



IBTISSAM OULAHCHOUM



JavaScript for Beginners Series

(مع شرح بالدارجة)

Part8





IBTISSAM OULAHCHOUM



Array Methods (Map, Filter, Reduce)





Technical Explanation

In modern JavaScript (and especially in React), we rarely use for loops to process lists. Instead, we use Higher Order Functions that result in cleaner, more readable code

هاد ال Methods بـ 3 هوما الساروت باش تخدم بـ Arrays بطريقة احترافية (Functional Programming). فـ React، غادي تخدم بيهم بزاف باش تأفشي (Display) القوائم

- **.map() (Transform):** Creates a new array by applying a function to every element in the calling array. It's used to transform data (e.g., converting raw data into HTML).

map(): كتخدم بيها إيلا بغيتي تدوز على ال Elements كاملين وتبدل فيهم شي حاجة (مثلاً عندك قائمة د الأسعار وبغيتي تزيد عليهم الضريبة). هي "كتحول" الداتا من شكل لشكل آخر.




in

- **.filter() (Select):** Creates a new array with only the elements that pass a specific test (return true).

.filter(): كتخدم بيها إيلا بغيتي "تصفي" الداتا. كتقول ليها "عطيني غير المنتوجات اللي الثمن دياهم كبر من 200 درهم"، وهي كترجع ليك Array جديد فيه غير داكشي اللي بغيتي.

- **.reduce() (Accumulate):** Executes a reducer function on each element, resulting in a single output value (like a sum).

.reduce(): كتجمع ليك الحساب كامل ف قيمة وحدة (مثلاً إيلا بغيتي تحسب الطووال ديال شي Panier).



A circular profile picture of a woman with dark hair and glasses, wearing a dark blazer over a light-colored top. A small blue square with the white letters 'in' is overlaid on the bottom left of the circle.

```
const prices = [100, 200, 300, 400];

// 1. MAP: Double the prices
// Returns a NEW array, does not change the original
const doubled = prices.map(price => price * 2);
console.log(doubled); // Output: [200, 400, 600, 800]

// 2. FILTER: Get prices above 250
const expensive = prices.filter(price => price > 250);
console.log(expensive); // Output: [300, 400]

// 3. REDUCE: Calculate the Total
// 'acc' is the accumulator (total), 'curr' is the current
// item
// '0' is the initial value
const total = prices.reduce((acc, curr) => acc + curr, 0);
console.log(total); // Output: 1000
```





QUESTION FOR YOU:

Since `.map` and `.filter` create a new array, what happens to the original prices array in the example above? Does it change or stay the same? Why is this important for React State?

////////////////////////////////////





IBTISSAM OULAHCHOUM

See you in other parts :)

