

可以整合出C++或java，它被称为胶水语言

功能强大 抢票

数据分析

游戏开发

秒处理表格

开发于1989年

①跨平台的计算机程序设计语言

②解释型语言

③交互式语言

④面向对象语言

⑤初学者最好的语言

首先是安装python解释器

www.python.org

属性中看你的操作系统的类型(下载对应的操作位数)

zip可嵌入式压缩文件

下载可执行的安装文件

简单的可以写在<<<后

复杂的必须新建代码

重新写要重新创造项目

Python模板 点出file 然后点settings 找出editor

Editor当中file and code temples 即是模板

找python script 来编写注释

# print（）

课程在5.输出函数print P6 - 10:25

将展示物质于IDIE和标准控制台

①向计算机发出指令打印XXX

②把编码编译成计算机，能听懂的继续语言

③做出相应的执行，在控制台上输出结果

A

print会输出①数字②字符串（必须加引号）③含有运算符的表达式（例如 3+1 其中3，1是操作数，+是运算符）（进行运算）

B

print（）函数还可以将内容输出的目的地

①显示器

②文件  fp=open(‘D:/text.txt’,”a+”)#a＋如果文件不存在就创建，存在就在文件内容后边继续追加

        print(‘helloworld‘,file=fp)

        fp.close()

①所指定的盘符存在②使用file=fp

C

print（）函数的输出形式

①换行

②不换行

不进行换行就用逗号分割 要如print（‘hello’,’world‘，’Python‘）

# 转义字符

6.转义字符与原字符 P7 - 00:02

含义：反斜杠+想要实现的转义功能首字母

**[\\ = \](\\\\--\\)**

http:\\www.baidu.com 就会输出http:\\www.baidu.com

**\’ = ' (使得‘成为输出内容)**

**\+转义功能的首字母：**

**换行：\n（newline）**

**回车：\r (return)**

如print(‘hello\rworld’)         在hello上输出world

**水平制表符：\t (tab)**

如print(‘hello\tworld’)     输出 hello  world，中间有四个空格的位置

而print(helloooo\tworld)  输出helloooo  world

因为一个字表位是4个字符，hell这四个字母成为一个字表位，o后\t 占用了其余三个，若是无剩余 则会新开一个字表位

**退格：\b (backspace)**

如print(‘hello\bworld’)          退格输出hellworld

\*原字符：不希望字符串中转义字符起作用，就会使用原字符，即在字符串上加上R或者r。注意最后一个字符不能是反斜线。

如print(r‘hello\nworld’)         输出hello\nworld

# 二进制与字符编码

7.二进制与字符编码 P8 - 00:01

8bit（位） — 1 byte （字节）——256种状态——256种符号

1024byte——1KB 千

1024KB——1MB 兆

1024MB——1GB 吉

1024GB——1TB 太

print(chr(0b100111001011000)——输出‘乘‘

chr为character的缩写，翻译为“汉字”；

print（ord（’乘‘）——输出十进制数

ord是order，是编码序号；

# 标识符和保留字

8.Python中的标识符和保留字 P9 - 03:16

保留字：被赋予了特殊意义，无法在对象起名中使用

如何查看

新建 keyword.demo

import（导入） keyword

print(keyword.kwist) 输出关键字列表（不需要掌握，系统会报错）

变量/函数/类/模块和其他对象 所起的名字叫做标识符

规则

①字母、数字、下划线

②不能以数字开头

③不能是保留字

④须严格区分大小写

# 变量的定义和使用

9.变量的定义和使用 P10 - 00:04

（可以改变的量）

name  =       ‘玛丽亚’

变量名  赋值运算符      值

变量要素  标识 id（name）

类型 type（name）

值  name

name里面存放的是id

变量的多次赋值指向新的空间

name=‘玛丽亚’（成为内存垃圾）

name=‘楚留冰’（指向新的空间）

print（name）

程序输出楚留冰

name=‘玛丽亚’

print（name）

name=‘楚留冰’

print（name）

程序输出:

玛丽亚

楚留冰

# 常见的数据类型

11.Python中常见的数据类型 P12 - 00:02

变量三部分

标识+类型+值

常用数据类型

整数类型—int ——98

浮点数类型——float ——3.1415926

布尔类型——bool——True 、False

字符串类型——str——人生苦短，’我用Python’（加上引号就是）

# 整数类型

12.数据类型-整数类型 P13 - 00:09

﻿英文为为integer，简写为int

可以表示正数、负数和零

整数不同进制的表示方式

十进制——默认

二进制——以0b开头

八进制——以0o开头

十六进制——以0x开头

n1=90

n2=-76

n3=0

print(n1,type(n1))

print(n2,type(n2))

print(n3,type(n3))

输出

90 <class, ‘int’>

-76 <class, ‘int’>

0 <class, ‘int’>

print（118）

print（0b10101111）

print（0o176）

print（0x1EAF）

输出 118

    175

126

7855

# 浮点类型

13.数据类型\_浮点类型 P14 - 00:01

﻿浮点数由整数部分和小数部分组成.

浮点数储存具有不精确性质.

使用浮点数计算时，可以既可以出现小数位数的不精确的情况

a=314159

print（a，type(a)）

输出

3.14159<class, ‘float’>

n1=1.1

n2=2.2

n3=2.1

print(n1+n2)

print(n1+n3)

输出 3.3000000000003

     3.2

因为计算机是二进制存储方式，底层问题，所以存储浮点数会有误差。解决方案：导入模块decimal

from decimal import Decimal

print(Decimal(‘1.1’) +Decimal(‘2.2’))

输出3.3

# 布尔类型

bool（全称Boolean）

14.数据类型\_布尔类型 P15 - 00:04

﻿用来表示真或假的值

Ture是真 False是假

布尔值可以转化为整数

Ture——1

False——0

f1=True

f2=False

print（f1，type（f1））

print（f2，type（f2））

输出True <class ‘bool’>

False <class ‘bool’>

print（f1+1）

print（f2+1）

输出 2

  1

# 字符串类型

15.数据类型\_字符串类型 P16 - 00:05

**①字符串又被称为不可变的字符序列**

**②可以使用单引号，双引号，三引号（‘’‘或者“”“）来定义**

**③单引号和双引号定义的字符串必须在同一行**

**④三引号定义的字符串，可以分布在连续的多行**

str1= ‘人生苦短 我用Python’

str2= “人生苦短 我用Python”

str3= ‘‘‘人生苦短

我用Python’’’

str1= “““人生苦短

我用Python”””

输出

人生苦短 我用Python

人生苦短 我用Python

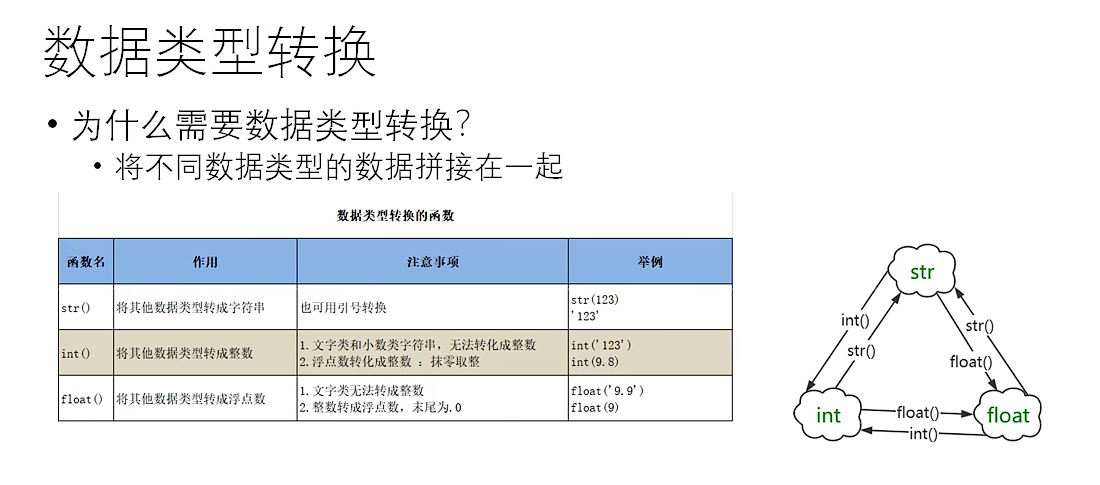
人生苦短

我用Python

人生苦短

我用Python

# 类型转换



str（）函数——int（）函数

16.类型转换\_str()函数与int()函数 P17 - 00:09

name=‘ 张三’

age=20

print（type（name），type（age））

输出：<class ‘str’> <class ‘int’>

补加：print（‘我叫’+name+‘今年’+age+‘岁’）

报错 （str无法与int连接）

修正 使用类型转换

print（‘我叫’+name+‘今年’+str（age）+‘岁’）

（将int函数转为str函数）

输出

我叫张三今年20岁

## str（）函数

a=10

b=198.8

c=False

print（type（a），type（b），type（c））

<class ‘int’><class ‘float’> <class ‘bool’>

print（str(a),str(b),str(c),type（a），type（b），type（c））

输出

10 198.8 False <class ‘str’> <class ‘str’> <class ‘str’>

## int（）函数

s1=’128’

s2=’76.7’

s3=’hello’

f1=98.7

ff=True

print（int(s1)，type(int(s1)））

输出

128 \*str—int 字符串必须为整数串才不会报错

print（int(f1)，type(int(f1)））

输出98 \*float—int 截取整数部分

print（int(s2)，type(int(s2)））

报错 \*str--int 如果字符串是小数串便无法转换成int

print（int(s3)，type(int(s3)））

报错  \*str—int 字符串如果是字符，报错。 因此字符串必须是整数串才能转换

print（int(ff)，type(int(ff))）

1 \*bool—int

## float（）函数

17.类型转换\_float()函数 P18 - 00:03

将其他数据转成浮点数

①文字类无法转成整数

②整数转成浮点数，末尾.0

s1=’128.98’

s2=’76’

ff=True

s3=’hello’

