



<Talento
Tech/>

Node.JS

Clase 3: Funciones, Arrays y Métodos en JavaScript

¡Les damos la bienvenida!

 Vamos a comenzar a grabar la clase.

Índice

A yellow arrow pointing to the right, containing the number 1.

1

A teal arrow pointing to the right, containing the number 2.

2

Funciones, Arrays y Methods

- **Funciones:**
 - Declaración de funciones
 - HOF: Higher order functions (funciones como argumentos)
- **Arrays:**
 - Declaración
 - Métodos de arrays más utilizados
- **Template Literals:**
 - Uso de backticks para strings dinámicos y multilínea

Objetos, clases y operadores avanzados

- 1. **Objetos:**
 - Definición/ Características de los objetos
 - Tipos de objetos
- 2. **Clases:**
 - Sugar Syntax
 - Similitud con objetos funcionales
 - Método constructor
- 3. **Operadores avanzados:**
 - Destructuring operator - Spread operator

Objetivos de la Clase



1 Repasar Funciones

Tipos y declaraciones de funciones en JavaScript.

2 Higher Order Functions

Entender el concepto y propósito de las HOF.

3 Estructura de Arrays

Aprender sobre arrays y sus métodos más útiles.

4 Template Literals

Crear cadenas de texto dinámicas con backticks.

Funciones

¿Qué son las Funciones?

Definición

Las funciones son bloques de código reutilizables que encapsulan acciones específicas.

Importancia

Permiten organizar el código, evitar repeticiones y facilitar el mantenimiento.

Tipos de Declaración de Funciones



```
function search() {  
    // proceso encapsulado  
}
```



```
const search = function () {  
    // proceso encapsulado  
}
```



```
const search = () => {  
    // proceso encapsulado  
}
```

Funciones Declaradas

Utilizan la palabra clave 'function' seguida del nombre y paréntesis.

Funciones Expresadas

Se asignan a variables, se las conoce como funciones anónimas.

Arrow Functions

Sintaxis más concisa, introducida en ES6.



```
const subtract = (numA, numB) => {
  const resultado = numA - numB;

  return resultado;
}

subtract(20,10); // 10
```

Parámetros y Argumentos

- **Parámetros:** Variables listadas en la declaración de la función.
- **Argumentos:** Valores pasados a la función cuando es llamada.

Permiten crear funciones flexibles y reutilizables.

return

Determina qué datos devuelve una función al ser ejecutada.

El resultado de una función es elevado al scope donde está siendo invocada.

Si una función no incluye una instrucción **return** se considera del tipo **void** y devuelve el valor **undefined** por defecto.

```
function addition(a, b) {  
    return a + b;  
}
```

Higher Order Functions (HOF)

Las **HOF** son un concepto fundamental donde las funciones son pasadas como parámetros.

También son conocidas como **callbacks**.

```
function calculator(a, b, action) {
    return action(a, b);
}

console.log(calculator(20, 10, (a, b) => a + b)); // 30
console.log(calculator(20, 10, (a, b) => a - b)); // 10
```

Arrays

Introducción a Arrays

Definición

Son colecciones de datos en formato de matriz o vector.

Nos permiten agrupar conjuntos de valores relacionados en una misma variable.

Declaración

```
const fruits = ['Pera', 'Manzana', 'Frutilla', 'Durazno'];
```

Propiedades de Arrays

1 index

2 Acceso a elementos

3 .length

Los elementos de un array se identifican por su índice. Comienzan en 0 para el primer elemento.

Se usa `arrayName[index]` para acceder a elementos específicos.
También podemos utilizar `arrayName.at(index)`

Devuelve la cantidad total de elementos en el array.

Métodos de Array: Añadir y Eliminar

Métodos de Array (Array Methods)

Son funciones que poseen los array de forma nativa, con ellas seremos capaces de agregar, quitar, buscar, filtrar, modificar, cortar, y muchas cosas más sobre los elementos internos de nuestros arreglos.



Push y Unshift

Añaden elementos al final e inicio del array respectivamente.

Pop y Shift

Eliminan elementos del final e inicio del array respectivamente.



Métodos de Array: Transformación

.join()

Une elementos de un array en una cadena.

.map()

Crea un nuevo array con los resultados de una función.

.filter()

Crea un nuevo array con elementos que pasan una prueba.

.reduce()

Reduce el array a un solo valor acumulado.

Iteración de Arrays

Si bien es posible recorrer arrays con los ciclos tradicionales como un for, existen una estructura para este fin que simplifica mucho el trabajo a realizar.

ForEach

Ejecuta una función para cada elemento del array.

For...of

Permite recorrer elementos de forma más sencilla que un for tradicional.

Template Literals

Template Literals

Definición

Cadenas de texto que permiten expresiones incrustadas.

Sintaxis

Se utilizan **backticks** (`) en lugar de comillas simples o dobles.

Interpolación

Permite insertar variables usando `${expresión}` .

Ventajas de Template Literals

Multilinea

Permiten strings de varias líneas sin caracteres especiales.

Expresiones Complejas

Se pueden incluir operaciones y funciones directamente.

Legibilidad

Mejoran la claridad del código al trabajar con strings.

Recursos Adicionales y Próximos Pasos

1 MDN Web Docs

Guías detalladas sobre funciones, arrays y template literals.

2 Eloquent JavaScript

Libro interactivo para profundizar en estos conceptos.

3 Próximos Temas

Clases, objetos, módulos y manejo de promesas.



Storytelling

 **Situación inicial en TechLab**

¡Bienvenido a **TechLab**! Tenemos un proyecto emocionante entre manos: un cliente necesita una aplicación backend robusta para gestionar su tienda en línea de manera eficiente. Pero antes de que puedas unirte a nuestro equipo y formar parte de este desafío, deberás demostrar que tienes lo necesario para ser uno de nosotros.

Equipo TechLab:



Silvia

Product Owner



Luis

Diseñador UX UI



Matias

Desarrollador



Sabrina

Desarrolladora

Tu misión comienza aquí. Antes de que Matías y Sabrina, nuestros desarrolladores estrella, puedan trabajar contigo, tendrás que superar una serie de retos diseñados para poner a prueba tus habilidades.



Ejercicios Prácticos



Ejercicio Práctico

Obligatorio

Después de demostrar tus primeros pasos con Node.js, Matías y Sabrina te plantean un reto más interesante.



"Imaginá que los precios de los productos del cliente están listos para cargarse en el sistema", te dicen. Pero hay un detalle: antes de enviarlos, tenés que calcular el IVA del 21% y presentarlos de manera clara.



Tu misión:

1. Creá un array con 10 números que representen los precios de los productos.
2. Utilizá un método de array para calcular el precio con IVA incluido para cada valor.



Ejercicio Práctico

Obligatorio



Ahora que tenés los valores con IVA calculados, es momento de mostrarlos como un desarrollador profesional. Matías te desafía a usar template literals para estructurar tu salida de esta forma:

El precio es: \${valor}.- IVA incluido.

Imprimí cada precio ajustado en la consola siguiendo este formato. Cada mensaje debe ser claro y profesional, como si estuvieras preparando un reporte para el cliente.

¿Puedes demostrar tu capacidad para manejar datos y presentar resultados de manera impecable?

¡Este es tu momento!