

String. Ejercicios resueltos

1. Crea una función que reciba como argumento una cadena de caracteres y devuelva una cadena con todas las vocales y el número de veces que aparece cada una en la cadena argumento.

Procura usar prioritariamente los métodos del objeto string

```
function cuentaVocales(frase){
  const vocales="aeiou";
  let veces =[0,0,0,0,0];
  let res="", car, pos;
  frase = frase.toLowerCase();
  for(let i=0; i < frase.length; i++){
    car = frase.charAt(i);
    pos = vocales.indexOf(car);
    if (pos >=0){
      veces[pos]++
    }
  }
  for(let j=0; j < vocales.length; j++){
    res += vocales.charAt(j)+":"+veces[j]+"/";
  }
  return res ;
}
```

2. En esta ocasión se trata de crear una función que reciba como argumento una frase con letras y números. La función debe devolver la suma de todos los dígitos contenidos en la frase

```
function sumaDigs(frase){  
  
  let digito="", suma=0;  
  
  for(let i = 0; i < frase.length; i++){  
    digito = parseInt(frase.charAt(i));  
    if (!isNaN(digito)){  
      suma += digito;  
    }  
  }  
  
  return suma  
}
```

3. Crea una función javascript que reciba como argumento una frase y que devuelva el número de espacios, dígitos y letras contenidos en la frase. Puedes usar expresiones regulares

```
function contarCars(frase){  
  let esp,dig, let, lista;  
  lista = frase.match(/\s/g);  
  esp = lista.length;  
  lista = frase.match(/[0-9]/g);  
  dig = lista.length;  
  lista = frase.match(/[a-z]/gi);  
  let = lista.length;
```

```
console.log("Hay "+esp+" espacios");
console.log("Hay "+dig+" digitos");
console.log("Hay "+let+" letras");
}
```

4. Escribe una función que reciba como argumento una frase y devuelva la misma frase pero con la inicial de cada palabra en mayúsculas

```
function cambiarIni(frase){

let lista=frase.split(" ");

let ini, pal, cambiada;

cambiada = lista.map(function(v,i,l){
ini = v.charAt(0).toUpperCase();
pal = ini.concat(v.substring(1));
return pal

})

return cambiada.join(" ");
}
```

5. Se trata de crear una función que decodifique un código de producto. El código contiene tres partes separadas por un guion:

Dos caracteres: CP: cliente particular, CE: empresa

Dos dígitos: 10 Local, 11 Autonómico, 12 Nacional, 20 Internacional

Una cifra que indica el número de años de antigüedad del cliente.

Si el código es erróneo la función deberá informar del tipo de error o errores.

Por ejemplo, descifrar("CP-12-3") deberá devolver Cliente particular nacional con 3 años de antigüedad

```
function descifrar(codigo){
  let tipo, local, ant;
  let msg = "";
  switch (codigo.substring(0,2)){
    case "CP":
      tipo = "Particular";
      break;
    case "CE":
      tipo = "Empresa";
      break;
    default:
      msg += " Error de cliente";
  }
  switch (codigo.substring(3,5)){
    case "00":
      local = "Local";
      break;
    case "01":
      local = "Nacional";
      break;
```

```

case "11":
    local = "Internacional";
    break;
default:
    msg += " Error en origen";
}
ant = codigo.substring(6);
if (msg.length == 0){

    msg = "cliente "+tipo+" "+local+" con "+ant+" de antigüedad" ;
}
return msg;
}

```

6. Diseña una función que sea capaz de encontrar los caracteres comunes entre dos palabras. La función recibe como argumentos dos palabras o frases y devuelve una cadena con los caracteres que haya en común entre ambas ordenados de menor a mayor. Se desechan los espacios en blanco. No se tiene en cuenta el caso (mayúsculas o minúsculas)

```

function comunes(pal1, pal2){
    let comunes = "", car;
    pal1 = pal1.toLowerCase();
    pal2 = pal2.toLowerCase();
    for(let i = 0; i < pal1.length; i++){

```

```

car = pal1.charAt(i);
if(pal2.includes(car) && !comunes.includes(car)){
    comunes += car;
}
}

comunes = comunes.split("")
comunes.sort();
comunes = comunes.join("");
return comunes;
}

```

7. En este caso se trata de escriir una función javascript que acepte como argumento una cadena de caracteres y reúna los espacios repetidos en un solo espacio. Una especie de compresora de espacios. Resolverlo sin usar expresioens regulares ni arrays, solo el objeto string.

```

function unirEspacios(frase){
    let cambiada="", i = 0;
    frase = frase.trim();
    //cada vez que encuentra un espacio
    //extrae la subcadena hasta ese punto y
    //continua buscando desde el siguiente caracter
    slatando los espacios repetidos
    while ( i >= 0){
        frase = frase.substring(i);
        i = frase.indexOf(" ");
        switch (true){

```

```
case i>0:
    cambiada += frase.substring(0, i+1);
    i++;
    break;
case i<0 :
    cambiada += frase;
    break
case i==0:
    i++;
}
}
return cambiada
}
```