freeCodeCamp(🔥)

Sum All Numbers in a Range

We'll pass you an array of two numbers. Return the sum of those two numbers and all numbers between them.

The lowest number will not always come first.

Remember to use Read-Search-Ask if you get stuck. Try to pair program. Write your own code.

Here are some helpful links:

Math.max()

Math.min()

Array.reduce()

```
1
2 function sumAll(arr) {
3   return 1;
4 }
5
6 sumAll([1, 4]);
7
```

Run tests (ctrl + enter)

Reset | Help | Bug

Sum the Range

1,4

= 1 +2+3+4=10

The **`Math.max()`** function returns the largest of zero or more numbers.

Syntax

```
Math.max([value1[, value2[, ...]]])
```

Parameters

`value1, value2, ...`
Numbers.

Description

Because `max()` is a static method of `Math`, you always use it as `Math.max()`, rather than as a method of a `Math` object you created (`Math` is not a constructor).

If no arguments are given, the result is `-Infinity`.

If at least one of arguments cannot be converted to a number, the result is `NaN`.

Examples

Using `Math.max()`

```
Math.max(10, 20);   //  20
Math.max(-10, -20); // -10
Math.max(-10, 20);  //  20
```

The **`Math.min()`** function returns the smallest of zero or more numbers.

Syntax

```
Math.min([value1[, value2[, ...]]])
```

Parameters

`value1, value2, ...`
Numbers.

Description

Because `min()` is a static method of `Math`, you always use it as `Math.min()`, rather than as a method of a `Math` object you created (`Math` is not a constructor).

If no arguments are given, the result is `Infinity`.

If at least one of arguments cannot be converted to a number, the result is `NaN`.

Examples

Using `Math.min()`

This finds the min of `x` and `y` and assigns it to `z`:

```
var x = 10, y = -20;
var z = Math.min(x, y);
```

Sum all the values of an array

```
1 | var total = [0, 1, 2, 3].reduce(function(a, b) {
2 |   return a + b;
3 | });
4 | // total == 6
```

Reduces the numbers in an array to one number using the math function assigned

How to solve?

How can we get from the minimum number to the maximum?

in a Range

of two numbers.
two numbers
n them.

not always

-Search-Ask
pair program.

nks:

```
//add numbers in a range--find start and stop(math min and math max)
//how to get from start to stop--loop

function sumAll(arr){
  var max = Math.max(arr[0], arr[1]);
   //uses Math.max method to find max through index of array
  var min = Math.min.apply(null,arr);
  //uses Math library to find min through entire array
  var total = 0;
  //hold our count for our loop
  for(var i= min; i<=max; i++) {
    total += i;
  }

  //basic for loop...starts at min, iterates through maxium...adds 1(goes to next time) total=add each iteration

  return total;
}

sumAll([1,4]);
```

+ enter)

Using the Reduce Method

More Examples

Example

Add more than one item:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Kiwi", "Lemon", "Pineapple");
```

The output of the code above will be:

```
Banana,Orange,Apple,Mango,Kiwi,Lemon,Pineapple
```

Try it Yourself »

JavaScript Array Reference

```
function sumAll(arr) {
  var max = Math.max(arr[0], arr[1]);
  var min = Math.min.apply(null, arr);
  var newArray = [];
 // holds the numbers from our for loop
  for(var i= max; i>=min; i--) {
    newArray.push(i);
  }
  //iterates through the array and pushes the numbers into our newArray
  return newArray.reduce(function(a, b) {
  return a + b;
    //reduces our newArray into one number (sums it because of +)
});
}
sumAll([1,4]);
```

ks:

```
function sumAll(arr) {
  // Step 1. We use the Math.min() method to return the smallest number of the array
  var minimum = Math.min(arr[0], arr[1]); // In this case arr[0] => 1

  // Step 2. We use the Math.max() method to return the largest number of the array
  var maximum = Math.max(arr[0], arr[1]); // In this case arr[1] => 4

  // Step 3. We need to know all the numbers in the array to return the total
    // We create an empty array that will store the numbers
    var allNumbers = []; // => [1,2,3,4]

    // We create a FOR loop that will push each numbers into the empty array
    var num = minimum; // We start with the smallest number of the array
    var len = maximum;

    for (; num <= len; num++) {
      allNumbers.push(num); // Each iteration will push the new number at the end of the empty array
    }

  // Step 4. We use the reduce() method to sum all the numbers
  var sum = allNumbers.reduce(function(a, b) {
    return a + b; // => 1 + 2 + 3 + 4
  });

  return sum; // => 10

}

sumAll([1, 4]);
```

enter)

Bug

]) should

]) should

]) should

0]) should

5]) should