第3讲回顾

1、**科学抽象的三个过程**是（　　）、（　　）和（　　）。

2、**科学抽象的逻辑方法**主要包括（　　）、（　　）和（　　）。（　　）是从**个别事实**中概括出**一般原理**的思维方法；（　　）是从**一般性原理**出发，推出关于个别或**特殊事物**的思维方法。

3、**模型**是把对象实体通过适当的（　　），用适当的表现规则描绘出的原型的简洁（　　）。

4、**数学模型**是参照某种事物或系统的特征或数量依存关系，采用（　　），概括地或近似地表述出的一种（　　）。

第3讲回顾（续）

5、**数学建模**的7个步骤分别是模型的（　　）、（　　）、（　　）、（　　）、（　　）和（　　）。

6、常用的**数学建模方法**包括（　　）、（　　）和（　　）。

7、（　　）建模是通过（　　）建立数学框架，通过（　　）确定模型中包含的参数或关系。

8、简单来说，**最优化问题**就是在一定的约束条件下，求一个函数的（　　）或（　　）。对最优化问题建立数学模型，必须写出（　　）和（　　）。

预习课件

北京航空航天大學
BEIHANG UNIVERSITY

大学计算机基础

（理科类）

第4讲 程序设计与Python
简单数据类型及词法

北京航空航天大学

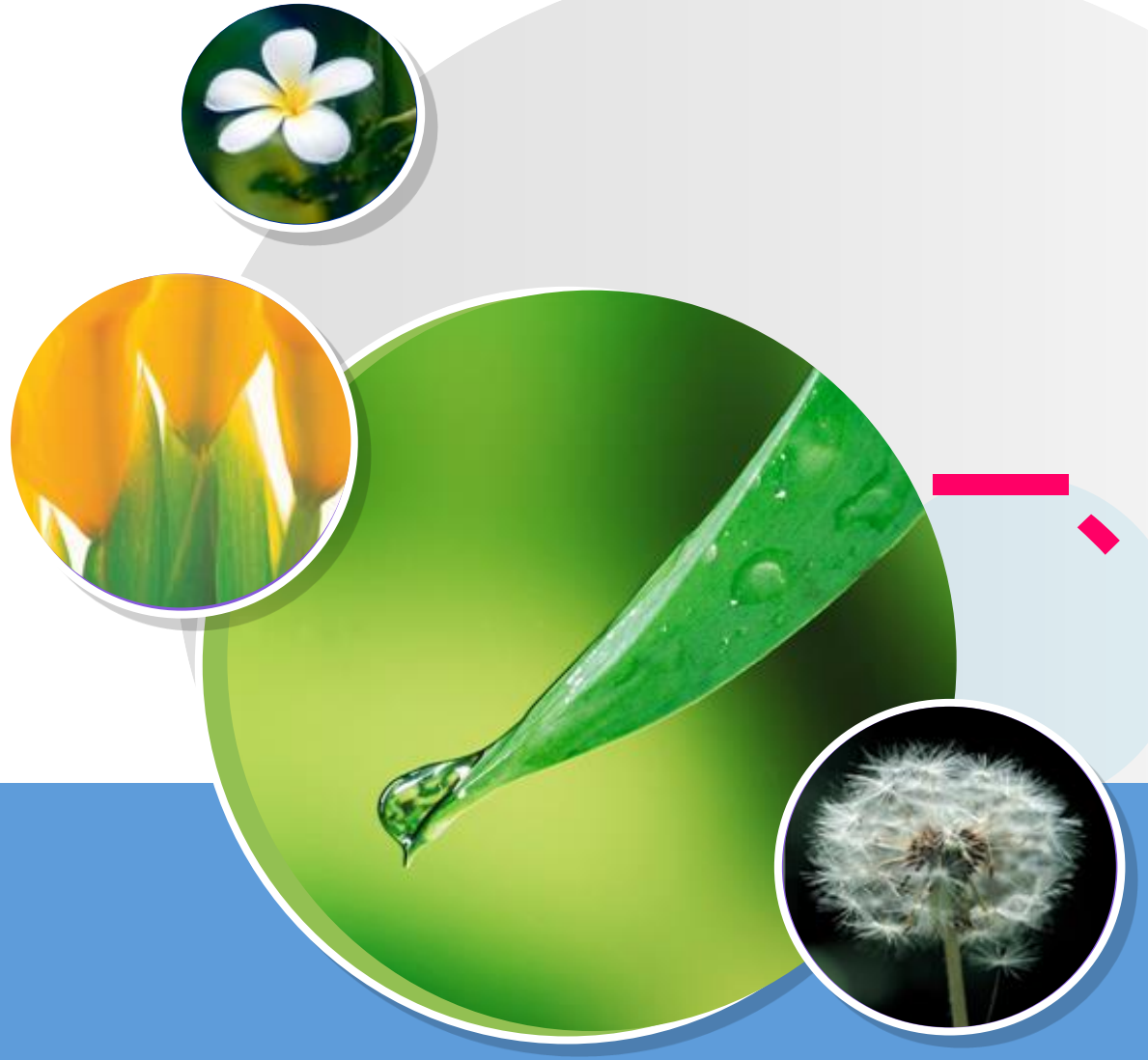
第4讲 程序设计与Python简单数据类型及词法

4.1 程序与程序设计语言（自学）

4.2 Python简单数据类型及词法

4.2 Python简单数据类型及词法

- 一、Python的内建函数
- 二、Python的标准库
- 三、简单数据类型
- 四、Python词法



北京航空航天大學
BEIHANG UNIVERSITY

一、Python的内建函数

北京航空航天大学

内建函数分类

- **Python解释器提供了69个内建函数**（内置函数）（Build-in Function），这些函数用户不需要导入库，而可以直接使用
- **分为六类**
  - **数学运算类**：**abs(x)**求绝对值，**float(x)**将一个字符串或数转换为浮点数，**bin(x)**将整数x转换为二进制字符串，**oct(x)**将整数x转换为八进制字符串，**hex(x)**将整数x转换为十六进制字符串
  - **集合类操作**：**max**(iterable[, args...][key])返回集合中的最大值，**min**返回序列中最小的元素，**str**([object]) 转换为string类型

内建函数分类（续）

- **逻辑判断：cmp**(x, y)比较x和y的大小
- **反射**：**len(s)**返回集合长度，**type(object)**返回该object的类型
- **IO操作**：**input**([prompt])获取用户输入，**print** ([prompt])打印函数
- **其他**：**help()**获取帮助信息

- **说明**：**cmp()函数是Python 2中的一个用于比较两个列表、数字或字符串等的大小关系的函数**
- **但在Python 3中已经没有这个函数了**

