Array_Single

January 2, 2025

[1]:
```
// Creating arrays of different types
val intArray: Array[Int] = Array(5, 10, 15, 20)
val stringArray = Array("apple", "banana", "cherry")

// Iterating through arrays using for loop
for (item <- intArray) {
  print(s"$item ")
}
println("")
for (fruit <- stringArray) {
  print(s"$fruit ")
}
```

```
5 10 15 20
apple banana cherry
```

```
intArray = Array(5, 10, 15, 20)
stringArray = Array(apple, banana, cherry)
```

[1]:
```
Array(apple, banana, cherry)
```

[2]:
```
// Array initialized with fill
val zeros: Array[Int] = Array.fill(3)(0)
val ones: Array[Int] = Array.fill(4)(1)

// Printing elements
zeros.foreach(num => print(s"$num "))
println("")
ones.foreach(num => print(s"$num "))
```

```
0 0 0
1 1 1 1
```

```
zeros = Array(0, 0, 0)
ones = Array(1, 1, 1, 1)
```

[2]:
```
Array(1, 1, 1, 1)
```

[3]:
```
// Accessing elements in an array
val values: Array[Double] = Array(2.5, 4.5, 6.5, 8.5)
val secondValue = values(1)

println(s"Array: ${values.mkString(", ")}")
println(s"Second element: $secondValue")
```

```
Array: 2.5, 4.5, 6.5, 8.5
Second element: 4.5
```

```
values = Array(2.5, 4.5, 6.5, 8.5)
secondValue = 4.5
```

[3]:
```
4.5
```

[4]:
```
// Modifying array elements
val nums: Array[Int] = Array(1, 2, 3, 4)
nums(2) = 10 // Change the third element

println("Modified Array:")
nums.foreach(num => print(s"$num "))
```

```
Modified Array:
1 2 10 4
```

```
nums = Array(1, 2, 10, 4)
```

[4]:
```
Array(1, 2, 10, 4)
```

[5]:
```
// Array properties
val nums: Array[Int] = Array(5, 10, 15, 20)
val arrayLength = nums.length

println(s"Array: ${nums.mkString(", ")}")
println(s"Array Length: $arrayLength")
```

```
Array: 5, 10, 15, 20
Array Length: 4
```

```
nums = Array(5, 10, 15, 20)
arrayLength = 4
```

[5]:
```
4
```

[6]:
```
// Using map and filter
val numbers = Array(2, 4, 6, 8)
val tripled = numbers.map(_ * 3)
val odds = numbers.filter(_ % 2 != 0)
```

```
println(s"Tripled: ${tripled.mkString(", ")}")
println(s"Odds: ${odds.mkString(", ")}")
```

```
Tripled: 6, 12, 18, 24
Odds:
```

```
numbers = Array(2, 4, 6, 8)
tripled = Array(6, 12, 18, 24)
odds = Array()
```

[6]:
```
Array()
```

[7]:
```
// Aggregating values
val scores = Array(10, 20, 30, 40)
val total = scores.reduce(_ + _)
val maxScore = scores.max
val minScore = scores.min

println(s"Scores: ${scores.mkString(", ")}")
println(s"Total: $total, Max: $maxScore, Min: $minScore")
```

```
Scores: 10, 20, 30, 40
Total: 100, Max: 40, Min: 10
```

```
scores = Array(10, 20, 30, 40)
total = 100
maxScore = 40
minScore = 10
```

[7]:
```
10
```

[8]:
```
// Array transformations with map
val tempsInCelsius = Array(0, 10, 20, 30)
val tempsInFahrenheit = tempsInCelsius.map(c => c * 9 / 5 + 32)

println(s"Temps in Celsius: ${tempsInCelsius.mkString(", ")}")
println(s"Temps in Fahrenheit: ${tempsInFahrenheit.mkString(", ")}")
```

```
Temps in Celsius: 0, 10, 20, 30
Temps in Fahrenheit: 32, 50, 68, 86
```

```
tempsInCelsius = Array(0, 10, 20, 30)
tempsInFahrenheit = Array(32, 50, 68, 86)
```

[8]:
```
Array(32, 50, 68, 86)
```

[ ]: