1. 摩尔定律：

其内容为:当价格不变时，集成电路上可容纳的元器件的数目，约每隔18-24个月便会增加一倍，性能也将提升一倍。换言之，每一美元所能买到的电脑性能，将每隔18-24个月翻一倍以上。这一定律揭示了信息技术进步的速度。

1. 编译和解释：

对源代码的处理不同，一个是直接翻译，一个是逐条翻译（对应静态语言和脚本语言）

1. 两种编程方式：

交互式和文件式

1. IPO的原则：对于编程。
2. 注释：

单行注释：以#开头

多行注释：以‘’‘开头以’‘’结尾

1. 命名与保留字：

变量：用来保存和表示数据的占位符号

命名规则：大小写字母，数字，下划线，汉族等组合；首字母不能是数字，不与保留字相同，大小写区分

保留字：一共33个，已经定义了，比如：if.elif （每一个保留字最后存在一个冒号：）

1. 数据类型：给计算机理解

字符串：在单双引号中。（字符串的序号：正向递增和反向递减）。

索引：<字符串>[M] #取出该字符串中第M个字符 例：取倒数第一个C[-1]

切片：返回字符串中一串字符：<字符串>[M:N] #不到第N个

浮点数和整数：

列表：由0个或多个数据组成的有序数列，用[]表示,用,分隔开元素，用in来判断一个元素是否在列表中

1. 语句和函数

赋值语句，分支语句（if，elif，else）

函数:格式<函数名>（<参数>）（print）

1. 程序的输入和输出

input（输入）：

print（输出）：print函数的格式化，

eval()：去掉参数最外侧引号并执行余下的语句

1. 整数类型

可正可负，没有取值范围，例如：pow(x,y)计算x的y

四种进制：十进制：

二进制：以0b或者0B开头：0b1010，-0b101

八进制：以0O或者0o开头

十六进制：以0x或者0X开头

1. 浮点数类型:

带有小数点和小数，取值范围大

浮点数之间运算会出现不确定尾数

其中round(x,d):对x取四舍五入，d表示小数截取的位数

科学计数法：e和E，例如4.3E-3

1. 复数类型

与数学中的复数一致

例如z=10+235j 可以用z.real获取实部，用z.imag获取虚部

1. 数值运算操作符

+ - \* /

其中/产生浮点数结果

其中x//y为整数除 x%y取模 x\*\*y求幂，y为小数时就是开方 二元操作符x op=y

类型间可以混合运算，生成结果为最宽的类型 100+4.0=104.0

1. 数值运算函数：  
   abs（x）求绝对值

divmod（x，y）求商余 divmod（10，3）结果为（3，1）

pow（x，y）求幂 还有pow（3，pow（3，99），10000）

round（x，d）四舍五入

max（）/min（）里面的数不限

int（x）转换为整数，舍弃小数部分

float（x）转换为浮点数，加上小数部分，也可以转换字符串

complex(x)转换为复数

1. 字符串

2类4种表示方式：（使得引号在字符串中可以出现）

由一对单引号或双引号，仅表示单行字符串

由一对三单引号或三双引号，可表示多行字符串

用<字符串>[M]得到想要的值

字符串切片：[M:N:K]从M到N间隔K切片，其中[::-1]将字符串逆序

转义符\，表示特定字符的本意 \b回退 \n换行 \r回车

1. 字符串操作

用x+y，连接两个字符串

用n\*x，复制n次字符串

用x in s，如果x是s子串，返回true

用len（x）返回字符串长度

用str（x）转换为字符串形式，和eval（）相反为一对

用hex（x）和oct（x）：整数x的十六进制或八进制的小写形式字符串

用chr（x）和ord（x）：Unicode编码和字符转换

1. 方法<a>.<b>

字符串方法：str.lower()或者str.upper可以转化大小写

用str.splid(sep):返回一个列表，把该字符串里用sep分隔开

用str.count(sub):计算sub在字符串中出现的次数

用str.replace(old,new):替换

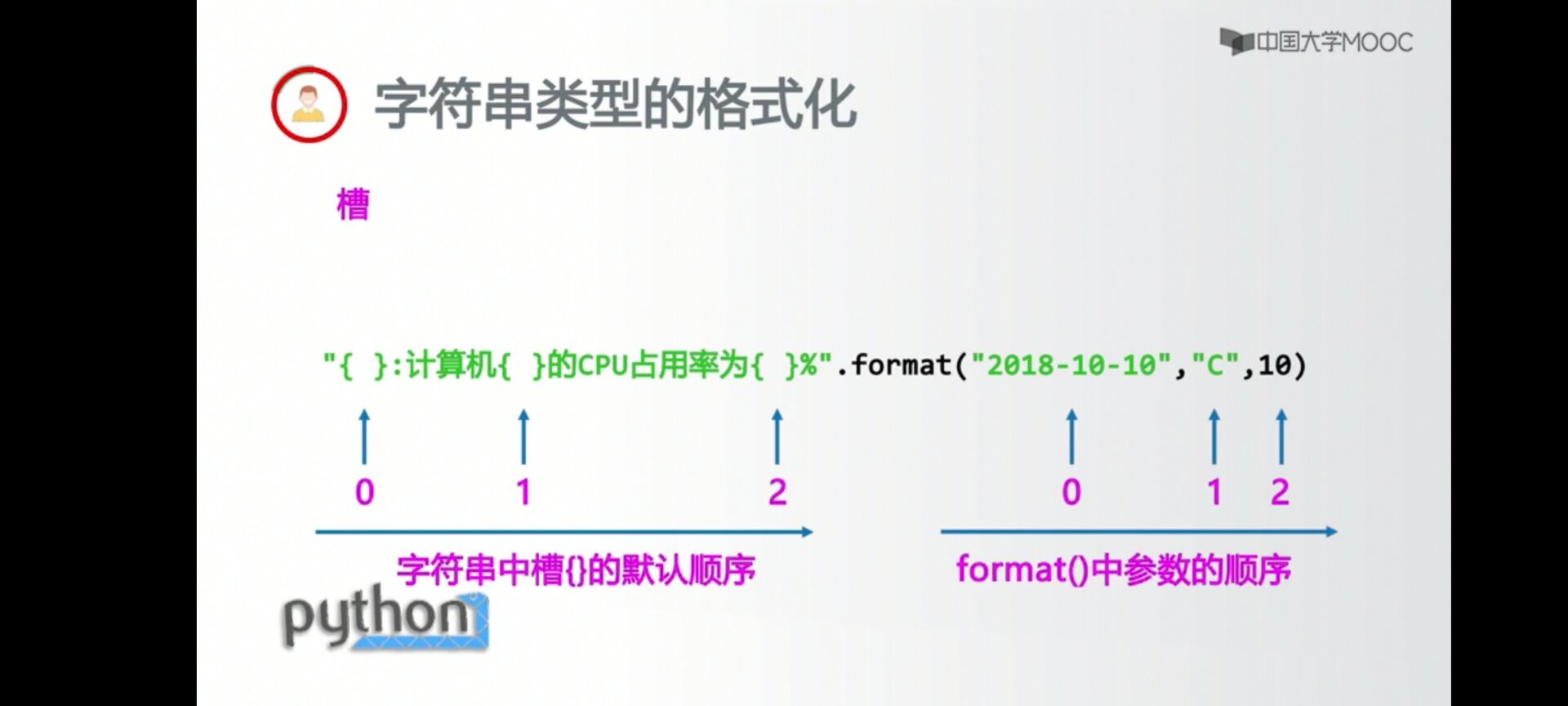
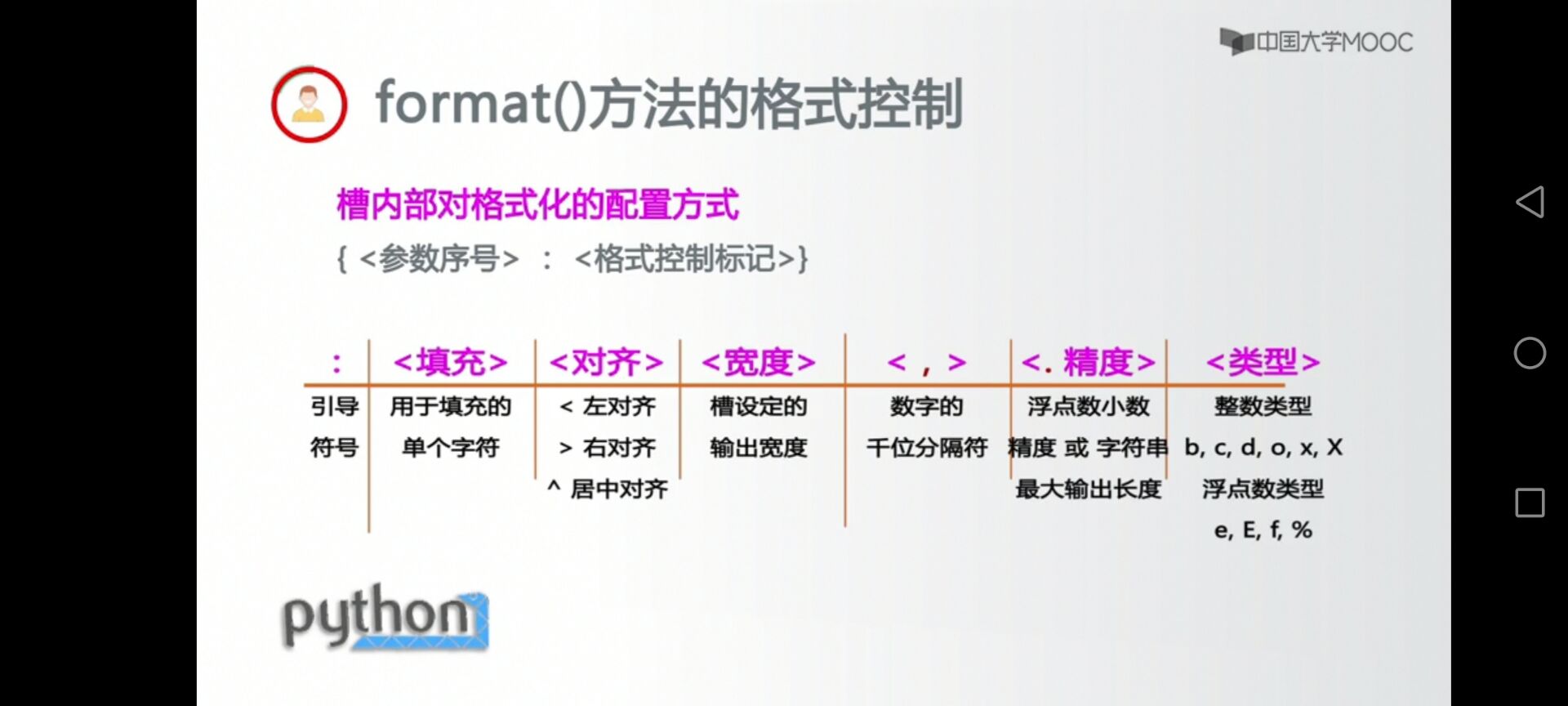
用str.center(width,fillchar)使字符居中