常用内建函数：（1）数学运算类

函数名	含义	示例	说明
abs(x)	求一个数的绝对值	>>>abs(-10.02) 10.02	
bin(x)	将整数x转换为二进制字符串	>>> bin(9) '0b1001'	可用来将十进制数转换为二进制数
divmod(x,y)	商余，（x//y，x%y），同时输出商和余数	>>> divmod(10,3) (3, 1)	
int(x)	把浮点数**向下取整**为整数；或把一个数字字符串转换为整数	>>>int(32.9) 32 >>> int('123') 123	简单舍弃小数部分，**只取整数部分**。对于**负数取整**是按照**朝向0**的方向进行

float(x)	将一个字符串或数转换为浮点数	>>> float(12)
		12.0
		>>> float('12.3')
		12.3	

常用内建函数：（1）数学运算类（续）

函数名	含义	示例	说明
pow(x,y)	指数运算，x的y次方	>>>pow(2,3) 8	

round(x,[d])	把浮点数**四舍五入**为最接近的整数值。d是保留小数位数，默认值为0。
	round(x)则是保留整数。
	在Python 3中该函数表示对x**四舍六入五凑偶**
	
	建议谨慎使用！取整一般采用int函数。
		>>>round(1.0/2.0)
		0
		>>> round(7.5)
		8
		>>> round(8.5)
		8
		>>> round(9.5)
		10
		>>> round(10.5)
		10
		>>> round(8.51)
		9
		>>> round(3.1415926,5)
		3.14159
			当需要四舍五入后保留n位小数时，宜使用该函数。
			注意：当d=0时，小数部分必须大于“0.5”，才能入“1”；**小于“0.5”的小数部分都舍掉；小数部分等于0.5时，整数取最靠近的偶数。**

常用内建函数：（2）集合类操作

函数名	含义	示例	说明

max	返回序列中最大的元素	>>> numbers=[99, 54, 387]
		>>> max(numbers)
		387
		>>> max(2,7,3,8,1,5)
		8
			序列可以是列表或元组、字符串

min	返回序列中最小的元素	>>> numbers=[99, 54, 387]
		>>> min(numbers)
		54
		>>> min(2,7,3,8,1,5)
		1
			序列可以是列表或元组、字符串

len	返回序列中所包含元素的数量	>>> numbers=[99, 54, 387]
		>>> len(numbers)
		3
		>>> len('world')
		5
			序列可以是列表或元组、字符串

常用内建函数：（2）集合类操作（续）

函数名	含义	示例	说明
list	对一个序列（如字符串）创建一个列表	>>> list('Hello') ['H', 'e', 'l', 'l', 'o']	适用于所有类型的序列，而不只是字符串
range	**范围函数**，用于产生某个整数范围内的整数数字	>>>range(0, 10) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	类似于分片，**指定的范围不包括上限**。常用在**for**语句中指定迭代范围
sum	求和函数，用于对序列进行求和计算	>>>sum([0,1,2,3,4], 2)	列表计算总和后再加 2 12

补充：字符串的join方法

- 字符串的**join方法**： 将列表中的各个字符，连接成一个字符串

  <'连接符'>.join(<字符串列表>)

  - 若在引号中给出某个连接符，则是用该连接符连接各字符

    ```
    >>> alist=['1', '2', '3', '4', '5']
    >>> '+'.join(alist)
    '1+2+3+4+5'
    ```

  - 若连接符为空，则直接连接各字符

    ```
    >>> alist=['H', 'e', 'l', 'l', 'o']
    >>> ''.join(alist)
    'Hello'
    ```

常用内建函数：（3）IO操作

函数名	含义	示例	说明

input	要求用户输入合法的Python表达式，可以是字符串或数字或中文	>>> input ('Enter your name：')
		Enter your name：John
		'John'
		>>> input ('Enter your weight(kg)：')
		Enter your weight(kg)：49.5
		'49.5'
			Python把用户输入当成是字符串，故在Shell窗口显示出来的内容均用**单引号**或**双引号**（当你输入的内容本身包括单引号时）括起来了

print	打印表达式	>>> print ('Age:', 42)
		Age: 42
		>>> name = input("请输入你的姓名：")
		>>> print ("你的姓名是：" +name)
		>>> print(1,2,3,end='!')
		1 2 3!
		>>> print(1,2,3,end='!',**sep=''**)
		123!
			可以打印多个表达式，之间用逗号隔开，这里的“42”认为是字符串。当需要打印多个**字符串**时，可用“+”连接。
			用**sep参数**控制多个值的**分隔符**（默认是**空格**），sep=‘’则使得值之间没有空格 。
			使用end参数控制打印内容的结尾。

默认多个值之间有一个空格

input函数

- **input函数**
  - **input函数**用来提供用户输入合法的Python表达式，一般用于**传递输入数据**给程序
  - input函数使用一个**字符串**作为参数，用来**提示用户应输入的内容——一般必须有此提示语！括号中不要为空！**

格式 变量名=*input*(" <字符串> ")

- 但是，**OJ（在线评测系统）无法评测input中的字符串**，故在提交到OJ上时，要求input后的括号中为空
- **input函数一般配合赋值语句使用**

```
name = input("Please enter your name: ")
```

输入及类型转换

- input函数的返回值为字符串
- 但可以通过类型转换，将其转换为整型或浮点型

```
ID = input("Please enter your Student ID: ")
age = int(input("Please enter your age: "))
height = float(input("Please enter your height(m): "))
```

字符串

整型

浮点数

- 通过input函数输入的内容默认其类型为字符串
- 如果需要进行算术运算，则必须先将其转换为int或float

print函数

print函数

- **print函数**用于在Shell窗口**显示**用户希望显示的内容（**字符串**或**变量的值**）

格式 *print*(" <字符串1>", <变量1>, " <字符串2> ", <变量2>,……)

- 当需要显示的内容较多时，最好在变量名的前面加上说明其含义的**字符串**，否则，难以判断输出的是哪个变量的值！

```
print("正态分布样本值为：", normal)
```

使用print函数注意事项

- 但是，**OJ（在线评测系统）无法评测print中的字符串**，故在提交到OJ上时，要求print后的括号中只有变量名
- 故可以在IDLE或Spyder中**调试**时在变量名前**加上提示字符串**
- 调试通过后，再将这些语句**注释掉**，重新写几行**print语句**

- **如果希望空一行，可以使用print()**
- **如果希望换行，也可以在上一条print语句后面的括号中最后加上换行符'\n'**

不同形式的输出

- 不同形式的输出

print-不同形式输出.py

```
name="Alice"
age=15
print ("Are you really", name , "? ")
print ("Are you really" + name + "? ")
print ("Today", name , "is" , age , "years old. ")
print ("Ten years later",name,"will be",age+10,"years old.")
```