i=98

print（type（s1），type（s2），type（ff）, type（s3），type（i））

输出

<class ‘str’> <class ‘str’> <class ‘bool’> <class ‘str’> <class ‘int’>

print（float(s1)，type(float (s1)））

128.98<class ‘float’>

print（float(s2)，type(float (s2)））

76.0<class ‘float’>

print（float(ff)，type(float (ff)））

1.0<class ‘float’>

print（float(s3)，type(float (s3)））

报错 字符串中的数据如果是非数字串，则不允许转换

print（float(i)，type(float (i)））

98.0

# Python中的注释

18.Python中的注释 P19 - 00:05

增强可读性且会被Python解释器忽略

含有三种类型的注释

①单行注释——以＃开头，直到换行结束 #输入功能（单行注释）

②多行注释——一对三引号代码 多行注释

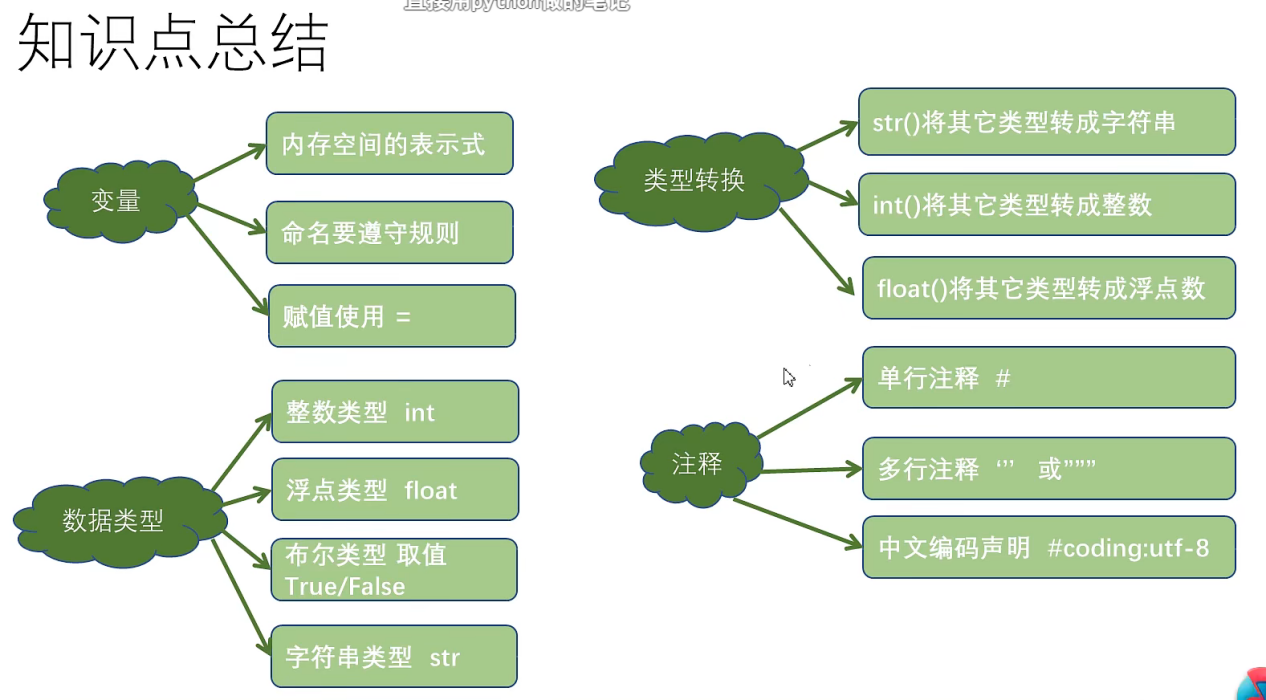
’’’嘿嘿嘿，

我是多行注释

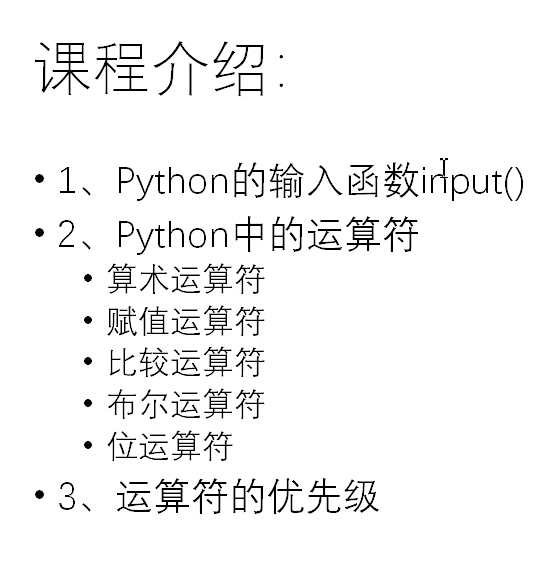
噢’’’

③中文编码声明注释——在文件开头 以指定源码文件的编码格式#coding:gbk

# 第二章总结



# 第三章 算你赢



# input函数的使用

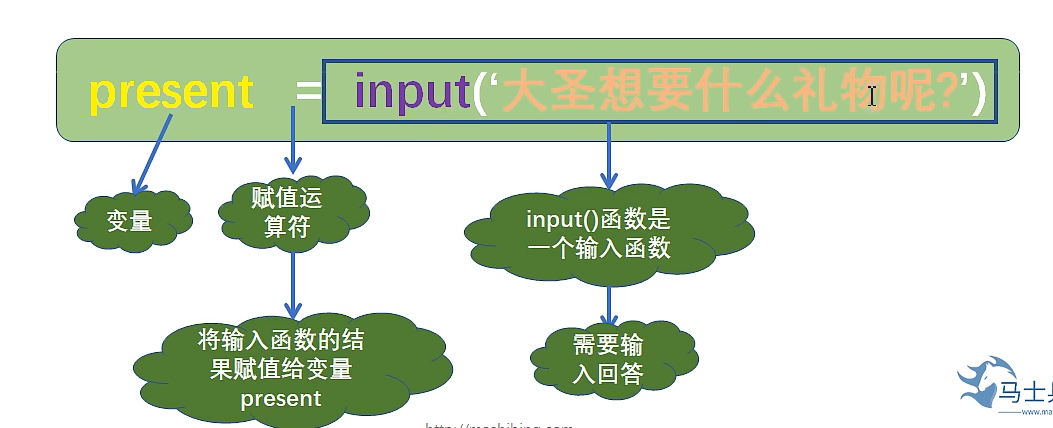
19.input函数的使用 P20 - 00:04

作用——接收来自用户的输入

返回值类型——输入值的类型为str

值的存储——使用=对输入值进行存储

赋值运算符——将输出的结果赋给变量



present=input（‘大圣想要什么礼物呢’）

print（present）

输出 大圣想要什么礼物呢

在输出结果后回答定海神针

自动回答 大圣想要什么礼物呢

present=input（‘大圣想要什么礼物呢’）

print（present，type(present))

重复上一回答

输出

大圣想要什么礼物呢

定海神针 <class 'str'>

# input（）函数的高级使用

20.input函数的高级使用 P21 - 00:05

a=input（‘请输入一个加数：’）

b=input（‘请输入另一个加数’）

print（type（a),type（b))

print（a+b）

请输入一个加数：10

请输入另一个加数：20

<class,’str’> <class ‘str’> #在获取数字时，数字格式为字符串

1020

a=int（input（‘请输入一个加数：’））#在获取数字时，格式为字符串，并将它转化为整形

b=int（input（‘请输入另一个加数’））

print（type（a),type（b))

print（a+b）

请输入一个加数：10

请输入另一个加数：20

<class,’int’> <class ‘int’>

30

# 运算符算术运算符



21.运算符\_算术运算符 P22 - 00:27

分类为

标准算术运算符 + 取余运算符 + 幂运算符

(+、-、\*、/ 整除//） % \*\*

基本运算符 （输出值用绿色）

print（1+1）2

print（1-1）0

print（2\*4）8

print（11/2）5.5

print（11//2）5（整除取整）

print（11%2）1 （取余）

print（2\*\*3）8（2\*2\*2） #幂运算

一正一负 向下取整

print（9//4）2

print（-9//-4）2

print（9//-4）-3 即为后文的商

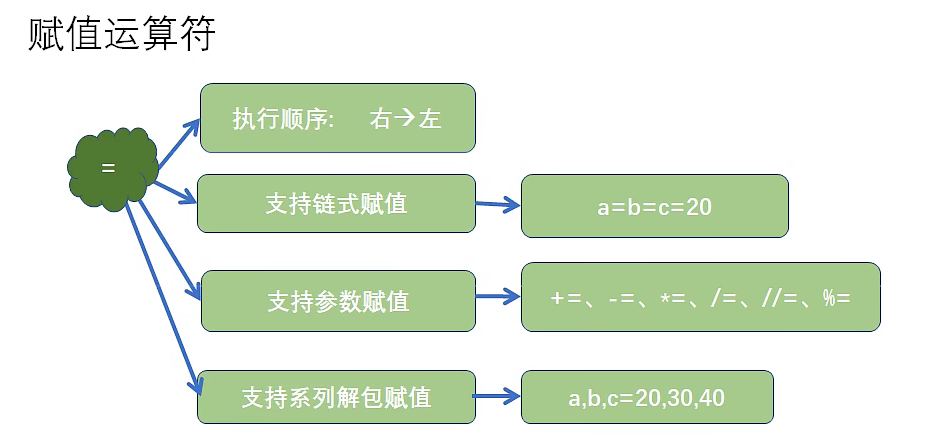
print（-9//4）-3 就是说在一正一负的除法运算之后，有余数就向下取整

print（9%4）1

print（9%-4）-3 （余数=被除数-除数\*商）9-（-4）\*（-3）=-3

print（-9%4）3 -9-4\*（-3）=3

# 赋值运算符



22.运算符\_赋值运算符 P23 - 00:10

赋值运算符 运算顺序从左到右

i=3+4

print（i）

7

支持链接赋值

a=b=c=20

print(a,id(i),b,id(b),c,id(c))

20 14645450672

20 14645450672

20 14645450672

支持参数赋值

+=、-=、\*=、/=、//=、%=

a=20

a+=30

print（a）

50

a-=10

print（a）40

a\*=2

print（a）

print(type(a)) 80 <class 'int' >

a/=3

print（a）

print(type(a)) 26.66666666668 <class 'float'>

a//=2

print(a)

print(type(a)) 13.0 <class 'float' >

a%=3

print(a) 1.0

支持系列解包赋值

a,b,c=20,30,40

print(a,b,c)

