**Python 基础语法**

**一、基础**

**1、字符串拓展**

**（1）字符串拼接**

name = “程序员”

age = “18”

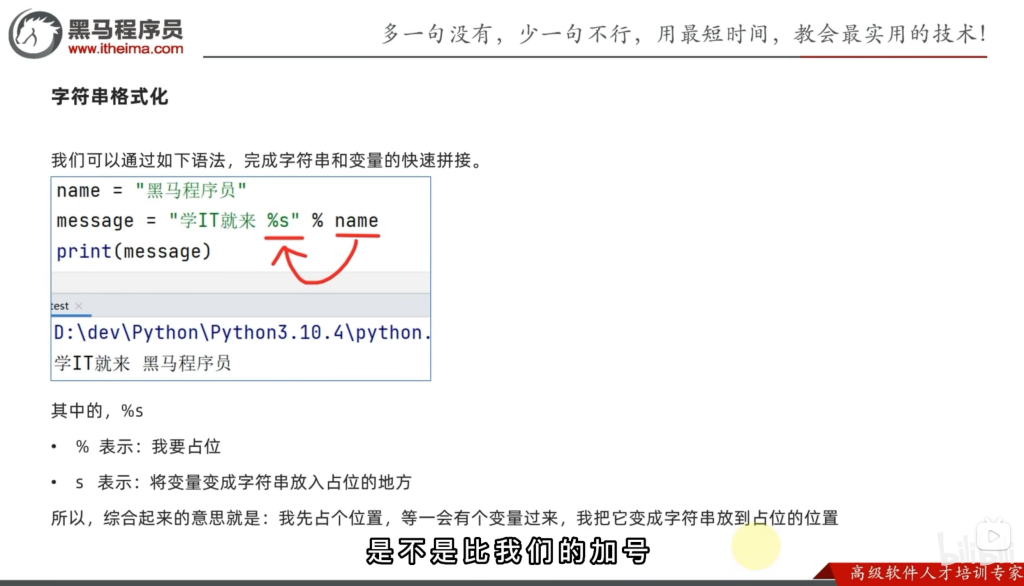
tel = 12345

print (“我是” + name + “, 我的年龄” + age + “岁”) # 正确

print (“我是” + name + “, 我的年龄” + age + “岁” + “我的电话是” + tel)