显示时逗号分隔的各个值之间间隔一个空格

可以用加号连接多个字符串，但打印时彼此之间无空格

表达式

```
Are you really Alice ?
Are you reallyAlice?
Today Alice is 15 years old.
Ten years later Alice will be 25 years old.
```

比较它们的区别

技巧：使用sep参数和end参数控制输出格式

- **print函数传入多个待打印的值时，用sep参数控制值的分隔符（默认是空格），sep=''则使得值之间没有空格**
- **使用end参数控制打印内容的结尾**
  - **end='\t'**表示输出的末尾以**Tab键**结束，则下一条print语句打印的内容将与刚才打印的内容的第一个字符**间隔8个字符输出，不换行**
  - **end=' '**表示以**空格**结尾，则下一条print语句打印的内容与刚才打印的内容末尾空一格输出，**不换行**
  - **end=''（两个单引号）**表示下一条print语句打印的内容**紧接着**刚才打印的内容末尾输出，**不换行**

```
>>> print(1,2,3,end='!')
1 2 3!
>>> print(1,2,3,end='!',sep='')
123!
>>> print(1,2,3,end='!',sep='+')
1+2+3!
>>> a=7
>>> b=9
>>> print(a,b,end='!',sep='')
79!
```

技巧：格式化字符串输出

- **当需要打印的变量值为不同类型（如浮点数）时，可以采用格式化字符串的方法，来指明变量值的类型、位数**

例：

```
print('估算2018年的人口数为%0.3f' % Result) #浮点数保留小数点后3位小数
```

```
>>>
估算2018年的人口数为1394.035
```

字符串格式化操作符

- %称为**字符串格式化操作符**，在%的左侧放置一个字符串（**格式化字符串**），右侧放置**希望格式化的对象**（可以是**数字**、**字符串**、**变量**或**表达式**）

格式化输出

格式 <模板串> % <值元组>

- **模板串中用%标记“空位”，输出时用值填入**
- **值元组可以是一个数值或多个数值，或者一个或多个变量名**
- **格式化运算的结果是一个字符串**

- **例如**

```
>>> print ("The price is RMB%.2f" %(100))
The price is RMB100.00
```

模板串

值元组

深刻理解

格式描述

- **空位：格式指示符（转换说明符），描述了填入的值的输出形式**

  %<宽度>.<精度><类型字符>　　例如：**%4.2f**

  - **三种常用类型字符：decimal, float, string**
    - ✓ **%d**表示被格式化为**带符号十进制整数**
    - ✓ **%f**（或**%F**）表示被格式化为**浮点数**
    - ✓ **%s**表示被格式化为**字符串**
    - ✓ **%c**表示格式化**字符**及其 **ASCII 码**
    - ✓ **%o**表示转换为**带符号八进制整数**
    - ✓ **%x**（或**%X**）表示转换为**带符号十六进制整数**

格式描述（续1）

宽度：指定用多少位显示数值

- ✓ **省略宽度**或指定为**0**：根据值的**实际宽度**显示

```
>>> x= 3.141592653
>>> print('pi=%.4f' % x)      #按值的实际宽度显示，小数点后保留4位
pi=3.1416
```

- ✓ 当**宽度>值的实际长度**时：**右对齐**显示

```
>>> y= 3.141592653
>>> print('pi=%10.4f' % y)      #最小宽度为10，小数点后保留4位
pi=    3.1416
```

- ✓ 当**值的实际宽度>指定宽度**时：按照值的**实际宽度**输出

```
>>> print('pi=%6.5f' % y)      #最小宽度为6，小数点后保留5位
pi=3.14159
```

- ✓ 若在指定宽度前**加负号**：**左对齐**显示

```
>>> print('pi=%-10.4f' % y)      #最小宽度为10，宽度前加负号，小数点后保留4位
pi=3.1416
```

格式描述（续2）

- **精度：指示浮点数值的小数位数**
  - ✓ **省略时：**按系统默认的小数位数显示

```
>>> import math
>>> print ("pi=%0.30f" % (math.pi))
pi=3.141592653589793115997963468544
>>> print ("pi=%20s" % math.pi)
pi=   3.141592653589793
```

宽度为20位

课后练习、体会

格式描述（续3）

- 当希望**格式化的对象**有**多个**时，只需在“%”右边，采用**元组**形式将变量名（或表达式）按顺序写出来即可。

```
#data_convert【方法一】.py
x,y=map(float, input().split())     #一个点的横坐标和纵坐标
a,b,c=map(float, input().split())  #直线方程参数

print('x=%.1f, y=%.1f' % (x,y))
print('a=%.1f, b=%.1f, c=%.1f' % (a,b,c))
```

```
1 2
2 3 4
x=1.0, y=2.0
a=2.0, b=3.0, c=4.0
```

北京航空航天大學
BEIHANG UNIVERSITY



二、Python的标准库

北京航空航天大学

Python的标准库

- **标准库**（standard liberary）：Python的标准安装提供的一组**模块**。在每个模块中，提供了若干**函数**，方便用户调用它们，进行相应运算
  - **math和cmath模块**：包含用于计算实数和复数的数学函数（如向下取整函数**floor**、求平方根函数**sqrt**、正弦函数**sin**等）
  - **random模块**：包含产生随机数的函数（如**randint**函数， **uniform**函数）
    - ✓ **randint(a,b)**：产生指定范围**(a,b)**内的一个随机**整数**
    - ✓ **uniform(a,b)**：生成指定范围内**(a,b)**的一个随机**浮点数**

```
>>> import random
>>> a=random.randint(0,10)
>>> a
3
>>> b=random.randint(0,10)
>>> b
5
```

```
>>> x=random.uniform(0,100)
>>> x
48.939112172354385
>>> y=random.uniform(0,100)
>>> print('%.4f' % y)
81.6205
```

格式化字符串输出

Python的标准库（续）

- **copy模块**：包含deepcopy函数（深拷贝）
- **datetime模块**：提供显示日期和时间的格式化方法
  - ✓ **datatime.date**：**日期**表示类，可以表示年、月、日等；
  - ✓ **datatime.time**：**时间**表示类，可以表示小时、分钟、秒、毫秒等；
  - ✓ **datatime.datetime**：**日期和时间**表示类，功能覆盖datatime.date和datatime.time

今天是哪天?

```
>>> import time
>>> from datetime import date
>>> today = date.today()
>>> today
datetime.date(2020, 10, 18)
```

