

# Problem 2794: Create Object from Two Arrays

## Problem Information

Difficulty: [Easy](#)

Acceptance Rate: 0.00%

Paid Only: No

## Problem Description

Given two arrays

keysArr

and

valuesArr

, return a new object

obj

. Each key-value pair in

obj

should come from

keysArr[i]

and

valuesArr[i]

If a duplicate key exists at a previous index, that key-value should be excluded. In other words, only the first key should be added to the object.

If the key is not a string, it should be converted into a string by calling

`String()`

on it.

Example 1:

Input:

```
keysArr = ["a", "b", "c"], valuesArr = [1, 2, 3]
```

Output:

```
{"a": 1, "b": 2, "c": 3}
```

Explanation:

The keys "a", "b", and "c" are paired with the values 1, 2, and 3 respectively.

Example 2:

Input:

```
keysArr = ["1", 1, false], valuesArr = [4, 5, 6]
```

Output:

```
{"1": 4, "false": 6}
```

Explanation:

First, all the elements in keysArr are converted into strings. We can see there are two occurrences of "1". The value associated with the first occurrence of "1" is used: 4.

Example 3:

Input:

```
keysArr = [], valuesArr = []
```

Output:

```
{}
```

Explanation:

There are no keys so an empty object is returned.

Constraints:

keysArr

and

valuesArr

are valid JSON arrays

$2 \leq \text{JSON.stringify(keysArr).length, JSON.stringify(valuesArr).length} \leq 5 * 10^5$

5

keysArr.length === valuesArr.length

## Code Snippets

JavaScript:

```
/**  
 * @param {Array} keysArr  
 * @param {Array} valuesArr  
 * @return {Object}  
 */  
var createObject = function(keysArr, valuesArr) {
```

```
};
```

## TypeScript:

```
type JSONValue = null | boolean | number | string | JSONValue[] | { [key: string]: JSONValue };

function createObject(keysArr: JSONValue[], valuesArr: JSONValue[]): Record<string, JSONValue> {
    ...
}
```

## Solutions

### JavaScript Solution:

```
/**
 * Problem: Create Object from Two Arrays
 * Difficulty: Easy
 * Tags: array, string
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */

/**
 * @param {Array} keysArr
 * @param {Array} valuesArr
 * @return {Object}
 */
var createObject = function(keysArr, valuesArr) {
    ...
};
```

### TypeScript Solution:

```
/**
 * Problem: Create Object from Two Arrays
```

```
* Difficulty: Easy
* Tags: array, string
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/
type JSONValue = null | boolean | number | string | JSONValue[] | { [key: string]: JSONValue };

function createObject(keysArr: JSONValue[], valuesArr: JSONValue[]): Record<string, JSONValue> {
}
```