## **03. Sub-string in text**

## **Description**

Write a JavaScript function that finds how many times a substring is contained in a given text (perform case insensitive search).

#### Input

[

'in',

'We are living in an yellow submarine. We don\'t have anything else. inside the submarine is very tight. So we are drinking all the day. We will move out of it in 5 days.'

]

#### Output

9

**Source code**

function *solve*(*args*) {  
 var find = (*args*[0] + **''**).toLowerCase(),  
 parse = (*args*[1] + **''**).toLowerCase(),  
 counter = 0;  
  
 var index = parse.indexOf(find);  
 while (index >= 0) {  
 counter += 1;  
 index = parse.indexOf(find, index + 1);  
 }  
  
 **console**.log(counter);  
}

## **04. Parse tags**

## **Description**

You are given a text. Write a function that changes the text in all regions:

<upcase>text</upcase> to uppercase.

<lowcase>text</lowcase> to lowercase

<orgcase>text</orgcase> does not change casing

#### Input

[ 'We are <orgcase>liViNg</orgcase> in a <upcase>yellow submarine</upcase>. We <orgcase>doN\'t</orgcase> have <lowcase>anything</lowcase> else.' ]

#### Output

We are liViNg in a YELLOW SUBMARINE. We doN't have anything else.

**Source code**

function *solve*(*args*) {  
 const upOpen = **'upcase'**,  
 upClose = **'/upcase'**,  
 lowOpen = **'lowcase'**,  
 lowClose = **'/lowcase'**,  
 orgOpen = **'orgcase'**,  
 orgClose = **'/orgcase'**,  
 upScope = **'up'**,  
 lowScope = **'low'**,  
 orgScope = **'initial'**;  
  
 let isTag = false,  
 scopeStack = [],  
 currentScope = orgScope,  
 strLength = *args*[0].**length**,  
 currentTag = **''**,  
 output = **''**;  
  
 for (let i = 0; i < strLength; i += 1) {  
 let c = *args*[0][i] + **''**;  
  
 if (c === **'<'**) {  
 isTag = true;  
 } else if (c === **'>'**) {  
 isTag = false;  
 // Evaluate Tag  
 if (currentTag === upOpen) {  
 scopeStack.push(currentScope);  
 currentScope = upScope;  
 } else if (currentTag === lowOpen) {  
 scopeStack.push(currentScope);  
 currentScope = lowScope;  
 } else if (currentTag === orgOpen) {  
 scopeStack.push(currentScope);  
 currentScope = orgScope;  
 } else if (currentTag === upClose) {  
 currentScope = scopeStack.pop();  
 } else if (currentTag === lowClose) {  
 currentScope = scopeStack.pop();  
 } else if (currentTag === orgClose) {  
 currentScope = scopeStack.pop();  
 } else {  
 output += currentTag;  
 }  
 currentTag = **''**;  
 } else if (isTag) {  
 currentTag += c;  
 } else if (!isTag) {  
 if (currentScope === upScope) {  
 output += c.toUpperCase();  
 } else if (currentScope === lowScope) {  
 output += c.toLowerCase();  
 } else {  
 output += c;  
 }  
 }  
 }  
 **console**.log(output);  
}

## **05. nbsp**

## **Description**

Write a function that replaces non breaking white-spaces in a text with &nbsp.

#### Input

[ 'hello world' ]

#### Output

hello&nbsp;world

**Source code**

function *solve*(*args*) {  
 var output = (*args* + **''**).replace(/ /g, **'&nbsp;'**);  
 **console**.log(output);  
}

## **06. Extract text from HTML**

**Input**

[

'<html>',

' <head>',

' <title>Sample site</title>',

' </head>',

' <body>',

' <div>text',

' <div>more text</div>',

' and more...',

' </div>',

' in body',

' </body>',

'</html>'

]

**Output**

Sample sitetextmore textand more...in body

**Source code**

function *solve*(*args*) {  
 let output = **''**;  
 let matchTags = /<.\*?>/ig;  
  
 for (let line of *args*) {  
 output += line.replace(matchTags, **''**).trim();  
 }  
  
 **console**.log(output);  
}

## **07. Parse URL**

### Input

[ 'http://telerikacademy.com/Courses/Courses/Details/239' ]