# 错误，字符串只能拼接字符串

**（2）字符串格式化（占位拼接）**

****

多个占位：print (“我的电话和名字是 %s , %s” % (tel , name))（顺序不能变）

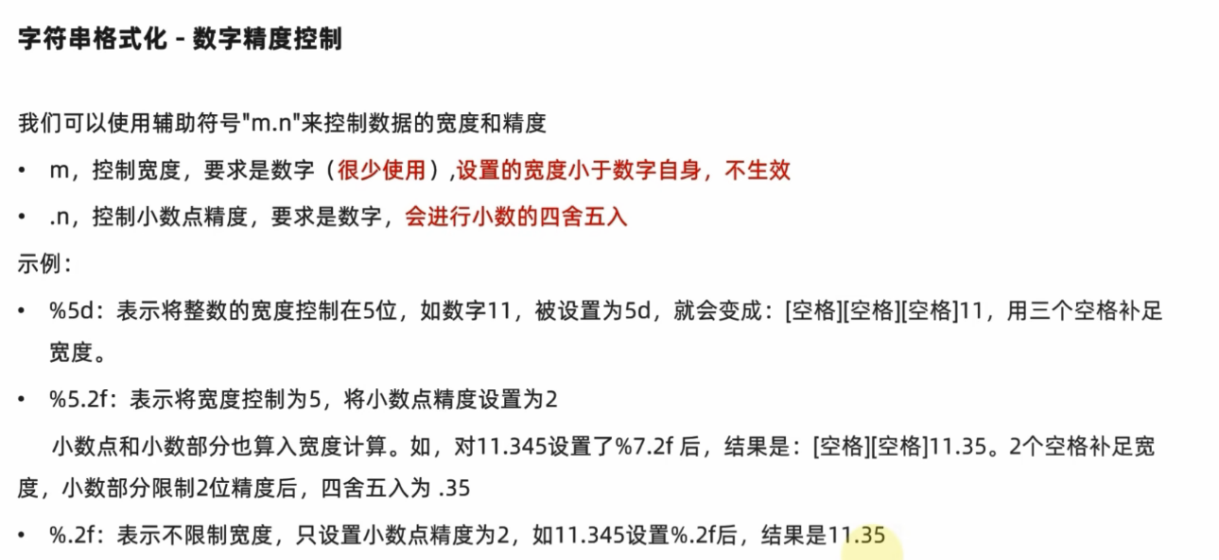
好处：可以拼接数字，拼接跟方柏霓

占位符号：%s：转换成字符串占位

%d：转换成整型占位

%f：转换成浮点型占位

1. **格式化精度控制**

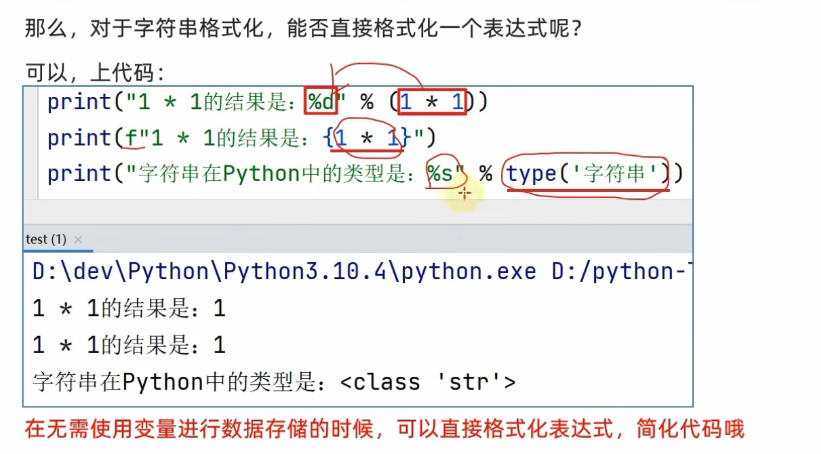


1. **快速格式化：无法控制精度，不限数据类型**



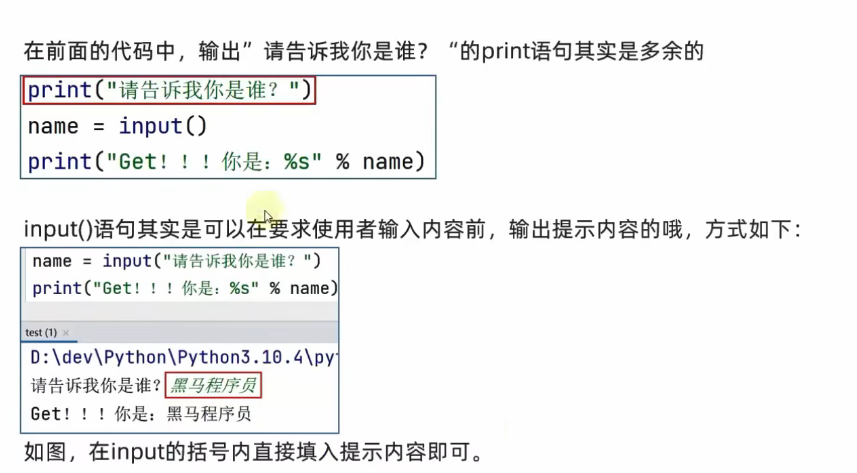
1. **对表达式进行格式化**

表达式：一条具有明确执行结果的代码语句



1. **数据输入**

input()：默认是字符串类型，需要时要数据类型转换：int(name)等等

****

1. **循环**
2. **布尔类型和比较运算符**

布尔类型的数据不仅可以通过定义得到，也可以通过比较运算符进行内容比较得到

result = 10 > 5

print (f”10 > 5的结果是：{result}，类型是：{type (result)}”)



1. **if语句**

格式：if 条件1:

（4个空格缩进）条件1成立时执行

elif 条件2:

（4个空格缩进）条件2成立时执行

......

else :

