

# What are the uses and differences between the “break” and “next” (continue) statements in R?

Authored by  
**stats writer**

June 23, 2024

## RECOMMENDED CITATION

stats writer (2024). *What are the uses and differences between the “break” and “next” (continue) statements in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=149194>

The "break" and "next" statements are control flow statements used in the R programming language. These statements are used to alter the normal flow of a loop, allowing for more efficient and dynamic execution of code. The "break" statement is used to terminate a loop prematurely, while the "next" statement is used to skip over a specific iteration of a loop.

The "break" statement is commonly used when a certain condition is met and the loop needs to be stopped. This can save processing time and prevent unnecessary iterations. On the other hand, the "next" statement is useful when specific iterations of a loop need to be skipped, but the loop should continue until completion. This can be helpful in cases where certain data values need to be excluded from the loop.

In summary, the "break" statement is used to stop a loop completely, while the "next" statement is used to skip over specific iterations within the loop. Both statements have different uses and can be effectively used to control the flow of code and improve efficiency in R programming.

The Break and Next are jump statements in R that are used in control statements to terminate the loop at a particular iteration or to skip a particular iteration.

The looping in R is used to repeatedly execute a set of statements or block of code a specified number of times or until a specified condition is satisfied. There are three types of loops in R Programming, the `for` loop, the `while` loop and the `repeat` loop. You can use these jump statements with any of these loop statements.

Let's understand the jump statements with examples.

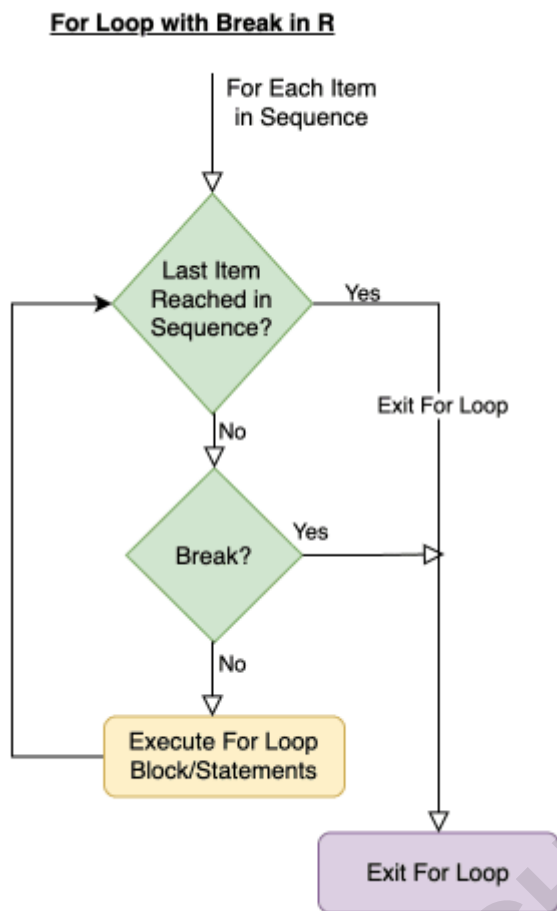
The break statementThe next statement (similar to continue)

In simple words, the `break` is used within the scope of the control statements to stop and exit the iteration without looping through all the iterations. And the `next` is similar to `continue` from other programming languages that is used to skip the current iterator and jump to the next iteration.

The `next` is similar to `continue` from other programming languages that is used to skip the rest of the iterator and jump to the next item in the sequence.

## 1. Break Statement in R

The break statement in R is used to exit or terminate the loop and the control goes to the very next statement after the loop. When a break statement is used in the nested loops, it exits from the innermost loop, and control transfers to the outer loop.



## 1.1 Using Break Statement with For Loop

```
# Using break statement
numbers <- c('One', 'Two', "Three", "Four", "Five")
for(i in numbers) {
  if( i == "Four")
  break
  print(i)
}
```

Yields below output.

```
[1] "One"
[1] "Two"
[1] "Three"
```

## 1.2 Using Break Statement in Nested For Loop

Quickly let's see what is nested for loop in R. If a for loop (inner loop) exists inside the body of the outer loop is called a nested loop. In each iteration of the outer loop, the inner loop will be re-started and the inner loop must finish all of its iterations before the outer loop can continue to its next iteration.

