

Overview

- 基本数据类型
 - 数学库math的使用
 - 内置转换函数
 - 赋值，条件和for语句
 - 格式化输出
 - 数据表示

2.1 数字类型

- 计算机内部用位序列表示数据，类型为这些位序列赋予了意义。
同时，类型限制了数据的取值范围。
 - 整数
 - 浮点数
 - 复数

整数

- ➊ >>>66
66
 - ➋ >>>-175
-175
 - ➌ >>>07
SyntaxError: invalid token
 - ➍ >>>0
0

二进制，八进制，十六进制

- 0b或0B 代表二进制
 - 0o或0O 代表八进制
 - 0x 或0X 代表十六进制
 - >>>0b10
2
 - >>>0o10
8
 - >>>0x10
16

整数可以表示很大的数

- >>>google=10**50
 - >>>google

运算符

运算符	说明	示例	运算结果
+	加法	5+10	15
-	减法	100-5	95
*	乘法	8*9	72
/	浮点数除法	100/5	20.0
//	整除	100//5	20
%	模（求余）	9%4	1
**	幂	2**3	8

浮点数

- 浮点数也就是小数
- 1.23, 3.14, -9.01
- 科学计数法:
- 1.23×10^9 就是 $1.23e9$,
- 0.000012 可以写成 $1.2e-5$ 。
- "e" 的前后都不能空, "e" 的后面要整数。

Python程序设计

8

浮点数运算

- 浮点数运算有误差
- False
- 5.428571428571429
- 浮点数的整除还是浮点数
- 5.0
- $\text{0.30000000000000004}$

Python程序设计

9

复数 $|ab|(\text{复数}) \rightarrow \sqrt{\text{取模}}$

- 所谓复数，就是由实部 (real) 和虚部 (imaginary) 两部分组成的数，虚部用 j 表示。
- $2+3j$
- $8j$
- $(7+1j)*1j$
- real 方法取实部， imag 方法取虚部， complex() 函数用于创建一个值为 $\text{real} + \text{imag} * j$ 的复数。

Python程序设计

10

数学库(math)

- math库是一个数学库，包含了很多的数学常数和数学函数
- 要使用math库，先要用“import math”语句引入math库
- math.pi
- math.sqrt(9)
- math.floor(3.0)
- math.ceil(3.0)
- math.fabs(3.0)
- math.factorial(5)

Python程序设计

11

函数和方法

- 方法是与数据对象关联的函数，是在相对应数据对象名字空间中定义的函数
- 用“.”记法调用的函数也称方法
- math.cos(5)
- cos() 是函数， cos() 也称为 math 对象的方法

Python程序设计

12

2.2 字符串

- 字符串是以"或""括起来的任意文本，比如'abc', "xyz"等等。请注意，"或""本身只是一种表示方式，不是字符串的一部分
- 'hello world'
- "hello world"
- "#空字符串"

Python程序设计

13

多行字符串

- ◎ >>>'''hello python
- ◎ 人身苦短
- ◎ 我用python'''
- ◎ >>>'hello python\n 人身苦短\n 我用python'
- ◎ 可用这种方式表示多行注解

Python程序设计

转义字符

转义字符	描述
\	\
'	'
\"	"
\a	响铃
\b	退格(Backspace)
\n	换行
\t	横制表符
\r	回车
\f	换页
\ooo	最多三位八进制数, 例如: \12代表换行
\xyy	两位十六进制数, 例如: \x0a代表换行

Python程序设计

15

转义字符举例

- ◎ >>>" \12" #八进制
- ◎ '\n'
- ◎ >>>" \x0a" #十六进制
- ◎ '\n'
- ◎ >>>" \141" #八进制
- ◎ 'a'
- ◎ >>> print("\\")
- ◎ \
- ◎ >>> "\\""
- ◎ ""

Python程序设计

字符串运算符: +, *

- ◎ >>>"人生苦短"+" 我用Python" "人生苦短 我用Python"
- ◎ >>>'2'*3
'222'
- ◎ type 函数是一个内置的函数, 调用它就能知道想要查询对象的类型信息
- ◎ >>>type(1)
<class 'int'>
- ◎ >>>type("python")
<class 'str'>

