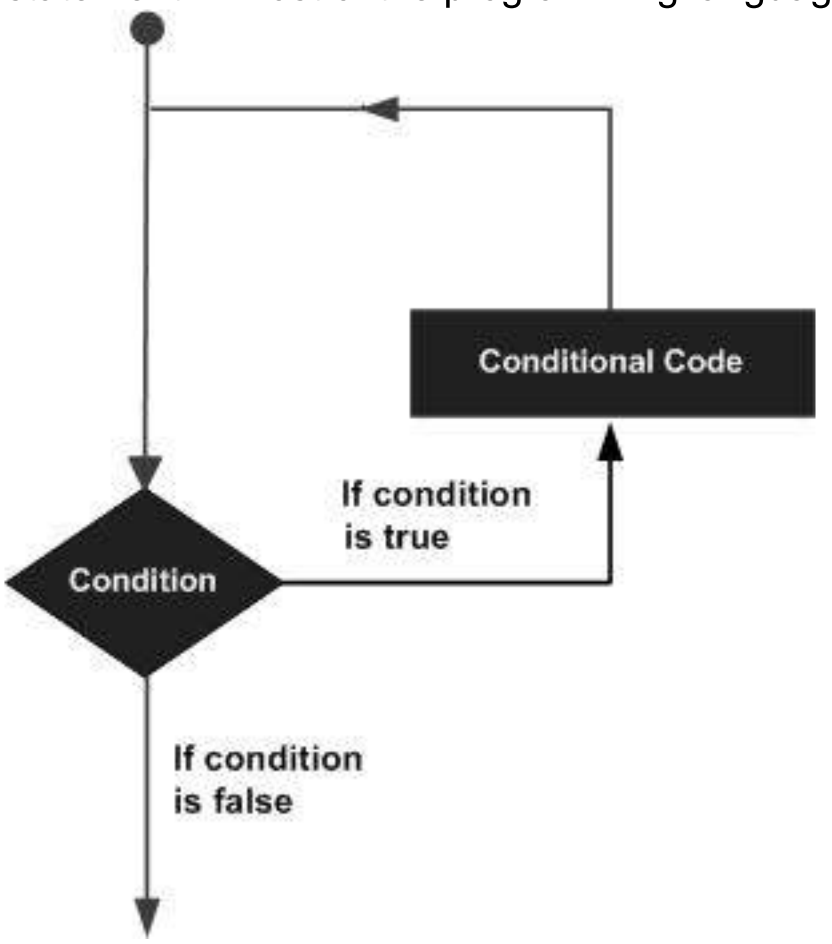
R - Loops

There may be a situation when you need to execute a block of code several number of times. In general, statements are executed sequentially. The first statement in a function is executed first, followed by the second, and so on.

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times and the following is the general form of a loop statement in most of the programming languages −



R programming language provides the following kinds of loop to handle looping requirements. Click the following links to check their detail.

Sr. No.	Loop Type & Description
1	repeat loop Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.

2	while loop Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
3	for loop Like a while statement, except that it tests the condition at the end of the loop body.

Loop Control Statements

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

R supports the following control statements. Click the following links to check their detail.

Sr. No.	**Control Statement & Description**
1	break statement Terminates the **loop** statement and transfers execution to the statement immediately following the loop.
2	Next statement The **next** statement simulates the behavior of R switch.

R - Repeat Loop

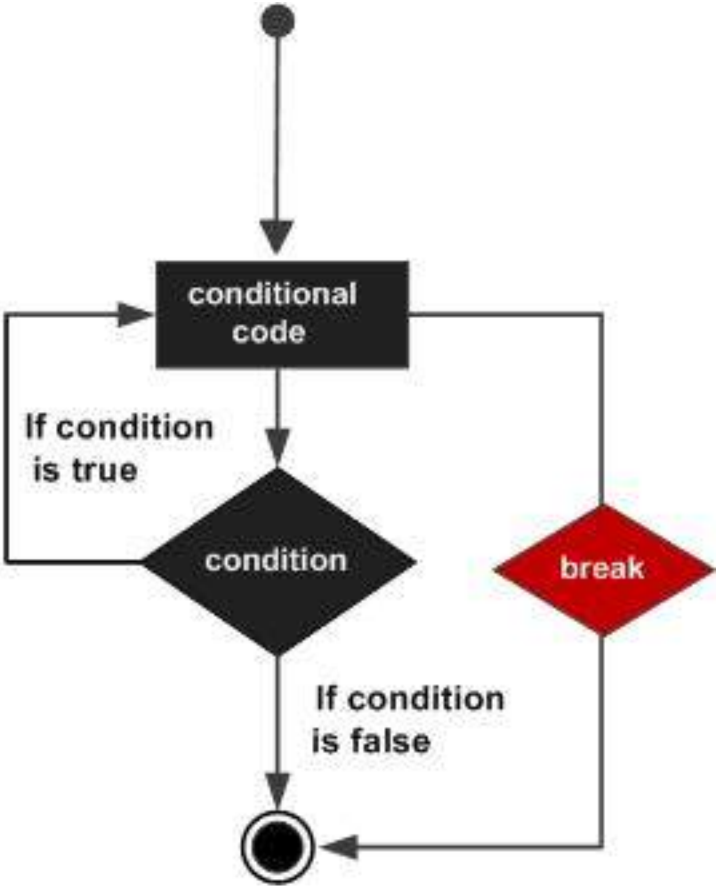
The **Repeat loop** executes the same code again and again until a stop condition is met.

