**Mini Linter**

In this project, you will use what you know about iterating over arrays to improve the quality of a paragraph and gather some information about that paragraph.

This is the same type of work that word processing software does. Additionally, you may have heard of *linting*, a process by which text is evaluated and improved by an application. In this project, you will create a miniature version of a linter using array methods that you have learned.

If you get stuck during this project or would like to see an experienced developer work through it, click “**Get Help**“ to see a **project walkthrough video**.

**Tasks**

**8/8Complete**

Mark the tasks as complete by checking them off

**1.**

In the code editor, there is a string called story. We want to gather some information about the individual words and sentences in the string. Let’s split the string into individual words and save them in a new array called storyWords.

Hint

string.split(' ');

**2.**

Log how many words there are in this story to the console.

Hint

Find the length of the storyWords array.

**3.**

There is an array of words that are unnecessary. Iterate over your array to filter out these words. Save the remaining words in an array called betterWords. There are several ways that you could achieve this.

Hint

As you iterate over storyWords (you can use the .filter() method to do this), if unnecessaryWords does not include the current word, you can return that word to the betterWords array.

The .includes() method may prove useful.

**4.**

There is an array of words called overusedWords. These are words overused in this story. You want to let the user of your program know how many times they have used these overused words. There are two ways to achieve this. Try it on your own first. If you need help, consult the hint.

Hint

1. You can iterate over the betterWords array three separate times (once for each of the words in the overusedWords array). Create a variable that represents the total times that word appears. Add 1 to the variable every time the current word is the same as that word.
2. You can make this simpler by using one if, and two else if statements in the function code block of your iterator. That way, you can gather the counts of all three overused words at one time.

**5.**

Now, count how many sentences are in the paragraph.

This may seem tricky, but remember that all of the sentences in this paragraph end with a period (.) or an exclamation mark (!). You could iterate over the array and add 1 to a sentence counter variable for each word that has a period or exclamation mark as its final character.

Hint

let sentences = 0;

array.forEach(word => {

if (word[word.length-1] === '.' || word[word.length-1] === '!') {

sentences+=1;

}

});

**6.**

Log these items to the console:

* The word count
* The sentence count
* The number of times each overused word appears

You could choose to simply log them one by one or, for a challenge, create a function that logs all of them with some formatting.

Hint

const logInfo = (param1, param2, param3) => {

console.log('Word count: ' + param1);

//The rest of your code goes here

};

**7.**

Now, log the betterWords array to the console as a single string.

Hint

array.join(' ')

**8.**

Congratulations! You’ve improved the original paragraph and given the user some important information about his or her work. Think about ways in which you can extend this project, potentially by using other JavaScript knowledge you have.

Here are some ideas:

1. For the overused words, remove it every other time it appears.
2. Write a function that finds the word that appears the greatest number of times.
3. Replaced overused words with something else.