**QUE ES UNA FUNCION CALLBACK EN JS**

* Una función de callback es una función que se pasa a otra función como un argumento, que luego se invoca dentro de la función externa.

*function imprimir(callback) {*

*callback();*

*}*

* Los callbacks aseguran que una función no se va a ejecutar antes de que se complete una tarea 🡪 ayuda a desarrollar código asíncrono (el código se ejecuta después de que ocurra una cosa y de forma secuencial en algunos casos--- js ejecuta el código de forma secuencial en orden descendente)
* LAS FUNCIONES SON OBJETOS – por lo que se pueden pasar funciones como parámetros, así como pasar objetos a funciones como parámetros
* **CALLBACK SINCRONICO:** se ejecuta inmediatamente

*function saludar(nombre) {*

*alert('Hola ' + nombre);*

*}*

*function procesarEntradaUsuario(callback) {*

*var nombre = prompt('Por favor ingresa tu nombre.');*

*callback(nombre);*

*}*

*procesarEntradaUsuario(saludar);*

* CALLABACK ASINCRONICO: son ejecutados d

**QUE ES UNA FUNCION FLECHA – arrow function**

**Descomposición de una función tradicional:**

**// Función tradicional**

function (a){

return a + 100;

}

**// Desglose de la función flecha**

**// 1. Elimina la palabra "function" y coloca la flecha entre el argumento y el corchete de** apertura.

(a) => {

return a + 100;

}

**// 2. Quita los corchetes del cuerpo y la palabra "return" — el return está implícito.**

(a) => a + 100;

**// 3. Suprime los paréntesis de los argumentos**

a => a + 100;

* Estructura básica: () => { sentencias}
* Cuando no hay argumentos o hay más de un argumento los paréntesis son obligatorios
* Si tiene más líneas de procesamiento adicionales hay que poner el return
* En funciones con nombre las expresiones de flecha (arrow function) funcionan como VARIABLES: let name = a => a + 100
* Permite escribir funciones de manera más corta: hello = () => "Hello World!";
* MANEJO DE THIS: no hay vinculación de this, en regulares representa el objeto que llamo la función, en arrow siempre representa el objeto que define el arrow function (se toma el contexto del exterior si se intenta acceder a this) – más bien, toma el valor del ámbito en el que se crean --- las tradicionales hacen referencia al objeto
* No se pueden llamar como new
* La flecha => no crea ningún enlace 🡪 SIMPLEMENTE no tiene this
* Son anónimas, por ello se le asigna a una variable si querremos llamarla