

# Problem 2619: Array Prototype Last

## Problem Information

**Difficulty:** [Easy](#)

**Acceptance Rate:** 0.00%

**Paid Only:** No

## Problem Description

Write code that enhances all arrays such that you can call the

`array.last()`

method on any array and it will return the last element. If there are no elements in the array, it should return

-1

You may assume the array is the output of

`JSON.parse`

Example 1:

Input:

```
nums = [null, {}, 3]
```

Output:

Explanation:

Calling nums.last() should return the last element: 3.

Example 2:

Input:

```
nums = []
```

Output:

```
-1
```

Explanation:

Because there are no elements, return -1.

Constraints:

arr

is a valid JSON array

$0 \leq \text{arr.length} \leq 1000$

## Code Snippets

JavaScript:

```
/**  
 * @return {null|boolean|number|string|Array|Object}  
 */  
Array.prototype.last = function() {  
  
};  
/**
```

```
* const arr = [1, 2, 3];
* arr.last(); // 3
*/
```

## TypeScript:

```
interface Array<T> {
    last(): T | -1;
}

Array.prototype.last = function() {

};

/** 
 * const arr = [1, 2, 3];
 * arr.last(); // 3
 */
```

## Solutions

### JavaScript Solution:

```
/**
 * Problem: Array Prototype Last
 * Difficulty: Easy
 * Tags: array
 *
 * 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
 */

/**
 * @return {null|boolean|number|string|Array|Object}
 */
Array.prototype.last = function() {

};
```

```
/**  
 * const arr = [1, 2, 3];  
 * arr.last(); // 3  
 */
```

## TypeScript Solution:

```
/**  
 * Problem: Array Prototype Last  
 * Difficulty: Easy  
 * Tags: array  
 *  
 * 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  
 */  
  
interface Array<T> {  
    last(): T | -1;  
}  
  
Array.prototype.last = function() {  
};  
  
/**  
 * const arr = [1, 2, 3];  
 * arr.last(); // 3  
 */
```