**Control Statements**

Control statements in R allow you to dictate how your program flows based on certain conditions or iterations. These statements include conditional statements (if, else if, else), loops (for, while, repeat), and other controls like break and next.

**1. Conditional Statements**

These are used to make decisions in a program based on some condition. They include if, else if, and else.

**a) if Statement**

The if statement evaluates a condition, and if it is TRUE, it executes the block of code associated with it.



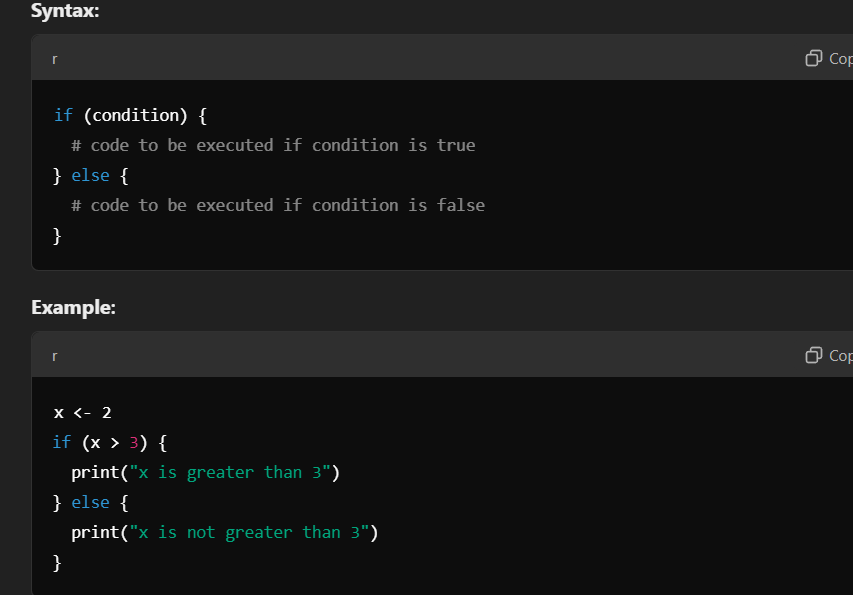
In this example, since x is greater than 3, the message "x is greater than 3" will be printed.

**Explanation:**

The condition inside the parentheses is evaluated. If the condition evaluates to TRUE, the block of code inside the if statement is executed. If it is FALSE, the code is skipped.

**b) else Statement**

The else statement is used to define an alternative block of code to execute if the if condition is FALSE.



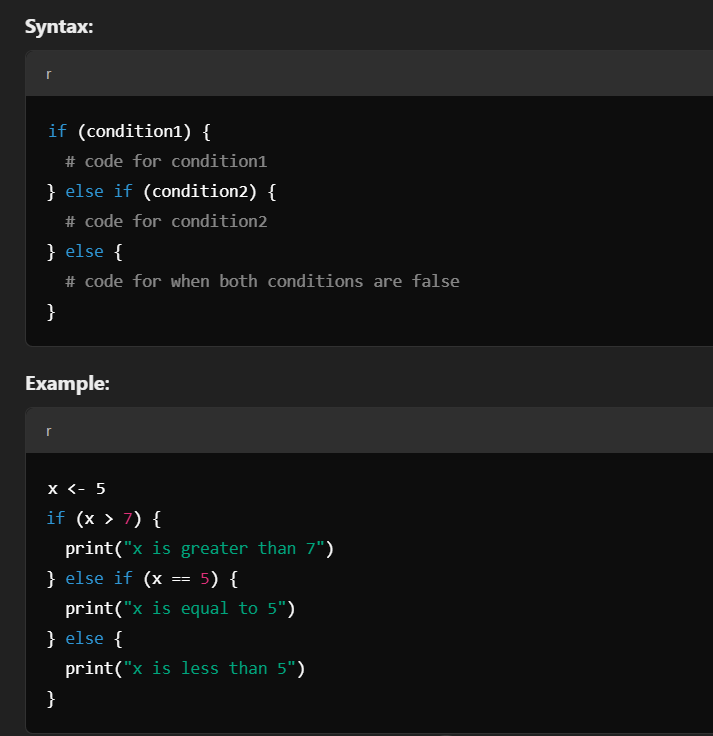
Here, since x is not greater than 3, the else block will execute, printing "x is not greater than 3."