（4个空格缩进）没有条件不成立时执行

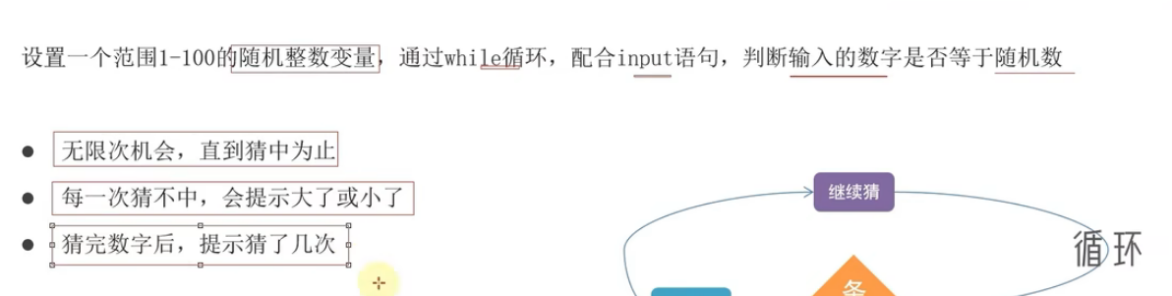
可以直接在if中写input，让代码更简洁

1. **while语句**

格式：while 条件:

（4个空格缩进）条件1成立时执行

案例：猜数字



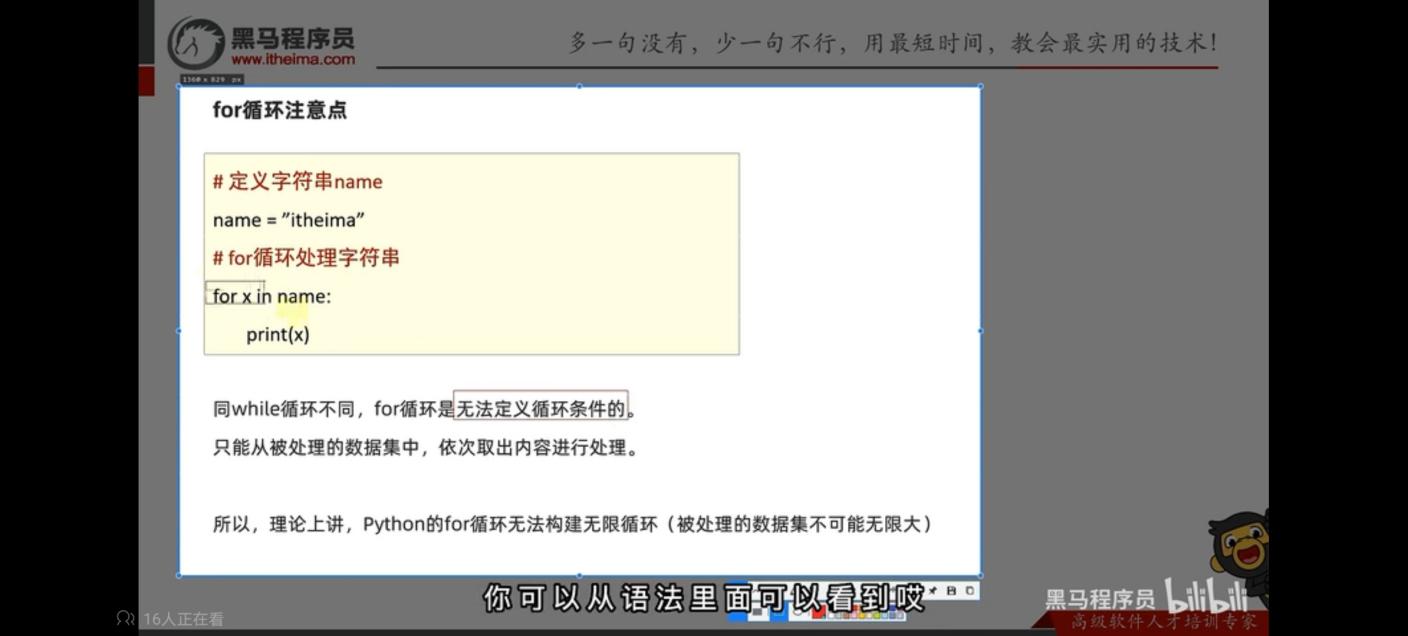
答案：

****

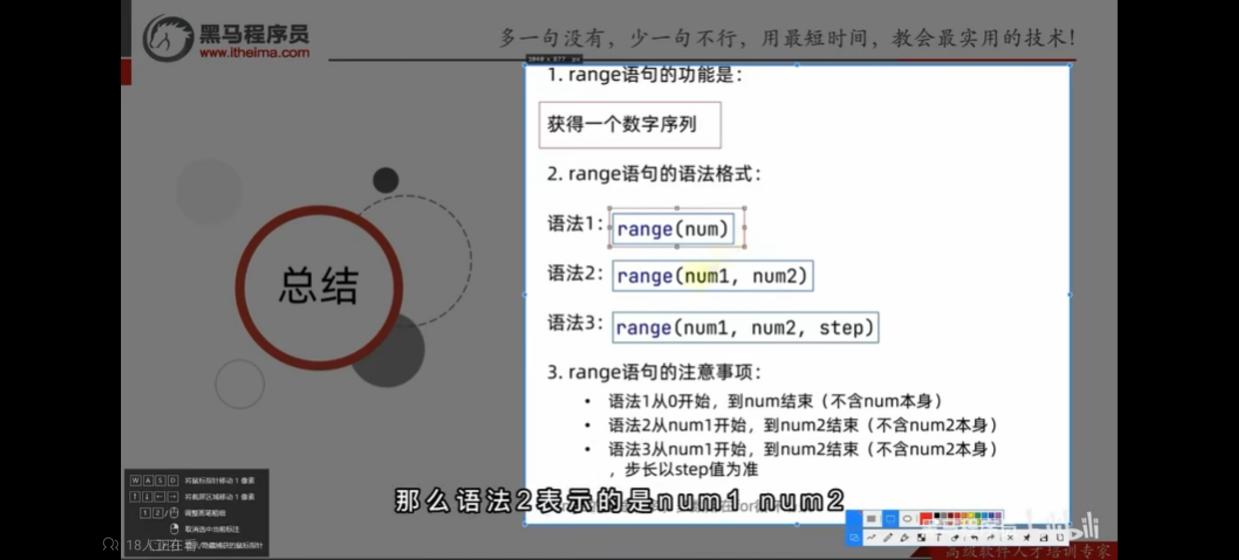
**4、for语句**

语法：for 参数 in 作用位置:

