for Loops

In this lesson, we will learn another type of loop: the for loop.

Syntax of a `for` loop

The syntax for `for` loop in R language:

```
for(value in vector)
{
   statements
}
```

Let's begin with printing every value in a vector:

```
myVector <- c(1, 2+2i, "3", 4, 5+5i, "6")

for (v in myVector)
{
  print(v)
}
```
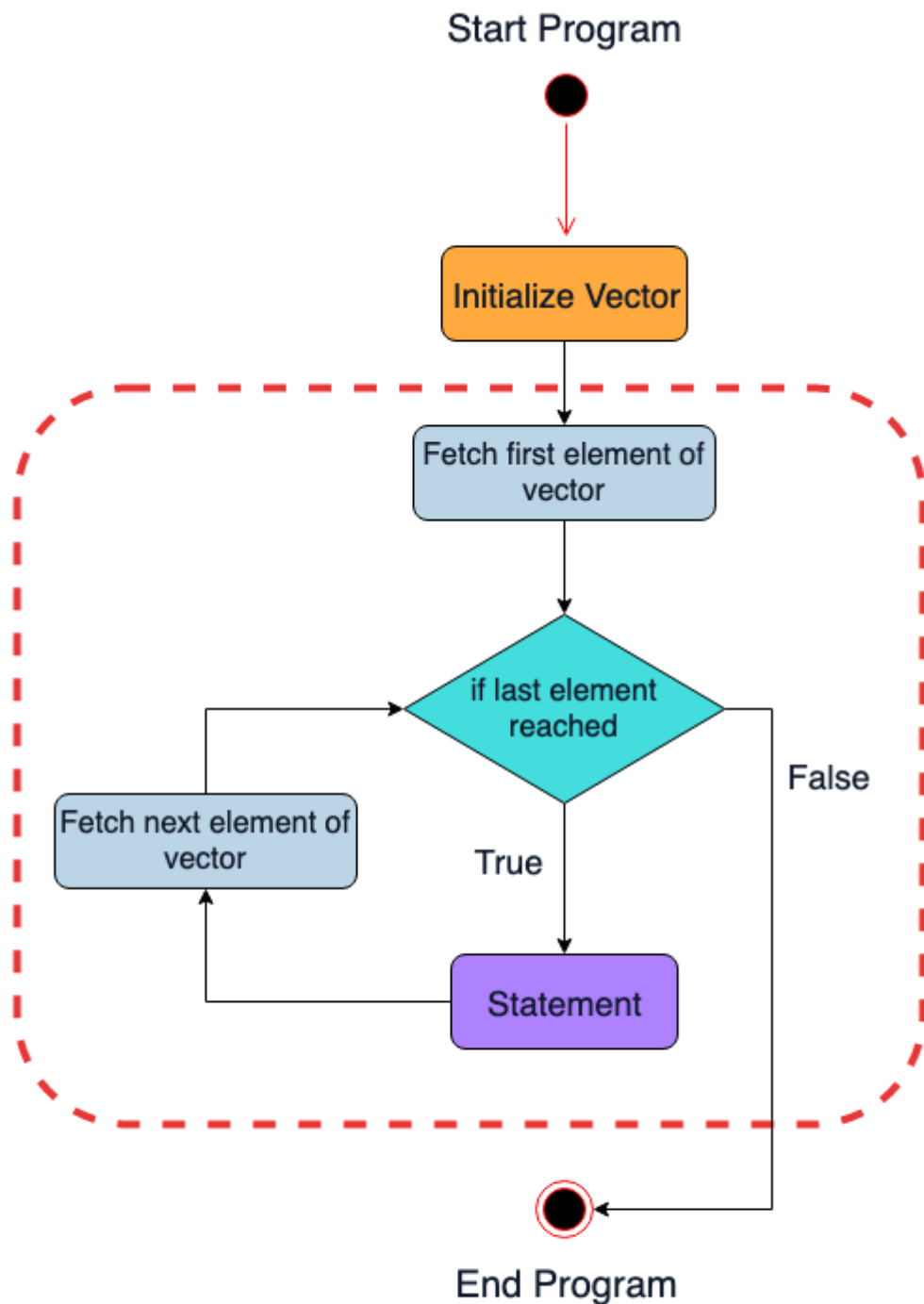
Basic for loop syntax

In the above code, the statement `print(v)` is executed for every element v in the vector `myVector`.

A `for` loop is used to apply the same **statements** or **function calls** to a collection of objects.

Let's visualize the code flow of `for` loop:

Start Program

Illustration of a for loop

Key:

Statements executed/handled by the for loop on its own

We can also use `for` loop on **lists** the same way we do on **vectors**. Furthermore, `for` loops can be applied to **matrices**, as the following example demonstrates. This will also give us an idea of **nested** `for` **loops**.

```
myMatrix <- matrix(c(1:12), nrow = 4, byrow = TRUE)

for (r in 1:nrow(myMatrix)) {
  for (c in 1:ncol(myMatrix))
```

```
        print(paste("Row", r, "and column",c, " = ", myMatrix[r, c]))
}
```

Here, the first loop

```
for (r in 1:nrow(myMatrix))
```

keeps track of the row index and the second nested `for` loop

```
for (c in 1:ncol(myMatrix))
```

keeps track of the column index.

The nested `for` loop allows us to iterate over the complete matrix one element at a time.

In the next lesson, we have a quick exercise for you.