



## **Coner Murphy**







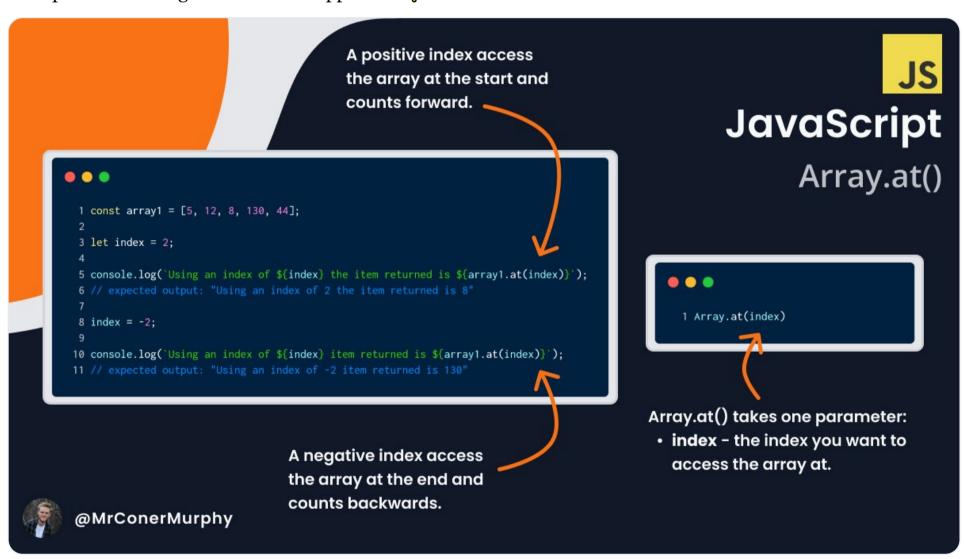
Massive JavaScript Resource Collection

All 37 JavaScript Array methods (current & experimental) explained in images 👇

1 Array .at()

.at() lets us access elements in an Array by providing an index.

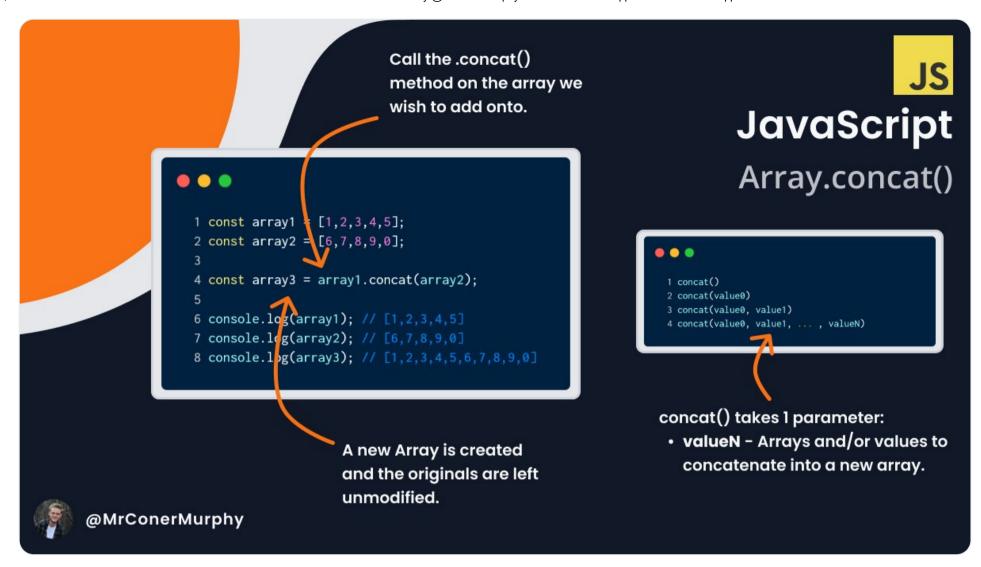
Both positive and negative ones are supported.



2 Array .concat()

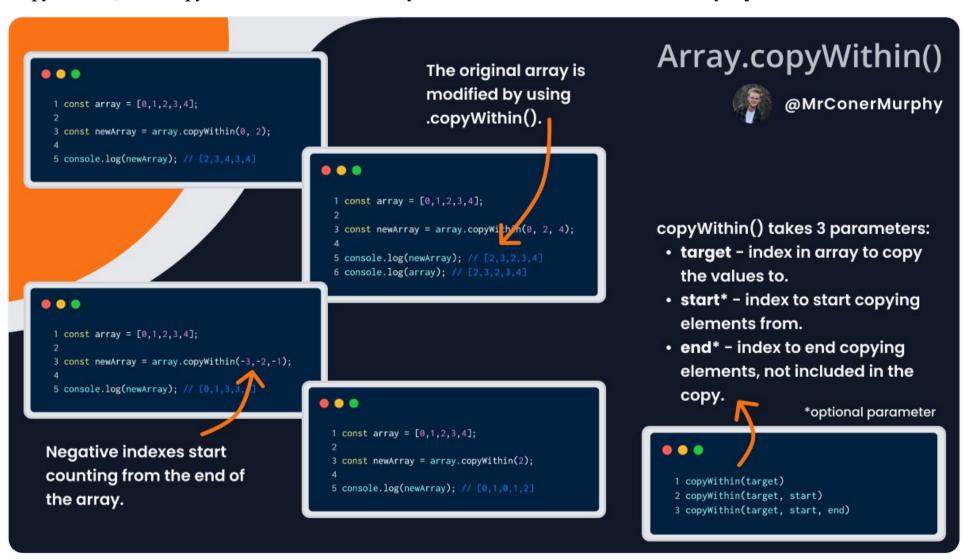
This method lets us join multiple arrays into one while leaving the original one untouched.

New arrays are added to the end of the previous ones.  $\P$ 



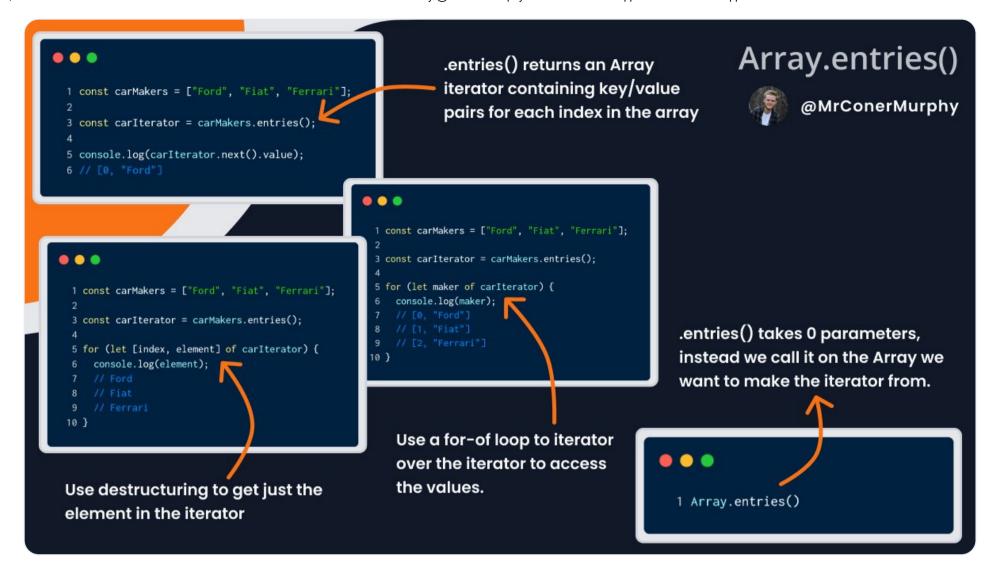
3 Array .copyWithin()

.copyWithin() let's copy elements within an array to another location within that array.

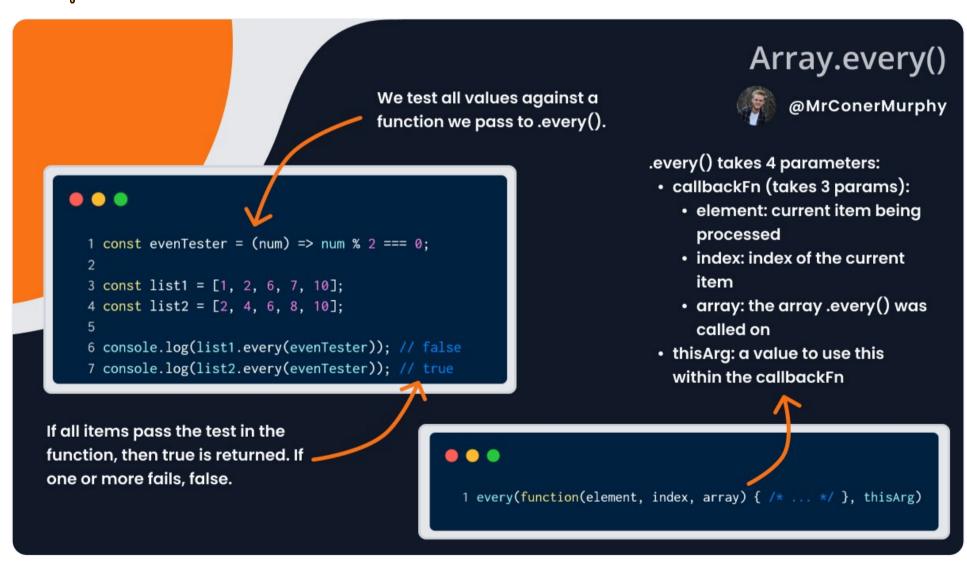


4 Array .entries()

.entries() returns a key/value pair for each index in the array, we can then loop through the iterator using a for...of loop.

