

# 异常处理

在开始之前，请注意这里我们将会尽可能详细阐述异常处理，一定要分清楚异常和错误的区别

## 什么是错误？什么是异常？

### 首先，我们来看一下，什么是错误

错误又称分析错误，意味着带么不符合规范，根本无法运行

In [4]:

```
# 以下，错误样例：
while True print('Hello world')

File "<ipython-input-4-42ea4a5d202b>", line 2
    while True print('Hello world')
            ^
```

**SyntaxError:** invalid syntax

解析器会输出出现语法错误的那一行，并显示一个“箭头”，指向这行里面检测到第一个错误。错误是由箭头指示的位置 上面 的 token 引起的（或者至少是在这里被检测出的）：在示例中，在 print() 这个函数中检测到了错误，因为它前面少了个冒号（:）。文件名和行号也会被输出，以便输入来自脚本文件时你能知道去哪检查。

### 接下来，我们来看一下，什么是异常

即使代码在表达上正确，在运行时，依然有可能会引发错误

In [10]:

```
# 以下，异常代码
b=123/0
```

```
-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-10-608ae74cd304> in <module>
      1 # 以下，异常代码
----> 2 b=123/0
```

**ZeroDivisionError:** division by zero

上面的代码是ZeroDivisionError,一个异常。b=123/0这段代码表达上并没有错误，但是运行时一定会发生错误，这就是异常

关于内置异常的各种信息，我们可以参阅：<https://docs.python.org/zh-cn/3/library/exceptions.html#bltin-exceptions> (<https://docs.python.org/zh-cn/3/library/exceptions.html#bltin-exceptions>)

## 处理异常

Python提供了可以处理异常的方法。下面我会提供一个例子，要求用户一直输入，直到输入了一个有效的整数（用户可以中断操作）。

In [1]:

```
while True:
    try:
        x = int(input('请输入一个数字: '))
        break
    except ValueError:
        print('这不是一个数字，请重新输入!')
    except KeyboardInterrupt:
        print('这不是一个数字，请重新输入!')
```

```
请输入一个数字: a
这不是一个数字，请重新输入!
请输入一个数字: x
这不是一个数字，请重新输入!
请输入一个数字: 123
```

try语句的工作原理如下：-首先，执行 try 子句（try 和 except 关键字之间的（多行）语句）。  
-如果没有异常发生，则跳过 except 子句 并完成 try 语句的执行。  
-如果在执行try 子句时发生了异常，则跳过该子句中剩下的部分。然后，如果异常的类型和 except 关键字后面的异常匹配，则执行 except 子句，然后继续执行 try 语句之后的代码。  
-如果发生的异常和 except 子句中指定的异常不匹配，则将其传递到外部的 try 语句中；如果没有找到处理程序，则它是一个 未处理异常，执行将停止并显示如上所示的消息。

<https://docs.python.org/zh-cn/3/tutorial/errors.html> (<https://docs.python.org/zh-cn/3/tutorial/errors.html>).