返回当前的本地日期

- **要想使用标准库中的某个模块，必须先使用import语句导入该模块；再对模块中的函数进行调用**

import语句

- **import语句**用于从外部模块导入**名称**（绑定到函数、类或其他值的变量）
- 要想使用math库中的某个函数，必须先使用“**import math**”语句导入math库；再使用形如“**math.<函数>(参数)**”的语句对该函数进行调用
  - 【**例**】math库中**log函数**实现对数运算

    如math.log(8,2)实现$\log_2 8$

```
>>> import math
>>> math.log(8,2)
3.0
```

数学类函数库math

- math库为Python语言处理实数**基本数学计算**提供了一些常用**数学函数**

函数名	数学表示	含义	示例	说明
pi	π	**圆周率π**的近似值，15位小数	>>> math.pi 3.141592653589793	
ceil(x)		浮点数**向上取整**	math.ceil(32.9) 33	
floor(x)		对浮点数**向下取整**	>>>math.floor(32.9) 32 **>>> math. floor(-12.51)** **-13**	对于负数，不是简单取整数部分，而是取小于x的整数
log(x)	lgx	求x的对数，以e为基	>>> math.log(10) 2.302585092994046	x可为整数或浮点数，结果为**浮点数**
log10(x)	$\log_{10}x$	求x的对数，以10为基	>>> math.log10(100) 2.0	x可为整数或浮点数，结果为**浮点数**

log(x,n)	$\log_n x$	求x的对数，以n为基	>>> math.log(8,2)
			3.0
			>>> math.log(9,2)
			3.169925001442312 6
				x可为整数或浮点数，结果为**浮点数**

floor函数与int函数完全相同吗?

floor函数与int函数完全相同吗？

```
>>> int(12.51)
12
>>> import math
>>> math. floor(12.51)
12
>>> int(-12.51)
-12
>>> math. floor(-12.51)
-13
```

向下取整，因为(-13)<(-12.51)，所以取-13

- math.floor与int函数对于正的浮点数运算结果相同。但floor函数使用之前必须导入math库；而int函数是内建函数，可以直接使用，所以更方便。
- 但是，对于负的浮点数来说，二者结果是不同的！

数学类函数库math（续）

函数名	数学表示	含义	示例	说明
e	e	自然常数e的近似值，15位小数	>>> math.e 2.718281828459045	
sqrt(x)	$\sqrt{x}$	计算一个浮点数的平方根	>>> math.sqrt(9) 3.0	该函数不能计算复数或虚数的平方根
degrees(x)		弧度转换为角度	>>> math.degrees(2) 114.59155902616465	
radians(x)		角度转换为弧度	>>> math.radians(90) 1.5707963267948966	
sin(x)	sinx	正弦函数	>>> math.sin(math.pi/2) 1.0	结果为浮点数
asin(x)	arcsinx	反正弦函数	>>> math.asin(1) 1.5707963267948966	x∈[-1.0,1.0]

- math库还有cos(x)、tan(x)、acos(x)、atan(x)
- 上述函数都封装在math库里，**不能直接使用，必须先导入math库**
- **思考**：用什么运算可以代替sqrt(x)函数?

【课堂练习】

- **【课堂练习】**尝试Python编程，输入一个整数n（n<=15），调用math库的pi函数获得π，再将π保留**n位**小数输出。
- **提示**：使用内建函数**round(x,[d])**）

三分钟内完成

总结：几个取整函数的比较

- **int(x)**：把浮点数**向下取整**为整数（简单舍弃小数部分，**只取整数部分）**；或把一个数字字符串转换为整数
- **round(x,[d])**：把浮点数**四舍五入**为最接近的整数值。d是保留小数位数，默认值为0。在Python 3中该函数表示对x**四舍六入五凑偶**
  - 如round(3.5)=4，round(2.5)=2
- **math.ceil(x)**：浮点数**向上取整**
- **math.floor(x)**：浮点数**向下取整**
  - 对于正的浮点数，运算结果与int函数相同：int(12.3)= math.floor(12.3)=12
  - 对于**负**的**带小数**的浮点数，运算结果与int函数**不同**： int(-12.3)=-12， **math.floor(-12.3)=-13**

【讨论1】

- **请思考，以下几个函数的计算结果是什么?**

（1）int(12.51)=?　　int('789')=?

（2）**round(12.5)=?**　　round(12.51)=?

（3）math.ceil(12.49)=?　　math.ceil(12.51)=?

（4）math.floor(12.49)=?　　math. floor(12.51)=?

（5）math.floor(-12.49)=?　　math.floor(-12.51)=?