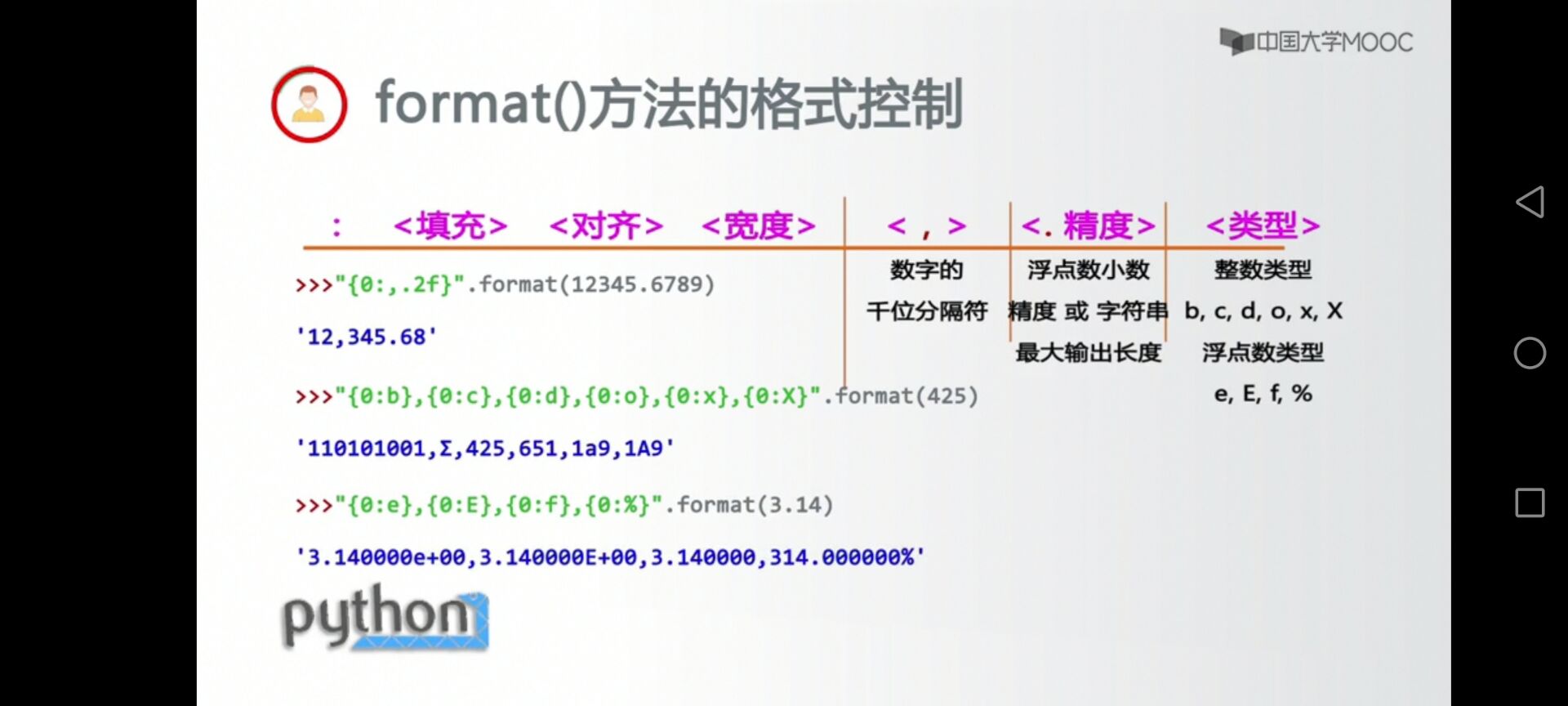
用str.strip(chars),去掉字符串两端特定的字符

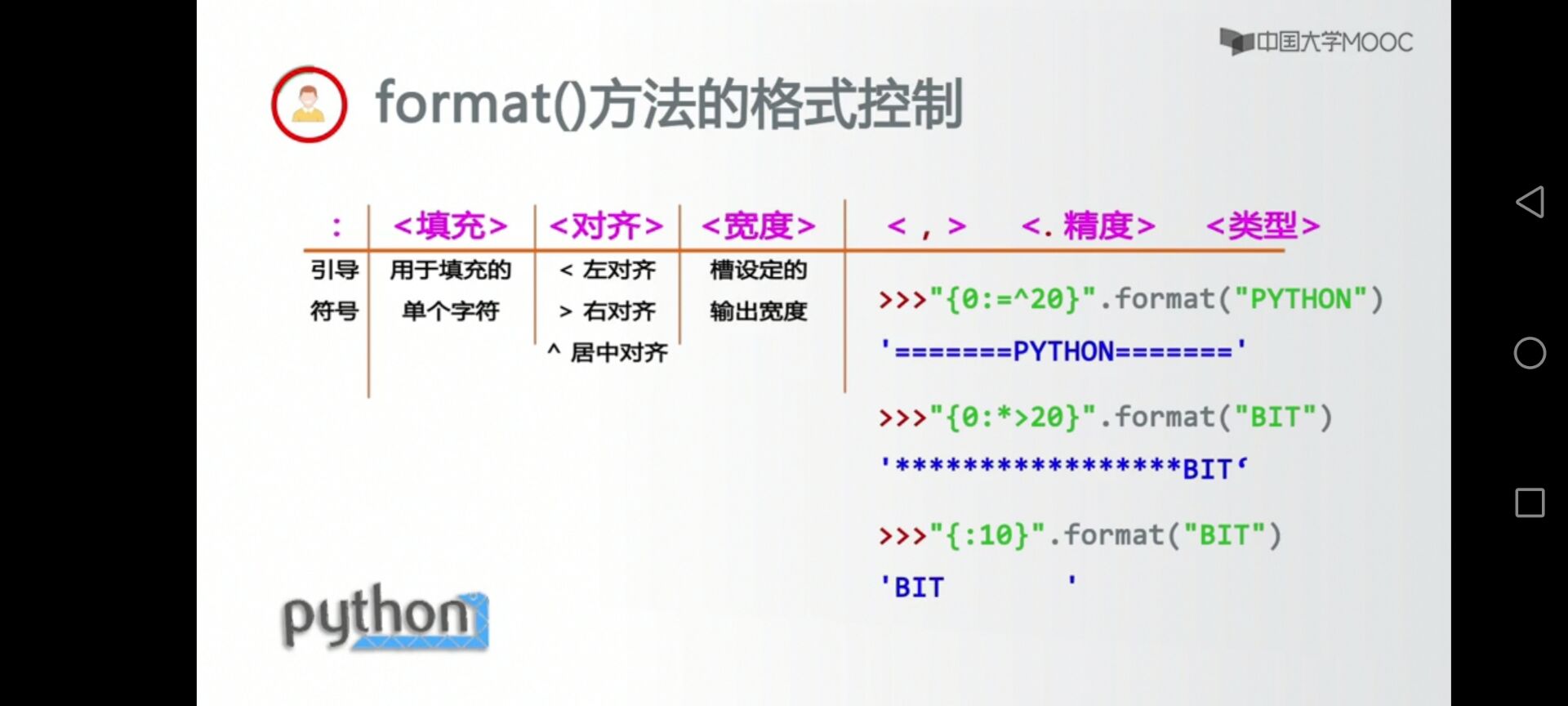
用str.join(iter),在每一个字符后面添加一个特定字符

1. 字符串类型的格式化

槽{}和.format()





