

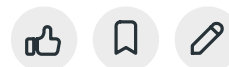


[← JS-Array](#) [JS-String](#) [JS-Function](#) [JS-Map](#) [JS-Math](#) [JS-Date](#) [JS-Number](#) [JS-Object](#) [JS-Promise](#) [→](#)

# How to clone array in ES6 ?



Srishtibajpai



[Read](#) [Discuss](#) [Courses](#) [Practice](#) [Video](#)

The [spread operator](#) in ES6 is used to clone an array, whereas the [slice\(\)](#) method in JavaScript is an older way that provides 0 as the first argument. These methods create a new, independent array and copy all the elements of oldArray to the new one i.e. both these methods do a shallow copy of the original array.

## Syntax:

```
// Older way
var clonedArray= oldArray.slice(0)

// ES6 way: spread operator
var clonedArrayES6= [...oldArray]
```

**Example:** This example uses the spread operator to copy an array.

## javascript

```
// Cloning array using spread
// operator- ES6

const oldArray = ["dog1", "dog2", "dog3"];

const clonedArrayES6 = [...oldArray];

// ["dog1", "dog2", "dog3"]
console.log(clonedArrayES6);
```

## Output

```
[ 'dog1', 'dog2', 'dog3' ]
```

**Equality and sameness:** Unlike the “=” operator, which creates a new variable that just points to the original array instead of copying its elements, the spread operator creates a new, cloned array, with different references but the same values. Hence “=” operator creates a deep copy of the original array but the spread operator does a shallow copy. The array created by the spread operator has the same value as that of the old array but, is not as same as the old array.

**Example:** This example shows the use of the above-explained approach.

---

## javascript

```
// Equality and sameness in cloning array

const oldArray = ["dog1", "dog2", "dog3"];

const clonedArrayES6 = [...oldArray];
const newArray = oldArray;

// False, i.e. shallow copy
console.log(clonedArrayES6 === oldArray)

// True, i.e. deep copy
console.log(newArray === oldArray)
```

## Output

```
false
true
```

**Note:** All the above examples can be tested by typing them within the script tag of HTML or directly into the browser’s console.

Last Updated : 14 Apr, 2023