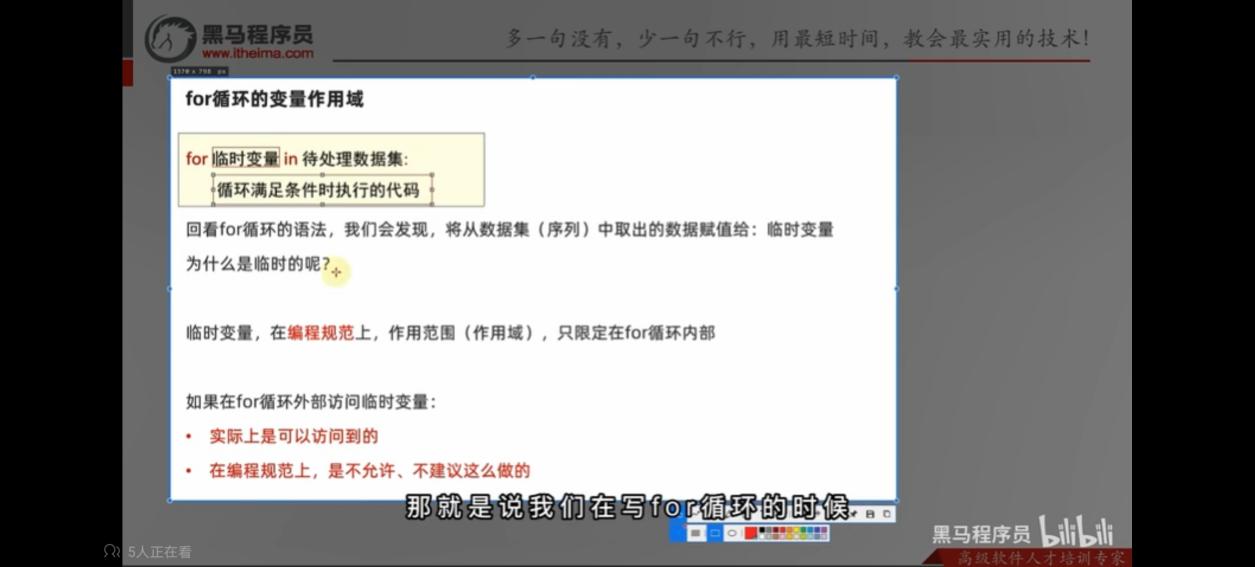
（4个空格缩进）执行



range语句：



变量作用域：



continue关键字：跳过本次循环进入下一次

break关键字：直接结束循环

**5、补充知识**

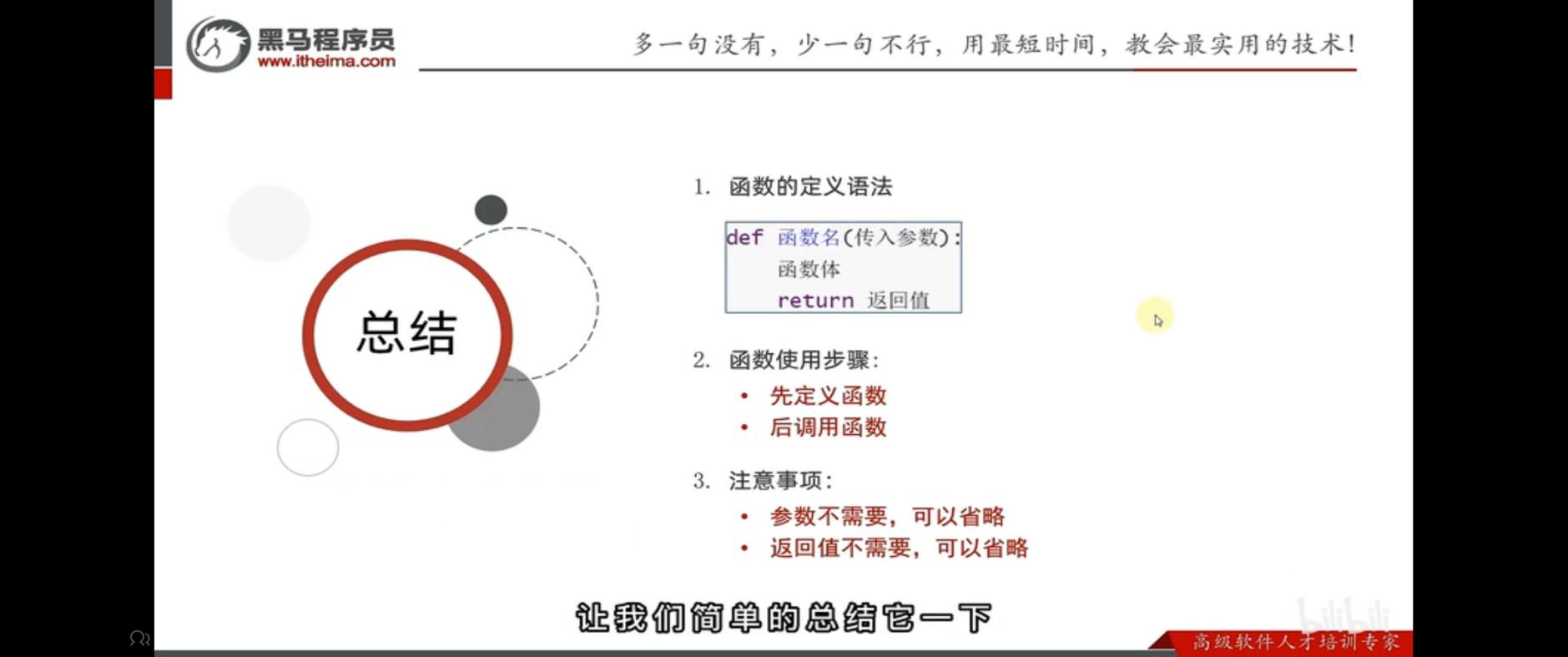
print不换行输出：print(…… , end＝'' '')