1. Time库的使用

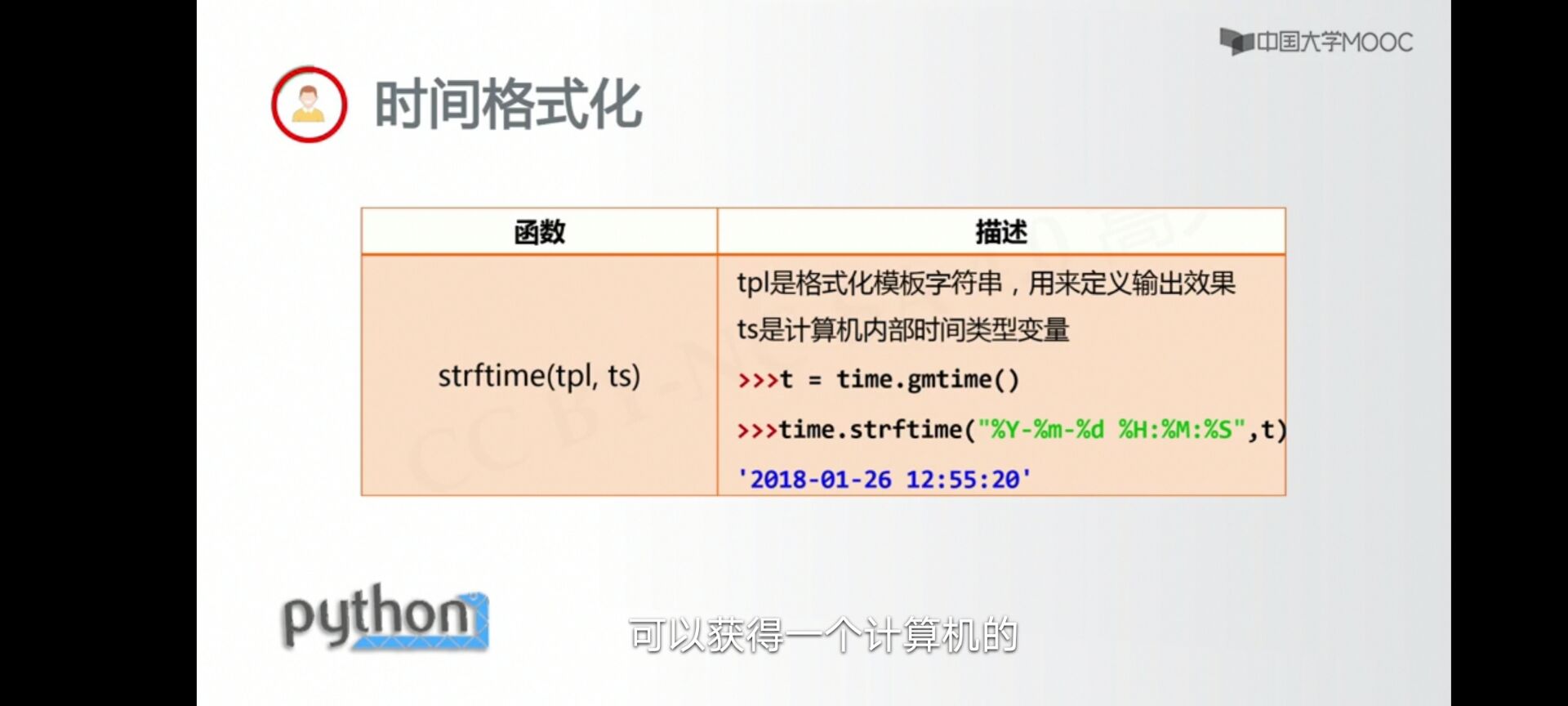
用import time

用time.time()获取时间戳

用time.ctime()获取具体的易读的

用time.gmtime()获取程序可以利用的时间程序

时间格式化：

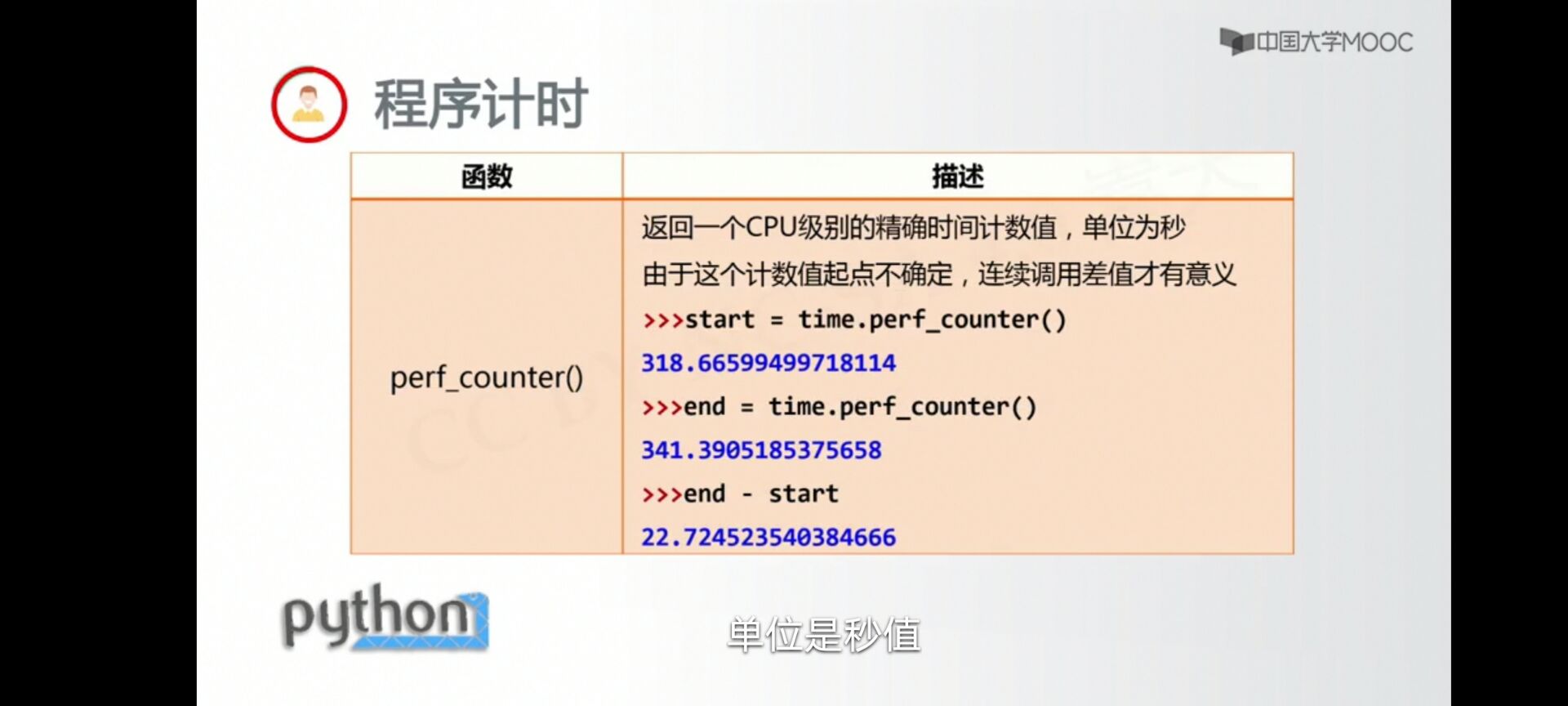
用







程运行时间的计算





1. \r回到行首进行刷新
2. 分支结构

单分支用if <条件>：+结构

二分支用if 和else

简洁形式：<表达式1> if <条件>else<表达式2>

多分支结构：if elif else

1. 用于条件组合的保留字

用and表示与

用or表示或

用not表示非

1. 异常处理

用try：语句一 except：语句二

如果语句一发生异常，就会执行语句二；或者在except后加上异常名字来表示，如果发生这种异常，则执行语句二，例如：except NameError：

高级引用



1. 遍历循环

用for <循环变量> in <遍历结构>:来遍历

1. 计数循环for I in range (N):
2. 用for in range(N,M,K) 从N到M，以K为步长
3. 用for I in c: 字符串一个一个取出
4. 用 for item in ls: 遍历列表
5. 用 for line in fi: 对文件遍历
6. 无限循环

用while <条件>:

