STRING

1. Написать функцию, к-я принимает первый аргумент - строку, второй вргумент - кол-во повторений. Возвращает строку в к-й содержится исходная строка заданное кол-во раз.

function multiplyString(str, num) {

var resultString = "";

for(i=0; i < num; i++) {

resultString += str;

}

return resultString;

}

2. Напишите функцию ucFirst(str), которая возвращает строку str с заглавным первым символом, например:

ucFirst("вася") == "Вася";

ucFirst("") == ""; // нет ошибок при пустой строке

function ucFirst(str) {

if (str == "") {

return "";

} else {

var string = str.toString();

string = string[0].toUpperCase() + string.slice(1);

return string;

}

}

3. Создайте функцию truncate(str, maxlength), которая проверяет длину строки str, и если она превосходит maxlength – заменяет конец str на "...", так чтобы ее длина стала равна maxlength.

function truncate(str, maxlength) {

str = str.toString();

len = str.length;

if (len <= maxlength) {

return str;

} else {

return str.slice(0, len - (len - maxlength) - 3) + "...";

}

}

4. Написать функцию turnMeBaby, к-я принимает один аргумент - исходную строку и возвращает строку перевернутую задом на перед.

function turnMeBaby(str) {

var str = str.toString();

var result = "";

for(var i = str.length; i > 0; i--) {

result = result + str[i - 1];

}

return result;

}

5. Реализовать функцию, которая принимает на вход строку и возвращает число слов в этой строке. Слова разделены пробелами (и, возможно, не одним), а знаки пунктуации не учитываются. Допускается только посимвольный перебор строки.

function splitString(str) {

var str = str.toString();

var wordArray = [];

var oneWord = "";

for(var i = 0; i <= str.length - 1; i++) {

if(str[i] != " ") {

oneWord += str[i];

} else {

if(str[i+1] != " ") {

wordArray.push(oneWord);

oneWord = "";}

}

}

return wordArray.length + 1;

}

NUMBERS

1. Написать функцию, к-я принимает первый аргумент - число, второй вргумент - степень.

Возвращает число возведенное в заданную степень.

function powFunction(num1, num2) {

return Math.pow(num1, num2);

}

На випадок якщо не дозволяється використовувати метод pow(), то :

function powFunction2(num1, num2) {

num1 = parseInt(num1);

num2 = parseInt(num2);

var result = 1;

var sum = 1;

if(num2 < -1) {

for(var i = 0; i > num2; i--) {

sum \*= num1;

result = 1 / sum;

}

} else if(num2 == -1) {

result = 1 / num1;

} else if(num2 == 0) {

result = 1;

} else if(num2 == 1) {

result = num1;

} else if(num2 > 1) {

for(var i = 0; i < num2; i++) {

result \*= num1;

}

}

return result;

}

2. Написать функцию, к-я выводит в консоль/html-документ таблицу умножения.

1 \* 1 = 1;

1 \* 2 = 2;

1 \* 3 = 3;

...

==========

2 \* 1 = 2;

2 \* 2 = 4;

2 \* 3 = 6;

...

==========

3 \* 1 = 3;

3 \* 2 = 6;

3 \* 3 = 9;

...

==========

...

function pifagoreTable() {

var multiple = 0;

for(var i = 1; i <= 10; i++) {

for(var j = 1; j <= 10; j++) {

multiple = i \* j;

console.log( i + " \* " + j + " = " + multiple);

if(j==10) {

console.log("================");

}

}

}

}

Є питання: чому в кінці таблиці ще додатково вибиває undefined?

3. Модифицировать функцию вывода таблицы умножения таким образом, чтобы она принимала на вход два параметра: 1-й число, второй - кол-во строк таблицы умножения.

multiTable(2, 3);

2 \* 1 = 2;

2 \* 2 = 4;

2 \* 3 = 6;

function multipleTable(num1, num2) {

var multiple = 0;

for(var i = 1; i <= num2; i++) {

multiple = num1 \* i;

console.log(num1 + " \* " + i + " = " + multiple);

}

}

4. Напишите функцию doubleNum, которая возвращает переданное ей число увеличенное вдвое,

если число не состоит из "двойников" - в таком случае функция просто возвращает исходное число.

"Двойники" - числа состоящие из двух одинаковых чисел, например 77, 22, 200200, 4343.

doubleNum(2); // 4

doubleNum(6); // 12

doubleNum(66); // 66

doubleNum(100100); // 100100

doubleNum(111111); // 111111

function doubleNum(num) {

var str = num.toString(),

len = str.length,

slice1 = str.slice(0, len/2),

slice2 = str.slice(len/2);

if(slice1 == slice2) {

return num;

} else {

return num \* 2;

}

}

Зауваження: дана функція не працює з числами, які починаються на 0 (наприклад 010010), я не розумію чого . Цікавий момент - з таких чисел typeOf() викидає помилку (ReferenceError: typeOf is not defined), в свою чергу метод isNaN() видає false, отже виходить, що такі числа, таки є числами.