13 Useful Array Methods in Swift

Use Swift's built-in array methods to make your life easier and code prettier

An *array* is a commonly used data type in Swift. It is like a list in which you store items (for example numbers). You are going to use arrays a lot as an iOS engineer.

To help you master arrays and improve your code, I wrote this guide that shows you built-in array methods and examples.

1. Array.filter()

Instead of creating a loop, you can use the array's built-in `filter()` method to filter elements to a new array.

For instance, let's filter numbers lower than or equal to `3` to a new array:

Output:

[1,2,3]

2. Array.map()

The `map()` method takes each value of an array, performs some action on it, and finally places the new value into a new array. This is handy and can be used instead of a `for` or `while` loop.

For instance, let's square an array of numbers (into a new array):

Output:

[1, 4, 9, 16, 25]

3. Array.forEach()

The `forEach()` method is super handy. You can use it to replace regular `for`loops, which might be more suitable in some cases.

As an example, let's print out all the numbers of an array:

Output:

```
1
2
3
4
5
```

4. Array.reduce()

This is probably the "trickiest" method on the list. According to Apple's docs:

> "Returns the result of combining the elements of the sequence using the given closure."

You can, for example, use it to calculate the sum of an array without using `for` or `while` loops. It works like you'd calculate the sum in real life: Go through the list number by number and keep track of the sum while adding the numbers to it.

In Swift, reducing looks like this:

Output:

```
15
```

Let's go through the code in a bit more detail:

- `.reduce(0)` means that you start reducing (i.e. summing) from `0`, which is reasonable.
- Then `(sum, num) -> Int` means for each element in the list, you perform an action that adds two integers and returns an integer (i.e. add each number to the sum).
- `sum + num` is the action that adds the current number `num` to the `sum`, which is the sum so far.

5. Array.sort()

As the name suggests, you can use the sort() method to sort an array. For instance, let's sort the numbers in descending order:

Output:

[5, 4, 3, 2, 1]

The sort() method directly modifies the original array (*in-place sort*). To create a **new** array with the sorted values, you can use the sorted() method.

6. Array.first()

To get the first element of the array, you can access the array's first *property*(nope, it is not a method):

```
var nums = [1, 2, 3, 4, 5]
print(nums.first!)
```

Output:

1

But you are here to learn array methods — not properties — so let's examine the first() method. You can use the first() method to get the first value that satisfies a condition.

As an example, let's pick the first even number from an array of integers:

Output:

2

7. Array.firstIndexOf()

Similarly to the first() method, firstIndexOf() can be used to get the first *index* of a possible element that satisfies a condition.

For instance, let's get the *index* of the first even number:

Output:

The first even number is at index 1

8. Array.shuffle()

To randomly mix up an array, use the shuffle() method.

For instance:

Output example:

[2, 3, 4, 5, 1]

9. Array.reverse()

You can reverse the order of an array by using the reverse() method.

As an example:

Output:

[5, 4, 3, 2, 1]

10. Array.allSatisfy()

The allSatisfy() method is handy, yet you'd probably think it wouldn't exist as a built-in method. You can use allSatisfy() to easily check if *all* the elements of an array satisfy a condition.

For example, let's see if all the numbers are lower than 10:

Output:

true

11. Array.contains()

Use the contains() method to check if:

- A specific element exists in an array.
- There's an element that satisfies a condition.

For instance, you can check if the number 2 is in the list of numbers:

Output:

true

Or you can check if there are any even numbers in the list:

Output:

true

12. Array.isEmpty()

Use the isEmpty() method to see if an array is empty.

For example:

Output:

false

13. Array.randomElement()

Pick a random element from an array using the built-in randomElement()method. For instance:

Output:

5

Conclusion

Thanks for reading. I hope you learned something useful today.

Happy coding!