Data and Computation

17

2.3 布尔类型、空类型和列表

- ◎ 布尔类型的变量只有True、False两种值, 要么是True, 要么是False (请注意大小写)
- ◎ 布尔值可通过逻辑运算符和关系运算符计算出来。关系运算符是 <、<=、>、>=、==和!=, 逻辑运算符是and、or和not。

Python程序设计

关系运算符

运算符	表达式	含义	实例	结果
==	x==y	x 等于 y	"ABCD" == "ABCDE"	False
!=	x!=y	x 不等于 y	"ABCD" != "abcd"	True
>	x>y	x 大于 y	"ABC" > "ABD"	False
>=	x>=y	x 大于等于 y	"123" >= "23"	False
<	x<y	x 小于 y	"ABC" < "DEF"	True
<=	x<=y	x 大于等于 y	"123" <= "23"	True

表 2-5 关系运算符

18

Python程序设计

19

逻辑运算符

逻辑量1	逻辑量2	结果	逻辑量1	结果
False	False	False	False	True
False	True	False		True
True	False	False	True	False
True	True	True		True
and 运算			not 运算	
逻辑量1	逻辑量2	结果	not 运算	
False	False	False		
False	True	True		
True	False	True		
True	True	True		
or 运算				

Python程序设计

20

关系运算符实例

- ① `>>> 1<3<5` #等价于1<3 and 3<5
- ② `True`
- ③ `>>> 3<5>2`
- ④ `True`
- ⑤ `>>> 1>6<8`
- ⑥ `False`
- ⑦ `>>> import math` #sqrt是math模块下的函数，导入math模块
- ⑧ `>>> 1>6<math.sqrt(9)` #注意：不能直接对整数比较
- ⑨ `False`
- ⑩ `>>> 'Hello'> 'world'` # ascii('H') = 72 < ascii('w') = 119
- ⑪ `False`
- ⑫ `>>> 'Hello'> 3` #字符串和数字不能比较
- ⑬ `TypeError: unorderable types: str()>int()`

Python程序设计

21

逻辑、关系运算符实例

- and 最后一个真第一个假*
- #注意，此时并没有定义变量a*
- or 最后一个假第一个真*
- ① `>>> 3>5 and a>3`
 - ② `False`
 - ③ `>>> 3>5 or a>3` #3>5的值为False，所以需要计算后面的表达式
 - ④ `NameError: name'a'is not defined`
 - ⑤ `>>> 3<5 or a>3` #3<5的值为True，不需要计算后面的表达式
 - ⑥ `True`
 - ⑦ `>>> 3 and 5` #最后一个计算的表达式的值作为整个表达式的值
 - ⑧ `5`
 - ⑨ `>>> 3 and 5>2`
 - ⑩ `True`
 - ⑪ `>>> not 3`
 - ⑫ `False`
 - ⑬ `>>> not 0`
 - ⑭ `True`
 - ⑮ `>>> 3 and 0 and 5` #练习

Python程序设计

22

空类型

- 空类型*
- ① 如把类型用集合表示：
 - ② `datatype={整数, 浮点数, 复数, 字符, ...}`
 - ③ `datatype的幂集: {}, {整数}, {浮点数}, ...`
 - ④ 幂集的空集合代表什么？
 - ⑤ 空类型只有一个值，空值。
 - ⑥ 空值是Python里一个特殊的值，用None表示。None不能理解为0
 - ⑦ `>>>bool(None)`
 - ⑧ `False`
 - ⑨ `>>>None ==0`
 - ⑩ `False`

