Principio de Inversión de Dependencias (Dependency Inversion Principle)
Este principio establece que las clases de alto nivel no deben depender de clases de bajo nivel, sino de abstracciones. Y las abstracciones no deben depender de detalles, sino que los detalles deben depender de las abstracciones.

Ejemplo:
En el código que has proporcionado, la estructura del código está definida para que la lógica de cada ejemplo de métodos sea independiente de otras implementaciones de bajo nivel, como los detalles de la manipulación del DOM. Esto significa que el procesamiento de cada ejemplo se encuentra desacoplado y puede ser fácilmente modificado sin afectar otras partes del sistema.

```
const container = document.getElementById("examples");

examples.forEach(({ method, description, example, execute }) => {
   const section = document.createElement("div");
   section.className = "method-section";
   section.innerHTML = `
      <div class="method-header"><h2>${method}</h2></div>
      <p><strong>Descripción:</strong> ${description}</p>
      <pre><code>${example}</code></pre>
      <button class="btn btn-primary" onclick="alert('${execute()}')">Ejecutar</button>
   `;
   container.appendChild(section);
});
```

Cada uno de los métodos en el array está perfectamente segregado y no tiene dependencias innecesarias entre ellos, lo que ayuda a mantener el código limpio y organizado. Además, la posibilidad de extender los métodos sin modificar el comportamiento existente sigue el principio de abierto/cerrado.