JavaScript Isomorphic Strings

Challenge

Given two strings `s` and `t`, determine if they are isomorphic.

Two strings `s` and `t` are isomorphic if the characters in `s` can be replaced to get `t`.

All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character, but a character may map to itself.

1st Example

```
Input: s = 'egg', t = 'add'
Output: true
```

2nd Example

```
Input: s = 'warm', t = 'cool'
Output: false
```

3rd Example

```
Input: s = 'paper', t = 'title'
Output: true
```

Constraints

- `1 <= s.length <= 5 * 10⁴`
- `t.length == s.length`
- `s` and `t` consist of any valid ascii character.

Solution

```
const isIsomorphic = (s, t) => {
    const hash1 = {},
          hash2 = {};

    for (let i = 0; i < s.length; i++) {
        if (hash1[s[i]] !== hash2[t[i]]) {
            return false;
        }

        hash1[s[i]] = i;

        hash2[t[i]] = i;
    }

    return true;
};
```

Explanation

I've created a function called `isIsomorphic` that takes in two strings `s` and `t` as parameters. Its purpose is to check if the two strings are isomorphic, meaning that each character in `s` can be replaced by a corresponding character in `t` while preserving the order of characters.

Inside the function, two empty objects called `hash1` and `hash2` are declared. These objects will be used to map characters from `s` and `t`, respectively.

The function then enters a loop that iterates through each character in `s`. The loop variable `i` represents the index of the current character.

Within the loop, it checks if the corresponding characters in `s` and `t` have different values in `hash1` and `hash2`, respectively. If the values are different, it means that the characters are not isomorphic, so the function returns `false` immediately.

If the characters are isomorphic, the function updates values in `hash1` and `hash2` to be the index `i`.

After the loop ends, it means that all characters in `s` and `t` have been checked and found to be isomorphic. Therefore, the function returns `true`.

In summary, the `isIsomorphic` function checks if two strings are isomorphic by using two hash tables to map characters from each string and comparing their values. If any mismatch is found, it returns `false`, otherwise it returns `true`.

Author: Trevor Morin