

Python语言程序设计

Design and Programming of The Python Language

主讲教师：张小东

联系方式：z_xiaodong7134@163.com

答疑地点：宋健研究院514

第2章 数据表示和基本运算

主要内容

- 常量、变量和对象
- 数据类型
- 运算符
- 内置函数
- 应用实例

➤ 常量

- ◆ 常量是内存中用于保存固定值的单元，在程序中常量的值不能发生改变。
- ◆ **Python**常量包括数字、字符串、布尔值和空值等。例如，数字**7**和'**abc**'都是常量。
- ◆ 没有定义常量的保留字。
- ◆ 可以自定义一个命名常量，即有名字的常量。命名常量有一个特点，就是一旦绑定，不能更改。

```
>>> from math import *  
>>> pi  
3.141592653589793  
>>> e  
2.718281828459045
```

➤ 对象

- ◆ 某种类型事务的一个具体实例，**python—一切皆对象。**
- ◆ 获取对象的标识(**id**)和类型

```
>>> id(11)  
1657845120
```

```
>>> id('python')  
48172648
```

```
>>> type(12)
```

```
<class 'int'>
```

```
>>> type(1.2)
```

```
<class 'float'>
```

- ◆ 对象的类型是不能改变的。可改变值的对象称为可变对象(**Mutable**)，否则，称为不可变对象(**Immutable**)。

➤ 变量

- ◆ 运行过程中，值可以改变的量称为变量。
- ◆ 变量是一个标识符，通过赋值运算符(=)创建，指向一个对象。

```
>>> a=112
```

创建一个变量a

```
>>> type(a)
```

```
<class 'int'>
```

a指向整数类型对象

```
>>> id(a)
```

```
1657848352
```

```
>>> id(112)
```

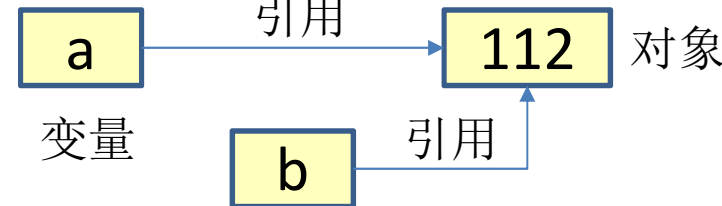
```
1657848352
```

id标识一样

```
>>> b=112
```

```
>>> id(b)
```

```
1657848352
```



===常量、变量和对象===

➤ 变量

◆python变量的特色

(1)不需要声明

(2)可以随时赋不同类型的值

(3)可以使用一个赋值符号给多个变量赋值

如, >>> x,y,z=5,7,8

>>> x

5

>>> y

7

>>> z

8

➤ 对象和数据

◆ 对象包含属性和方法

◆ 数据是某类对象的属性值

5为对象，通过a指向5

```
>>> a=5  
>>> a.bit_length() 调用对象的方法  
3  
>>> a=254  
>>> a.bit_length()  
8
```

◆ 调用对象的方法为<变量名>.<方法> (<参数>)

第2章 数据表示和基本运算

主要内容

- 常量、变量和对象
- 数据类型
- 运算符
- 内置函数
- 应用实例

===数据类型===

➤ 数字类型

◆ 整数型

0X-, 0o-

(1) 标准类型 $-2^{31} \sim 2^{31}-1$

(2) 长整型 内存的长度

◆ 浮点型

(1) 表示实数。科学记数法：<实数>E<整数>，如

1.2E3表示 1.2×10^3

(2) IEEE754 占8个字节， $-1.8E308 \sim +1.8E308$

===数据类型===

➤ 数字类型

◆ 复数类型

不能省略

复数的实部和虚部都是浮点数，且至少有一个虚部。

3+4j, 8-7j, 0.0j, 1j

区分大小写

◆ 布尔型

用于表示逻辑判断的结果，如True和False

◆ 高精度数与分数

(1) Decimal类型

```
from decimal import Decimal
```

```
Decimal(value='0', Context=None)
```

===数据类型===

➤ 数字类型

- ◆ 复数类型
- ◆ 布尔型
- ◆ 高精度数与分数

```
from decimal import Decimal  
>>> a=Decimal(1)/Decimal(3)  
>>> a  
Decimal('0.3333333333333333333333333333333')  
>>> type(a)  
<class 'decimal.Decimal'>
```

