

Principio de Responsabilidad Única (Single Responsibility Principle)

Este principio establece que una clase o módulo debe tener una única razón para cambiar, es decir, debe tener una sola responsabilidad. En el contexto de tu código, cada función que define los ejemplos de métodos de Array tiene una única tarea: ejecutar un ejemplo relacionado con un método específico del array.

Ejemplo:

Cada método en el array, como `Array.prototype.at()`, `Array.prototype.concat()`, `Array.prototype.copyWithin()`, tiene una responsabilidad única de ejecutar la operación y devolver los resultados.

```
const examples = [
  {
    method: "1. Array.prototype.at()",
    description: "Accede al elemento en una posición específica (soporta índices negativos).",
    example: `let array = [10, 20, 30, 40, 50];\nlet result1 = array.at(1); // Devuelve 20\nlet result2 = array.at(-1); // Devuelve 50`,
    execute: () => {
      let array = [10, 20, 30, 40, 50];
      let result1 = array.at(1);
      let result2 = array.at(-1);
      return `Resultado 1: ${result1}, Resultado 2: ${result2}`;
    }
  }
]
```

