

# PYTHON

## 编程基础

# 异常处理-断言和自定义异常

# 断言

使用assert可以判断一个条件是否成立，如果成立则继续执行后面的语句；如果不成立则会引发AssertionError异常。

例

assert使用示例。

```
for i in range(2): #循环2次
    try:
        num=int(input('请输入一个数字：'))
        assert num!=0
        print(10/num)
    except AssertionError:
        print('断言失败！')
```

请输入一个数字：0  
断言失败！  
请输入一个数字：10  
1.0

# 自定义异常

自定义异常，实际上就是以BaseException类作为父类创建一个子类。

例

自定义异常示例。

```
class ScoreError(BaseException): #以BaseException类作为父类创建ScoreError类
    def __init__(self,msg): #定义构造方法
        self.msg=msg
    def __str__(self): #定义__str__方法，将ScoreError类对象转换为字符串时自动调用
        return self.msg
```

# 自定义异常

```
if __name__ == '__main__':  
    for i in range(2): #循环2次  
        try:  
            score=int(input('请输入一个成绩 : '))  
            if score<0 or score>100:  
                raise ScoreError('输入成绩为%d , 成绩应在0-100之间'%score)  
            print('输入成绩为%d'%score)  
        except ScoreError as e:  
            print('分数错误 : ',e)
```

# 自定义异常

请输入一个成绩：90

输入成绩为90

请输入一个成绩：-1

分数错误：输入成绩为-1，成绩应在0-100之间