第2章 基本数据类型与表达式

2.1 数字类型

2.2 字符串类型

2.3 变量与赋值

2.4 运算符与表达式

- 掌握数字类型的特点及其操作方法
- 掌握字符串类型的特点及其操作方法
- 熟悉并能运用系统函数
- 逐步熟悉Python的基本运算、表达式和优先级

数据类型

- 类型：是编程语言对数据的一种划分。不同的编程语言，对数据类型有不同的规定。
- Python语言的数据类型：
  - 数字类型
  - 字符串类型

  基本数据类型
  - 列表
  - 元组
  - 字典
  - 集合

  组合数据类型

2.1 数字类型

- 数字类型对Python语言中数字的表示和使用进行了定义和规范。
- Python语言包括四种数字类型
  - 整数类型
  - 浮点数类型
  - 复数类型
  - 布尔值

两个内置函数

- `type(obj)`：返回obj的数据类型
- `isinstance(obj,class)`：判断obj是否属于某个类型

1. 整数类型

- 不带小数点的数字。Python的整数没有长度限制
- pow(x,y) 函数：计算x^y
- 打开IDLE，进行以下计算：
  - pow(2, 10)
  - pow(2, 1000)
  - pow(2, pow(2,15))

1. 整数类型

- Python整数支持4种计数制
  - 1010　99　-217　(10进制)
  - 0x9a　-0X89　(0x, 0X开头表示16进制数)
  - 0b010　-0B101　(0b, 0B开头表示2进制数)
  - 0o123　-0O456　(0o, 0O开头表示8进制数)

2. 浮点数类型

- 带有小数点或小数部分的数字。
- Python语言中浮点数的数值范围存在限制，小数精度也存在限制。若超出范围会导致溢出错误。

```
>>> pow(100.0, 100)
1e+200
>>> pow(100.0, 1000)
```

```
>>> import sys
>>> sys.float_info
sys.float_info(max=1.7976931348623157e+308, max_exp=1024,
max_10_exp=308, min=2.2250738585072014e-308, min_exp=-102
1, min_10_exp=-307, dig=15, mant_dig=53, epsilon=2.220446
049250313e-16, radix=2, rounds=1)
```

2. 浮点数类型

- 带有小数点或小数部分的数字。
  - 0.0　　-77.　　-2.17　（小数形式）
  - 96e4　　4.3e-3　　9.6E5（指数形式，即科学计数法）
  - 科学计数法：使用字母'e'或'E'作为幂的符号。

$$4.3e-3 = 4.3*10^{-3}$$

- 注意：计算机不一定能精确表示程序中的实数

```
>>> 2/3
0.6666666666666666
>>> 1-1/3
0.6666666666666667
```

3. 复数类型

- 与数学中的复数概念一致。

```
z = a + bj
```

  - a是实部，b是虚部，a和b都是浮点类型，虚部用j或J标识。
  - 对于复数z，可以用z.real获得实部，用z.imag获得虚部。
  - 例如：

```
z = 1.23e-4+5.6e+89j
```
(实部和虚部是什么？)

4. 布尔型

- 布尔型数据只有True和False，分别代表逻辑“真”和“假”，是两个逻辑值。
- 逻辑值True和False的值实为1和0，可以与数字类型的值进行算术运算。

```
>>> 1 == 1.0
True
>>> 123 == '123'
False
>>> 2 < 5 < 7
True
>>> (2 > 5) or (5 > 7)
False
```

```
>>> x = False
>>> x + (5 < 7)
1
```

数字类型的操作

- Python内置的算术运算符

操作符	描述
x+y	x与y之和
x-y	x与y之差
x*y	x与y之乘积
x/y	x与y之商（浮点除法）
x//y	x与y之整数商（整除）
x%y	x与y之商的余数（取模）
x**y	x的y次幂
-x	x的负值
+x	x本身

