



链滴

`0.1 + 0.2 === 0.3` 的计算结果是什么?

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1547786948595>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

2019-01-17

回答

他的计算结果为 `false`，因为 JavaScript 中的 `Math` 遵循 IEEE 754 标准使用了 64 位的浮点数。简言：计算机使用二进制来存储小数，而大部分小数转换成二进制后都是无限循环的值，因此需要进行取，这样一来在进行十进制计算时就会导致精度丢失。

```
0.1 + 0.2 // 0.30000000000000004
```

要解决精度问题，可以通过以下函数对误差进行忽略：

```
const approxEqual = (n1, n2, epsilon = 0.0001) => Math.abs(n1 - n2) < epsilon  
approxEqual(0.1 + 0.2, 0.3) // true
```

加分回答

- 对于范围内的浮点数计算可以先转换为整数后再进行计算，如：

```
const add = (num1, num2) => {  
  const num1Digits = (num1.toString().split('.')[1] || '').length;  
  const num2Digits = (num2.toString().split('.')[1] || '').length;  
  const baseNum = Math.pow(10, Math.max(num1Digits, num2Digits));  
  return (num1 * baseNum + num2 * baseNum) / baseNum;  
}
```

- 可以使用 [number-precision](#) 或 [mathjs](#) 库来完成计算

返回总目录

每天 30 秒