1. 用cttl c可以退出循环
2. 用break和 continue
3. 循环与else

如果正常退出，则实现else后面的语句

1. Random库与随机数

（标准库，伪随机数）

用seed（）产生种子

用random（）产生随机数

用randint（a，b）产生一个ab间的整数

用randrange（m，n，k）

用getrandbits（k）

用uniform（a，b）

用choice（seq）从序列中选择

用shuffle（seq）把序列打乱然后返回

1. 函数

用def <函数名> （<非可选参数>,<可选参数>）

<函数体>

return <返回值>

用def <函数名> （<参数>,\*b） #不限定参数个数

<函数体>

return <返回值>

1. 局部变量和全局变量

用global转换为全局变量

1. Lambda函数(一种匿名函数)

函数名就是返回结果

例如：f=lambda x,y:x+y

1. 递归：链条和基例

函数加分支

1. 库PyInstaller

把.py的源代码转换为可直接执行的文件

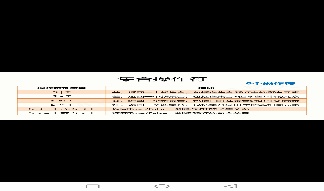


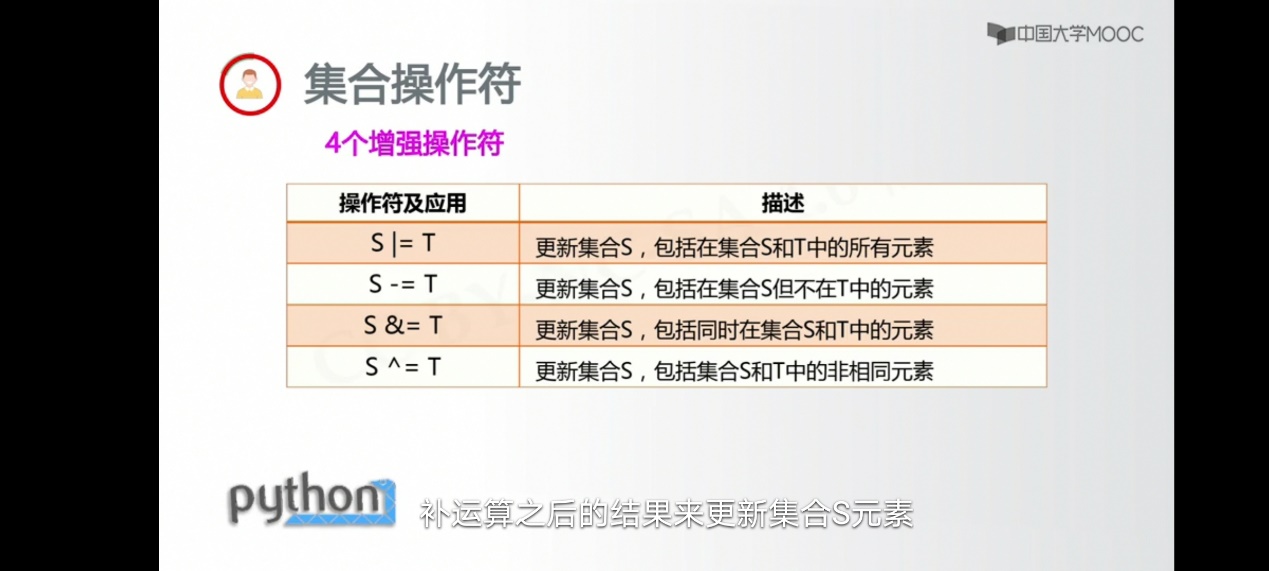
1. 组合数据类型

集合：放入集合的元素是不可以被修改的，独一无二，不存在相同元素，用{}建立集合

用set（）建立集合

集合之间的操作：六种





可以利用集合来数据去重

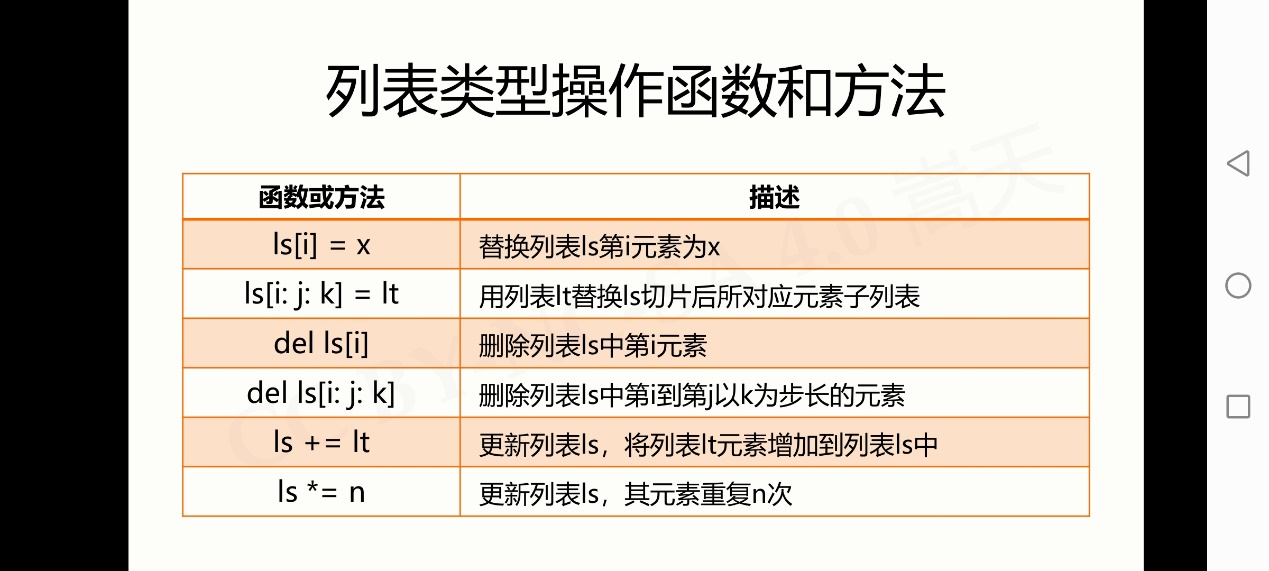




1. 序列类型（元组和列表）：具有先后关系

Ls[::-1]取反

元组：用（）或tuple（）创建，一旦创建不可以改变，有顺序，继承了序列的操作

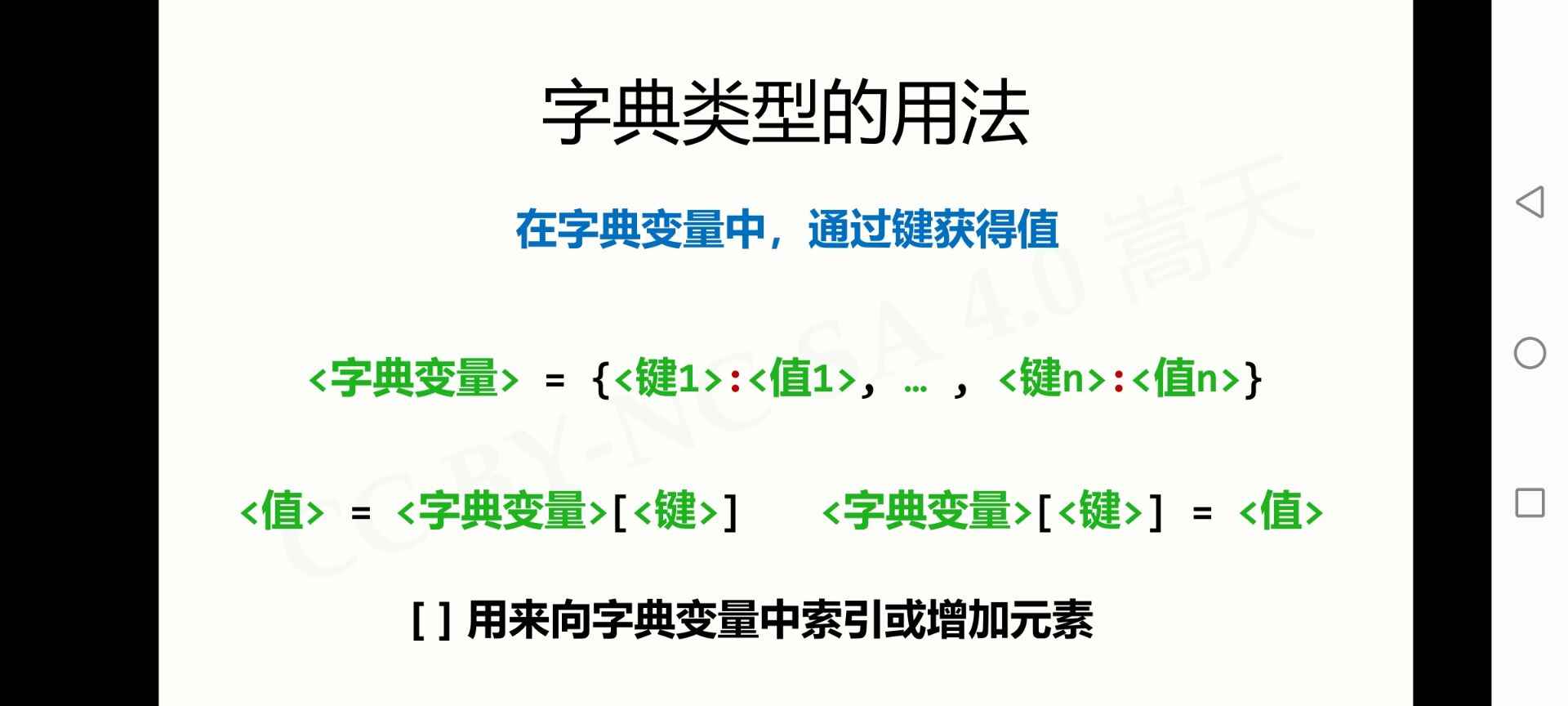
列表：序列类型，用[]或list（）来创建

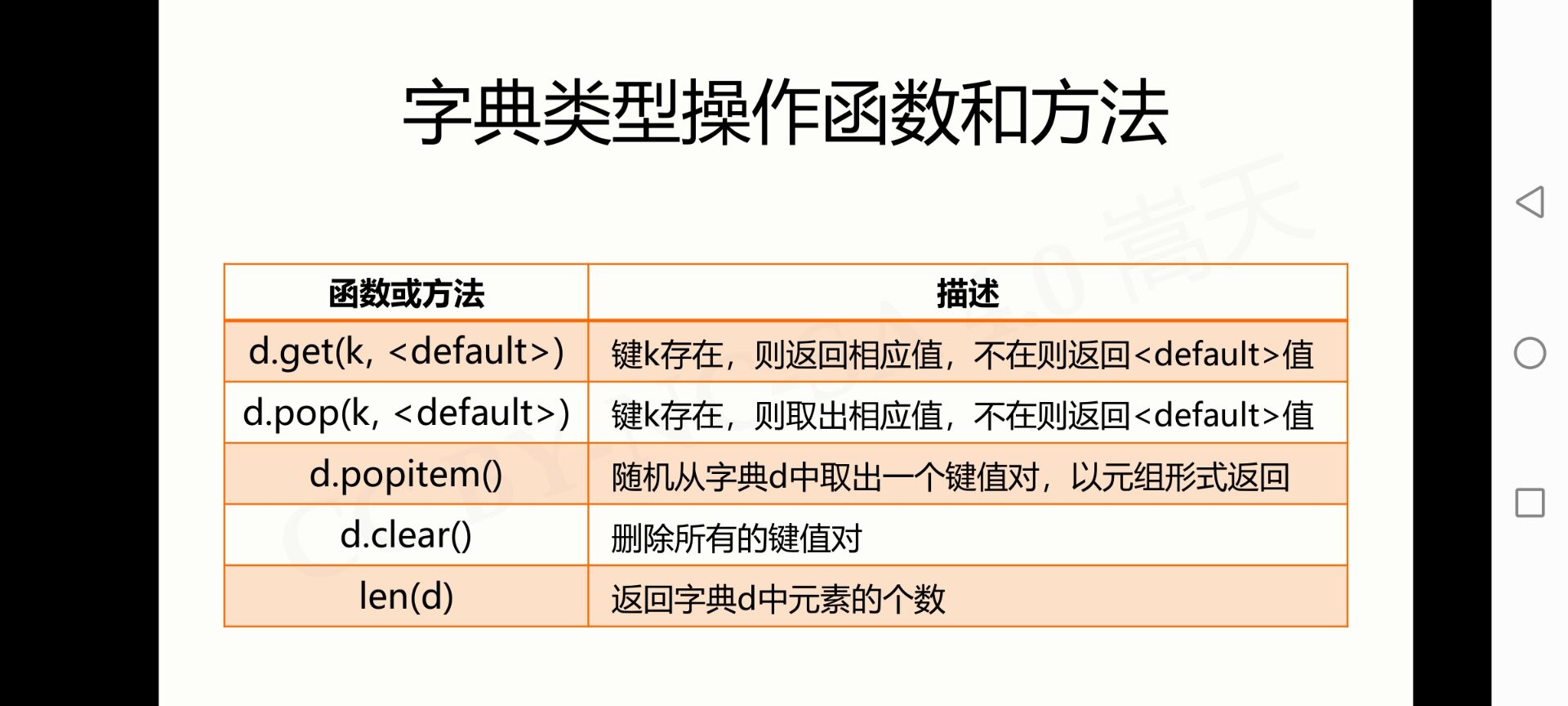
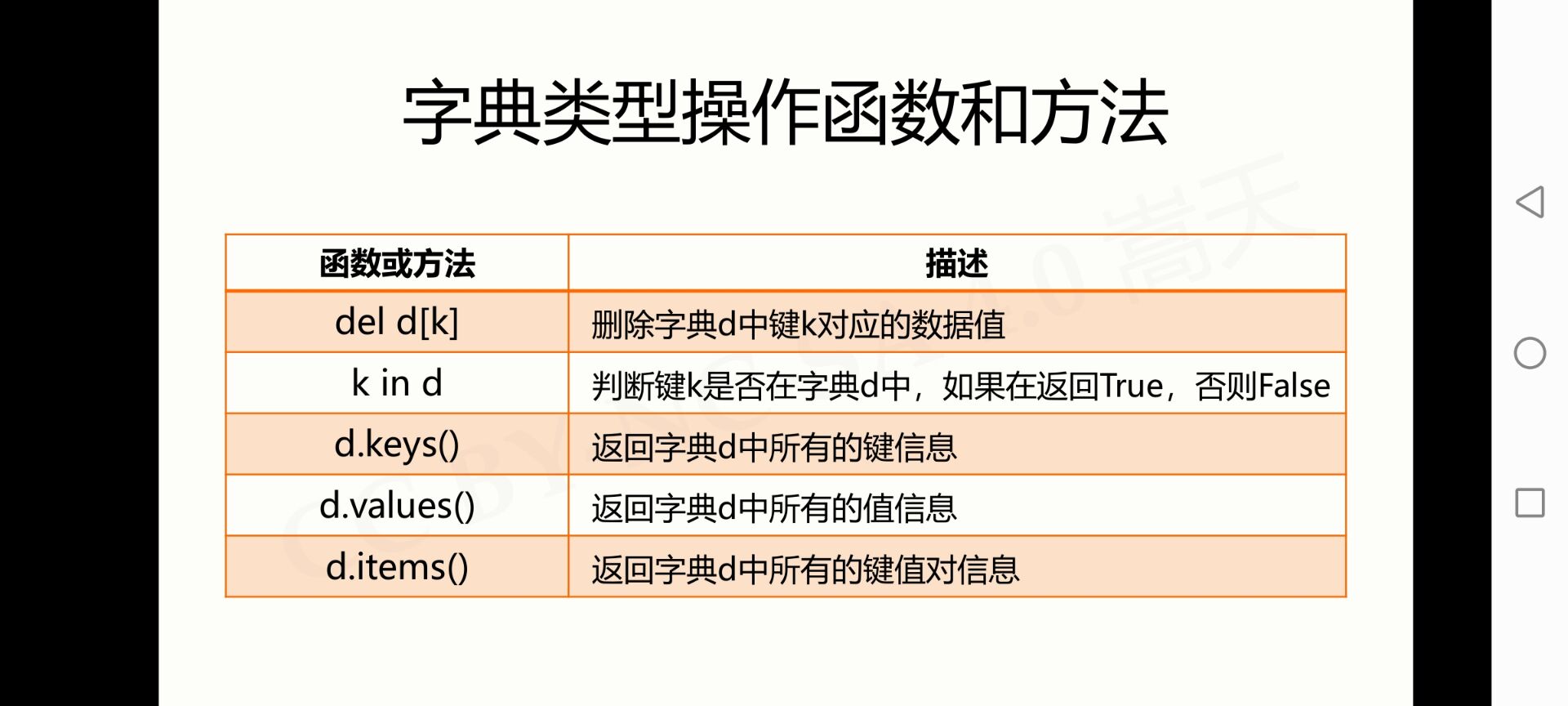
1. 字典类型

