

## 1. Funciones Callback

**Definición:** Una función callback es una función que se pasa como argumento a otra función y se invoca en algún momento dentro de esa función, generalmente después de que se completa alguna operación (por ejemplo, una operación asíncrona).

**Ejemplo:**

```
function procesarDatos(data, callback) {  
    // Procesa los datos (simulación de un procesamiento)  
    console.log("Datos procesados:", data);  
    // Llama a la función callback  
    callback();  
}  
  
function finalizar() {  
    console.log("Proceso finalizado.");  
}  
  
// Uso de la función callback  
procesarDatos("Ejemplo de datos", finalizar);
```

**Ventajas:**

Permite ejecutar código en respuesta a la finalización de otra operación.  
Facilita la programación asíncrona.

## 2. Funciones Arrow

**Definición:** Las funciones arrow son una forma concisa de escribir funciones en JavaScript. Se introdujeron en ECMAScript 6 (ES6) y permiten una sintaxis más corta, además de mantener el contexto del `this` del entorno en el que fueron definidas.

**Sintaxis:**

```
const suma = (a, b) => a + b;
```

**Ejemplo:**

```
const numeros = [1, 2, 3, 4, 5];  
const dobles = numeros.map(num => num * 2); // Uso de función arrow  
console.log(dobles); // [2, 4, 6, 8, 10]
```

### 3. Uso de this

**Definición:** El valor de this en JavaScript se refiere al contexto de ejecución de una función. Su valor depende de cómo se llama a la función.

#### Ejemplos de contexto de this:

##### En una función normal:

```
function mostrarNombre() {  
  console.log(this.nombre);  
}  
  
const persona = {  
  nombre: "Juan",  
  mostrarNombre: mostrarNombre  
};  
  
persona.mostrarNombre(); // "Juan"
```

##### En una función arrow:

```
const persona = {  
  nombre: "Juan",  
  mostrarNombre: () => {  
    console.log(this.nombre); // `this` no se refiere al objeto `persona`  
  }  
};  
  
persona.mostrarNombre(); // undefined
```

#### Características del uso de this:

**Funciones normales:** El valor de this se determina por cómo se llama a la función.

**Funciones arrow:** Heredan el valor de this del ámbito en el que fueron definidas, no del contexto de ejecución.