JavaScript Valid Palindrome

Challenge

A phrase is a palindrome if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include letters and numbers.

Given a string `s`, return `true` if it is a palindrome, or `false` otherwise.

1st Example

```
Input: s = 'A man, a plan, a canal: Panama'
Output: true
Explanation: 'amanaplanacanalpanama' is a palindrome.
```

2nd Example

```
Input: s = 'race a car'
Output: false
Explanation: 'raceacar' is not a palindrome.
```

3rd Example

```
Input: s = ' '
Output: true
Explanation: s is an empty string '' after
             removing non-alphanumeric characters.
             Since an empty string reads the same
             forward and backward, it is a palindrome.
```

Constraints

- `1 <= s.length <= 2 * 10^5`
- `s` consists only of printable ASCII characters.

Solution

```
const isPalindrome = (s) => {
    let sanitizedString = s.replace(/\W|_/g,''),
        reversedString  = sanitizedString.split('')
                                         .reverse()
                                         .join('');

    return sanitizedString.toLowerCase() == reversedString
                                            .toLowerCase();
};
```

Explanation

I've built a function expression, using ES6 syntax, that is defined by an anonymous function that accepts `s` as a string of words.

Within the anonymous function, I establish a variable for the input string called `sanitizedString` which takes the string `s` and applies the string prototype method `replace` to remove the non-alphanumeric characters with the help of regex.

I also set a variable called `reversedString` to reverse the `sanitizedString`. I apply the string prototype method `split` to create an array for every character of the `sanitizedString`, then I use the array prototype `reverse` method to reverse the order of the string characters within the array, and then I use the array prototype `join` method to turn the reversed character array back into a string that can be referred to as `reversedString`.

The `sanitizedString` and `reversedString` variables have the string prototype `toLowerCase` method applied to make sure all the characters are in lower case. The two variables are then compared to each other for basic equality and a Boolean value is returned.

Author: Trevor Morin