

## Enunciado

### Enunciado

Más abajo vas a encontrar un fragmento de un programa Javascript. El objetivo de la función *FormatString* es quitar todos los caracteres especiales de la cadena de caracteres que viene como parámetro. Los únicos caracteres permitidos son las 26 letras del alfabeto inglés (de la A-Z y de la a-z), dígitos (0-9), espacios ( ), guiones medios (-) y guiones bajos (\_). El resto se consideran caracteres especiales. La función debe retornar la cadena "ERROR" si la cadena de caracteres de entrada no tiene caracteres permitidos.

Veamos unos ejemplos de valores de entrada y salida esperados:

**Entrada:** Lavese las manos, por favor!

**Salida:** Lavese las manos por favor

**Entrada:** Consultar al #0800-999-100.

**Salida:** Consultar al 0800-999-100

Tener en cuenta que en este programa los caracteres están codificados utilizando el estándar ASCII. Cada caracter de una secuencia ASCII es un número entre 0 y 127.

Podemos dividir a los caracteres en varios grupos:

- **Caracteres de control (0–31 y 127):** caracteres no imprimibles, utilizados para dar comandos. La mayoría cayeron en desuso.
- **Caracteres Especiales Primer Grupo (32-47):** "espacio en blanco", "!", "acento", "#", "\$", "%", "&", "apóstrofe", "(", ")", "\*", "+", ",", "-", ".", "/" en este mismo orden.
- **Números (48-57):** Números Árabigos 0-9.
- **Caracteres Especiales Segundo Grupo (91-96):** "[", "\", "A", "\_", "" en este mismo orden.
- **Otros Caracteres especiales: (58-64 / 123-126) :** otros caracteres imprimibles que no son ni números ni letras.
- **Letras Mayúsculas (65-90):** A-Z.
- **Letras Minúsculas: (97-122):** a-z



Programa desarrollado para cumplir con la funcionalidad anterior



```
1 function FormatString(sentence) {  
2   let result = [];  
3  
4   sentence = sentence.toUpperCase();  
5  
6   let i = 0;  
7   let j = 0;  
8   while (i < sentence.length) {  
9     if (  
10      (sentence.charCodeAt(i) >= 65 && sentence.charCodeAt(i) <= 90) ||  
11      (sentence.charCodeAt(i) >= 48 && sentence.charCodeAt(j) <= 57) ||  
12      sentence.charCodeAt(i) == 32 ||  
13      sentence.charCodeAt(i) == 45  
14    ) {  
15      sentence[j] = result[i];  
16      j += 1;  
17    }  
18    i += 1;  
19  }  
20  
21  return result.join('');  
22 }
```



### Aclaraciones adicionales

- La función `charCodeAt` permite recuperar el código ASCII del carácter en la posición indicada. En este ejercicio, vamos a pedir realizar las comparaciones utilizando esta función.
- La función `toUpperCase` devuelve un string donde todas las letras minúsculas pasan a ser mayúsculas.
- `length` devuelve el largo de la cadena.
- Los corchetes se utilizan para acceder a la posición *i*-ésima de un Array. `a[i] = b[j]` quiere decir: "asignar en la posición *i* del array **a** el elemento presente en la posición *j* del array **b**".
- La sintaxis de Javascript es muy sencilla y podés utilizar este sitio como referencia: <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>. Si a pesar de eso tenés dificultades, podés utilizar pseudocódigo o el lenguaje que te resulte más conveniente.
- Agrega los comentarios que consideres necesarios.

El programa presentado no cumple la funcionalidad esperada. ¿Podés encontrar los errores que tiene? Tu objetivo es explicar con tus palabras los errores presentes y reescribir el código completo sin errores. \*

Tu respuesta

Se enviará un correo electrónico con una copia de tus respuestas a la dirección que suministraste.

Enviar

Borrar formulario



Nunca envíes contraseñas a través de Formularios de Google.

reCAPTCHA