C语言的浮点数

- **两种精度**
  - `float` 单精度
  - `double` 双精度
- **类型转换**
  - `int`, `float`, `double` 间转换，将改变位模式
  - `double`/`float` → `int`
    - 截掉小数部分
    - 类似向0舍入
    - 当数值超范围或NaN时无定义：通常设置为 TMin
  - `int` → `double`
    - 精确转换,只要`int`的位宽 ≤ 53 bit，即可精确转换
  - `int` → `float`
    - 将根据舍入模式进行舍入

浮点数习题

- 针对下列C表达式:
  - 证明对所有参数值都成立
  - 或什么条件下不成立

```
int x = …;
float f = …;
double d = …;
```

假定d 和 f都不是NaN

- x == (int)(float) x
- x == (int)(double) x
- f == (float)(double) f
- d == (double)(float) d
- f == -(-f);
- 2/3 == 2/3.0
- d < 0.0 ⇒ ((d*2) < 0.0)
- d > f ⇒ -f > -d
- d * d >= 0.0
- (d+f)-d == f

浮点数习题答案

- x == (int)(float) x　　No: 无法表示24 位尾数, x=0x1000001
- x == (int)(double) x　　Yes: 53位尾数
- f == (float)(double) f　　Yes: 增加精度
- d == (float) d　　No: 损失精度
- f == -(-f);　　Yes: 仅仅改变符号位
- 2/3 == 2/3.0　　No: 2/3 == 0
- d < 0.0⇒ ((d*2) < 0.0)　　Yes!
- d > f ⇒ -f < -d　　Yes
- d *d >= 0.0　　Yes!
- (d+f)-d == f　　No: 不具备结合性