

Falsehoods Programmers Believe About Money



Sarah Dayan
NodeConf Remote 2021



Dinero.js

stability

alpha

build

passing

license

MIT

Dinero.js lets you create, calculate, and format money safely in JavaScript and TypeScript.

v2.dinerojs.com/docs

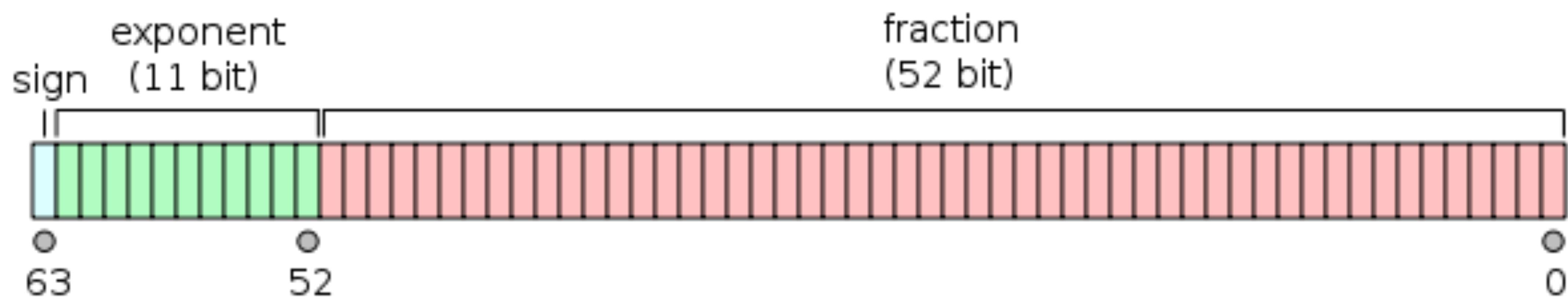
Money is complex, and the primitives of the language aren't enough to properly represent it. Dinero.js is a JavaScript library that lets you express monetary values, but also perform mutations, conversions, comparisons, formatting, and overall make money manipulation easier and safer in your application.



You can safely store money
as a fractional value

```
const cart = [
  {
    name: 'Mass Effect: Legendary Edition',
    platform: 'Xbox One',
    price: 69.99,
  },
  {
    name: 'The Legend of Zelda: Breath of the Wild',
    platform: 'Nintendo Switch',
    price: 51.91,
  }
];
```

```
cart.reduce( (acc, { price }) => acc + price, 0);  
// 121.89999999999999 😕
```



JS

- ▶ **number (doubles)**
- ▶ **bigint (arbitrary precision integers)**

```
dinero( { amount: 6999, currency: USD } );
```

\$69.99
¢6999

```
type Money<TAmount> = {  
    amount: TAmount;  
    currency: Currency<TAmount>;  
    // ...  
}
```



All currencies are decimal and
split into 100 sub-units



$x = 100x / 100$

```
function toMajorUnit(minorUnitAmount) {  
    return minorUnitAmount / 100;  
}
```



```
const USD = {  
  code: 'USD',  
  base: 10,  
  exponent: 2,  
};
```

```
// Represents USD 10.00  
dinero({ amount: 1000, currency: USD });
```

```
// Iraqi dinar
const IQD = {
  code: 'IQD',
  base: 10,
  exponent: 3,
};
```

