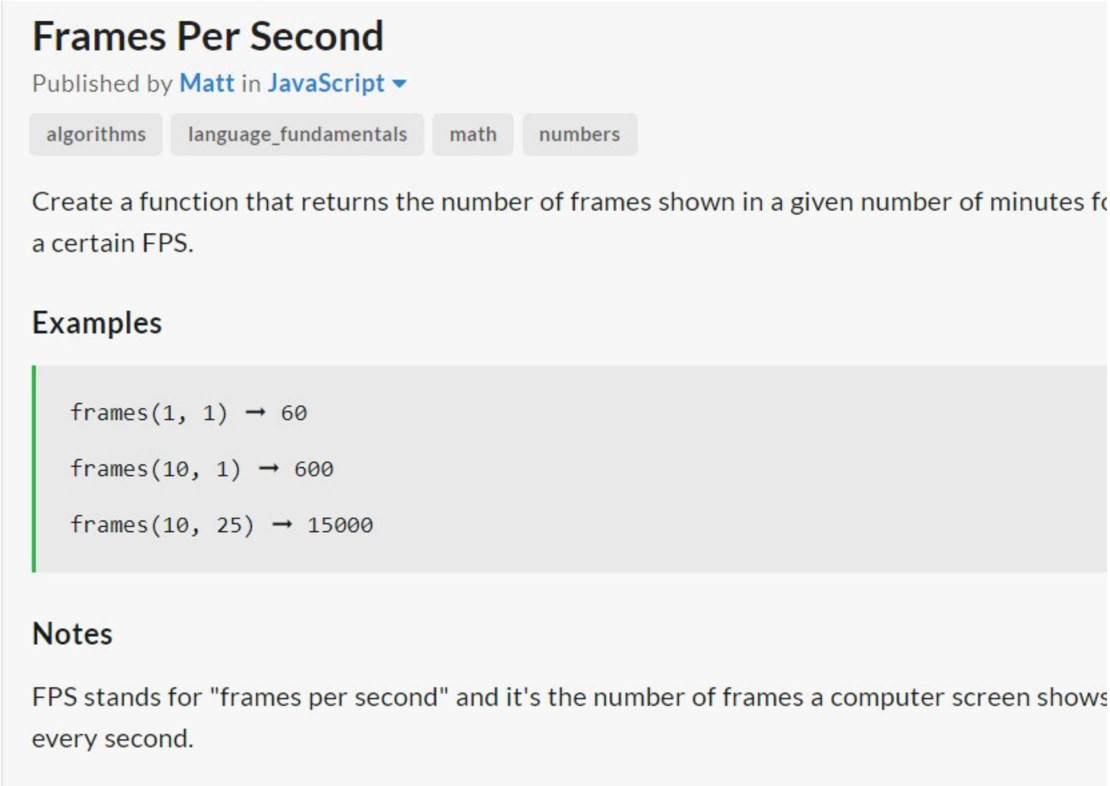
Question 1:



Frames Per Second

Published by Matt in JavaScript

Create a function that returns the number of frames shown in a given number of minutes fo a certain FPS.

