

Лабораторная работа №3. Скрипты и первые программы на R

Контрольные вопросы и задания

Задание 1 Создать скрипт и перенести в него результаты самостоятельного задания № 3 из Лабораторной работы № 2: создать несколько констант и переменных разных типов, выполнить арифметические расчеты по самостоятельно подготовленным формулам с этими константами и переменными. В формулах проверить работу операторов `%%` и `%/%`.

Ответ

Листинг 1: `./scripts/lab3_zad1.R`

```
1 library(stringi) # использовал для конкатенации строк
2
3 a <- 7
4 b <- 3
5 c <- 5
6 d <- 10
7
8 print(stri_c("(a + b) * c) = ", (a + b) * c))
9 print(stri_c("(a + b) * (c - d)) = ", (a + b) * (c - d)))
10 print(stri_c("a / b * c = ", d / c * 7))
11 print(stri_c("a / b * c ^ d = ", a / b * c ^ d))
12 print(stri_c("d * a %% b = ", d * a %% b))
13 print(stri_c("(d * a) %% b = ", (d * a) %% b))
14 print(stri_c("d * a %/% b = ", d * a %/% b))
15 print(stri_c("(d * a) %/% b = ", (d * a) %/% b))
16 print(stri_c("(d * a) ^ 2 %/% b = ", (d * a) ^ 2 %/% b))
17 print(stri_c("((d * a) ^ 2) %/% b = ", ((d * a) ^ 2) %/% b))
18 print(stri_c("a %% b = ", a %% b))
19 print(stri_c("a %/% b = ", a %/% b))
```

```
1 [1] "(a + b) * c) = 50"
2 [1] "(a + b) * (c - d)) = -50"
3 [1] "a / b * c = 14"
4 [1] "a / b * c ^ d = 22786458.33333333"
5 [1] "d * a %% b = 10"
6 [1] "(d * a) %% b = 1"
7 [1] "d * a %/% b = 20"
8 [1] "(d * a) %/% b = 23"
9 [1] "(d * a) ^ 2 %/% b = 1633"
10 [1] "((d * a) ^ 2) %/% b = 1633"
11 [1] "a %% b = 1"
12 [1] "a %/% b = 2"
```

Задание 2 Используя различные внешние программы-редакторы, подготовить в них скрипты с одним и тем же содержанием:

```
print("Проверяю работу скрипта из редактора . . .")
```

где вместо многоточия вписать название программы-редактора. Использовать программы, как установленные на компьютере (Word, WordPad, блокнот и т.д.), так и найденные в интернете (AkelPad, Notepad++ и др). При сохранении скрипта проверять, где это возможно, в какой кодировке выполняется сохранение. Всегда выбирать UTF-8. После написания скрипта в редакторе открывать его в среде R и запускать на выполнение. Цель задания – сравнить удобство и возможности написания скриптов в разных редакторах.

Ответ

Листинг 2: ./scripts/lab3_zad2_*.R

```
1 print("Checking the script from the editor IntelliJ IDEA")
```

```
1 >Rscript ./scripts/lab3_zad2_intellij.R
```

```
2 [1] "Checking the script from the editor IntelliJ IDEA"
```

```
1 print("Checking the script from the editor PyCharm")
```

```
1 >Rscript ./scripts/lab3_zad2_pycharm.R
```

```
2 [1] "Checking the script from the editor PyCharm"
```

```
1 print("Checking the script from the editor Sublime Text 3")
```

```
1 >Rscript ./scripts/lab3_zad2_sublime.R
```

```
2 [1] "Checking the script from the editor Sublime Text 3"
```

```
1 print("Checking the script from the editor Vim")
```

```
1 >Rscript ./scripts/lab3_zad2_vim.R
```

```
2 [1] "Checking the script from the editor Vim"
```

Задание 3 Используя скрипты, написать программы, в которых требуется

- получить вводом с клавиатуры два числа
- первое число возвести в степень второго, вывести эти числа и результат
- затем второе число возвести в степень первого, вывести результат
- разделить первое число на второе, вывести результат

- выполнить деление на ноль, посмотреть на результат

Указание. Для ввода строки с клавиатуры использовать функцию `readline()`. Для преобразования строки в число использовать функции `as.integer()`, `as.double()`. Для формирования строки с результатами использовать функции `paste()`, `paste0()`. Для вывода результатов функцию `print()`.