数字类型的操作

- 不同数字类型之间可以进行混合运算，运算时会发生隐式类型转换，转换规则：（低类型向高类型转换）

低 bool → int → float → complex 高

```
>>> 5 + 3
8
>>> 5 + 3.0
8.0
>>> True + 1.5
2.5
>>> True + 1.5j
(1+1.5j)
```

```
>>> 5 / 3
1.6666666666666667
>>> 5 // 3
1
>>> 5 % 3
2
>>> 5 ** 3
125
```

数字类型的操作

❖ Python内置的数值运算函数

Python解释器提供了一些内置函数，与数值运算相关的如下：

函数	描述
abs(x)	x的绝对值
divmod(x, y)	返回x与的商和余数，二元组 (x//y, x%y)
pow(x, y)	乘方，即x**y
round(x[, ndigits])	对x四舍五入，保留ndigits位小数。round(x)返回四舍五入取整值
max(x_1,x_2,…,x_n)	x_1,x_2,…,x_n的最大值
min(x_1,x_2,…,x_n)	x_1,x_2,…,x_n的最小值

数字类型的操作

- Math模块中的函数

函数	描述
fabs(x)	返回x的绝对值（float类型）
ceil(x)	向上取整，返回大于等于x的最小整数
floor(x)	向下取整，返回小于等于x的最大整数
trunc(x)	返回x的整数部分
factorial(x)	返回整数x的阶乘
sqrt(x)	返回x的平方根
exp(x)	返回e的x次幂，e是自然对数
log(x)	返回x的对数值，即ln x

共提

数字类型的操作

- Math模块中的函数
  - math库是Python提供的内置数学类函数库，一共提供了4个数学常量和44个函数。

常量	数学表示	描述
math.pi	π	圆周率，值为3.141592653589793
math.e	e	自然对数，值为2.718281828459045
math.inf	∞	正无穷大，负无穷大为-math.inf
math.nan		非浮点数标记，NaN（Not a Number）

数字类型的操作

- Math模块中的函数

使用math库前，要用import先导入该库。

- 第一种： import math

对math库中函数采用 math.<函数名>()的形式

- 第二种：from math import *

对math库中函数可以直接采用<函数名>()的形式

```
>>> import math
>>> math.pi * math.sqrt(2)
4.442882938158366
```

```
>>> from math import *
>>> pi * sqrt(2)
4.442882938158366
```

数字类型的转换

- 数值运算操作符可以**隐式**地转换输出结果的数字类型。

  如： 1/2 的结果是 0.5

- Python还提供了**内置的类型转换函数**，可以进行**显式**转换。

类型转换函数	描述
int(x)	
float(x)	
str(x)	
complex(x,y)	
bool(x)	为布尔型
chr(x)	换为对应的字符
ord(x)	将一个字符转换成对应ASCII码或Unicode值

```
>>> x = input('请输入：')
```

```
>>> bool(0)
False
```

```
>>> str(1)
'1'
>>> str(1.2e-3)
'0.0012'
```

```
>>> chr(65)
'A'
>>> ord('2')
50
>>> ord('中')
20013
```

数字类型的判断

- 内置函数type(obj)：可以返回obj的数据类型。

```
>>> type(1.0)
<class 'float'>
>>> type('1')
<class 'str'>
```

- 内置函数isinstance(obj, class)：用来测试obj是否为指定类型class的实例。

```
>>> isinstance(1, int)
True
>>> isinstance('1.0', float)
False
```

试写出以下数学算式的Python表达式

$$\sin 45° + 10^{-5}|a - b|$$

$$\frac{\sqrt[3]{c}}{a + b}$$

$$\frac{e^2 + \ln 10}{\sqrt{xy}}$$

包/库的调用

lib 里有 fun1, fun2

```
import lib
lib.fun1( )
```

```
from lib import *
fun1( )
fun2( )
```

```
import lib as x
x.fun1( )
```

```
from lib import fun1
fun1( )
fun2( )
```

