

任务6-1 函数的定义和调用

什么是函数？

函数是组织好的，可重复使用的，用来实现单一或相关联功能的**代码段**，它能够提高应用的模块性和代码的重复利用率。

```
print(" * ")  
print(" *** ")  
print("*****")
```

如果在一个程序的不同地方需要输出这个图形，每次使用print语言输出的做法显然**不可取**。

函数的定义和调用

Python定义函数使用def关键字，格式如下：

```
def 函数名（参数列表）：
```

```
    函数体
```

函数的定义和调用

这是一个自定义的函数：

```
def printInfo():  
    print('-----')  
    print('    生命苦短，我用Python    ')  
    print('-----')
```

函数的定义和调用

定义了函数之后，想要让这些代码能够执行，需要调用函数。通过“**函数名()**”即可完成调用。

```
# 调用刚才定义的函数  
printInfo()
```

函数命名应该符合标识符命名规则：

- 可以由字母、下划线和数字组成
- 不能数字开头
- 不能与关键字重名

函数的定义和调用

```
1 # coding=utf-8
2 # 定义函数
3 def say_hello():
4     print("hello world")
5
6 # 调用函数
7 say_hello()
```

运行结果：

```
hello world
```

函数的定义和调用

Python定义函数使用def关键字，格式如下：

```
def 函数名（参数列表）：
```

```
    函数体
```


函数的定义和调用

这是一个自定义的函数：

```
def printInfo():  
    print('-----')  
    print('    生命苦短，我用Python    ')  
    print('-----')
```

函数的定义和调用

定义了函数之后，想要让这些代码能够执行，需要调用函数。通过“**函数名()**”即可完成调用。

```
# 调用刚才定义的函数  
printInfo()
```

函数的定义和调用举例

```
1 # coding=utf-8
2 # 定义函数
3 def say_hello():
4     print("hello world")
5
6 # 调用函数
7 say_hello()
```

运行结果:

```
hello world
```

函数的参数是什么呢？

先看一段代码：

```
def add2num():  
    c = 11+22  
    print(c)
```

这个函数计算的只是固定的两个数，没有什么意义。

函数的返回值

如果希望定义的函数，可以计算任何两个数和，我们在定义函数的时候，让函数接收数据，这就是**函数的参数**。

```
def add2num(a, b):  
    c = a+b  
    print(c)
```

a和b就是函数的参数，调用的时候，可以传入任何两个数。

```
add2num(11, 22)
```

函数的返回值

参数调用过程

定义接收 2 个参数的函数

```
def add2num (a, b):
```

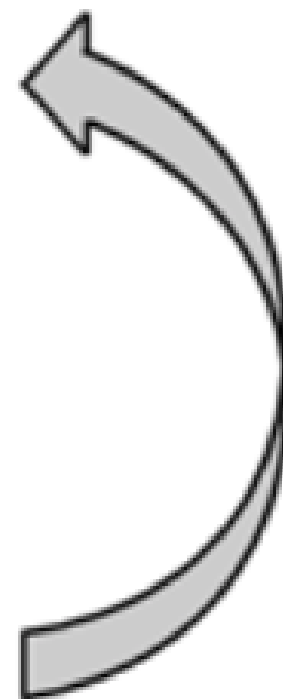
```
    c=a+b
```

```
    print (c)
```

调用带有参数的函数

```
add2num(11, 22)
```

此时：
a=11
b=22
c=33



默认参数

调用函数时，如果没有传递参数，则会使用默认参数。

```
def printinfo( name, age = 35 ):  
    print("Name:", name)  
    print("Age:", age)  
printinfo(name="miki" )  
printinfo(age=9,name="miki" )
```

带有默认值的参数一定要位于参数列表的最后面。否则程序会报错。

不定长参数

有时可能需要一个函数能处理比当初声明时更多的参数，这些参数叫做**不定长参数**，声明时不会命名。

```
def functionname([formal_args,] *args, **kwargs):
```

```
    "函数_文档字符串"
```

```
    function_suite
```

```
    return [expression]
```

- 加了星号 (*) 的变量args会存放所有未命名的变量参数，args为元组；
- 加**的变量kwargs会存放命名参数，即形如key=value的参数，kwargs为字典。

不定长参数

下面代码的运行结果是怎样的呢？

```
def test(a,b,*args):  
    print(a)  
    print(b)  
    print(args)  
test(11,22)
```

```
def test(a,b,*args):  
    print(a)  
    print(b)  
    print(args)  
test(11,22,33,44,55,66,77,88,99)
```