Ответ

Листинг 3: `./scripts/lab3_zad3.R`

```
1 # возводит a в степень b
2 exp_ <- function(a, b) {
3   res <- a ^ b
4   print(paste0(a, " ^ ", b, " = ", res))
5   return(res)
6 }
7
8 # делит a на b
9 div <- function(a, b) {
10  res <- a / b
11  print(paste0(a, " / ", b, " = ", res))
12  return(res)
13 }
14
15 main <- function() {
16   a <- NA; b <- NA
17   while (is.na(a)) {
18     a <- as.double(readline("a = "))
19   }
20   while (is.na(b)) {
21     b <- as.double(readline("b = "))
22   }
23   exp_(a, b)
24   exp_(b, a)
25   div(a, b)
26   div(a, 0)
27 }
28
29 main()
```

```
1 > a = 12
2 > b = 22
3 [1] "12 ^ 22 = 5.52061438912436e+23"
4 [1] "22 ^ 12 = 12855002631049216"
5 [1] "12 / 22 = 0.545454545454545"
6 [1] "12 / 0 = Inf"
7 [1] Inf
```

Задание 4 Для задания № 4 из Лабораторной работы № 2 написать программы, в которых Пользователь с клавиатуры вводит значения двух переменных разных типов, которые затем сравниваются между собой. Использовать функции `readline()`, `print()` и функции преобразования типов.

Ответ

Листинг 4: `./scripts/lab3_zad4.R`

```
1 # преобразование ввода с клавиатуры по цепочке
2 # приведения типов: logical – integer – double – character
3 input <- function(name) {
4   x <- readline(paste0(name, " = "))
5   t <- as.logical(x)
6   if (is.na(t)) {
7     t <- as.integer(x)
8     if (is.na(t)) {
9       t <- as.double(x)
10      if (is.na(t)) {
11        return(x)
12      }
13    }
14  }
15  return(t)
16 }
17
18 main <- function() {
19   a <- input("a")
20   b <- input("b")
21
22   print('Сравнение через "<":')
23   print('Неявное преобразование: logical – integer – double –
24     character')
25   print(a)
26   print(paste0("a(", typeof(a), ") < ", "b(", typeof(b), ") = ",
27     a < b))
28 }
29
30 main()
```

```
1 > a = -1
2 > b = True
3 [1] "Сравнение через \"<\": \"
4 [1] "Неявное преобразование: logical – integer – double – character"
5 [1] -1
6 [1] "a(integer) < b(logical) = TRUE" # b преобразовалось к integer и
7 # получило значение 1, теперь: ?1 > -1 - True
```

Задание 5 Изучить 5-7 любых функций языка R, исследовать передачу параметров в эти функции. Выяснить формальные имена параметров и значения параметров по умолчанию. Проверить работу этих функций, написав для этого простейшие тестирующие программы. По результатам проверки работы функций написать по каждой функции краткую справку в виде комментария в тестирующей программе.

Ответ

Листинг 5: ./scripts/lab3_zad5.R

```
1 library( stringi )  # подключает пакет package
2
3 from <- -6
4 to <- 6
5 by <- 0.5
6 print(seq( from , to , by = by ))  # генерирует последовательность чисел
7                                     # от from до to с шагом by
8
9 from <- -10
10 to <- 50
11 len <- 6
12 print(seq( from , to , len = len ))  # генерирует последовательность
13                                     # чисел от from до to длины len
14
15 x <- 7
16 times <- 3
17 rep(x, times)  # повторяет x ровно times раз
18
19 list( " string ", 12L, 12.2, FALSE, NULL)  # создаёт список объектов
20
21 x <- 1:10
22 rev(x)  # реверсирует порядок элементов
23
24 sort(x)  # сортирует элементы объекта по возрастанию
25
26 breaks <- 5
27 cut(x, breaks)  # делит вектор на равные интервалы
28
29 y <- seq(0, 9, by = 2)
30 match(y, x)  # ищет элементы x, которые есть в y
31
32 x <- rep(1:4, 5)
33 unique(x)  # исключает из объекта повторяющиеся элементы
```

```

1 [1] -6.0 -5.5 -5.0 -4.5 -4.0 -3.5 -3.0 -2.5 -2.0 -1.5 -1.0 -0.5
   0.0  0.5  1.0
2 [16]  1.5  2.0  2.5  3.0  3.5  4.0  4.5  5.0  5.5  6.0
3 [1] -10   2  14  26  38  50
4 [1]  7  7  7
5 [[1]]
6 [1] "string"
7
8 [[2]]
9 [1] 12
10
11 [[3]]
12 [1] 12.2
13
14 [[4]]
15 [1] FALSE
16
17 [[5]]
18 NULL
19
20 [1] 10  9  8  7  6  5  4  3  2  1
21 [1]  1  2  3  4  5  6  7  8  9 10
22 [1] (0.991,2.8] (0.991,2.8] (2.8,4.6] (2.8,4.6] (4.6,6.4]
   (4.6,6.4]
23 [7] (6.4,8.2] (6.4,8.2] (8.2,10] (8.2,10]
24 Levels: (0.991,2.8] (2.8,4.6] (4.6,6.4] (6.4,8.2] (8.2,10]
25 [1] NA  2  4  6  8
26 [1] 1 2 3 4

```