20 30 40

a,b=20,30,40 ##必须变量和赋值数个数一致

print(a,b)

报错

交换两个变量

a,b=10,20

print(‘交换之前:’,'a,b')

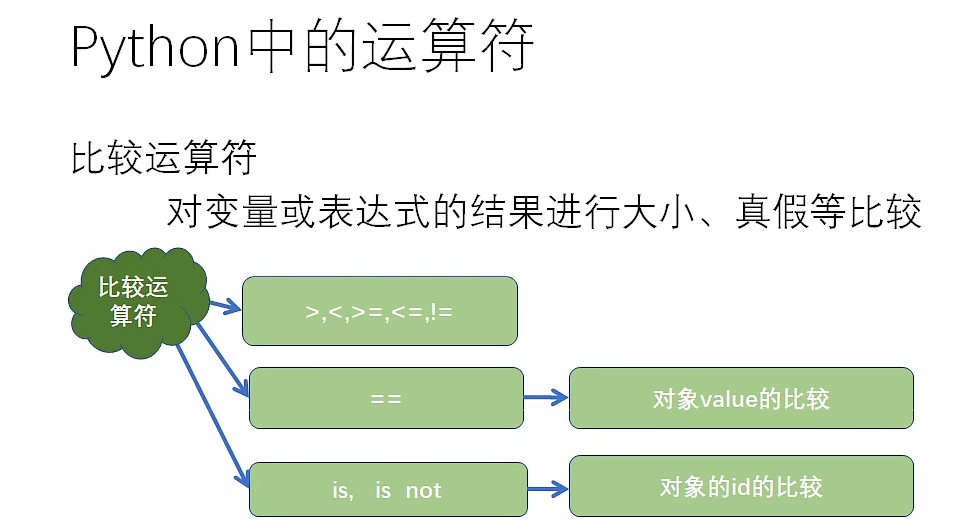
a,b=b,a

print('交换之后:‘,'a,b')

交换之前：10，20

交换之后：20，10

# 比较运算符



23.运算符\_比较运算符 P24 - 00:11

①>, <,>=,<=,!=

②= = 对象value的比较

③is,is not 对象id的比较

比较的结果是布尔类型

a,b=10,20

print('a>b吗?',a>b)

print('a<b吗?',a<b)

print('a<=b吗?',a<=b)

print('a>=b吗?',a>=b)

print('a==b吗?',a==b)

print('a!=b吗?',a!=b)

a>b吗？False

a<b吗？True

a<=b吗？True

a>=b吗？False

a==b吗？False

a!=b吗？True

=成为赋值运算符 ==比较运算符

变量= 标识+类型+值

== 对象value的比较

is,is not 对象id的比较

a=10

b=10 #说明a与b是指向的同一个id和值

print(a==b)

print（a is b）

True #a与b的value相等

True #a与b的id标识相等

lst1 [11,22,33,44]

lst2 [11,22,33,44]

print(lst1==lst2)

print(lst1 is lst2)

print(id(lst1))

print(id(lst2))

print(a is not b)

print(lst1 is not lst2)

True

False

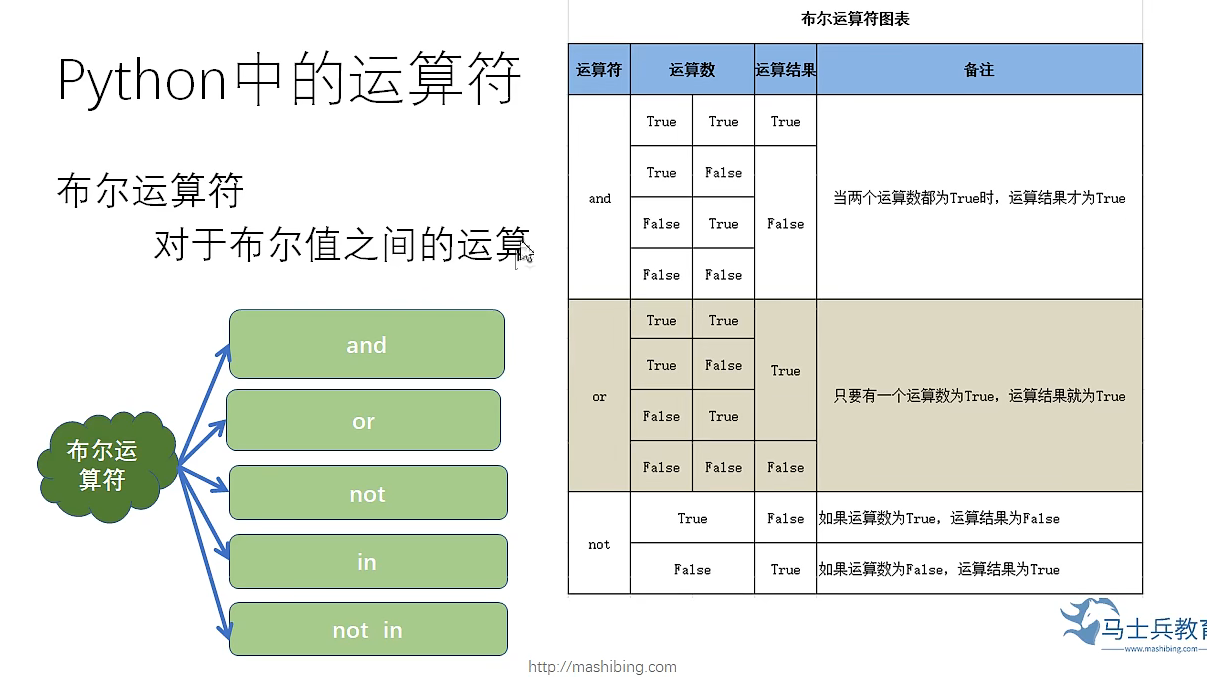
2525441564

2525665484

False

True

# 布尔运算符



24.运算符\_布尔运算符 P25 - 00:24

and/or/not/in/not in

**and**

a,b=1,2

print(a==1 and b==2) \*True and True=True

True

print(a==1 and b<2) \*False and True=False

False

print(a!=1 and b==2) \*False and True=False

False

print(a!=1 and b!=2) \*False and False=False

False

**or**

print(a==1 or b==2) \*True orTrue=True

True

print(a==1 or b<2) \*False or True=True

False

print(a!=1 or b==2) \*False or True=True

False

print(a!=1 and b!=2) \*False or False=False

False

**not(针对布尔类型)** \*对布尔类型操作数取反

f=True

f2=False

print(not f) False

print(not f2) True

**in/not in** \*在不在字符串中

s='helloworld'

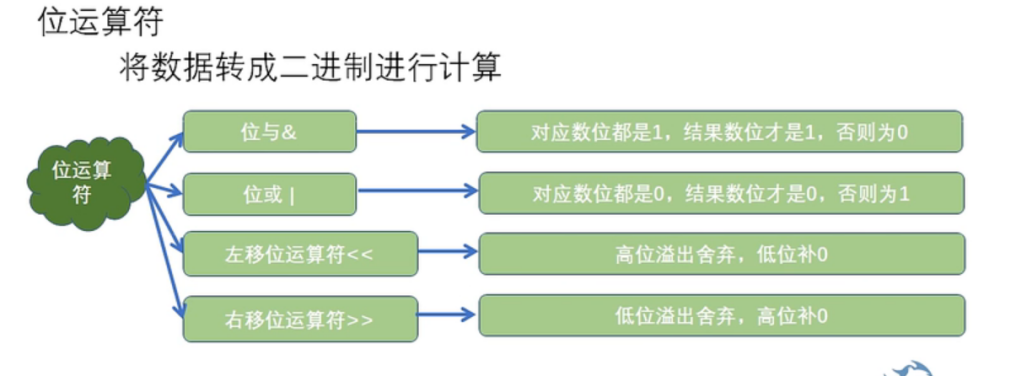
print('w' in s ) True

print('k' in s ) False

print('w' not in s) False

print('k'not in s) True

# 位运算符号



25.运算符\_位运算 P26 - 00:11

将数据转成二进制进行计算

**位与&（对应数位都是1，结果数位才是1，否则为0）**

4的二进制 00000100

8的二进制 00001000

4&8 00000000 —— 0

print（4&8） 0

**位或 |（对应数位都是0，结果数位才是0，否则为1）**

4的二进制 00000100

8的二进制 00001000

4|8 00001100 —12

print（4|8） 12

**左移位运算符<< （高位溢出舍弃，低位补0)**

4的二进制 00000100

左移位 000001000 8 （向左移动以为相等于\*2）

**右移位运算符>> （低位截断，高位补0)**

4的二进制 00000100

右移位 000001000 2（向右移动以为相等于/2）

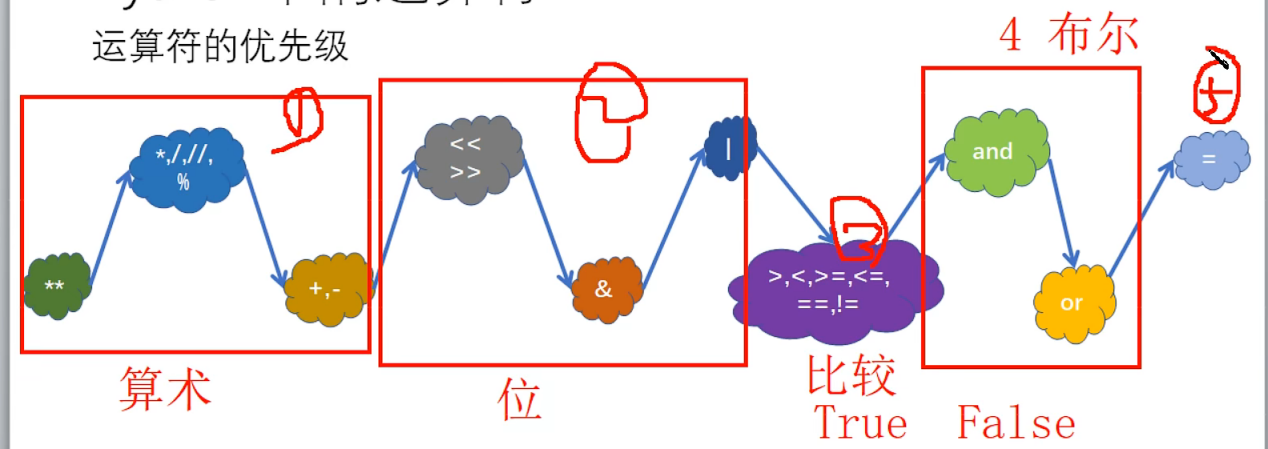
print (4<<1）

print (4<<2)

