

JS Basic

Bootcamp Frontend Developer

Ejercicio 1: Cálculo del DNI



En España el documento nacional de identidad se conoce como DNI. Cada DNI tiene un número de identificación único. Este número está compuesto por un número de 8 cifras, acompañado de una letra de seguridad. La letra asegura la integridad del DNI, ya que se calcula en base a los números.

Ejemplos de DNI: 12312312K, 78163312C, 12345678Z, 34667892S, 92234488A...

El cálculo de la letra se realiza de la siguiente forma:

Se toma el número y se divide entre 23. Según el resto de la división entre 23 miramos la letra correspondiente en la siguiente tabla:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

Realiza una función que reciba el número de un DNI y devuelva la letra correspondiente. Puedes realizar pruebas con los números de DNI anteriores.

Ejercicio 2: Cálculo del DNI V2



La función desarrollada en el ejercicio anterior cumple con su cometido.
Pero solo funciona correctamente en los casos óptimos.

¿Qué pasaría si le indicamos un número de 3 dígitos?

¿Qué pasaría si en vez de un número le enviamos un string?

Reutiliza la función del ejercicio anterior e introduce las siguientes mejoras:

- 1) Si introducimos un valor no numérico deberá mostrar el mensaje: "Debes introducir un valor numérico."
- 2) Si introducimos un valor numérico que no tenga 8 cifras deberá mostrar: "Debes introducir un número de 8 cifras."
- 3) Si introducimos un valor numérico negativo deberá mostrar: "Debes introducir un valor positivo."
- 4) Si el número del DNI tiene menos de 8 dígitos, debemos añadirle ceros por la izquierda hasta completarlo.



Ejercicio 3: Devuelve el string de mayor longitud

Realiza una función que reciba un array de strings y devuelva el string más largo y su longitud

Por ejemplo dado el siguiente array:

```
var array = ["Hola", "Frase corta", "Frase normalita", "Frase muy muy larga"];
```

Deberá devolver:

```
{  
  longitud: 19,  
  string: "Frase muy muy larga"  
}
```

Recuerda comprobar que todos los valores recibidos son de tipo string

Ejercicio 4: Media de longitudes



Vamos a complicar el ejercicio anterior:

Ahora cada vez que calculemos la longitud del string más largo, almacenaremos el resultado en un array de resultados.

Una vez ejecutados varios cálculos de longitud (4 en el ejemplo), vamos a calcular la media de los resultados.

```
var resultados = [];  
  
var arrayDeTest1 = ["Richie", "Joanie", "Greg", "Marcia", "Bobby"];  
var arrayDeTest2 = ["Blanka", "Zangief", "Chun Li", "Guile"];  
var arrayDeTest3 = ["Red", "Blue", "Green"];  
  
var calculoLongitudMasLargo(arrayDeTest1);  
var calculoLongitudMasLargo(arrayDeTest2);  
var calculoLongitudMasLargo(arrayDeTest3);  
  
media(resultados);  
  
// debe imprimir 6
```



Ejercicio 5: Cuenta las letras

Realiza una función que reciba un string y devuelva un objeto contando el número de apariciones de cada letra en el string. Almacena y devuelve el resultado en un objeto.

Asegúrate de que la función cumple su cometido haciendo uso de los tests aportados:

```
// AYUDA:
var arrayDeCaracteres = nombreDeString.split("");

// Tests

resultadoContador = contadorDeCaracteres("abbabcbdbabdbbabababcbcbab");
console.log( resultadoContador['a'] === 7 );
console.log( resultadoContador.b === 14 );
console.log( resultadoContador['c'] === 3 );
```



Ejercicio 6: Media de los valores del string

Escribe una función que reciba un string de números separados por dos puntos, cree un array a partir del string y devuelva la media de todos los valores

```
var stringDeNumeros = '80:70:90:100';  
// La función debe devolver 85
```

Una vez lo tengas resuelto, haz una función que haga lo mismo pero eliminando repetidos

```
// Misma funcionalidad pero eliminando los repetidos  
var stringDeNumeros = '80:70:90:100:100:100:100';  
// también deberá devolver 85
```



JS Basic

EJERCICIOS

Ejercicio 7: Diversas funciones con strings

```
// Esta función recibe un string y devuelve el string inverso
// Por ejemplo: para el string "Hola clase!" debe devolver "!esalc aloH"
function stringInverso(string) {
}

// Esta función debe recibir un string y devolver el mismo string sin espacios
function eliminarEspacios(string) {
}

// Esta función debe recibir un string y devolverlo con todas sus letras mayúsculas
function ponerTodasLasLetrasMayusculas(string) {
}
```




JS Basic

EJERCICIOS

Ejercicio 8: Palíndromo

```
// Esta función debe recibir un string y decir si es un palíndromo (true / false)
// Un palíndromo es una frase que se lee igual al derecho que al revés
// Haz uso de las tres funciones anteriores
function esPalindromo(string) {

}

// Ejemplos de palíndromos:

// Arde ya la yedra
// Ana lava lana
// Anita lava la tina
```

Ejercicio 9: Objeto Zoo

Realiza la modelización de un Zoológico. El zoológico deberá tener un nombre, una ubicación, un aforo máximo, un horario... ¡y todo lo que se te pueda ocurrir!

El zoológico deberá tener varias áreas, con distintos recintos, por ejemplo:

- Reptiles
 - Serpientes
 - Lagartos
- Aves
 - Aves pequeñas
 - Aves tropicales

Cada recinto debe tener un nombre, una capacidad máxima de animales, aforo máximo de personas y un conjunto de animales.

Modeliza el zoológico lo más completo que puedas.

Ejercicio 10: Zoo con vida (V1) / Página 1 de 2

Haciendo uso del zoo que definimos en el ejercicio anterior, vamos a añadirle funcionalidad:

1) Haz una función que añada un visitante nuevo:

Si el zoo está lleno no podrá entrar.

Para entrar deberá pagar la entrada que dependerá de:

Niños <14 años: gratis

Personas mayores >65 gratis

Resto: 5\$

Estudiantes: 3\$

Este importe deberá ser restado del dinero del visitante y añadido a la caja del zoo

El visitante irá a un área y una sección aleatoria,
si esta está llena, deberá buscar otro lugar

Ejercicio 11: Zoo con vida (V1) / Página 2 de 2

- 2) Crea una función que se llame ejecutarCiclo() que simule el paso de 1 hora en el zoo, deberá:
Añadir visitantes al parque y también los retire del parque
Deberá quedar reflejado que ha pasado un ciclo en el importe de las personas (tendrán menos dinero) y en la caja del parque (habrá ganado dinero)
(El ciclo simula ser una hora del parque, pero lo ejecutamos cada 3seg)
- 3) Crea una funcionalidad que simule el paso de un ciclo en un animal:
Su salud se verá afectada disminuyendo o aumentando 10 (de forma aleatoria).
Si la salud del animal desciende por debajo de 50, este deberá ir a la enfermería.
También el animal tendrá más hambre cada hora que pase (+10) cuando llegue a 100 deberá ser alimentado y pasará a tener hambre 0.
- 4) Asocia la funcionalidad anterior a la función de ejecutarCiclo() de manera que los animales vayan variando su salud y su hambre. De vez en cuando algunos animales deberán ir a la enfermería (salud menor de 50) donde recuperarán 10 de salud hasta llegar a 100. Al llegar a 100 deberán volver a su recinto.