

addVeryLargeNumbers

Write a program containing a function that takes in any two numbers but in the form of strings, and returns the sum of both numbers as string.

Quite useful for adding extremely large numbers in javascript.

The need for this program arises because you cannot store more than 53 bits as a number in pure javascript.

So, to do operations on larger numbers, we convert them into strings and do operations on it.

The algorithm implemented here is something everyone is familiar with, old school addition.

$$\begin{array}{r} 28 \\ +45 \\ \hline \end{array} \Rightarrow \begin{array}{r} 1 \\ 28 \\ +45 \\ \hline 3 \end{array} \Rightarrow \begin{array}{r} 1 \\ 28 \\ +45 \\ \hline 73 \end{array}$$

Yup, we implement this into our code. Here's how.

1. Take two numbers as string, create a variable
2. Find out which one is longer, we always need a longer number to be on top and shorter one to be on bottom, so if the second is longer than first, we swap two numbers.
3. Next, we need to iterate through every digit, from left to right of both numbers and add each pair of digits along with a carried digit.
4. Append the left-most digit of the sum of each pair of digits into the return variable 'sum'. Put rest of the digits into 'carry'
5. When we reach the rightmost digit, don't need to split sum of pair of digits into carry and a leftmost digits. we can just append that final sum to our return variable 'sum'.

Once we're done iterating, we get the sum of two numbers in the form of a string.

demo of how the function works

```
console.log(add("5987342879234789234897", "23489072349807239487")); // will return  
6010831951584596474384
```

Another example,

```
console.log(add("893427328497983427893248932498034289324", "234859234879342897893427893274"));  
// will return 893427328732842662772591830391462182598
```