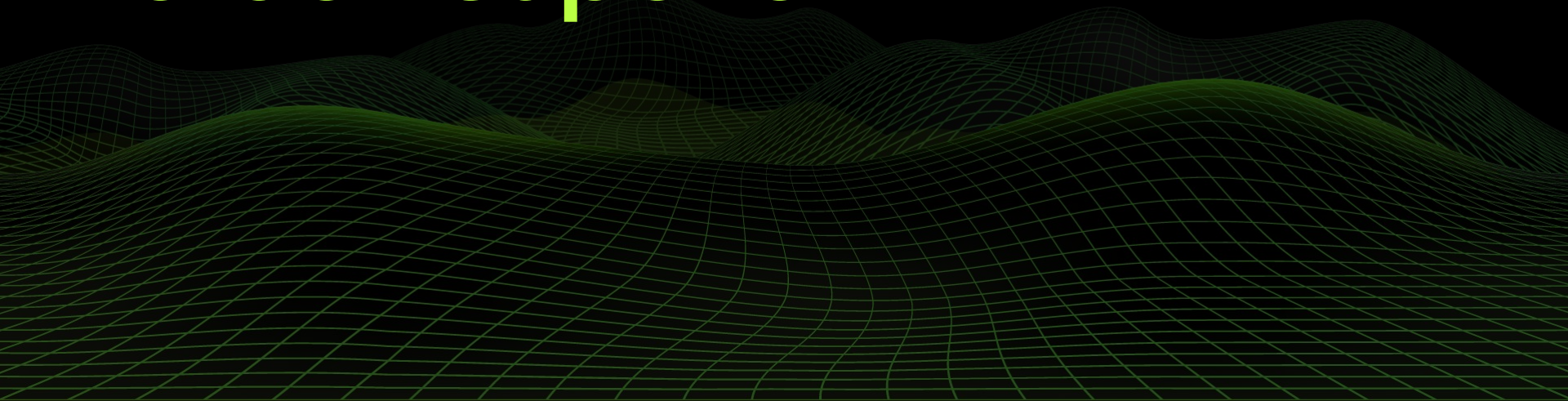
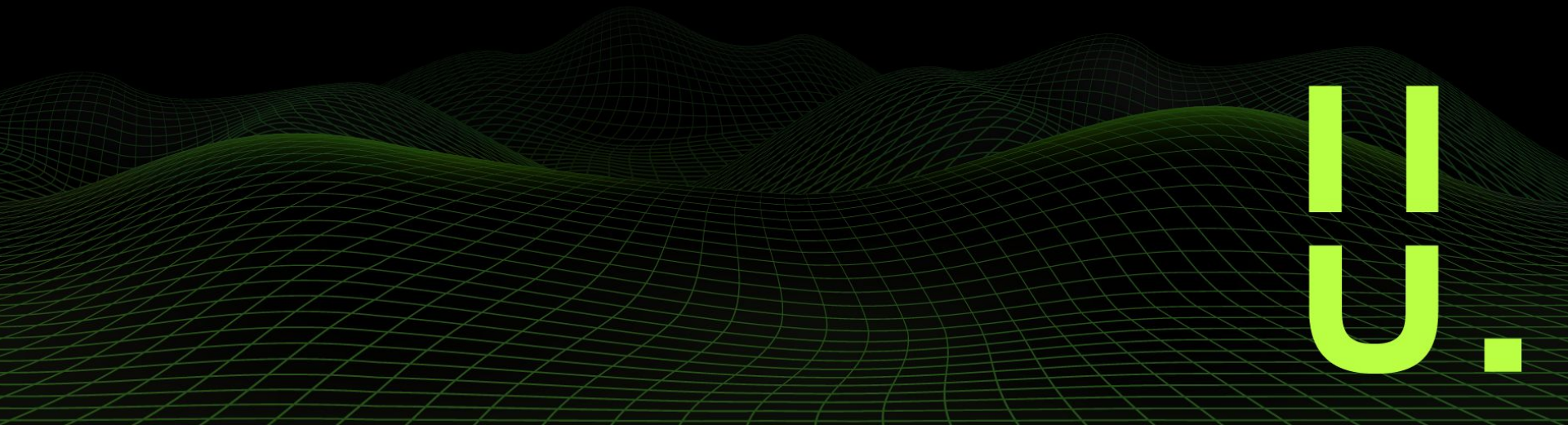
Ejercicios de funciones de orden superior



BIENVENIDOS



> Resumen Funciones Orden Superior

Qué son las Funciones de Orden Superior: Son funciones que pueden recibir otras funciones como argumentos y/o devolver funciones como resultados.

- Funciones como argumentos: Tal y como indica, al utilizar funciones de orden superior podemos utilizar otras funciones como argumentos.

  Ejemplo:

  const resultado = realizarOperacion(3, 5, suma);

  “suma” es otra función previamente creada y la utilizamos como argumento dentro de la función “realizarOperacion”

- Funciones como datos: Esto permite almacenarlas en variables, pasarlas como argumentos a otras funciones, retornarlas desde funciones y manipularlas de formas diversas.