2.2 字符串类型

- 字符串是用一对引号括起来的一个或多个字符（字符系列）。
- 字符串的界定符
  - 单引号　'hello world'
  - 双引号　"a"
  - 三引号字符串的界定符　（常用于多行字符串）

```
>>> print('''line1
line2
line3''')
line1
line2
line3
```

‘A’ 字符的编码：$(100\ 0001)_2$ 值为65

$b_6 b_5 b_4$ / $b_3 b_2 b_1 b_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	
0001	SOH	DC1	!	1	A	Q	a	
0010	STX	DC2	"	2	B	R	b	
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
01								
10								
10								
10								
11								
11								
11								
1111	SI	US	/	?	O	_	o	DEL

‘S’ 字符的编码：$(010\ 0100)_2$ 值为36

字符0～9、A～Z、a～z的编码是顺序排列的：

‘0’ 的编码十进制值为48，‘1’ 的编码为49；依次类推

‘A’ 的编码为十进制65，‘B’ 的编码为66；依次类推

‘a’ 的编码为十进制97，‘b’ 的编码为98；依次类推

2.2 字符串类型

- Python语言转义符：\ （反斜杠）

转义符是一些特殊字符。

```
>>> print('a\tb\nc\\')
a	b
c\
>>> print('\"Great!\"')
"Great!"
>>> print('\x41bc')
Abc
```

转义字符	
\t	
\v	
\n	换行
\ooo	八进制数ooo代表的字符，如：\012代表换行
\xhh	十六进制数hh代表的字符，如：\x0a也代表换行
\\	反斜杠符号
\’	单引号
\”	双引号

2.2 字符串类型

- **Python语言转义符： \（反斜杠）**

  转义符是一些特殊字符。

- **原始字符串：用r或R来定义**

  原始字符串用于显示字符串原来的样子，不让转义符生效。

```
>>> print(r'a\tb\nc\\')
a\tb\nc\\
>>> print(R'\x41bc')
\x41bc
```

2.2 字符串类型

- 字符串是一个字符序列：字符串最左端位置标记为0，依次增加，字符串中的编号称“索引”。


```
>>> s = 'Hello World'
>>> print(s[0], s[10])
H d
>>> print(s[-1], s[-11])
d H
>>> s[11]   # 下标越界了
Traceback (most recent call last):
  File "<pyshell#165>", line 1, in <module>
    s[11]   # 下标越界了
IndexError: string index out of range
```

2.2 字符串类型

❖ 字符串切片

可以通过两个索

格式： <strin

- start：切片起
- end：切片截止
- step：切片步

```
>>> a = 'Hello World'
>>> a[1:4]
'ell'
>>> a[:4]
'Hell'
>>> a[1:]
'ello World'
>>> a[::]
'Hello World'
>>> a[::2]
'HloWrd'
>>> a[::-1] # 步长为-1，得到倒序字符串
'dlroW olleH'
>>> a[:100] # 截止位置越界
'Hello World'
>>> a[100:] # 起始位置越界，返回空字符串
''
```

2.2 字符串类型

【例】输入一个月份数字，输出对应月份名称缩写。

如：>>> 1

Jan

【问题分析】

1）输入：输入一个表示月份的数字（1-12）

2）处理：利用字符串基本操作得到缩写字符串

3）输出：显示对应月份名称的缩写

2.2 字符串类型

将所有月份名称缩写存储在字符串中：

```
months = "JanFebMarAprMayJunJulAugSepOctNovDec"
```

在字符串中截取适当的子串来查找特定月份的缩写。


```
# month.py

months = "JanFebMarAprMayJunJulAugSepOctNovDec"
n = input("请输入月份数（1-12）：")
pos = (int(n)-1) * 3
mon = months[pos:pos+3]
print("月份缩写是：" + mon + ".")
```

