

Exercise No -4

Title of Experiment: String Operations using R

Solution Provider Name: Pranita Anil Kamble

College Name: V.P.Institute of Management Studies & Research, Sangli

#R Version: 3.6.3

#R Studio Version: 1.3.1093

Abstract

Any value written within a pair of single quote or double quotes in R is treated as a string. Internally R stores every string within double quotes, even when you create them with single quote.

Introduction

- Double quotes can be inserted into a string starting and ending with single quote.
- Single quote can be inserted into a string starting and ending with double quotes.

Prerequisite for experiment

- To work through the examples you will need the stringr package.
 1. install necessary packages:

```
install.packages("tidyverse")
```

```
library(tidyverse)#for str_view_all function to extract all string by word
```

```
library(stringi) #for string manipulation
```

Q1) Create string “Learning R Programming Is Very Interesting”

Explanation: To create a string we can write the text inside the quotes as below.

Solution:

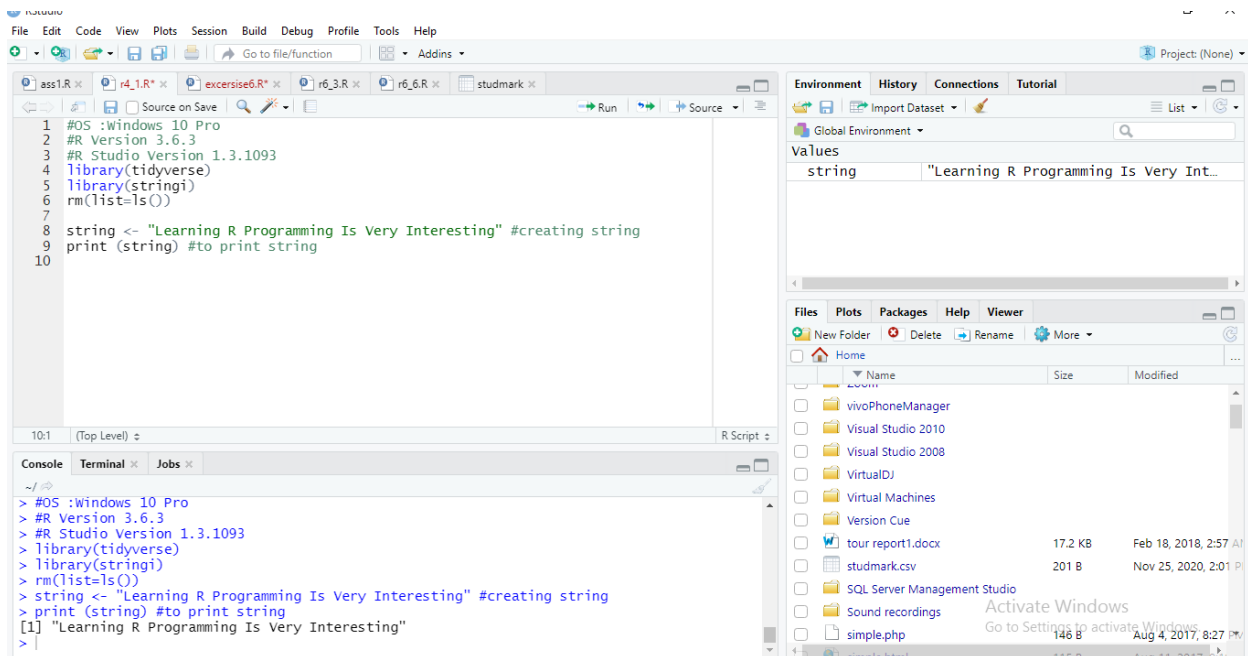
```
rm(list=ls())
```

```
string <- "Learning R Programming Is Very Interesting" #creating string
```

```
> print (string) #to print string
```

[1] "Learning R Programming Is Very Interesting"

Screen:



Q2) Count No. of Characters in string.

Explanation: To count the no. of characters from the given string we use the `nchar()` function of a `stringi` library package.