制表符对齐输出：print(''hello\tworld)

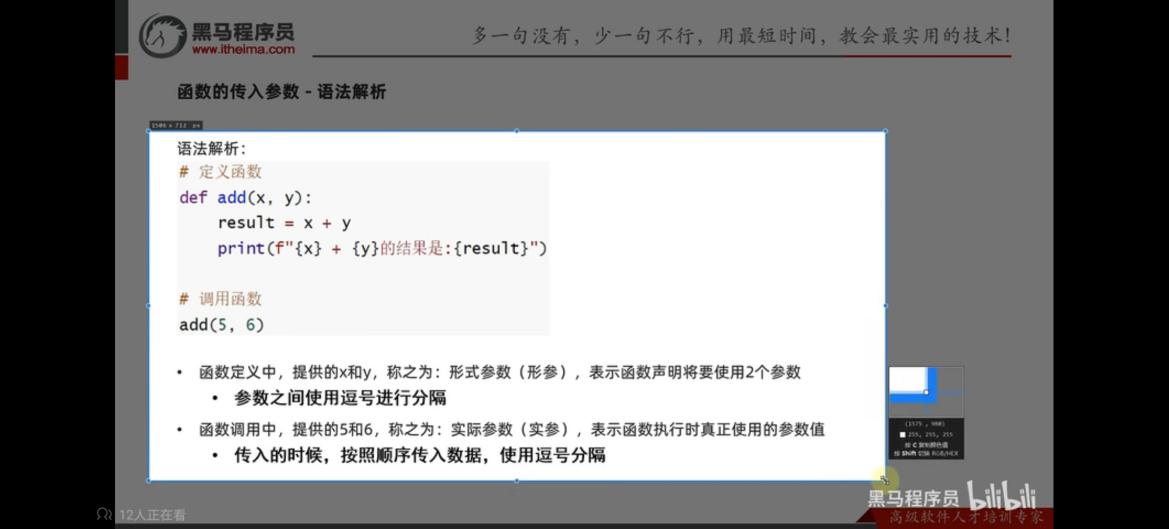
print(''你\t好)