北京航空航天大學
BEIHANG UNIVERSITY

三、简单数据类型

北京航空航天大学

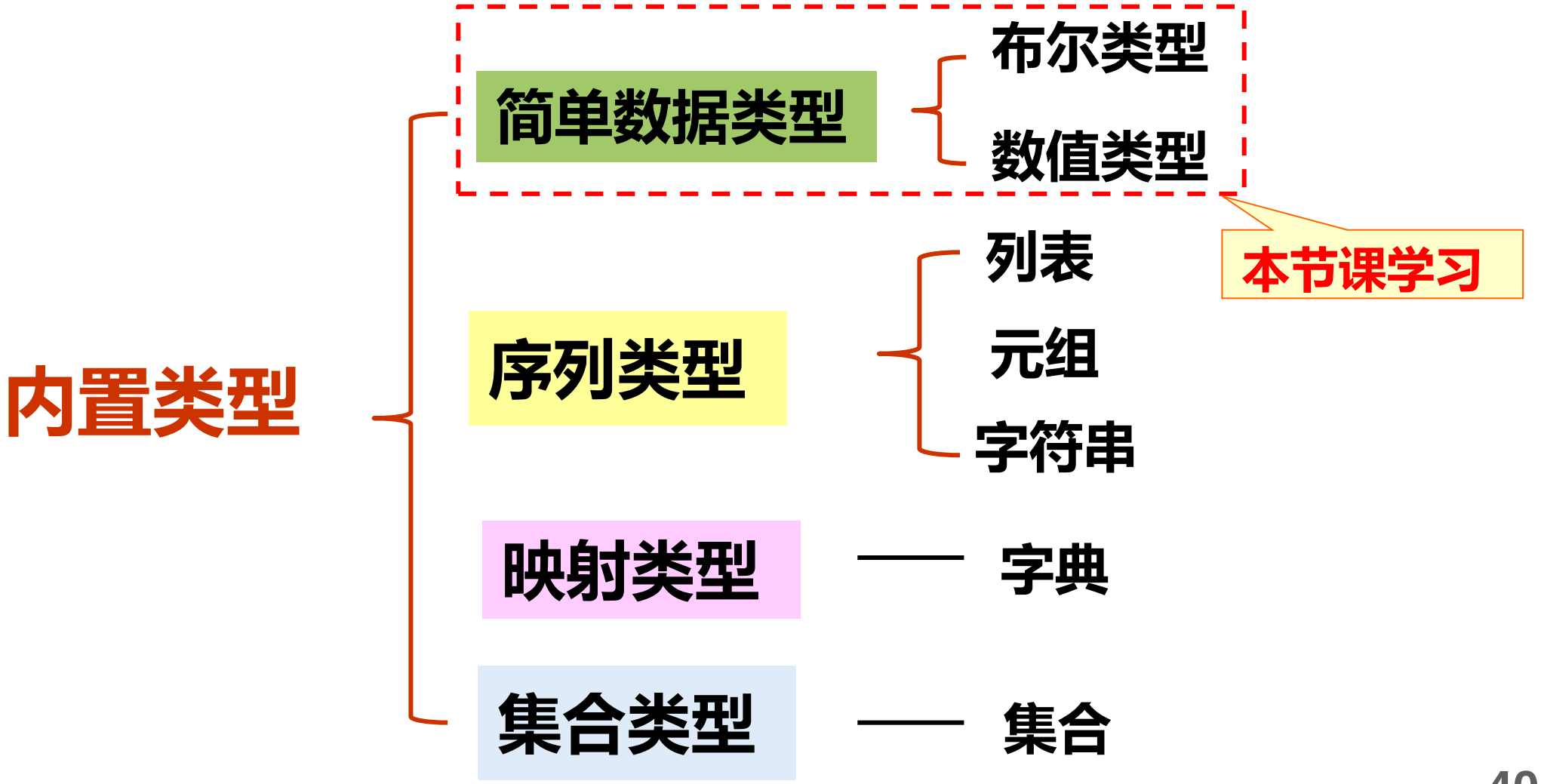
Python中的对象

- **Python程序中所有的数据均由对象或者对象间的关联关系来表示，对象几乎都具有属性（attributes）及方法（methods）**
- **每个对象包括标识（identity）、类型（type）及值(value)**
  - 在创建对象后，该对象的标识（即名称）不再改变
  - **对象的类型决定该对象支持哪些操作**，可以通过函数**int**（整型）、**float**（浮点型）、**str**（字符串）等进行**类型转换**
  - 一些对象的值可以改变，如列表list[ ]、字典dict{ }；而有些对象的值在创建后不可改变，如整型int、字符串str' '、元组tuple()。**值的可变性由对象的类型决定**

常用Python内置类型

- 在Python程序中，每个数据都是**对象**，每个对象都有自己的数据类型。不同类型有不同的操作方法
- Python提供了多个内置的数据类型（**Build-in Types**）
- Python 是一种**动态类型化**语言，**无需事先声明变量类型；**由解释器根据其值的表示形式自动匹配其类型
- 在同一程序中，**变量的类型是可以改变**（多次）的，主要通过相应的**内建函数**（如 **int()**、**float()**、**str()**）改变

常用Python内置类型（续）



简单数据类型

- Python简单数据类型包括**布尔类型**和**数值类型**

（1）布尔类型（bool type）

- Python中最简单的内置类型，该类型的**对象值**只可能为**True**或**False**（称为**布尔值**）
  - 【例】逻辑表达式“x and y”中，x、y都是布尔类型
- **表达式可以评值为布尔值**
  - 【例】将关系表达式“10 < 11”的值评值为“True”

```
>>> b=10<11
>>> print(b)
True
```

（2）数值类型

（2）数值类型

- **int**（整型），**float**（浮点型），**complex**（复数型）
- 除复数型外，所有的数值类型都支持**算术运算**
- **在Python中，参数或变量不需要事先声明类型**
  - 一个数字或变量，根据数字或变量被赋值的数字的**书写形式**，Python就可以知道它的类型
- 同样一个数字，如果写成不同的形式，则表示不同的数值类型
  - 例如，4表示整数，4.0表示实数

如何获取对象的类型？

类型	描述	语法示例
int（整型）	无小数部分的数	42， -5，1024
float（浮点型）	有小数部分的数	42.5，-5.25，**4.25e-3（科学计数法）**或4.25*10**(-3)（即$4.25*10^{-3}=0.00425$）
complex（复数型）	实数（整数或浮点数）与虚数的和	59+4j，42j

- **使用Python的内建函数type可以获取对象的类型**

```
>>> type(4)
<class 'int'>
>>> type(4.0)
<class 'float'>
>>> type(True)
<class 'bool'>
```

北京航空航天大學
BEIHANG UNIVERSITY

四、Python词法

北京航空航天大学

词法符号

- **Python程序由空白符分隔的词法符号流组成**
- **词法符号**包括
  - **空白符**（White space）
  - **常量**（Constant）
  - **操作符**（运算符，Operator）
  - **标识符**（Identifier）
  - **关键字**（Key word）
  - **注释**（Comment）

1、空白符

- **空白符**包括**空格**、**Tab**、**换行**
- 起分隔词法符号的作用；同时使代码错落有致，提高**可读性**
- **Python可以智能缩进**
  - ✓ 用def关键字定义一个函数后，回车，则自动缩进4格，以输入函数中的执行语句（**函数体**）
  - ✓ for语句、while语句等，也是在第2行自动缩进4格，以便输入**循环体**

```
for n in range(2,25):                    #for循环语句，range为范围函数
    fib[n]=fib[n-1]+fib[n-2]             #计算第n项斐波那契数
    print('第',n,'个月：',fib[n])          #打印
```

2、常量

- **常量：在程序运行过程中，其值不能改变的量**
- Python的常量包括**整数**和**实数**
- **常量可以表示为二进制、十进制、八进制或十六进制**
  - 一个数字，如果仅仅由阿拉伯数字0~9以及小数点组成，如483、2795、343.7，则表示**十进制数**
  - **二进制数用0b开头**，如**0b10**=2
  - **八进制数用0o开头**，如**0o10**=8
  - **十六进制数用0x开头**，如**0x10**=16；**0xAF**=175（**A~F**或**a~f**代表十进制数10~15）

如何将一个十进制数转换其他进制的字符串?

