All the variables that we use within a program needs to be stored somewhere. That somewhere can be called as an environment in R language. They are closely related to lists in which they are used to store diverse types of variables together. There may arise two situations where you might encounter environments. 1st, whenever a function is called, all the variables described by the function are stored in an environment belonging to that specific function (a function along with its environment is sometimes referred to as closure). 2nd, when you load a package, the functions in that package gets stored in an environment on the search path.

**What is a Function in R Programming?**

A function can be defined as a collection of statements structured together for carrying out a definite task. R provides a huge number of in built functions and also user can create their own functions.

In R, a function is treated as object so the R interpreter is capable of passing control to the function, along with arguments which may be essential to the function for achieving the actions. The function has the capability to turn its performance and returns control to the interpreter that may be stored in other objects.

The keyword ‘function’ is used to create a function in R. The basic structure of a function can be:

function\_name <- function (argu\_1, argu\_2, .... argu\_N)

{

#Function body

}

**Various Components of a Function**

The function in R is having various parts and each of them is having its own characteristics. These are:

* **Function Name**: is the real name of the function with which you can call it in some other part of the program. It is stored as an object with this name given to it.
* **Arguments**: is a placeholder for that specific function. As a function gets invoked, you can pass a value to the argument. Arguments are not mandatory to be used within the function; i.e. a function may not contain any arguments. Arguments can contain default values also.
* **Function Body**: It may contain a set of statements which specifies what the function does and how it will work along with its use.
* **Return Value**: Return value of any function is the last expression in the function which tells what that function is able to return.

**Built in Functions**

Built in functions are those functions whose meaning and working is already defined within the function's body and they are kept somewhere within the packages or libraries of R language. These pre- defined functions make programmers task easier.

Some common examples of in built functions are:

seq(),max(), mean(), sum(x), paste(...) etc.

They are directly called and used by programmers who are writing programs.

Example:

print (seq (12,30))

This creates a sequence of number from 12 to 30 using the predefined function seq().

print (mean (4:26))

This calculates the mean of all the numbers ranging from 4 to 26

Here is a

hypotenuse <- function (x, y)

{

sqrt (x ^ 2 + y ^ 2)

}

You can now call this function as like this:

hypotenuse (3, 4)

## [1] 5

hypotenuse (y = 24, x = 7)

## [1] 25

Read about functions from

<https://www.geeksforgeeks.org/functions-in-r-programming/>

<https://www.geeksforgeeks.org/function-arguments-in-r-programming/>

<https://www.geeksforgeeks.org/types-of-functions-in-r-programming/>