

PYTHON

编程基础

异常处理-else、 finally和raise

else

else子句是try except语句中的一个可选项。

如果try子句执行时没有发生异常，则在try子句执行结束后会执行else子句；
否则，如果发生异常，则else子句不会执行。

else

例

else子句示例。

```
for i in range(3): #循环3次
    try:
        num=int(input('请输入一个数字：'))
        print(10/num)
    except ValueError:
        print('值错误！')
    except:
        print('其他异常！')
    else:
        print('else子句被执行！')
```

请输入一个数字：abc
值错误！
请输入一个数字：0
其他异常！
请输入一个数字：10
1.0
else子句被执行！

finally

finally子句是try except语句中的另一个可选项。

无论try子句执行时是否发生异常，finally子句都会被执行。

finally

例

finally子句示例。

```
for i in range(3): #循环3次
    try:
        num=int(input('请输入一个数字：'))
        print(10/num)
    except ValueError:
        print('值错误！')
    except:
        print('其他异常！')
    finally:
        print('finally子句被执行！')
```

```
请输入一个数字：abc
值错误！
finally子句被执行！
请输入一个数字：0
其他异常！
finally子句被执行！
请输入一个数字：10
1.0
finally子句被执行！
```

raise

除了系统遇到错误产生异常外，我们也可以使用raise产生异常。

例

raise使用示例。

```
1 for i in range(2): #循环2次
2     try:
3         num=int(input('请输入一个数字：'))
4         if num==0:
5             raise ValueError('输入数字不能为0！')
6         print(10/num)
7     except ValueError as e:
8         print('值错误：',e)
```

```
请输入一个数字：0
值错误： 输入数字不能为0！
请输入一个数字：10
1.0
```