Python

极夜酱

目录

Chapter 1 函数

1.1 函数

1.1.1 函数（Function）

函数执行一个特定的任务，Python 提供了大量内置函数，例如 print() 用来输出字符串、len() 用来计算序列长度等。

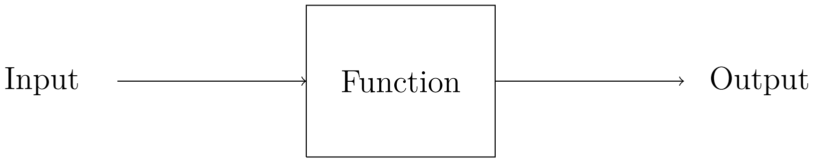

图 1.1: 函数

当调用函数时，程序控制权会转移给被调用的函数，当函数执行结束后，函数会把程序序控制权交还给其调用者。

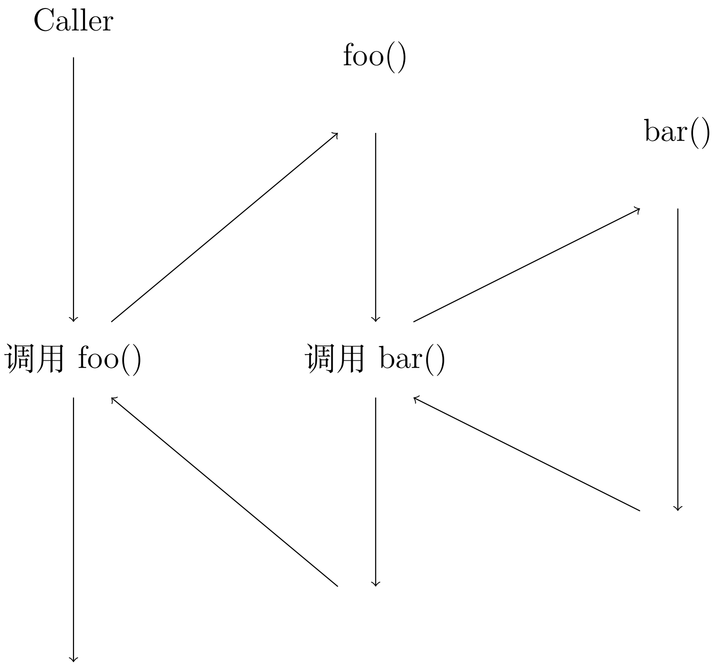

图 1.2: 函数调用

使用 def 关键字可以定义函数：

```python
def func_name([param_list]):
    # code
```

1.1.2 函数设计方法

为什么不把所有的代码全部写在一起，还需要自定义函数呢?

使用函数有以下好处:

1. 避免代码复制，代码复制是程序质量不良的表现
2. 便于代码维护
3. 避免重复造轮子，提高开发效率

在设计函数的时候需要考虑以下的几点要素:

1. 确定函数的功能
2. 确定函数的参数
    - 是否需要参数
    - 参数个数
    - 参数类型
3. 确定函数的返回值
    - 是否需要返回值
    - 返回值类型

函数实现返回最大值

```python
def get_max(num1, num2):
    # if num1 > num2:
    #     return num1
    # else:
```

```python
    #     return num2

    return num1 if num1 > num2 else num2

print(get_max(4, 12))
print(get_max(54, 33))
print(get_max(0, -12))
print(get_max(-999, -774))
```

运行结果

```
12
54
0
-774
```

函数实现累加和

```python
def get_sum(start, end):
    total = 0
    for i in range(start, end+1):
        total += i
    return total

print("1-100的累加和 = %d" % get_sum(1, 100))
print("1024-2048的累加和 = %d" % get_sum(1024, 2048))
```

运行结果

```
1-100的累加和 = 5050
1024-2048的累加和 = 1574400
```

函数实现输出 i 行 j 列由自定义字符组成的图案

```python
def print_chars(row, col, c):
    for i in range(row):
        for j in range(col):
            print(c, end='')
        print()

print_chars(5, 10, '?')
```

运行结果

```
??????????
??????????
??????????
??????????
??????????
```

1.2 主函数

1.2.1 主函数

Python 是为数不多的直接定义完源代码就可以执行的编程语言，很多编程语言对于程序的执行都有非常严格的标准，例如 C、C++、Java 等都有主函数（主方法）来标记程序的起点。

在现实的开发之中主函数是很有必要的，可以区分出其它的结构，在模块中更需要主函数的使用。

在 Python 中如果想实现主函数的定义，必须借助于全局变量 __name__ 的返回内容，而后采用自定义的函数形式实现，返回的 __main__ 是一个字符串，这个内容是会改变的，跟程序身处的结构有关。

在很多的编程语言都将主函数通过 main 这个标识符来定义，所以定义的 main() 是符合一般的习惯的。在进行复杂开发的时候，强烈建议使用主函数作为程序的起点。

主函数

```python
def main():
    pass

if __name__ == "__main__":
    main()
```