

BOLETÍN DE EJERCICIOS DE DESESTRUCTURACIÓN EN JS

- Implementa en JS un código equivalente al del ejemplo con asignación directa pero con desestructuración.

```
let foo = ["uno", "dos", "tres"];

// sin desestructuración
let uno  = foo[0];
let dos  = foo[1];
let tres = foo[2]; // asignación en tres líneas

// con desestructuración
```

- Para extraer los valores de un array usando asignación desestructuración, sigue los siguientes pasos:
 1. Crea un array con tres valores 1, 2 y 3 y guárdalos en una variable llamada **datos**.
 2. Desestructura el array creado con una expresión simple.
Desestructura el primer valor del array en una variable llamada **a**. Salta el segundo valor del array
Desestructura el tercer valor en una variable llamada **b**.
Intenta desestructurar el cuarto valor en una variable llamada **c** y proporciona el valor 4.
 3. Imprime de la forma que desees el valor de todas las variables.
- *Intercambia los valores de dos variables usando desestructuración*
- Dado el siguiente código que representa un objeto de la predicción del tiempo con las propiedades sobre el tiempo de ayer, hoy y mañana que guardan las temperaturas más altas y bajas para cada día, emplea desestructuración para que podamos guardar la información que corresponde en unas variables llamadas bajaHoy y altaHoy. ¿Cómo sería con asignación directa?

```
const PREDICCION_LOCAL = {
  ayer: { baja: 61, alta: 75 },
  hoy: { baja: 64, alta: 77 },
  mananha: { baja: 68, alta: 80 }
};
```

- Dado el siguiente objeto que representa las temperaturas más altas de ayer, hoy y mañana, utiliza desestructuración para asignar los valores que correspondan a unas variables llamadas hoy y mañana.

```
const ALTAS_TEMPERATURAS = {
  ayer: 75,
  hoy: 77,
  mañana: 80
};
```

- Dado el siguiente código que parte de una lista de números, crea una función que elimine los dos primeros números del array y se quede con el resto. Utiliza desestructuración.

```
const origen = [1,2,3,4,5,6,7,8,9,10];
```

- Usando desestructuración y partiendo de dos arrays inicializados directamente, implementa el código que guarda los valores del array en las variables nombre, edad y trabajo.

```
const param = ['Pedro', 21 , 'Desarrollador'];
const infoUsuario = ['Juan', 'Cocinero', 34];
```

- Aunque todavía no hemos creado objetos en JS, utilizando los apuntes, vamos a ver una forma de desestructurar (extraer los datos) las propiedades de un objeto. A partir del ejemplo de código, crea el código equivalente que corresponda con la desestructuración.

```
let obj = {};
obj.nombre = 'Pepe';
obj.apell = 'Paz';
//Asignacion directa
let n = obj.nombre;
let a = obj.apell;
//otra forma de definir el objeto
obj = { nombre: 'Paco', apell: 'Zas' };
//Incluye el codigo de la desestructuración
```

- Siguiendo la línea del ejercicio anterior, desestructura el objeto dirección que se proporciona a continuación y extrae los valores de la calle y la ciudad a dos variables.

```
const direccion = {
  calle : 'Rua Caballeros',
  numero : 7,
```

```
    ciudad : 'A Coruña',  
    país : 'España',  
    cp : 15107,  
};
```

- *Desestructuración de una función. Vamos a utilizar un objeto persona que tiene las propiedades peso y alto. Y vamos a crear una función que calcule el índice de masa corporal ($BMI = \text{peso}/\text{alto}$). Deberás crear el objeto persona y la función que corresponda no desestructurada y desestructurada. Utiliza el ejercicio anterior para guiarte y los apuntes.*

//Crear el objeto siguiendo el código ejemplo anterior

// Funcion no desestructurada BMI

// Funcion desestructurada BMI