# **Resumen Conceptual sobre Arreglos en JavaScript**

 Definición de Arreglos

Un arreglo es una colección de datos almacenados en una sola variable. Se identifica por las llaves cuadradas `[]`, lo que indica que es una estructura de lista.  
➡️ Ejemplo: `const arreglo = [1, 2, 3];` crea un arreglo con tres elementos.

 Creación de Arreglos y Uso del new Array

Aunque se puede crear un arreglo usando `new Array()`, la sintaxis preferida es utilizando las llaves cuadradas (`[]`). Se recomienda `new Array()` solo cuando se necesita definir un tamaño específico para el arreglo.  
➡️ Ejemplo: `const arreglo = new Array(100);` crea un arreglo con 100 posiciones vacías.

 Operador push y Alternativa Spread

El operador `push` permite añadir elementos a un arreglo existente, pero modifica el arreglo original. Como alternativa, es preferible usar el operador spread (`...`) para evitar modificar el arreglo original.  
➡️ Ejemplo: `const arreglo2 = [...arreglo, 5];` crea un nuevo arreglo con el contenido de `arreglo` más el valor `5`.

 Operador Spread para Copias

El operador spread (`...`) permite crear una copia del arreglo, evitando modificar el arreglo original cuando se hacen cambios. Esto es útil en programación funcional y en React.  
➡️ Ejemplo: `const arreglo2 = [...arreglo];` crea una copia independiente de `arreglo`.

 Método map para Transformar Arreglos

El método `map` permite aplicar una función a cada elemento de un arreglo y devolver un nuevo arreglo con los resultados. Este método es útil para modificar o transformar datos sin alterar el arreglo original.  
➡️ Ejemplo: `const arreglo3 = arreglo.map(num => num \* 2);` multiplica cada valor en `arreglo` por 2.

 Buenas Prácticas con Arreglos

Es recomendable usar métodos funcionales como `map` y el operador spread (`...`) para evitar modificar el arreglo original, mejorando la consistencia del código y previniendo errores.  
➡️ Ejemplo: Usar `const arregloNuevo = [...arreglo, nuevoElemento];` para añadir elementos sin modificar `arreglo`.

 Respaldo del Archivo

Es recomendable realizar copias de seguridad del código para evitar pérdidas de información. Guardar los archivos en carpetas organizadas permite mantener un registro de los ejercicios y avances.  
➡️ Ejemplo: Copiar `index.js` en la carpeta `bases` y renombrarlo como `04-arreglos.js`.