1. Crear una función en JS que dado un parámetro string, se extraiga el subconjunto de ese string de sus índices **pares**.*(utilizar los métodos substr o substring, ciclo for para iterar sobre el string)* Ej: someFunction(‘abcdefg’) -> ‘aceg’;
2. Crear una función en JS que dado un parámetro string que está separado por comas, devuelva otro string separado por puntos *(utilizar los métodos de split de un string y un join de un array)* someFunction(‘asd,fbdbfd,sdfdsfds’) -> ‘asd.fbdbfd.sdfdsfds’
3. Crear una función en JS que dado un parámetro string que determine si es palindrome devuelve **true** de lo contrario que devuelva **false** *(utilizar los métodos split de un string, métodos join y reverse de un array)* someFunction(‘oruro’) -> **true**
4. Crear una función en JS que dado un parámetro número que determine si ese número es mayor a un número aleatorio (determinado dentro de la función) que devuelva **el string ‘Mi número {numeroDelParametro} es mayor al número {numeroAleatorio}’** de lo contrario que devuelva **false** *(utilizar los métodos random, max y floor del constructor Math)* someFunction(9) -> (suponiendo que 9 salió mayor al número aleatorio) **‘Mi número 9 es mayor al número 5’**; someFunction(1) ->(suponiendo que salió menor) **false**
5. Crear una función en JS que dado un parámetro de tipo vector me devuelva otro nuevo vector determinando si el vector pasado como parámetro, contiene elementos de tipo booleano u objeto. *(utilizar el método filter de un array)* Ej: someFunction([1, true, false, {}]) -> [true, {}]
6. Crear una función en JS que dado un parámetro de tipo string, determine si los primeros 3 caracteres (subconjunto del string) son iguales a los últimos 3 caracteres, en ese caso devuelve **true**, de lo contrario devuelve **false**. *(utilizar el método charAt del constructor String o acceso al elemento de un string mediante corchetes)* Ej: someFunction(“alaHelloJSWorldala”) -> **true**
7. Crear una función de JS que dado un parámetro de tipo Array con elementos de tipo number, retorne un nuevo array SIN TENER ELEMENTOS DUPLICADOS. *(una manera sencilla de realizar esto es utilizar los métodos filter e indexOf del constructor Array. El método indexOf en un array devuelve el índice del vector de la primera ocurrencia del valor pasado como parametro: [1,5,4,3].indexOf(5) //1 primer índice | Pista: Retornar dentro del callback la igualación del índice del callback ‘segundo parametro’ del filter con el del mismo array ‘tercer parámetro’ invocando el método indexOf)* Ej: someFunction([1,3,1,4,3,2,5,6,7,4,3,6,9,8,6,7]) -> [1,3,4,2,5,6,7,9,8]
8. Crear una función de JS que dado un parámetro de tipo string, retorna un nuevo string de los **valores distintos y separados por un espacio**, al **string de ejemplo que lo contiene(“BLA”)**. *(utilizar los métodos split y filter del constructor Array, el método join del constructor String. Otra manera de realizar es utilizando los métodos replace y trim de el constructor String)* Ej: someFunction(“BLABLAIBLAAMBLAABLAMASTERBLA”) -> “I AM A MASTER”.
9. Crear una función en JS que dado un parámetro de tipo number, retorne si el número pasado como parámetro es igual al número aleatorio que se calcula dentro de la función y si son iguales que devuelve el string “Adivinaste mi número”, caso contrario que devuelva “Sigue participando”. *(utilizar los métodos random y floor para sacar el numero aleatorio, LIMITAR el NÚMERO ALEATORIO AL NÚMERO PASADO COMO PARÁMETRO AL MOMENTO DE REALIZAR EL CÁLCULO DEL NÚMERO ALEATORIO)* Ej someFunction(9) -> “Adivinaste mi número” (en el caso que haya adivinado).
10. Crear una función en JS que dado un parámetro de tipo string, este string debe estar separado, en cada caracter o conjunto de caracteres con el caracter (‘|’), devuelva otro string que este separado por el caracter (‘-’) Ej: someFunction(‘hola|que|tal|e|r|t’) -> ‘hola-que-tal-e-r-t’ *(utilizar los métodos split de String y join de Array)*