Entering Data from the Keyboard

In this lesson, we will learn about getting data from the user.

We'll cover the following

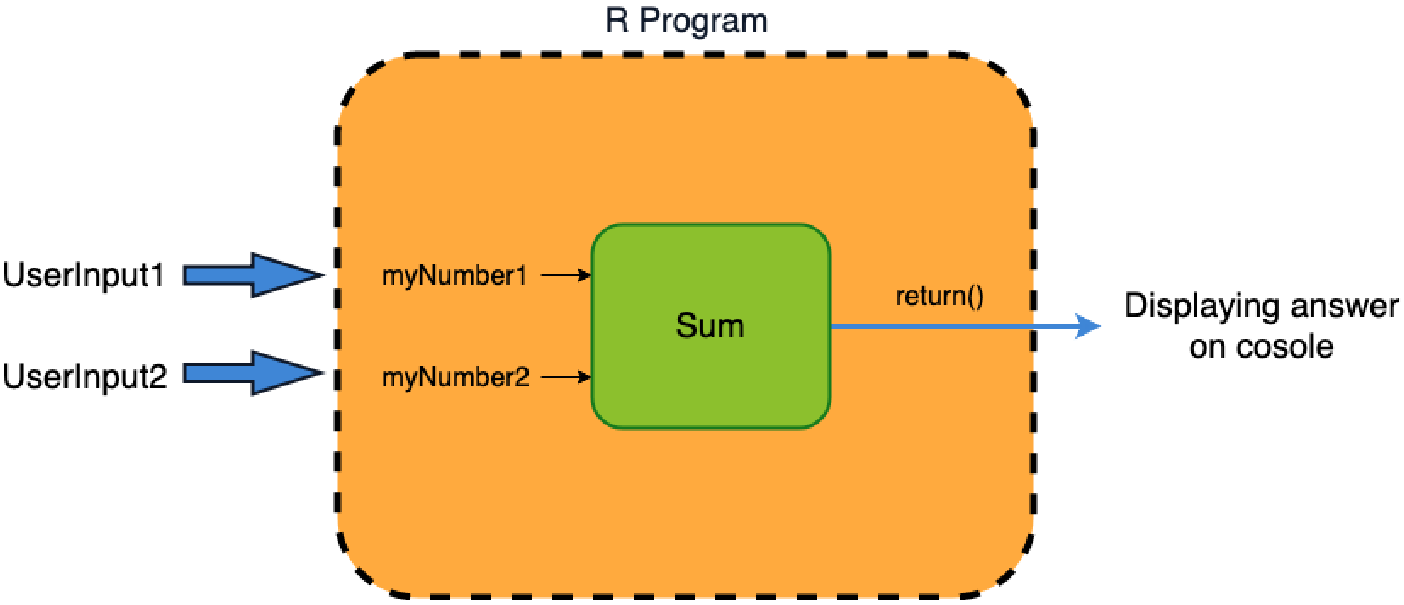
- Using readLine() Function to Get User Input
  - Steps for fetching data using STDIN

Sometimes while we are writing a program, we may want to take input from the user who is actually using our program. We design tasks so that they run on any input that the user may give.

Take the example of a calculator. We put some input into it and tell it what to do with the input. It performs the respective task and displays the result.

Now suppose we are also creating a calculator that performs only **addition**.

Let’s have a look at an example:



Taking user's input and passing to a function.

In the illustration, above we take two inputs from the user, perform a certain task on it and then display the result on the console.

Using `readLine()` Function to Get User Input

Using `readLine()` Function to Get User Input

The `readLine()` function **reads** a line from the console, this is referred as `stdin` file in R. The data fetched through this function can be stored in a variable and later used by the program.

Steps for fetching data using `STDIN`

1. First, fetch the file using

```
input <- file("stdin")
```

2. Then use the function `readLines()`

```
UserInput <- readLines(input, <numberOfInputs>)
# Here the variable `UserInput` is a list and each of its elements can be accessed using square brackets.
```

Let's look at the code for fetching data from a user and then passing it to a function.

> Follow the following steps for executing the code below:
>
> - Click on `>_ STDIN` and enter two values on separate lines. The first number should be typed in on the first line and the second number on the second line.
> - Then click `RUN`.

```
Sum <- function(value1, value2)
{
    x = as.integer(value1)
    y = as.integer(value2)
    x+y
}

# Driver Code
input <- file("stdin")
UserInput <- readLines(input, 2)
print(Sum(UserInput[1], UserInput[2]))
```

In the code above, we fetch data from the file `stdin` and then read all its lines in the variable `UserInput`. The number of inputs is 2. Next, we pass these inputs to the `Sum()` function.

> The input that is taken from a file using the `readLines()` function is always of type **string**.

Therefore, to perform the **addition operation** we need to convert it to an **integer** using the `as.integer()` function. This function converts the passed parameter to **integer** type object.

The core strength of a typical R application is **manipulating large chunks of data**. Such data cannot be easily entered by a user; therefore, **external files** are used to pass data to programs in R language. Such a method is more efficient and easy to use.

We will be learning how to get input from `.csv` and `.txt` files in the coming lessons.