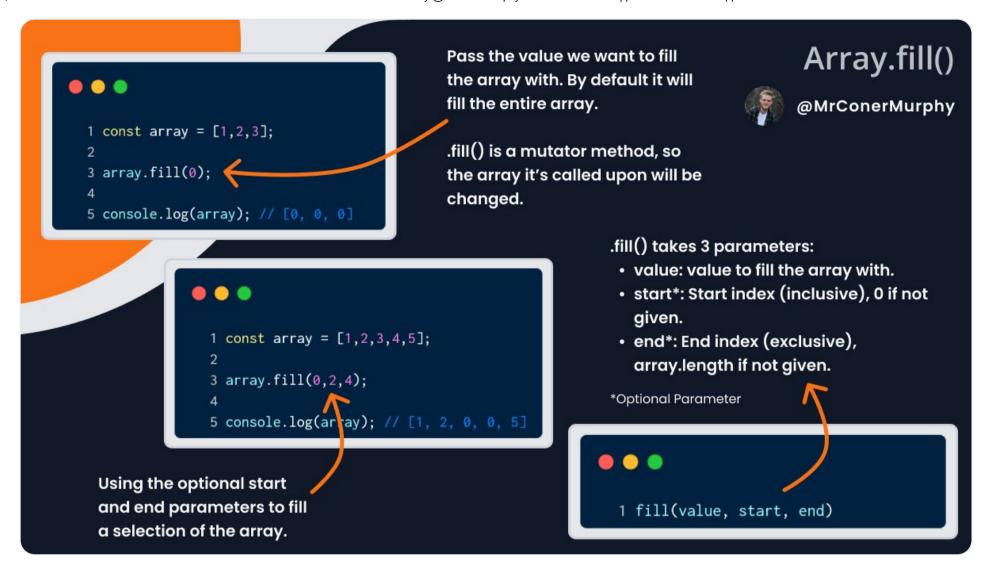


5 Array .every()



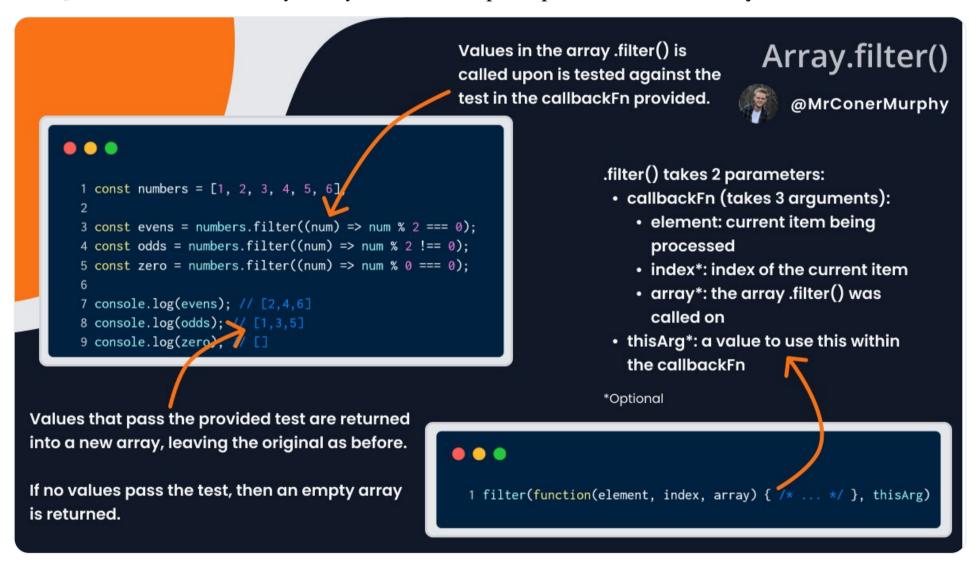
6 Array .fill()

.fill() lets us fill either an entire array or part of one with a given value.



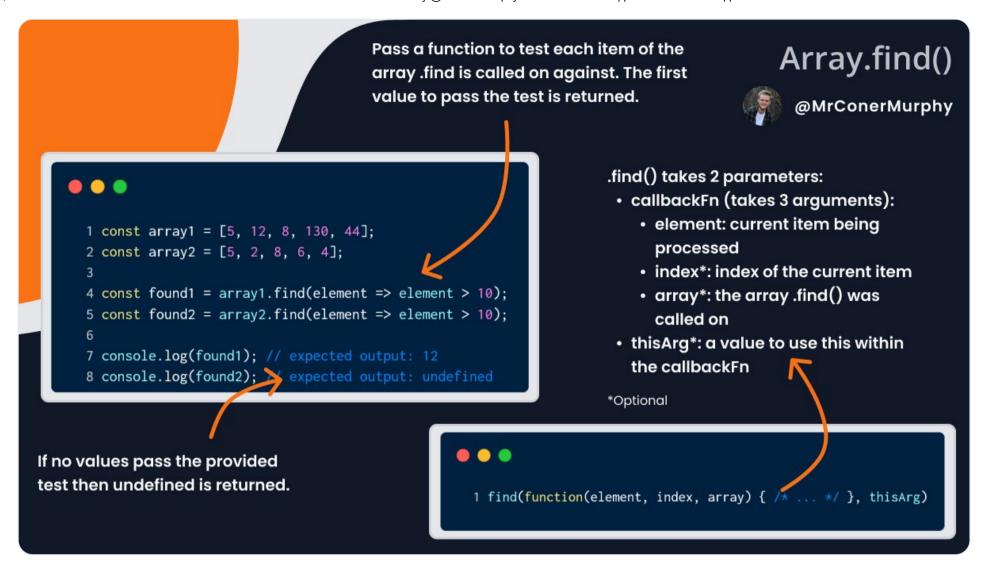
7 Array .filter()

.filter() allows us to filter an array to only the values that pass a provided test function.



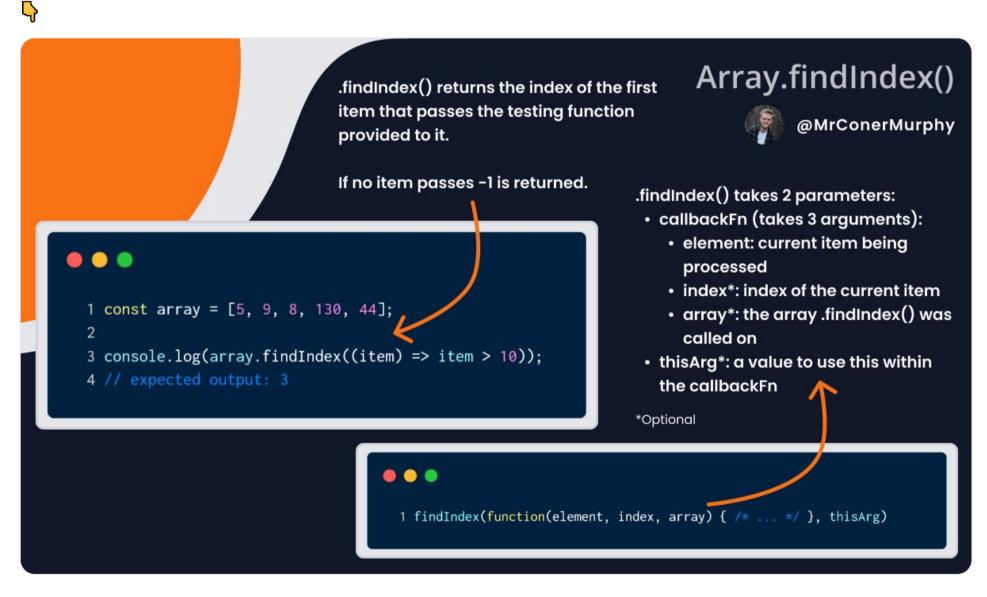
8 Array .find()

.find() allow us to test each item in an array, the first value that passes the test is returned.  $\P$ 