Python程序设计

23

运算符的优先级和结合性

优先级（1最高，8最低）	运算符	描述	结合性
1	<code>x**y</code>	幕	从右向左
2	<code>+x,-x</code>	正，负	
3	<code>x*y, x/y, x%y</code>	乘，除，取模	从左向右
4	<code>x+y, x-y</code>	加，减	从左向右
5	<code>x<y, x<=y, x==y, x!=y, x>=y, x>y</code>	比较	从左向右
6	<code>not x</code>	逻辑否	从左向右
7	<code>x and y</code>	逻辑与	从左向右
8	<code>x or y</code>	逻辑或	从左向右

Python程序设计

24

优先级和结合性实例

- ① `>>> 3+5 * 4` #先乘后加
- ② `23`
- ③ `>>> 5 * 3/2` #从左向右
- ④ `7.5`
- ⑤ `>>> 2**3**2` #从右向左
- ⑥ `512`
- ⑦ `>>> 3<5 or a>3` #从左向右
- ⑧ `True`

Python程序设计

25

列表

- 列表可以由零个或多个元素组成，元素之间用逗号分开，整个列表被方括号所包裹
- >>> empty_list = [] #空列表
- >>> weekdays = ['Monday', 'Tuesday', '\nWednesday', 'Thursday', 'Friday']
- >>> weekdays[2] #下标从0开始
'Wednesday'

Python程序设计

26

列表运算

- >>> weekdays[4]=5
- >>> weekdays
['Monday', 'Tuesday', 'Wednesday',\n'Thursday', 5]
>>> [1,2,3]<[1,2,4] #比较列表大小
True
- >>> [1,2,3]+['c','java','python'] #加法
[1, 2, 3, 'c', 'java', 'python']
- >>> [1]*10 #可以用作列表初始化
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

Python程序设计

27

2.4 内置转换函数

函数名	含义
bool	根据传入的参数的逻辑值创建一个新的布尔值
int	根据传入的参数创建一个新的整数
float	根据传入的参数创建一个新的浮点数
complex	根据传入参数创建一个新的复数
str	创建一个字符串
ord	返回Unicode字符对应的整数
chr	返回整数所对应的Unicode字符
bin	将整数转换成2进制字符串
oct	将整数转化成8进制数字符串
hex	将整数转换成16进制字符串
list	根据传入的参数创建一个新的列表

Python程序设计

28

内置转换函数实例

- >>> bool('str')
True
- >>> int(3.6)
3 (注释)
- >>> float(3)
3.0
- >>> complex(1,2) #传入数值创建复数
(1+2j)

Python程序设计

29

int函数用法

- >>> int() #不传入参数时，得到结果0。
0
- >>> int("02") #去掉0
2
- >>> int(" 35 ") #去掉空格
35
- >>> int(" 3 5 ") #无法转换
Traceback (most recent call last):
File "<pyshell#0>", line 1, in <module>
 int(" 3 5 ")
ValueError: invalid literal for int() with base 10: ' 3 5 '
>>> int("35",8) #八进制
29

Python程序设计

30

ord函数和chr函数

- >>> ord('a') #ASCII码值
97
- >>> ord('中') #汉字‘中’的Unicode码
20013
>>> chr(97) #参数类型为整数
'a'
8与"8"如何转换？

Python程序设计

31

bin函数，oct函数，hex函数

- ① >>> bin(3) #0b为默认
'0b11'
- ② >>> oct(10)
'0o12' → 10r
- ③ >>> hex(15)
'0xf'

Python程序设计

32

str函数和list函数

- ④ >>> str(123)
'123'
- ⑤ list('abcd') #传入字符串，创建列表
['a', 'b', 'c', 'd']

Python程序设计

33

表达式

- ⑥ 表达式是可以计算的代码片段，由常量、变量和运算符或函数按规则构成，返回运算结果
- ⑦ >>> 7 #表达式
7
- ⑧ 计算表达式 cos(a(x+1)+b)/2, a等于2, x等于5, b等于3
>>> import math
>>> math.cos(2*(5+1)+3)/2
-0.37984395642941066
- ⑨ >>> 0 and 1 or not 2<True