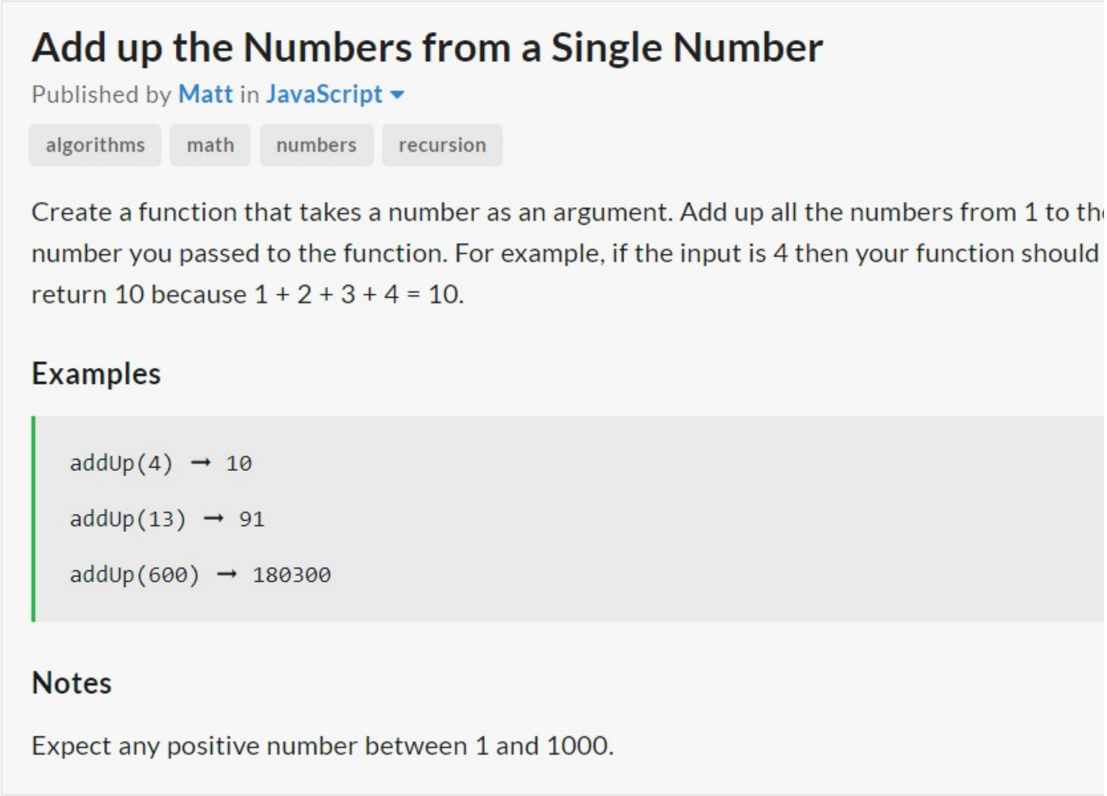
Examples

```
frames(1, 1) ➞ 60

frames(10, 1) ➞ 600

frames(10, 25) ➞ 15000
```

Notes

FPS stands for "frames per second" and it's the number of frames a computer screen shows every second.

Question 2:



Add up the Numbers from a Single Number

Published by Matt in JavaScript ▾

Create a function that takes a number as an argument. Add up all the numbers from 1 to th
number you passed to the function. For example, if the input is 4 then your function should
return 10 because 1 + 2 + 3 + 4 = 10.

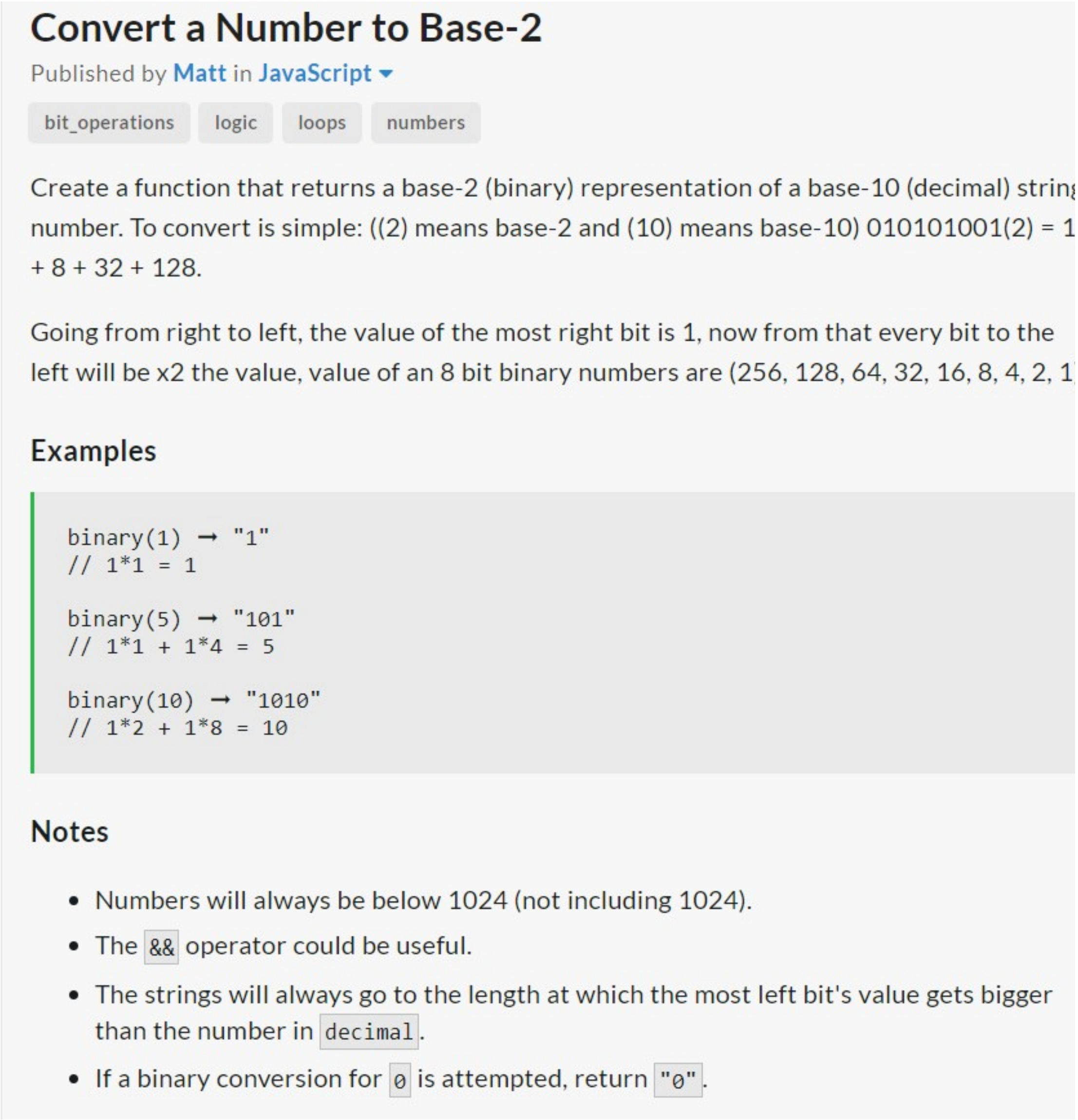
Examples

```
addUp(4) ➞ 10

addUp(13) ➞ 91

addUp(600) ➞ 180300
```