===数据类型===

➤ 数字类型

◆ 高精度数与分数

(2) 分数

from fractions import Fraction**Fraction(numberator = 0, denominator=1)****Fraction(other_fraction)****Fraction(float)****Fraction(decimal)****Fraction(string)**

```
>>> from fractions import Fraction
>>> Fraction(1,2)    #Fraction(1, 2)
>>> Fraction(12,20)  #Fraction(3, 5)
>>> Fraction(1.25)   #Fraction(5, 4)
>>> a=Fraction(1,3)
>>> b=Fraction(2,3)
>>> print(a+b)       #1
```

===数据类型===

➤ 序列类型

若干有序的数据，分为不可变序列类型和可变序列类型

- ✓ 不可变序列：数据不能再改变。字符串、元组、字节序列
- ✓ 可变序列：数据能再改变，如列表、字节数组等

◆ 字符串

Unicode字符序列。如'abcde'、"super"、"""z"""

◆ 元组类型

写在一对圆括号中，用逗号隔开的一组数据。如(1,2,3)

元组中的多个数据的类型可以不同。如(1,2,"ab","c")

===数据类型===

➤ 序列类型

◆ 字节序列

数据是一系列的字节。以**'b'**开头的字符串。

```
>>> str="abcd字节序列"
>>> print(str)
abcd字节序列
>>> a=str.encode("utf-8")
>>> a
b'abcd\xe5\xad\x97\xe8\x8a\x82\xe5\xba\x8f\xe5\x88\x97'
>>> a=str.encode("gb2312")
>>> a
b'abcd\xd7\xd6\xbd\xda\xd0\xf2\xc1\xd0'
```

===数据类型===

➤ 序列类型

◆ 列表

在一对方括号中用逗号隔开的若干数据。

多个数据的类型可以不同。如[1,2],[“February”,[1,2]]

```
>>> l=[1,2,3]
>>> l
[1, 2, 3]
>>> l[0]='zxd'
>>> l
['zxd', 2, 3]
```

===数据类型===

➤ 序列类型

◆ 字节数组

可修改的字节序列。

```
>>> str="abcd字节序列"
>>> a=str.encode("utf-8")
>>> a
b'abcd\xe5\xad\x97\xe8\x8a\x82\xe5\xba\x8f\xe5\x88\x97'
>>> a=bytearray(str,"utf-8")
>>> a[0]=101
>>> a
bytearray(b'ebcd\xe5\xad\x97\xe8\x8a\x82\xe5\xba\x8f\xe5\x88\x97')
```


===数据类型===

➤ 其他类型

◆ 集合数据类型

由若干数据组成，集合中的项目没有顺序，且不重复。

在一对大括号中，用逗号隔开，如{1, 2, 3}。

集合数据可变。

◆ 字典数据类型

字典中的每一项数据包括两部分：键和值。

字典中的值是可变的

◆ Python一切皆有类型

如模块、类、对象、函数等都属于某种类型。

===数据类型===

➤ 其他类型

◆ 字典数据类型：键和值。

```
>>> c={'name':"zhang sanfeng",'func':"太极神功"}
>>> c
{'name': 'zhang sanfeng', 'func': '太极神功'}
>>> c['name']="zhang wuji"
>>> c
{'name': 'zhang wuji', 'func': '太极神功'}
>>> c['age']=75
>>> c
{'name': 'zhang wuji', 'func': '太极神功', 'age': 75}
```

第2章 数据表示和基本运算

主要内容

- 常量、变量和对象
- 数据类型
- 运算符
- 内置函数
- 应用实例

===运算符===

➤ 基本概念

- ◆ 运算：对数据的变换
- ◆ 运算符：运算的符号
- ◆ 操作数：运算数、操作对象
- ◆ 表达式：按运算规则将值、变量和运算符组合
- ◆ 学习要点：优先级、操作数

➤ lambda

```
>>> f=lambda x,y:x+y  
>>> f(1,2)      #3  
>>> f(3,8)      #11
```

➤ if ...else

```
>>> x=1;y=2;z=3  
>>> m = x if y else z    #m=1  
>>> y=0  
>>> m = x if y else z    #m=3
```

===运算符===

➤ 逻辑运算符

◆ 符号: or, and, not

◆ 示例

```
>>>x=0;y=2
>>>x or y
>>>x and y
>>>not x
```

➤ 关系运算符

◆ 符号: <, <=, >, >=, !=, ==

◆ 示例

```
>>>x=0;y=2
>>>x > y
>>>x <= y
```

➤ 复合运算符

◆ 符号: +=, -=,, Computing & Service

➤ 成员运算符

◆ 符号: in, not in, is, is not

◆ 示例

```
>>> y={1,2,3}
>>> x in y      #True
>>> x not in y  #False
>>> x is y      #False
>>> x is not y  #True
```