**Explanation:**

The else block is executed if the if condition is FALSE. It provides an alternative code execution path.

**c) else if Statement**

The else if statement allows you to test multiple conditions. If the initial if condition is FALSE, the else if statement checks another condition.



Since x is equal to 5, the second condition will evaluate to TRUE, and "x is equal to 5" will be printed.

**Explanation:**

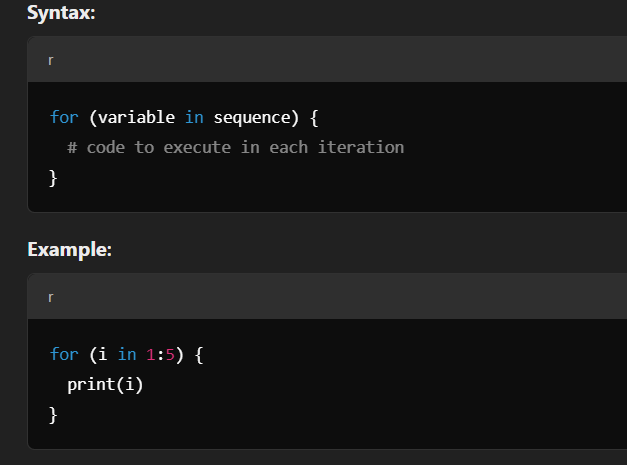
The else if statement allows for multiple checks. The first condition to evaluate as TRUE will trigger its associated block of code, and no other conditions will be evaluated.

**2. Loops**

Loops are used to repeat a block of code multiple times, either a set number of times or until a condition is met.

**a) for Loop**

The for loop in R is used to iterate over elements in a sequence (like vectors or lists). Each iteration applies the code block to the current element in the sequence.



This will print the numbers from 1 to 5, as the loop iterates through the sequence 1:5.

**Explanation:**

The loop assigns each value in the sequence to the variable i, and the block of code is executed with that value. Once all elements have been processed, the loop ends.

**b) while Loop**

The while loop continues to execute a block of code as long as the specified condition remains TRUE.



This will print numbers from 1 to 5, as the loop continues until x becomes greater than 5.

**Explanation:**

The while loop checks the condition before each iteration. If the condition is TRUE, the code inside the loop is executed. When the condition becomes FALSE, the loop terminates.

**c) repeat Loop**

The repeat loop is an infinite loop that continues to execute until a break statement is encountered.



This will print numbers from 1 to 5, and the loop will terminate when x exceeds 5.

**Explanation:**

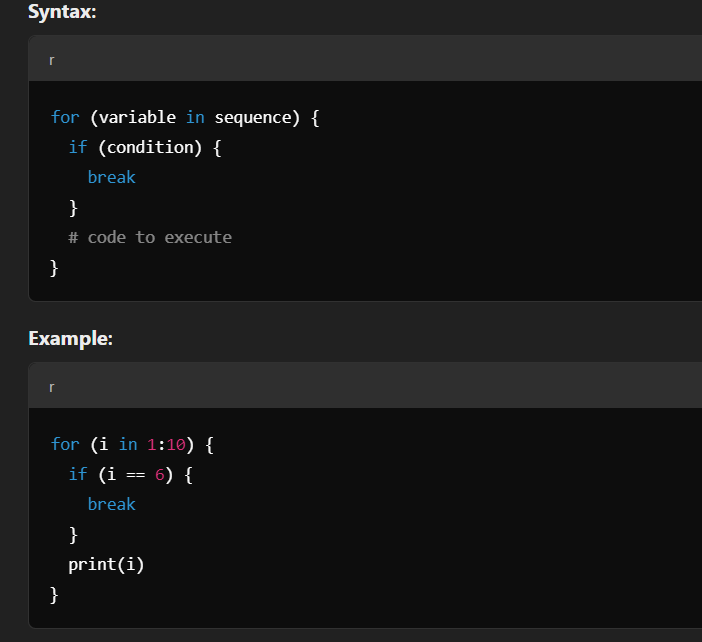
The repeat loop will run indefinitely unless it is explicitly terminated using a break statement. This makes it useful in situations where you want the loop to run until a certain condition is met.