(1) 如何将一个十进制数转换为二进制表示的字符串?

bin(number)

(2) 如何将一个十进制数转换为八进制表示的字符串?

oct(number)

(3) 如何将一个十进制数转换为十六进制表示的字符串?

hex(number)

```
>>> bin(15)
'0b1111'
>>> oct(15)
'0o17'
>>> hex(15)
'0xf'
```

3、运算符

- **运算符**也称为**操作符**，是**Python预定义的函数符号，这些函数对被操作的对象进行规定的运算，得到一个结果**
  - **逻辑**运算符
  - **成员测试**运算符
  - **关系**运算符（布尔比较运算符）
  - **位**运算符
  - **移位**运算符
  - **算术**运算符等
- 共36个

运算符（续1）

运 算 符	描　　述		优 先 级
lambda	lambda表达式		1
or	逻辑或	逻辑运算符	2
and	逻辑与		3
not	逻辑非		4
in	成员资格测试	成员测试运算符	5
not in	非成员资格测试		5
is	一致性测试		6
is not	非一致性测试		6

运 算 符	描　　述		优 先 级
<	小于	关系运算符	7
>	大于		7
<=	小于或等于		7
>=	大于或等于		7
==	等于		7
!=	不等于		7
|	按位或	位运算符	8
^	按位异或		9
&	按位与		10

运算符（续2）

运 算 符	描 述	优 先 级
&	按位与	10
<<	左移	11
>>	右移	11
+	加法	12
-	减法	12
*	乘法	13
/	除法	13
%	求余	13
+	一元一致性	14
-	一元不一致性	14
~	按位补码	15
**	幂	16
x.attribute	特性引用	17
x[index]	项目访问	18
x[index1:index2[:index3]]	切片	19
f(args...)	函数调用	20
(...)	将表达式加圆括号或元组显示	21
[...]	列表显示	22
{key:value, ...}	字典显示	23
'expressions...'	字符串转换	24

移位运算符

算术运算符

算术运算符使用说明

运算符	含义	示例	说明
+	加法	1+2=3 1.0+2=3.0 1+2.0=3.0	如果参与运算的两个数都是整数，则结果也是整数； 如果参与运算的数中有一个为浮点数，则结果也是浮点数
-	减法	8-5=3 8.0-5=3.0 8-5.0=3.0	如果参与运算的两个数都是整数，则结果也是整数； 如果参与运算的数中有一个为浮点数，则结果也是浮点数
*	乘法	3*5=15 3.0*5=15.0 3*5.0=15.0	如果参与运算的两个数都是整数，则结果也是整数； 如果参与运算的数中有一个为浮点数，则结果也是浮点数
/	**除法**	1/2=0.5 10/3=3.3333333333333335 1/2.0=0.5	普通除法，两个整数相除即使能除尽，**结果**也为**浮点数**。如果除不尽，保留**16**位小数

算术运算符使用说明（续）

运算符	含义	示例	说明
//	**整除**	1//2=0 10//3=3 1.0//2.0=0.0	无论对整数或者浮点数，都是整除，只保留**商的整数部分**。即使是浮点数，也会执行整除，但结果形式也为浮点数：1.0//2.0=0.0
%	求模，取余	1%2=1 10%3=1 2.75%0.5=0.25	取余运算符对浮点数同样适用
**	幂（乘方）	2**3=8 (-3)**2=9 **9**(1/2)=3**	幂运算符比取反（一元减运算符）优先级高：-3**2等同于-（3**2）=-9。 幂运算可以实现**求平方根**

运算结果的类型与操作数相同

- **思考：普通除法与整除有何区别?**

【讨论2】

- **除法“/”**：普通除法。商为多少，全部保留。无论能否除尽，**结果都为浮点数**
- **整除“//”**：无论对整数或者浮点数，都**只保留**商的**整数**部分。**运算结果的类型与操作数相同**

- **请思考，以下几个表达式的计算结果是什么?**

（1）10//4 =?　　10//4.0=?

（2）10/4= ?　　10/4.0 =?　　4/2=?

运算符的优先级

- **不同的运算符有不同的优先级**
  - 同一个表达式中如果有多个运算符，应按照**优先级高**的运算符**先运算**、**优先级低**的运算符**后运算**的规则进行
  - 例：x+y*3

    首先计算y*3，之后加上x
- **利用括号可以改变优先顺序**
  - (x+y)*3

    首先计算x+y，结果值再乘以3

4、表达式

- 由对象和运算符（操作符）组成的式子称为表达式
- 表达式包括
  - 算术表达式
  - 关系表达式
  - 布尔表达式

- 采用不同的运算符，则构成不同的表达式

（1）算术表达式

- **算术表达式**：**用算术运算符连接的表达式**
  - “4+5” 指示int型对象9
  - “4.00+5.00” 指示float型对象9.00
  - (2 + tmp)**0.5
  - pi_0 * 2/tmp
- **算术表达式的运算结果为数值类型**

（2）关系表达式

- **关系表达式**：**表示两个对象之间关系的表达式**
  - 使用**关系运算符**（**<, <=, >,>=, ==, !=** ）连接
  - 当关系为**真**时表达式的值为布尔值**True**，当关系为**假**时表达式的值为布尔值**False**
  - 如“8 > 9”，其值为False，“2 < 3”，其值为True
- **关系表达式的运算结果为布尔类型**

（3）布尔表达式

- **布尔表达式（逻辑表达式）**：**由布尔运算量和逻辑运算符按一定语法规则组成的式子**
  - 参与逻辑运算的**对象**（布尔运算量）
    - ✓ **逻辑值**（True 或False）
    - ✓ **布尔变量**（又称为**逻辑**变量）
    - ✓ **关系表达式**
    - ✓ 由括号括起来的**布尔表达式**

- 编写程序时可利用**布尔表达式**或**关系表达式**实现**程序控制**（用作条件语句中的条件）

【例4. 1】布尔表达式示例

【例4.1】判断某年份year是否为闰年

2000年

- **世纪闰年**：**世纪年**（能被100整除的年份）**能被400整除**的是闰年

  year % 400==0

- **普通闰年**：**普通年**（不能被100整除的年份）**能被4整除**的年为闰年

  ✓ (year % 4==0) and (not(year % 100==0))

1904年，2004年

- **整合**：

z=(year % 400==0) or (year % 4==0) and (not(year % 100==0))

若z=**True**，说明year是**闰年**；否则不是闰年

【举手发言】

- **问题：**

判断下面哪些年份是闰年，哪些不是。为什么?