Syntax

The basic syntax for creating a repeat loop in R is −

```
repeat { 
   commands 
   if(condition) {
      break
   }
}
```

Flow Diagram



Example

```
v <- c("Programming","Fiesta")
cnt <- 2

repeat {
   print(v)
   cnt <- cnt+1

   if(cnt > 5) {
      break
   }
}
```

When the above code is compiled and executed, it produces the following result −

```
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
```

R - While Loop

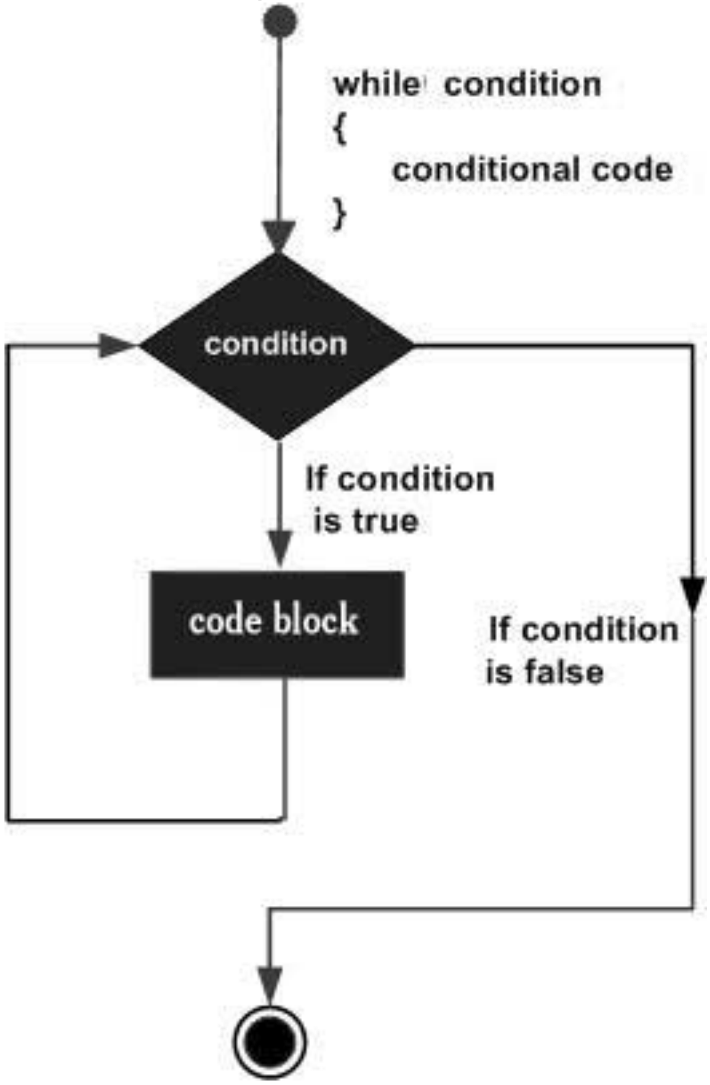
The While loop executes the same code again and again until a stop condition is met.

Syntax

The basic syntax for creating a while loop in R is −

```
while (test_expression) {
   statement
}
```

Flow Diagram



Here key point of the **while** loop is that the loop might not ever run. When the condition is tested and the result is false, the loop body will be skipped and the first statement after the while loop will be executed.

