Python语言基础与应用

数据类型 / 基本类型：数值

陈斌 北京大学 gischen@pku.edu.cn

基本类型：数值

- 整数类型
- 浮点数类型
- 复数类型
- 更多的数学函数

整数类型：int

最大特点是不限制大小

- 无论多复杂的算式都可以直接得到结果

常见的运算

运算符	功能	备注
m + n	加法	
m - n	减法	
m * n	乘法	
m // n	整数除法	结果是商的整数部分
m / n	除法	“真”除法，得到小数
m % n	求余数	
divmod(m, n)	求整数除法和余数	会得到两个整数，一个是 m // n，另一个是 m % n
m ** n	求乘方	整数 m 的 n 次方
abs(m)	求绝对值	

整数类型：int

- 大小比较
- 连续比较判断

```
>>> 7 > 3 >= 3
True
>>> 12 < 23 < 22
False
```

m == n	相等比较	m 是否等于 n
m > n	大于比较	m 是否大于 n
m >= n	大于或等于比较	m 是否大于或者等于 n
m < n	小于比较	m 是否小于 n
m <= n	小于或等于比较	m 是否小于或者等于 n

整数类型：int

数的进制

我们用多少个不同符号来表示数？

通常用的十进制是0-9，十个不同符号

逢十进一

十进制	0	1	2	3	4	5	6	7	8
二进制	0	1	10	11	100	101	110	111	1000
八进制	0	1	2	3	4	5	6	7	10
十六进制	0	1	2	3	4	5	6	7	8
十进制	9	10	11	12	13	14	15	16	17
二进制	1001	1010	1011	1100	1101	1110	1111	10000	10001
八进制	11	12	13	14	15	16	17	20	21
十六进制	9	a	b	c	d	e	f	10	11

整数类型：int

整数的各种进制表示

- Python语言中可以直接用二进制、八进制和十六进制来表示整数，只要加一个前缀用以标识几进制即可

进制	表示	例子
十进制 decimal	无前缀数字	367
二进制 binary	0b 前缀	0b101101111
八进制 octal	0o 前缀	0o557
十六进制 hexadecimal	0x 前缀	0x16f

浮点数类型：float

- 操作与整数类似
- 浮点数受到17位有效数字的限制
- 特点
  - 科学记数法
  - 有效位数
- 特性
  - 进制转换导致精度误差

```
>>> 355/113
3.1415929203539825
>>> 3.14159265358979323
84626
3.141592653589793
>>> 1234123423412342312
341234.0
1.2341234234123424e+24
>>> 0.000001231231231231
23123
1.2312312312312312e-06
>>> 4.2 + 2.1 == 6.3
False
>>> 4.2 + 2.1
6.300000000000001
```

复数类型

复数生成

- Python内置复数数据类型

复数运算

- 支持所有常见计算

```
>>> 1+3j
(1+3j)
>>> (1+2j)*(2+3j)
(-4+7j)
>>> (1+2j)/(2+3j)
(0.6153846153846154+0.07692307692307691j)
>>> (1+2j)**2
(-3+4j)
>>> (1+2j).imag
2.0
>>> (1+2j).real
1.0
>>>
```

复数类型

复数比较

- 复数之间只能比较是否相等

复数应用

- 求平面上两个点(x1, y1)和(x2, y2)的距离

```
>>> (1 + 2j) == (3 + 4j)
False
>>> (1 + 2j) < (3 + 4j)
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    (1 + 2j) < (3 + 4j)
TypeError: '<' not supported between instances of 'complex'
and 'complex'
>>> abs((3 + 3j) - (4 + 4j))
1.4142135623730951
```

更多的数学函数：math模块

数学常数

- 圆周率π、自然对数的底e等

数学函数

- 三角函数、对数、最大公约数、最小公倍数等

```
>>> import math
>>> dir(math)
['__doc__', '__loader__', '__name__', '__package__', '__spec__',
'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil
', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp
', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum'
, 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf',
'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'mod
f', 'nan', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan',
'tanh', 'tau', 'trunc']
```

更多的数学函数：cmath模块

专门面向复数计算

- math模块中的数学函数只能用于计算整数和浮点数，对于复数就无能为力了

平面直角坐标和极坐标之间的转换

```
>>> cmath.polar(1 + 1j)
(1.4142135623730951, 0.7853981633974483)
>>> cmath.rect(1, cmath.pi / 2)
(6.123233995736766e-17+1j)
```