

## API Documentation

Function	Function Description	Inputs / Outputs
def read_file(filename):	This function is responsible for reading the contents of the user specified file	<b>Input:</b> filename (str) - Name of the input file specified by the user. <b>Output:</b> chars (str) - Content of the file.
def write_file(output_Filename, output_data):	This function is responsible for writing the results of this program to a new file, the name of the file will be specified by the user.	<b>Input:</b> output_Filename (str) - Name of the output file specified by the user. output_data (str) - Results of the program to be written to the output file. <b>Output:</b> None / returns the results of the program in the output file
def wordCount(chars):	This function counts the number of total words in the user-specified file.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> wordCount (int) - The total count of words in the file.
def num_sentences(chars):	This function counts the total number of sentences in the file, considering periods and question marks as sentence delimiters.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> num_of_sent (int) - The total number of sentences in the input file.
def avg_words_in_sent(chars):	This function calculates the average number of words in a sentence in the input file, returning 0 if the file is empty.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> avg_word_count (float) - The average number of words in a sentence in the entire file.
def longest_word(chars):	This function returns the longest word in the text file.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> longest (str) - The longest word in the input file.
def length_longest_word(chars):	This function returns the length of the longest word in the text file.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> len(longest) (int) - The length of the longest word in the input file.
def shortest_word(chars):	This function returns the shortest word in the text	<b>Input:</b> chars (str) - Content of the input file.

	file.	<b>Output:</b> shortest_word (str) - The shortest word in the input file.
def count_chars(chars):	This function calculates the frequency of each character in the file and saves it in a DataFrame.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> table (Pandas DataFrame) - A DataFrame containing the frequency of each character in the file and saves it in a DataFrame. Also, a new file called <b>alphabet_frequency.csv</b> will be created and it will contain the frequency of each "character" in the file.
def frequency(chars):	This function counts the total number of times each word appears in the input text file, considering word frequencies and removing stopwords and punctuation.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> word_freq (dict) - The total number of times each word appears in the input file.
def user_word_frequency(chars):	This function counts the total number of times a specific word, which is specified by the user, appears in the file.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> freq_of_word (str) - The frequency of the word specified by the user.
def frequent_words(chars):	This function counts the total number of times a word appears in the file, sorts them in descending order, and outputs the top 10 most occurring words.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> df (DataFrame) - A DataFrame consisting of the top 10 words that occurred most frequently in the file.
def term_frequency(chars)	This function is responsible for calculating the Term Frequency value for each word in the document and then displaying it as a data frame. Term frequency is the ratio of the number of times the word appears in a document compared to the total number of words in that document. It is an indicator of the term's importance in a given text.	<b>Input:</b> chars (str) - Content of the input file. <b>Output:</b> df (DataFrame) - A data frame consisting of the term frequencies of each word in the document.