JavaScript Reverse Words in a String III

Challenge

Given a string `s`, reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

1st Example

```
Input: s = 'Here's a string of words'
Output: 'sdrow fo gnirts a s'ereH'
```

2nd Example

```
Input: s = 'Superklok Labs'
Output: 'sbaL kolkrepuS'
```

Constraints

- `1 <= s.length <= 5 * 10⁴`
- `s` contains printable ASCII characters.
- `s` does not contain any leading or trailing spaces.

- There is at least one word in `s`.
- All the words in `s` are separated by a single space.

Solution

```
const reverseWords = (s) => {
    return s.split(' ').map((w) => w.split('')
                                    .reverse()
                                    .join(''))
                                    .join(' ');
};
```

Explanation

I've written 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, the string `s` is split into an array of string words everywhere there is a space `' '` by using the string prototype `split` method.

The array prototype `map` method is then applied to the array that was returned by the `split` method. The string words `w` in the array are then split into their own arrays of string characters using the string prototype `split` method at `''`.

The string character arrays are then reversed using the array prototype `reverse` method. The newly reversed arrays are then

joined together to form words `w` again, and this process is effectuated for each word from the array derived from `s`.

Once the array prototype `map` method is complete, the reversed words are then joined into a string using the array prototype `join` method that adds a space `' '` between each reversed word.

Author: Trevor Morin