Solution:

```
library(tidyverse)
```

```
library(stringi)
```

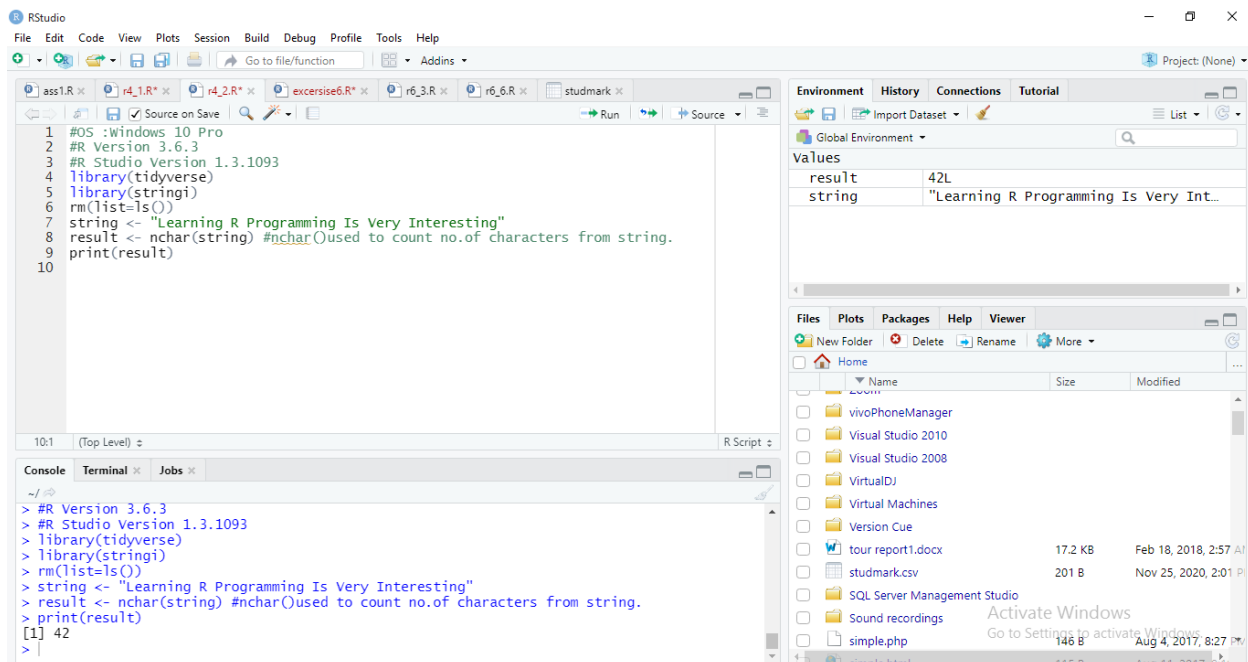
```
rm(list=ls())
```

```
string <- "Learning R Programming Is Very Interesting"
```

```
result <- nchar(string) #nchar()used to count no.of characters from string.
```

```
print(result)
```

```
[1] 42
```



Q3) Display each word of string separately.

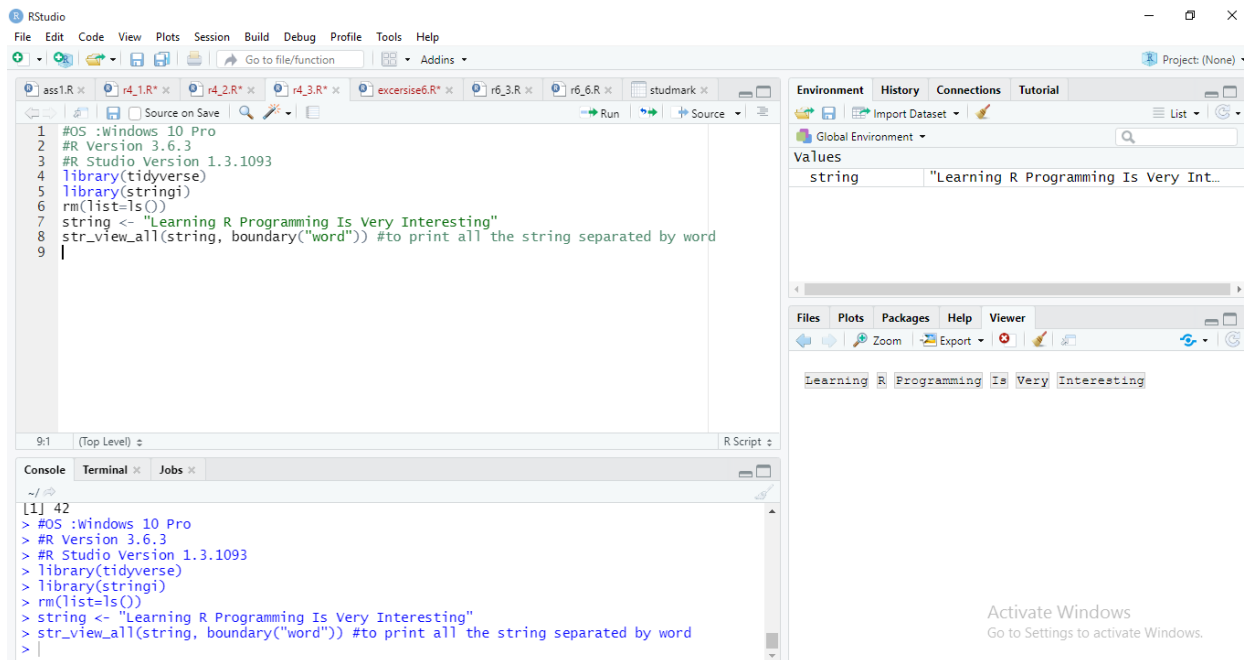
Explanation: To display each word of a string separately we use `str_extract_all()` function with arguments as name of the variable/vector which hold the whole string and the second argument is like boundary which should specify by word because we want to separate each word of a string.