Example

```
v <- c("Programming","Fiesta")
cnt <- 2

while (cnt < 7) {
   print(v)
   cnt = cnt + 1
}
```

When the above code is compiled and executed, it produces the following result −

```
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta”
```

R - For Loop

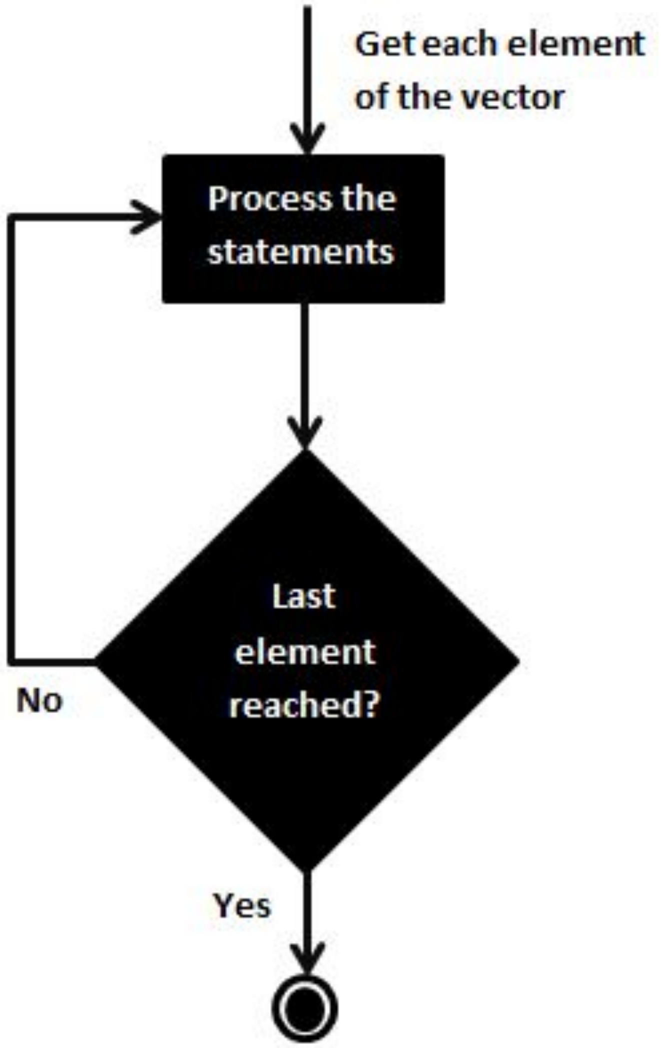
A **For loop** is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax

The basic syntax for creating a **for** loop statement in R is −

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

Flow Diagram



R’s for loops are particularly flexible in that they are not limited to integers, or even numbers in the input. We can pass character vectors, logical vectors, lists or expressions.

Example

```
v <- LETTERS[1:4]
for ( i in v) {
   print(i)
}
```

When the above code is compiled and executed, it produces the following result −

```
[1] "A"
[1] "B"
[1] "C"
[1] “D"
```

R - Break Statement

The break statement in R programming language has the following two usages −

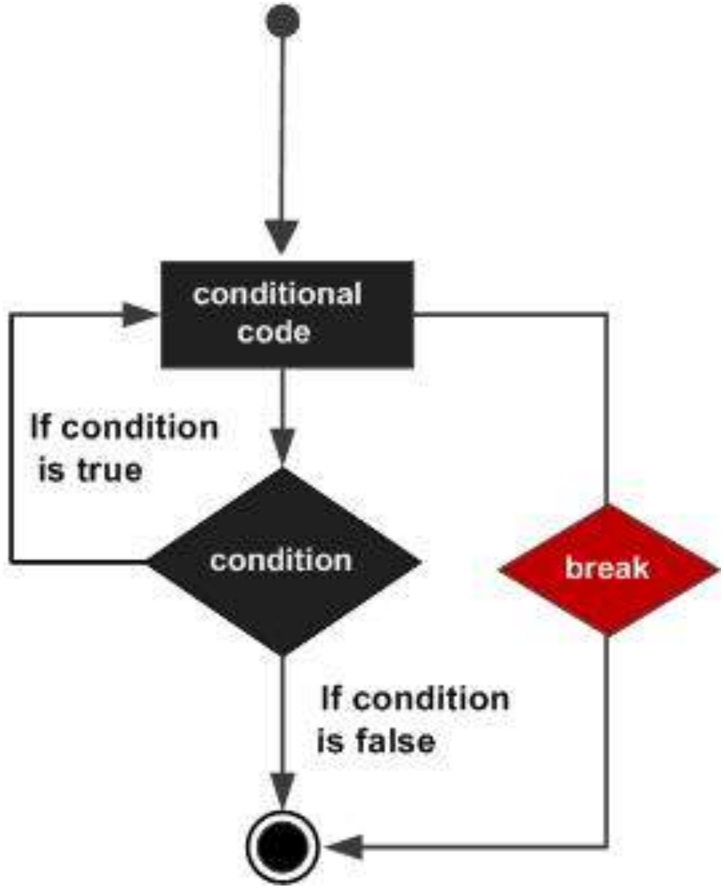
- When the break statement is encountered inside a loop, the loop is immediately terminated and program control resumes at the next statement following the loop.
- It can be used to terminate a case in the switch statement (covered in the next chapter).