- 1600年，1900年，2400年
- 1998年，2016年，2018年，2019年，2020年

【例4.1】Python程序

例4.1-闰年.py

```
year=int(input('请输入年份：'))

z=(year % 4==0) and (not(year % 100==0)) or (year % 400==0)

if z==True:
    print('%d 是闰年' % year)
else:
    print('%d 不是闰年' % year)
```

```
>>>
请输入年份：2004
2004 是闰年
>>> ============
>>>
请输入年份：2000
2000 是闰年
>>> ============
>>>
请输入年份：1900
1900 不是闰年
>>> ============
>>>
请输入年份：2018
2018 不是闰年
>>> ============
>>>
请输入年份：2020
2020 是闰年
>>> ============
>>>
请输入年份：2016
2016 是闰年
```

5、标识符

- **标识符：用户编程时给Python语言描述的对象所起的名字。**标识符可由**字母**、**数字**、**下划线**和**$符号**构成
- 如**变量名**、**常量名**、**函数名**等
- **定义标识符时应遵循如下规则**

① **首字符必须是字母或下划线，不能是数字或$符号！**

② 标识符是**区分大小写**的：“LENGTH”与“length”不同

③ **标识符不要与关键字同名！**

变量及变量的赋值

- **变量：在程序运行过程中，其值可以改变的量**
- 变量提供将**名**（name）与**对象**（object）关联的方式
  - pi_0 = 2　　# pi的初值
  - pi_1 = 3　　# pi的计算值
  - n = int(input())
- **变量是代表（或者引用）某值的名字**
- **赋值：将某个值（或对象）赋给某个变量**，或者说，**将变量绑定到某个值（或对象）上面**
  - 例：x=3，y=2*x

变量及变量的赋值（续）

- 在使用变量之前，必须对其赋值！
- **变量被赋值之后，才可以在表达式中使用该变量**

```
>>> x=3  #赋值语句
>>> x*2
6
```

```
>>> y*3

Traceback (most recent call last):
  File "<pyshell#18>", line 1, in <module>
    y*3
NameError: name 'y' is not defined
```

关于变量的使用规则

- 一个变量记一个名，赋值语句将**赋值操作符“=”**左侧的**名**与“=”右侧的表达式所指示的**对象**进行关联

  - 在Python中，变量通常以**字母**或者“_”开始，变量名称可以包含**字母、数字**以及特殊字符_
  - 变量**不能以数字或$符号开头**
  - 系统所使用的**关键字**（如if, for, def, import），**不能作为变量名使用**
  - 变量是**大小写敏感**的，如world和World代表不同的变量
  - 通常**使用小写字母**进行变量命名

6、赋值语句

- **赋值语句**：**将等号右边的值（或对象）赋给等号左边的变量的语句**

  格式 `<variable name> = <expression>`

  - 通过赋值语句，可以设置变量的**初始值**，也可以将**新值**关联至变量
    - **变量初始化：第一次对某变量名进行赋值**
    - **变量引用：变量初始化之后，在程序后续的表达式中使用该变量名**

赋值语句示例

- **在一条赋值语句中给多个变量赋值**

```
x,y = 1,2
print ('x=',x)        x= 1
print ('y=',y)        y= 2

x,y = y,x
print ('x=',x)        x= 2
print ('y=',y)        y= 1
```

交换两个变量的值

思考：还可以采用什么方法?

注意：分隔符只能是英文半角逗号

【例4.2】赋值语句使用示例

【例4.2】利息。已知本金m（美元），年利率（人民币）r，存款年数n，存款日美元兑人民币汇率a。假如小明今天把**美元**换成人民币再**存**入银行，n年后从银行取出。每年利息按**复利**计算（即每年的利息也作为下一年的本金计算）。

试输出小明取款日拿到的**人民币**，**保留两位小数**。

【例4.2】设计思路

设计思路

- 根据题意，首先将美元兑换成人民币存入银行，已知存款日美元兑人民币汇率为a

  则本金m（美元）换成**人民币后 = m*a**

- 每年利息按复利计算，已知年利率（人民币）为r，存满一年的**本金+利息**为多少？

$$x = (m \times a) \times (1 + r)$$

- 存满**2年**的本金+利息为多少？

$$x = [(m \times a) \times (1 + r)] \times (1 + r) = (m \times a) \times (1 + r)^2$$

- 进一步地，存满**n年**的本金+利息为多少？

$$x = (m \times a) \times (1 + r)^n$$

【例4.2】 Python程序

例4.2-利息.py

```
#例4.2-利息.py

#  （1）输入
m = float(input())                    #本金m（美元）
r = float(input())                    #年利率
n = int(input())                      #存款年数
a = float(input())                    #存款日美元兑人民币汇率

#  （2）计算存款n年复利后的人民币
x = (m*a) * ( (1+r) ** n ) #把美元换成人民币后，存n年

#  （3）输出
print('%d年后共有人民币x=%.3f' % (n,x))
```

```
>>>
1000
0.02
10
6.5
10年后共有人民币x=7923.46
>>>
```

7、关键字

- **关键字：是某种程序设计语言中事先定义好的确认符，用来组织语言结构，或者定义内建函数**
- **关键字**含有内建的含义，**用户不能随便使用作为标识符**
- 不同版本的Python其关键字不尽相同
- Python3.4.0中的关键字
  - 组织语言结构的单词
  - 所有内建函数名

  abs，max，min，bin，

  oct，hex，pow，int，float等

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

8、注释

- **增强代码可读性**的有效方法是在程序中添加必要的注释（对程序中**每个变量、关键语句必须添加注释！**）
- 在Python中，**注释不会被解释执行**
  - “#”为**单行注释符**，跟在每一行的末尾，后跟注释文字。注释行也可以单独作为一行
  - **多行注释符**：**一对三引号'''**

当需要注释掉某一段程序时，使用一对三引号'''

8、注释（续）

建议把题目要求和解题思路写在程序最前面；把测试样例写在最下面

```
#例4.2-利息.py
'''
已知本金（美元）m，年利率（人民币）r，存款年数n，存款日美元兑人民币汇率a，取款日
假如小明今天把美元换成人民币再存入银行，n年后从银行取出。每年利息按复利计算（即
试输出小明取款日拿到的人民币，保留两位小数。
'''

#（1）输入
m = float(input())  #本金m（美元）
r = float(input())  #年利率
n = int(input())    #存款年数
a = float(input())  #存款日美元兑人民币汇率

#（2）计算存款n年复利后的人民币
x = (m*a) * ( (1+r) ** n ) #把美元换成人民币后，存n年

#（3）输出
print('%d年后共有人民币x=%.2f' % (n,x))

'''
输入样例:
1000
0.02
10
6.5
输出样例:
7923.46
'''
```

- **如何打印由多行字符串组成的图形?**
  - 采用一对**三引号**将要输出的字符串括起来，然后赋给一个变量
  - 再采用**格式化字符串**方法，打印该变量

【例4.3】多行注释符使用示例

【例4.3】将键盘输入的一个字符插入到固定的多行字符串的中间位置，打印输出。

例4.3-print-多行字符串.py

```
x = input()        #输入一个字符

display = """--------------------
--------
------
----                  -
--         %s         --
-                  ----
                 ------
               --------
--------------------"""
print(display % x) #x的值插入%s所在位置
```

```
>>>
B
--------------------
--------
------
----                  -
--         B          --
-                  ----
                 ------
               --------
--------------------
>>>
```

