Code Academy Grammar Checker

**Start**

*let* story =

'Last weekend, I took literally the most beautifull bike ride of my life. The route is called "The 9W to Nyack" and it stretches all the way from Riverside Park in Manhattan to South Nyack, New Jersey. It\'s really an adventure from beginning to end! It is a 48 mile loop and it literally took me an entire day. I stopped at Riverbank State Park to take some artsy photos. It was a short stop, though, because I had a freaking long way to go. After a quick photo op at the very popular Little Red Lighthouse I began my trek across the George Washington Bridge into New Jersey. The GW is a breathtaking 4,760 feet long! I was already very tired by the time I got to the other side. An hour later, I reached Greenbrook Nature Sanctuary, an extremely beautifull park along the coast of the Hudson. Something that was very surprising to me was that near the end of the route you literally cross back into New York! At this point, you are very close to the end.';

*let* storyWords = story.split(" ");

*let* unnecessaryWord = "literally";

*let* misspelledWord = "beautifull";

*let* badWord = "freaking";

1) In main.js the ‘story’ variable holds the paragraph that needs to be edited.

In order to edit the story, we turn it into an array on line 3. The .split() method separates the story string by suing the space character (‘ ‘) and stores each word as an element of the array.

2) We now want to view the edited storyWords array as a string. To change the storyWords array back into a readable string, we can invoke the .join() method on storyWords.

The .join() method needs an argument of an empty space character (‘ ‘) to separate each array element with a space in the string.

console.log(storyWords.join(" "));

Place the .join(‘ ’) method invocation as an argument of a console.log() statement to log the final story to the console.

3) Above the console.log() that uses the .join() method, create a variable named count that stores the number 0.

*let* count = 0;

Directly below ‘count’, use a .forEach() method to iterate over the storyWords array. As an argument of the forEach() method, create an empty function to be used as the call-back function.

storyWords.forEach(() *=>* {

});

4) For each word in the storyWords array, we want the count variable to increment by one.

Add a parameter named word to the callback function of the .forEach() method to be used to store the current element when iterating over the storyWords array. Each time storyWord iterates, increment count by one.

Below the .forEach() method, log count to see how many words are in the story.

storyWords.forEach((*word*) *=>* {

console.log(count);

return count++;

});

5). 181 word count is too long for this story. It’s time to filter out all instances of the word ‘literally’ to shorten the story and remove the unnecessary words. We will reassign the filtered story to the same storyWords variable by applying the .filter() method. Throughout this project, we’ll use this approach of reassigning the storyWords variable for each revision of the story.

Below where you logged the count variable, reassign the sotryWords variable to equal the invocation of the .filter() method on the storyWords array. Give the .filter() method a callback function with a parameter of word.

storyWords = storyWords.filter((*word*) *=>* {

return *word* != unnecessaryWord;

});

6) Reassign the storyWords to equal the invocation of the .map() method on the storyWords array.

Set word as a parameter of .map()’s callback function.

In the callback’s body, define a conditional statement to check if the word argument is equal to the misspelledWord variable. If it is, return the correct spelling of the string, ‘beautiful’. If not return word.

storyWords = storyWords.map((*word*) *=>* {

return *word* === misspelledWord ? "beautiful" : *word*;

});

7). Apply the .findIndex() method to storyWords to find the index of a bad word.

Start by declaring a variable called badwWordIndex and setting it to the invocation of .findIndex() on the storyWords array.

The .findIndex() callback function should check each word to see if it equals the badWord variable declared on line 6, and return the index of the found word.

Log badWordIndex to the console.

*let* badWordIndex = storyWords.findIndex(*word* *=>* {

return *word* === badWord;

})

console.log(badWordIndex);

8) Now we have the index of the bad word, it needs to be replaced.

Access the element inside the storyWords array that has the index of badWordIndex using bracket notation. Set the accessed element equal to the more appropriate word; ‘really’.

*let* story =

'Last weekend, I took literally the most beautifull bike ride of my life. The route is called "The 9W to Nyack" and it stretches all the way from Riverside Park in Manhattan to South Nyack, New Jersey. It\'s really an adventure from beginning to end! It is a 48 mile loop and it literally took me an entire day. I stopped at Riverbank State Park to take some artsy photos. It was a short stop, though, because I had a really long way to go. After a quick photo op at the very popular Little Red Lighthouse I began my trek across the George Washington Bridge into New Jersey. The GW is a breathtaking 4,760 feet long! I was already very tired by the time I got to the other side. An hour later, I reached Greenbrook Nature Sanctuary, an extremely beautifull park along the coast of the Hudson. Something that was very surprising to me was that near the end of the route you literally cross back into New York! At this point, you are very close to the end.';

*let* storyWords = story.split(" ");

*let* unnecessaryWord = "literally";

*let* misspelledWord = "beautifull";

*let* badWord = "freaking";

*let* count = 0;

storyWords = storyWords.filter((*word*) *=>* {

return *word* != unnecessaryWord;

});

storyWords = storyWords.map((*word*) *=>* {

return *word* === misspelledWord ? "beautiful" : *word*;

});

*let* badWordIndex = storyWords.findIndex((*word*) *=>* {

return *word* === badWord;

});

console.log(`BadwordIndex: ${badWordIndex}`);

storyWords.forEach((*word*) *=>* {

//console.log(count);

return count++;

});

*const* lengthCheck = storyWords.every((*word*) *=>* {

*word*.length <= 10;

});

storyWords = storyWords.map((*word*) *=>* {

return *word*.length <= 10 ? *word* : "huge";

});

//console.log(storyWords);

console.log(storyWords.join(" "));