2.2 字符串类型

❖ 字符串连接

- 加法操作 +：将两个字符串首尾相连成一个新字符串。
- 乘法操作 *：让字符串自身多次重复拼接而成一个新字符串。

❖ 求字符串长度

使用len( )函数，可以得到字符串包含多少个字符。

2.2 字符串类型

❖ 内置的字符串处理函数

◆ Python提供了丰富的字符串处理函数

```
>>> dir(s)
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__'
, '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__ge
tnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter_
_', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__'
, '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__seta
ttr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold
', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format',
'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigi
t', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istit
le', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition'
, 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', '
split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate
', 'upper', 'zfill']
```

操作	含义
+	连接
*	重复
<string>[ ]	索引
<string>[:]	剪切
len(<string>)	长度
<string>.upper()	字符串中字母大写
<string>.lower()	字符串中字母小写
<string>.strip()	去两边空格及去指定字符
<string>.split()	按指定字符分割字符串为数组
<string>.join()	连接两个字符串序列
<string>.find()	搜索指定字符串
<string>.replace()	字符串替换
for <var> in <string>	字符串迭代

2.2 字符串类型

❖ 字符串类型的格式化

◆ 方式一：使用 % 来格式化字符串

格式字符	含义
%d	输出十进制整数
%s	输出字符串
%c	输出字符格式
%m.nf	输出浮点数，长度为m（默认为0），保留n位小数（默认为6）
%o	输出八进制格式的整数
%x 或 %X	输出十六进制格式的整数
%e 或 %E	以科学计数法格式输出

2.2 字符串类型

❖ 字符串类型的格式化

◆ 方式二：使用字符串的format()方法，通过{ }来代替%

例如：

```
'Hi {0}, your score is {1}.'.format('Bob', 85.7)
'Hi {0}, your score is {1:.0f}.'.format('Bob',85.7)

'{name} is {age} years old.'.format(age=20, name='Bob')
```

练习：小明的成绩从去年的72分提升到了今年的85分，请计算小明成绩提升的百分点，并用字符串格式化显示出‘**xx.x%**’，只保留小数点后1位。

```
s1 = 72
s2 = 85
r = ???
print('???' % r)
```

3. 变量与赋值

❖ 变量的定义

Python通过对变量第一次进行赋值，来实现变量定义。变量一旦被定义，系统就会给变量分配内存空间。

```
>>> x
Traceback (
  File "<py
    x
NameError:
```

```
>>> x = 1
>>> x
1
>>> type(x)
```

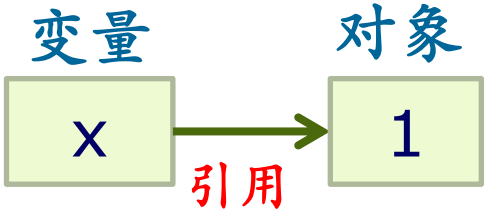
❖ 变量

使用

```
>>> del x
>>> x
Traceback (most recent call last):
  File "<pyshell#65>", line 1, in <module>
    x
NameError: name 'x' is not defined
```

