

Lab 2

In this lab we will get started with using Javascript. You will work by adding your functions directly into the script tag of an HTML file. But first, we'll need to update your project.

Updating your project

- Start up GitKraken.
- Make sure that you have committed all files related to your project. When you look at your project in GitKraken, you should be seeing no WIP section and no modified.
- Right-click at the “upstream” item in the “REMOTE” section on the left side. Choose “Fetch upstream”. You should now be seeing that your graph in the middle has two “master branches”. One corresponds to “my repository” (the upstream) and the other corresponds to your master branch with the changes you’ve made on it for Lab1. This should be the “active” branch, with a little checkbox next to it. These two branches should appear to “deviate” from a common start labeled “Initial files for Lab 1”.
- Right-click *my* master branch (upstream). You can do so either in the main window or in the upstream->master section in REMOTE on the left. Choose “Merge upstream/master onto master”. You should see the two branches merge into a new “merge” commit.
- You are now ready to work on Lab 2! After you add your changes, save and create a commit in GitKraken, then push your changes.
- You will need to repeat these steps when a new lab is released.

Assignment

In your project you will find a Lab2 folder and an index.html file within that folder. You will find in that file a <script> section. You will add your Javascript code there to answer the following questions. Every time you save the file and reload the page on the browser, you can watch your progress.

The questions you need to answer follow. When you are satisfied with your answers, save the HTML and make a commit and push it.

1. Write a function “countWords” that takes as input one value, s, which is a string. It then returns an integer, the number of “words” in the string s. A “word” is defined as a sequence of one or more letters (a through z, A through Z) that extends as much as possible (i.e. until it encounters a non-letter character). The string and regular expressions section should help for this problem. If you have implemented this correctly, then typing text in the little text area window should correctly show you the number of words in the box next to it.

2. Write a function “fix2” that does the following: It takes as input an array of numbers, and looks through the array to find two values that are in decreasing order (i.e. the earlier value is bigger than the later value). It then swaps the two values in the array, and returns the value `true`. If there is no such pair of values, it returns `false`. If you have implemented this correctly, then when you load the page you will see a series of horizontal bars that will start swapping themselves until they are in decreasing order of length.
3. Write a “computeTotal” function. It is given as input an array `arr` whose entries are objects that have three properties: `name`, `quantity` and `price` (per item). Your function must return the total value (multiplying each quantity with the price and adding them all up), returning 0 for an empty array. If you have done this correctly, the table in the page should show the correct total (and you can play with the quantity/price values to see it dynamically change).
4. Exercise 4 asks you to write a function `surroundMatches`. It is given a string and a regular expression and it is asked to return the string where each match of the expression is surrounded in the tags `` and ``. If you have done this correctly, then the textarea and view below it should work correctly in highlighting the matches of the regular expression. You can change the textarea and/or regular expression and observe the changes in the text at the bottom.