**Unidad 3: Objetos y funciones. Avanzado**

#### RECURSIVIDAD

#### EJERCICIO 40: SUMAR TODOS LOS NÚMEROS HASTA EL ELEGIDO

Escribe una función sumTo(n) que calcule la suma de los números 1 + 2 + ... + n.

Por ejemplo:

sumTo(1) = 1

sumTo(2) = 2 + 1 = 3

sumTo(3) = 3 + 2 + 1 = 6

sumTo(4) = 4 + 3 + 2 + 1 = 10

...

sumTo(100) = 100 + 99 + ... + 2 + 1 = 5050

Escribe 3 soluciones diferentes:

1. Utilizando un bucle for.
2. Usando la recursividad, pues sumTo(n) = n + sumTo(n-1) para n > 1.

Un ejemplo del resultado:

function sumTo(n) { /\*... tu código ... \*/ }

alert( sumTo(100) ); // 5050

#### EJERCICIO 41: FACTORIAL

El [factorial](https://es.wikipedia.org/wiki/Factorial) de un número natural es un número multiplicado por "número menos uno", luego por "número menos dos", y así sucesivamente hasta 1. El factorial de n se denota como n!

Podemos escribir la definición de factorial así:

n! = n \* (n - 1) \* (n - 2) \* ...\*1

Valores de factoriales para diferentes n:

1! = 1

2! = 2 \* 1 = 2

3! = 3 \* 2 \* 1 = 6

4! = 4 \* 3 \* 2 \* 1 = 24

5! = 5 \* 4 \* 3 \* 2 \* 1 = 120

La tarea es escribir una función factorial(n) que calcule n! usando llamadas recursivas.

alert( factorial(5) ); // 120

P.D. Pista: n! puede ser escrito como n \* (n-1)! Por ejemplo: 3! = 3\*2! = 3\*2\*1! = 6

#### EJERCICIO 42: SUCESIÓN DE FIBONACCI

La secuencia de [sucesión de Fibonacci](https://es.wikipedia.org/wiki/Sucesi%C3%B3n_de_Fibonacci) tiene la fórmula Fn = Fn-1 + Fn-2. En otras palabras, el siguiente número es una suma de los dos anteriores.

Los dos primeros números son 1, luego 2(1+1), luego 3(1+2), 5(2+3) y así sucesivamente: 1, 1, 2, 3, 5, 8, 13, 21....

La sucesión de Fibonacci está relacionada la [proporción áurea](https://es.wikipedia.org/wiki/N%C3%BAmero_%C3%A1ureo) y muchos fenómenos naturales alrededor nuestro.

Escribe una función fib(n) que devuelve el número n-th de Fibonacci.

Ejemplo:

function fib(n) { /\* your code \*/ }

alert(fib(3)); // 2

alert(fib(7)); // 13

alert(fib(77)); // 5527939700884757

#### EJERCICIO 43: GENERAR UNA LISTA DE UN SOLO ENLACE

Digamos que tenemos una lista de un solo enlace (como se describe en el capítulo [Recursión y pila](https://es.javascript.info/recursion)):

let list = {

value: 1,

next: {

value: 2,

next: {

value: 3,

next: {

value: 4,

next: null

}

}

}

};

Escribe una función printList(list) que genere los elementos de la lista uno por uno.

Haz dos variantes de la solución: utilizando un bucle y utilizando recursividad.

#### EJERCICIO 44: GENERAR UNA LISTA DE UN SOLO ENLACE EN ORDEN INVERSO

Genere una lista de un solo enlace a partir de la tarea anterior [Generar una lista de un solo enlace](https://es.javascript.info/task/output-single-linked-list) en orden inverso.

Escribe dos soluciones: utilizando un bucle y utilizando recursividad.