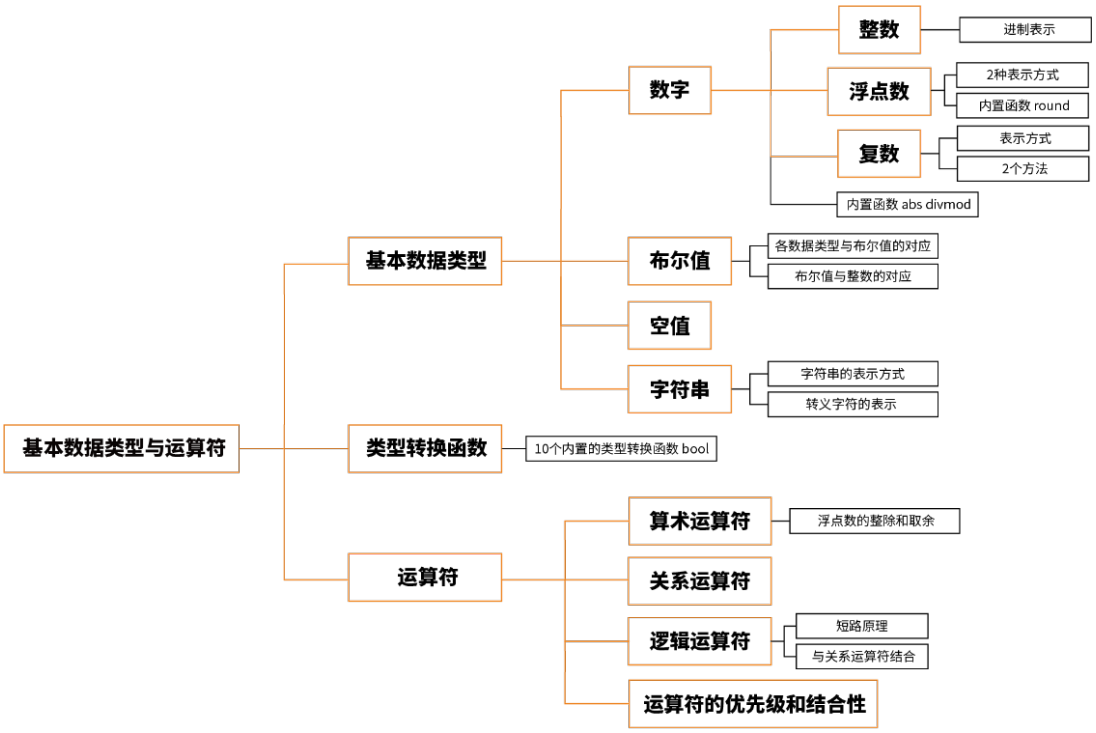
第 1 讲　基本数据类型与表达式

知识脉络



一　基本数据类型

1. 数字类型

① 整数 int

```
108             # 十进制 108（不需要加前缀，但不能以 0 开头）
0b1101100       # 二进制 108（0b 或 0B 前缀）
0o154           # 八进制 108（0o 或 0O 前缀）
0x6C            # 十六进制 108（0x 或 0X 前缀），不管用何种进制表示，它们都是同一个整数对象
```

② 浮点数 float

```
10.8            # 一般表示法
1.2e-5          # 科学计数法　xey 等价于 x × 10^y，其中 x 不能为 0，y 必须为非 0 整数
```

相关函数

```
round(a, n)     # 返回浮点数 a 保留 n 位小数的形式（四舍五入），n 默认值为 0
```

③ 复数 complex

```
2 + 3j          # 2 为实部，3 为虚部，类型为 int 或 float
```

相关方法

```
z.real()        # 返回复数 z 的实部
z.imag()        # 返回复数 z 的虚部
```

数字类型相关函数

```
abs(a)          # 返回数字 a 的绝对值（整数、浮点数）或模长（复数）
```

```
divmod(a, b)    # 返回元组(a // b, a % b)，也就是整除和取余的结果
```

2.布尔值 bool

```
True            # 真
```

```
False  # 假
```

· 所有的数据都具有布尔属性，并且只有以下取值对应的布尔值为 False

整数	浮点数	复数	字符串	空值	列表	元组	集合	字典
0	0.0	0+0j	''	None	[]	()	set()	{}

· **布尔值具有数字属性：** True 与 1 对应，False 与 0 对应

3.空值

```
None            # 对应的布尔值为 False，但并不与 0 等价
```

4.字符串 str（此处仅是字符串的一部分知识点）

① 单行字符串

· 字符串是字符组成的序列，在程序中用（'）或（"）括起来表示（两种符号没有区别，但必须匹配）

如果在程序文件中要换行，用**反斜杠**分隔开，在**输出时不视为换行**（本质还是单行字符串）

② 多行字符串

· 前后用三个引号（"""）括起来，这种表示在输出时程序文件换行的地方就会换行

```
"Programming is fun."
""  # 空字符串
```

单行字符串

```
>>> "Programming \
is fun."
Programming is fun.
```

换行字符串

```
>>> """Programming
is fun."""
Programming
is fun.
```

多行字符串

③ 转义字符

换行、Tab 等字符无法表示，引号等会引起歧义

因此引入转义字符，以“\”作为前缀

转义字符	意义
\\	反斜杠符号
\'	单引号
\"	双引号
\b	退格(Backspace)
\n	换行
\v	纵向制表符
\t	横向制表符
\最多 3 位 8 进制数字	该 8 进制数对应的编码字符
\x 两位 16 进制数字	该 16 进制数对应的编码字符

④ 字符串的前缀

写在字符串的引号前，起指示作用

前缀	意义
r	内部的 \ 不会被视为转义字符前缀
f	允许用{}嵌入变量
u	按照 unicode 编码

二　类型转换函数

1. 基本类型转换函数

① `bool(x)`　　# 将 x 转换为对应的布尔值

② `int(x, base)`　# 将 x 按照 base 进制（默认为 10）解读，转换为对应的整数值

- 若不输入任何参数：返回 0
- 若 x 是数字类型，则不允许传入 base 参数，结果是对 x 向下取整
- 若 x 是 str：允许两侧有空格，但去除两侧空格后 x 只能包含选定进制范围以内的字符（允许 0 开头）
- 若 x 是 bool：按照对应规则（True 为 1，False 为 0）处理

③ `float(x)`　　# 将 x 转换为对应的浮点数

④ `complex(x, y)`　# 创建一个新的复数 x + yj，要求 x 和 y 必须是整数或浮点数

⑤ `str(x)`　　# 将对象 x 转换为对应的字符串

- 若 x 是数字，不管在程序中是以几进制表示的，都返回以 10 进制形式表示的字符串

  若 x 是列表、字典等，返回的字符串会带上它们对应的括号

2. Unicode 编码与整数间转换

① `ord(x)`　　# 将字符 x 转换为其 Unicode 码对应的十进制整数

② `chr(x)`　　# 将整数 x 转换为对应的 Unicode 字符

3. 非 10 进制数字字符串

① `bin(x)`　　# 将整数 x 转换为对应的用 2 进制表示的字符串（含前缀 0b）

② `oct(x)`　　# 将整数 x 转换为对应的用 8 进制表示的字符串（含前缀 0o）

③ `hex(x)`　　# 将整数 x 转换为对应的用 16 进制表示的字符串（含前缀 0x）

三　基本运算符

1. 算术运算符

运算符	说明	使用	结果
+	加法	2 + 3	5
-	减法	2 - 3	-1
+/-	单目正负	-2	-2
*	乘法	2 * 3	6
/	浮点数除法	10 / 5	2.0
//	整除	10 // 5	2
%	模（求余数）	10 % 4	2
**	幂	2 ** 3	8

① 浮点数由于在计算机内存储表示的问题，运算可能存在误差

② 浮点数也可以整除和取余，**整除的结果类型是浮点数，但数学上是整数值**

```
>>> 3.8 // 0.7
5.0
```

```
>>> 3.8 % 0.7
0.30000000000000004
```

③ 浮点数与整数运算的结果一定是浮点数类型

2. 关系运算符

运算符	说明	使用	结果
==	等于	2 == 3	False
!=	不等于	2 != 3	True
>	大于	2 > 3	False
<	小于	2 < 3	True
>=	大于或等于	2 >= 3	False
<=	小于或等于	2 <= 3	True

· 除此之外还有 in 和 not in，这两个运算符不能用于数字之间，但也属于关系运算符

· 字符串只能和字符串比较：比较第一个字符的编码，编码大则字符串大，相同则比较下一个直至某一字符串的字符比较完毕，若另一字符串还有字符，则另一字符串大

3. 逻辑运算符

```
表达式 1 and 表达式 2      # 若表达式 1 的结果为假，返回表达式 1 的值，否则处理并返回表达式 2 的结果
```

```
表达式 1 or 表达式 2       # 若表达式 1 的结果为真，返回表达式 1 的值，否则处理并返回表达式 2 的结果
```

```
not 表达式                 # 若表达式的结果为真，返回 False，否则返回 True
```

· 因此逻辑表达式虽然是根据布尔逻辑作运算，但**返回结果不一定是布尔值**

· 关系运算符与逻辑运算符结合：`a 关系符 1 b and b 关系符 2 c` → `a 关系符 1 b 关系符 2 c`

```
1 < 3 < 5    # 等价于 1 < 3 and 3 < 5
```

4. 运算符的优先级与结合性

① 当表达式中有多个运算符时，需要按照优先级和结合性，按顺序处理各个运算符：

`** → + -（单目） → * / // % → + -（双目） → < <= == != >= > → not → and → or`

② 单双目+-的判别：运算符左边是数值的是双目运算符（如 1+1），否则是单目运算符（如-1）

③ ** 和 + -（单目）的结合性是从右向左，指有多个同等级运算符并列时，从右往左处理运算符

④ 圆括号（）可以改变优先级，括号内的表达式计算优先级为最高

⑤ 后面几讲将涉及到.（属性）、()（函数调用）、[]（取值、切片），它们的优先级高于上述运算符

经典例题

例 1 表达式 3**2**3 的值为_____。

解 **为从右向左结合，因此先处理右边的**：将左边的 2 与右边的 3 结合：2**3 → 2^3 → 8

结果覆盖掉这一部分表达式：3**8 → 3^8 → 6561

例 2 表达式---3 的值为_____。

解 3 个-都是单目运算符，从右向左结合，一个技巧是相邻的两个-可以直接抵消，因此结果为-3

例 3 表达式(2>=2 or 2<2) and 2 的值为_____。

解 由于括号，要先处理两个关系表达式，再处理 or，最后处理 and

关系表达式 2>=2 的结果为 True，因此根据 or 的短路原理，该 or 表达式的值直接为 True

表达式简化为 True and 2，根据 and 的短路原理，该 and 表达式的值为 and 后面的 2

例 4 表达式 32.2//6-24//6 的值是_____。

解 -的左边是数字 6，因此是双目运算符，优先级低于//

32.2//6 的结果是 5.0，24//6 的结果是 4，因此 5.0 - 4 的结果是 1.0

例 5 表达式 32.2//6-24//6 的值是_____。

解 -的左边是数字 6，因此是双目运算符，优先级低于//

32.2//6 的结果是 5.0，24//6 的结果是 4，因此 5.0 - 4 的结果是 1.0

例 6 python 语句 print(int('20', 16), int('101', 2))的输出结果是_____

解 第一个 int 将字符串'20'当成 16 进制数，则这个数用 10 进制表示是 32

第二个 int 将字符串'101'当成 2 进制数，则这个数用 10 进制表示是 5

因此输出结果为 `32 5`

例 7 print(hex(16), bin(10))的输出结果是_____（进制用小写字母表示）

解 hex(16)将十进制数 16 转化为十六进制数 10，加上前缀为 0x10

bin(10)将十进制数 10 转化为二进制数 1010，加上前缀为 0b1010

因此输出结果为 `0x10 0b1010`