

Array Methods

- To solidify understanding of higher order functions and callbacks, we are going to look at some of the more popular iterative array methods provided for us by JavaScript.
- Some of the most popular are `forEach`, `map`, and `reduce`

array.forEach()

- The forEach method takes a callback and runs it once for each element in the array.
- forEach can also take an optional index argument to keep track of the index that you are currently working with
- forEach function does not mutate the array that it is being operated on, and does not return any value itself.

array.map()

- The map method takes a callback function and creates a new array by performing the callback on each array element.
- map can also take an optional index array.
- map does not mutate the original array. It instead returns a new array of the same length as the original array with the result of operating the callback function.

array.reduce()

- The reduce method takes a callback function and an iterator (which can be any data type) and runs the callback on each array element to reduce it to a single value.
- Your callback should take at least two arguments, which are regularly known as *previous* and *next*. These will be used to reduce each value in the array into the iterator, and as such your callback must return a value to be used on the next iteration.
- Reduce does not mutate the original array, but it does return a new value based on the callback function.