字典是键值对的集合，键值对之间无序

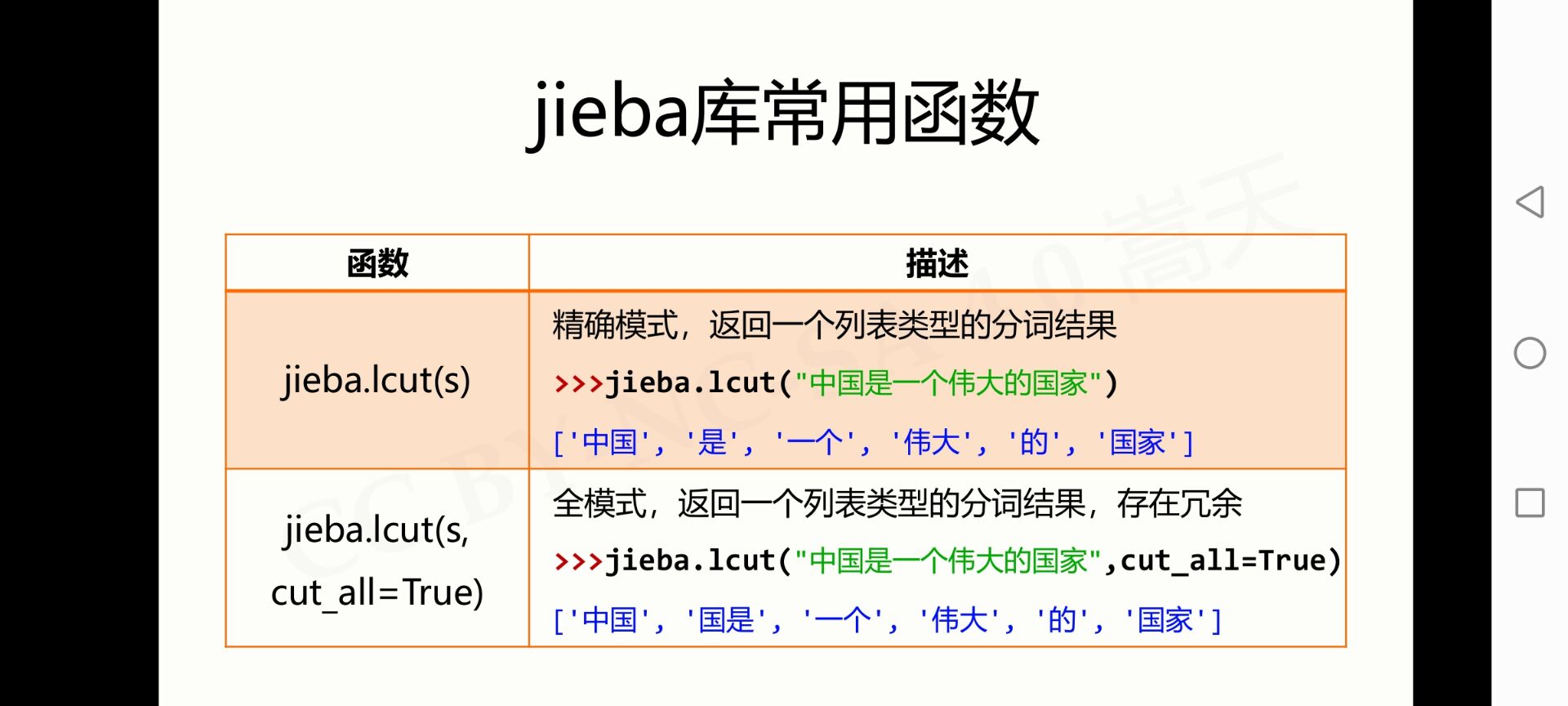
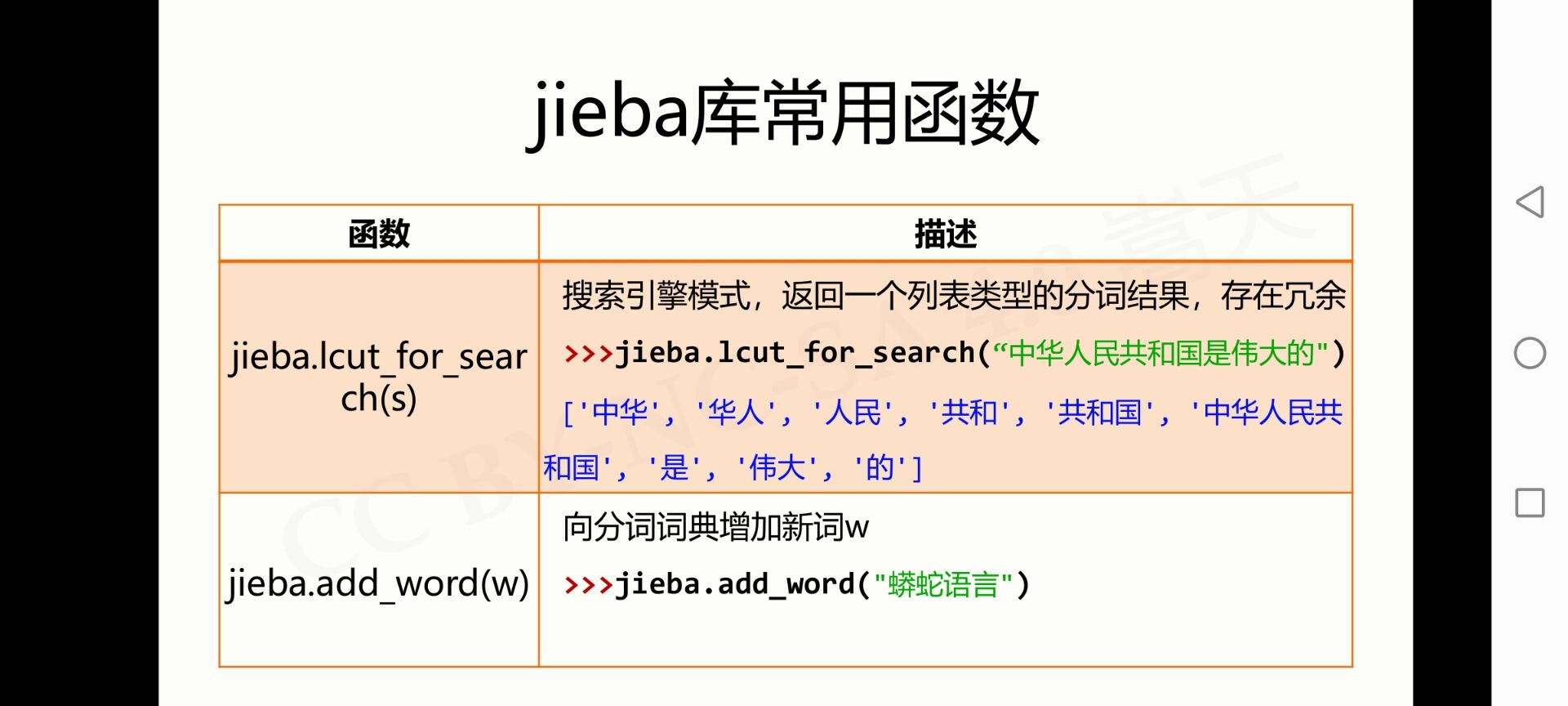
用大括号{}和dict（）创建，键值对用冒号：表示，可以通过【】获得增加键和值

空的{}保留位生成字典

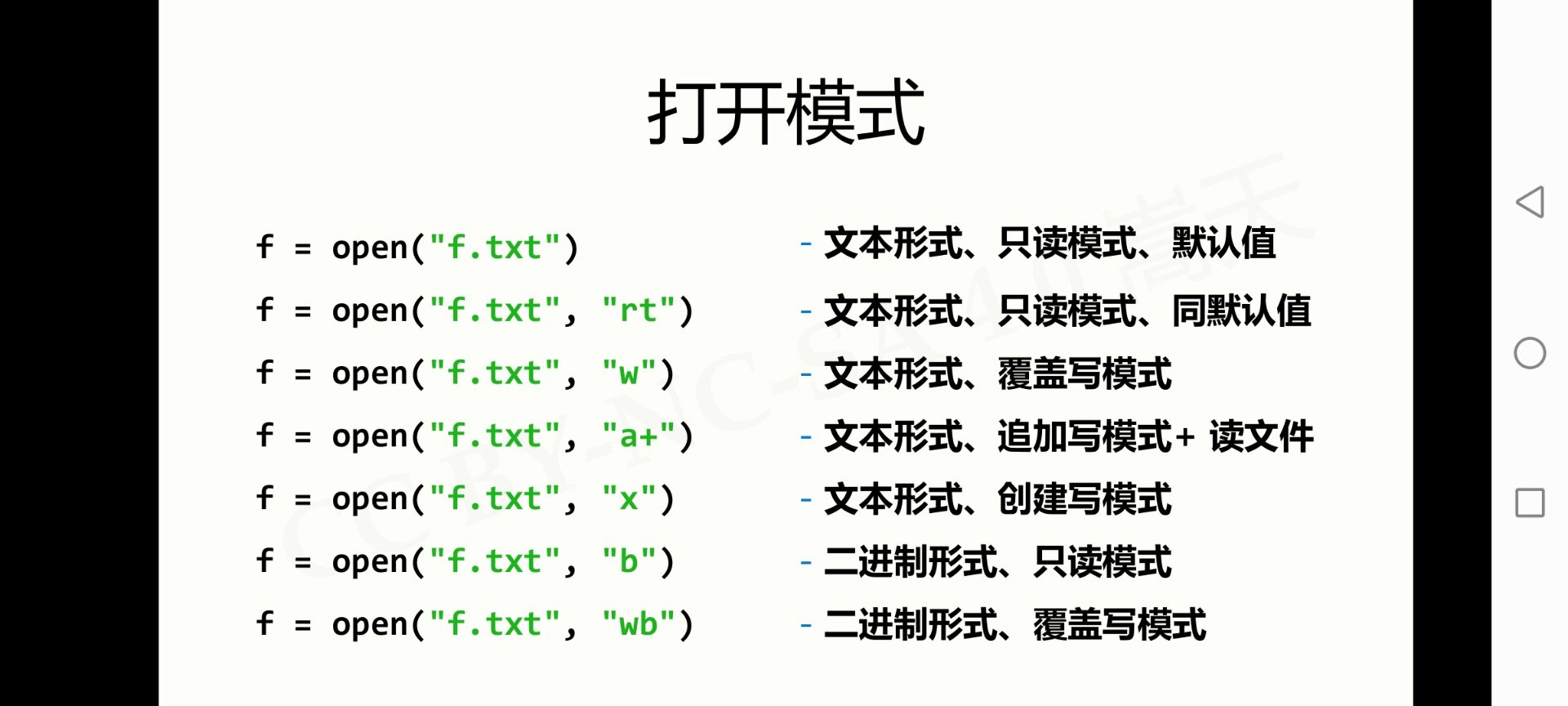
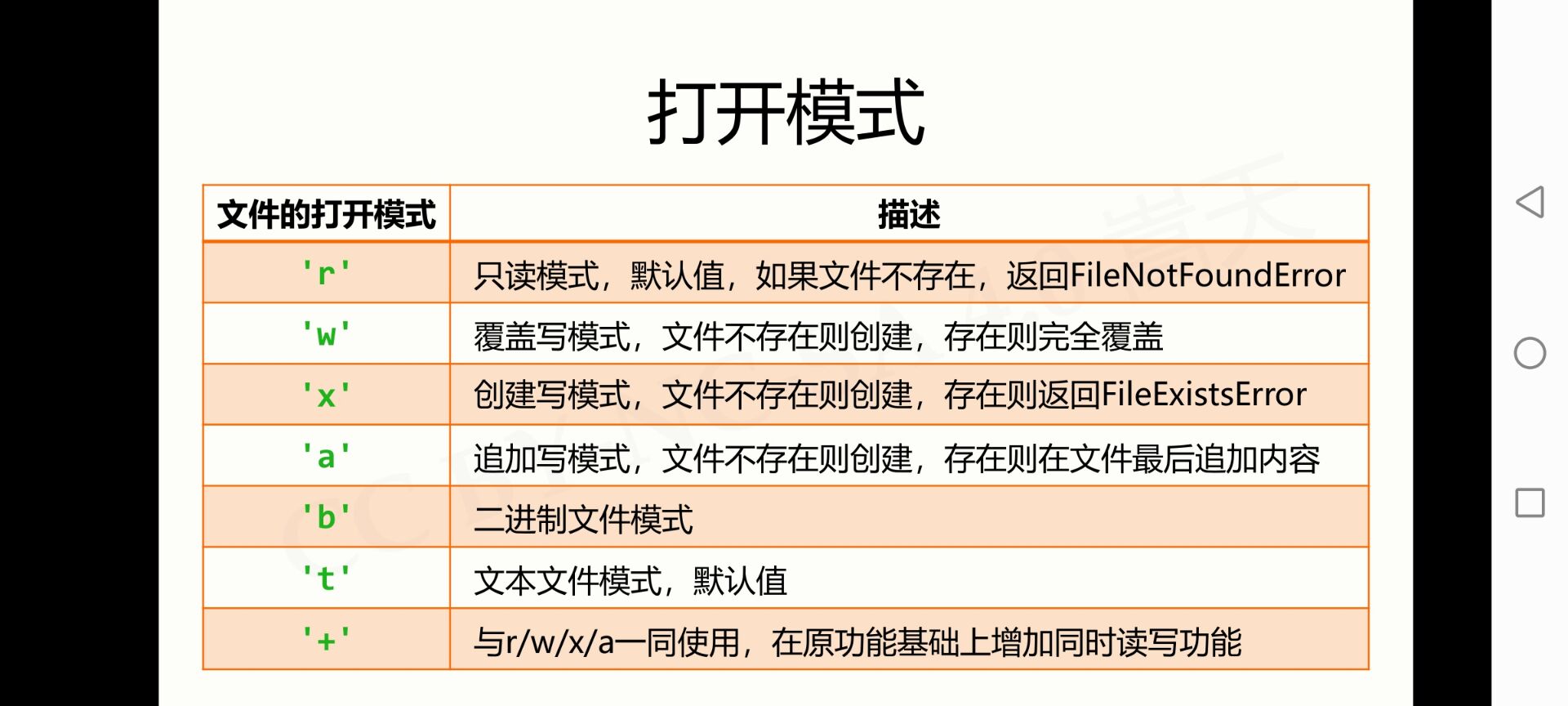
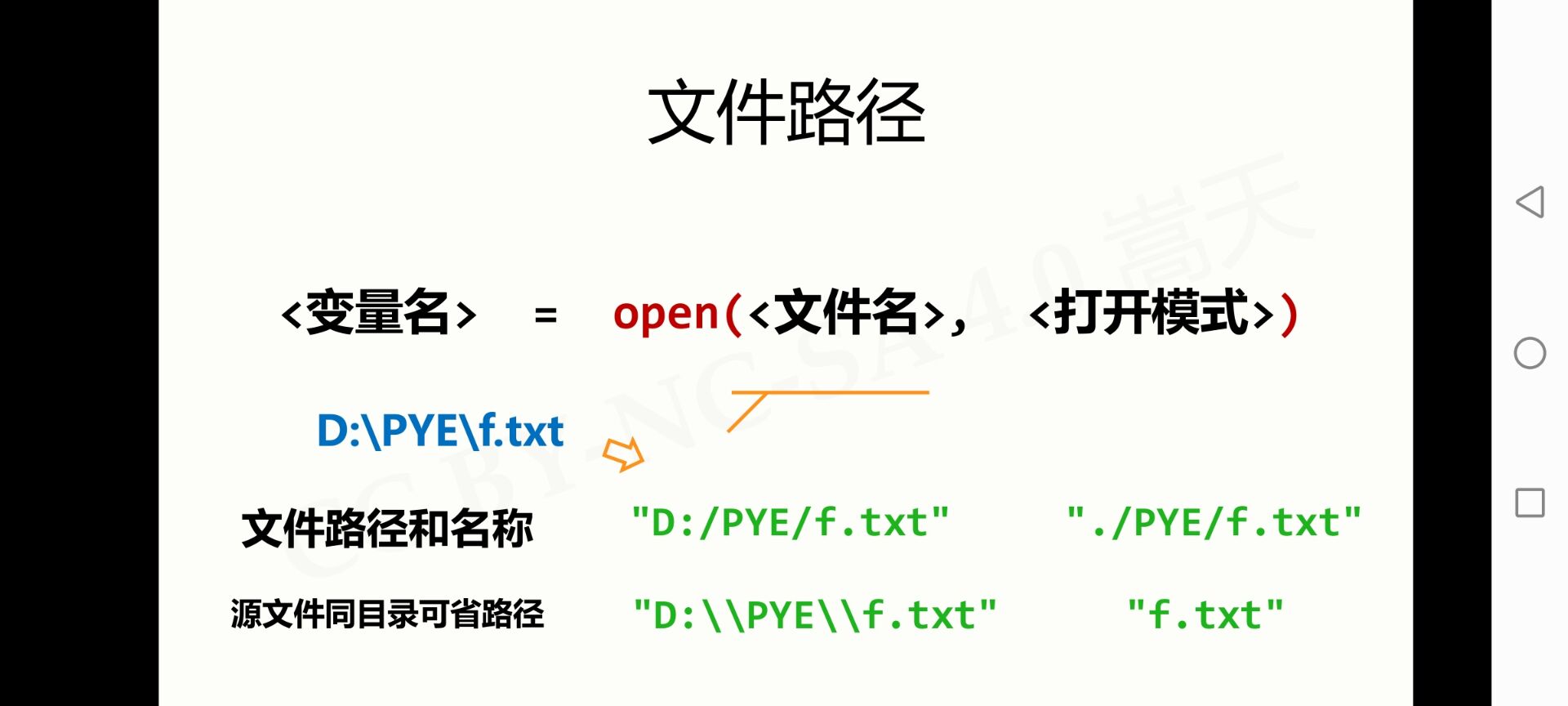
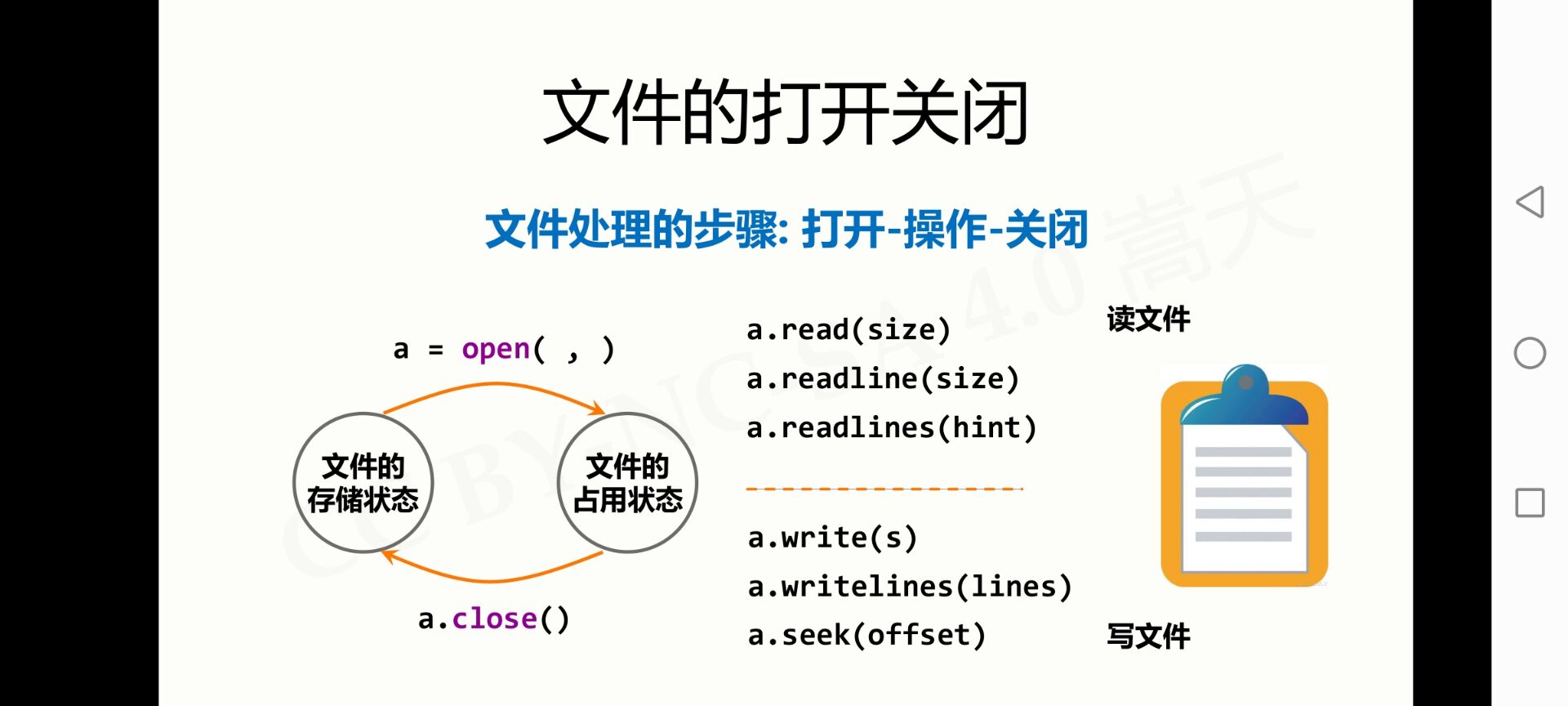