**3. Control Statements within Loops**

These are statements used to alter the normal flow of a loop.

**a) break Statement**

The break statement is used to exit a loop prematurely when a certain condition is met.



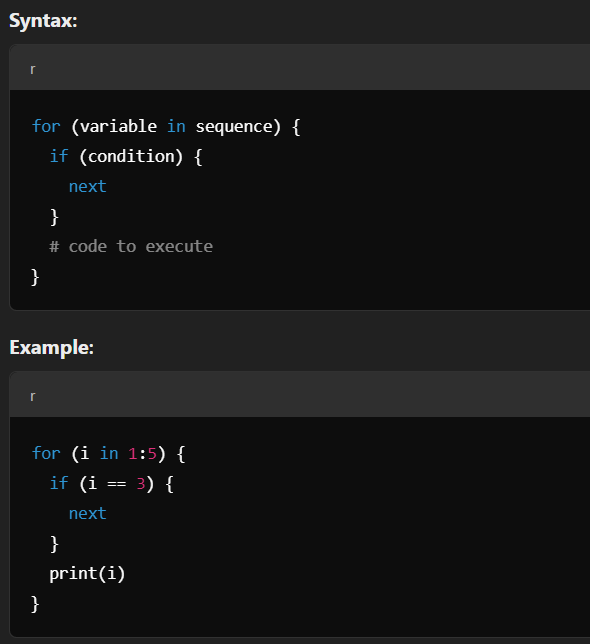
This will print numbers from 1 to 5, and then stop when i equals 6 due to the break statement.

**Explanation:**

The break statement is used to exit a loop when a specified condition is satisfied. It immediately terminates the loop, regardless of where the loop counter is.

**b) next Statement**

The next statement is used to skip the current iteration and move on to the next iteration of the loop.



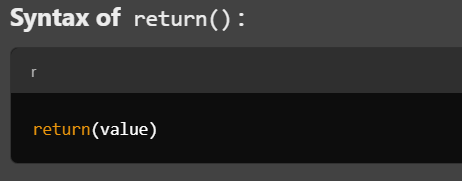
This will print 1, 2, 4, and 5, skipping the iteration where i == 3.

**Explanation:**

The next statement causes the loop to skip the remainder of the code in the current iteration and continue with the next iteration. It is commonly used when you want to avoid specific cases in a loop.

c) **return Statement**

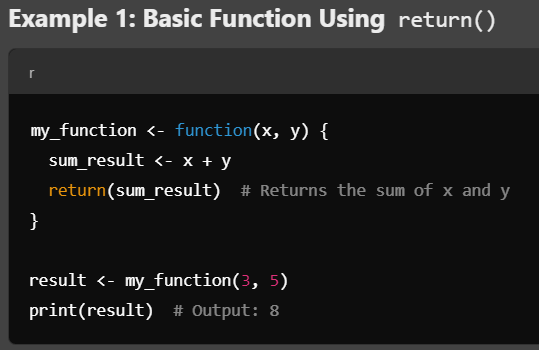
In R programming, the return() statement is used to explicitly return a value from a function. When a function is executed, the return() statement immediately exits the function and sends the specified value back to the caller. Although R automatically returns the result of the last evaluated expression in a function, using return() can make the code more readable and allow for early exits in certain cases.



