# 函数高级

## 课程目标

* 能够对函数进行默认参数的定义
* 能够使用关键字参数形式给函数形参赋值
* 能够使用 \*args 实现函数接收不定个数的位置参数
* 能够使用 \*\*kwargs 实现函数接收不定个数的关键字参数
* 能够说出递归函数定义的两个要素

## 课程内容

* 默认参数
* 关键字参数
* 可变参数
* 字典参数
* 递归函数
* 匿名函数

### 默认参数

#### 1.1默认参数概念

默认参数指函数/方法在定义时为形参赋值，对应的形参称为默认参数。

默认参数是一个参数定义期的概念，与调用无关。

#### 1.2默认参数作用

如果参数定义默认参数，在调用函数/方法时，未对该参数进行传值，则使用默认值作为该参数的值。

#### 1.3默认参数基本语法

定义格式:

def 函数名(形参1 = 值1，…)：

函数体

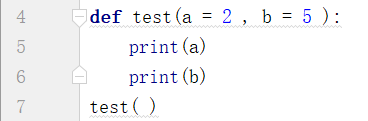
……

调用格式一（同普通参数，无特殊）：

函数名(实参) 使用实参作为形参的值，不使用默认值

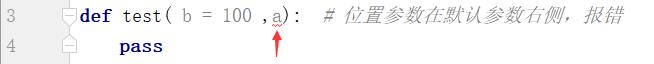
调用格式二：

函数名() **使用默认值作为形参的值**

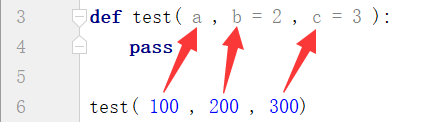


#### 1.4注意事项

1.默认参数的定义必须在位置参数的右侧



2.默认参数在调用传值时，按照位置参数进行对待，从左至右依次赋值，不能跳过左侧的某个参数直接为后面的参数赋值



### 关键字参数

#### 2.1关键字参数概念

函数/方法在调用时为指定名称的形参进行赋值，对应实参称关键字参数。

关键字参数是一个参数调用期的概念，与定义无关。

#### 2.2关键字参数作用

按名称确认实参为指定的形参赋值。

#### 2.3关键字参数基本语法

定义格式（同普通参数定义，无特殊）:

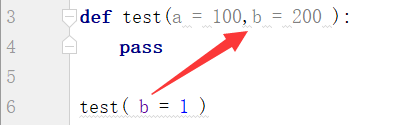
def 函数名(形参 ，…)：

函数体

……

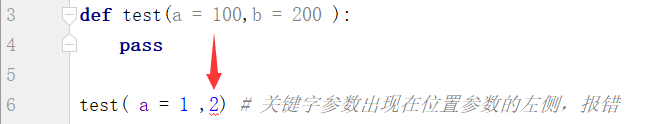
调用格式：

函数名( 形参名 = 实参)

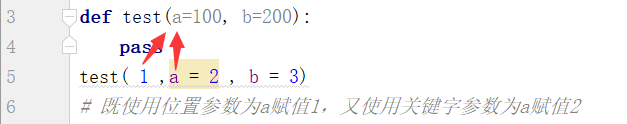
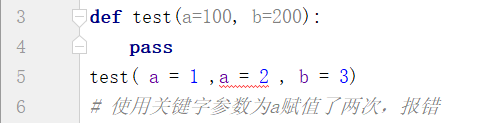


#### 2.4注意事项

1.关键字参数必须出现在位置参数的后面



2.不允许使用关键字参数对同一个形参进行多次赋值



3.关键字参数既可以为位置参数赋值，也可以为默认参数赋值。通常使用关键字参数解决默认参数选择性赋值问题。

### 可变参数

#### 3.1可变参数概念

函数/方法在定义时，定义一个特殊的形参，用于接收调用时传入的任意数量的实参，对应的形参称为可变参数。

可变参数是一个参数定义期与调用期都有效的概念。

#### 3.2可变参数作用

简化函数定义过程，定义可以接收无数个实参的形参

形参接受到的数据，无论数量多少，包装成一个元组对象

#### 3.3可变参数基本语法

定义格式:

