BRAINSTER

# Workshop 23: JavaScript Intro

Brainster Web Development Academy

# Exercise 01

Define two variables. Initialize them with values taken as user input  from the prompt() function.

Write a JavaScript function that accepts those numbers as parameters and returns the bigger one. If they are equal it returns a message “Numbers are equal”.

Inside the HTML part of the solution, make a div with an id="result".

With JavaScript code, after you’ve found out which number is bigger create a new paragraph tag with text: “The numbers {x} and {y} are equal” or “The number {x} is bigger than number {y}”.

Then append this new paragraph to the div with id="result".

Add a button that when clicked will refresh the page and thus the program will start over.
Hint: add a button that will have onClick attribute where you will write the code to refresh the page (location.reload()).



# Exercise 02

Write an unordered list of items in html.

Define one variable in JavaScript. Initialize it with values taken as user input  from the prompt() function.

Define a function that accepts this variable as parameter.

The function should color red all the elements of the unordered list up to the given number. For example, if the user enters 5 in the prompt, the first 5 items (first 5 <li> tags) should be colored red.

If the user enters value other than a number in the prompt window show an alert box saying that it's not a valid value and that the program only accepts numbers.

.

If the entered number is bigger than the number of <li> elements, color all of them

Hint: Number.isInteger();

Hint: parseInt();



# Exercise 03

Make a program which for an entered amount of money in an ATM (prompt) will calculate how many banknotes should the ATM return to you. The banknotes can be: 1000, 500, 100, 50, 10, 5. The entered money amount can be only whole number ending in 5 or 10 (ex. 1335, 235, 45, 10, 2500).

- Get a number from prompt.
- Make an arrow function checkBanknotes which takes a number(money amount) as a parameter.
- Make another arrow function isValidAmountForATM which checks if a number is valid for ATM use. A valid number for ATM use in our case is a number which ends with 0 or 5.
- Use the isValidAmountForATM inside checkBanknotes to check if the provided amount is valid for ATM use. If it is, proceed in the next step, and if it isn't show an alert saying: 'You cannot withdraw money amount that isn’t divisible by 5'.
- If the amount is valid for withdraw, find out how many 1000, 500, 100, 50, 10, 5 banknotes should the ATM return to match the amount.
- The code should return the minimal amount of banknotes. For example, if the user asked for 1800, that would be 1 x 1000, 1 x 500 and 3 x 100. Valid solution would be for example 3 x 500 and 3 x 100, but that is not the solution with the least number of banknotes.
- In a new html elements inside body (dynamically) write: That would be: *number* in 1000s, *number* in 500s, *number* in 100s, *number* in 50s, *number* in 10s and *number* in 5s.
- Call the checkBanknotes with the number from the prompt.



# Exercise 04

Define an array containing negative and positive numbers.

Write code that will extract all negative numbers  from this array into a new one. The original array should be left with only positive numbers.

Iterate through the new array with negative numbers and make all of them positive.

Calculate the sums of both arrays.

Print the 2 arrays in console, print their sums and print information which has larger sum of numbers, like: “Sum of the numbers in the array1 is greeted than the sum of numbers in array2”.

* Try using some functions like map/filter/find.



# Break a leg!



If there is not enough time, try to finish the exercises at home.