8

16

# 运算符的顺序



26.运算符的优先级 P27 - 03:53

①\*\* A 赋值运算符

②\*、/、//、%

③+、-

④<<、>> B 位运算符

⑤&

⑥|

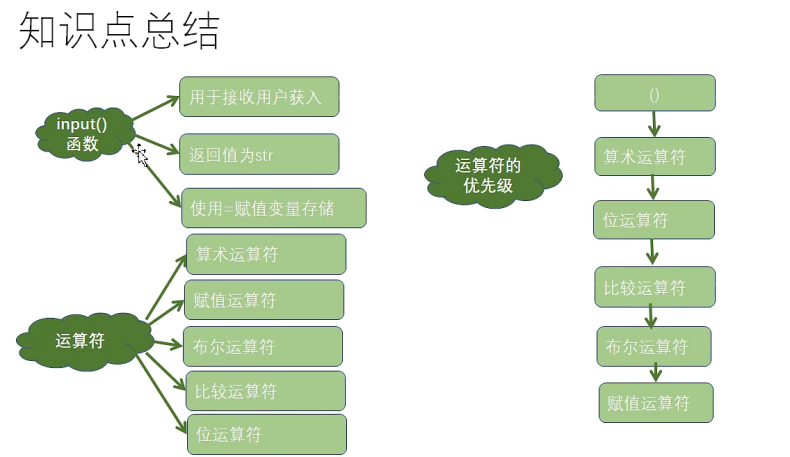
⑦>、< 、>=、<=、==、！= C 比较运算符

⑧and D 布尔运算符

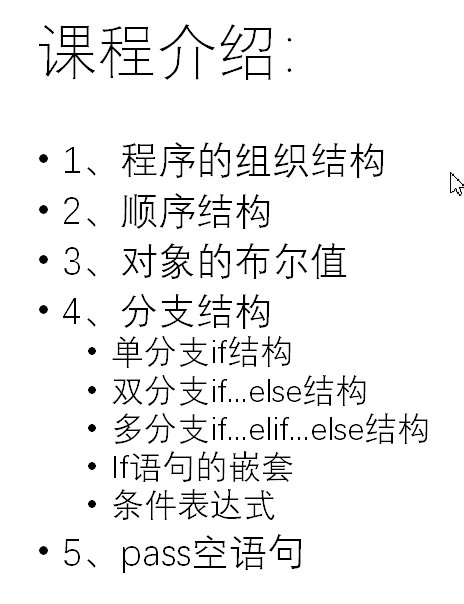
⑨or

⑩= E 赋值运算符

# 第三章知识点总结



# 第四章 往哪走呢？



# 程序的组织结构

27.程序的组织结构\_顺序结构 P28 - 00:40

任何简单或者复杂的算法

可以由顺序结构、选择结构和循环结构组成

if语句 while for-in语句

顺序结构

程序开始-代码1-代码2-代码3……代码N——程序结束

单击左键

可以在（每）一句打上断点 在向下运行观察程序结构

# 对象的布尔值

28.对象的布尔值 P29 - 00:08

Python一切为对象 所有对象都有一个布尔值

获取对象的布尔值

使用内置函数bool（）

以下对象的布尔值为False

\*False

\*数值0/0.0

\*None

\*空字符串（‘ ’ “ ”)

\*空列表（[ ] / list() ）

\*空元组(tuple() / () )

\*空字典 ({ } / dict() )

\*空集合 ( set（）)

除了以上的对象 都是True

# 选择结构

程序根据判断条件的布尔值选择性地执行部分代码

明确的让计算机知道在什么条件、该去做什么﻿

# 单分支结构

29.分支结构\_单分支结构 P30 - 02:33

中文语义：如果……就……

语法结构：if 条件表达式

条件执行体

money=1000

s=int（input（‘请输入取款金额：’））

if money>=s：

money=money-s #(缩进)

print（‘取款成功，余额为：’,money）

请输入取款金额：200

取款成功，余额：800

请输入取款金额：1200 （无结果）

# 双分支结构

30.分支结构\_双分支结构 P31 - 01:04

中文语义：如果…不满足…就

语法结构：if 条件表达式：

条件执行体1

else：

条件执行体2

num=int(input('请输入一个整数'))

if num%2==0:

print(num,'是偶数’)

else:

print(num,'是奇数')

请输入一个整数：6

6 是偶数

请输入一个奇数：7

7 是奇数

# 多分支结构

31.分支结构\_多分支结构 P32 - 01:23

中文语义： ……是……？ 不是

……是……？ 不是

……是……？ 不是

……是……? 不是

……是……？ 是

语法结构：

if 条件表达式1：

条件执行体1

elif 条件表达式2：

条件执行体2

elif 条件表达式N：

条件执行体N

[else:] #放[]里代表可以省略这一条

条件执行体N+1

（elif=else if）

类型1

if score>=90 and score<= 100

print('A级')

elif score>=80 and score<= 89

print('B级')

elif score>=70 and score<= 79

print('C级')

elif score>=60 and score<= 69

print('D级')

elif score>=0 and score<= 60

print('E级')

else:

print('对不起，成绩有误，不在成绩的有效范围')

类型2 （Python语言独有方式）

if 90<=score<= 100

print('A级')

elif 80<= score<= 89

print('B级')

elif 70<= score<= 79

print('C级')

elif 60<=score<= 69

print('D级')

elif 0<=score<= 60

print('E级')

else:

print('对不起，成绩有误，不在成绩的有效范围')

输出效果：

请输入一个成绩：98

A级

请输入一个成绩：180

对不起，成绩有误，不在成绩的有效范围

# 嵌套if

32.分支结构\_嵌套if的使用 P33 - 00:02

语法结构：

if 条件表达式1：

if 内层条件表达式：

内存条件执行体1

else：

内存条件执行体2

else：

条件执行体

answer=input（‘您是会员吗？y/n’）

money=float（input（‘请输入您的购物金额：’））

if answer=‘y’

if money>=200

print('打八折，付款金额为：'，money\*0.8)

elif money>=100

print(‘打九折，付款金额为：’，money\*0.9)

else：

print（‘不打折，付款金额为：’money）

else：

if money>=200

print('打九五折，付款金额为：'，money\*0.95)

else：

print（‘不打折，付款金额为：’money）

您是会员吗？y/n y

请输入您的购物金额： 120

打九折，付款金额为：108

# 条件表达式

33.条件表达式 P34 - 00:37

条件表达式是if……else（简写）

语法结构：

if 判断条件 else y

运算规则：

如果判断条件的布尔值为True，条件表达式的返回值为x，否则条件表达式的返回值为False

num\_a=int(input('请输入第一个整数'))

num\_b=int(input('请输入第二个整数'))

if num\_a>=num\_b

print(num\_a,'大于等于'，num\_b)

else:

print(num\_a,'小于‘，num\_b)

print（’输入条件表达式进行比较‘）

print(str(num\_a)+'大于等于'+str(num\_b) if num\_a >=num\_b else str(num\_a)+'小于'+str(num\_b))

请输入第一个整数：10

请输入第二个整数：20

使用条件表达式进入比较

10小于20

# Pass语句

34.pass语句 P35 - 00:35

语句什么都不做，只是一个占位符，用在需要语句的地方

使用时间：先搭建语法结构，还没好代码怎么写的时候

配套使用：

\*if 语句的条件执行体

\*for--in语句的循环体

\*定义函数的函数体

answer=input（’您是会员吗？y/n‘）

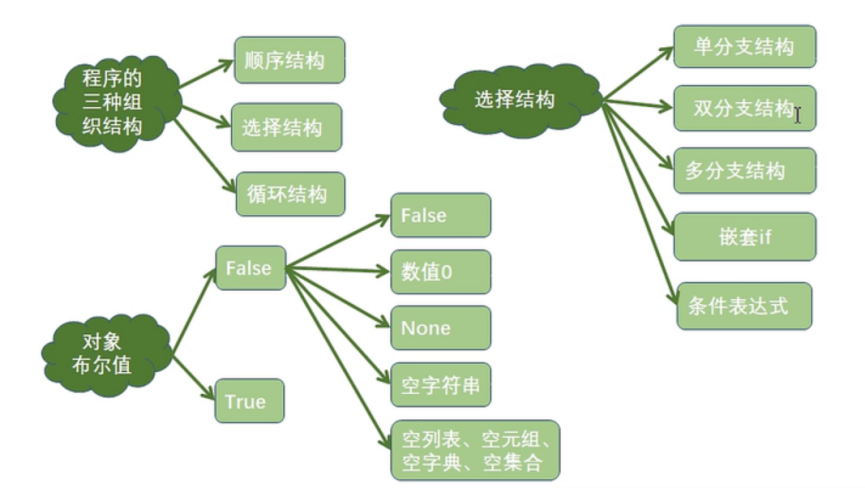
if answer='y'

pass

else

pass (搭建结构)

# 第四章总结



age=int(input(’请输入您的年龄：‘))

if age

print（age）

else

print('年龄为：'，age)

