

Nested Functions

We are going to learn all about nested functions in this lesson.

We'll cover the following

- What are Nested Functions?
 - Syntax of a Nested Function
- Defining a Function in Another Function
 - Syntax for defining one function in another

What are Nested Functions?

As we learned in the previous [lessons](#), a function is a set of statements that are executed together to achieve a specific goal or task.

In such a situation, we might have to use another function in the current function. This is called **nested functions**. We will be learning about **user-defined functions** being used in another **user-defined** function.

Syntax of a Nested Function

```
functionName1 <- function(argument1, argument2, ..., argumentN)
{
  # Statements
}

functionName2 <- function(argument1, argument2, ..., argumentN)
{
  # Statements
  functionName1(argument1, argument2, ..., argumentN)
}
```

Let's rewrite our previous example of finding the **maximum** and **minimum** number in a **vector** using multiple functions.

```
maxNumber <- function(myVector)
{
  max <- -1
```



```

max <- 1
for(v in myVector)
{
  if(max < v)
  {
    max = v
  }
}
max

minNumber <- function(myVector)
{
  min <- Inf

  for(v in myVector)
  {
    if(min > v)
    {
      min = v
    }
  }
  min
}

maxMinNumber <- function(myVector)
{
  max <- maxNumber(myVector) # calling the maxNumber() function
  min <- minNumber(myVector) # calling the minNumber() function

  print(paste("The maximum value in the vector is: ", max), quote = FALSE)
  print(paste("The minimum value in the vector is: ", min), quote = FALSE)
}

# Driver Code
input <- c(2, 5, 4, 10)
maxMinNumber(input) # calling the maxMinNumber() function

```



Calling one function from another

In this code, we have split the task of finding the maximum number in `maxNumber()` function and the task of finding the minimum number in `minNumber()` function.

Later, we call both these functions in another function called `minMaxNumber()`.

Defining a Function in Another Function

We can also create one function in another function and use it in the same function. How cool is that!

Syntax for defining one function in another

```

functionName1 <- function(argument1, argument2, ..., argumentN)
{
  # Creating a Function Inside a Function
  functionName2 <- function(argument1, argument2, ..., argumentN)
  {
    # Statements
  }

  # Using the created function
  functionName2(argument1, argument2, ..., argumentN)
}

```

Let's explore how we can implement this in R:

```

driverCode <- function()
{
  input <- c(2, 5, 4, 10)

  # Creating the maxNumber function inside the driverCode function
  maxNumber <- function(myVector)
  {
    max <- -1
    for(v in myVector)
    {
      if(max < v)
      {
        max = v
      }
    }
    max
  }

  # Creating the minNumber function inside the driverCode function
  minNumber <- function(myVector)
  {
    min <- Inf
    for(v in myVector)
    {
      if(min > v)
      {
        min = v
      }
    }
    min
  }

  # Creating the maxMinNumber function inside the driverCode function
  maxMinNumber <- function(myVector)
  {
    max <- maxNumber(myVector) # calling the maxNumber() function
    min <- minNumber(myVector) # calling the minNumber() function

    print(paste("The maximum value in the vector is: ", max), quote = FALSE)
  }
}

```

```
    print(paste("The maximum value in the vector is: ", max), quote = FALSE)
    print(paste("The minimum value in the vector is: ", min), quote = FALSE)
  }

  # calling the maxMinNumber function from inside the driverCode function
  maxMinNumber(input) # calling the maxMinNumber() function
}

driverCode() # Calling the driverCode function.
  # This is where actual execution of code starts
```

