类型。对于类，可以是另一个类或者抽象类；对于接口， <a href="#">继承</a> |
| <code>final</code>         | 用来说明最终属性，表明一个类不能派生出子类，或者成员方法不能被覆盖，成员域的值不能被改变               |
| <code>finally</code>       | 有没有异常都执行   |
| <code>float</code>         | 单精度浮点数类型   |
| <code>for</code>           | 一种循环结构的引导词   |
| <code>goto</code>          | 指定跳转语句   |
| <code>if</code>            | 条件语句的判断词   |
| <code>implements</code>    | 表明一个类实现了给定的接口  |
| <code>import</code>        | 表明要访问指定的类或包  |
| <code>instanceof</code>    | 用来测试一个对象是否是指定类型的实例对象                                       |
| <code>int</code>           | 整数类型   |
| <code>interface</code>     | 接口   |
| <code>long</code>          | 长整数类型  |
| <code>native</code>        | 用来声明一个方法是由与计算机相关的语言（如 C/C++/FORTRAN 语言）实现的                 |

|              |   |
|--------------|---|
| new          | 用来创建新实例对象   |
| package      | 包   |
| private      | 私有的   |
| protected    | 受保护的  |
| public       | 公有的   |
| return       | 返回数据  |
| short        | 短整数类型   |
| static       | 表明具有静态属性  |
| strictfp     | 用来声明 FP_strict（单精度或双精度浮点数）表达式遵循 IEEE 754 算术规范，严格，精准 |
| super        | 表明当前对象的父类型的引用或者父类型的构造方法                             |
| switch       | 分支语句结构的引导词 开关                                       |
| synchronized | 表明一段代码需要同步执行  |
| this         | 指向当前实例对象的引用   |
| throw        | 抛出一个异常对象  |
| throws       | 声明一个异常可能被抛出   |
| transient    | 声明不用序列化的成员域   |
| try          | 异常的程序块  |
| void         | 空返回值  |
| volatile     | 表明两个或者多个变量必须同步地发生变化 易失                              |
| while        | 循环语句  |
| null         | 空值  |
| true         | 真   |
| false        | 假   |