

R break and next (With Syntax and Examples)



R break and next (With Syntax and Examples)

In this article, you'll learn about break and next statements in R programming. You'll learn their syntax and how they work with the help of examples.

In R programming, a normal looping sequence can be altered using the `break` or the `next` statement.

break statement

A `break` statement is used inside a loop (repeat, for, while) to stop the iterations and flow the control outside of the loop.

In a nested looping situation, where there is a loop inside another loop, this statement exits from the innermost loop that is being evaluated.

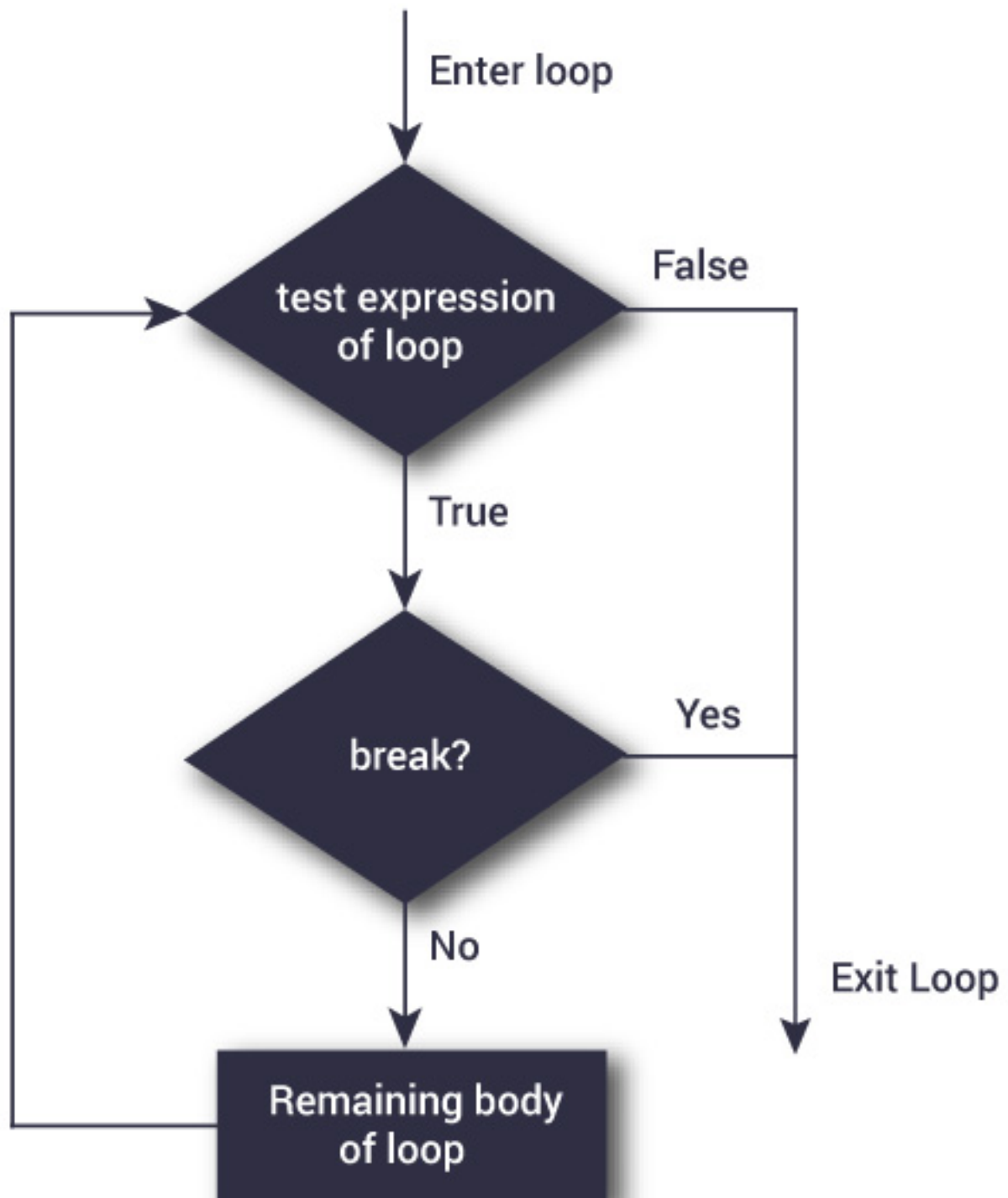
The syntax of break statement is:

```
if (test_expression) {  
  break  
}
```