Notes

Expect any positive number between 1 and 1000.

Question 3:



Convert a Number to Base-2

Published by Matt in JavaScript ▾

bit_operations logic loops numbers

Create a function that returns a base-2 (binary) representation of a base-10 (decimal) string number. To convert is simple: ((2) means base-2 and (10) means base-10) 010101001(2) = 1 + 8 + 32 + 128.

Going from right to left, the value of the most right bit is 1, now from that every bit to the left will be x2 the value, value of an 8 bit binary numbers are (256, 128, 64, 32, 16, 8, 4, 2, 1)

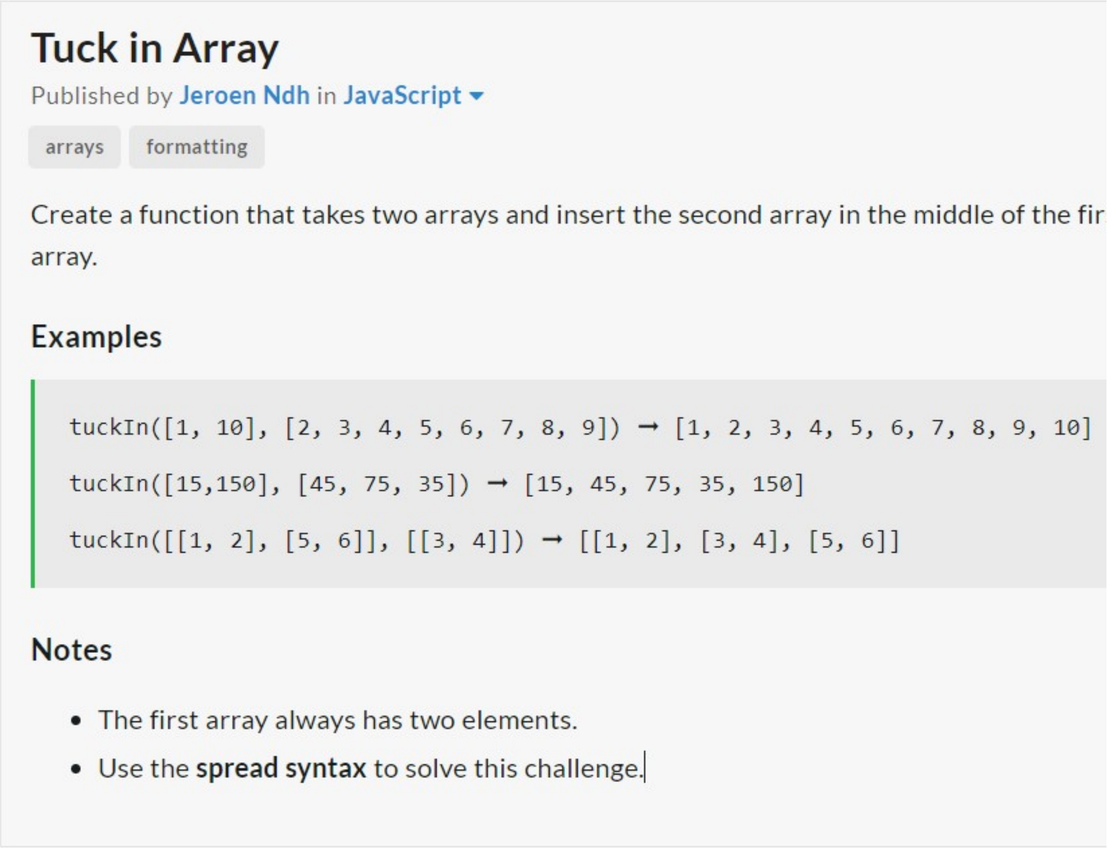
Examples

```
binary(1) ➞ "1"
// 1*1 = 1

binary(5) ➞ "101"
// 1*1 + 1*4 = 5

binary(10) ➞ "1010"
// 1*2 + 1*8 = 10
```