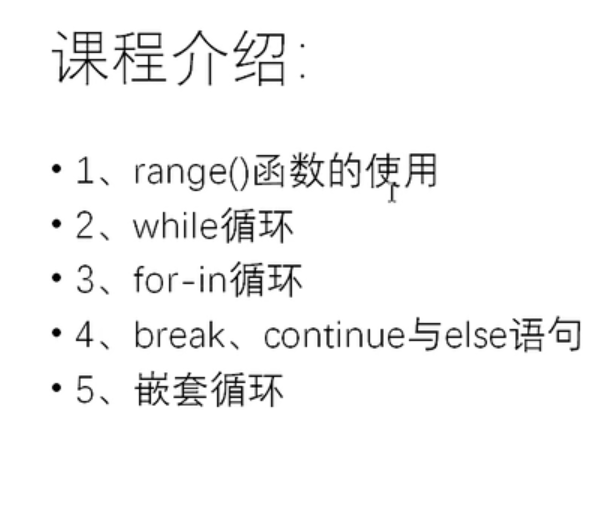
请输入您的年龄：18

18 \*18的布尔值是True

请输入您的年龄：0

年龄为：0 \*0的布尔值是False

# 第五章 转圈圈



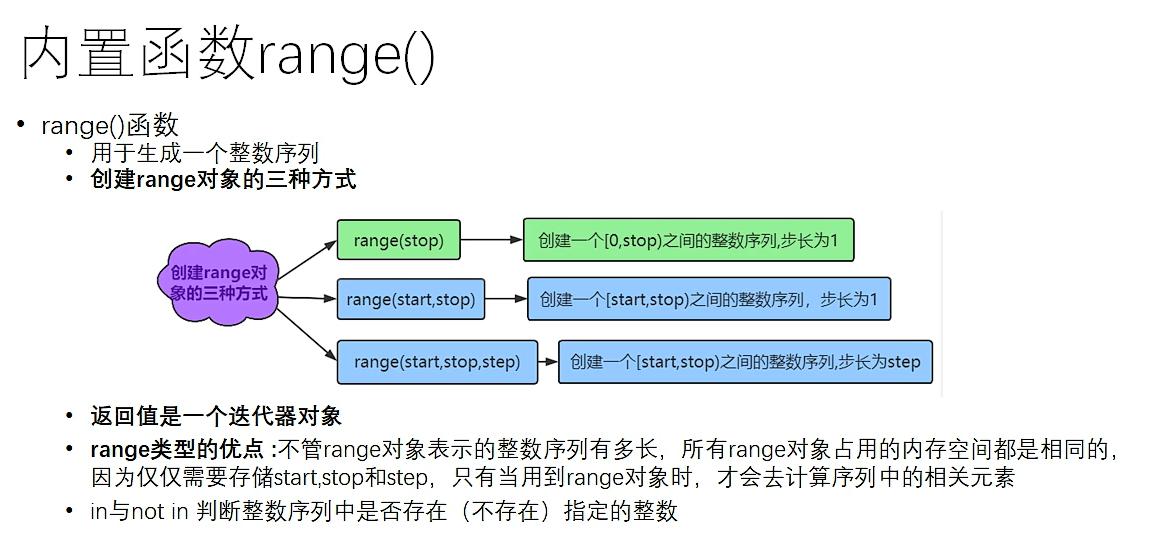
# range 函数的使用

35.range函数的使用 P36 - 00:06

课前介绍

内置函数（系统自带，无需下载任何模板）

range（）



创建range（）的三种方式

①range(stop)——创建一个以(0,stop)之间的整数序列，步长为1

r=range（10）

print（r）

range（0，10） \*结果是一个迭代器对象

print(list(r)) \*用于查看range对象中的整数序列

[0,1,2,3,4,5,6,7,8,9] \*默认从零开始

②range(start,stop)----创建一个(start,stop)之间的整数序列，步长为1

r=range(1,10) \*range右边是开区间，取不到

print(list(r))

[1,2,3,4,5,6,7,8,9]

③range(start,stop,step)----创建一个(start,stop)之间的整数序列，步长为step

r=range(1,10,2)

print(list(r))

[1,3,5,7,9]

in与not in判断整数序列中是否存在（不存在）指定的整数

print（10 in r）

print（9 in r）

print（10 not in r）

print （9 not in r）

False

True

True

False

优点：不论range中的整数序列有多长，占有的内存空间是一样的。只有计算时，才会计算相关元素

# 循环结构

﻿

36.while循环 P37 - 00:25

循环的分类：

\*while \*for-in

语法结构：

while 条件表达式

条件执行体（循环体）

选择结构的if 与循环结构while的区别

if是判断一次，条件为True执行一行

while是判断N+1次，条件为True执行N次

a=1

if a<10:

print (a)

a+=1

1

a=1

while a<10:

print (a)

a+=1

1

2

3

4

5

6

7

8

9

四步循环法

①初始化变量

②条件判断

③条件执行体（循环体）

④改变变量

\*while的循环的执行流程

\*计算0--4之间的累加和

sum=0

\*初始变量为0

a=0

\*条件判断

while a<5:

\*条件执行体（循环体）

sum+=a

\*改变变量

a+=1

print（'和为'，sum）

和为10

while练习题

37.whil练习题\_1到100之间的偶数和 P38 - 00:49

1--100的偶数和

sum=0

\*初始变量为0

a=1

\*条件判断

while a<100:

\*条件执行体（循环体）

if a%2==0

sum+=0

\*改变变量

a+=1

print（’1--100之间的偶数和‘，sum）

print（'和为'，sum）

1--100之间的偶数和 2550

1--100的奇数和

sum=0

a=1

while a<100:

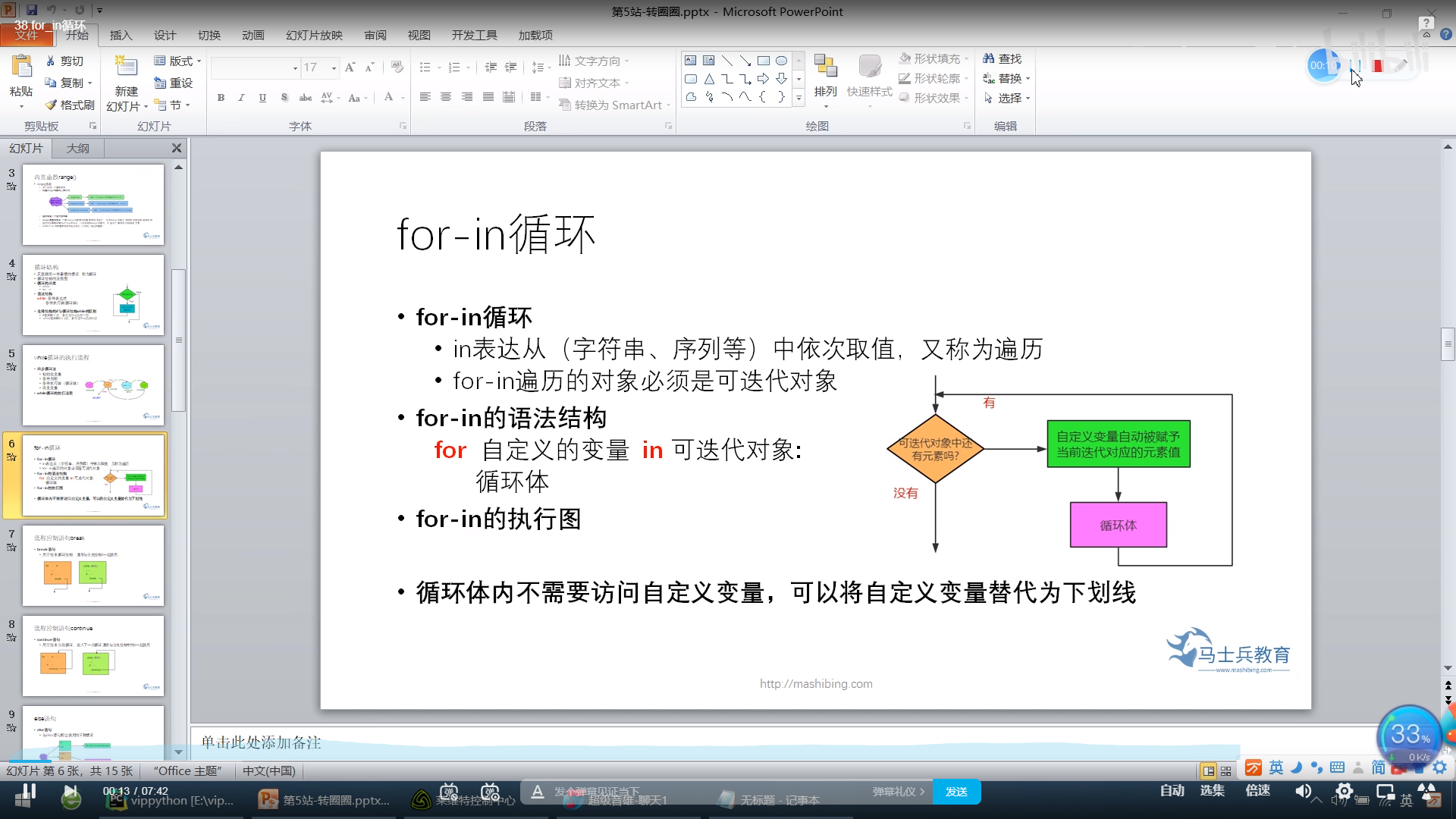
if a%2： \*0是False，奇数进行

sum+=0

a+=1

1--100之间的偶数和 2500

# for--in循环

38.for\_in循环 P39 - 00:33

字符串和序列都是可迭代对象

for item in ‘Python’:

print (item)

P

y

t

h

o

n

\*range()产生的整数序列，也是一个可迭代对象

for i in range(10):

print(i)

0

1

2

3

4

5

6

7

8

9

\*不使用变量，可将添加 "\_"

for \_ in range（5） \*循环5次

print('人生苦短，我用Python')

人生苦短，我用Python

人生苦短，我用Python

人生苦短，我用Python

人生苦短，我用Python

人生苦短，我用Python

print（使用for循环，计算1到100之间的偶数和）

sum=0

for item in range(1,101):

if item%2==0:

sum+=item

for--in循环练习题

# 100到999的水仙花数

39.for\_in练习题\_100到999之间的水仙花数 P40 - 00:13

水仙花数

每个位数的三次方的和与之相等

例如：153=3^3+5^3+1^3 370 371 407

for item in range（100，1000）：

ge=item%10

shi=item//10%10

bai=item//10

if ge\*\*3+shi\*\*3+bai\*\*3==item:

print(item)

