Challenge-5

CHENJUNYURYAN

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:

```
x <- 5 #global variable that is define outside of function
shadowing_function <- function(x) {x <- 10} #local variable that is define within function
print(x)
```

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

```
print(shadowing_function(x))
```

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

```
# variables initialise inside the function will not be available outside the function
# sprintf
```

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:

```
total <- 0 # global variable that is define outside of function
add_to_total <- function(x){ total <<- total + x}
add_to_total(5) #call function with argument 5 (after inputting this function does (global) total <- 5

print(total) # total <- (0+5)
```

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

```
add_to_total(10) #call function with argument 10 (after inputting this function is the (global) total <

print(total) # using last total which was (0+5) hence total <- ((0+5)+10)
```

```
## [1] 15
```

```r
add_to_total(15) #argument 3 is 15, call function with argument 15

print(total)
```

```
## [1] 30
```

```r
# (<<-) creates a global variable inside a function
```

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
total <- 100 #global variable
updated_total <- function(x){total <- total+x
return(total)} #dont understand the return part help ! so it


print(updated_total(50))
```

```
## [1] 150
```

```r
print(total)
```

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

```r
#return, returns the last output of a function atuomatically
```

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
outer_function <- function() {x <- 5
inner_function <- function(){print(x)}
inner_function() }  # local variable

#what is the difference between function() and function(x), function() is fixed whereas function(x)
print(outer_function())
```

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

```r
?return
```

```
## starting httpd help server ... done
```

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
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
```

```r
create_meme <- function(text,image_path,output_path) {
  meme <- image_read(image_path)
  meme <- image_annotate(meme,text,gravity="center",size = 30, color = "black")
  image_write(meme, path = output_path)
}

create_meme("A potato flew around my room","POTATO.jpg","output_meme.jpg")
```

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
text_analysis_game <- function() {
  cat("Welcome to the Text Analysis Game!\n")

  # Prompt the user for input
  user_input <- readline(prompt = "Enter a sentence: ")

  # Tokenize the input sentence into words
  words <- unlist(strsplit(user_input, "\\s+"))
  # Calculate statistics
  num_words <- length(words)
  num_characters <- nchar(user_input)
  average_word_length <- mean(nchar(words))

  # Determine communication skill level
  skill_level <- if (average_word_length >= 7) {
    "Excellent"
  } else if (average_word_length >= 5) {
    "Good"
  } else {
    "Needs Improvement"
  }

  # Display the statistics and reward
  cat("\n--- Text Analysis Results ---\n")
  cat("Number of words: ", num_words, "\n")
  cat("Number of characters: ", num_characters, "\n")
  cat("Average word length: ", round(average_word_length, 2), "\n")
  cat("Communication skill level: ", skill_level, "\n")

  # Provide feedback based on skill level
  cat("\nCongratulations! Your communication skills are ", skill_level, ".\n")
}

# Run the game
```