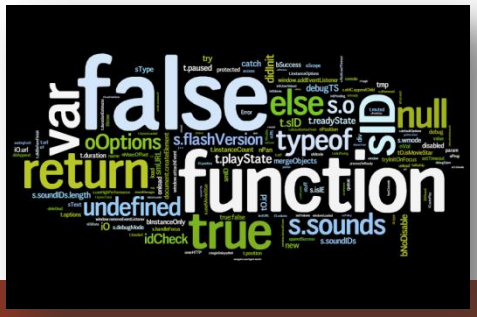
JavaScript



OVERVIEW

In this assignment we will write a handful of JavaScript functions. These functions are mostly about familiarizing you with the basic elements of JavaScript. You can use any plain-text editor you would like to create a file named **homework1.js**. You must write a function that solves each of the problems listed below. Each of the functions must be placed within the file **homework1.js** which is the only deliverable for this assignment.

FOLLOW THE LEADER

lettersThatFollow(text, ch)

This function accepts two strings: a string named text and a string name ch. The function must search through the text for all occurrences of ch in text and, for each occurrence of ch, the character that immediately follows ch in text if such a character exists. The result must be a string that contains no duplicate characters.

Examples

- **lettersThatFollow**("I nearly laughed when the fat man sat on his hat.", "a") => "rutn"
- **lettersThatFollow**("I nearly laughed when the fat man laughed.", "z") => ""
- **lettersThatFollow**("I nearly laughed when the fat man laughed.", ".") => ""

YOU ARE UNIQUE; JUST LIKE EVERYONE ELSE

containsDuplicate(text)

This function takes a string (text) as input and returns true if and only if the text string contains at least one character that occurs twice in the text.

Examples

- **containsDuplicate**("I nearly laughed when the fat man sat on his hat.") => true
- **containsDuplicate** ("3825") => false
- **containsDuplicate** ("ABC") => false

THE ALL-CAPS REPLACEMENT

highlight(text, phrase)

This function takes two strings: some text and a phrase that may or may not occur in the text. The function must return a string that is a duplicate of the input text except that every occurrence of phrase in text is replaced by the phrase surrounded in a span tag. The code must be case insensitive.

Examples

- **highlight** (“Faith, Hope and Love.”, “hoPe”) => “Faith,Hope and Love.”
- **highlight** (“Jimmy”,”m”) => “Jimmy”
- **highlight** (“ABC”, “D”) => “ABC”

OVER AND OVER AGAIN

repeat(text, n)

This function accepts a string (text) and an integer number (n). The function must return a string that has n repetitions of text. If n is non-positive this function must return the empty string.

Examples

- **repeat**(“cow “, 3) => “cowcowcow”
- **repeat**(“alf”,10)=> “alfalfalfalfalfalfalfalfalf”
- **repeat**(“cat”,0) =””
- **repeat**(“repeat”, -3) =””

COUNT ME IN

count(text, phrase)

This function accepts a string (text) and a string (phrase) that may or may not occur in the text. The function must return the number of times that the phrase occurs in the text.

Examples

- **count**(“alfalfalfalfalfalfalfalfalf “, “alf”) => 10
- **count**(“”,”cat”)=> 0
- **count**(“cccc”,”cc”) = 3

HOME ON THE RANGE

range(low, high, step)

This function accepts three integer numbers: low, high and step. The function returns an array that contains the integers in the closed interval low to high in increments of step. If step is non-positive, the function must return the empty array. If low is greater than high, the function must return the empty array.

Examples

- **range**(1, 3, 1) => [1,2,3]
- **range**(3, 1, 0)=> []
- **range**(4, 4, 10) =>[4]
- **range**(-10, 0, 3) => [-10, -7, -4, -1]