# Solutions-Lab: Strings and Regular Expressions

## Pascal or Camel Case

function solve() {

let input = document.getElementById('str1').value;

let currentCase = document.getElementById('str2').value;

doPasscalOrCamelCase(input, currentCase);

function doPasscalOrCamelCase(input, currentCase){

let splitedInput = input.toLowerCase().split(' ').filter(w => w !== '');

let output = '';

if (currentCase === 'Pascal Case'){

for(let word of splitedInput){

if (word[0] !== word[0].toUpperCase()){

word = word.replace(word[0], word[0].toUpperCase());

}

output += word;

}

} else if (currentCase === 'Camel Case'){

for(let word of splitedInput){

if (word[0] !== word[0].toUpperCase()){

word = word.replace(word[0], word[0].toUpperCase());

}

output += word;

}

output = output.replace(output[0], output[0].toLowerCase())

} else {

output = 'Error!'

}

console.log(output)

document.getElementById('result').innerHTML = output;

}

}

## 2.Find ASCII Equivalent

function solve() {

let input = document.getElementById('str').value;

convertFromAndToAscii(input);

function convertFromAndToAscii(input){

let resultElement = document.getElementById('result');

let splitedInput = input.split(' ').filter(w => w !== '');

let output = '';

for(let element of splitedInput){

if (Number(element)){

output += String.fromCharCode(element);

} else {

let numbersFromCharacters = [];

for (let i = 0; i < element.length; i++) {

numbersFromCharacters.push(element[i].charCodeAt(0));

}

let p = document.createElement('p');

p.textContent = numbersFromCharacters.join(' ');

resultElement.appendChild(p);

}

}

let p = document.createElement('p');

p.textContent = output;

resultElement.appendChild(p);

}

}

## 3.Split String Equally

function solve() {

let string = document.getElementById('str').value;

let number = Number(document.getElementById('num').value);

splitStringEqually(string, number);

function splitStringEqually(string, number){

let outputArray = [];

if (string.length % number !== 0){

let stringLength = string.length;

let symbolsCount = 0;

while(stringLength % number !== 0){

stringLength %= number;

stringLength++;

symbolsCount++;

}

for (let i = 0; i < symbolsCount; i++) {

string +=string[i];

}

}

for (let i = 0; i < string.length; i += number) {

outputArray.push(string.substr(i, number));

}

document.getElementById('result').innerHTML = outputArray.join(' ');

}

}

## 4.Replace a Certain Word

function solve() {

let inputArray = JSON.parse(document.getElementById('arr').value);

let wordToReplacing = document.getElementById('str').value;

replaceWord(inputArray, wordToReplacing);

function replaceWord(arr, replacement){

let resultElement = document.getElementById('result');

let wordToReplaced = arr[0].split(' ').filter(w => w !== '')[2];

let regex = new RegExp(wordToReplaced, 'gi');

for (let element of arr){

element = element.replace(regex, replacement);

let p = document.createElement('p');

p.textContent = element;

resultElement.appendChild(p);

}

}

}

## 5.Extract User Data

function solve() {

let inputArr = JSON.parse(document.getElementById('arr').value);

extractUserData(inputArr);

function extractUserData(arr){

let resultElement = document.getElementById('result');

let pattern = /^([A-Z][a-z]\* [A-Z][a-z]\*) (\+359 \d{1} \d{3} \d{3}|\+359-\d{1}-\d{3}-\d{3}) ([a-z0-9]+@[a-z]+\.[a-z]{2,3})$/;

let match;

for(let data of arr){

match = pattern.exec(data);

if(match){

let p1 = document.createElement('p');

p1.textContent = `Name: ${match[1]}`;

resultElement.appendChild(p1);

let p2 = document.createElement('p');

p2.textContent = `Phone Number: ${match[2]}`;

resultElement.appendChild(p2);

let p3 = document.createElement('p');

p3.textContent = `Email: ${match[3]}`;

resultElement.appendChild(p3);

} else {

let pError = document.createElement('p');

pError.textContent = 'Invalid data';

resultElement.appendChild(pError);

}

let pDashes = document.createElement('p');

pDashes.textContent = '- - -';

resultElement.appendChild(pDashes);

}

}

}