# Challenge-5

Jaren Ong

2023-09-13

## 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:**

```
# Define a global variable x with a value of 5
x <- 5

# Define a function that demonstrates variable shadowing
shadowing_function <- function() {
  # Declare a local variable x with a value of 10 (shadows the global x)
  x <- 10

  # Print the value of x inside the function (local x)
  cat("Inside the function, x =", x, "\n")
}

# Call the function
shadowing_function()
```

```
## Inside the function, x = 10
```

```
# Print the value of x outside the function (global x)
cat("Outside the function, x =", x, "\n")
```

```
## Outside the function, x = 5
```

**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:**

```
# Initialize the global variable total
total <- 0

# Define a function to add a value to the global total
```

```r
add_to_total <- function(value) {
  # Use the <<- operator to modify the global variable
  total <<- total + value
}

# Call the function multiple times with different arguments
add_to_total(5)

cat("1st Total is:", total, "\n")
```

```
## 1st Total is: 5
```

```r
add_to_total(10)

cat("2nd Total is:", total, "\n")
```

```
## 2nd Total is: 15
```

```r
add_to_total(7)

cat("3rd Total is:", total, "\n")
```

```
## 3rd Total is: 22
```

```r
# Print the accumulated total
cat("The accumulated total is:", total, "\n")
```

```
## The accumulated total is: 22
```

**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:**

```r
# Initialize the global variable total
total <- 100

# Define a function to add a value to total and return the updated total
add_to_total_and_return <- function(value) {
  # Use the <<- operator to modify the global variable
  total <<- total + value

  # Return the updated total
  return(total)
}

# Print the initial value of total
cat("Initial total:", total, "\n")
```

```
## Initial total: 100
```

```r
# Call the function with an argument and store the updated total
new_total <- add_to_total_and_return(25)

# Print the updated total
cat("Updated total:", new_total, "\n")
```

```
## Updated total: 125
```

```r
# Demonstrate that the global total has been updated
cat("Global total after function call:", total, "\n")
```

```
## Global total after function call: 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:**

```r
# Define the outer function
outer_function <- function() {
  # Declare a local variable x with a value of 5
  x <- 5

  # Define the inner function
  inner_function <- function() {
    # Print the value of x from the outer function's scope
    cat("Value of x from inner_function:", x, "\n")
  }

  # Call the inner function
  inner_function()

  # Print a message from the outer function
  cat("Message from outer_function\n")
}

# Call the outer function
outer_function()
```

```
## Value of x from inner_function: 5
## Message from outer_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:**

```r
# Enter code here
library(magick)

generate_meme <- function(input_text, image_path) {
  # Load the image
  img <- image_read(image_path)

  # Define text properties
  text_color <- "white"
  font_size <- 40
  text_x <- 50
  text_y <- 50

  # Annotate the image with text
  image_annotate(
    img,
    input_text,
    size = font_size,
    color = text_color,
    location = "+x+y",
    gravity = "northwest",
    boxcolor = "black",
    strokecolor = "black",
  )
}

# Example usage
input_text <- "HELP I DONT UNDERSTAND 2207 :("
image_path <- "//Users/jaren/Desktop/NM2207/W5/Meme Picture.png"
generate_meme(input_text, image_path)
```

**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:**

```r
# Define a function for the text analysis game
text_analysis_game <- function() {
  # Get user input
  user_input <- readline(prompt = "Enter a sentence: ")

  # Split the sentence into words
  words <- unlist(strsplit(user_input, " "))

  # Calculate the number of words
  num_words <- length(words)

  # Calculate the number of characters
  num_chars <- nchar(user_input)

  # Calculate the average word length
  avg_word_length <- sum(nchar(words)) / num_words
```

```
  # Determine communication skill level based on average word length
  communication_skill_level <- ifelse(avg_word_length < 4, "Novice",
                                      ifelse(avg_word_length < 6, "Intermediate", "Advanced"))

  # Display statistics and communication skill level
  cat("\nStatistics:\n")
  cat("Number of words:", num_words, "\n")
  cat("Number of characters:", num_chars, "\n")
  cat("Average word length:", round(avg_word_length, 2), "\n")
  cat("\nCommunication Skill Level:", communication_skill_level, "\n")
}

# Run the text analysis game
text_analysis_game()
```

```
## Enter a sentence:
##
## Statistics:
## Number of words: 0
## Number of characters: 0
## Average word length: NaN
##
## Communication Skill Level: NA
```