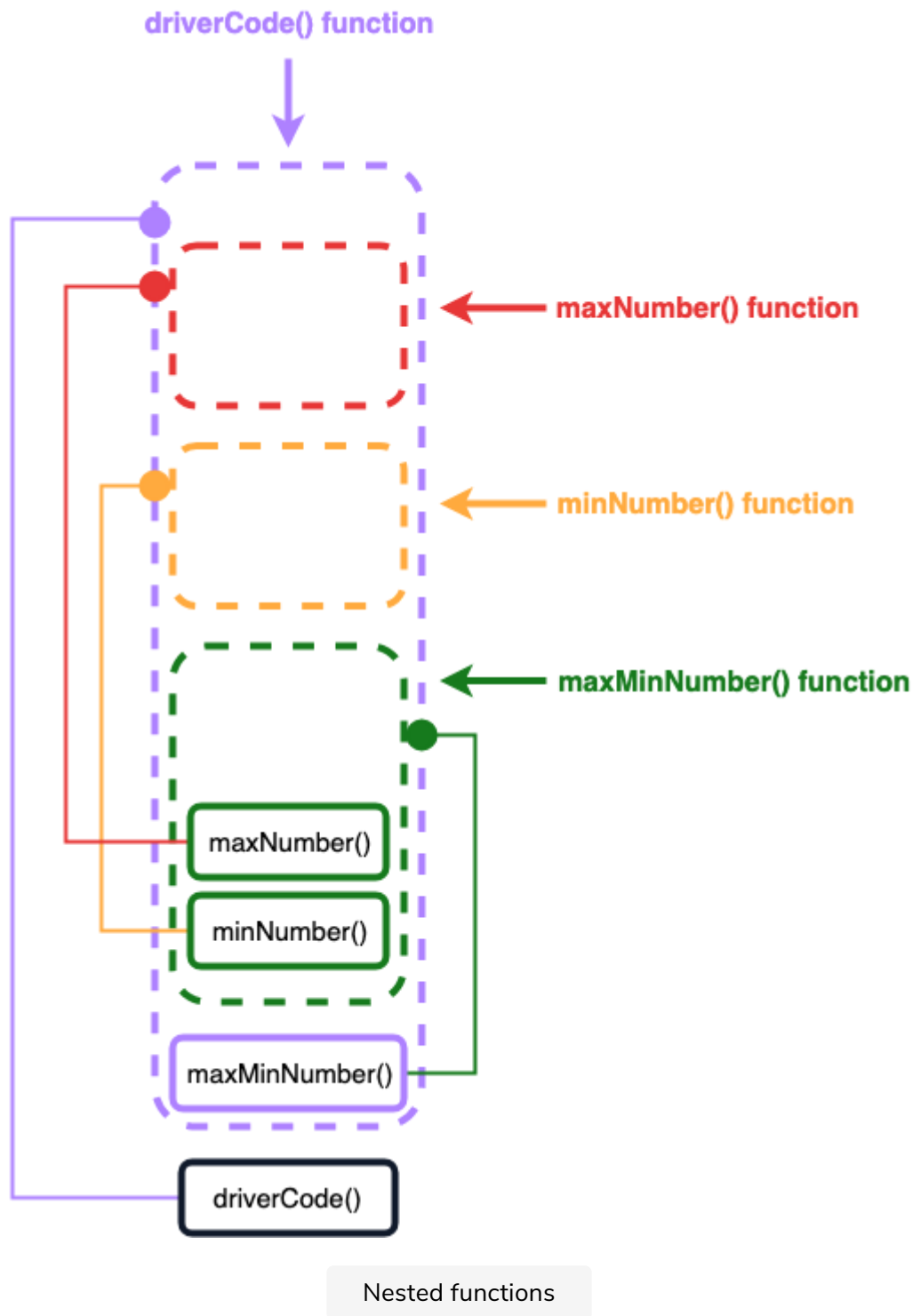


Calling one function from another

As you can see from the code above, the functions `maxNumber()`, `minNumber()` and `maxMinNumber()` are all created in the function `driverCode()`.

We also call the function `maxMinNumber()` from the `driverCode()` function which later goes on to call `maxNumber()` function and `minNumber()` function.

Let's have a look at an illustration:



In the above illustration, the dotted line shows the function, whereas the solid lines indicate *function calls*. For example, `maxNumber()` and `minNumber()` function are called from inside `maxMinNumber()` function.

Also, the `drivercode()` function is *called* from outside all these functions. In the `driverCode()` function, all the other functions are called and created.

In the next lesson, we will be learning Recursion.