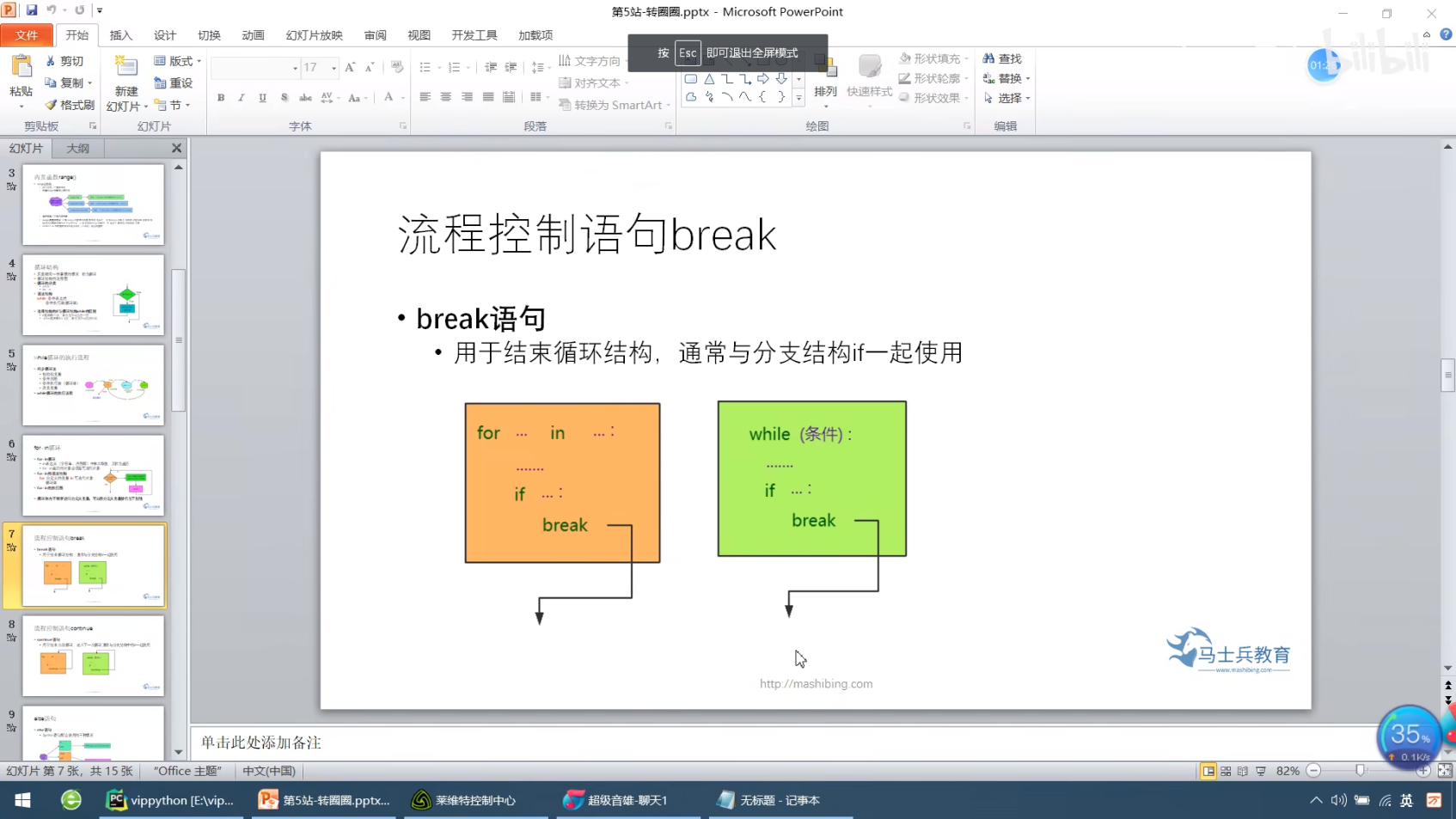
153

370

371

407

# 流程控制语句break

.

40.流程控制语句break P41 - 00:42

（非正常结束循环）

for item in range（3）：

pwd=input（‘请输入密码：’）

if pwd==’8888’：

print（‘密码正确’）

break

else：

print（‘密码不正确’）

请输入密码：8888

密码正确

请输入密码：4803

密码不正确

请输入密码：4803

密码不正确

请输入密码：4803

密码不正确

a=0

while a<3:

pwd=input(' 请输入密码')

if pwd ='8888'

print('密码正确')

break

else:

print('密码不正确')

a+=1

请输入密码：8888

密码正确

请输入密码：4803

密码不正确

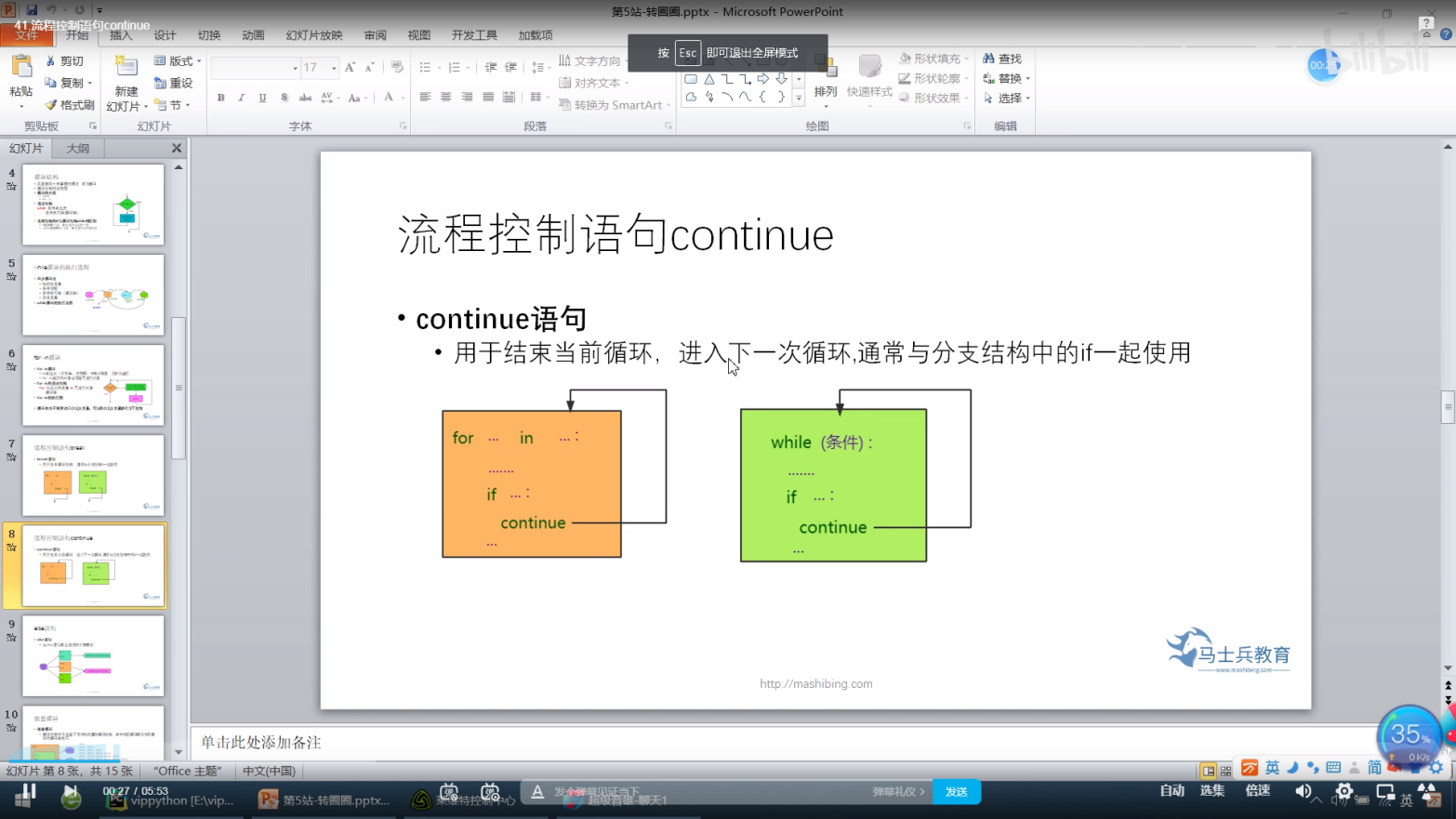
请输入密码：4803

密码不正确

请输入密码：4803

密码不正确

# 流程控制语句continue



41.流程控制语句continue P42 - 00:39

continue 语句

用于结束当前循环，进入下一次循环，通常与分支结构的if一起使用

for item in range(1,11)

if item%5==0

print(item)

