函数

定义函数

- def关键字,依次写出函数名、括号、括号中的参数和冒号：
- 在缩进块中编写函数体,函数的返回值用 return 语句返回。

```
def func():
        pass
        return
```

调用函数

Python 内置了很多有用的函数,我们可以直接调用:

- abs()
- cmp()
- 数据类型转换函数:

  int(), float(), str(),unicode(),bool()

理解函数名

- 函数名与变量名类似，其实就是指向一个函数对象的引用；
- 给这个函数起了一个“别名”：函数名赋给一个变量

```
In [53]: cmp(1,10)
Out[53]: -1

In [54]: a = cmp

In [55]: a(1,20)
Out[55]: -1
```

空函数

- 定义一个什么事也不做的空函数,可以用 pass 语句;
- pass 可以用来作为占位符,还没想好怎么写函数的代码,

就可以先放一个 pass ,让代码能运行起来

```
def nofunc():
        pass
```

参数检查

- 调用函数时,如果参数个数不对,Python 解释器会自动检查出来,并抛出 TypeError;
- 如果参数类型不对,Python 解释器就无法帮我们检查。
- 数据类型检查可以用内置函数 isinstance 实现

函数返回值

函数可以返回多个值吗?那编写python程序，思考下:

定义一个函数func，传入两个数字，返回两个数字的平均值与最大值。

函数返回值

- 函数返回值用return关键字;
- 返回一个 tuple 可以省略括号;
- Python 的函数返回多值其实就是返回一个 tuple
- 函数执行完毕也没有 return 语句时,自动 return None

函数参数

- 默认参数可以降低调用函数的难度。

定义一函数，计算x值的n次方。那如果计算x平方时只需要传入x值时怎么解决?

函数参数

默认参数注意事项：

- 有多个参数时,变化大放前面,变化小的放后面；
- 必选参数在前,默认参数在后

默认参数

默认函数容易出错点:

试一试:先定义一个函数,传入一个 list,添加一个 END 再返回.

默认参数

默认函数容易出错点:

试一试:先定义一个函数,传入一个 list,添加一个 END 再返回.

可变参数

- 可变参数就是传入的参数个数是可变的,可以是 1 个、2 个到任意个,还可以是 0 个。*args

以数学题为例子,给定一组数字 a,b,c……,

请计算 $a^2 + b^2 + c^2 + \ldots\ldots$

可变参数

如果已经有一个 list 或者 tuple,要调用一个可变参数怎么办?

- Python 允许你在 list 或 tuple 前面加一个 * 号;

- 把 list 或 tuple 的元素变成可变参数传进去;

```
largs = [1,2,3]

func(largs[0],largs[1],largs[2])

func(*largs)
```

关键字参数

- 关键字参数允许你传入 0 个或任意个含参数名的参数;
- 这些关键字参数在函数内部自动组装为一个 dict;
- 关键字参数用**kwargs

参数组合

- 参数组合是指可以必选参数、 默认参数、 可变参数和关键字参数一起使用。
- 参数定义的顺序必须是:必选参数、 默认参数、可变参数和关键字参数。

参数总结

- 必选参数
- 默认参数：默认参数一定要用不可变对象
- 可变参数：*args 是可变参数,args 接收的是一个 tuple
- 关键字参数：**kw 是关键字参数,kw 接收的是一个 dict
- 对于任意函数,都可以通过类似 func(*args, **kw) 的形式调用它

变量的作用域

- 局部变量：只能在函数内部使用的变量
- 全局变量：在整个程序中使用的变量
- global关键字：强制将局部变量转换为全局变量

over !