主要用于对映射的表达

1. 库jieba
2. 对文件的操作

文本文件（由单一编码 UTF-8）和二进制文件（直接由01组成）

步骤：打开，操作，关闭

1. 一维数据

如果有序：用列表

无序：用集合

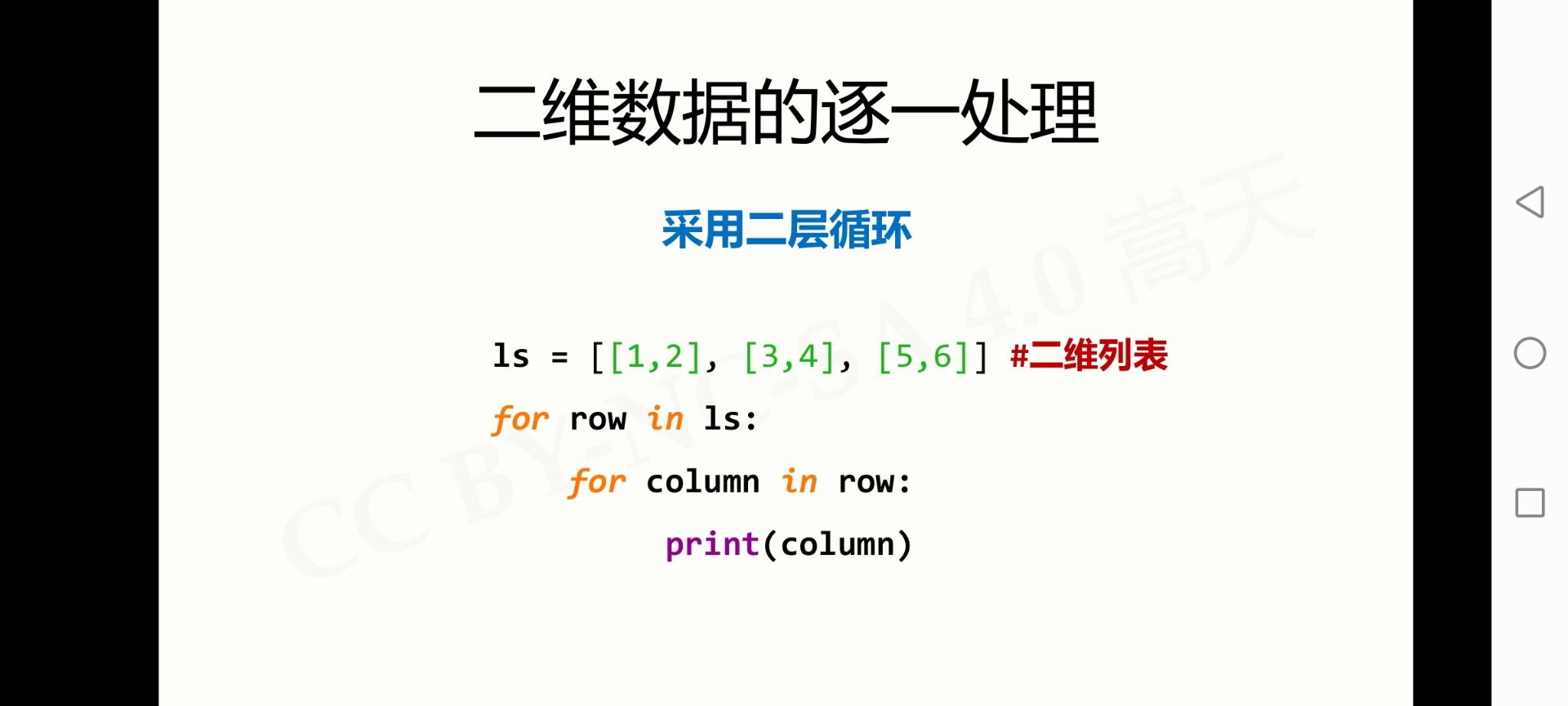
文件写入和读取：.join()和.splid()

