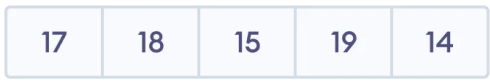
Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.

Swift Arrays

In this tutorial, you will learn about Swift arrays with the help of examples.

An array is a collection of similar types of data. For example,

Suppose we need to record the age of **5** students. Instead of creating **5** separate variables, we can simply create an array:



Array of Age

Elements of an array

Create a Swift Array

Here's how we create an array in Swift.

```
// an array of integer type
var numbers : [Int] = [2, 4, 6, 8]

print("Array: \(numbers)")
```

Output

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.

Swift is a type inference language that is, it can automatically identify the data type of an array based on its values. Hence, we can create arrays without specifying the data type. For example,

```
var numbers = [2, 4, 6, 8]
print("Array: \(numbers)")   // [2, 4, 6, 8]
```

Create an Empty Array

In Swift, we can also create an empty array. For example,

```
var value = [Int]()
print(value)
```

Output

```
[ ]
```

In the above example, `value` is an empty array that doesn't contain any element.

It is important to note that, while creating an empty array, we must specify the data type inside the square bracket `[]` followed by an initializer syntax `()`. Here, `[Int]()` specifies that the empty array can only store integer data elements.

> **Note:** In Swift, we can create arrays of any data type like `Int`, `String`, etc.

Access Array Elements

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.


```
var languages = ["Swift", "Java", "C++"]

// access element at index 0
print(languages[0])   // Swift

// access element at index 2
print(languages[2])   // C++
```

In the above example, we have created an array named `languages`.

Array indexing in Swift

Here, we can see each array element is associated with the index number. And, we have used the index number to access the elements.

> **Note:** The array index always starts with **0**. Hence, the first element of an array is present at index **0**, not **1**.

Add Elements to an Array

Swift Array provides different methods to add elements to an array.

1. Using append()

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.


```
// using append method
numbers.append(32)

print("After Append: \(numbers)")
```

Output

```
Before Append: [21, 34, 54, 12]
After Append: [21, 43, 54, 12, 32]
```

In the above example, we have created an array named `numbers`. Notice the line,

```
numbers.append(32)
```

Here, `append()` adds **32** at the end of the array.

We can also use the `append()` method to add all elements of one array to another. For example,

```
var primeNumbers = [2, 3, 5]
print("Array1: \(primeNumbers)")

var evenNumbers = [4, 6, 8]
print("Array2: \(evenNumbers)")

// join two arrays
primeNumbers.append(contentsOf: evenNumbers)

print("Array after append: \(primeNumbers)")
```

Output

```
Array1: [2, 3, 5]
Array2: [4, 6, 8]
Array after append: [2, 3, 5, 4, 6, 8]
```

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.

ADVERTISEMENTS

Here, we are adding all elements of `evenNumbers` to `primeNumbers`.

> **Note:** We must use `contentOf` with `append()` if we want to add all elements from one array to another.

2. Using insert()

The `insert()` method is used to add elements at the specified position of an array. For example,

```
var numbers = [21, 34, 54, 12]

print("Before Insert: \(numbers)")

numbers.insert(32, at: 1)

print("After insert: \(numbers)")
```

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.

Here, `numbers.insert(32, at:1)` adds **32** at the **index 1**.

Modify the Elements of an Array

We can use the array index to modify the array element. For example,

```
var dailyActivities = ["Eat","Repeat"]
print("Initial Array: \(dailyActivities)")

// change element at index 1
dailyActivities[1] = "Sleep"

print("Updated Array:  \(dailyActivities)")
```

Output

```
Initial Array: ["Eat", "Repeat"]
Updated Array:  ["Eat", "Sleep"]
```

Here, initially the value at **index 1** is `Repeat`. We then changed the value to `Sleep` using

```
dailyActivities[1] = "Sleep"
```

Remove an Element from an Array

We can use the `remove()` method to remove the last element from an array. For example,

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.


```
let removedValue = languages.remove(at: 1)

print("Updated Array: \(languages)")
print("Removed value: \(removedValue)")
```

Output

```
Initial Array: ["Swift", "Java", "Python"]
Updated Array: ["Swift", "Python"]
Removed value: Java
```

Similarly, we can also use

- `removeFirst()` - to remove the first element
- `removeLast()` - to remove the last element
- `removeAll()` - to remove all elements of an array

Other Array Methods

Method	Description
`sort()`	sorts array elements
`shuffle()`	changes the order of array elements
`forEach()`	calls a function for each element
`contains()`	searches for the element in an array
`swapAt()`	exchanges the position of array elements
`reverse()`	reverses the order of array elements

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.


```
// an array of fruits
let fruits = ["Apple", "Peach", "Mango"]

// for loop to iterate over array
for fruit in fruits {
  print(fruit)
}
```

Output

```
Apple
Peach
Mango
```

Find Number of Array Elements

We can use the `count` property to find the number of elements present in an array. For example,

```
let evenNumbers = [2,4,6,8]
print("Array: \(evenNumbers)")

// find number of elements
print("Total Elements: \(evenNumbers.count)")
```

Output

```
Array: [2, 4, 6, 8]
Total Elements: 4
```

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.


```
let numbers = [21, 33, 59, 17]
print("Numbers: \(numbers)")

// check if numbers is empty
var  result = numbers.isEmpty
print("Is numbers empty? : \(result)")

// array without elements
let evenNumbers = [Int]()
print("Even Numbers: \(evenNumbers)")

// check if evenNumbers is empty
result = evenNumbers.isEmpty
print("Is evenNumbers empty? : \(result)")
```

Output

```
Numbers: [21, 33, 59, 17]
Is numbers empty? : false
Even Numbers: []
Is evenNumbers empty? : true
```

In the above example, we have used `isEmpty` property to check if arrays `numbers` and `evenNumbers` are empty. Here, `isEmpty` returns

- `true` - if the array is empty
- `false` - if the array is not empty

Array With Mixed Data Types

Till now, we have been using arrays that hold elements of a single data type.

However, in Swift, we can also create arrays that can hold elements of multiple data types. For example,

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.

Output

```
["Scranton", 570]
```

In the above example, we have created an array named `address`.

```
var address: [Any] = ["Scranton", 570]
```

Here, `[Any]` specifies that `address` can hold elements of any data type. In this case, it stores both `String` and `Integer` data.

Next Tutorial:
Swift Sets (/swift-programming/sets)

Previous Tutorial:
Swift guard Statement (/swift-programming/guard-statement)

Share on:

(https://www.facebook.com/sharer/sharer.php?u=https://www.programiz.com/swift-programming/arrays)

(https://twitter.com/intent/tweet?text=Check%20this%20amazing%20programming/arrays)

Did you find this article helpful?

😀 ☹️

ADVERTISEMENTS

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.

Related Tutorials

Swift Tutorial

Swift Sets

(/swift-programming/sets)

Swift Tutorial

Swift Dictionary

(/swift-programming/dictionary)

Swift Tutorial

Swift Ranges

(/swift-programming/ranges)

Swift Tutorial

Swift Tuple

(/swift-programming/tuples)

Thank you for printing our content at www.domain-name.com. Please check back soon for new contents.

(/)

Search tutorials and examples

www.domain-name.com