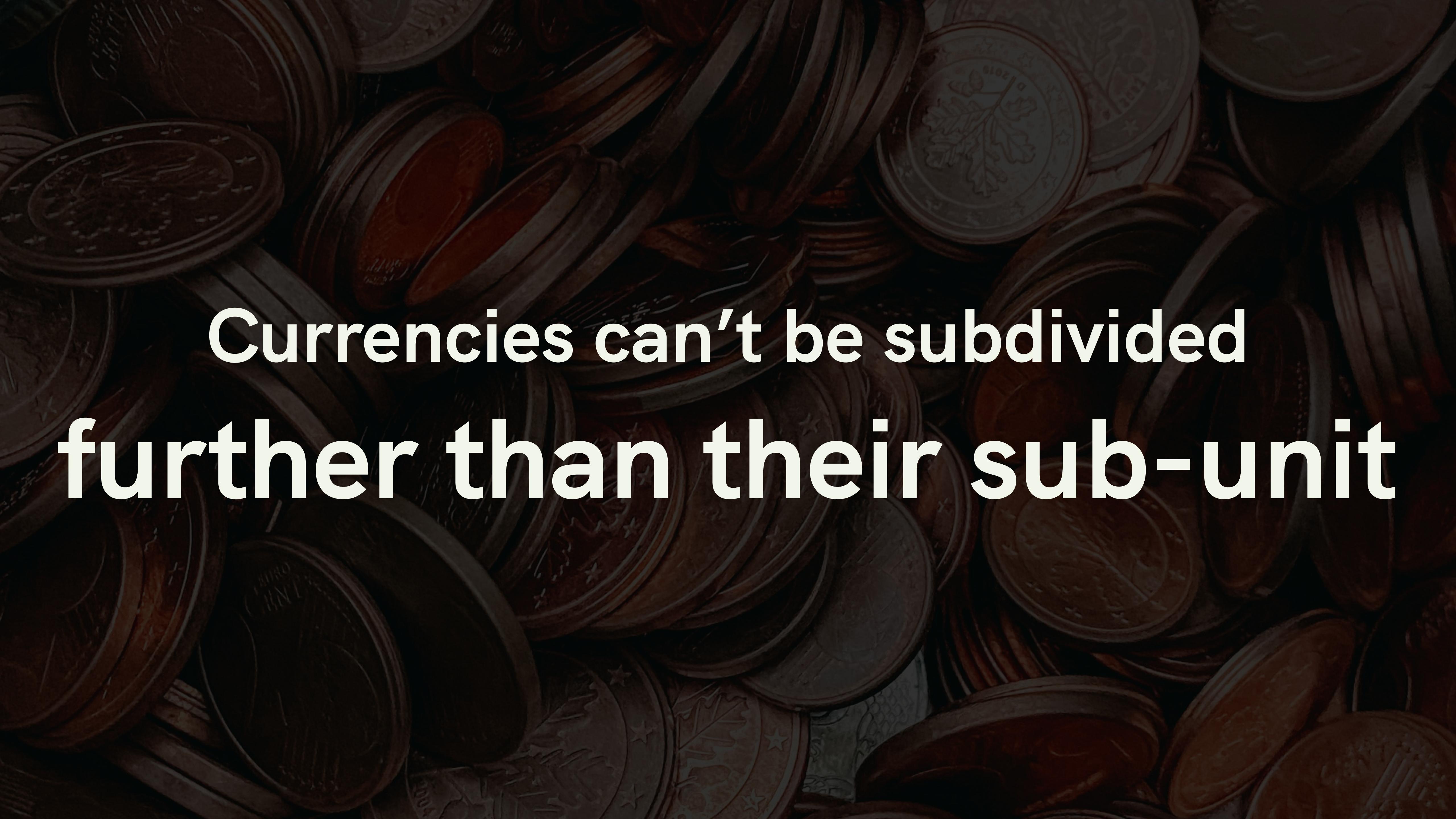
```
// Represents IQD 1.000
dinero({ amount: 1000, currency: IQD });
```

```
// Japanese yen  
const JPY = {  
  code: 'JPY',  
  base: 10,  
  exponent: 0,  
};
```

```
// Represents JPY 1000  
dinero({ amount: 1000, currency: JPY });
```

```
// Mauritanian ouguiya
const MRU = {
  code: 'MRU',
  base: 5,
  exponent: 1,
};

// Represents 2 ouguiyas and 3 khoums
dinero({ amount: 13, currency: MRU });
```



Currencies can't be subdivided
further than their sub-unit



D 1.130

95 1.330

LPG 0.6 14

0-24

SHOP

CAFÉ

Discovery
KIDS

Slovnaft

€19.95 + 5.5% VAT

€21.04725

```
const EUR = {  
  code: 'USD',  
  base: 10,  
  exponent: 3,  
};  
  
// Represents EUR 1.131  
const diesel = dinero({ amount: 1131, currency: EUR });
```

```
// Represents EUR 1.131
dinero({ amount: 1131, currency: EUR, scale: 3 });
```

```
const diesel = dinero({ amount: 1131, currency: EUR, scale: 3 });
const fullTank = multiply(diesel, 50);
const coffee = dinero({ amount: 99, currency: EUR });

// Amount 57540 and scale 3 (aka EUR 57.540)
const total = add(fullTank, coffee);
```



All currencies have
a single subdivision level





```
// Pre-decimal Great Britain pound sterling
const GBP = {
  code: 'GBP',
  base: [20, 12],
  exponent: 1,
};
```

```
// 267 pence, or 1 pound, 2 shillings and 3 pence
const d = dinero({ amount: 267, currency: GBP });
toUnits(d); // [1, 2, 3]
```



Splitting money is as simple
as dividing it

```
const purchase = {  
  title: 'Microsoft Xbox Series S',  
  price: 369.99,  
};
```

```
const length = 2;  
  
Array.from({ length }).map(() => purchase.price / length);  
// [184.995, 184.995]
```

```
Array.from({ length: count }).map(  
  () => Math.round((purchase.price / count) * 100) / 100  
) ; // [185, 185]
```

```
const purchase = {  
  title: 'Microsoft Xbox Series S',  
  price: dinero({ amount: 36999, currency: USD }),  
};
```

```
const [d1, d2] = allocate(purchase.price, [50, 50]);  
  
d1; // a Dinero object with amount 18500  
d2; // a Dinero object with amount 18499
```



