# JavaScript Uncommon Words from Two Sentences

## Challenge

A sentence is a string of single-space separated words where each word consists only of lowercase letters.

A word is uncommon if it appears exactly once in one of the sentences and does not appear in the other sentence.

Given two sentences `s1` and `s2`, return a list of all the uncommon words. You may return the answer in any order.

### 1st Example

```
Input: s1 = 'this apple is sweet',
       s2 = 'this apple is sour'
Output: ['sweet','sour']
```

### 2nd Example

```
Input: s1 = 'apple apple',
       s2 = 'banana'
Output: ['banana']
```

### Constraints

- `1 <= s1.length, s2.length <= 200`

- `s1` and `s2` consist of lowercase English letters and spaces.
- `s1` and `s2` do not have leading or trailing spaces.
- All the words in `s1` and `s2` are separated by a single space.

## Solution

```
const uncommonFromSentences = (s1, s2) => {
    const arrayS1 = s1.split(' '),
          arrayS2 = s2.split(' '),
          hashMap = new Map (),
          output  = [];

    for (let i = 0; i < arrayS1.length; i++) {
        if (!hashMap.has(arrayS1[i])) {
            hashMap.set(arrayS1[i], 1);
        } else {
            hashMap.set(arrayS1[i],
                        hashMap.get(arrayS1[i]) + 1);
        }
    }

    for (let i = 0; i < arrayS2.length; i++) {
        if (!hashMap.has(arrayS2[i])) {
            hashMap.set(arrayS2[i], 1);
        } else {
            hashMap.set(arrayS2[i],
                        hashMap.get(arrayS2[i]) + 1);
        }
    }
```

Solution continues on next page...

```
    hashMap.forEach((value, key) => {
        if (value === 1) output.push(key);
    });

    return output;
};
```

## Explanation

I've written a function called `uncommonFromSentences` that takes two strings, `s1` and `s2`, as input. Its purpose is to find the uncommon words between the two sentences.

Inside the function, the strings `s1` and `s2` are split into arrays of words using the `split()` method and assigned to `arrayS1` and `arrayS2` respectively.

A new `Map` object called `hashMap` is created to store the words and their frequencies.

An empty array called `output` is initialized to store the uncommon words.

The function then iterates through each word in `arrayS1` using a `for` loop. It checks if the word is already present in the `hashMap` using the `has()` method. If the word is not present, it adds it as a key in the `hashMap` with a value of `1` using the `set()` method. If the word is already present, it increments the value of the key by `1` using the `get()` and `set()` methods.

Next, the function does the same iteration and check for each word in `arrayS2`. It checks if each word is already present in the `hashMap` and adds it if not or increments its value if already present.

After that, the function uses the `forEach()` method on the `hashMap` to iterate through each key-value pair. Inside the `forEach` callback function, it checks if the value is equal to `1`. If the value is `1`, it means that the corresponding word is unique and not present in the other string. In this case, it adds the word to the output array using the `push()` method.

Finally, the function returns the output array containing all the unique words that are uncommon between the two sentences.

In summary, the `uncommonFromSentences` function finds the uncommon words between two sentences by using a `Map` object to count the frequencies of words in the sentences. It then iterates through the `Map` and adds the words with a frequency of `1` to an output array, which is returned as the result.

Author: Trevor Morin