⑩ 计算表达式：当n是奇数时为1，偶数时为0
>>> n=int(input())
5
>>> 1 if n%2==1 else 0 #条件表达式
1

Python程序设计

34

条件表达式表示短路运算

- ⑪ a or b
(如果a为真，则b不执行)
- ⑫ a if a为真 else b
a为假b为真
- ⑬ a and b
如果a假
- ⑭ b if a为真 else a
(b if a为真 else a)

Python程序设计

35

2.5 语句

- ⑮ Python语言常用的有赋值、if语句和for语句。语句通常是一行一条语句。如一行中有多条语句，则用分号（；）分开，如语句太长要跨行时，可以用续行符（\）跨行表示一个语句。

赋值语句

赋值语句用于将名称绑定到特定对象
(值)

基本形式是“变量=值”的形式

Python程序设计

36

【例 2-1】 基本赋值语句

```
x=1  
y=2  
k=x+y  
print(k)
```

程序输出：

3

Python程序设计

37

多变量赋值

- >>>x,y=4,8
>>>print(x,y)

4 8

例 2-2 交换a,b值

```
a=int(input())  
b=int(input())  
print(a,b)  
a,b=b,a #a和b交换  
print(a,b)
```

链式赋值

- >>>a=b=c=5
- >>>print(a,b,c)
- 5 5 5
- >>>b=b+6
- >>>print(a,b,c)
- 5 11 5
- a=(b=5)结果?

错误! b=5是语句, 无返回值,

赋值和“+”、“-”、“*”和“\”组合

- >>>i=2
- >>>i*=3
- >>>i
- 6
- >>>j=5
- >>>j*=3+1 # j=j*(3+1)
- >>>j
- 20

if语句

- if 逻辑表达式:
 语句块1
- else:
 语句块2

```
x=int(input())  
if x%2==0:  
    print("偶数")  
else:  
    print("奇数")
```

计算水费

- 为鼓励居民节约用水, 自来水公司采取按用水量阶梯式计价的办法, 居民应交水费y(元)与月用水量x(吨)相关: 当x不超过15吨时, $y=4x/3$; 超过后, $y=2.5x-17.5$, 小数部分保留2位。请编写程序实现水费的计算。

```
x=float(input())  
if x<=15:  
    y=4*x/3  
else:  
    y=2.5*x-17.5  
print("水费 = {:.2f}".format(y))
```

for语句

- for variable in 列表:
 语句块

for后面的变量先被赋给列表的第一个值, 并执行下面的代码块。然后变量被赋给列表中的第二个值, 再次执行代码块。该过程一直继续, 直到穷尽这个列表。语句块缩进表示它是属于for代码块

【例 2-5】遍历列表

```
for i in [1,2,3,4]:  
    print(i)
```

输出：
1
2
3
4

Python程序设计

44

range函数

- ◎ range(start, stop, step)。
- ◎ start: 计数从start开始。默认是从0开始。
例如range(5)等价于range(0, 5)
- ◎ stop: 计数到stop结束, 但不包括stop。
例如: list(range(0, 5))是[0, 1, 2, 3, 4]
~~没有5~~
- ◎ step: 步长, 默认为1。例如: range(0, 5)等价于range(0, 5, 1)

Python程序设计

45

range函数实例

- ◎ >>> list(range(10)) # 从 0 开始到 10, 不包括10
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- ◎
- ◎ >>> list(range(1,11)) # 从 1 开始到11
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- ◎
- ◎ >>> list(range(0, 30, 4)) # 步长为 4
[0, 4, 8, 12, 16, 20, 24, 28]
- ◎
- ◎ >>> list(range(0, -10, -1)) # 步长为负数
[0, -1, -2, -3, -4, -5, -6, -7, -8, -9]

Python程序设计

46

sum函数

它可以求列表的和
如何求 $1+2+3+\dots+10$ 的和?

```
>>>sum([1,2,3,4,5,6,7,8,9,10])  
55  
或:
```

```
>>>sum(list(range(1,11)))
```

Python程序设计

47

【例 2-6】输入n ($n \geq 10$) ,求 $1+2+\dots+n$ 之和。

```
n=int(input())  
s=sum(list(range(n+1)))  
print(s)
```

