

Ejercicios Puntuales

1. Crear un array vacío. Utilizando la función **push()** agregar al array los números del 1 al 521. Mostrar todos los números pares, la cantidad que hay y lo mismo con los impares.
 2. Utilizando la función **filter** realizar un programa que filtre los textos que tengan en algún lugar un segundo texto. Este segundo texto tiene que estar definido en una variable.
 3. Utilizando la función **reduce** realizar un programa que dado un array con strings los concatene en una nueva variable.
 4. Resolver los siguientes puntos utilizando objetos:
 - a. Generar un objeto **persona** que cuente con el nombre, apellido, dni y diasVividos por una persona.
 - b. Generar un objeto **comida** que cuente con nombre y calorías.
 - c. Agregar al objeto **persona** caloríasConsumidas y caloríasConsumidasHoy.
 - d. Agregar al objeto **persona** la función **crecer()** que incremente sus días vividos y reinicie las caloríasConsumidasHoy.
 - e. Agregar al objeto **persona** la función **comer(alimento)** que recibe una **comida** y actualice las calorías consumidas.
 - f. Agregar al objeto **persona** la función **imprimir()** que imprima todos sus datos.
 - g. Borrar la propiedad **diasVividos** y agregar **fechaDeNacimiento** y **fechaActual**. El resto de los métodos debe seguir funcionando exactamente igual incluyendo que el método **imprimir()** imprima la cantidad de **diasVividos** y crecer modifique la **fechaActual**.
-
1. Vamos a crear una (objeto) ficha de animal y rellenarlo con sus datos. Debe incluir:
 - a. Nombre
 - b. Especie
 - c. color/colores
 - d. Cantidad de patas
 - e. Habitat (domestico/salvaje)
 - f. Dieta (carnivoro/herbivoro)
 - g. Abrimos una función llamada sonido y dentro de la función que se imprima el sonido que hace el animal.
 2. Ahora imprimimos animal utilizando la función **console.log()**.
 3. Ahora imprimimos cada dato de animal individualmente utilizando la función **console.log(objeto.propiedad)**.
 4. Definir el objeto selva. Dentro de selva vamos a definir:
 - a. Continente
 - b. Pais/es (array)
 - c. Tipo (Selva/bosque/tropical/jungla)
 - d. Ficha de animal (ejercicio anterior)
 5. Imprimimos el objeto selva.
 6. Imprimimos la ficha animal.

7. Imprimimos las propiedades DENTRO de la ficha animal.
8. Medir el largo del nombre del animal utilizando length.
9. Considerando el siguiente código:

```
function asignacion() {  
    var numero = 5;  
}  
  
console.log(numero);
```

Qué valor se imprimiría?

10. Crear una función que repita la palabra “hola” con un espacio entre cada palabra 20 veces. Probarlo tanto creando una función como asignando a una variable.
11. Crear dos funciones. La primera debe llamarse **asignarAnimal()** en donde se reciba un número al azar y según ese número la función retorna un string con el nombre de un animal. La segunda debe llamarse **hacerSonido()** la cual recibirá al animal retornado por la función anterior y según qué animal sea imprimirá su sonido.
12. Creemos una función llamada **conciertoAnimal()** que asigne 10 animales al azar e imprima sus sonidos.
13. Considerando el siguiente código:

```
function test() {  
    console.log(a);  
    console.log(foo());  
  
    var a = 1;  
    function foo() {  
        return 2;  
    }  
}  
  
test();
```

Que devuelve? Porque? Esta bien hacer esto? Se puede declarar una variable después de definirla en JS?

14. Crear una función llamada “interior” dentro de otra función llamada “exterior” y que ambas imprimen un string. Intentar correr para que se impriman ambos mensajes. Se logró imprimir ambos? Porque o porque no? Lograr el efecto deseado en el caso

que no se haya logrado.

15. Crear una función que dentro de dicha función llamada “definir” que defina una variable nueva y se le asigne un valor. Ahora creamos otra función llamada “impresion” que imprime la variable definida en la otra función. Funciona la segunda función? porque? Lograr que funcione.

16. Considerando el siguiente código:

```
function imprimi() {  
  var numero = 2;  
  console.log(numero);  
};  
  
imprimi();  
console.log(numero);
```

Qué sucede? Modificarlo para que la variable número sea global.

17. Hacer una funcion para imprimir los múltiplos de 5 del 0 al 500

18. Considerando el siguiente código:

```
function aumenta(numero) {  
  numero = numero + 1;  
};  
  
var numero = 2;  
aumenta();  
console.log(numero);
```

Qué imprime? Porque?

