

JavaScript Jewels & Stones

Challenge

You're given strings `jewels` representing the types of stones that are jewels, and `stones` representing the stones you have. Each character in `stones` is a type of stone you have. You want to know how many of the stones you have are also jewels.

Letters are case sensitive, so `'a'` is considered a different type of stone from `'A'`.

1st Example

```
Input: jewels = 'aA', stones = 'aAAbbbb'
Output: 3
```



2nd Example

```
Input: jewels = 'z', stones = 'ZZ'
Output: 0
```



Constraints

- `1 <= jewels.length, stones.length <= 50`

- `jewels` and `stones` consist of only English letters.
- All the characters of `jewels` are unique.

Solution

```
const numJewelsInStones = (jewels, stones) => {  
  const hashMap = new Map();  
  let output = 0;  
  
  for (let i = 0; i < jewels.length; i++) {  
    hashMap.set(jewels[i], i);  
  }  
  
  for (let i = 0; i < stones.length; i++) {  
    if (hashMap.has(stones[i]))  
      output++;  
  }  
  
  return output;  
};
```



Explanation

I've coded a function called `numJewelsInStones` that takes two parameters: `jewels` and `stones`. Its purpose is to count the number of jewels (characters) from the `jewels` parameter that are present in the `stones` parameter, and then return the count.

Inside the function, a new `Map` object called `hashMap` is created.

This `Map` object will be used to store key-value pairs.

A variable called `output` is initialized to `0`. This variable will be used to store the count of jewels found in stones.

The first for loop iterates through each character in the `jewels` parameter. It uses the `set` method of the `Map` object to add each character as a key, with the value being the index of that character in the `jewels` parameter.

The second for loop iterates through each character in the `stones` parameter. It checks if the `hashMap` contains the current character as a key using the `has` method of the `Map` object.

If the current character in `stones` is found in the `hashMap`, the `output` variable is incremented by `1`.

After both loops have finished executing, the final value of the `output` variable is returned as the result of the function. This value represents the count of jewels found in stones.

In summary, the `numJewelsInStones` function counts the number of jewels present in stones by using a `Map` object to store the jewels and their indices. It then iterates through the stones, checking if each stone is a jewel, and increments the count accordingly.