Ej 1 -Crear las expresiones regulares necesarias para resolver los siguientes puntos:

Valida si una cadena introducida es un número entero.

/^[0-9]+$/

Validar DNI, 8 números y una letra al final.

/^[0-9]{8}[a-z]$/gi

Validar una matrícula de un coche con formato 0000XXX

/^[0-9]{4}[a-z]{3}$/gi

Valida nombre de usuario en twitter, debe de empezar por @ y puede contener, números,letras mayúsculas y minúsculas, “-“y “_”.

/(^@)([a-z 0-9 \- _])+$/gi

Crear una expresión regular que valide una fecha en formato "XX/XX/XXXX", donde cada"X" es un dígito. Se puede probar con expresiones tipo:"El día 29/11/2019 tenemos examen.".

/(\d{2}[/]\d{2}[/]\d{4}$)/

Crear una expresión regular para la validación de direcciones de correo electrónico. Para simplificar, los valores anteriores a @ pueden contener cualquier carácter alfanumérico, y los caracteres “.”y “-”, mientras que los valores después de la @ pueden contener caracteres alfanuméricos, y el tipo de dominio tendrá una longitud de 2 o 3 caracteres.

/([a-z \- _ 0-9])@([\\.])([a-z]{2,3}$)/i

Crear una expresión regular que elimine las etiquetas potencialmente peligrosas (<script>...</script>) y todo su contenido de una cadena HTML.

/((<script>).*(<\/script>))/

Crea una expresión regular que dado un número de cuenta IBAN en formato ESXXXXXXXXXXXXXXXXXXXXX nos lo devuelva en porciones de 4 dígitos separado por un “-“.

Comprobar Formato:
/^(ES)(\d{22}$)/

Separando con guiones (javascript):

```
function separarIban(iban) {

  let resultado = "";
  let contador = 0;

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

    resultado += iban.charAt(i);
    contador++;

    if (contador == 4 && i < iban.length - 1) {
      resultado += '-';
      contador = 0;
    }
  }
 return resultado;
}
```

Ej2 –Genera dos ejercicios donde te parezca interesante el uso de expresiones regulares