Python程序设计

48

【例 2-7】输入n ($n \geq 5$) 求n!

```
n=int(input())  
fac=1  
for i in list(range(1,n+1)):  
    fac=fac*i  
print(fac)
```

Python程序设计

49

列表元素的计算

- >>> a=3
- >>> b=7
- >>> lst=[a+2,b]
- >>> lst
- [5,7]

Python程序设计

50

列表和for结合

- >>> a=1;b=2;c=3;d=4
- >>> [a,b,c,d]
- [1,2,3,4]
- for i in list(range(1,5)):
- print(i,end=" ")
- 1 2 3 4
- >>> lst=[i for i in list(range(1,5))]
- lst
- [1,2,3,4]

Python程序设计

51

列表推导式

- 列表推导式是从一个或者多个列表快速简洁地创建列表的一种方法，又被称为列表解析。它可以将循环和条件判断结合，从而避免语法冗长的代码，同时提高程序性能。
- [expression for item in iterable]
- **输出表达式 变量**
- >>> nl = [2*number for number in [1,2,3,4,5]]
- >>> nl
- [2, 4, 6, 8, 10]

Python程序设计

52

带条件的列表解析

- [expression for item in iterable **if condition**]
- >>> nl=[number for number in range(1,8) if number % 2 == 1] 可选
 - >>> number_list
 - [1, 3, 5, 7]

Python程序设计

53

级数求和编程的一般模式

- $a_0 + a_1 + \dots + a_{n-1} = \sum_{i=0}^{i=n-1} a_i$
- sum---- Σ
- for i in range(n)----i=0到i=n-1
- 合起来：
- sun([a_i for i in list(range(n))])
- 级数求和就变成如何写通项公式a_i

Python程序设计

54

【例 2-8】 【例 2-9】求和

- 求 $1+1/2+\dots+1/20$ 之和
- print(sum([1/i for i in list(range(1,21))]))
- 通项公式: $a_i = 1/i$
- 求 $1-1/2+1/3-1/4+\dots$ 之前n项和($n \geq 10$)
n=int(input())
print(sum([1/i if i%2==1 else -1/i for i in list(range(1,n+1))]))

Python程序设计

55

列表推导式的if条件和条件表达式同时使用

【例 2-10】求 $1-1/3+1/5-1/7+\dots-1/47+1/49$

```
• >>> [i if i%4==1 else -i for i in list(range(1,50))]\n•     if i %2==1] \n[1, -3, 5, -7, 9, -11, 13, -15, 17, -19, 21, -23, 25,\n-27, 29, -31, 33, -35, 37, -39, 41, -43, 45,\n-47, 49]\nprint(sum([1/i if i%4==1 else -1/i for i in \\\n    list(range(1,50)) if i%2==1]))
```

Python程序设计

56

求 $6+66+666+\dots+666\dots666$

• 生成5个6

• >>>int('6'*5)

66666

n=int(input())

print(sum([int('6'*i) for i in list(range(1,n+1))]))

Python程序设计

57

2.6 格式化输出

- 当输出计算结果时，经常需要控制它的显示形式
- format()函数是Python的内置函数，用来设置输出格式，返回值是字符串
- 一般形式：“需格式化字符串”.format(参数表)
- >>> name="John"
- >>> "Hello,{:>6s}".format(name)
- >>> 'Hello, {} John'
- {:6s}是格式限定符，描述参数name如何显示，普通字符串原样输出，{}表示空格

Python程序设计

58

格式限定符举例

格式限定符	输出	说明
"{:d}".format(24)	24	格式化一个整数,d代表十进制整数
"{:b}".format(45)	101101	b代表二进制整数
"{:o}".format(24)	30	o代表八进制整数
"{:x}".format(24)	18	x代表十六制整数
"{:5d}".format(24)	□□□24	宽度指定5，增加了空格
"{:4d}".format(24879)	24879	超过宽度全部输出
"{:.2f}".format(1.2449)	1.24	小数点保留2位, f代表浮点数
"{:.2e}".format(53.2453)	5.32e+01	E表示科学计数法
"{:.6.2f}".format(1.2449)	□□1.24	宽度为6，增加了空格
"{:9s}".format("hello")	hello□□□□	宽度为9,s代表字符串，左对齐
"{:>9s)".format("hello")	□□□□hello	宽度为9,右对齐

