程序中的错误一般被称为 Bug，无可否认，这几乎总是程序员的错。。。

程序员的一生，始终伴随着一件事 - 调试（错误检测、异常处理）。反反复复，最可怕的是：不仅自己的要改，别人的也要改。。。一万头草泥马奔腾而过！

**错误**

程序错误，主要分为三类：

1. 语法错误
2. 逻辑错误
3. 运行时错误

**语法错误**

语法错误（也称：解析错误）：是指不遵循语言的语法结构引起的错误（程序无法正常编译/运行）。  
在编译语言（例如：C++）中，语法错误只在编译期出现，编译器要求所有的语法都正确，才能正常编译。不过对于直译语言（例如：Python）来说，语法错误可能在运行期才会出现，而且不太容易区分语法错误及语义错误。

常见的 Python 语法错误有：

1. 遗漏了某些必要的符号（冒号、逗号或括号）
2. 关键字拼写错误 prink('hello,Underworld')
3. name 'prink' is not defined
4. 缩进不正确 unexpected indent
5. 空语句块（需要用 pass 语句）
6. 标识符中的无效字符 invalid character in identifier

例如，编写这样一个条件判断语句：

[?](https://www.jb51.net/article/133918.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | >>> if n < 5   File "<stdin>", line 1    if n < 5        ^  SyntaxError: invalid syntax |

由于 if 语句中缺少一个冒号（:），不符合 Python 语法，因此程序无法正常运行。

**逻辑错误**

逻辑错误（也称：语义错误）：是指程序的执行结果与预期不符（程序可以正常运行，不会 Crash）。  
与语法错误不同的是，逻辑错误从语法上来说是正确的，但会产生意外的输出或结果，并不一定会被立即发现。逻辑错误的唯一表现就是错误的运行结果。

常见的逻辑错误有：

1. 运算符优先级考虑不周
2. 变量名使用不正确
3. 语句块缩进层次不对
4. 在布尔表达式中出错

例如，要计算两个数的平均值：

[?](https://www.jb51.net/article/133918.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | >>> def average(a, b):  ...   return a + b / 2 # 应为 (a + b) / 2  ...  >>> |

虽然程序能够正常运行，但代码中缺少括号，由于运算符优先级（先乘除，后加减）的缘故，运算结果并不正确。

**运行时错误**

运行时错误：是指程序可以运行，但是在运行过程中遇到错误，导致意外退出。  
当程序由于运行时错误而停止时，通常会说程序崩溃了。在 Python 中，这种运行时错误被称为异常。

Python 中的一些运行时错误：

1. 使用未定义的标识符（NameError）
2. 除数为 0（ZeroDivisionError）
3. 打开的文件不存在（FileNotFoundError）
4. 导入的模块没被找到（ImportError）

例如，除数为 0：

[?](https://www.jb51.net/article/133918.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | >>> 5 / 0  Traceback (most recent call last):  追溯（计算机病毒名）   File "<stdin>",line 1, in <module>  标准输入（设备）文件  ZeroDivisionError: division by zero |

每当出现这类运行时错误时，Python 就会创建一个异常对象。如果处理不当，会输出一个跟踪（Traceback）到那个错误，以及关于为什么会出现错误的一些细节。

**内置异常的层次结构**

Python 中有许多内置的异常，当出现相应的错误时，就会引发这些异常。

在 [Built-in Exceptions](https://docs.python.org/3/library/exceptions.html) 中，有一个内置异常的完整层次结构：

BaseException  
+-- SystemExit  
+-- KeyboardInterrupt  
+-- GeneratorExit  
+-- Exception  
+-- StopIteration  
+-- StopAsyncIteration  
+-- ArithmeticError  
| +-- FloatingPointError  
| +-- OverflowError  
| +-- ZeroDivisionError  
+-- AssertionError  
+-- AttributeError  
+-- BufferError  
+-- EOFError  
+-- ImportError  
+-- ModuleNotFoundError  
+-- LookupError  
| +-- IndexError  
| +-- KeyError  
+-- MemoryError  
+-- NameError  
| +-- UnboundLocalError  
+-- OSError  
| +-- BlockingIOError  
| +-- ChildProcessError  
| +-- ConnectionError  
| | +-- BrokenPipeError  
| | +-- ConnectionAbortedError  
| | +-- ConnectionRefusedError  
| | +-- ConnectionResetError  
| +-- FileExistsError  
| +-- FileNotFoundError  
| +-- InterruptedError  
| +-- IsADirectoryError  
| +-- NotADirectoryError  
| +-- PermissionError  
| +-- ProcessLookupError  
| +-- TimeoutError  
+-- ReferenceError  
+-- RuntimeError  
| +-- NotImplementedError  
| +-- RecursionError  
+-- SyntaxError  
| +-- IndentationError  
| +-- TabError  
+-- SystemError  
+-- TypeError  
+-- ValueError  
| +-- UnicodeError  
| +-- UnicodeDecodeError  
| +-- UnicodeEncodeError  
| +-- UnicodeTranslateError  
+-- Warning  
+-- DeprecationWarning  
+-- PendingDeprecationWarning  
+-- RuntimeWarning  
+-- SyntaxWarning  
+-- UserWarning  
+-- FutureWarning  
+-- ImportWarning  
+-- UnicodeWarning  
+-- BytesWarning  
+-- ResourceWarning

注意： 当捕获异常或者决定应该继承哪个异常类时，这个层次关系非常有用。

**总结**

以上就是本文关于Python 错误和异常代码详解的全部内容，希望对大家有所帮助。感兴趣的朋友可以继续参阅本站其他相关专题，如有不足之处，欢迎留言指出。感谢朋友们对本站的支持！