一旦被删除，系

象就不存在了。



3. 变量与赋值

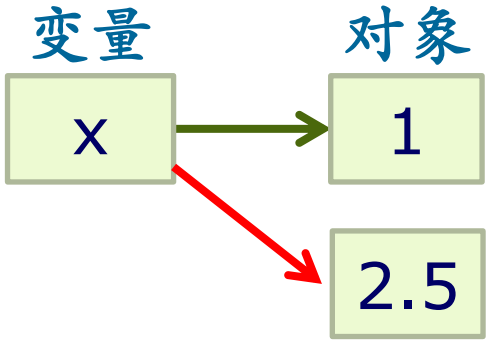
❖ 变量的引用　　x = 1

变量里存放的并非“值”本身，而是“值”对象的内存地址。通过地址去访问对象的方式，称为引用。

❖ 变量修改赋值

实质是改变了变量的引用（即变量

```
x = 1
x = 2.5
```

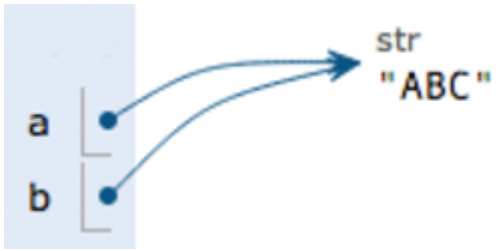



```
>>> x = 1
>>> x
1
>>> id(x)
271702192
>>> x = 2.5
>>> x
2.5
>>> id(x)
37388944
```

3. 变量与赋值

❖ 变量的引用

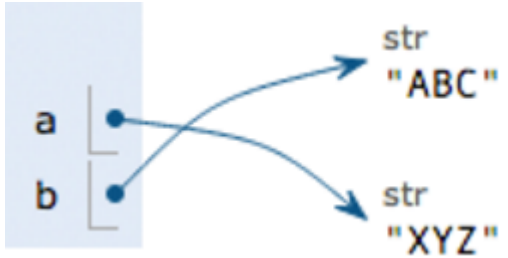
```
>>> a = 'ABC'
>>> b = a
>>> a = 'XYZ'
>>> print(b)
```



思考：

打印出的变量b的内容是

‘ABC’ 还是 ‘XYZ'?



4 运算符和表达式

```
>>> x = 2.5
>>> y = 2.5
>>> x == y
True
>>> x is y
False
```

```
>>> x = 2.5
>>> y = x
>>> x is y
True
```

❖ 关系运算符

也称比较运算符，可对两个数值型或字符串型数据进行大小比较，返回一个“真”或“假”的布尔值。

运算符	描述	运算符	描述
>	大于	！=	不等于
>=	大于等于	is	(引用) 同一个对象
<	小于	is not	(引用) 不同的对象
<=	小于等于		

4 运算符和

```
>>> "abc" > "Abc"
True
>>> "abcd" > "abc"
True
>>> " abc" > "a"
False
```

```
>>> a = 0.1 * 3
>>> b = 0.3
>>> a == b
False
>>> abs(a-b) < 1e-6
True
```

❖ 关系运算符

- 两个字符串比较

  逐个字符依次比较，规则是：

  空字符<空格<'0'~'9'<'A'~'Z'<'a'~'z'<汉字

- 两个浮点数比较

  如果两个浮点数之差小于一个极小值，即认为相等。

```
>>> a = 1 + 2j
>>> b = 2 + 3j
>>> a == b
False
>>> a < b
Traceback (most recent call last):
  File "<pyshell#98>", line 1, in <module>
    a < b
TypeError: '<' not supported between instances of 'complex' and 'complex'
```

- 两个复数比较

  复数不能比大小，只能比较是否相等。

- Python允许链式比较

  x<y<z　等价于　x<y and y<z

```
>>> 1 < 3 < 5
True
>>> 1 < 3 and 3 < 5
True
```

4 运算符和表达式

❖ 逻辑运算符

运算符	描述
and	逻辑与，两者都为真，结果才为真
or	逻辑或，两者都为假，结果才为假
not	逻辑非，非真即假，非假即真

- **优先级从低到高：** `or < and < not`

4 运算符与表达式

- 表达式：由运算符和操作数组成，操作数可以是常量、变量，可以是函数的返回值。
  - 很多运算对操作数的类型有要求，当操作类型不一致时，可能发生隐式类型转换。
  - 差别较大的数据类型，则需要进行显式类型转换。
  - 表达式结果的类型由操作数和运算符共同决定。

4 运算符与表达式

- 常见运算符的优先级

低

逻辑或or
逻辑与and
逻辑非not
关系运算符 > >= < <= == != is is not
加减 + -
乘除 * / // %
单目+ 单目-
幂次 **
索引[ ]

高

本次课后任务

- 完成课题例子
- 复习本课内容