Python程序设计

59

多个参数格式化输出

- ```
• >>>x=3.14159\n• >>>y=2*x*3\n• >>>print("{0:.2f} {1:.2f}".format(x,y))\n• 3.14 18.85\n• 0和1表示format函数中的第一和第二个参数，.2f表示小数部分保留两位，四舍五入
```

Python程序设计

60

## 字符串格式化

- ```
• >>> s="hello"\n• >>> "{}".format(s)\n• 'hello'\n• >>> "{:10s}".format(s)\n• 'hello' '\n• >>> "{^10s}".format(s)\n• ' hello ' \n• >>> "{:>10s}".format(s)\n• ' hello'
```

Python程序设计

61

华氏-摄氏温度转换表

输入2个正整数lower和upper ($lower < upper < 100$)，请输出一张取值范围为 $[lower, upper]$ 、且每次增加2华氏度的华氏-摄氏温度转换表，小数部分保留一位。温度转换的计算公式： $C=5\times(F-32)/9$ ，其中：C表示摄氏温度，F表示华氏温度。

```
lower,upper=input().split() # 一行输入两个数，是字符串类型
lower,upper=int(lower),int.upper() # 字符串变成整数
for i in range(lower,upper,2):
    print(i,"{:5.1f}".format(5*(i-32)/9))
    #print("fahr={0:d} Celsius={1:5.1f}".format(i, 5*(i-32)/9))
```

程序输入：

30 40

程序输出：

30 -1.1

32 0.0

34 1.1

36 2.2

38 3.3

Python程序设计

62

2.7 位运算

⑤ 位运算符用于按二进制位进行逻辑运算，操作数必须是整数。

运算符	说明	示例
&	按位与运算符：参与运算的两个值，如果两个相应位都为1，则该位的结果为1，否则为0。类似二进制乘法。	a & b 等于 1
	按位或运算符：只要对应的二个二进位有一个为1时，则该位就为1。类似二进制加法。	a b 等于 21
^	按位异或运算符：当两个对应的二进位相异时，结果为1。	a ^ b 等于 20
~	按位取反运算符：对数据的每个二进制位取反，即把1变为0，把0变为1。	~a 等于 -6
<<	左移动运算符：运算数的各二进位全部左移若干位，<<右边的数字指定了移动的位数，高位丢弃，低位补0。	a <<2 等于 20
>>	右移动运算符：运算数的各二进位全部右移若干位，>>右边的数字指定了移动的位数。	a >>2 等于 1

Python程序设计

63

* 数据表示

数据：数字、文字、图形图像和声音等

对象：Python用对象表示数据

对象三要素：id, 对象存储的位置

type, 对象类型

value, 对象的值

对象数字“1”：(id(1), type(1), 1)

【例2-13】a & b运算的二进制表示

```
a=5
b=17
print(" ", " ", "{:>08s} ".format(bin(a)[2:]))
print("& ", " ", "{:>08s} ".format(bin(b)[2:]))
print("-----")
print(" ", " ", "{:>08s} ".format(bin(a&b)[2:]))
```

运行程序，显示：

```
 000000101
& 00010001
-----
```

Python程序设计

64

Python程序设计

65

*对象的可变性

可变对象：对象的值可变，修改它的值对象的id不变

不可变对象：对象的值不可变，修改它的值对象的id要变，即创建另一个对象

```
>>> a=5;lst=[3,9,78]
>>> print(id(a),id(lst))
1549134400 55838560
>>> a=8;lst[1]=45
>>> print(id(a),id(lst))
1549134448 55838560
```

数字、字符串是不可变对象，列表是可变对象

Python程序设计

66