

1 `map()` Notes

You can use the `map()` method to transform each item in an array into something else.

Like `filter()` it returns a new array leaving the original array untouched. Unlike `filter()`, the new array is returned with the same number of array elements. In this case, each item in the new array will be mutated.

To transform an array's items with `map`, you use a callback function to return the data you want:

```
let strings = ['1', '2', '3'];  
  
const numbers = strings.map(string => parseInt(string, 10));  
  
[1, 2, 3]
```

Note: `parseInt()` takes a second argument, called a *ratex*. A *ratex* value of 10 means we want the returned numbers to be in base 10. Base 10 is basically the way we normally think of numbers. You can use different values for different kinds of numbers, like binary or octal numbers.