#### Output

protocol: http

server: telerikacademy.com

resource: /Courses/Courses/Details/239

#### Source code

function *solve*(*args*) {  
 var protocolSeparator = **'://'**;  
 var serverSeparator = **'/'**;  
 var parse = *args*[0] + **''**;  
 var index = 0;  
 var len = protocolSeparator.**length**;  
  
 var protocol = **''**;  
 var server = **''**;  
 var resource = **''**;  
  
  
 index = parse.indexOf(protocolSeparator);  
 protocol = parse.substr(0, index);  
  
 len = index + len;  
 index = parse.indexOf(serverSeparator, len);  
  
 server = parse.substr(len, index - len);  
 resource = parse.substr(index);  
  
 **console**.log(**'protocol: '** + protocol);  
 **console**.log(**'server: '** + server);  
 console.log(**'resource: '** + resource);  
}

## **08. Replace tags**

## Description

Write a JavaScript function that replaces in a HTML document given as string all the tags <a href="…">…</a> with corresponding tags [TEXT](URL).

#### Input

[ '<p>Please visit <a href="http://academy.telerik.com">our site</a> to choose a training course. Also visit <a href="www.devbg.org">our forum</a> to discuss the courses.</p>' ]

#### Output

<p>Please visit [our site](http://academy.telerik.com) to choose a training course. Also visit [our forum](www.devbg.org) to discuss the courses.</p>

#### Source code

function *solve*(*args*) {  
 var regex = new RegExp(**'<a href="(.\*?)">(.\*?)</a>'**, **"g"**);  
 var text = *args*[0].replace(regex, function(*none*, *href*, *content*) {  
 return **'['** + *content* + **']('** + *href* + **')'**;  
 });  
  
 **console**.log(text);  
}

## **1. Format with placeholders**

## **Description**

* Write a function that formats a string. The function should have placeholders, as shown in the example
  + Add the function to the **String.prototype**

| **Input** | **Output** |
| --- | --- |
| [ '{ "name": "John" }', "Hello, there! Are you #{name}?" ] | 'Hello, there! Are you John' |
| [ '{ "name": "John", "age": 13 }', "My name is #{name} and I am #{age}-years-old" ] | 'My name is John and I am 13-years-old' |

#### Source code

function *solve*(*args*) {  
 const options = **JSON**.parse(*args*[0]);  
  
 let inputString = *args*[1] + **''**,  
 matchProps = /#\{(.\*?) \}/g;  
  
 for (let item in options) {  
 inputString = inputString  
 .replace(new RegExp(**`#{**${item}**}`**, **'g'**), options[item]);  
  
 }  
  
 **console**.log(inputString);  
}

## **2. HTML Binding**

## **Description**

* Write a function that puts the value of an object into the content/attributes of HTML tags.
  + Add the function to the String.prototype and use it in the following way:

var html = '<div data-bind-content="name"></div>';

var data = { name: 'Pesho' };

var result = html.bind(data);

#### Input 1

[

'{ "name": "Steven" }',

'<div data-bind-content="name"></div>'

]

#### Output 1

<div data-bind-content="name">Steven</div>

#### Input 1

[

'{ "name": "Elena", "link": "http://telerikacademy.com" }',

'<a data-bind-content="name" data-bind-href="link" data-bind-class="name"></а>'

]

#### Output 1

<a data-bind-content="name" data-bind-href="link" data-bind-class="name" href="http://telerikacademy.com" class="Elena">Elena</а>

#### Source code

function *solve*(*args*) {  
 var obj = **JSON**.parse(*args*[0]);  
 var text = **''** + *args*[1];  
 text = text.replace(/'/, **'"'**);  
 var regex = /data-bind-(.\*?)="(.\*?)"/gmi;  
 var currentMatch;  
 while (currentMatch = regex.exec(text)) {  
 var index = text.indexOf(**'>'**);  
 if (text[index - 1] === **'/'**) { // if tag is self closing  
 index--;  
 }  
 var field = currentMatch[1];  
 if (field.toLowerCase() === **'content'**) {  
 var arr = text.split(**''**);  
 var x = arr.splice(index + 1, 0, obj[currentMatch[2]]);  
 text = arr.join(**''**);  
 }  
 else {  
 var arr2 = text.split(**''**);  
 var x2 = arr2.splice(index, 0, **" "** + field + **'="'** + obj[currentMatch[2]] + **'"'**);  
 text = arr2.join(**''**);  
 }  
 }  
  
 **console**.log(text);  
}