➤ 位运算符

◆ 符号: |, ^, &, >>, <<, ~

➤ 算术运算符

◆ 符号: +, -, *, /, %, //, **

===运算符===

➤ 运算符优先级

由高到低

- ◆ 函数调用、寻址、下标
- ◆ 幂运算 (******)
- ◆ 翻转运算符 (**~**)
- ◆ 正负号 (**+X, -X**)
- ◆ 算术运算符 (***, /, %, //**)
- ◆ 算术运算符 (**+, -**)
- ◆ 位运算符 (**>>, <<; &, |, ^**)
- ◆ 关系运算符 (**<, <=, >, >=, !=, ==**)

◆ 赋值运算符

(**= %= /= //= -= += *= **=**)◆ 同一性运算符 (**is, is not**)◆ 成员运算符 (**in, not in**)◆ 逻辑运算符 (**not, and, or**)

```
>>> a='A'
>>> b=0
>>> c=[1,2,3,0]
>>> a and c
>>> c and a
>>> a and b and c
>>> not b and c
>>> c and not b
```

===运算符===

思考题：求任意三角形的面积

提示：设任意三角形的面积A，半周长为s，则有：

$$A = \sqrt{s \times (s - a) \times (s - b) \times (s - c)}$$

第2章 数据表示和基本运算

主要内容

- 常量、变量和对象
- 数据类型
- 运算符
- 内置函数
- 应用实例

===内置函数===

Python提供67个内置函数

➤ 转换函数

- ◆ 函数名: `abs(x)`, `bin(x)`, `bool(x)`, `int(x)`,
`complex([real[,imag]])`,
`ord(x)`, `bytes([source[,encoding[,errors]])`

- ◆ 示例

```
>>> x=-1
>>> abs(x)           #1
>>> bool(x)          #True
>>> complex(x)        #(-1+0j)
```

➤ `divmod(a,b)`

- ◆ 作用: 返回a除以b的商和余数

```
>>> x,y=divmod(23,5)
```

===内置函数===

Python提供67个内置函数

➤ eval(expression)

◆ **expression**是字符串，返回 字符串表达式的值

◆ 示例

```
>>> a=1;b=23  
>>> eval("a+b")
```

➤ pow(x,y)

◆ 作用：返回**x**的**y**次方。

```
>>> pow(3,3) #2
```

➤ sorted()

◆ 作用：返回有序列表

```
>>> a=[5,1,6]  
>>> sorted(a)  
[1, 5, 6]
```

第2章 数据表示和基本运算

主要内容

- 常量、变量和对象
- 数据类型
- 运算符
- 内置函数
- 应用实例

【2-1】判断一个4位整数是否为回文数

- 问题分析：回文数例子，**1221**是回文数，因为其千位与个位互换，百位和十位互换后，仍是它本身
- 计算模型：设输入数为**n**，则有
 - (1) 取出每一位的值
 - n**的千位： $a=n//1000$
 - n**的百位： $b=n//100\%10$
 - n**的十位： $c=n//10\%10$
 - n**的个位： $d=n\%10$
 - (2) 构成逆序数
$$m=d*1000+b*100+c*10+a$$
 - (3) $n==m?$

【2-1】判断一个4位整数是否为回文数

- 问题分析：回文数例子，**1221**是回文数，因为其千位与个位互换，百位和十位互换后，仍是它本身
- 程序实现

```
n=input("请输入一个四位数：")
n=int(n)
a=n//1000
b=n//100%10
c=n//10%10
d=n%10
m=d*1000+c*100+b*10+a
result=(n==m)
print("回文数的判断结果是：", result)
```

===应用实例===

【2-2】判断是否为闰年

➤ 问题分析:

(1)能被4整除不能被100整除

(2)能被400整除

➤ 计算模型: 设输入数为y, 则有

(1) $y \% 4 == 0$ and $y \% 100 != 0$

(2) $y \% 400 == 0$

```
stry=input("请输入年份:")  
y=int(stry)  
result=(y%4==0 and y%100!=0) or  
(y%400==0)  
print("闰年的判断结果为: ",result)
```

【思考题】将输入的字符转换为小写，并输出其ASCII值

➤ 提示：

- (1) 条件表达式为 **z if x else y**
- (2) 获得字母的**ASCII**值函数**ord()**
- (3) 字符转换函数**chr()**
- (4) 大写字母为'A'~'Z'

```
c=input("请输入一个字符： ")
y=(c if(c>='a' and c<='z') else chr(ord(c)+32))
print("字符： ",c,",ASCII:",ord(c),"转换为： ",
y,",ASCII: ",ord(y))
```

本章小结

- 常量、变量和对象
- 数据类型
- 运算符
- 内置函数
- 应用实例