5

10

for item in range（1,11）

if item%5！=0：

continue

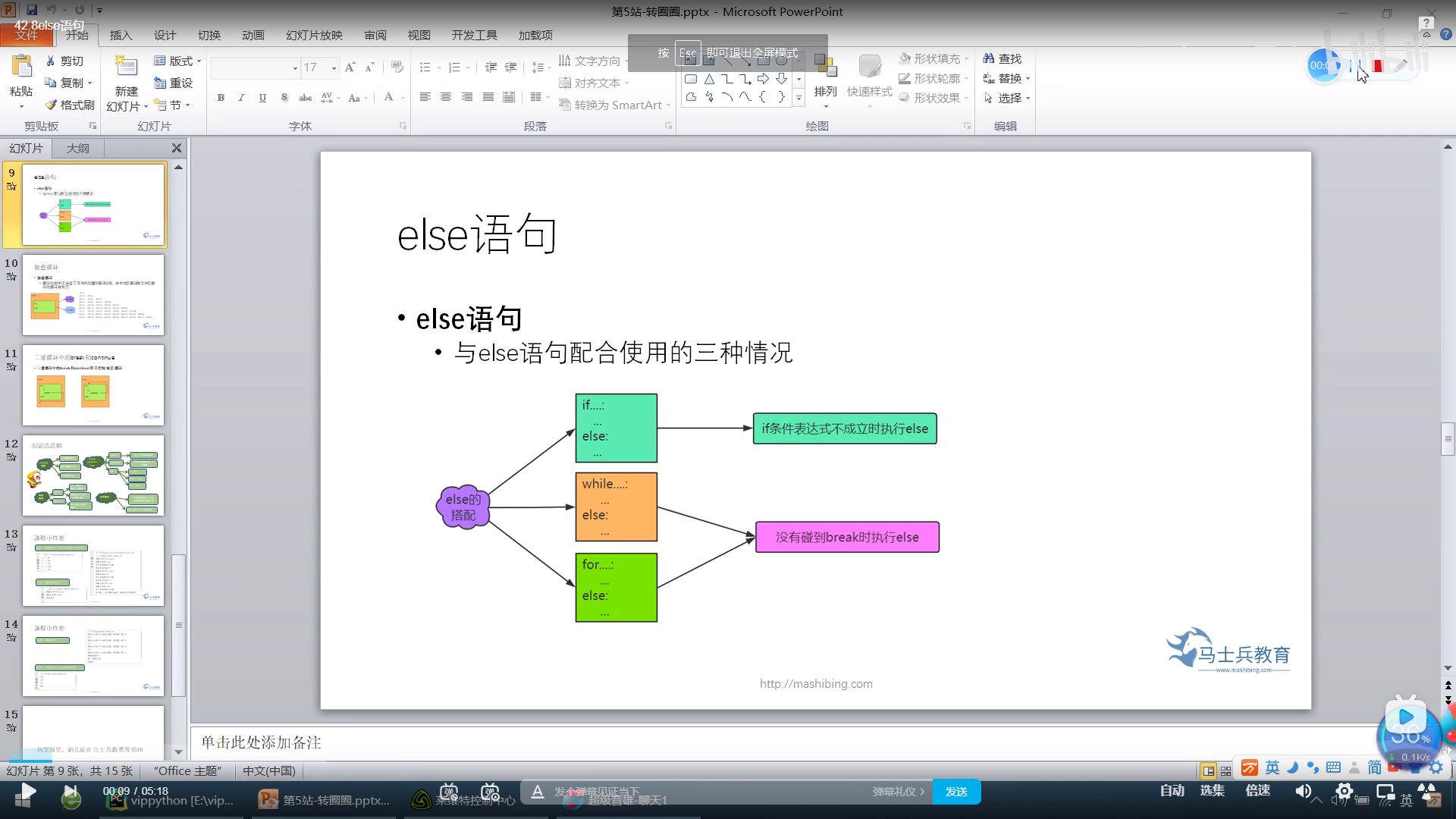
print（item）

5

10

# else 语句

42.8 else语句 P43 - 00:02



①if else ——if 条件不成立时执行else

②while else—— 循环正常执行完毕且没有碰到break 执行else

③for else —— 循环正常执行完毕且没有碰到break 执行else

for item in range(3)

pwd=input('请输入密码')

if pwd='8888'

print('密码正确')

break

else:

print('密码不正确')

a+=1

else：

print（‘对不起，三次密码均输入错误’）

请输入密码：8888

密码正确

请输入密码：4803

密码不正确

请输入密码：4803

密码不正确

请输入密码：4803

密码不正确

对不起，三次密码均输入错误

a=0

while a<3:

pwd=int(input('请输入密码：'))

if pwd==8888:

print('密码正确')

break

else:

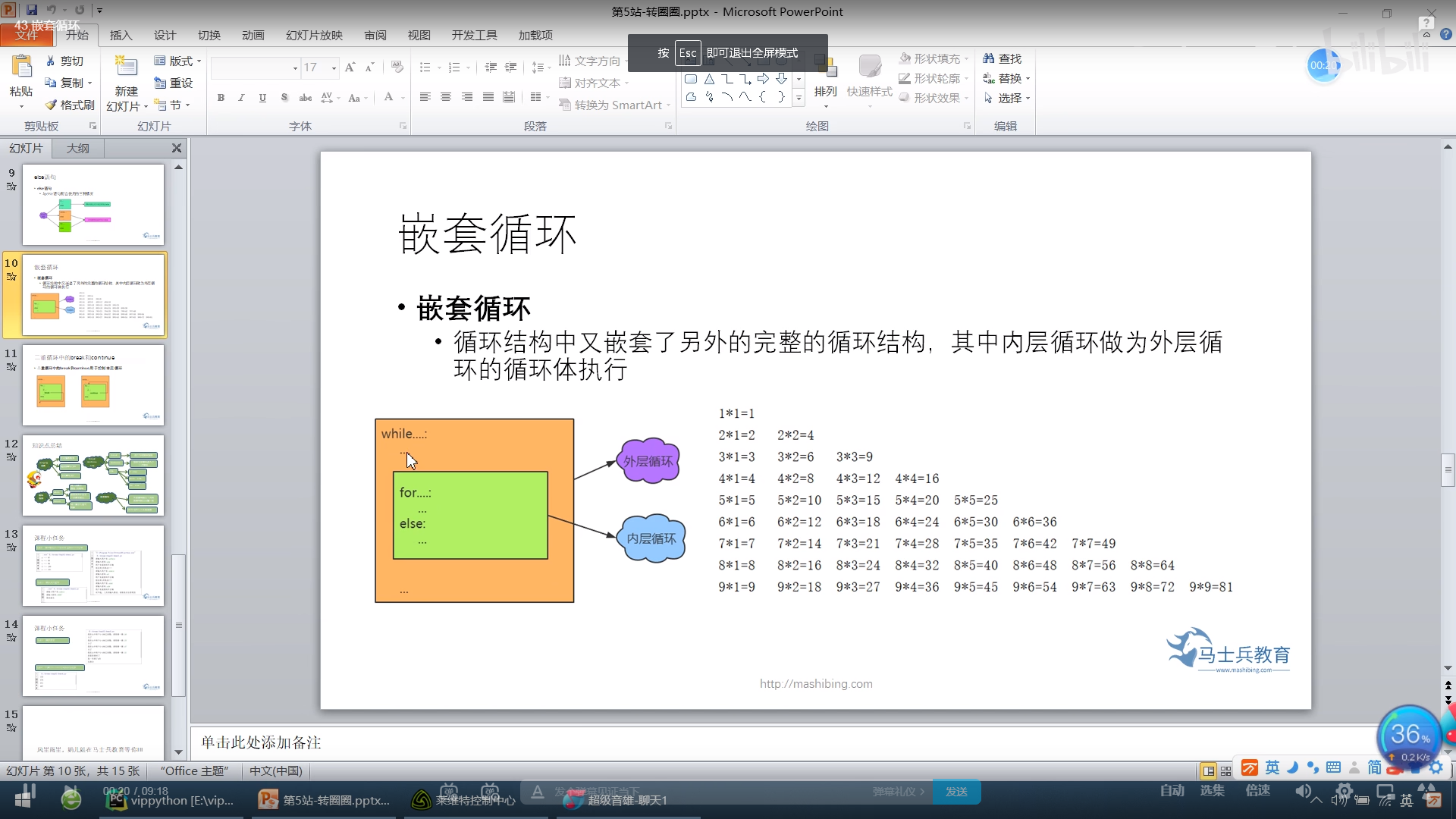
print('密码错误')

a+=1

else:

print('三次密码都输错了')

# 嵌套循环



43.嵌套循环 P44 - 00:38

嵌套循环

\*循环结构中又嵌套了另外的完整循环结构，其中内层循环做为外层循环的循环体执行

for i in range(1,4):

for j in range (1,5)

print('\*',end='\t') # end=’’是取消换行并加上空格

print() # print函数默认打印一行，结尾自动换行。

\*\*\*\*

\*\*\*\*

\*\*\*\*

for i in range(1,10):

for j in range (1,i+1)：

print('\*',end='')

print()

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

for i in range(1,10):

for j in range (1,i+1)：

print(i,'\*',j,'=',i\*j,end='')

print()

九九乘法表

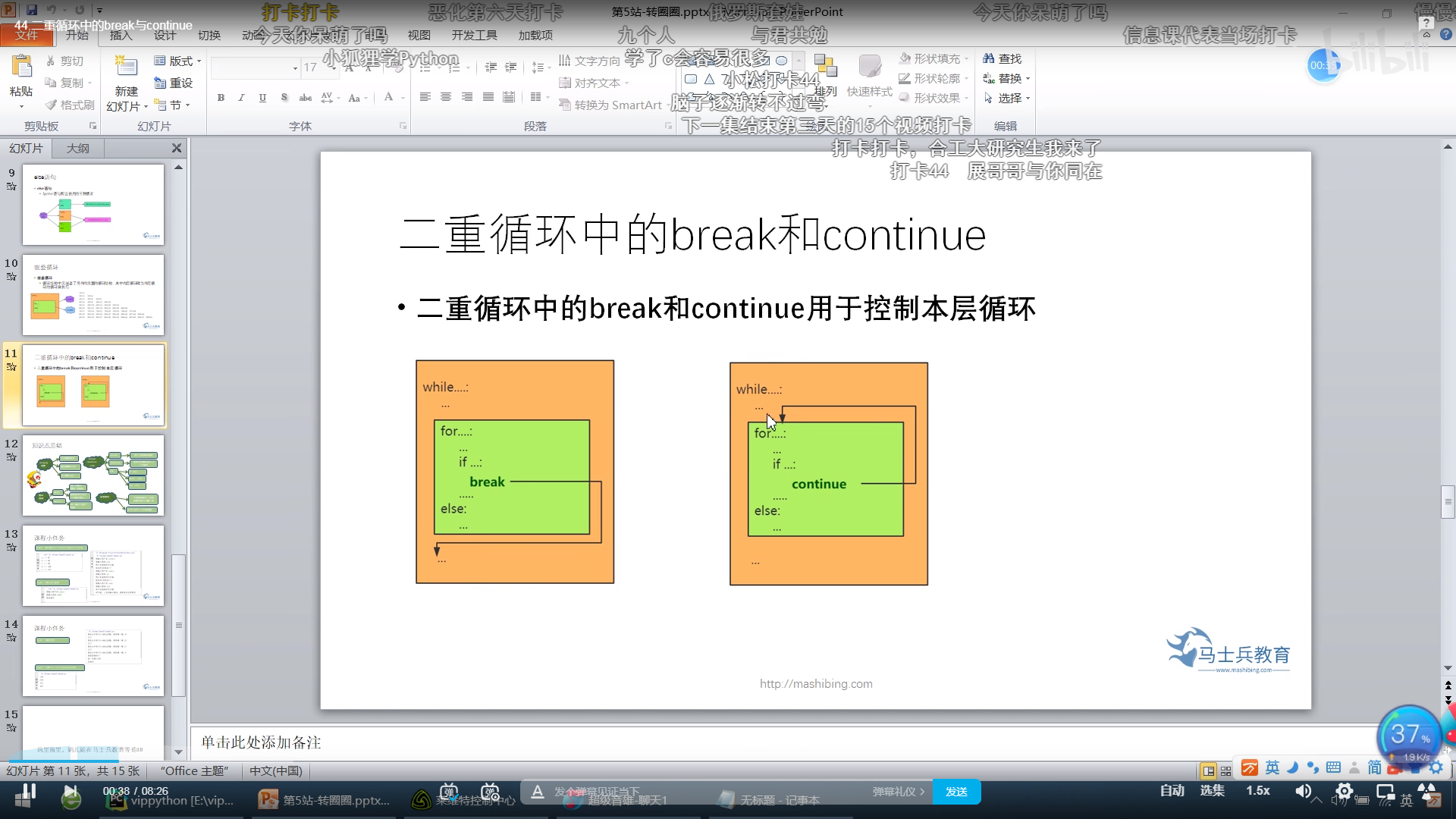
for i in range(10):

for j in range(1,i+1):

print(i,'\*',j,'=',i\*j,end=' ')

print()

