

3.5.4 乘性操作符

ECMAScript 定义了 3 个乘性操作符：乘法、除法和求模。这些操作符与 Java、C 或者 Perl 中的相应操作符用途类似，只不过在操作数为非数值的情况下会执行自动的类型转换。如果参与乘性计算的某个操作数不是数值，后台会先使用 `Number()` 转型函数将其转换为数值。也就是说，空字符串将被当作 0，布尔值 `true` 将被当作 1。

1. 乘法

乘法操作符由一个星号 (*) 表示，用于计算两个数值的乘积。其语法类似于 C，如下面的例子所示：

```
var result = 34 * 56;
```

在处理特殊值的情况下，乘法操作符遵循下列特殊的规则：

- ❑ 如果操作数都是数值，执行常规的乘法计算，即两个正数或两个负数相乘的结果还是正数，而如果只有一个操作数有符号，那么结果就是负数。如果乘积超过了 ECMAScript 数值的表示范围，则返回 `Infinity` 或 `-Infinity`；
- ❑ 如果有一个操作数是 `NaN`，则结果是 `NaN`；
- ❑ 如果是 `Infinity` 与 0 相乘，则结果是 `NaN`；
- ❑ 如果是 `Infinity` 与非 0 数值相乘，则结果是 `Infinity` 或 `-Infinity`，取决于有符号操作数的符号；
- ❑ 如果是 `Infinity` 与 `Infinity` 相乘，则结果是 `Infinity`；
- ❑ 如果有一个操作数不是数值，则在后台调用 `Number()` 将其转换为数值，然后再应用上面的规则。

2. 除法

除法操作符由一个斜线符号 (/) 表示，执行第二个操作数除第一个操作数的计算，如下面的例子所示：

```
var result = 66 / 11;
```

与乘法操作符类似，除法操作符对特殊的值也有特殊的处理规则。这些规则如下：

- ❑ 如果操作数都是数值，执行常规的除法计算，即两个正数或两个负数相除的结果还是正数，而如果只有一个操作数有符号，那么结果就是负数。如果商超过了 ECMAScript 数值的表示范围，则返回 `Infinity` 或 `-Infinity`；
- ❑ 如果有一个操作数是 `NaN`，则结果是 `NaN`；
- ❑ 如果是 `Infinity` 被 `Infinity` 除，则结果是 `NaN`；
- ❑ 如果是零被零除，则结果是 `NaN`；
- ❑ 如果是非零的有限数被零除，则结果是 `Infinity` 或 `-Infinity`，取决于有符号操作数的符号；
- ❑ 如果是 `Infinity` 被任何非零数值除，则结果是 `Infinity` 或 `-Infinity`，取决于有符号操作数的符号；

图灵社区会员 StinkBC(StinkBC@gmail.com) 专享 尊重版权

3. 求模

求模（余数）操作符由一个百分号（%）表示，用法如下：

```
var result = 26 % 5;    // 等于 1
```

与另外两个乘性操作符类似，求模操作符会遵循下列特殊规则来处理特殊的值：

- ❑ 如果操作数都是数值，执行常规的除法计算，返回除得的余数；
- ❑ 如果被除数是无穷大值而除数是有限大的数值，则结果是 NaN；
- ❑ 如果被除数是有限大的数值而除数是零，则结果是 NaN；
- ❑ 如果是 Infinity 被 Infinity 除，则结果是 NaN；
- ❑ 如果被除数是有限大的数值而除数是无穷大的数值，则结果是被除数；
- ❑ 如果被除数是零，则结果是零；
- ❑ 如果有一个操作数不是数值，则在后台调用 Number() 将其转换为数值，然后再应用上面的规则。