

Lab 1

In this lab we will get started with using Javascript. You will download an HTML file, then add Javascript code in its `<script>` tag. When you open the HTML file in the browser, you will be able to see how you've done.

The file to use will be this¹. Download it and save it somewhere on your computer, then open it in both the web browser and in SublimeText.

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 email it to me.

1. Define an object called "myself". It should contain three properties: a "name", whose value would be your name (a string), an "age" which should be a number, and "likes", which should be an array of strings of things you like.
2. Write a function "findMax" that takes as input one value, `arr` which is meant to be an array. It then proceeds to find the maximum value in the array, and return that maximum value:
 - Your function should skip over values that are not numbers. Use `typeof` to determine that.
 - If there are no numbers in the array, the result should be `-Infinity`. Ideally this should be a natural result of the algorithm you write, and not an exceptional behavior (i.e. if you write your algorithm well, this will just naturally happen).
 - You should not use any fancy functions. You can do this with a `for` loop and an accumulator.
3. 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.
4. 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.
5. 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

¹<https://github.com/skiadas/WebAppsCourse/blob/gh-pages/labs/lab0.html>

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).