

JavaScript Tömb Metódusok Magyarázata

JavaScript tömbökhöz számos beépített metódus létezik, amelyek lehetővé teszik a tömbök módosítását és kezelését. Az alábbiakban részletesen bemutatom a `splice`, `slice`, `indexOf`, `lastIndexOf`, `concat` és `reverse` metódusokat.

1. splice Metódus

A `splice` metódus lehetővé teszi elemek hozzáadását, eltávolítását vagy cseréjét egy tömbben.

Szintaxis:

```
array.splice(start, deleteCount, item1, item2, ..., itemN);
```

- **start:** Az a pozíció, ahol a módosítás kezdődik.
- **deleteCount:** Az eltávolítandó elemek száma.
- **item1, item2, ..., itemN:** Azok az elemek, amelyeket a tömbbe kell illeszteni (opcionális).

Példa:

```
let gyumolcsok = ["alma", "banán", "cseresznye"];
gyumolcsok.splice(1, 1, "körte", "narancs"); // Az 1-es indexű elemről kezdve 1 elemet töröl, és hozzáad két új elemet
console.log(gyumolcsok); // ["alma", "körte", "narancs", "cseresznye"]
```

2. slice Metódus

A `slice` metódus egy új tömböt hoz létre, amely az eredeti tömb egy részét tartalmazza. Az eredeti tömb nem változik.

Szintaxis:

```
array.slice(start, end);
```

- **start:** A kezdő index (beleértve).
- **end:** A befejező index (kizárva). Ha nincs megadva, a tömb végéig tart.

Példa:

```
let gyumolcsok = ["alma", "banán", "cseresznye", "körte"];
let reszlet = gyumolcsok.slice(1, 3); // Új tömb az 1-es és 2-es indexű elemekkel
console.log(reszlet); // ["banán", "cseresznye"]
```

3. indexOf Metódus

Az `indexOf` metódus megkeresi az első előfordulását egy adott elemnek a tömbben, és visszaadja annak indexét. Ha az elem nem található, `-1`-et ad vissza.

Szintaxis:

```
array.indexOf(searchElement, fromIndex);
```

- **searchElement:** Az a keresendő elem.
- **fromIndex:** Az index, amelytől kezdve a keresést elindítjuk (opcionális).

Példa:

```
let gyumolcsok = ["alma", "banán", "cseresznye"];
let index = gyumolcsok.indexOf("banán");
console.log(index); // 1
```

4. lastIndexOf Metódus

A `lastIndexOf` metódus megkeresi az utolsó előfordulását egy adott elemnek a tömbben, és visszaadja annak indexét. Ha az elem nem található, `-1`-et ad vissza.

Szintaxis:

```
array.lastIndexOf(searchElement, fromIndex);
```

- `searchElement`: Az a keresendő elem.
- `fromIndex`: Az index, amelytől kezdve visszafelé a keresést elindítjuk (opcionális).

Példa:

```
let gyumolcsok = ["alma", "banán", "cseresznye", "banán"];
let index = gyumolcsok.lastIndexOf("banán");
console.log(index); // 3
```

5. concat Metódus

A `concat` metódus összekapcsol egy vagy több tömböt az eredeti tömb másolataival, és egy új tömböt ad vissza. Az eredeti tömb nem változik.

Szintaxis:

```
let newArray = array1.concat(array2, array3, ..., arrayN);
```

- `array1, array2, ..., arrayN`: Az összekapcsolandó tömbök.

Példa:

```
let gyumolcsok = ["alma", "banán"];
let zoldsegek = ["répa", "brokkoli"];
let etelek = gyumolcsok.concat(zoldsegek);
console.log(etelek); // ["alma", "banán", "répa", "brokkoli"]
```

6. reverse Metódus

A `reverse` metódus megfordítja a tömb elemeinek sorrendjét. Az eredeti tömb módosul.

Szintaxis:

```
array.reverse();
```

Példa:

```
let gyumolcsok = ["alma", "banán", "cseresznye"];
gyumolcsok.reverse();
console.log(gyumolcsok); // ["cseresznye", "banán", "alma"]
```