Notes

- Numbers will always be below 1024 (not including 1024).
- The `&&` operator could be useful.
- The strings will always go to the length at which the most left bit's value gets bigger than the number in `decimal`.
- If a binary conversion for `0` is attempted, return `"0"`.

Question 4:



Tuck in Array

Published by Jeroen Ndh in JavaScript

arrays formatting

Create a function that takes two arrays and insert the second array in the middle of the fir array.

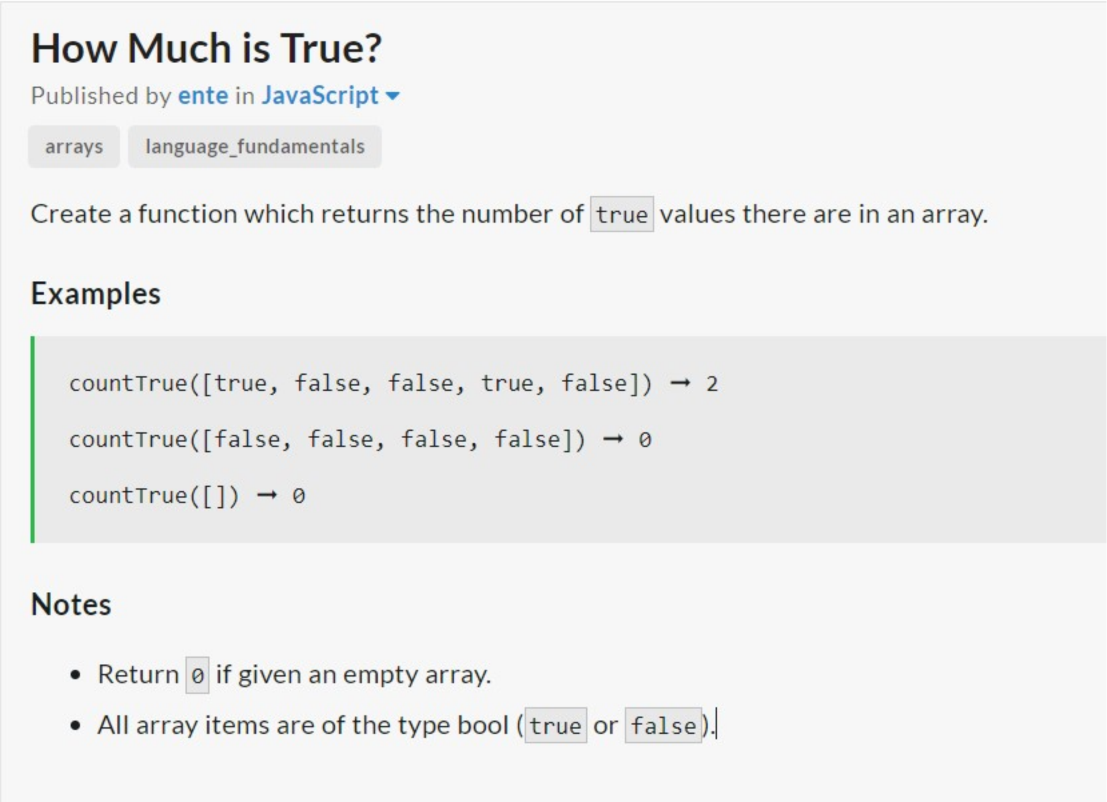
Examples

```
tuckIn([1, 10], [2, 3, 4, 5, 6, 7, 8, 9]) ➞ [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

tuckIn([15,150], [45, 75, 35]) ➞ [15, 45, 75, 35, 150]

tuckIn([[1, 2], [5, 6]], [[3, 4]]) ➞ [[1, 2], [3, 4], [5, 6]]
```