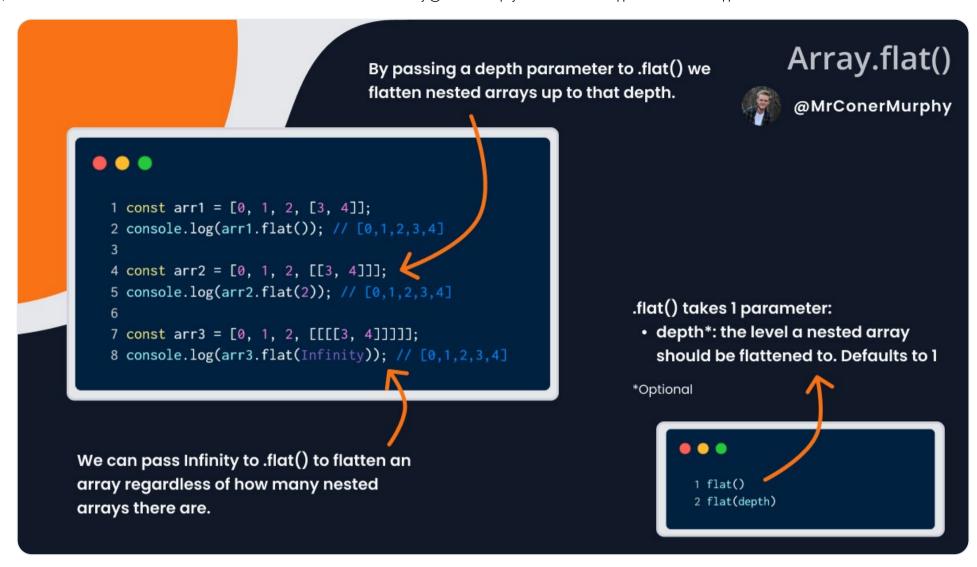
9 Array .findIndex()

Similar to .find() but this time instead of returning the first element that passes the test, we return the index of it.

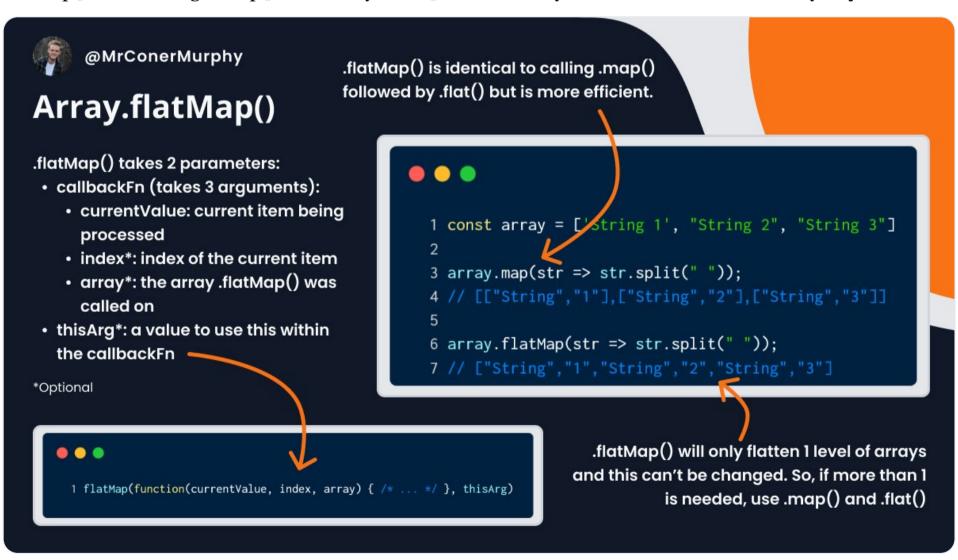


1 O Array .flat()

.flat() lets us flatten nested arrays down levels. We can control the number of levels falttened by using the depth parameter.  $\bigcirc$ 



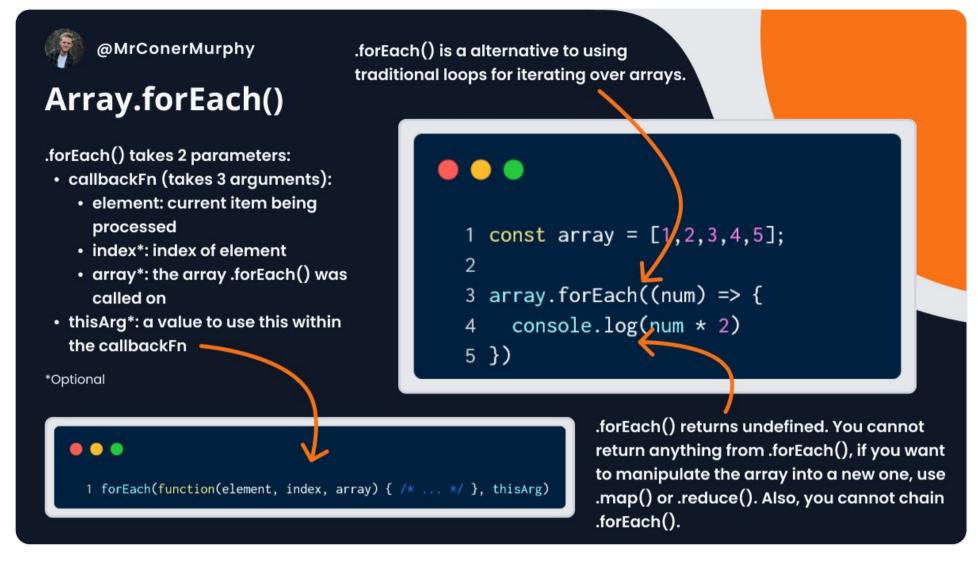
1 Array .flatMap()



1 2 Array .forEach()

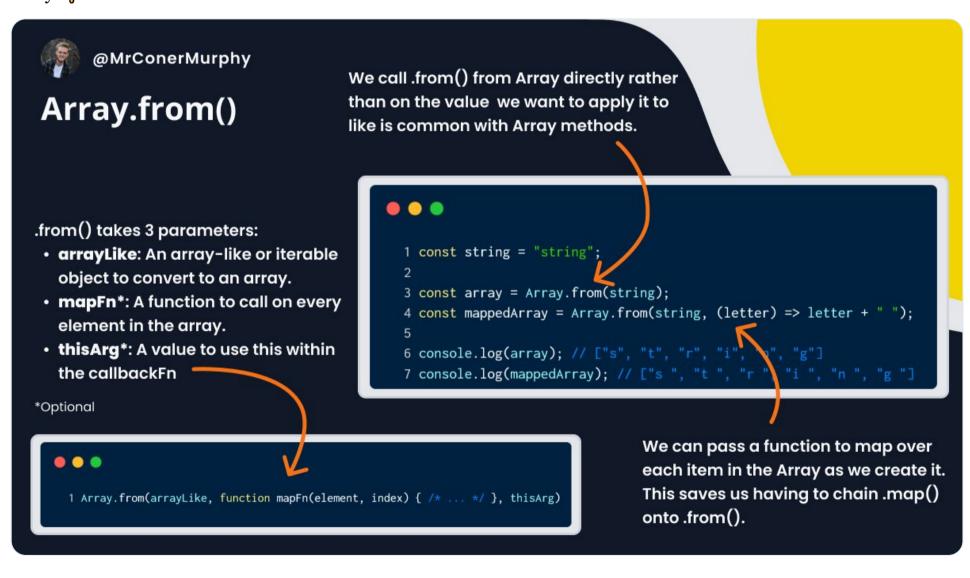
.forEach() allows us to iterate over an array. .forEach() does not return any values from the array, only undefined.





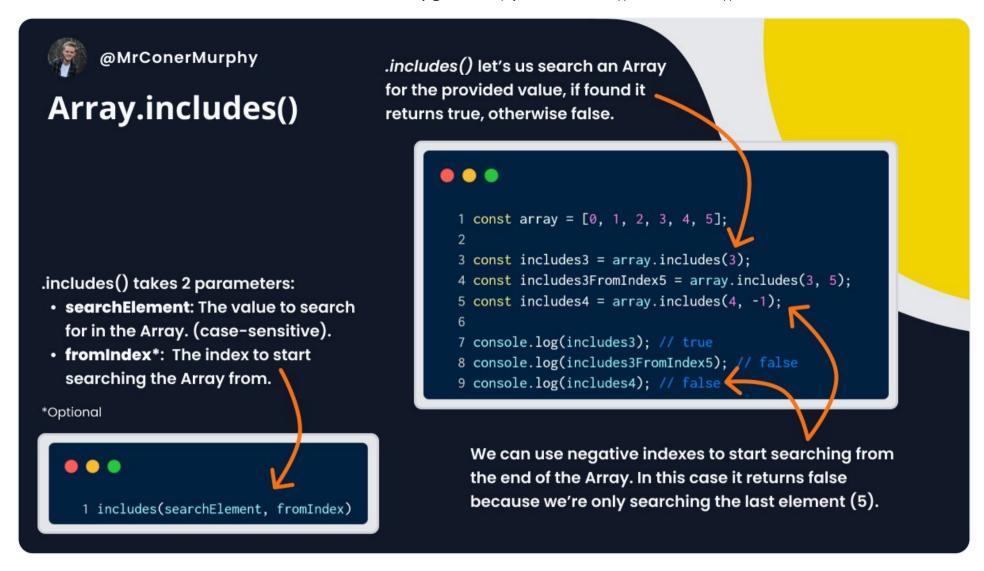
1 3 Array .from()

.from() lets us create a new array from an array-like value, we can also map over each item as we create the new array \$\infty\$



