Swift Higher Order Function

Higher-order functions are functions that take other functions as parameters or return functions as their result. Swift provides several powerful higher-order functions that can be used to manipulate collections and perform common tasks more succinctly and expressively. Here are some of the most commonly used higher-order functions in Swift:

map

The **map** function applies a given function to each element in a collection and returns a new collection containing the results.

```
let numbers = [1, 2, 3, 4, 5]
let squaredNumbers = numbers.map { $0 * $0 }
print(squaredNumbers)  // Output: [1, 4, 9, 16, 25]
```

filter

The **filter** function returns a new collection containing only the elements that satisfy a given predicate.

```
let numbers = [1, 2, 3, 4, 5]
let evenNumbers = numbers.filter { $0 % 2 == 0 }
print(evenNumbers)  // Output: [2, 4]
```

reduce

The **reduce** function combines all elements of a collection into a single value using a given closure.

```
let numbers = [1, 2, 3, 4, 5]
let sum = numbers.reduce(0) { $0 + $1 }
print(sum)  // Output: 15
```

flatMap

The **flatMap** function applies a given transformation to each element in a collection and flattens the resulting collections into a single collection.

```
let nestedNumbers = [[1, 2, 3], [4, 5], [6]]
let flatNumbers = nestedNumbers.flatMap { $0 }
print(flatNumbers)  // Output: [1, 2, 3, 4, 5, 6]
```

compactMap

The **compactMap** function is similar to **map**, but it removes **nil** values from the resulting collection.

```
let strings = ["1", "2", "three", "4", "5"]
let numbers = strings.compactMap { Int($0) }
print(numbers)  // Output: [1, 2, 4, 5]
```

sorted

The **sorted** function returns the elements of the collection, sorted according to the given predicate.

```
let names = ["Chris", "Alex", "Ewa", "Barry", "Daniella"]
let sortedNames = names.sorted { $0 < $1 }
print(sortedNames)  // Output: ["Alex", "Barry", "Chris", "Daniella", "Ewa"]
```

forEach

The **forEach** function executes a given closure on each element of the collection.

```
let numbers = [1, 2, 3, 4, 5]
numbers.forEach { print($0) }
// Output:
// 1
// 2
// 3
// 4
// 5
```