JAVASCRIPT QUESTIONS

Deadline: 19/08/2020 (11:59 pm)

(Very Easy)

Q1. Convert Minutes into Seconds

Write a function that takes an integer minutes and converts it to seconds.

Examples

convert (5) ➡ 300
convert (3) ➡ 180
convert (2) ➡ 120

Notes

- Don't forget to return the result.
- If you get stuck on a challenge, find help in the Resources tab.
- If you're really stuck, unlock solutions in the Solutions tab.

Q2. Divides Evenly

Given two integers, a and b, return true if a can be divided evenly by b. Return false otherwise.

Examples

dividesEvenly (98, 7) ➡ true

# 98/7 = 14

dividesEvenly (85, 4) ➡ false

# 85/4 = 21.25

Notes

- a will always be greater than or equal to b.

(Easy)

Q1. Count Instances of a Character in a String

Create a function that takes two strings as arguments and returns the number of times the first string (the single character) is found in the second string.

Examples

```
charCount ("a", "edabit") ➞ 1
charCount ("c", "Chamber of secrets") ➞ 1
charCount ("b", "big fat bubble") ➞ 4
```

Notes

- Your output must be case-sensitive (see second example).

Q2. Add up the Numbers from a Single Number

Create a function that takes a number as an argument. Add up all the numbers from 1 to the number you passed to the function. For example, if the input is 4 then your function should return 10 because 1 + 2 + 3 + 4 = 10.

Examples

```
addUp (4) ➞ 10
addUp (13) ➞ 91
addUp (600) ➞ 180300
```

Notes

- Expect any positive number between 1 and 1000.

Q3. Replace Vowel with Another Character

Create a function that takes a string and replaces the vowels with another character.

a = 1, e = 2, i = 3, o = 4, u = 5

Examples

```
replaceVowel("karachi") ➞ "k1r1ch3"
replaceVowel("chembur") ➞ "ch2mb5r"
replaceVowel("khandbari") ➞ "kh1ndb1ri"
```

Notes

- The input will always be in lowercase.

(Medium)

Q1. Reverse Words Starting with a Particular Letter

Write a function that reverses all the words in a sentence that start with a particular letter.

Examples

specialReverse ("word searches are super fun", "s")

➡ "word sehcraes are repus fun"

specialReverse ("first man to walk on the moon", "m")

➡ "first nam to walk on the noom"

specialReverse ("peter piper picked pickled peppers", "p")

➡ "retep repip dekcip delkcip sreppep"

Notes

- Reverse the words themselves, not the entire sentence.
- All characters in the sentence will be in lower case.

Q2. Hitting the Jackpot

Create a function that takes in an array (slot machine outcome) and returns true if all elements in the array are identical, and false otherwise. The array will contain 4 elements.

Examples

testJackpot (["@", "@", "@", "@"]) ➡ true
testJackpot (["abc", "abc", "abc", "abc"]) ➡ true
testJackpot (["SS", "SS", "SS", "SS"]) ➡ true
testJackpot (["&&", "&", "&&&", "&&&&"]) ➡ false
testJackpot (["SS", "SS", "SS", "Ss"]) ➡ false

Notes

- The elements must be exactly identical for there to be a jackpot.

Q3. Remove Duplicates from an Array

Create a function that takes an array of items, removes all duplicate items and returns a new array in the same sequential order as the old array (minus duplicates).

Examples

removeDups ([1, 0, 1, 0]) ➞ [1, 0]
removeDups (["The", "big", "cat"]) ➞ ["The", "big", "cat"]
removeDups (["John", "Taylor", "John"]) ➞ ["John", "Taylor"]

Notes

- Tests contain arrays with both strings and numbers.
- Tests are case sensitive.

(Hard)

Q1. Get Real Type

Create a function that takes a value as an argument and returns the type of this value. You should get the real type of a value (JavaScript typeof doesn't return the real object type of values and you need to fix that).

Examples

realType (1) ➡ "number"
realType("a") ➡ "string"
realType(true) ➡ "boolean"
realType ([]) ➡ "array"
realType(null) ➡ "null"

Notes

- N/A

Q2. Numbers in Strings

Create a function that takes an array of strings and returns an array with only the strings that have numbers in them. If there are no strings containing numbers, return an empty array.

Examples

numInStr (["1a", "a", "2b", "b"]) ➡ ["1a", "2b"]
numInStr (["abc", "abc10"]) ➡ ["abc10"]
numInStr (["abc", "ab10c", "a10bc", "bcd"]) ➡ ["ab10c", "a10bc"]
numInStr (["this is a test", "test1"]) ➡ ["test1"]

Notes

- The strings can contain white spaces or any type of characters.
- Bonus: Try solving this without regex.