# JavaScript Reverse Vowels of a String

## Challenge

Given a string `s`, reverse only all the vowels in the string and return it.

The vowels are `'a'`, `'e'`, `'i'`, `'o'`, and `'u'`, and they can appear in both cases.

### 1st Example

```
Input: s = 'hello'
Output: 'holle'
```

### 2nd Example

```
Input: s = 'leetcode'
Output: 'leotcede'
```

### Constraints

- `1 <= s.length <= 3 * 10^5`
- `s` consist of printable ASCII characters.

## Solution

```
const reverseVowels = (s) => {
    let vowels = ['a', 'A',
                  'e', 'E',
                  'i', 'I',
                  'o', 'O',
                  'u', 'U'],
        stack  = [];

    for (let i = 0; i < s.length; i++) {
        if (vowels.indexOf(s[i]) !== -1) {
            stack.push(s[i]);
        }
    }

    let ans = [];

    for (let i = 0; i < s.length; i++) {
        if (vowels.indexOf(s[i]) !== -1) {
            ans.push(stack.pop());
        } else {
            ans.push(s[i]);
        }
    }

    return ans.join('');
};
```

## Explanation

I've coded a function called `reverseVowels` that takes a string `s` as input. The purpose of this function is to reverse the vowels in the input string and return the modified string.

The function begins by creating an array called `vowels` that contains all the vowels in both uppercase and lowercase.

Next, an empty array called `stack` is created. This array will be used to store the vowels encountered in the input string.

A `for` loop is used to iterate through each character of the input string `s` . Within the loop, an `if` statement checks if the current character is a vowel by using the `indexOf` method to search for the character in the `vowels` array. If the character is a vowel, it is pushed into the `stack` array.

After that, an empty array called `ans` is created. This array will store the modified string with reversed vowels.

Another `for` loop is used to iterate through each character of the input string `s` . Within this loop, an `if` statement checks if the current character is a vowel by using the `indexOf` method to search for the character in the `vowels` array. If the character is a vowel, it is replaced with the last vowel in the `stack` array by using the `pop()` method. This effectively reverses the order of the vowels.

If the current character is not a vowel, it is simply pushed into the `ans` array.

Finally, the `ans` array is converted to a string using the `join()` method with an empty string as the separator. The resulting modified string is then returned as the output of the function.

In summary, this function iterates through the input string twice. The first iteration collects the vowels in the `stack` array, and the

second iteration replaces the vowels in the input string with the vowels from the `stack` array in reverse order. The modified string is then returned as the output of the function.

Author: Trevor Morin