

Challenge-5

Lew Bo Cong

2023-09-11

Questions

Question-1: Local Variable Shadowing Create an R function that defines a global variable called `x` with a value of 5. Inside the function, declare a local variable also named `x` with a value of 10. Print the value of `x` both inside and outside the function to demonstrate shadowing.

Solutions:

```
# Enter code here
x <- 5

glob_var <- function(x){
  x <- 10
  print(x)
}

print(x)
```

```
## [1] 5
```

```
glob_var(x)
```

```
## [1] 10
```

Question-2: Modify Global Variable Create an R function that takes an argument and adds it to a global variable called `total`. Call the function multiple times with different arguments to accumulate the values in `total`.

Solutions:

```
# Enter code here
total <- 0
mod_global <- function(add_to_global){
  total <-< total + add_to_global
  return(total)
}

mod_global(1)
```

```
## [1] 1
```

```
mod_global(3)
```

```
## [1] 4
```

```
mod_global(10)
```

```
## [1] 14
```

```
mod_global(15)
```

```
## [1] 29
```

Question-3: Global and Local Interaction Write an R program that includes a global variable `total` with an initial value of 100. Create a function that takes an argument, adds it to `total`, and returns the updated `total`. Demonstrate how this function interacts with the global variable.

Solutions:

```
# Enter code here
```

```
total <- 100
```

```
add_num <- function(num){  
  total <- total + num  
  return(total)  
}
```

```
add_num(10)
```

```
## [1] 110
```

```
add_num(15)
```

```
## [1] 125
```

Question-4: Nested Functions Define a function `outer_function` that declares a local variable `x` with a value of 5. Inside `outer_function`, define another function `inner_function` that prints the value of `x`. Call both functions to show how the inner function accesses the variable from the outer function's scope.

Solutions:

```
# Enter code here
```

```
outer_function <- function(){  
  x <- 5  
  inner_function <- function() print(x)  
  
  inner_function()  
}
```

```
outer_function()
```

```
## [1] 5
```

```
inner_function()
```

```
## Error in inner_function(): could not find function "inner_function"
```

Question-5: Meme Generator Function Create a function that takes a text input and generates a humorous meme with the text overlaid on an image of your choice. You can use the `magick` package for image manipulation. You can find more details about the commands offered by the package, with some examples of annotating images here: <https://cran.r-project.org/web/packages/magick/vignettes/intro.html>

Solutions:

```
# Enter code here
```

```
library(magick)
```

```
## Linking to ImageMagick 6.9.12.93
```

```
## Enabled features: cairo, freetype, fftw, ghostscript, heic, lcms, pango, raw, rsvg, webp
```

```
## Disabled features: fontconfig, x11
```

```
meme_generator <- function(text_input){  
  meme <- image_read("buttery meme credits imgflip.jpg")  
  meme_output <- image_annotate(meme, text_input, size = 15, color = "white",  
                                boxcolor = "black", location = "+320+200")  
  return(meme_output)  
}
```

```
meme_generator("is this Python")
```



Question-6: Text Analysis Game Develop a text analysis game in which the user inputs a sentence, and the R function provides statistics like the number of words, characters, and average word length. Reward the user with a “communication skill level” based on their input.

Solutions:

```
# Enter code here  
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v dplyr      1.1.3      v readr      2.1.4  
## v forcats    1.0.0      v stringr   1.5.0  
## v ggplot2    3.4.3      v tibble    3.2.1  
## v lubridate  1.9.2      v tidyr     1.3.0  
## v purrr      1.0.2  
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()      masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
text_game <- function(sentence){
  # sentence <- readline("Enter a sentence:") <-- Commented out to demo the output
  num_words <- lengths(strsplit(sentence, ' '))
  num_chars <- nchar(gsub(" ", "", sentence))
  ave_word_length <- num_chars/num_words
  comms_skill <- case_when(between(ave_word_length, 0, 5) ~ "Good.",
                           between(ave_word_length, 5, 10) ~ "Great!",
                           ave_word_length > 10 ~ "Excellent!")

  cat("Your sentence has ", num_words, " words, ", num_chars,
      " characters, and the average word length is ", ave_word_length,
      ".", "\n", "Your communication skill level is: ", comms_skill, sep = "")
}

text_game("Hello my name is Bo Cong")
```

```
## Your sentence has 6 words, 19 characters, and the average word length is 3.166667.
## Your communication skill level is: Good.
```