

$$0.1 + 0.2 \neq 0.3$$

IEEE 754

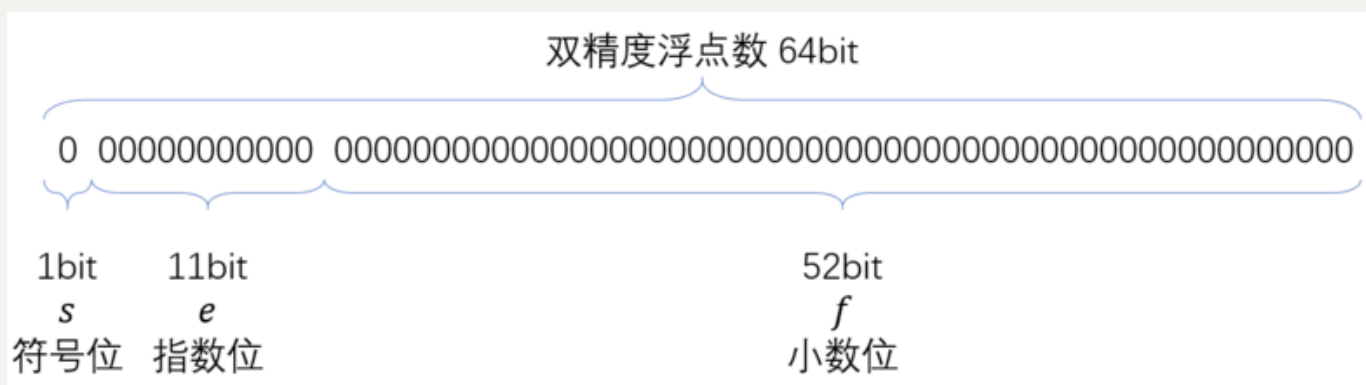
## 0.1与0.2的表示

$$0.1 + 0.2 \neq 0.3$$

计算机（Javascript）中存储浮点数的方式是采用IEEE754标准，这直接导致了浮点数精度的损失。

# IEEE 754

在IEEE754中，浮点数有单精度（32位）和双精度（64位）两种存储方法（另有扩展单（双）精度，与本文无关）。其中64位双精度浮点数存储方式如下所示：



而对于32位单精度浮点数，其由1位符号位、8位指数位和23位有效位组成。

## 0.1与0.2的表示

当我们使用IEEE754标准用64位表示0.1与0.2时，步骤如下（以0.1为例）：

1. 将十进制的0.1转换为二进制的0.1;
  - a. 将十进制小数转换为二进制的具体方式是：对小数乘2，分离其小数点前的数，并对取完数的部分重复乘2、模1取余数的操作，直至乘2得到的数为一，或根据需求截断；
  - b. 显然，0.1是无法通过乘2、模1取余的方式得到结果1的，因此它必然会截

断损失精度；

2. 用科学计数法表示二进制0.1，并只保留有效位52位；

通过上述方式，可以得到0.1的IEEE754双精度表示：

```
0 1.1001100110011001100110011001100110011001100110011001100110011010 01111111011
```

同理，0.2也无法通过乘2、模1取余的反复操作得到二进制的精确表示，0.2最终会被表示为：

```
0 1.1001100110011001100110011001100110011001100110011001100110011010 01111111100
```

将上述两数相加，显然因为两者都损失了一些精度，结果会不等于0.3。而最终结果通过浮点数转化后，为0.30000000000000004。