1. 二维数据

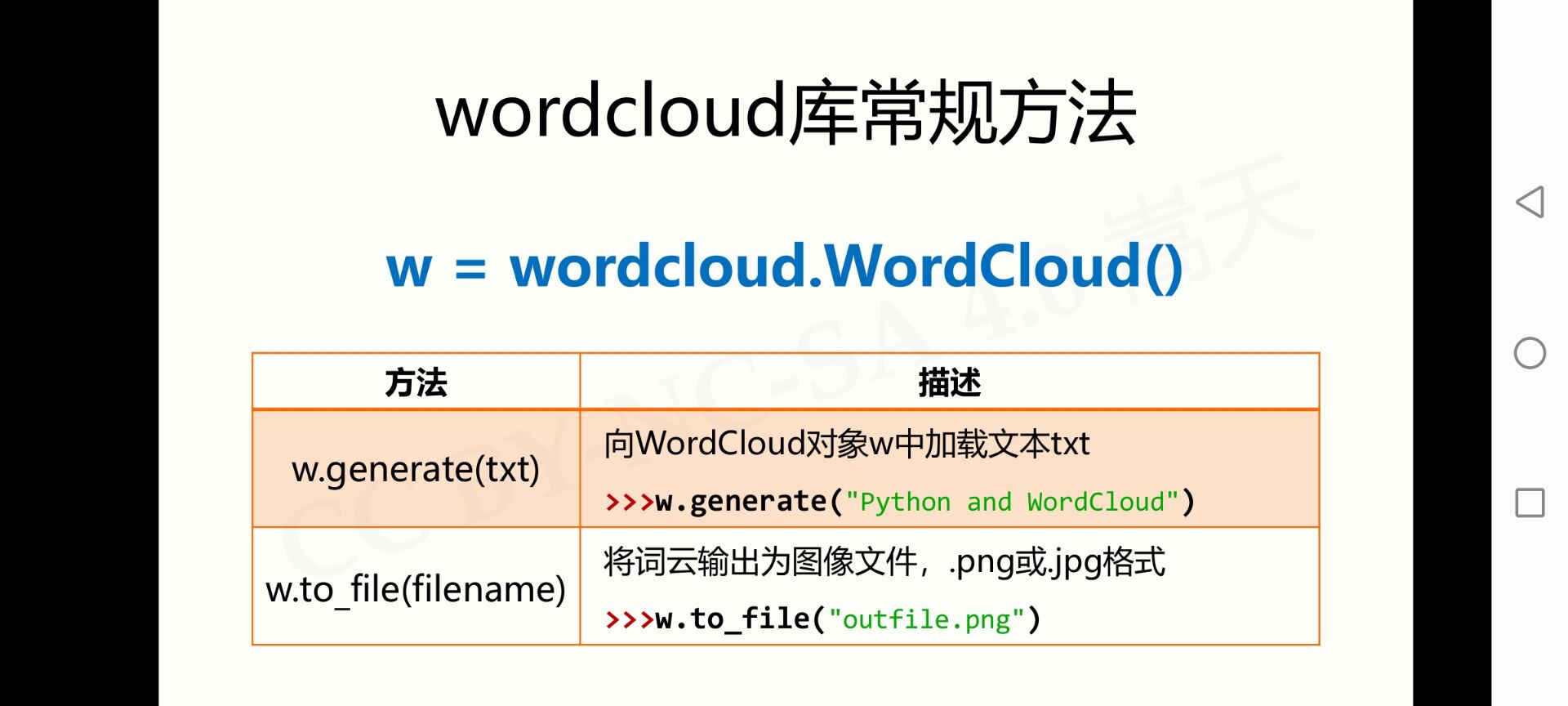
用列表形式表达（二维列表）

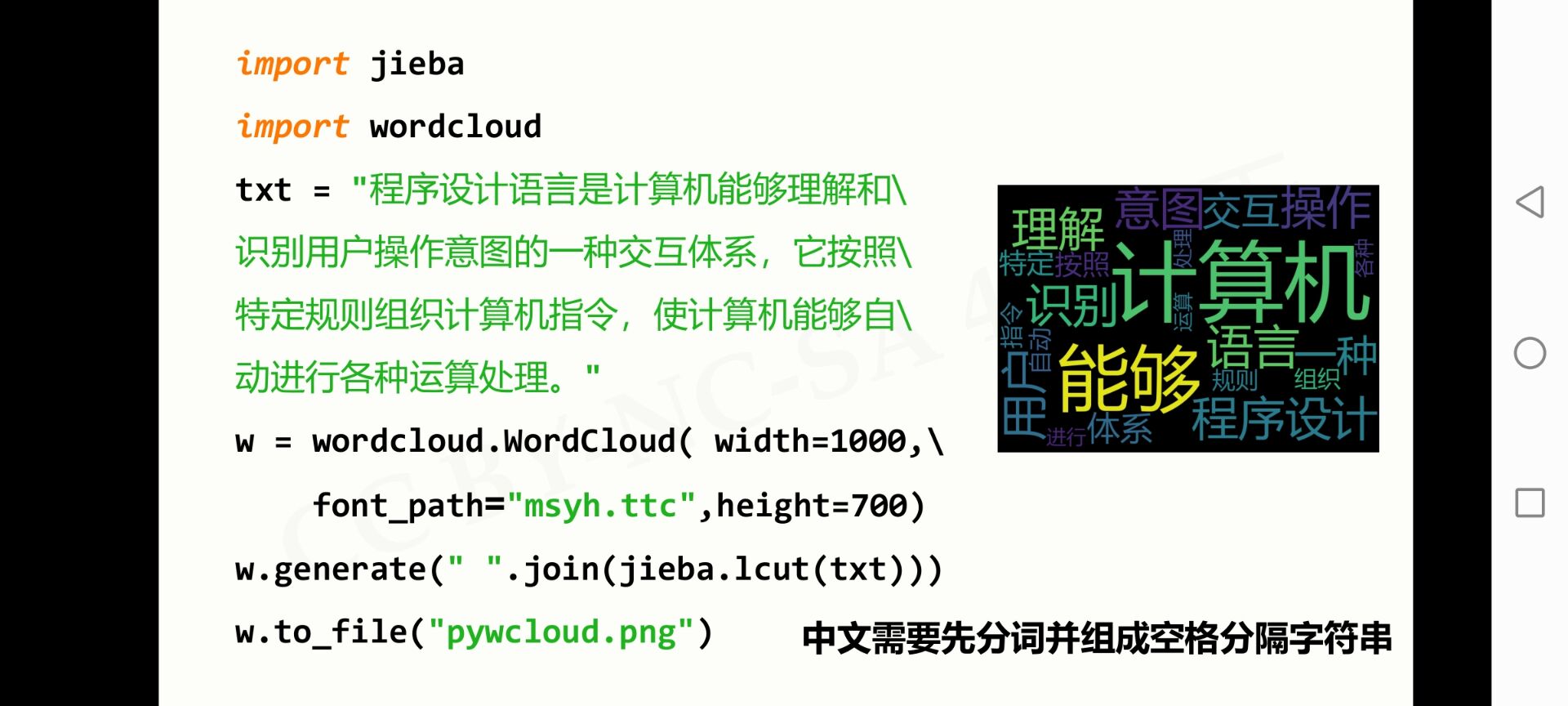
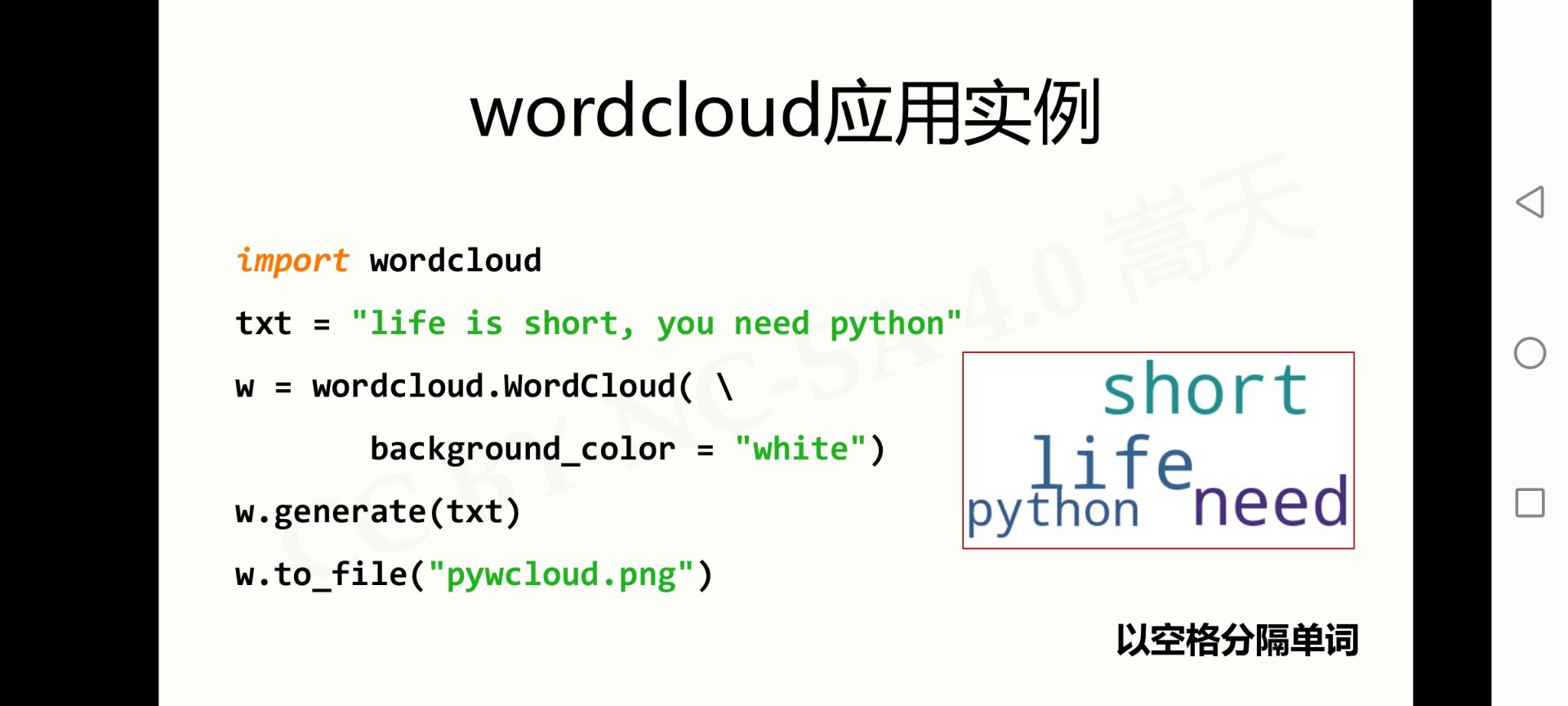
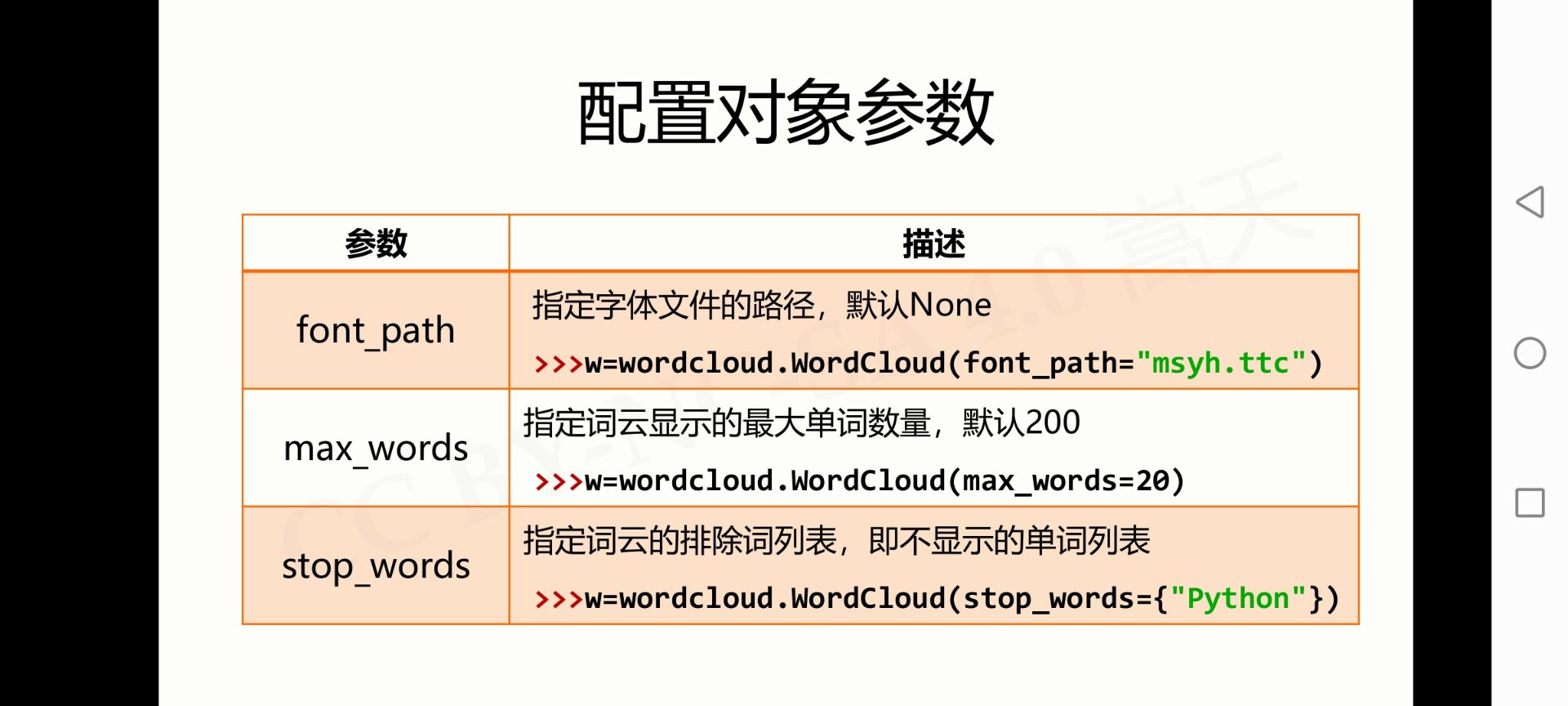
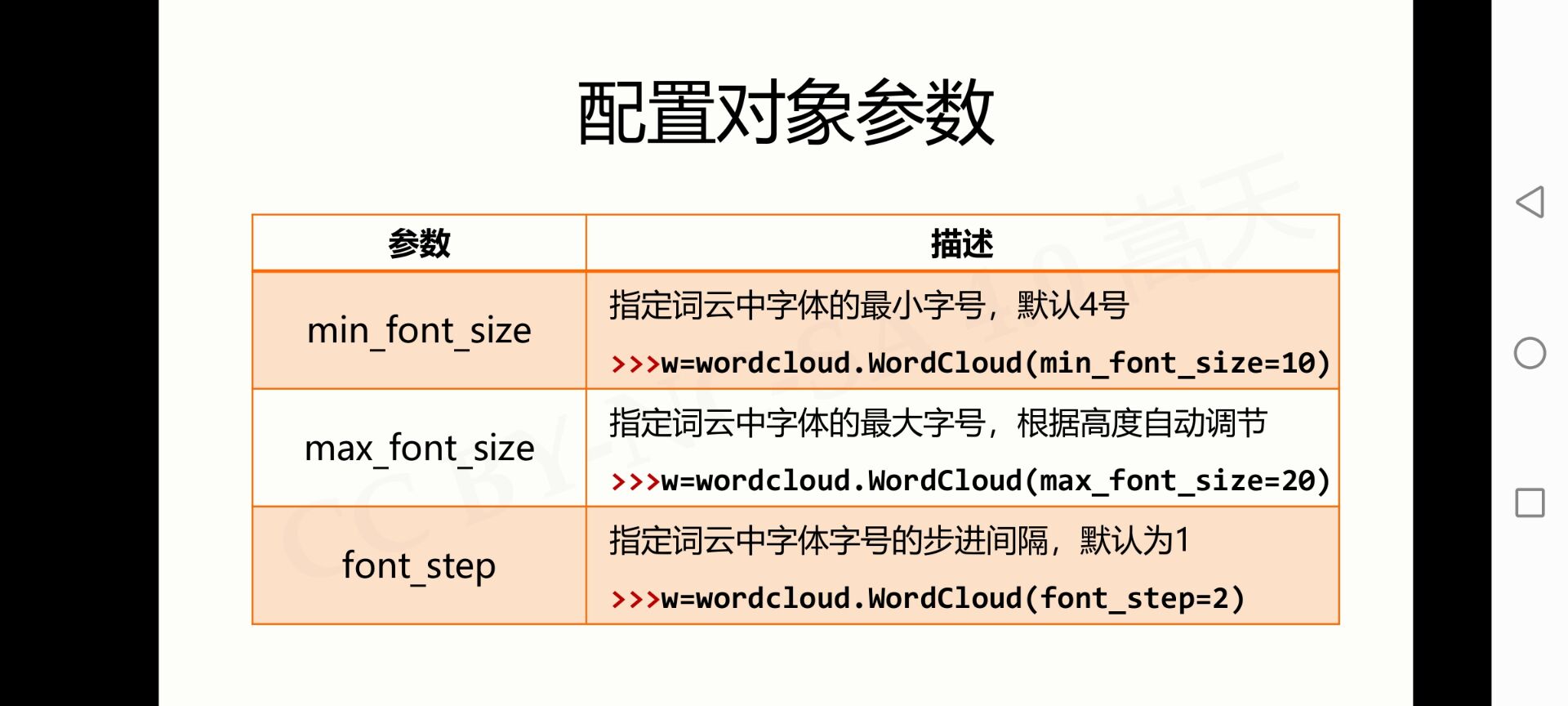
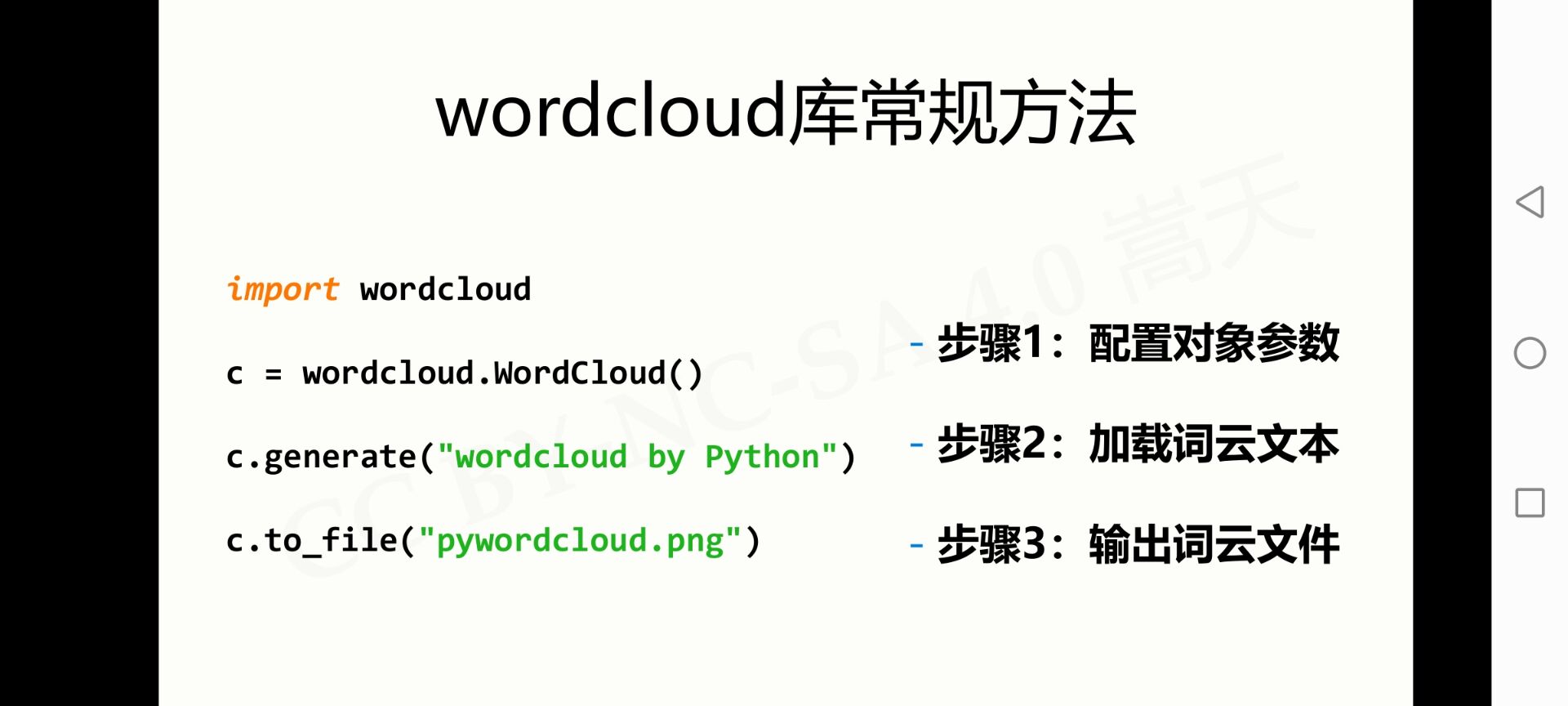
[[],[],[],[],[]]