19. Crear un array llamado “amigos” con X nombres dentro del mismo. utilizando foreach y una función anónima (callback) enumerar el nombre de cada amigo y separarlos con una coma y espacio a cada uno. Ej. “1.Juan, 2.Pepe, 3. Carolina, 4....”

Ejercicios Complementarios (aún más)

1. Haremos una función llamada **evaluarObjeto** que reciba un objeto e imprima el puntaje otorgado al objeto. Por cada atributo va a sumarle un punto al objeto si cumple alguna de las siguientes condiciones:
 - a. Si es número, ser positivo
 - b. Si es un string, tener más de 5 caracteres
 - c. Si es verdadero
2. Definiremos el objeto **pokemon** de la siguiente manera:

```
var pokemon = {  
  puntosDeVida: 100,  
  ataque: 20,  
  velocidad: 50,  
  nombre: "charmander"  
};
```

De esta manera podríamos definir distintos objetos **pokemon** con distintos puntosDeVida y distintos valores de ataque. Dado esto, realizaremos una función **pelea()** que reciba dos objetos **pokemon** y simule una batalla entre ellos. Para esto hay que cumplir las siguientes condiciones:

- Una batalla pokemon se sigue ejecutando siempre que alguno de los dos objetos siga teniendo puntos de vida.
 - En cada turno un pokemon ataca quitándole puntosDeVida a su rival. La cantidad de puntosDeVida a restar son los indicados en el ataque.
 - El pokemon con mayor velocidad comienza la batalla. En caso de contar con la misma velocidad se resuelve de forma aleatoria.
 - Sobre el final de la pelea se debe imprimir el nombre del pokemon ganado
3. Vamos a crear un juego de cartas para **n** jugadores. El objeto carta se define de la siguiente manera:

```
var carta = {  
  tipo: "instruccion",  
  puntaje: 0  
};
```

Las cartas pueden tener 4 tipos:

- Avanzar -> Su puntaje siempre es 0
- Descansar -> Su puntaje siempre es 0
- Terminar -> Su puntaje siempre es 0
- Puntaje -> Su puntaje es un número entre 1 y 50 que no se repite en ninguna otra carta.

Habrás 50 cartas de **Puntaje** en un array mientras que habrá 100 cartas de instrucciones en otro array (50 de **Avanzar**, 49 de **Descansar** y 1 de **Terminar**).

Cada jugador irá robando una carta de la pila de 100 cartas (la cual se descartará) y luego le tocará al jugador siguiente.

Si el jugador roba una carta de **Descansar** su turno termina y le toca al siguiente.

Si el jugador roba una carta de **Avanzar** roba una carta de la pila de puntajes y se le suma la cantidad de puntos que diga esta carta.

Si el jugador roba la carta de **Terminar** se termina el juego.

Al terminar el juego hay que indicar el número de jugador que tenga el mayor puntaje como el ganador.

Realizar este ejercicio utilizando tanto **break** como **continue**.

Ejercicios Complementarios (+++++)

1. Utilizando **for...in**
(<https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Sentencias/for...in>) realizar una función que dado un objeto imprima los nombres de cada uno de sus atributos y sus valores.
2. Modificar el punto 1 para que además, si algún atributo del objeto sea una función, este se ejecute.
3. Realizar una función que dado un objeto, el nombre de un atributo y un valor, se pise (o en caso de no existir se cree) con el valor dado.
4. Modificar el punto 3 para que si el atributo no existe previamente, no lo agregue.

Ejercicios Complementarios (:O)

1. Hacer una función para calcular la factorial de un número utilizando un **for**. Por ejemplo:
 $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$
2. Hacer una función para calcular el mínimo común múltiplo y el máximo común divisor de dos números.
3. Hacer una función que cada número subsiguiente es la suma de los dos anteriores.
ej. 0,1,1,2,3,5,8,13,21,34...
4. Hacer una función que al ingresarle x cantidad de número en un array que los separe en dos arrays, par e impar.
5. Rehacer el ejercicio 1 complementario (factorial) pero de manera recursiva
6. Escribir una función **recursiva** que de vuelta una palabra.
7. Crear una función **recursiva** que cuente la cantidad de movimientos necesarios para solucionar una torre de hanoi dependiendo de la cantidad de discos (3-8).
Se tienen que obedecer estas 3 reglas para lograr desplazar la torre de un palo a otro:
 - a. Sólo se puede mover un disco cada vez.
 - b. Un disco de mayor tamaño no puede descansar sobre uno más pequeño que él mismo.

- c. Solo puedes desplazar el disco que se encuentre arriba en cada varilla.



