

Iteratives

January 2, 2025

```
[1]: // Basic for loop
for (i <- 1 to 5) {
  println(s"Iteration: $i")
}
```

Iteration: 1
Iteration: 2
Iteration: 3
Iteration: 4
Iteration: 5

```
[2]: // for loop using `until`
for (i <- 1 until 5) {
  println(s"Iteration: $i")
}
```

Iteration: 1
Iteration: 2
Iteration: 3
Iteration: 4

```
[3]: // Nested for loop
for (i <- 1 to 2; j <- 1 to 3) {
  println(s"i: $i, j: $j")
}
```

i: 1, j: 1
i: 1, j: 2
i: 1, j: 3
i: 2, j: 1
i: 2, j: 2
i: 2, j: 3

```
[4]: // for loop with a condition
for (i <- 1 to 10 if i % 2 == 0) {
  println(s"Even number: $i")
}
```

Even number: 2
Even number: 4

Even number: 6
Even number: 8
Even number: 10

```
[5]: // Basic while loop  
var count = 1  
while (count <= 5) {  
    println(s"Count: $count")  
    count += 1  
}
```

Count: 1
Count: 2
Count: 3
Count: 4
Count: 5

count = 6

[5]: 6

```
[6]: // do-while loop  
var num = 1  
do {  
    println(s"Number: $num")  
    num += 1  
} while (num <= 5)
```

Number: 1
Number: 2
Number: 3
Number: 4
Number: 5

num = 6

[6]: 6

```
[7]: // Iterating over a collection using foreach  
val colors = List("Red", "Blue", "Green")  
colors.foreach(color => println(s"Color: $color"))
```

Color: Red
Color: Blue
Color: Green

colors = List(Red, Blue, Green)

[7]: List(Red, Blue, Green)

```
[8]: // Using map to transform a collection
val numbers = List(1, 2, 3, 4)
val squared = numbers.map(num => num * num)
println(s"Squared numbers: $squared")
```

Squared numbers: List(1, 4, 9, 16)

numbers = List(1, 2, 3, 4)

squared = List(1, 4, 9, 16)

[8]: List(1, 4, 9, 16)

```
[9]: // for comprehension to generate a new collection
val doubledNumbers = for (num <- numbers) yield num * 2
println(s"Doubled numbers: $doubledNumbers")
```

Doubled numbers: List(2, 4, 6, 8)

doubledNumbers = List(2, 4, 6, 8)

[9]: List(2, 4, 6, 8)

[]: