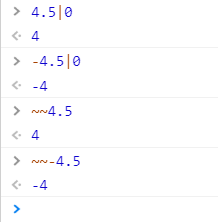
对于 ~~number 和 number|0 都能取整的问题思考(正数取上整、负数取下整)



4.5|0 过程分析

1. 将4.5转换为二进制 = 100.1

2. 将二进制数100.1 丢弃小数部分，并和0进行按位或 = 100

3. 将二进制数100转换为整数 = 4

先回顾大学学的关于补码的问题

计算机中进行减运算时，其实进行的是加上减数的相反数

那么如 4-1 应该怎么做

1. 首先将4-1 转成 4 + (-1)
2. 那么-1应该怎么表示？

Js中Number类型的数据占32位, 这里简化一下只占4位

那么4位能表示的最大数即 1111 = 15，15+1则进位，丢弃高位， 剩下的4位为0000 = 0

那么 4-1 就可以看做是 4 + (16-1)， 因为16的二进制为10000， 丢弃最高位就 = 0

计算 4+15， 0100 + 1111 = 10011, 丢弃最高位，则为0011 = 3

在计算机中为了标识正负数，就将第一位标识

而负数以补码的形式存储, 即 -1 = 1111

~~4.5 即对4.5进行两次按位非计算



~4.5 过程分析

1. 将4.5转换为二进制 = 0100.1 (最高位为符号位)
2. 丢弃小数部分，进行按位非 = 1011
3. 发现最高位为1， 将其余位取反 = 1100
4. 末位+1取补码 = 1101
5. 转换为10进制 = -5

~-5过程分析

1. 将 -5转换为二进制 = 1101 (最高位为符号位)
2. 发现最高位为1，将其余位取反 = 1010
3. 末位+1 = 1011
4. 按位非 = 0100
5. 转换为10进制 = 4

综上 number|0方式更简单，效率更快