Solution:

`str_extract_all(string, boundary("word"))` #to print all the string separated by word

```
[1] "Learning" "R" "Programming" "Is" "Very" "Interesting"
```

Screen:



Q4) Display String in uppercase.

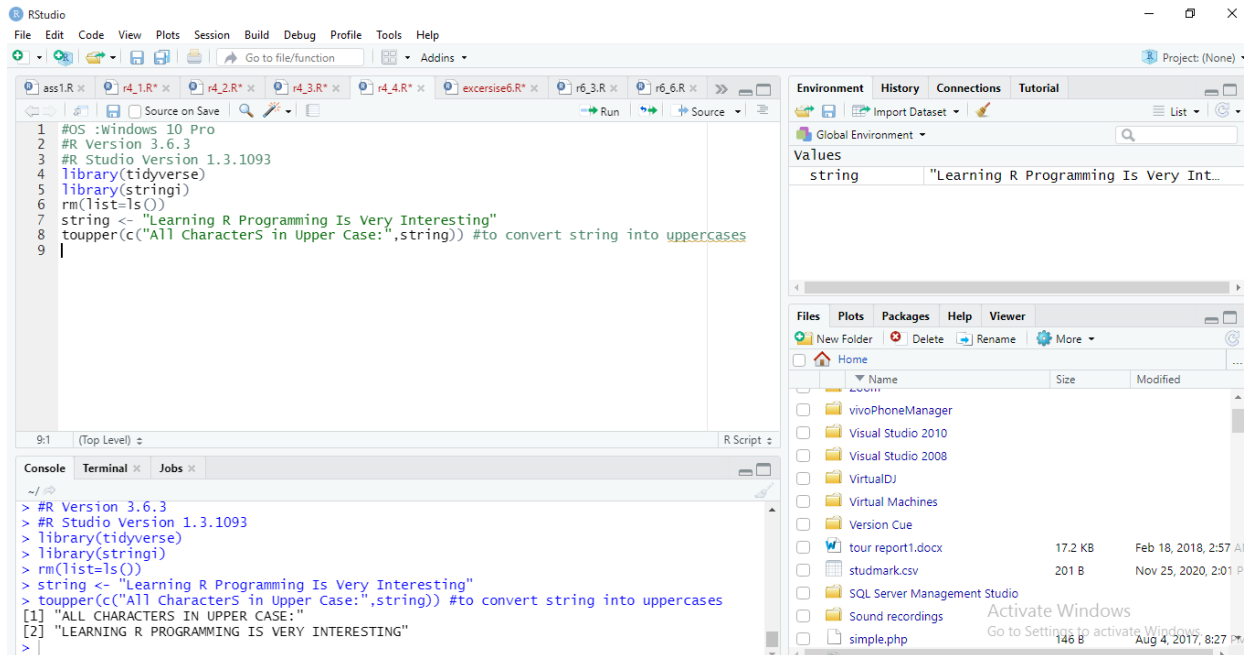
Explanation: To transform the string into the uppercase we have to use `toupper()` function for string manipulation.

Solution:

`toupper(c("All CharacterS in Upper Case:",string))` #to convert string into uppercases

[1] "ALL CHARACTERS IN UPPER CASE:" "LEARNING R PROGRAMMING IS VERY INTERESTING"

Screen:



Q5) Display String in lowercase.