CSV格式：用逗号分隔值，无空行，数据转换通用格式

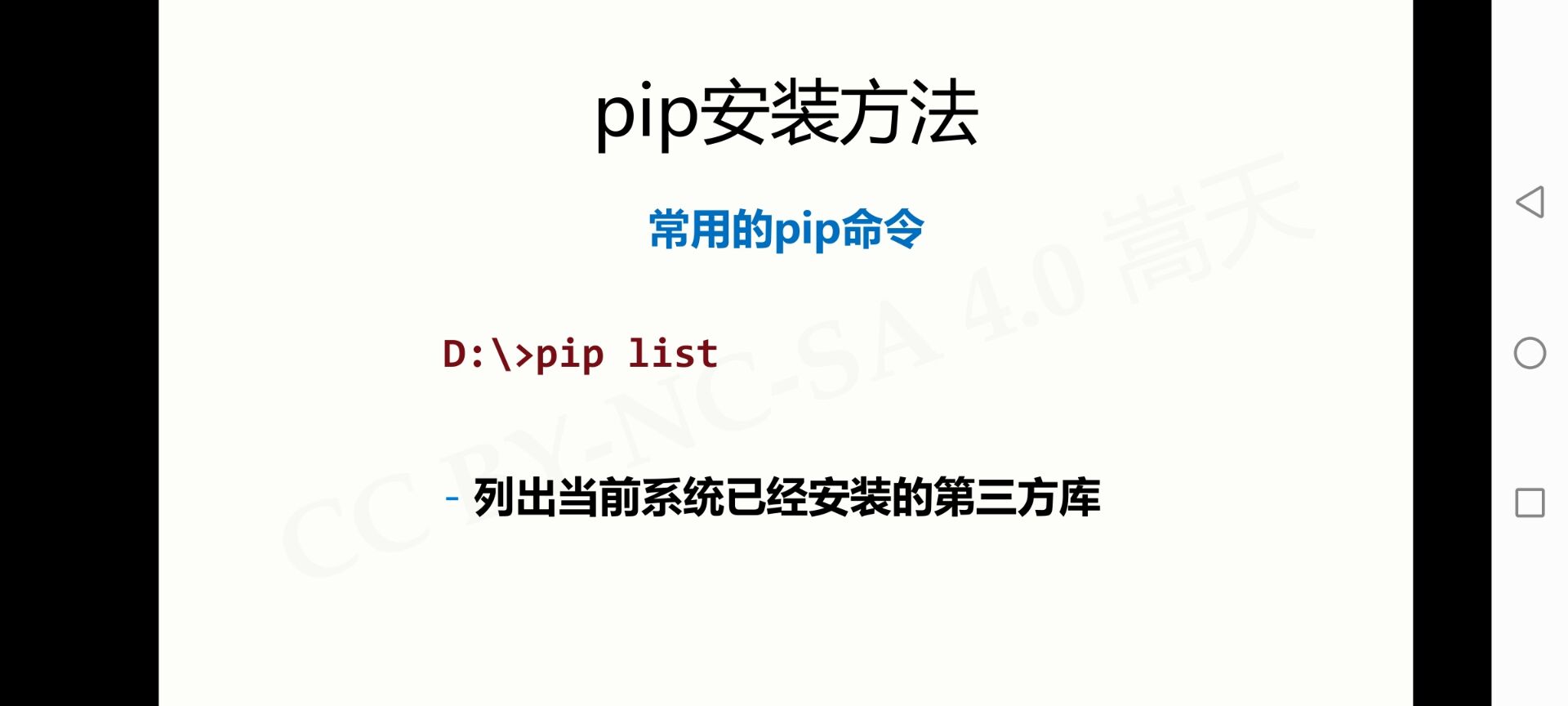
先行后列

1. 词云wordcloud库





1. 用pip安装第三方库



1. Os库的使用