Notes

- The first array always has two elements.
- Use the **spread syntax** to solve this challenge.

Question 5:



How Much is True?

Published by ente in JavaScript

Create a function which returns the number of `true` values there are in an array.

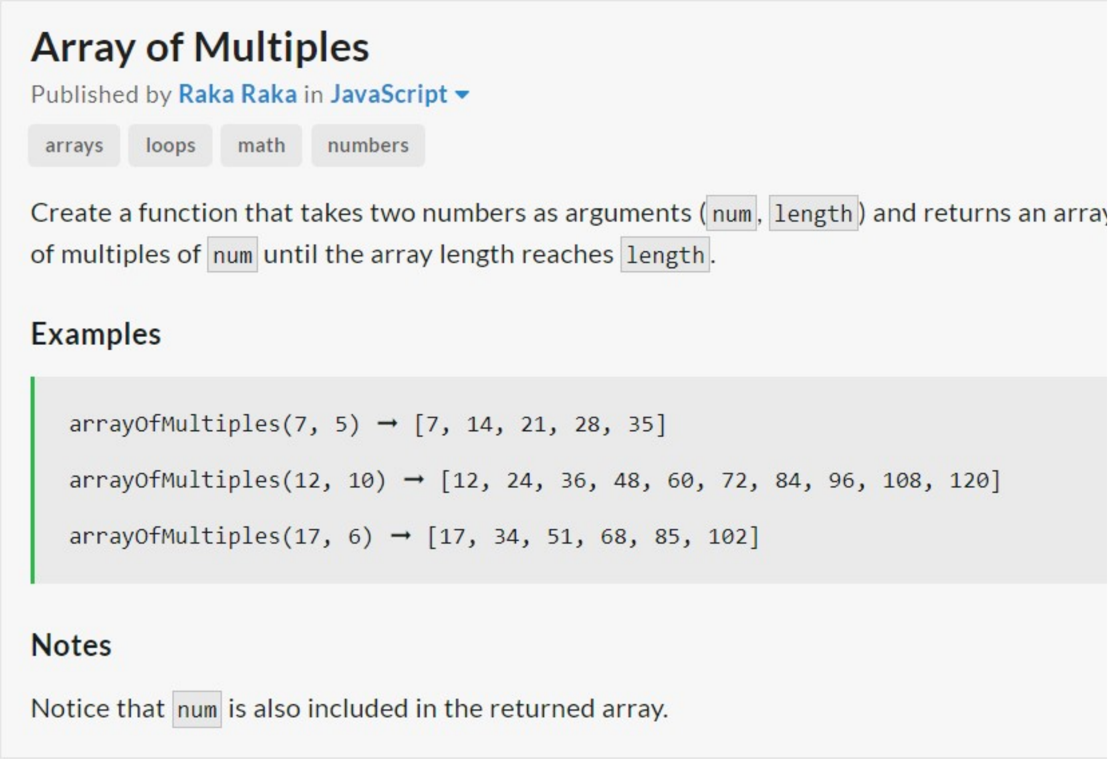
Examples

```
countTrue([true, false, false, true, false]) → 2

countTrue([false, false, false, false]) → 0

countTrue([]) → 0
```

Notes

- Return `0` if given an empty array.
- All array items are of the type bool (`true` or `false`).

Question 6



Array of Multiples

Published by Raka Raka in JavaScript ▾

arrays loops math numbers

Create a function that takes two numbers as arguments (`num`, `length`) and returns an array of multiples of `num` until the array length reaches `length`.

