

Explanation of my Code Challenge Solution

First of all, I took a look at every challenge and thought about solutions, writing some pseudocode on paper to remember the things I thought about.

I found the challenge number 3 most interesting for myself.

Before starting I needed to look up what a pangram is.

First I needed to store the DOM elements in constants for easier use in the script.

Then I needed a baseline and some variables. I chose an array of key-value pairs containing each individual character of the alphabet along with the count of how often it is used in the sentence. With this array I was able to count if the character is not present, count being 0, the character is present once, count being 1 or the character is there more than once, count being more or equal 2.

The Main part of the function checking the sentence is the first for-loop:

It checks every character on an index position starting at 0 and ending when the sentence is over. Every character gets compared to the alphabet array. The first match in the comparison adds 1 to the count of that character in the key-value array and stops the check, causing the for loop to initiate the same check on the next index position until the sentence's end is reached.

After I checked how many times each character was present, I looked at the count value of each array element with a for loop and counted missing, onetime and multiple characters. The missing characters get pushed into a different array.

The basic determination if a pangram or a perfect pangram is present, works like this.

The alphabet has 26 characters. Having counted unique characters and unique onetime characters, I added them together. If the value equals 26, the pangram condition is set to true. If only the unique onetime characters equal 26 the perfect pangram condition gets set to true, too and it's a perfect pangram.

After having worked with the data, I modified the DOM elements innerHTML property to display the values. After that, I put in code to reset the values for consecutive runs, so you don't have to reload the page every time.

While coding, I also added some debug console logs to check the values. After the script was finished, I styled the page.