

# Creating a Range of Numbers

In this lesson, you will learn how to create a range of numbers using `Range`.

## We'll cover the following

- Introduction
- Creating a Range Using to
- Specifying an Interval Using by
- Creating a Range Using until

## Introduction #

In the previous lessons, we used the `range` method to populate a collection. A **Range** is also a collection in itself. It is an ordered sequence of integers with the integers being equally spread apart from one another by an equal interval.

## Creating a Range Using `to` #

We can create a Range by fixing `to` between the lower limit and upper limit of a Range.

`lowerLimit to upperLimit`

This code requires the following environment variables to execute:

LANG C.UTF-8

```
val myFirstRange = 1 to 10

// Driver Code
myFirstRange.foreach(println)
```



The code snippet above creates a range of numbers starting from 1 all the way up to 10

## Specifying an Interval Using `by` `#`

We can specify the interval between each element using `by`.

`lowerLimit to upperLimit by interval`

This code requires the following environment variables to execute:

LANG C.UTF-8

```
val oddRange = 1 to 10 by 2
```

```
// Driver Code  
oddRange.foreach(println)
```



The code snippet above creates a range of numbers starting from **1** all the way to **10** with an interval of **2** between two numbers. In other words, this would give us all the odd integers from **1** to **10**.

## Creating a Range Using `until` `#`

When we want to exclude the upper limit in a Range, we can use `until` fixed between the lower limit and the excluded upper limit.

`lowerLimit until upperLimit`

This code requires the following environment variables to execute:

LANG C.UTF-8

```
val notLast = 1 until 3
```

```
// Driver Code  
notLast.foreach(println)
```



The code snippet above creates a range of numbers starting from **1** until **3**. In other

words, it's a range of numbers starting from **1** and ending at **2**.

---

In the next lesson, we will discuss Streams; the last collection covered in this course.