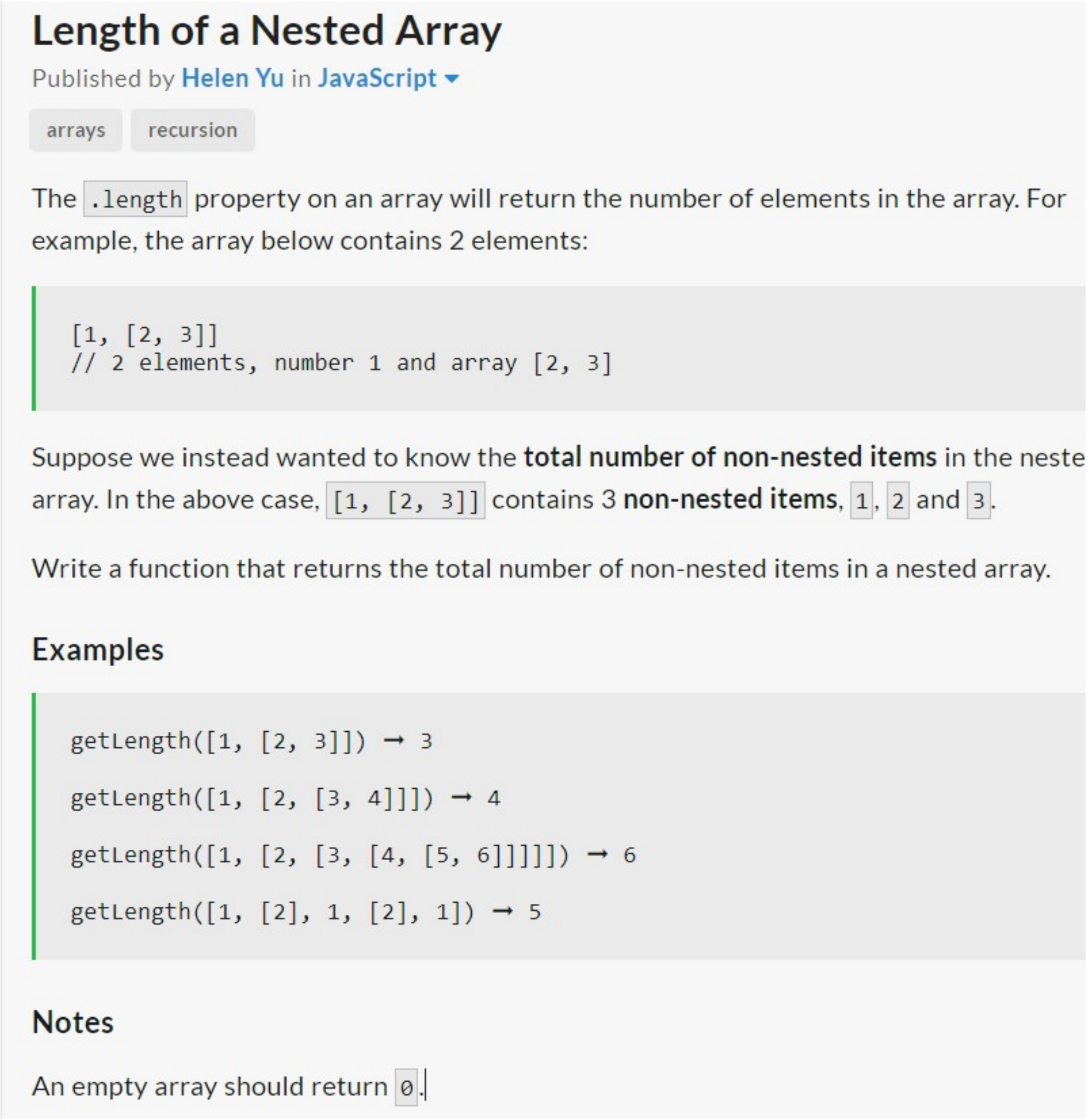
Examples

```
arrayOfMultiples(7, 5) ➞ [7, 14, 21, 28, 35]

arrayOfMultiples(12, 10) ➞ [12, 24, 36, 48, 60, 72, 84, 96, 108, 120]

arrayOfMultiples(17, 6) ➞ [17, 34, 51, 68, 85, 102]
```

Notes

Notice that `num` is also included in the returned array.

Question 7



Length of a Nested Array

Published by Helen Yu in JavaScript ▾

arrays recursion

The `.length` property on an array will return the number of elements in the array. For example, the array below contains 2 elements:

```
[1, [2, 3]]
// 2 elements, number 1 and array [2, 3]
```

Suppose we instead wanted to know the **total number of non-nested items** in the neste array. In the above case, `[1, [2, 3]]` contains 3 **non-nested items**, `1`, `2` and `3`.

Write a function that returns the total number of non-nested items in a nested array.

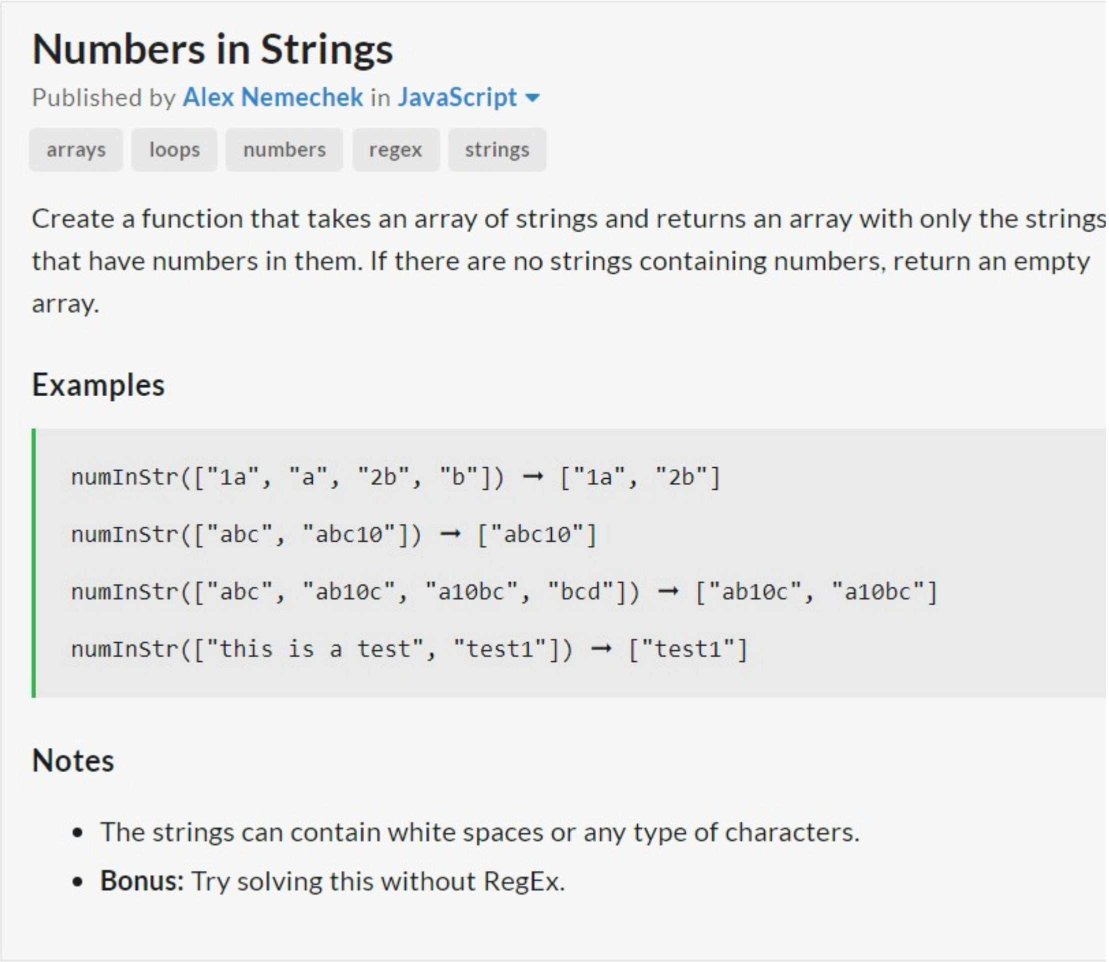
Examples

```
getLength([1, [2, 3]]) ➞ 3

getLength([1, [2, [3, 4]]]) ➞ 4

getLength([1, [2, [3, [4, [5, 6]]]]]) ➞ 6

getLength([1, [2], 1, [2], 1]) ➞ 5
```

Notes

An empty array should return `0`.

Question 8:



Numbers in Strings

Published by Alex Nemechek in JavaScript ▾

Create a function that takes an array of strings and returns an array with only the strings that have numbers in them. If there are no strings containing numbers, return an empty array.

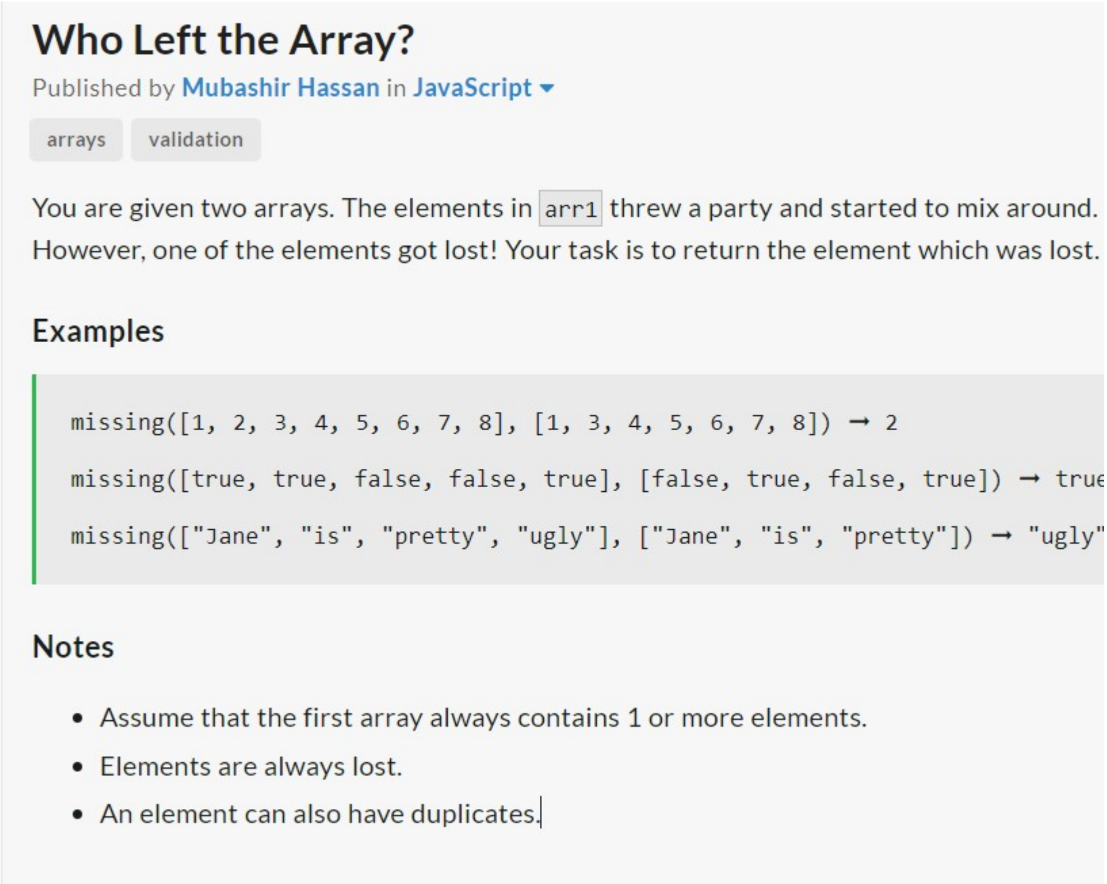
Examples

```
numInStr(["1a", "a", "2b", "b"]) ➞ ["1a", "2b"]

numInStr(["abc", "abc10"]) ➞ ["abc10"]

numInStr(["abc", "ab10c", "a10bc", "bcd"]) ➞ ["ab10c", "a10bc"]

numInStr(["this is a test", "test1"]) ➞ ["test1"]
```

Notes

- The strings can contain white spaces or any type of characters.
- **Bonus:** Try solving this without RegEx.

Question 9:



Who Left the Array?

Published by Mubashir Hassan in JavaScript ▾

arrays validation

You are given two arrays. The elements in `arr1` threw a party and started to mix around. However, one of the elements got lost! Your task is to return the element which was lost.

Examples

```
missing([1, 2, 3, 4, 5, 6, 7, 8], [1, 3, 4, 5, 6, 7, 8]) ➞ 2

missing([true, true, false, false, true], [false, true, false, true]) ➞ true

missing(["Jane", "is", "pretty", "ugly"], ["Jane", "is", "pretty"]) ➞ "ugly"
```

Notes

- Assume that the first array always contains 1 or more elements.
- Elements are always lost.
- An element can also have duplicates.