



**MehdiBouchachi**  
Web Developer

# THE COMPLETE FRONT-END DEVELOPMENT

● BY MEHDI BCH

**SECTION**

JAVASCRIPT REVIEW

**LECTURE**

WORKING WITH IMMUTABLE ARRAYS

# WORKING WITH IMMUTABLE ARRAYS

## BEFORE WE START

- 👉 In JavaScript, Arrays are a convenient way for storing and manipulating data.
- 👉 However, their **mutability** often leads to **unexpected changes** and **unintended side effects**, making debugging a headache.
- 👉 Fortunately, We have **Immutable Arrays** to answer this problem

# WORKING WITH IMMUTABLE ARRAYS

## IMMUTABLE ARRAYS

- 👉 An **immutable array** is an array that **never changes after creation**.
- 👉 Modifying elements **doesn't alter** the original array, but **creates a new one** instead.
- 👉 Immutable arrays ensure **predictable data behavior**, **facilitate easier debugging** by maintaining a clear data history, and enhance performance by **avoiding unnecessary re-renders**.

# WORKING WITH IMMUTABLE ARRAYS

## ADDING AN ELEMENT

- 👉 Immutable arrays ensure data integrity by preserving the original array.
- 👉 A new array is created with the added element.
- 👉 We use the **spread operator** (...) to create a new array with the added element.

```
1  const newMovie = {  
2    id: 6,  
3    title: "The Shawshank Redemption",  
4    publicationDate: "1994-10-14",  
5    director: "Frank Darabont",  
6  };  
7  const moviesAfterAdd = [...data, newMovie];
```

# WORKING WITH IMMUTABLE ARRAYS

## DELETING AN ELEMENT

- 👉 Immutable arrays maintain the original array structure while removing elements.
- 👉 A new array is created without the deleted element.
- 👉 We use the **filter method** to create a new array excluding the deleted element.



```
1 const moviesAfterDelete = moviesAfterAdd.filter((movie) => movie.id !== 2);
```

# WORKING WITH IMMUTABLE ARRAYS

## EDITING AN ELEMENT

- 👉 Immutable arrays facilitate editing elements without altering the original array.
- 👉 A new array is created with the edited element.
- 👉 We use the **map method** to create a new array with the edited element.



```
1  const updatedMovies = movies.map((movie) => {  
2    return movie.id == 1 ? { ...movie, director: "Rayan" } : movie;  
3  });
```

**SEE YOU SOON...**