1 4 Array .includes()

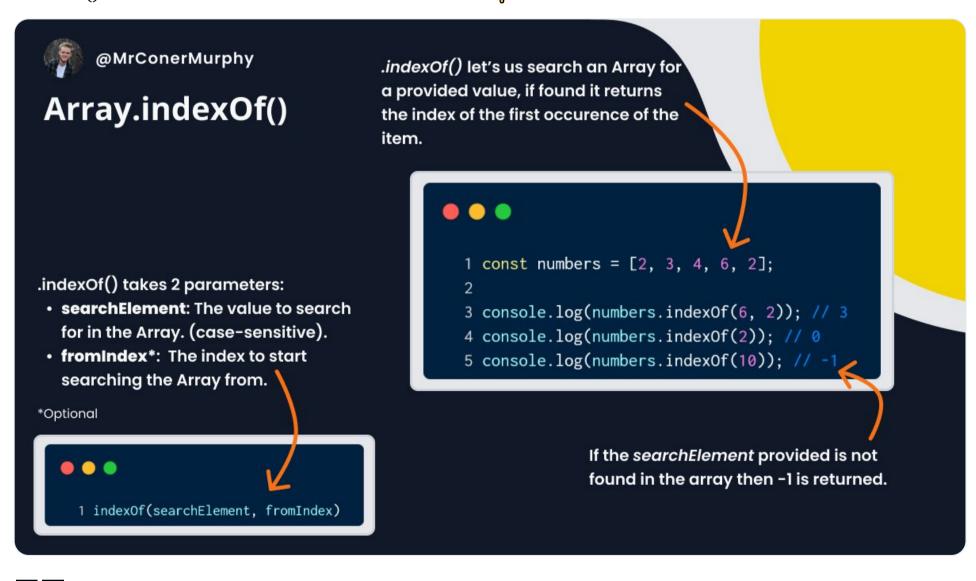
includes() lets us search an array for a provided value if it's found in the array, true is returned otherwise false



1 5 Array .indexOf()

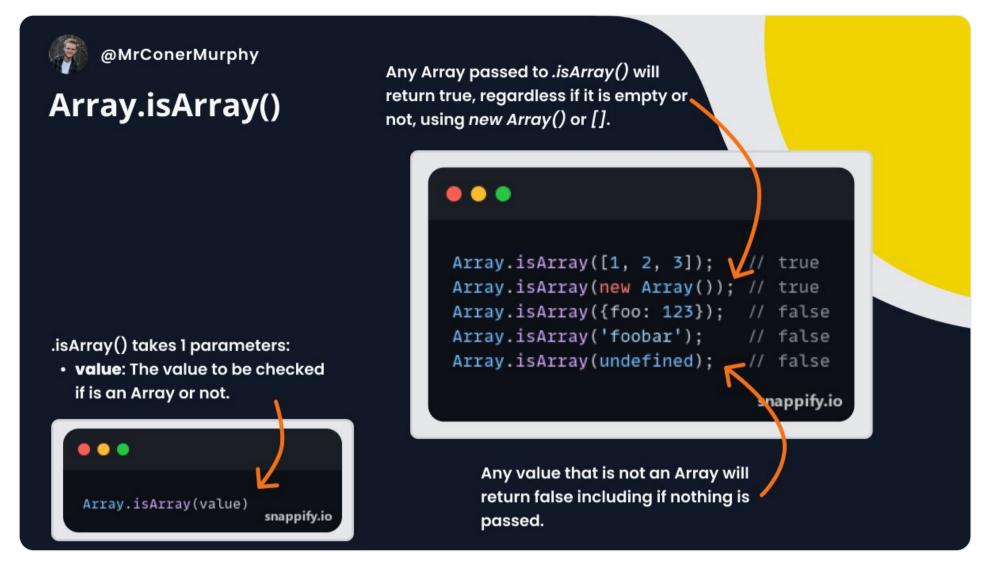
.indexOf() lets us search an array for a given value, if it's found it's index is returned.

.indexOf() will return on the first occurrence of the value



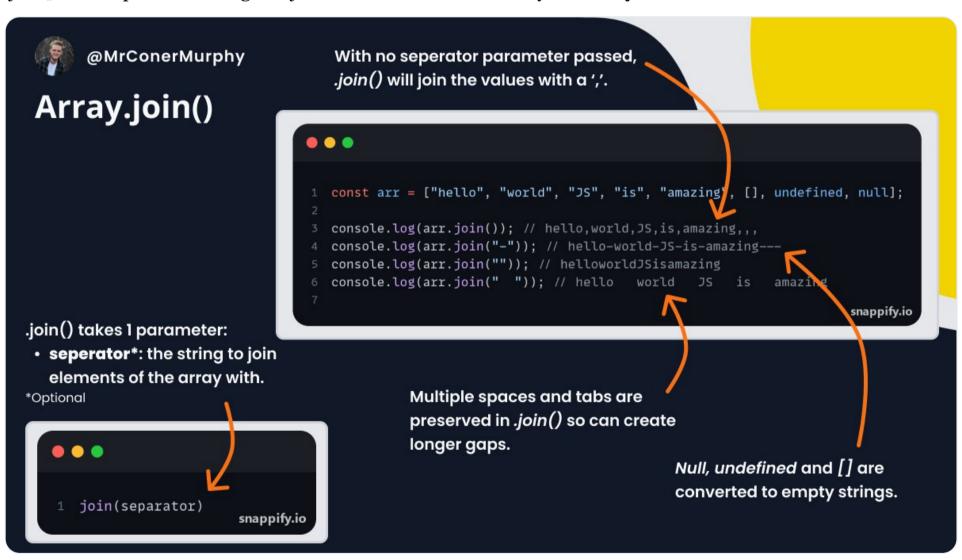
1 6 Array .isArray()

.isArray() allows us to test if a provided value is an Array or not. •



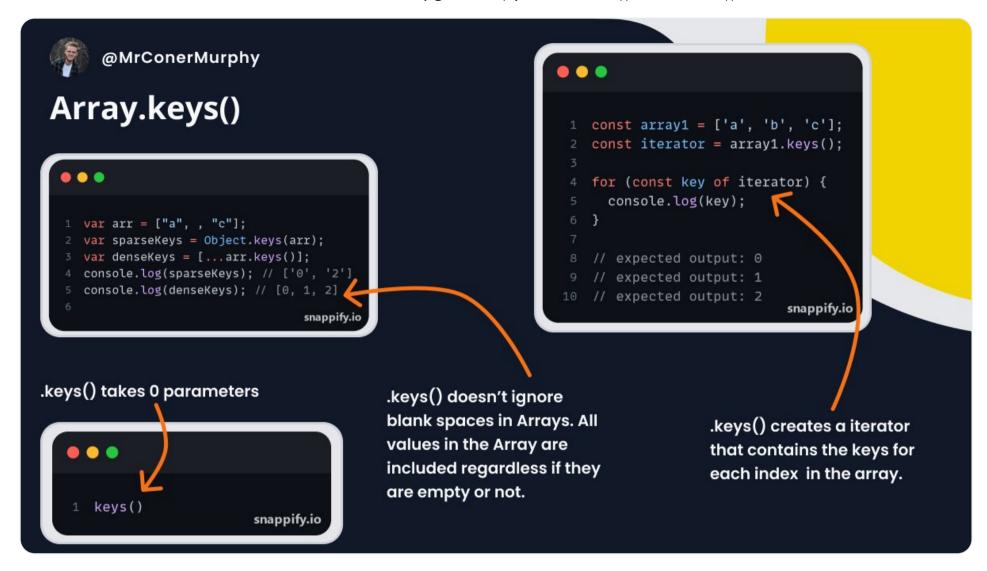
1 7 Array .join()

.join() takes a provided string and joins each element in the array with it.



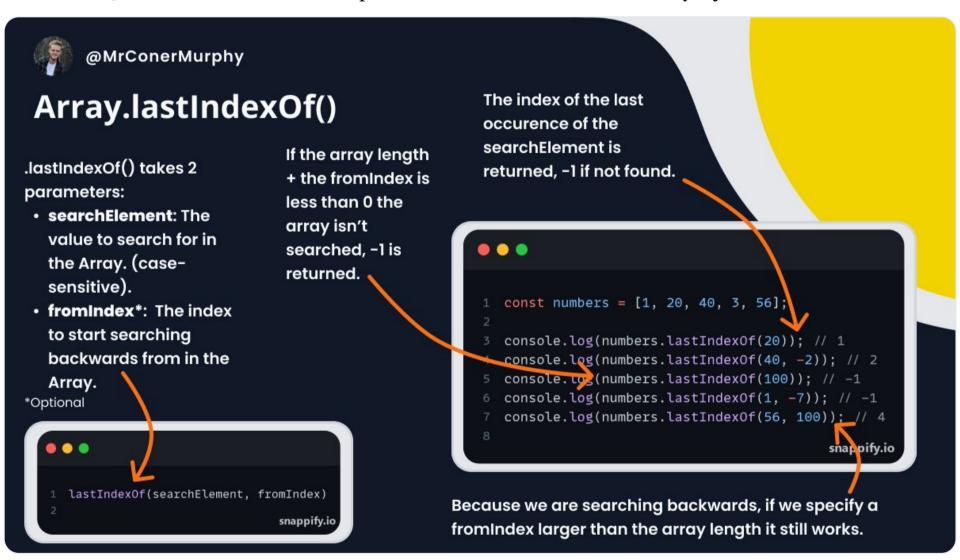
1 8 Array .keys()

.keys() creates an iterator that contains all of the indexes in the array, black spaces aren't ignored.