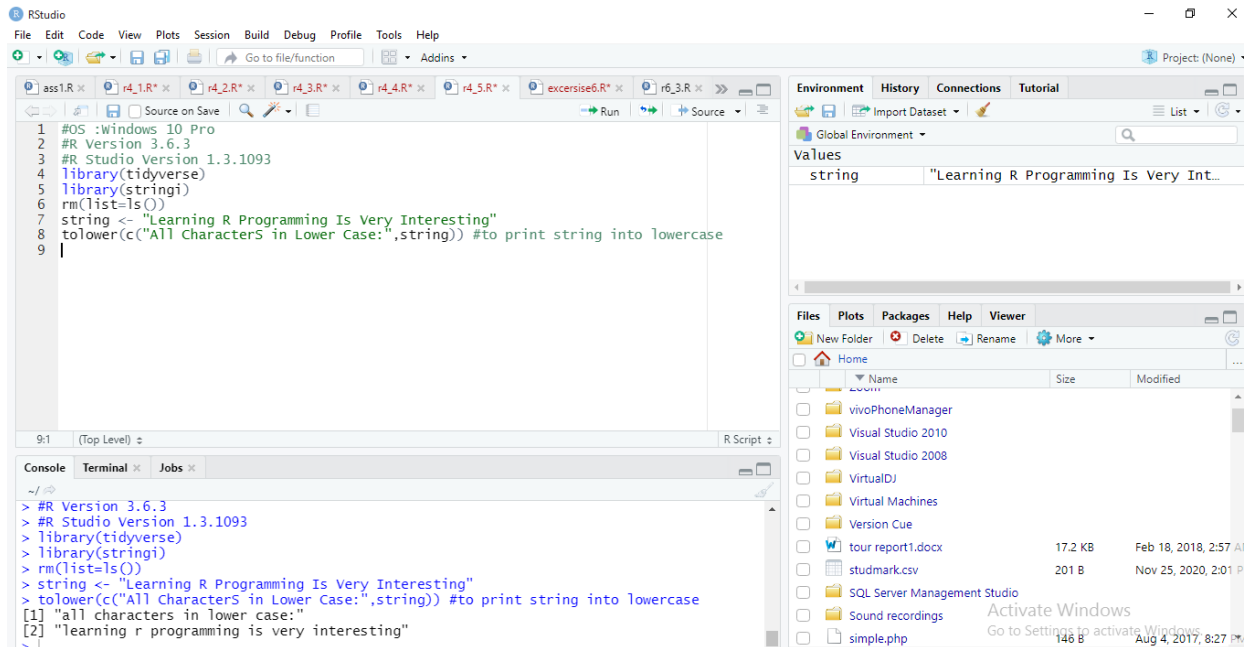
Explanation: To transform the string into the lowercase we have to use `tolower()` function for string manipulation.

Solution:

`tolower(c("All CharacterS in Lower Case:",string))` #to print string into lowercase

[1] "all characters in lower case:" "learning r programming is very interesting"

Screen:



Q6) Display Substring as a “R Programming”.

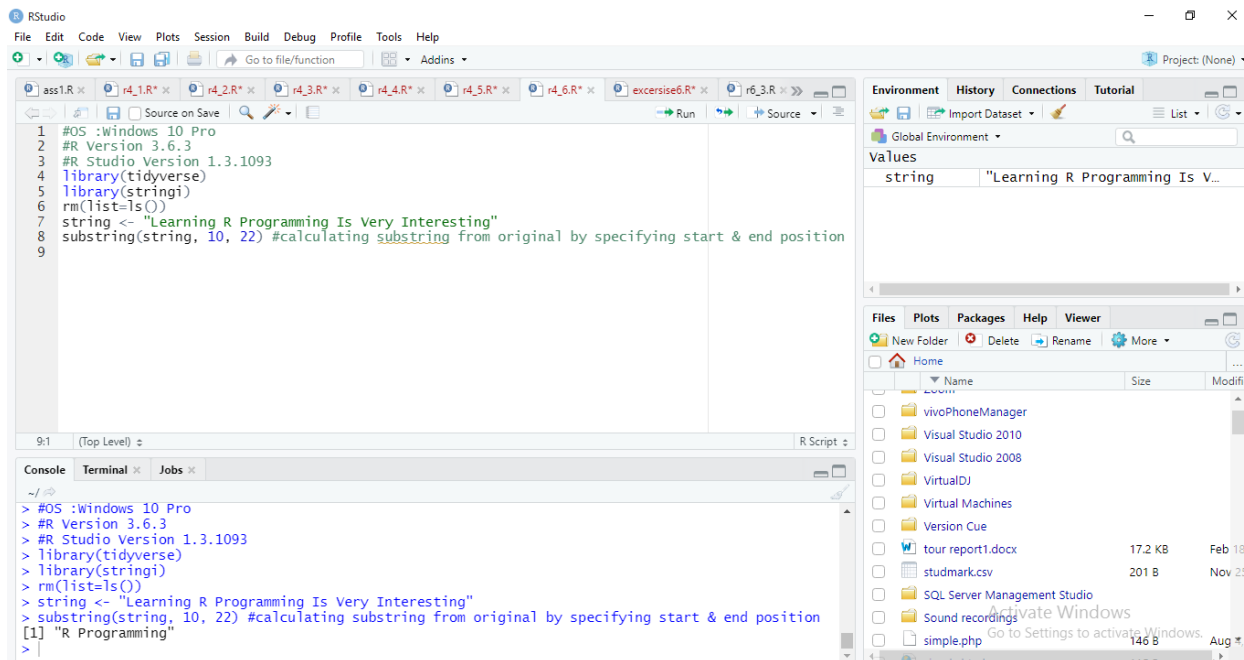
Explanation: To print the substring from the given string we have to use `substr` () function with arguments name of the string holding variable, the start and the end position count of the substring.

