

JavaScript First Unique Character in a String

Challenge

Given a string `s`, find the first non-repeating character in it and return its index. If it does not exist, return `-1`.

1st Example

Input: `s = 'super'`
Output: `0`



2nd Example

Input: `s = 'supsuperklok'`
Output: `6`



3rd Example

Input: `s = 'aabb'`
Output: `-1`



Constraints

- `1 <= s.length <= 105`
- `s` consists of only lowercase English letters.

Solution

```
const firstUniqChar = (s) => {  
  const map = {};  
  
  for (let char of s) {  
    map[char] ? map[char]++ : map[char] = 1;  
  }  
  
  for (let i = 0; i < s.length; i++) {  
    if (map[s[i]] === 1) return i;  
  }  
  
  return -1;  
};
```



Explanation

I've written a function called `firstUniqChar` that takes a string `s` as input. Its purpose is to find the index of the first unique character in the string. If there is no unique character, it returns `-1`.

Inside the function, an empty object called `map` is initialized. This object will be used to store the count of each character in the string.

A `for` loop is used to iterate through each character `char` in the string `s`. Within the loop, a ternary operator is used to check if the character `char` already exists as a key in the `map` object. If it exists, the count is incremented by `1`. If it doesn't exist, a new key `char`

is added to the `map` object with a value of `1`.

After the first loop, the `map` object will contain the count of each character in the string.

Another `for` loop is used to iterate through each character in the string using the index `i`. Within this loop, an if statement is used to check if the count of the character at index `i` in the `map` object is equal to `1`. If it is, it means that the character is unique and the index `i` is returned.

If no unique character is found in the string, the function returns `-1`.

In summary, the `firstUniqChar` function counts the occurrences of each character in the string using an object, and then finds and returns the index of the first unique character in the string.