1 9 Array .lastIndexOf()

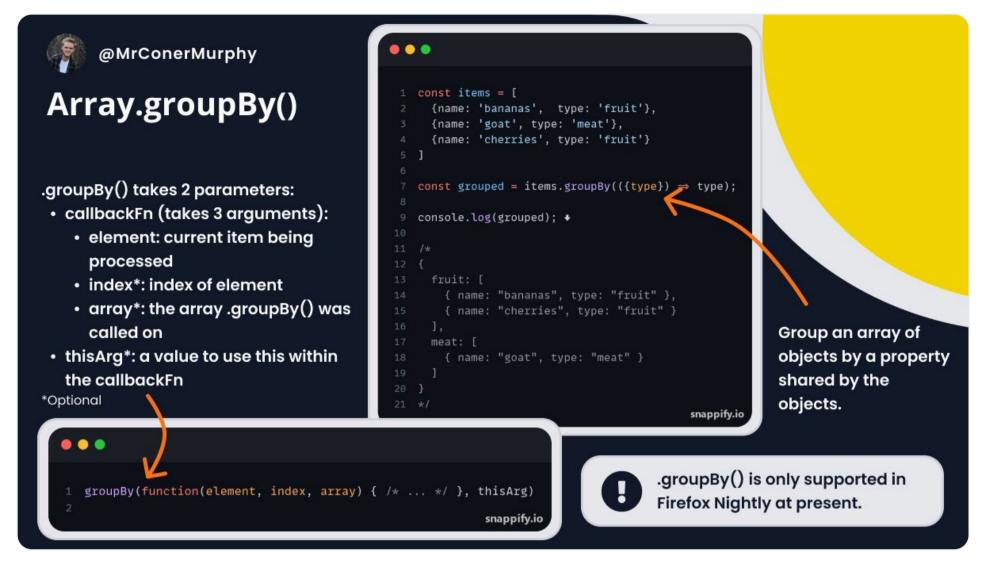
.lastIndexOf() returns the last index that a provided element is found at in an array.



2 O Array .groupBy()

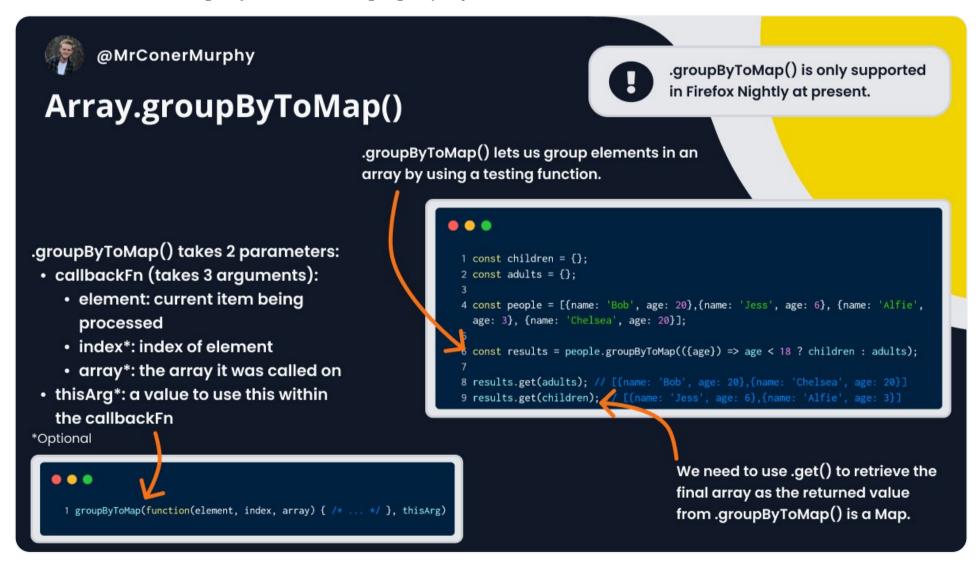
.groupBy() is one of the newest array methods, it allows us to group objects in an array by properties they share.

**-**



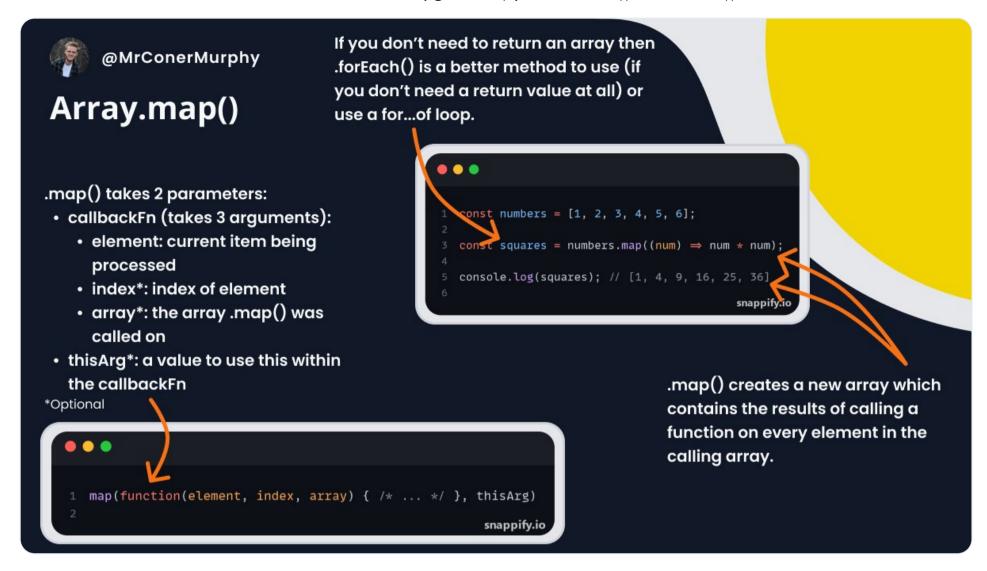
2 1 Array .groupByToMap()

.groupByToMap() was added along with .groupBy(), it allows us to group objects like .groupBy() but it then allows us to add them to a Map object based on a property.  $\cite{Q}$ 



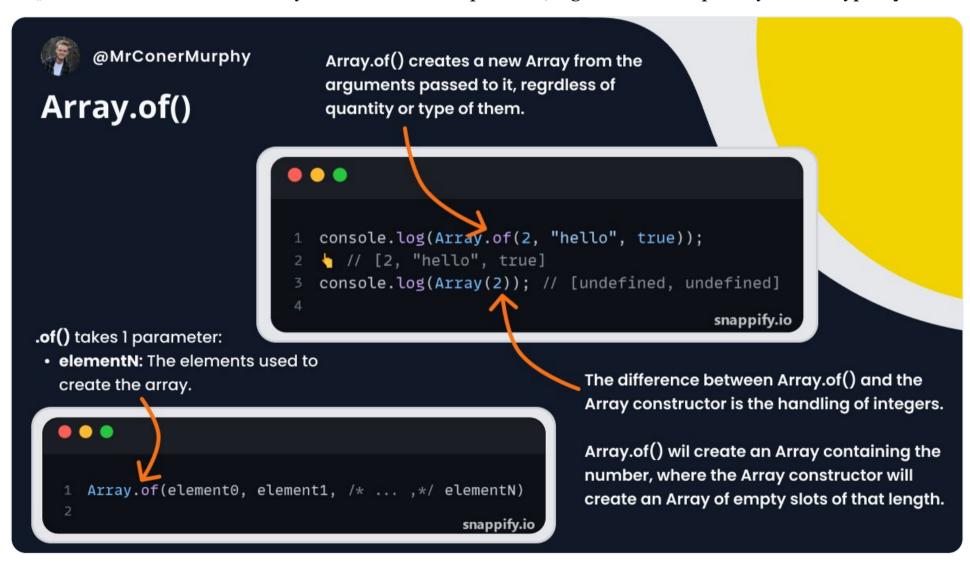
2 Array .map()

.map() lets us iterate over an array and return a new array with the new values from any manipulations we perform.  $\P$ 



