

实例 1：计算器

计算器极大地提高了人们进行数字计算的效率与准确性，无论是超市的收银台，还是集市的小摊位，都能够看到计算器的身影。计算器最基本的功能是四则运算。本实例要求编写程序，实现计算器的四则运算功能。

实例目标

- 熟练地定义函数
- 熟练地调用函数

实例分析

由实例描述可知，本实例的计算器具有最基本的加、减、乘、除 4 项功能，可实现计算两个数的和、差、积、商。因此，这里可以定义一个包含两个参数的函数，第 1 个参数接收用户输入的第 1 个数，第 2 个参数接收用户输入的第 2 个数，该函数主要实现的是加、减、乘、除 4 项功能，执行哪种功能需用户输入相应的运算符，再根据该运算符计算结果即可。

代码实现

```
def oper(parm_one, parm_two):  
    operator = input('请选择要执行的运算符: +、-、*、/' + '\n')  
    if operator == "+":  
        print("计算结果为:", +parm_one + parm_two)  
    elif operator == "-":  
        print("计算结果为:", parm_one - parm_two)  
    elif operator == "*":  
        print("计算结果为:", parm_one * parm_two)  
    elif operator == "/":  
        if parm_two == 0:  
            print('被除数不能为 0')  
        else:  
            print("计算结果为:", parm_one / parm_two)  
num_one = int(input('请输入第一个数:'))  
num_two = int(input('请输入第二个数:'))  
oper(num_one, num_two)
```

以上代码首先定义了一个包含两个参数 `parm_one` 和 `parm_two` 的函数 `oper()`，该函数中

接收了用户输入运算符 `operator`，并根据不同的运算符 `operator` 执行相应的运算，然后接收了用户输入的两个数 `num_one` 和 `num_two`，最后调用了 `oper()` 函数。

代码测试

运行程序，在控制台依次输入“10”、“10”、“*”之后的结果如下所示：

```
请输入第一个数:10
请输入第二个数:10
请选择要执行的运算符: +、-、*、/
*
计算结果为: 100
```

运行程序，在控制台依次输入“88”、“8”、“/”之后的结果如下所示：

```
请输入第一个数:88
请输入第二个数:8
请选择要执行的运算符: +、-、*、/
/
计算结果为: 11.0
```

运行程序，在控制台依次输入“43”、“0”、“/”之后的结果如下所示：

```
请输入第一个数:43
请输入第二个数:0
请选择要执行的运算符: +、-、*、/
/
被除数不能为 0
```