**四、函数**

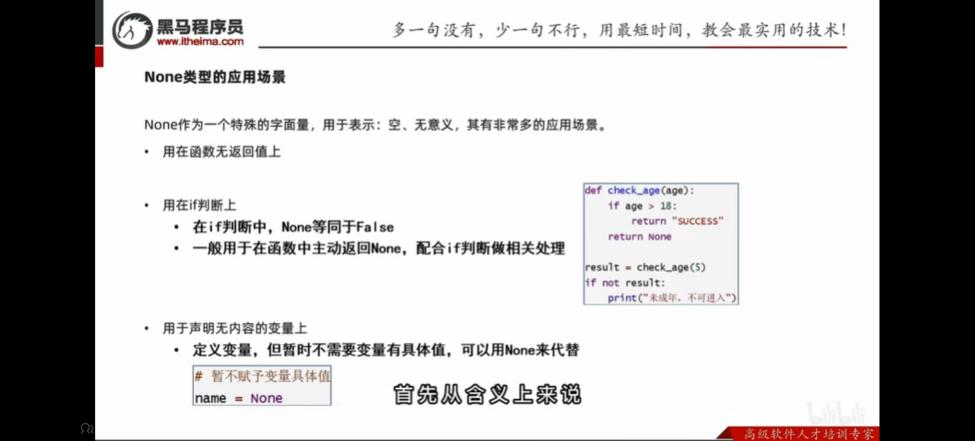
**1、函数的定义**



**2、传入参数**



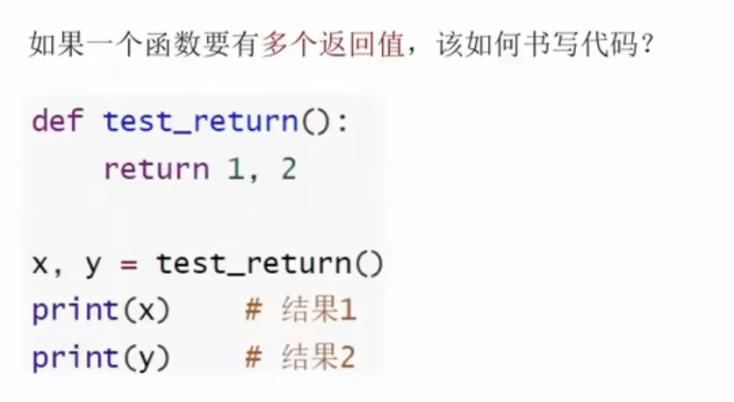
**3、None的使用**



**4、全局变量和局部变量**

global关键字让局部变全局

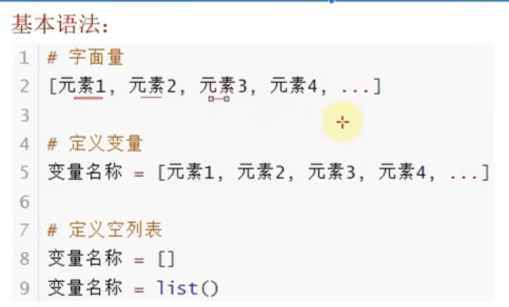
**5、多返回值**

****

按照返回值的顺序，用多个变量接收即可，且支持不同类型的返回值

**五、数据容器**

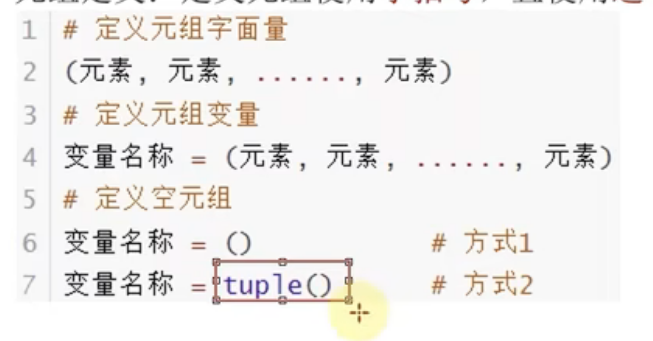
**1、列表：可修改，可重复，支持下标索引，支持任意数据类型**

****

基本操作：



**2、元组：不可修改，可重复，支持下标索引，支持任意数据类型**

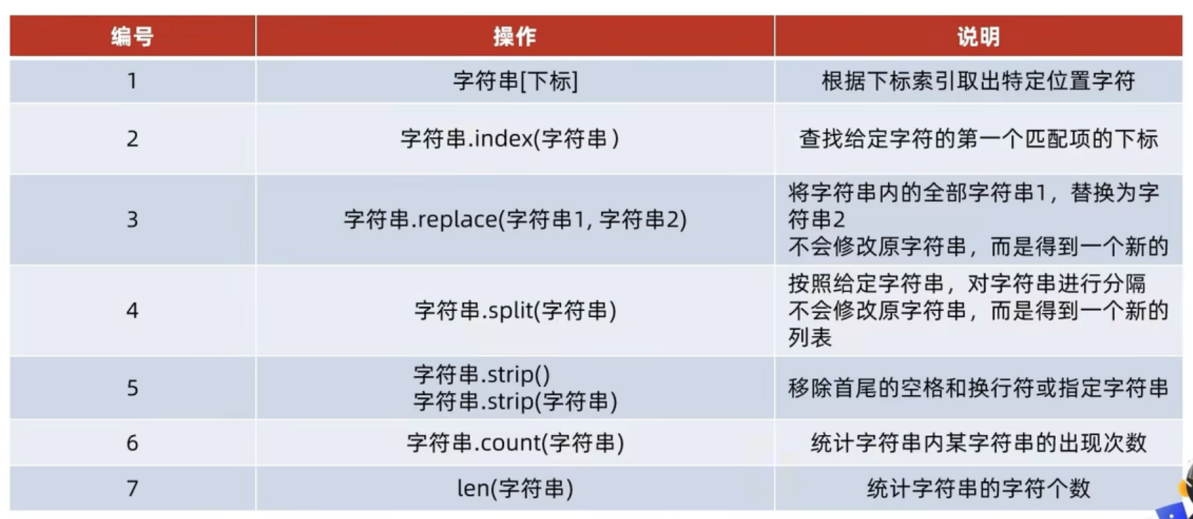
