

## 217. Contains Duplicate

Easy

 Topics

 Companies

Given an integer array `nums`, return `true` if any value appears **at least twice** in the array, and return `false` if every element is distinct.

### Example 1:

**Input:** `nums = [1,2,3,1]`

**Output:** `true`

**Explanation:**

The element 1 occurs at the indices 0 and 3.

### Example 2:

**Input:** `nums = [1,2,3,4]`

**Output:** `false`

**Explanation:**

All elements are distinct.

### Example 3:

**Input:** `nums = [1,1,1,3,3,4,3,2,4,2]`

**Output:** `true`

### Constraints:

- `1 <= nums.length <= 105`
- `-109 <= nums[i] <= 109`

## Python:

```
class Solution(object):
    def containsDuplicate(self, nums):
        """
        :type nums: List[int]
        :rtype: bool
        """
        # Using a set to track seen numbers
        seen = set()
        for num in nums:
            if num in seen:
                return True # Duplicate found
            seen.add(num)
        return False # All elements are distinct
```

## JavaScript:

```
/**
 * @param {number[]} nums
 * @return {boolean}
 */
var containsDuplicate = function(nums) {
    // Create a Set to store unique numbers
    let seen = new Set();

    // Loop through each number in the array
    for (let num of nums) {
        // If the number already exists in the Set, duplicate found
        if (seen.has(num)) {
            return true;
        }
        // Otherwise, add the number to the Set
        seen.add(num);
    }

    // If no duplicates found, return false
    return false;
};
```

## Java:

```
import java.util.HashSet;

class Solution {
    public boolean containsDuplicate(int[] nums) {
        HashSet<Integer> set = new HashSet<>();

        for (int num : nums) {
            if (set.contains(num)) {
                return true; // Duplicate found
            }
            set.add(num);
        }

        return false; // No duplicates found
    }
}
```