Solution:

`substr(string, 10, 22) #calculating substring from original by specifying start & end position`

`[1] "R Programming"`

Screen:



Q7) Display string in reverse order.

Explanation: To display the given string in reverse order we have to use the `stri_reverse()` function by passing a parameter name as a variable which should hold the whole string.

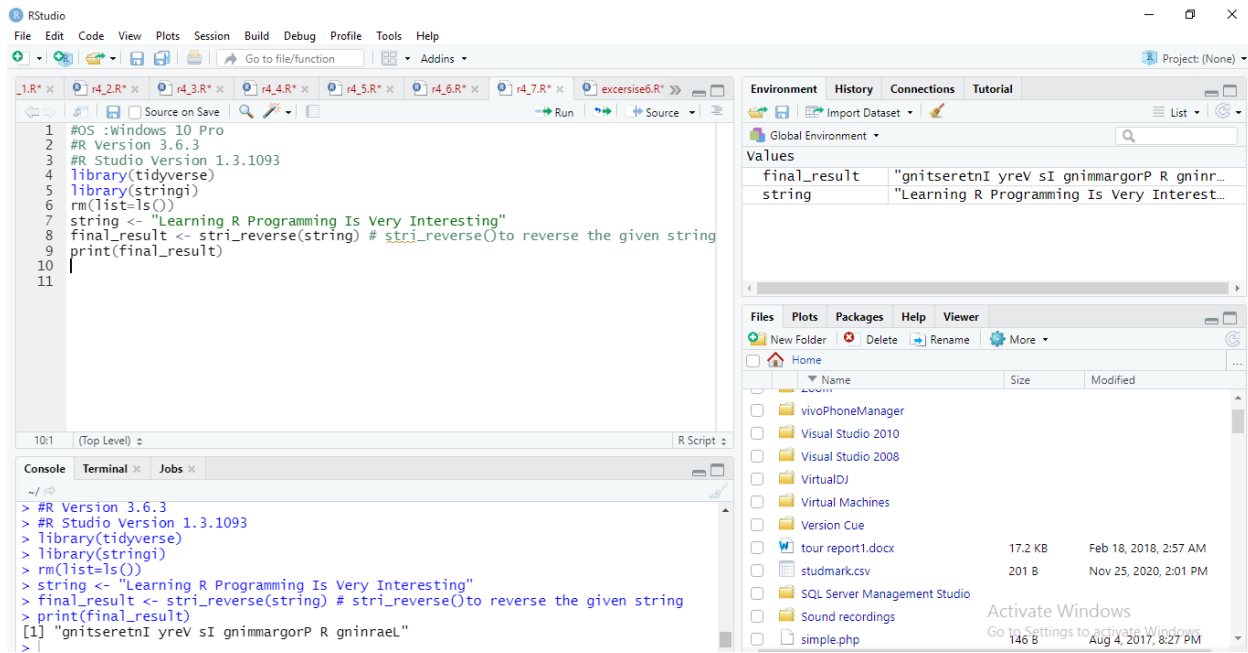
Solution:

```
final_result <- stri_reverse(string) # stri_reverse() to reverse the given string
```

```
> print(final_result)
```

```
[1] "gnitseretnI yreV sI gnimmargorP R gninraeL"
```

Screen:



XXXXXXXXXXXX