The `break` is used within the scope of either the outer or inner loop to exit the iteration without looping through all the items in sequence. When we use a `break` statement inside the inner loop, it just terminates the inner loop but not the outer loop. The below example demonstrates using the `break` statement in the outer as well as the inner loop.

```
# nested for loop using break
# outer loop & inner loop
for (i in c('A', 'B', 'C')){
  if( i == 'C')
  break
  # inner loop
  for (j in c('X', 'Y')){
    # print i & j
    if( j == 'Y')
    break
    print(c(i, j))
  }
}
```

# Output

# "A" "X"

# "B" "X"

## 1.3 Using Break Statement in While Loop

The following example demonstrates using a `break` statement in a while loop. Here, when `i` value

becomes 4 we are exiting from the while loop.

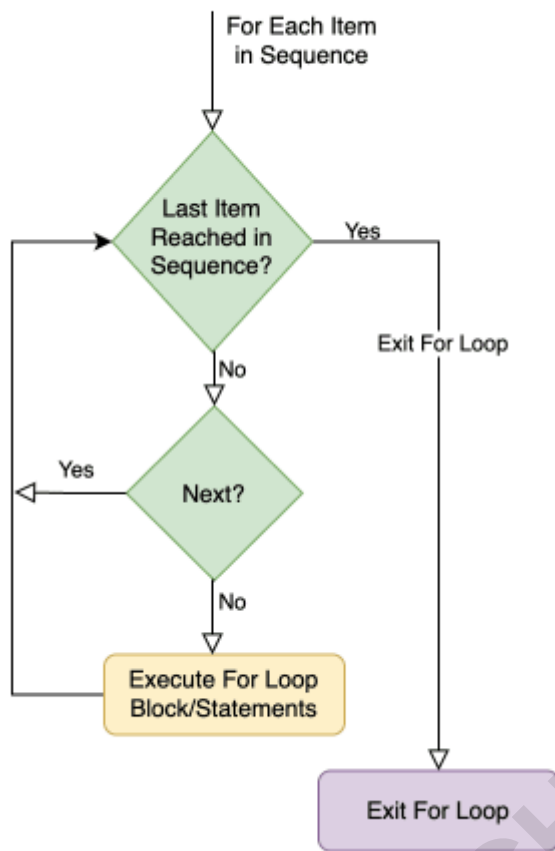
```
# Using break & next
i <- 1
n <- 5
while (i <= n) {
  if(i == 4){
    break
  }
  print(i)
  i = i + 1
}
```

Similarly, you can also use the break statement with a nested while loop.

## 2. Next Statement in R

The `next` is similar to `continue` from other programming languages that is used to skip the rest of the iterator and jump to the next item in the sequence.

**For Loop with Next in R**



**2.1 Using Break & Next Statement in R**

The below example demonstrated using break and next statement in R.

```

# Continue or next
numbers <- c('One', 'Two', "Three", "Four", "Five")
for(i in numbers) {
  if(i == "Two")
  next
  if( i == "Four" )
  break
  print(i)
}
    
```

Yields below output.

```
[1] "One"
[1] "Three"
```

## 2.2 Using Break & Next in Nested For Loop

```
# nested for loop using next
# outer loop
for (i in c('A','B','C')){
# inner loop
for (j in c('X','Y')){
# print multiplication
if( i == 'B' && j == 'X')
next
print(c(i,j))
}
}
```

```
# output
# "A" "X"
# "A" "Y"
# "B" "Y"
# "C" "X"
# "C" "Y"
```

## 2.3 Using Break & Next in While Loop

The below example uses both break and next statements in R with a while loop.

```
# Using
i <- 1
n <- 5
while (i <= n) {
if(i == 4){
break
}
if(i == 3){
i = i + 1
}
```

```
next  
}  
print(i)  
i = i + 1  
}
```

### 3. Conclusion

In this article, you have learned the break and next jumping statements in R and learned how to use them with for, and while statements. In simple words, the `break` is used within the scope of the control statements to stop and exit the iteration without looping through all the iterations. And the `next` is similar to `continue` from other programming languages that are used to skip the current iterator and jump to the next iteration.

### Related Articles

[Looping in R \(for, while, repeat\) With Examples](#)[For Loop in R With Examples](#)[Repeat Loop in R](#)[While Loop in R - With Examples](#)[Nested For Loop in R](#)[R Using For in Range with Example](#)[R if, if...else, if...else...if Usage with Examples](#)[R if...else with Multiple Conditions](#)[R ifelse\(\) Function](#)

### References

<https://cran.r-project.org/doc/manuals/r-release/R-lang.html>