You'll never need to represent
very large amounts

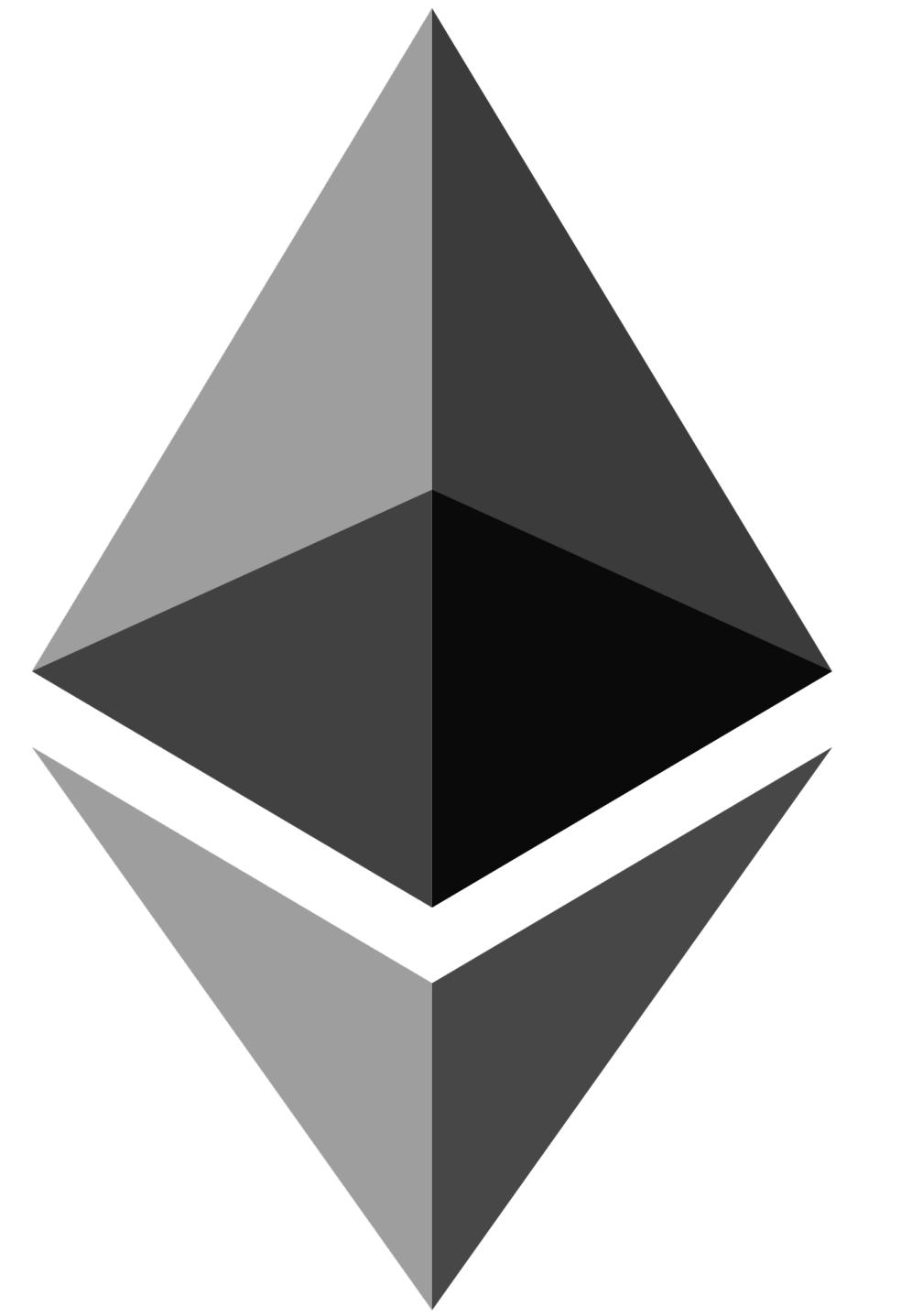
296000000000000

2960000000000000

9000000000000000

296000000000000000

900000000000000000



```
const dineroBigint = createDinero({  
  calculator: {  
    add: (a, b) => a + b,  
    // ...  
  },  
});
```

```
const dineroBigJs = createDinero({  
  calculator: {  
    add: (a, b) => a.plus(b),  
    // ...  
  },  
});
```

```
const USD = {  
    code: 'USD',  
    base: 10n,  
    exponent: 2n,  
};  
  
dineroBigint({ amount: 500n, currency: USD });
```



So, is JavaScript safe to
manipulate money?

Thank you!



@frontstuff_io