HS23 LTWA
Excercise 4:
JavaScript

Notes on the submission:

- **Submission format:** zip-folder containing a folder called *word-analytics* containing the *img* folder as well as the js,html and css files.
- Filename in the following format: gitlab-name_ltwa_exnumber,
  e.g. *max.muster_ltwa_ex4*
- Submit your zip file via the exercise module on OLAT if you want us to take a look at it. The module is only open until Wednesday, October 25, 12:00.
- The exercises are **not graded**.

If you have problems or questions just post in the OLAT forum or ask in the tutorials. If it is personal or urgent you can also contact us by e-mail. **Good luck!**

Language Technologies Web Application HS23 — Lecturer: Johannes Graën
Department of Computational Linguistics — Tutors: Elina Stüssi, Lea Müller

1 Word Analytics

Imagine you work for a company that specializes in word analytics and your boss wants you to create a website where users can enter a noun and perform some basic analysis. You have already created the basic html and css files for the new website (see *word-analytics.html* and *word-analytics.css*). However, the website is not interactive yet. When a user enters a word and clicks on the *"Analyze!"* button, no analysis is performed and no results are displayed. This is because the linked script *word-analytics.js* is still empty. Add the desired functionality as described below and make sure the analysis is being performed when the button is clicked. Try to achieve this without changing the provided html file.

1. The analysis includes language identification based on the suffix of the entered noun. The IT department already has the necessary code from a previous project, but unfortunately it is written in Python (see language *identification.py*). Your task is to translate the Python code into JavaScript. To do this, add your new JavaScript function to the *word-analytics.js* file. Also make sure that this function gets called when the user clicks on the button.
   (Hint: Use **document.querySelector** to select elements in the DOM, and **addEventListener** to make them reactive to user interaction such as clicking.)

2. Your company would like to offer further analyses to its users. Add additional functions to the same *word-analytics.js* file that provide the following information about the entered noun:

   - Length of the word.
   - Whether the first letter of the word is capitalized or not.
   - Count of distinct letters.

- Calculation of the ratio of consonants to vowels in the word.
- Inversion of the case of the word (lowercase becomes uppercase, and vice versa).
- Calculation of the worth of the word, where a=1, b=2, . . . , z=26.

3. Display the results of the above described analyses (upon the click of the button) on your website in the corresponding placeholders.

4. The company is also interested to know to which extent the entered word is a palindrome. For an easy visualization, they want you to display a rectangle in the DOM for each letter of the word. If the letter helps to achieve a palindrome, a green rectangle should be displayed, if not a red rectangle should be displayed.
   E.g. 'dog' should be:  ,
   and 'Bathtub' should be:  .
   (Hint: For this task, first add a placeholder in the HTML file to contain the boxes. Write styles for green and red boxes in the CSS file. Then, using JavaScript, dynamically create boxes based on the user input and put them in the placeholder. You might need to look at the **document.createElement** and **appendChild** methods.)

Have fun!