

Funciones 04.

```
1 const frutas = ['Plátanos', 'Bananas', 'Piñas', 'Manzanas', 'Peras',  
  'Uvas', 'Pomelos'];  
2  
3 const myIndexOf = (valor, array) => {  
4  
5 }  
6  
7 myIndexOf('Piñas', frutas) // 2  
8 myIndexOf('Chirimoyas', frutas) // -1
```

Crea una función que recibe como parámetro una fruta como string y devuelve la posición del array donde esté esa fruta.

Si no encuentra el elemento que devuelva `-1`;

Guarda el valor que retorna la función y imprimela en la consola.

Funciones 05.

Parte 1:

Crea una función que reciba el array de frutas y devuelva un string con todas las frutas separadas por una coma y un espacio.

¡No utilizar el método `join()`!

¡¡¡No tiene que haber una coma al final del string!!!

```
1 // Resultado → "Plátanos, Bananas, Piñas, Manzanas, Peras, Uvas,  
  Pomelos"
```

Parte 2

Haz una segunda versión en la que tú le pases el separador que quieras.

Funciones 06.

Crea una función que reciba el array de frutas y que te devuelva un nuevo array al revés.

No utilizar el método `reverse()`

Funciones 07. - ¡Guinda!

Crear una calculadora:

- Crea un bucle en el que el programa pregunta diferentes opciones con un `prompt` (como si fuera un menú).
 - Las opciones serán (s) sumar, (r) restar, (m) multiplicar, (d) dividir, (e) exponencial, (p) porcentaje y (x) salir.
- La calculadora seguirá ejecutándose hasta que el usuario aprete la letra `x` que será salir.
- Para cada letra entre las opciones, crear la función que se encargará de hacer cada operación.
- Dentro de cada función, pediremos al usuario los datos necesarios para relizar esa operación.
- Que se imprima por consola el resultado de cada operación.
- No hace falta que las funciones devuelvan nada, solo que impriman por consola.
- Refactoriza para que haya el menor número de cosas repetidas.