Note: the break statement can also be used inside the `else` branch of `if...else` statement.

R break and next (With Syntax and Examples)

Flowchart of break statement



R break and next (With Syntax and Examples)

Example 1: break statement

```
x <- 1:5
for (val in x) {
  if (val == 3){
    break
  }
  print(val)
}
```

Output

```
[1] 1
[1] 2
```

In this example, we iterate over the vector `x`, which has consecutive numbers from 1 to 5.

Inside the for loop we have used a if condition to break if the current value is equal to 3.

As we can see from the output, the loop terminates when it encounters the `break` statement.

next statement

A next statement 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.

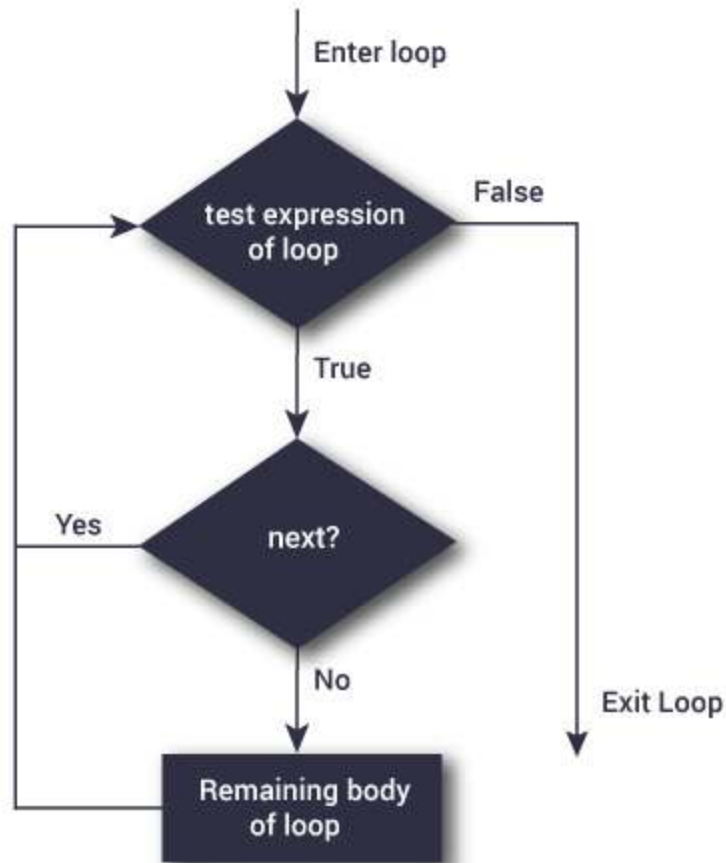
The syntax of next statement is:

```
if (test_condition) {
  next
}
```

Note: the next statement can also be used inside the `else` branch of `if...else` statement.

R break and next (With Syntax and Examples)

Flowchart of next statement



R break and next (With Syntax and Examples)

Example 2: Next statement

```
x <- 1:5
for (val in x) {
  if (val == 3){
    next
  }
  print(val)
}
```

Output

```
[1] 1
[1] 2
[1] 4
[1] 5
```

In the above example, we use the `next` statement inside a condition to check if the value is equal to 3.

If the value is equal to 3, the current evaluation stops (value is not printed) but the loop continues with the next iteration.

The output reflects this situation.