  Ejemplo:

```
// Definimos una función de suma
function suma(a, b) {
return a + b
}
...
// Almacenamos la función en una variable
const funcionSuma = suma

//La utilizamos para pasarla como argumento
 function ejecutarOperacion(funcion, num1,num2) {
return funcion(num1, num2)
}

// Finalmente la ejecutamos.
const resultado = ejecutarOperacion(funcionSuma, 5, 3);
```

> Resumen Funciones Orden Superior

- Función Flecha: Es una manera de declarar funciones de manera concisa

Al igual que las demás funciones puede o no tener parámetros, si tiene solo un parámetro puede no llevar paréntesis. Puede tener solo una línea (No necesita llaves y el return está implícito) o múltiples líneas (llaves)

Ejemplo:

```
//En una sola línea.
const suma = (a, b) => a + b

//En una multiples líneas.
const multiplicacion = (a, b) => {
    return a * b
    }
```

- Funciones comunes de Orden Superior : map, filter, reduce, forEach, sort, find, etc

Ejemplo:

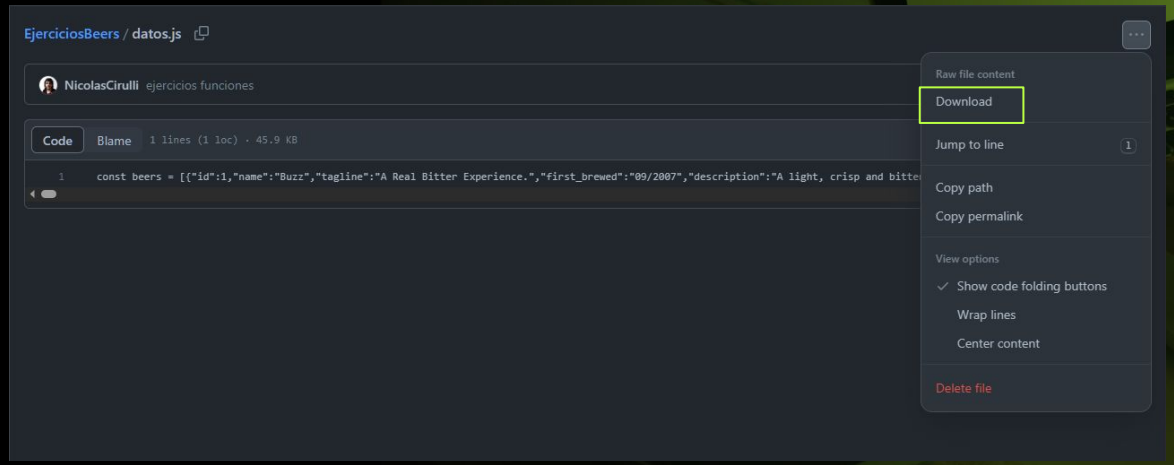
```
const numeros = [1, 2, 3, 4, 5];
const numerosAlCuadrado = numeros.map((numero) => numero * numero);
```

Material de trabajo

Funciones de orden superior

Utilizando el siguiente enlace:
EjerciciosBeers/datos.js at main · NicolasCirulli/EjerciciosBeers (github.com)

Descargue el archivo raw de datos.js para realizar la ejercitación correspondiente a las funciones de orden superior.



Ejercitación

Ejercitación

Funciones de orden superior

1

Convertir la siguiente función nombrada en una función flecha:

```
function imprimirMensaje( mensaje ){
    console.log( mensaje )
}
```

2

Convertir la siguiente función nombrada en una función flecha:

```
function crearMultiplicacion( numero1, numero2 ){
    let resultado = numero1 * numero2
    return resultado
}
```

3

Partiendo de un array const array = [1,2,3,4,5,6,7,8,9], aplicarle un map a ese array y pasarle como argumento la función nombrada mostrada en el ejemplo del punto 2. Mostrar por consola el nuevo array obtenido.

Ejercitación

Funciones de orden superior

4. Generar una función que reciba como parámetro un array de cervezas y devuelva un nuevo array con las 10 cervezas más alcohólicas

5. Generar una función que reciba como parámetro un array de cervezas y devuelva un nuevo array con las 10 cervezas menos amargas.

6. Generar una función que reciba como parámetro un array de cervezas y un nombre de una cerveza. La función deberá devolver el objeto completo que coincida con el nombre ingresado.

7. Generar una función que reciba como parámetro un array de cervezas, un valor y que devuelva el primer objeto que su propiedad ibu sea igual al valor ingresado, en caso de que no exista ninguna cerveza con ese ibu que muestre por consola un mensaje que diga “No existe cerveza con un ibu de (valor ingresado)”.

Ejercitación

Funciones de orden superior

8 Generar una función que reciba como parámetro un nombre de una cerveza y devuelva la posición en el array de esa cerveza. En caso de no encontrar el dicha cerveza se debe imprimir por consola un mensaje diciendo “(Nombre de la cerveza ingresada) no existe”.

9 Generar una función que reciba como parámetro el array de cervezas y un valor de alcohol. La función debe devolver un nuevo array con las cervezas que no excedan el nivel etílico. Cada elemento del nuevo array debe ser un objeto que tenga la propiedades nombre, alcohol (abv) y "amargor" (ibu)

10 Generar una función que reciba como parámetro un array de cervezas, un nombre de propiedad y un valor booleano. Debe devolver un nuevo array con 10 cervezas ordenadas por la propiedad ingresada como segundo argumento, de manera ascendente si el tercero es true o descendente si es false.

11 Generar una función que reciba como parámetro un array de cervezas y un id. La función debe renderizar (renderizar, dibujar, pintar, llenar, etc) en un archivo html una tabla que contenga las columnas "Name", "ABV", "IBU", y una fila por cada elemento del array. Cada fila debe tener los datos que se piden de cada una de las cervezas.

¡MUCHAS GRACIAS!

MIND
HUB.