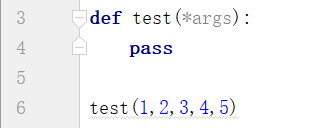
def 函数名(\*args ，…)：

函数体

……

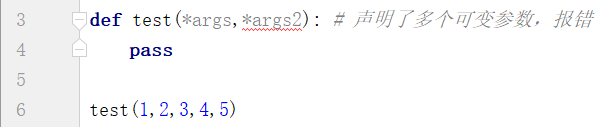
调用格式（同普通参数，无特殊）：

函数名( 实参1，…… )

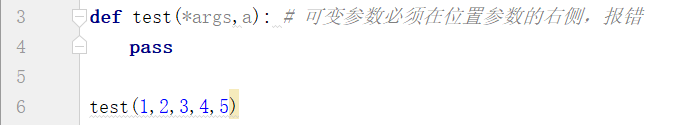


#### 3.4注意事项

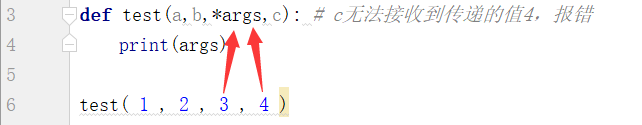
1.可变参数在一个函数/方法定义中只能出现一个



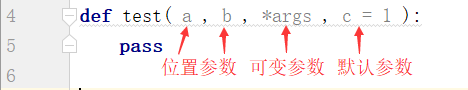
2.可变参数只能出现在位置参数的后面，位置参数在左侧，可变参数在右侧



3.关键字参数只能出现在位置参数的后面，赋值时必须使用关键字参数传值，否则值会被可变参数接收



4.定义函数的参数时，先定义位置参数，再定义可变参数，最后定义默认参数



### 字典参数

#### 4.1字典参数概念

函数/方法在定义时，定义一个特殊的形参，用于接收未定义直接使用的关键字参数，对应的形参称为字典参数。

#### 4.2字典参数作用

简化函数定义过程，接收未定义直接使用的关键字参数。

在多层级调用间进行关键字参数的逐级传递。

未定义的关键字参数，无论数量多少，接收后包装成一个字典对象。

#### 4.3字典参数基本语法

定义格式:

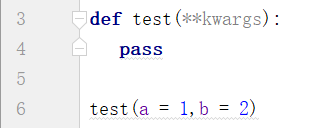
def 函数名(\*\*kwargs ，…)：

函数体

……

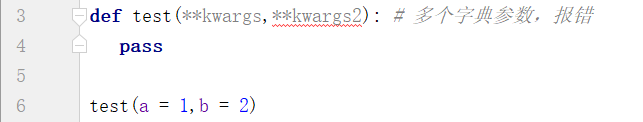
调用格式：

函数名( 未定义的变量名1 = 实参1，未定义的变量名2 = 实参2，…… )



#### 4.4注意事项

1.字典参数在一个函数/方法定义中只能出现一个



2.定义函数的参数时，先定义位置参数，再定义可变参数，然后定义默认参数，最后定义字典参数



#### 4.5其他

所有的容器都支持解包操作，格式：\*容器变量名

### 递归函数

#### 5.1递归函数概念

函数/方法在执行过程中出现了对自身函数/方法的调用，称该过程为递归调用，称这样的函数为递归函数。

#### 5.2递归函数基本语法

同普通函数

#### 5.3递归函数定义要求

1.函数体中具有对自身函数的调用

2.函数体中具有明确的函数执行结束判定条件

#### 5.4递归函数调用要求

Python语言中函数调用最大层级为1000层，超过该层级，程序将报错（不同语言设置不同）

### 匿名函数

#### 6.1匿名函数概念

匿名函数即没有名称的函数，在python语言中，匿名函数是使用lambda关键字定义的一个特殊表达式，也称为lambda表达式。

#### 6.2匿名函数基本语法

格式一:

定义： 函数名 = lambda 形参 ： 返回值

调用： 结果 = 函数名(实参)  


格式二：

定义并调用： 结果 = (lambda 形参 ： 返回值)(实参)



#### 6.3匿名函数特殊语法

无参格式: 结果 = (lambda ：返回值)()



多返回值格式: 无，可使用数据存储模型返回多个数据

无返回值格式： 无，必须有具体值或结果为None