# 二重循环中的break和continue



44.二重循环中的break与continue P45 - 00:05

二重循环中的break和continue用于控制本层循环

for i in range(5):

for j in range (1,11)：

if j%2==0

break

print(j)

1

1

1

1

1

for i in range(5):

for j in range (1,11)：

if j%2==0

continue

print(j，end='\t')

print()

1 3 5 7 9

1 3 5 7 9

1 3 5 7 9

1 3 5 7 9

1 3 5 7 9

# 总结



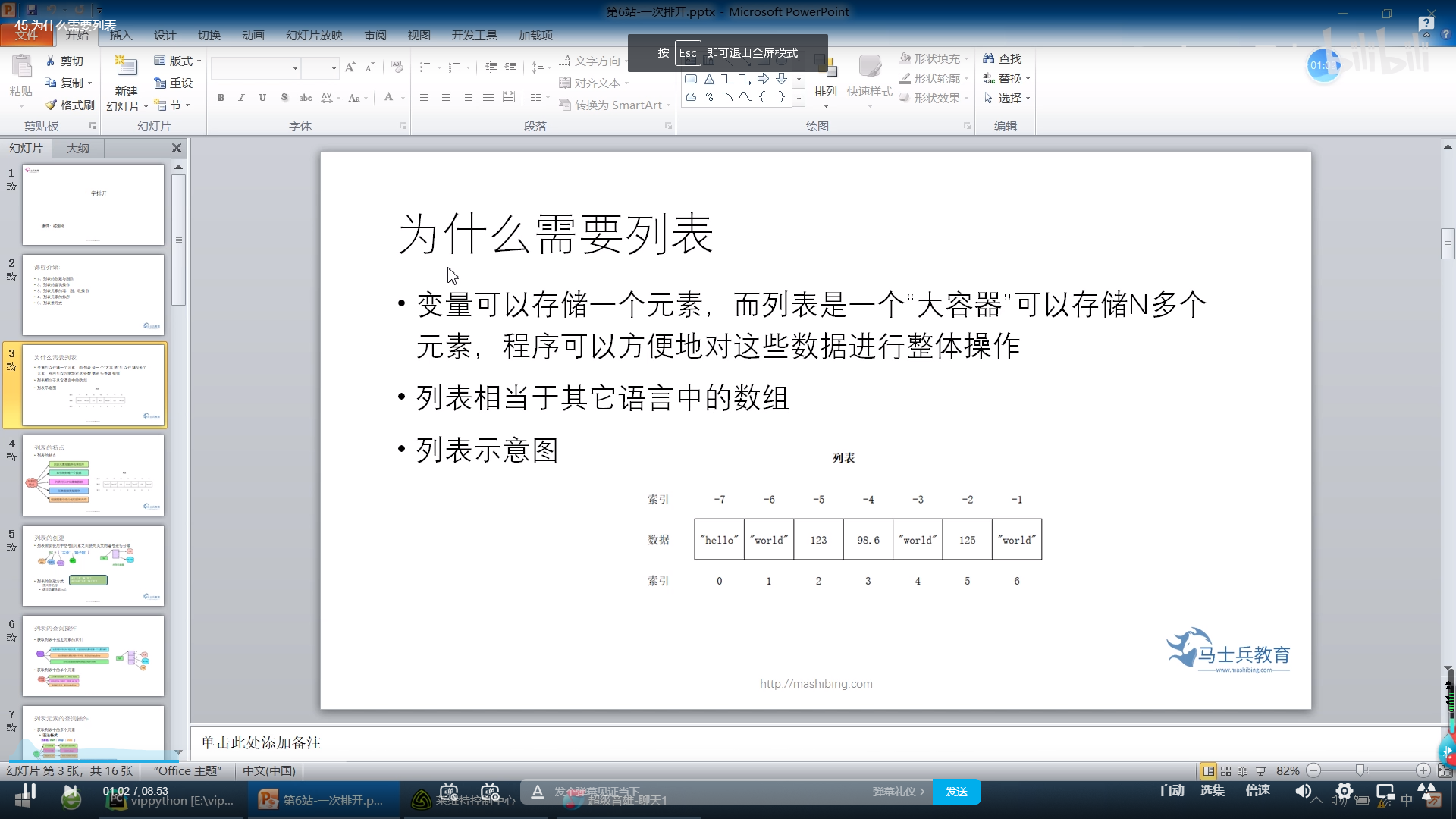
# 列表

为什么需要列表

45.为什么需要列表 P46 - 01:08

\*变量可以存储一个元素，而列表是一个“大容器”可以存储N个元素，程序可以方便地对这些数据进行整体操作

\*列表相当于其他语言的数组



a=10

lst['hello','world',98]

print(id(lst))'

print(type(lst))

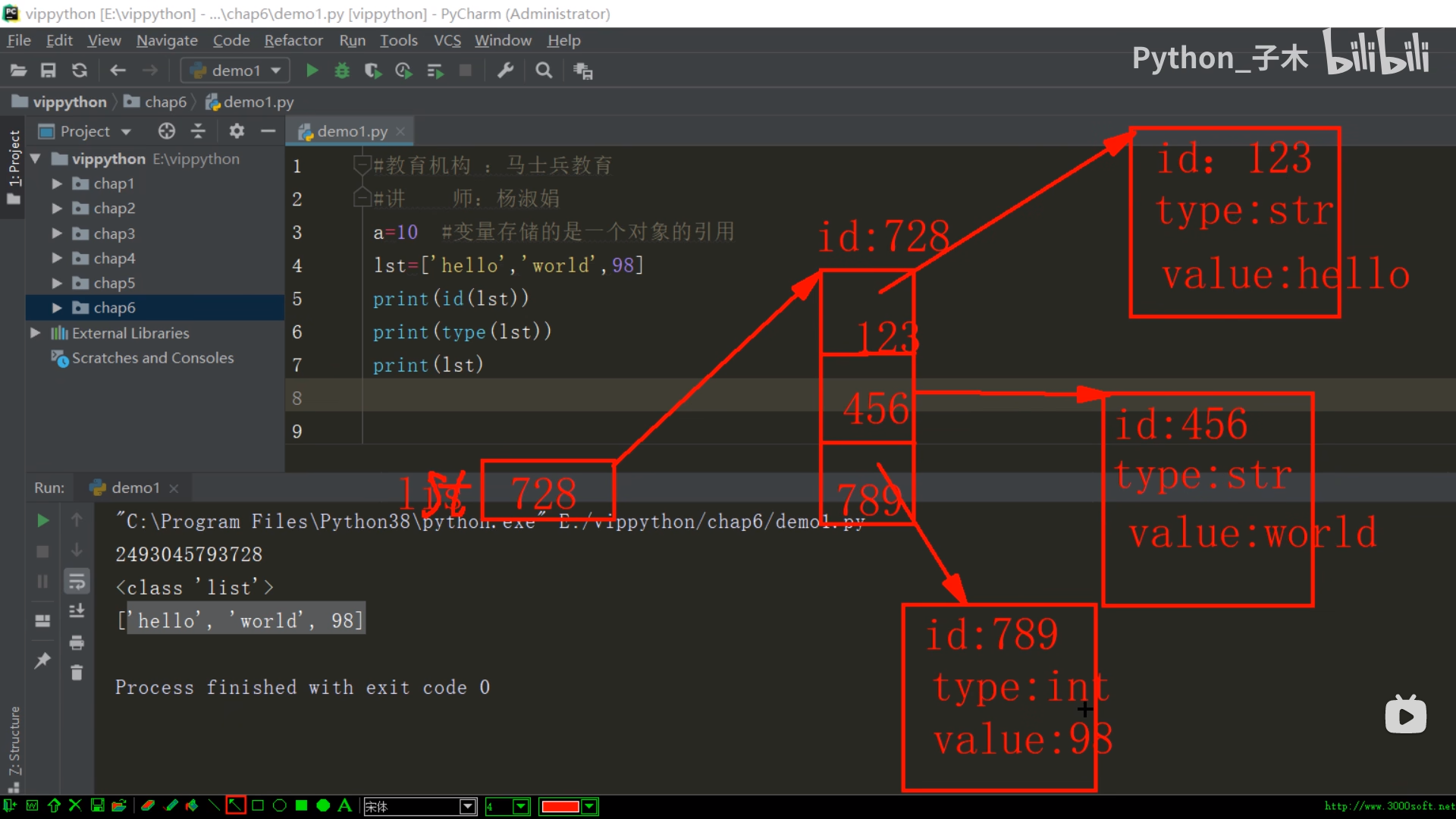
print(lst)

2546466131

<class,'list'>

['hellp','world','98']

内存示意图



# 列表的创建

46.列表对象的创建 P47 - 00:59

列表需要使用中的括号[ ] ,元素之间用英文的逗号分开

①使用方括号

lst=['hello','world',98]

②使用内置函数list（）

lst2=list(['hello','world',98])

# 列表的特点

47.列表的特点 P48 - 01:04

①列表元素按顺序有序排序

lst=['hello','world',98]

print（lst）

[‘hello’ ‘world’ 98]

②索引映射唯一个数据

lst=['hello','world',98]

print（lst[0],lst[-3]）

hello hello

③列表可以存储重复数据

lst=['hello','world',98,'hello']

print（lst[0],lst[-4]）

hello hello

④任意的数据类型混存

见上

⑤根据需要动态分配和回收内存

作者：调皮の姜太公 https://www.bilibili.com/read/cv14889923?spm\_id\_from=333.999.0.0 出处：bilibili