- %称为**字符串格式化操作符**，其**右边**为需要**格式化的对象**，**左边**的**%s**表示要插入的对象的**位置**和**格式**。这里即是将x的值插入变量display中“%s”所在位置。
- print语句输出时，是将三引号中的内容，**原封不动**地打印出来。必须将要打印的内容，**紧跟在第一个在三引号的后面**！而不能放在三引号的下一行！否则，打印时最前面会比预想的多一行空行。同理，**第二个三引号也必须紧跟在要打印内容的末尾**，而不是下一行。否则，打印内容的最后，会多一个空行。提交到OJ上时会WA。

关于学习Python的几点建议

1、重视课前、课上和课后各个环节

- **课前**预习课件，观看MOOC视频，仔细阅读教材相关章节
- **上课**认真听讲，积极**思考**，参与**讨论**和**练习**
- **课后**认真复习**精讲课件**，把课件上**例子**自己独立**做一遍**，仔细体会语法

仅仅带着耳朵听课是不够的！
必须预习+复习+实践！

2、实验怎么做？

2、实验怎么做?

提示很重要!

- **实验课前复习语法**；仔细阅读**实验指导**，了解**题目要求**（包括输入、输出的类型），**理解题意**
- **实验课上**逐题编程，先在本地调试，**所有**测试样例通过后再提交到OJ上测试
- **若遇到程序有问题**
  - ✓ **不要马上问老师或助教**——因为**考试时你只能靠自己**!
  - ✓ 先自己冷静下来，仔细阅读系统给出的**出错信息**，分析可能错在什么地方

2、实验怎么做？（续）

- ✓ 在可能有问题的地方**插入print**，打印**中间结果**，有助于查错
- ✓ 查阅课件或教材中相关知识和示例
- ✓ 如果还不能解决，一定要现场或在课程群里提问。**不要在一道题上纠缠太久！**
- ✓ 可以贴出**出错信息**和**相关源代码**，但**不要直接贴出全部代码**
- ✓ 可以私信发给助教或老师

- **实验课后**，尽快完成未完成的题，在OJ上提交
  - **确保每次实验的每道题都能独立完成**

北京航空航天大學
BEIHANG UNIVERSITY

本讲小结

北京航空航天大学

一、Python的内建函数

- **Python解释器提供了69个内建函数，用户可以直接使用**
  - **数学运算类**：**abs(x)**求绝对值；**int(x)**向下取整，**float(x)**将一个字符串或数转换为浮点数，**round(x,[d])**把浮点数四舍五入为最接近的整数值；**bin(x)**将整数x转换为二进制字符串，**oct(x)**将整数x转换为八进制字符串，**hex(x)**将整数x转换为十六进制字符串
  - **集合类操作**：**len(x)**返回序列中所包含元素的数量；**max(x)**返回集合中的最大值，**min(x)**返回序列中最小的元素，**str(x)** 转换为string类型
  - **IO操作**：**input**输入合法的Python表达式；**print**打印表达式

二、Python的标准库

- **标准库**是Python的标准安装提供的一组**模块**。每个模块中提供了若干**函数**，方便用户调用它们，进行相应运算
  - **math和cmath模块**：包含用于计算实数和复数的数学函数（如向下取整函数**floor**、求平方根函数**sqrt**、正弦函数**sin**等）
  - **random模块**：包含产生随机数的函数（如**randint**函数， **uniform**函数）
  - **copy模块**：包含deepcopy函数（深拷贝）
  - **datetime模块**：提供显示日期和时间的格式化方法
- 要想使用标准库中的某个函数，必须先使用“**import <库名>**”语句导入该库；再使用形如“**<库名>.<函数>(参数)**”的语句对该函数进行调用

总结：几个取整函数的比较

- **int(x)**：把浮点数**向下取整**为整数（简单舍弃小数部分，**只取整数部分）**；或把一个数字字符串转换为整数
- **round(x,[d])**：把浮点数**四舍五入**为最接近的整数值。d是保留小数位数，默认值为0。在Python 3中该函数表示对x**四舍六入五凑偶**
  - 如round(3.5)=4，round(2.5)=2
- **math.ceil(x)**：浮点数**向上取整**
- **math. floor(x)**：浮点数**向下取整**
  - 对于正的浮点数，运算结果与int函数相同：int(12.3)= math.floor(12.3)=12
  - 对于**负**的**带小数**的浮点数，运算结果与int函数**不同**： int(-12.3)=-12，**math.floor(-12.3)=-13**

三、简单数据类型

- Python简单数据类型包括**布尔类型**和**数值类型**
  - **布尔类型**：Python中最简单的内置类型，该类型的**对象值**只可能为**True**或**False**（称为**布尔值**）
  - **数值类型**：**int**（整型），**float**（浮点型），**complex**（复数型）
    - ✓ 除复数型外，所有的数值类型都支持**算术运算**
- **在Python中，参数或变量不需要事先声明类型**
- 使用Python的内建函数**type**可以获取对象的类型

四、Python词法

- Python程序由空白符分隔的词法符号流组成
- 词法符号
  - 空白符（空格、Tab、换行）
  - 常量（整数和实数）
  - 操作符
  - 标识符
  - 关键字
  - 注释
- 可以分别使用内建函数bin、oct、hex，将十进制数字转换为二进制、八进制和十六进制表示的字符串

运算符

- **运算符（操作符）**，是**Python预定义的函数符号，这些函数对被操作的对象进行规定的运算，得到一个结果**
  - **逻辑**运算符
  - **成员测试**运算符
  - **关系**运算符（布尔比较运算符）
  - **位**运算符
  - **移位**运算符
  - **算术**运算符等
- **普通除法“/”与整除“//”的区别**
- **不同的运算符有不同的优先级**

标识符

- **标识符：用户编程时给Python语言描述的对象所起的名字。**标识符可由**字母**、**数字**、**下划线**和**$符号**构成
- 如**变量名**、**常量名**、**函数名**等
- **定义标识符时应遵循如下规则**

① **首字符必须是字母或下划线，不能是数字或$符号！**

② 标识符是**区分大小写**的：“LENGTH”与“length”不同

③ **标识符不要与关键字同名！**

关键字与注释

- **关键字：**是某种程序设计语言中事先定义好的确认符，用来组织语言结构，或者定义内建函数
- 关键字含有内建的含义，**用户不能随便使用作为标识符！**
- **注释**：程序中的解释性文字，用来说明变量或语句的含义和作用，程序运行时不会被执行
  - 对程序中**每个变量、关键语句必须添加注释！**

- **建议把题目要求和解题思路写在程序最前面；把测试样例写在最下面**
- **当需要注释掉某一段程序时，使用多行注释符（一对三引号'''）**