**value**: The value or object you want the function to return.

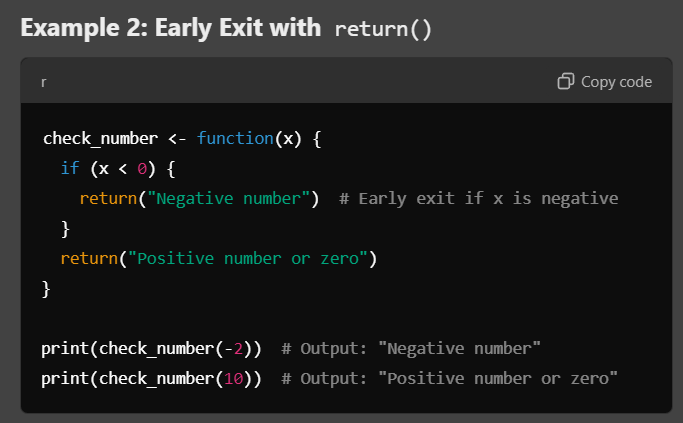
**When to Use return()?**

* When you want to **explicitly** state what value should be returned.
* To **exit a function early**, based on conditions.
* To **improve code readability** and structure.



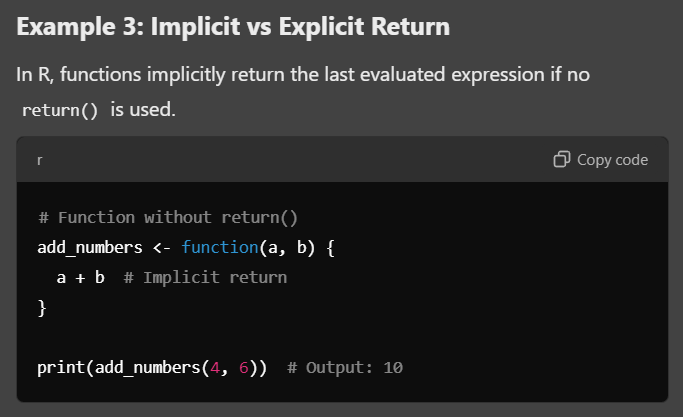
**Explanation**:

* The function my\_function adds x and y and explicitly returns the result using return(sum\_result).
* When we call my\_function(3, 5), it returns the sum 8



**Explanation**:

* The function check\_number uses return() to exit early if the input number is negative.
* If x < 0, the function returns "Negative number" and doesn't proceed further.
* If x >= 0, it returns "Positive number or zero".

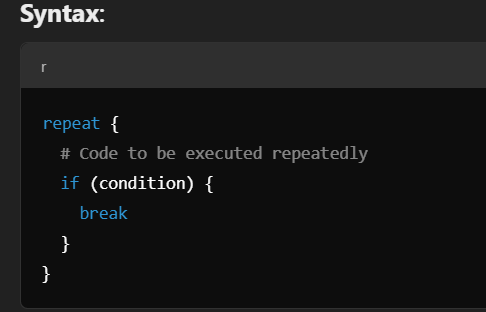


**Explanation**:

* Here, no return() is used, but the function still returns the result of a + b because it is the last evaluated expression

### ****Repeat Statement in R****

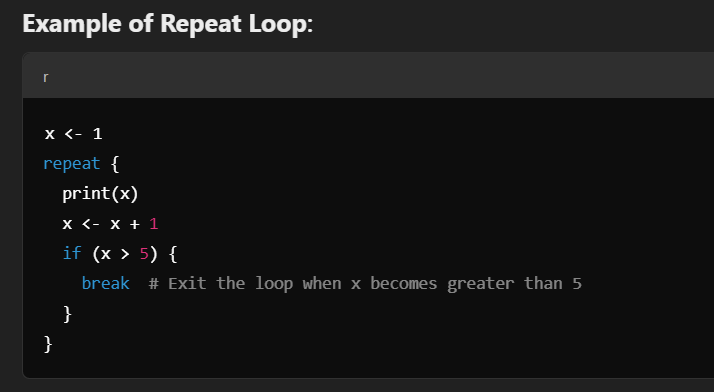
The repeat statement in R is used to create an infinite loop, where the loop continues indefinitely until a specific break statement is encountered to terminate it. Without the break statement, the repeat loop will keep running forever.



 **repeat**: This keyword begins the infinite loop.

 **break**: Used inside the loop to stop execution when a condition is met.

**condition**: Defines the criteria to exit the loop.



**Explanation**:

* This loop will keep printing the value of x and incrementing it by 1 in each iteration.
* When x becomes greater than 5, the break statement is encountered, which terminates the loop.