****

基本操作：

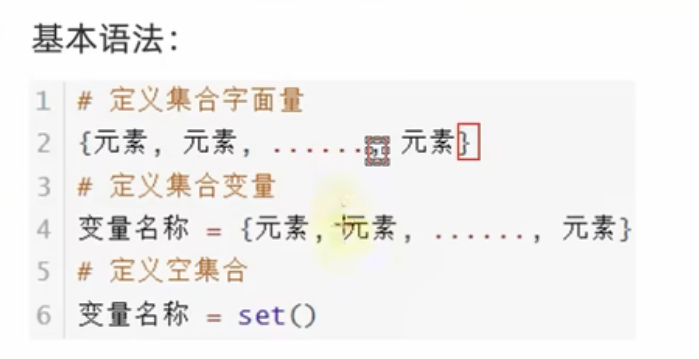
****

**3、字符串：不可修改，可重复，支持下标索引，仅支持字符串类型**

基本操作：

****

1. **集合：可修改，不可重复，不支持下标索引，支持任意数据类型，元素无序**

****

基本操作：



1. **字典：可修改，key不可重复，不支持下标索引，支持任意数据类型（key不能是字典）**

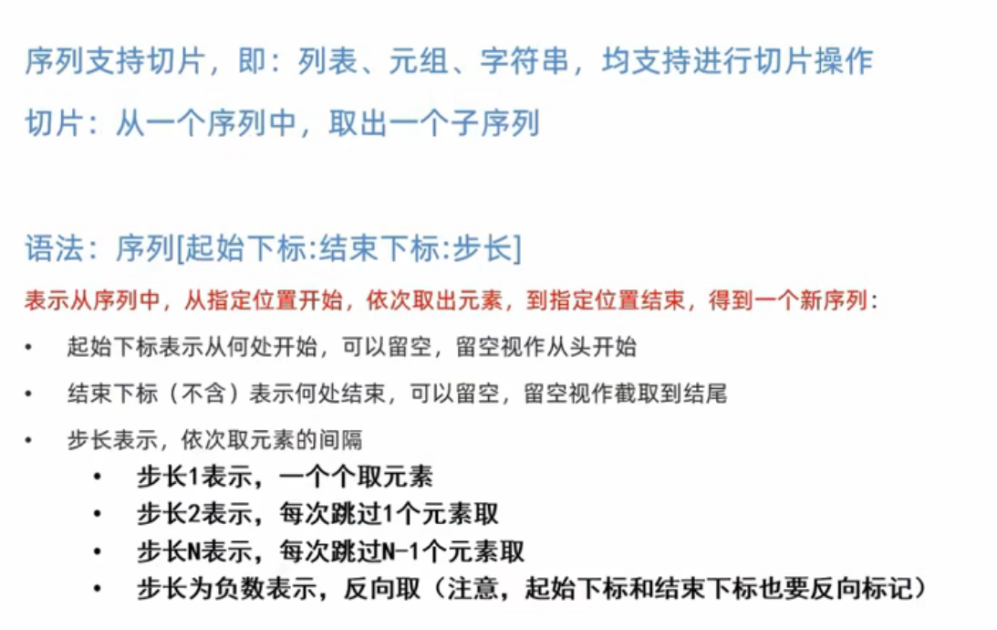
****

基本操作：



1. **序列的切片**

序列：列表，元组，字符串



1. **数据容器的通用操作：**

****

注意：字典转换为列表、元组和集合时会丢失value

所有数据容器排序后都是列表

1. **字符串大小比较：从头到尾，一位一位按照ASCII码进行比较，前面一位大，后面无需比较**