Functional Nuggets

hackingwithswift.com

UPPERCASE AN ARRAY OF STRINGS

```
let uppercase = fruits.map {
$0.uppercased() }
```

ARRAY BY RUNNING A FUNCTION 10 TIMES

```
let webViews = (0 ..< 10).map { _
in createSomeView() }
```

CONVERT URL ARRAY TO STRING ARRAY

```
let filenames = urls.map {
$0.lastPathComponent }
```

CREATE ARRAY OF INDIVIDUAL STRING LETTERS

```
let letters = string.map {
String($0) }
```

FORMAT ARRAY OF INTEGERS AS STRINGS

```
let formattedScores = scores.map {
“Your score was \($0)” }
```

REMOVE NIL VALUES FROM ARRAY

```
let res = albums.compactMap { $0 }
```

CONVERT ARRAY OF STRINGS TO INTEGERS, IGNORING ANY INVALID NUMBERS

```
let integerScores = scores.
compactMap { Int($0) }
```

CONVERT ARRAY OF STRINGS TO URLS, IGNORING ANY INVALID URLS

```
let validURLs = strings.compactMap
{ URL(string: $0) }
```

FIND ALL TAPPED NODES THAT WERE BUILT FROM THE CLASS “BALL”

```
let balls = nodes(at: point).
compactMap { $0 as? Ball }
```

CONVERT ARRAY OF DICTIONARIES INTO SIMPLE ARRAY USING ONE KEY

```
let names = categories.compactMap
{ $0[“name”] }
```

MAKE ARRAY FROM ALL DICTIONARY KEYS WHERE VALUE IS TRUE

```
let trueResults = allResults.
compactMap { $1 == true ? $0 : nil }
```

CREATE A URL FROM A STRING, THEN DOWNLOAD IT IF THE URL WAS VALID

```
let contentsOfURL = URL(string:
filePath).compactMap {
  try? Data(contentsOf: $0)
}
```

CONVERT OPTIONAL STRING ARRAY INTO REGULAR STRINGS, ONLY WHERE NON-EMPTY

```
let flattened = strings.compactMap {
  $0.count > 0 ? $0 : nil
}
```

CONVERT AN ARRAY OF ARRAYS INTO A ONE-DIMENSIONAL ARRAY

```
let oneDimensional =
Array(twoDimensional.joined())
```

CONVERT TWO-DIMENSIONAL ARRAY TO A STRING, SEPARATED BY LINE BREAKS

```
let gridString = rows.reduce(“”) {
$0 + $1.joined(separator: “ ”) +
“\n” }
```

SUM AN ARRAY OF NUMBERS

```
let sum = numbers.reduce(0, +)
```

CHECK ARRAY OF NAMES ALL CONTAIN AT LEAST FOUR CHARACTERS

```
let namesAreLongEnough =
names.reduce(true) {
$0 && $1.count > 4 }
```

FILTER OUT ALL INVISIBLE FILENAMES (FILES THAT START WITH A PERIOD / FULL STOP)

```
let visible = allFilenames.filter {
!$0.hasPrefix(“.”) }
```

FILTER OUT ALL CITIES NOT IN EUROPE

```
let europeanCities = cities.filter {
  $0.continent == “Europe”
}
```

SORT AN ARRAY OF STRINGS LONGEST FIRST

```
let sortedByLength = names.sorted {
  $1.count < $0.count
}
```

PULL OUT THE FIRST THREE ITEMS IN AN ARRAY

```
let firstThreeCities =
cities.prefix(upTo: 3)
```

Thanks for being part of the Swift Frequent Flyer Club!