Syntax

The basic syntax for creating a break statement in R is −

```
break
```

Flow Diagram



Example

```
v <- c("Programming","Fiesta")
cnt <- 2

repeat {
   print(v)
   cnt <- cnt + 1
   if(cnt > 5) {
      break
   }
}
```

When the above code is compiled and executed, it produces the following result −

```
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta"
[1] “Programming" “Fiesta”
```

R - Next Statement

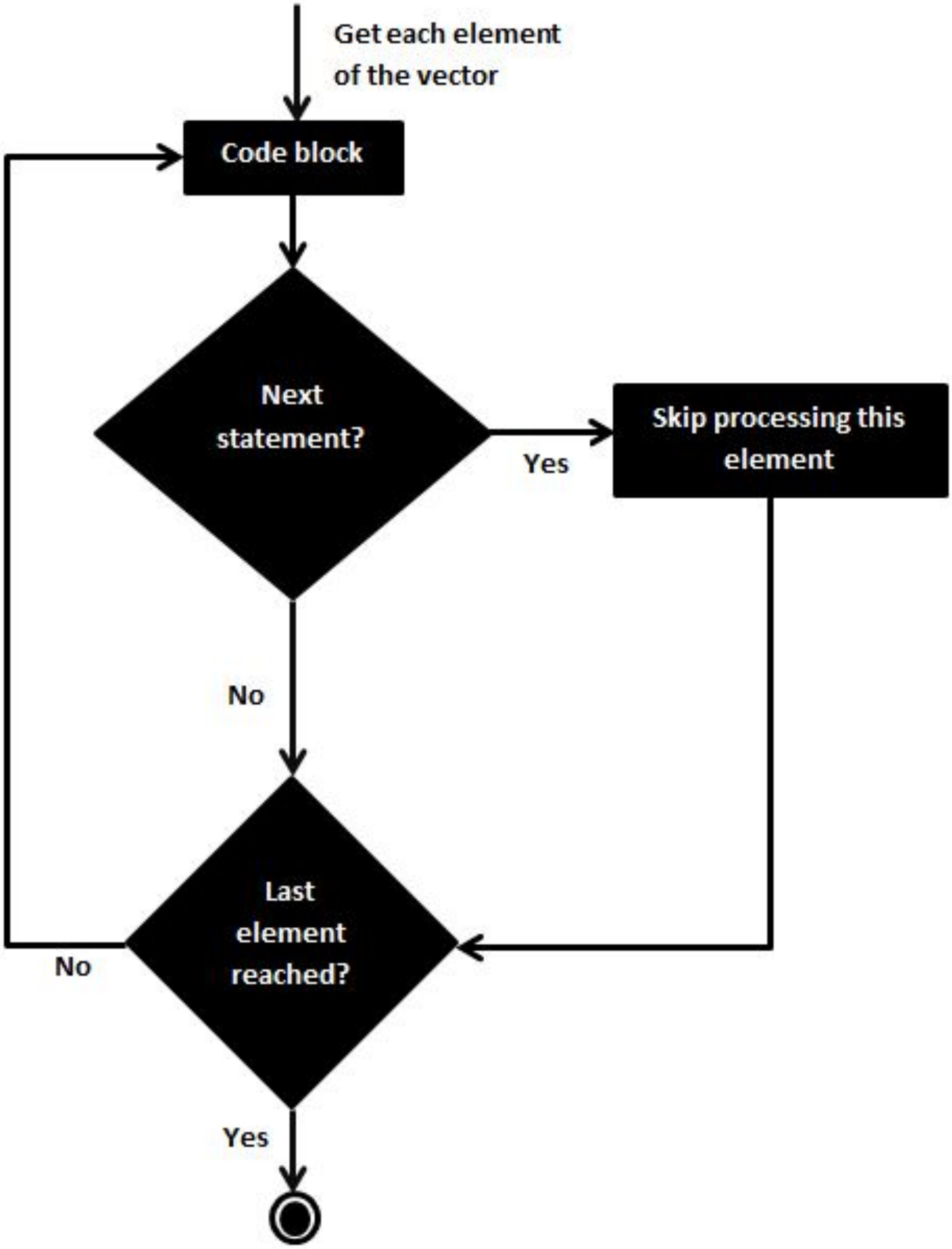
The **next** statement in R programming language is useful when we want to skip the current iteration of a loop without terminating it. On encountering next, the R parser skips further evaluation and starts next iteration of the loop.

Syntax

The basic syntax for creating a next statement in R is −

```
next
```

Flow Diagram



Example

```
v <- LETTERS[1:6
for ( i in v) {
   if (i == "D") {
      next
   }
   print(i)
}
```

When the above code is compiled and executed, it produces the following result −

```
[1] "A"
[1] "B"
[1] "C"
[1] "E"
[1] "F"
```