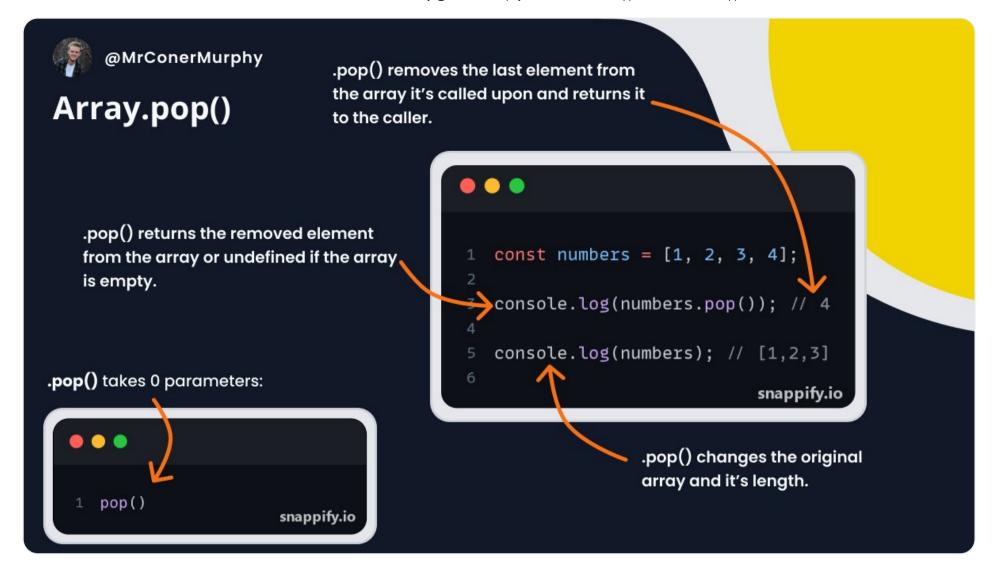
2 3 Array .of()

.of() allows us to create a new array from the values we pass to it, regardless of the quantity or their type.



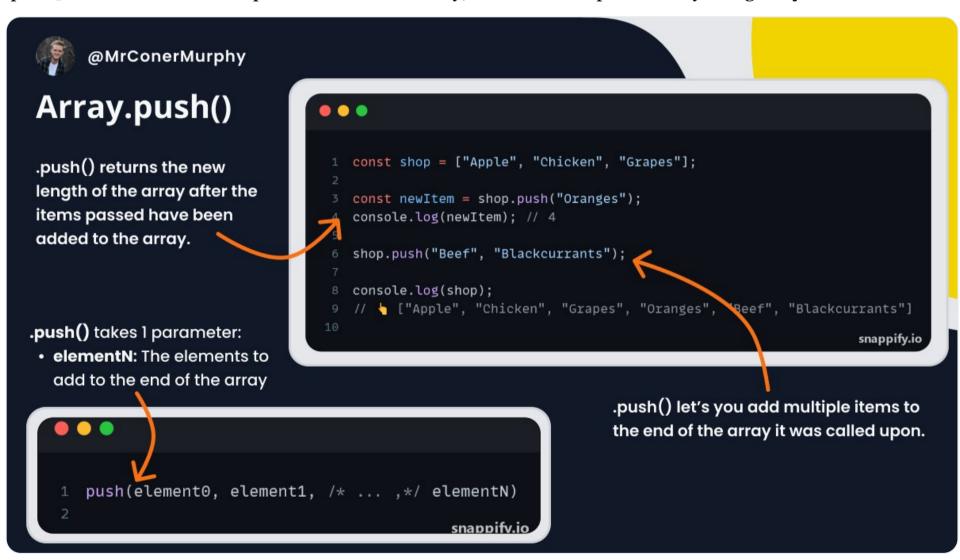
2 4 Array .pop()

.pop() removes the last item from the array and returns it.  $\P$ 



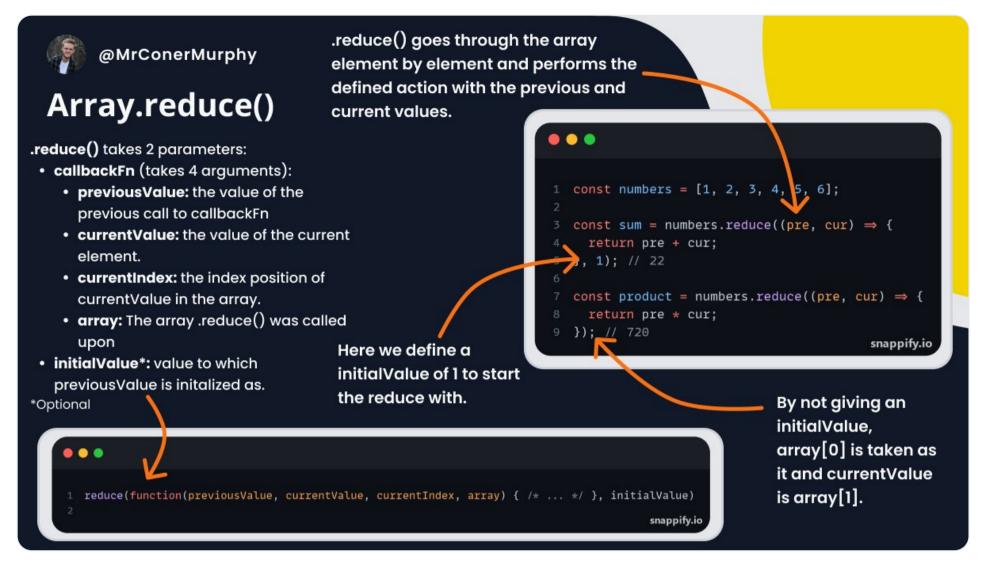
2 5 Array .push()

.push() allows us to add multiple new items to an array, it returns the updated array's length.



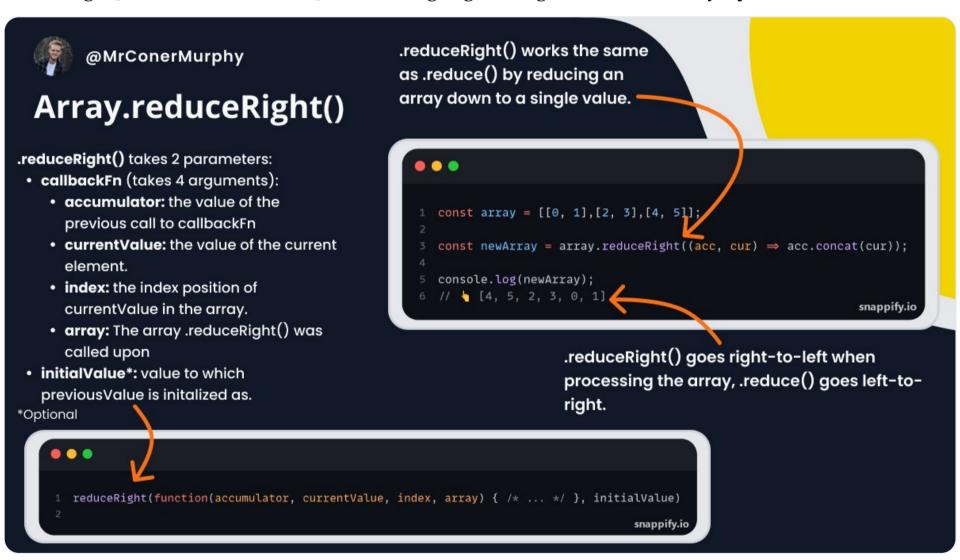
2 6 Array .reduce()

.reduce() lets us define a custom reducer function to be performed on each element in the array, going from left to right in the array.  $\P$ 



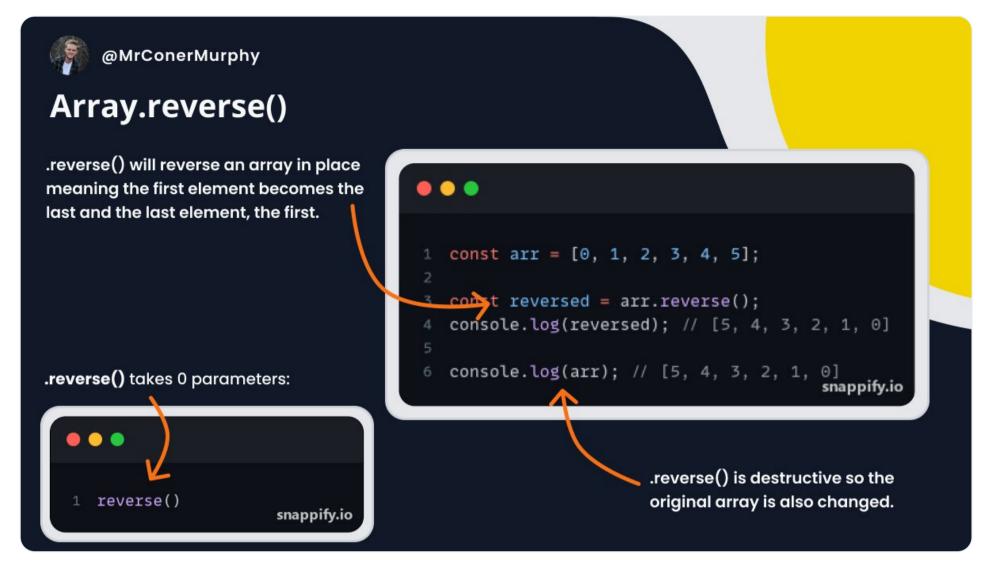
2 7 Array .reduceRight()

.reduceRight() is the same as .reduce() but instead going from right to left in the array.



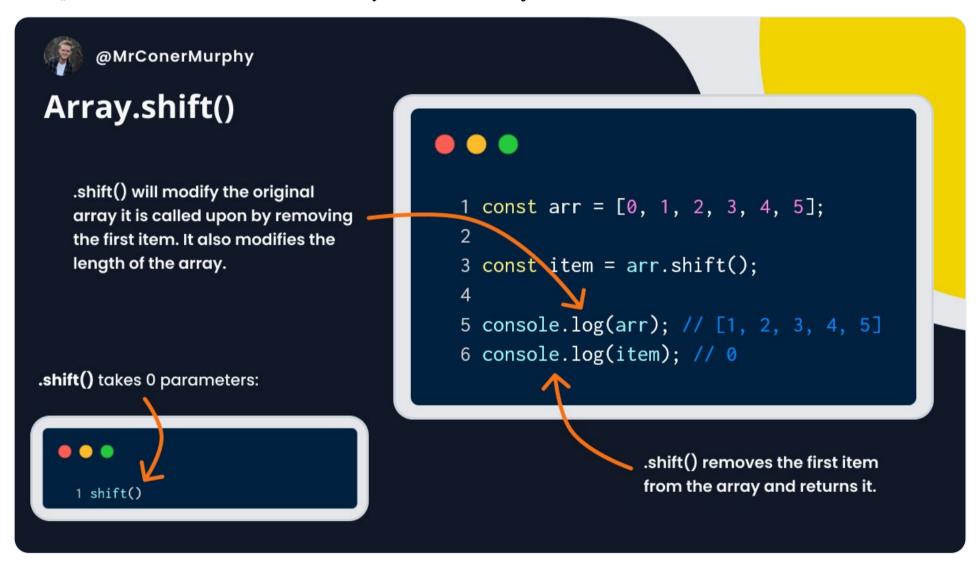
2 8 Array .reverse()

.reverse() will reverse an array in place, meaning the first item becomes the last and the last the first.



2 9 Array .shift()

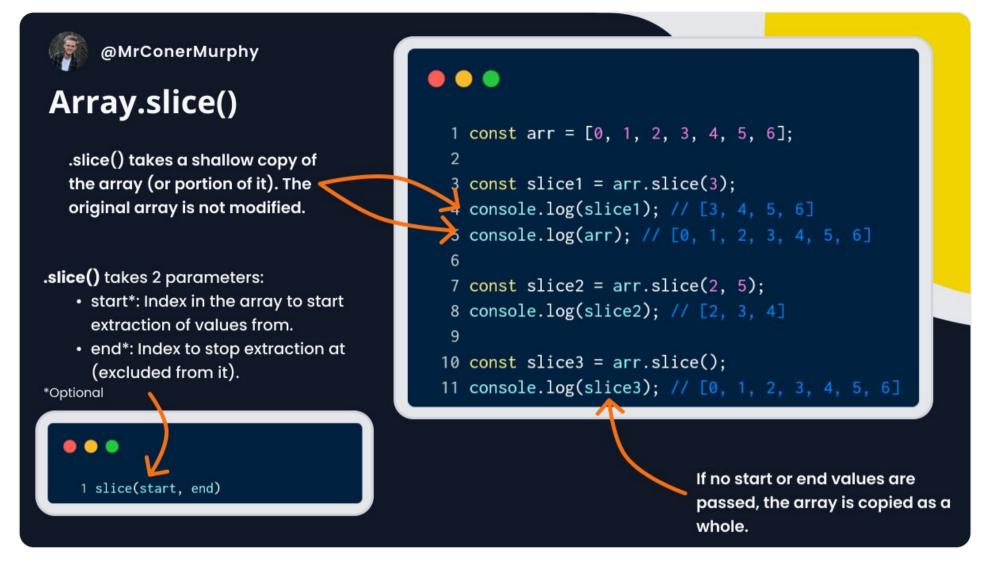
.shift() removes the first item from the array and returns it.



3 O Array .slice()

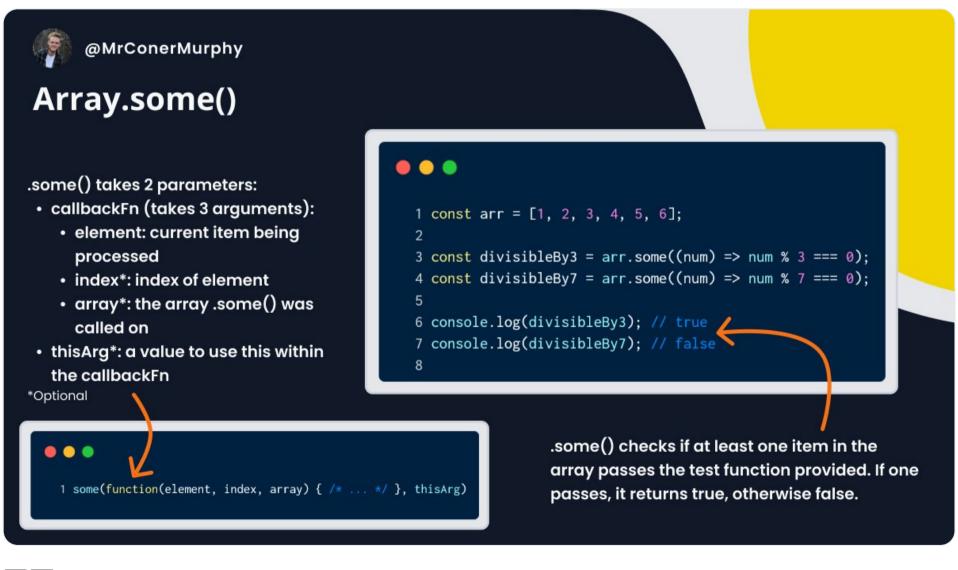
.slice() allows you to take a copy of an array either as a whole or part of it. The original array is left untouched.  $\P$ 

Follow Us on Twitter!



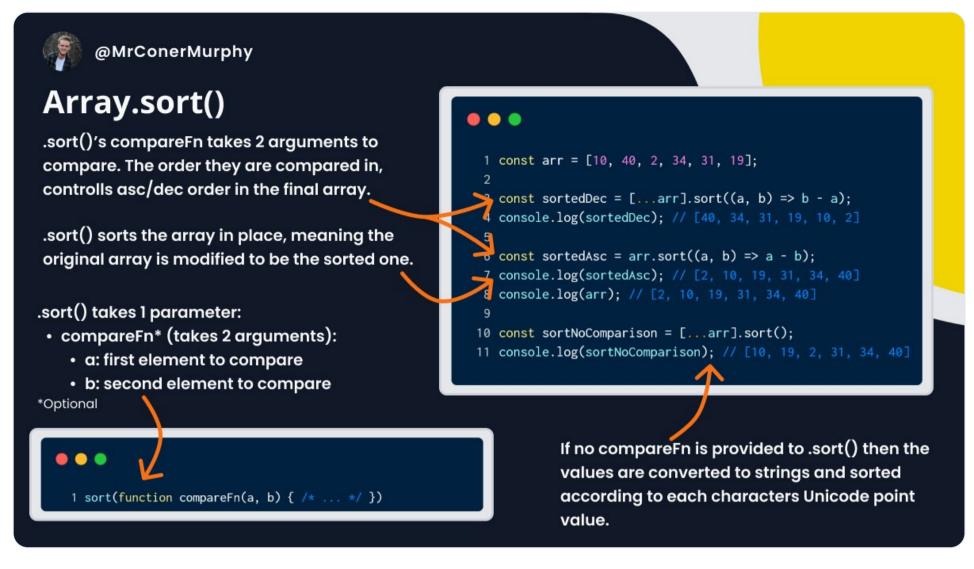
3 1 Array .some()

.some() checks if at least one item in an array passes the provided test function, if so it returns true otherwise false. \$\\ \\$



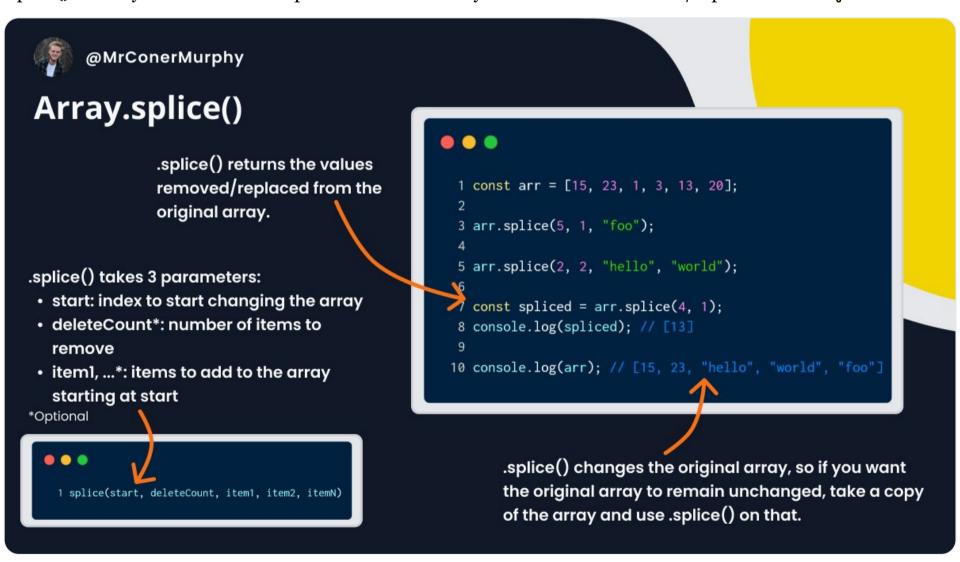
3 2 Array .sort()

.sort() allows you to sort an array based on a provided sorting function.



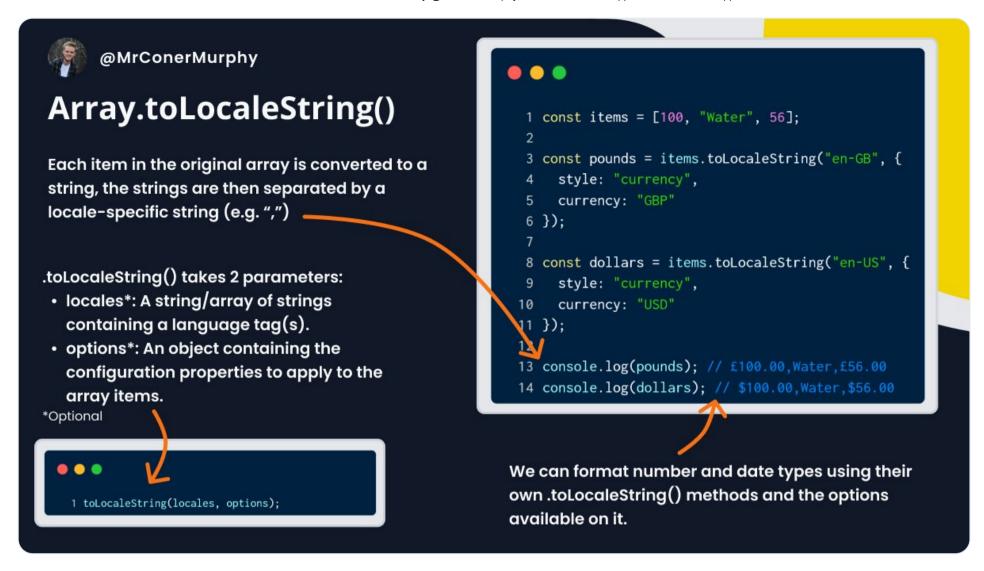
3 Array .splice()

.splice() allows you to remove or replace items in an array and return the removed/replaced items.



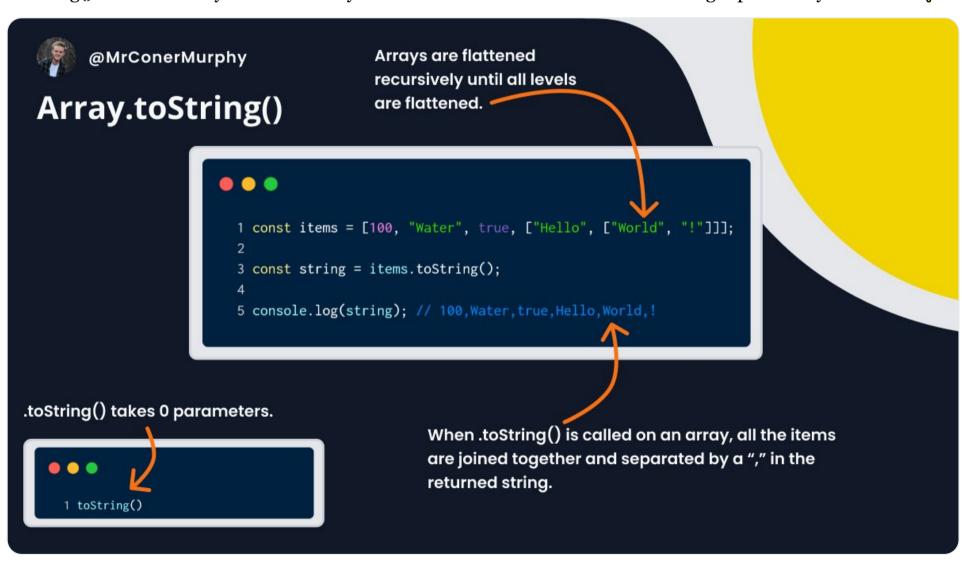
4 Array .toLocaleString()

.toLocaleString() will convert the array it's called on to a string and then separate the string with a locale-specific string (e.g. ',')



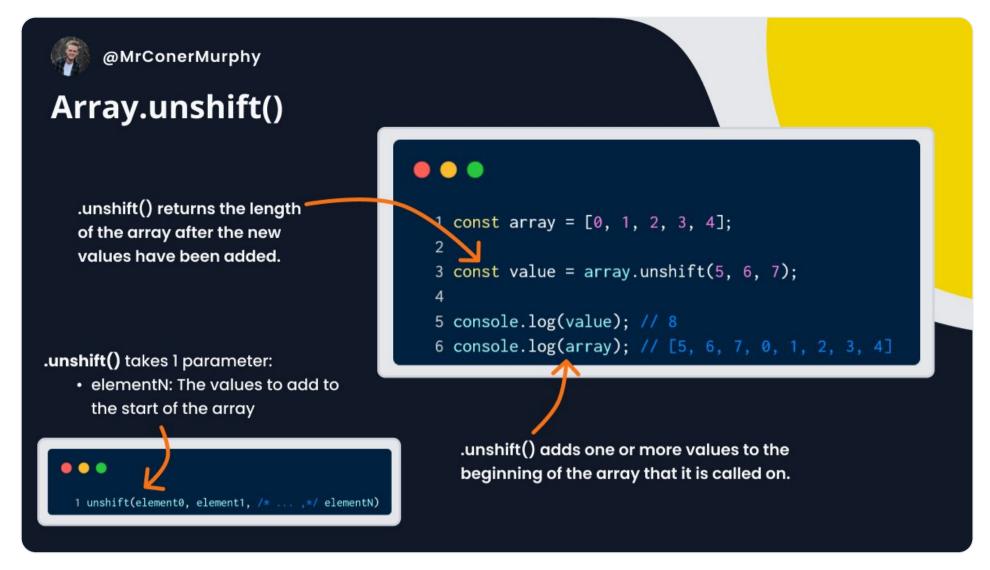
3 5 Array .toString()

.toString() will recursively flatten an array until it's one level and convert it to a string separated by a comma.



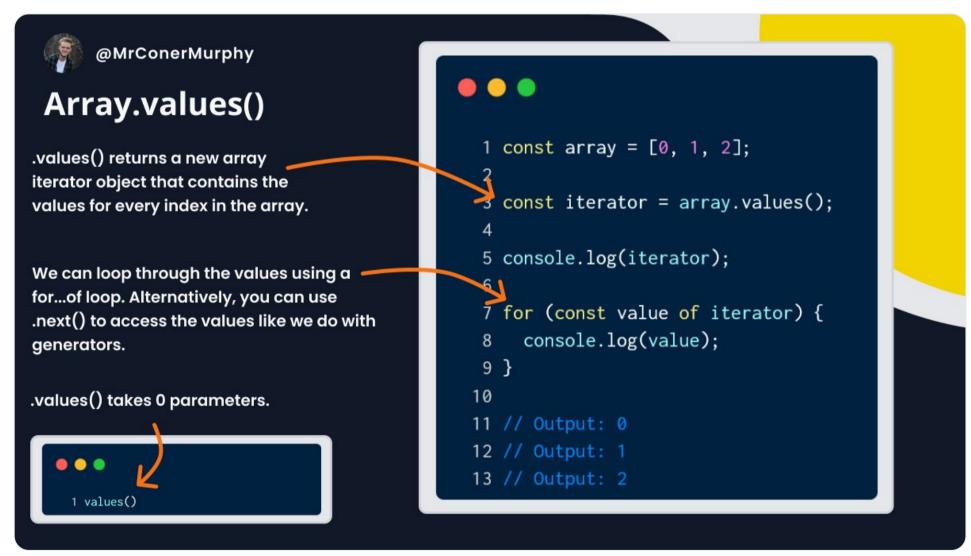
3 6 Array .unshift()

.unshift() adds one or more values provided to the beginning of the array and returns the updated length of the array.  $\blacksquare$ 



<sup>3</sup> Array .values()

.values() returns an iterator object containing a value for every index in the array which we can then loop over with a for...of loop.  $\P$ 



Did you enjoy this thread/find it helpful?

If so please consider doing one of the below to show your support:

- Like these tweets
- Retweet the 1st tweet in this thread so others can see it.
- Follow me ( <u>@MrConerMurphy</u> ) for more content like this.

