**27. While Loops in R:**

**1. How Does a While Loop Differ from a For Loop in R?**

A while loop in R is used to repeatedly execute a block of code as long as a specified condition is true. It is ideal when the number of iterations is not known in advance and depends on a dynamic condition. On the other hand, a for loop is used to iterate over a predefined sequence (such as a vector, range, or list). It is most suitable when the number of iterations is known beforehand. In a while loop, the user must manually update the variables involved in the condition to avoid infinite loops, whereas in a for loop, the control of iterations is handled automatically by R.

**Example:**

* while**:** Continue until a condition becomes false (e.g., sum less than 100).
* for**:** Loop through elements in a vector (e.g., print all items in a list).

**2. While Loop to Find the First Number Divisible by 7**

**Input:**

num <- 1

while (num <= 50) {

if (num %% 7 == 0) {

print(paste("The first number divisible by 7 is", num))

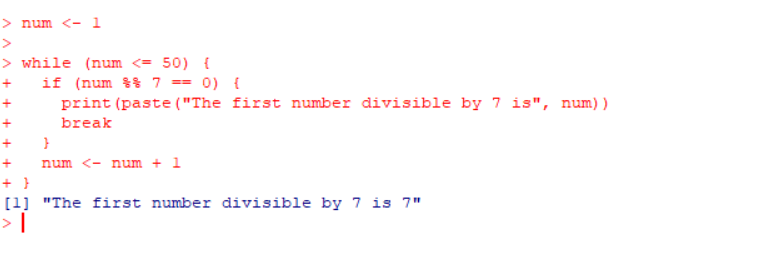
break

}

num <- num + 1

}

**Output:**

****

**3. Exit a While Loop Prematurely Using**break**.**

The break statement is used to terminate the loop prematurely, regardless of the condition.

**Example: Stop the loop once the sum exceeds 50:**

**Input:**

sum <- 0

counter <- 1

while (TRUE) {

sum <- sum + counter

if (sum > 50) {

print(paste("Sum exceeded 50 at counter:", counter))

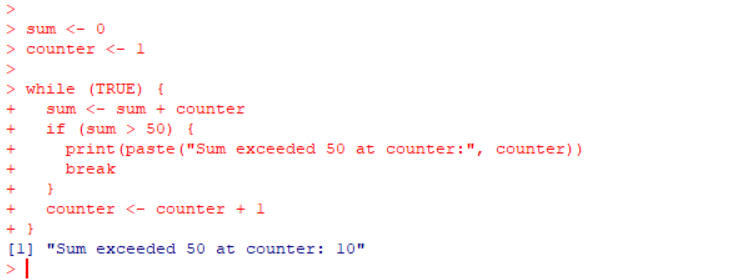
break

}

counter <- counter + 1

}

**Output:**

****

**4. Use Case of While Loop for Unknown Iterations**

A while loop is ideal when the number of iterations is not known beforehand, such as generating random numbers until a specific condition is met.

**Example:** Generate random numbers until a value greater than 0.9 is found:

**Input**:

random\_number <- 0

while (random\_number <= 0.9) {

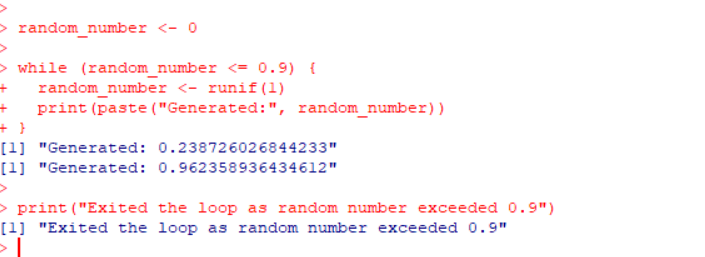
random\_number <- runif(1)

print(paste("Generated:", random\_number))

}

print("Exited the loop as random number exceeded 0.9")

**output:**

****