位运算

# Get Bit

给定两个正整数 number 和 bitPosition , 返回 number 的二进制形式上的第 bitPosition 位数

### **示例 1：**

**输入：number =** 4, bitPosition = 2

**输出：**1

### **示例 2：**

**输入：number =** 3, bitPosition = 1

**输出：**1

### **解题思路：**

把 number 的二进制形式的数向右移动 bitPosition 位，即把 bitPosition 位移动到 最右边(第 0 个) 的位置上。然后在该数上与 1的二进制形式的数(0001) 进行 AND(&) 操作。这会清理掉除了目标位的所有其它位的数据。如果目标位是 1 ，那么结果就是 1 ，反之，结果是 0

### **题解：**

<https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/getBit.js>

# Set Bit

给定两个正整数 number 和 bitPosition , 要求把 number 的二进制形式上的第 bitPosition 位数设为 1 并返回结果

### **示例 1：**

**输入：number =** 2, **bitPosition =** 0

**输出：**3

### **示例 2：**

**输入：number =** 2, **bitPosition =** 1

**输出：**2

### **解题思路：**

把 1(0001) 向左移动 bitPosition 位，生成一个二进制形如 0100(bitPosition = 2) 的值。然后我们拿该值与目标数字进行 OR(|) 操作，就能把目标位设置位 1 而不影响其它位。

### **题解：**

<https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/setBit.js>

# Clear Bit

给定两个正整数 number 和 bitPosition , 要求把 number 的二进制形式上的第 bitPosition 位数清除并返回结果

### **示例 1：**

**输入：number =** 2, **bitPosition =** 1

**输出：**0

### **示例 2：**

**输入：number =** 2, **bitPosition =** 0

**输出：**2

### **解题思路：**

把 1(0001) 向左移动 bitPosition 位，生成了一个二进制形如 0100(bitPosition = 2) 的值。然后反转每一位的数字，得到一个二进制形如 1011 的值。接着与目标值进行 AND(&) 操作，就能清除掉目标位的值。**题解：**

### **题解：**

<https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/clearBit.js>

# Update Bit

给定三个正整数 number 、 bitPosition 和 bitValue , 要求把 number 的二进制形式上的第 bitPosition 位数更新为 bitValue 并返回结果

### 说明:

1. bitValue 只能为 1 or 0

### **示例 1：**

**输入：number =** 2, **bitPosition =** 1, bitValue = 1

**输出：**2

### **示例 2：**

**输入：number =** 2, **bitPosition =** 1, bitValue = 0

**输出：**0

### **解题思路：**

组合 “Clear Bit” 和 “Set Bit” **：**

### **题解：**

<https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/updateBit.js>

**isEven**

给定一个正整数 number , 检测传入的 number 是否是偶数 , 返回布尔值

### **示例 1：**

**输入：number =** 2

**输出：**true

### **示例 2：**

**输入：number =** 1

**输出：**false

### **解题思路：**

二进制中奇数的最右边的位永远是1 **：**

### **题解：**

<https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/isEven.js>

**isPositive**

给定一个整数 number , 检测传入的 number 是否是正数 , 返回布尔值

### **示例 1：**

**输入：number = -**2

**输出：**true

### **示例 2：**

**输入：number =** 1

**输出：**false

### **解题思路：**

二进制中正数最左边的位永远是 0 。(JavaScript 使用 32 位按位运算数) 然而如果传入的 number 是 0 或者 -0 ，它也应该返回 false**：**

### **题解：**

<https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/isPositive.js>

**Multiply By Two**

给定一个整数 number ,返回 number \* 2 的值

### **示例 1：**

**输入：number =** 2

**输出：**4

### **示例 2：**

**输入：number =** 1

**输出：**2

### **解题思路：**

将 number 向左移动一位。因此所有位都将乘以 2，因此数字本身也将乘以 2**：**

### **题解：**

[https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/multiplyByTwo.js](%20https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/multiplyByTwo.js)

**Divide By Two**

给定一个整数 number ,返回 number / 2 并向下取整的值

### **示例 1：**

**输入：number =** 2

**输出：**1

### **示例 2：**

**输入：number =** 5

**输出：**2

### **解题思路：**

将 number 向右移动一位。因此所有位都将除以 2 , 因此数字本身也将除以 2 , 且不会产生余数**：**

### **题解：**

[https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/divideByTwo.js](https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/isPositive.js)

**Switch Sign**

给定一个正整数 number ,返回 number \* -1 的值

### **示例 1：**

**输入：number =** 2

**输出：**-2

### **示例 2：**

**输入：number =** 5

**输出：**-5

### **解题思路：**

使用了 “二进制补码” 的方法，即 取反(~) 所有位然后加 1**：**

### **题解：**

[https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/divideByTwo.js](https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/isPositive.js)

**Multiply Two Signed Numbers**

给定两个整数 a 和 b , 使用位运算符计算两个有符号数的乘积

### **示例 1：**

**输入：number =** 2

**输出：**-2

### **示例 2：**

**输入：number =** 5

**输出：**-5

### **解题思路：**

使用了 “二进制补码” 的方法，即 取反(~) 所有位然后加 1**：**

### **题解：**

[https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/divideByTwo.js](https://github.com/trekhleb/javascript-algorithms/blob/master/src/algorithms/math/bits/isPositive.js)