

0.1+0.2≠0.3 的原因在于这是数学运算中的浮点运算。

与许多其他编程语言不同，JavaScript 并未定义不同类型的数字数据类型，而是始终遵循国际 IEEE 754 标准，将数字存储为双精度浮点数。

这种格式以 64 位存储数字，其中数字（分数）存储在位 0 到 51 中，指数存储在位 52 到 62 中，符号存储在位 63 中。

我们按 IEEE754 标准用 64 位表示 0.1。第一步是将十进制的 0.1 转换为二进制的 0.1。首先将 0.1 乘以 2，然后将小数点前的数字分离出来，得到其相应的二进制数。

```
0.1 x 2 = 0.2 0
0.2 x 2 = 0.4 0
0.4 x 2 = 0.8 0
0.8 x 2 = 1.6 1
0.6 x 2 = 1.2 1
0.2 x 2 = 0.4 0
.....
```

重复此操作至 64 位。然后把它们按升序排列，获取尾数，再根据双精度标准，我们将把其四舍五入到 52 位。

```
0.000110011001100110011001100110011001100110011001100110011001100110...x20
```

用科学计数法表示二进制 0.1 并只保留前 52 位：

```
1.1001100110011001100110011001100110011001100110011001100110011001...x2-4
```

用下面的方式处理指数：

```
211-1 - 1 = 1023
1023 + (-4) = 1019
1019 base 10 ---> 1111111011 in binary
```

这里，11 代表我们要使用的 64 位表示的指数位数，-4 代表科学计数中的指数。

所以最终数字 0.1 的表示形式是：

0 1.1001100110011001100110011001100110011001100110011001100110011010 01111111011

0.2 表示为：

0 1.1001100110011001100110011001100110011001100110011001100110011010 01111111100

转换为浮点数，它变成：

0 0111111101 0011001100110011001100110011001100110011001100110100

所以 $0.1 + 0.2 = 0.30000000000000004$

参考链接：[Why is 0.1 + 0.2 Not Equal to 0.3 in Most Programming Languages? | by Parul Malhotra | Better Programming](#)