

Challenge-5

Sie Choon Hong

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
# Global Variable
x <- 5
new_function <- function(x) {
  #Local Variable
  x <- 10
  return(x)
}
x
```

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

```
new_function(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
accumulator <- function(x) {
  # Assign the argument `x` to the global variable `total`
  total <- total + x
}

# Initialize `total` to 0
total <- 0

# Call the function multiple times with different arguments
accumulator(1)
accumulator(2)
accumulator(3)

# Print the value of `total`
print(total)

```

```
## [1] 6
```

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

accumulator <- function(x) {
  # Add the argument to the global variable `total`
  total <- total + x
}

total

```

```
## [1] 100
```

```

# Call the function with the argument 10
total <- accumulator(10)
total

```

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

```

# Call the function with the argument 20
total <- accumulator(20)
total

```

```
## [1] 130
```

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()
}

# Call the function `outer_function`
outer_function()
```

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

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> (<https://cran.r-project.org/web/packages/magick/vignettes/intro.html>)

Solutions:

```
# Enter code here

library(magick)
```

```
## Warning: package 'magick' was built under R version 4.2.3
```

```
## Linking to ImageMagick 6.9.12.93
## Enabled features: cairo, freetype, fftw, ghostscript, heic, lcms, pango, raw, rsvg, webp
## Disabled features: fontconfig, x11
```

```
# Function to generate a meme
meme_generator <- function(text, image_path) {
  # Load the image file
  image <- image_read(image_path)

  # Add the text to the image
  image_annotate(image, text, size = 50, gravity = "center", color = "red")
}

# Call the function 'meme_generator'
meme_generator("Ahhhh!", "https://raw.githubusercontent.com/R-CoderDotCom/samples/main/homer.gif")
```



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 —
## ✓ dplyr      1.1.2      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr    1.5.0
## ✓ ggplot2    3.4.3      ✓ tibble     3.2.1
## ✓ lubridate  1.9.2      ✓ tidyr      1.3.0
## ✓ 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 be
come errors
```

```

# Function to analyze text
play_game <- function(user_input) {
  #user_input <- readline(prompt="Enter your sentence:")
  # Get the number of words
  num_words <- strsplit(user_input, " ")[[1]] %>% length()

  # Get the number of characters
  num_characters <- nchar(user_input)

  # Get the average word length
  average_word_length <- num_characters / num_words

  # Calculate the communication skill level
  communication_skill_level <- case_when(
    num_words >= 4 & average_word_length >= 1 ~ "Expert",
    num_words >= 3 & average_word_length >= 1 ~ "Good",
    num_words >= 2 & average_word_length >= 1 ~ "Average",
    TRUE ~ "Needs Improvement"
  )
  # Return the results
  return(list(
    num_words = num_words,
    num_characters = num_characters,
    average_word_length = average_word_length,
    communication_skill_level = communication_skill_level
  ))
}

# Call the function 'text_analysis'
play_game("This should be Expert.")

```

```

## $num_words
## [1] 4
##
## $num_characters
## [1] 22
##
## $average_word_length
## [1] 5.5
##
## $communication_skill_level
## [1] "Expert"

```