Challenge-5

Wu Xingyi (Kyra)

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:

```
x <- 5
sprintf("The value assigned to x outside the function is %d",x)
```

```
## [1] "The value assigned to x outside the function is 5"
```

```
inside_function <- function() {
  x <- 10
  message(sprintf("The value of x inside the function is %d", x))
}
sprintf("The value assigned to x outside the function is %d",x)
```

```
## [1] "The value assigned to x outside the function is 5"
```

```
inside_function()
```

```
## The value of x inside the function is 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:

```
total <- 0

accumulated_total <- function(value) {
  total <<- total + value
}

accumulated_total(5)
accumulated_total(10)
accumulated_total(7)

print(sprintf("Total: %d", total))
```

```
## [1] "Total: 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
# Interpreted this question's "updated total" differently from the previous questions
# "accumulated total" where we had to change the value of the global total versus
# just adding numbers on a one-time basis to the total.
total <- 100

updated_total <- function(val) {
  return(total + val)
}

updated_total(1)
```

```
## [1] 101
```

```r
updated_total(10)
```

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

```r
updated_total(100)
```

```
## [1] 200
```

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()
}

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

Solutions:

```
#install.packages("magick")
library(magick)
```

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

```
meme = function(text) {
  frink <- image_read("https://jeroen.github.io/images/frink.png")
  frink <- image_border(image_background(frink, "hotpink"), "#000080", "20x10")
  frink <- image_negate(frink)
  frink <- image_annotate(frink, text = text, size = 25, color = "red", boxcolor = "pink",
    degrees = 30, location = "+60+70")
  print(frink)
}

meme("me on mondays")
```

```
##   format width height colorspace matte filesize density
## 1    PNG   260    465       sRGB  TRUE        0   72x72
```

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:

```
# Unfortunately, the TA and I were unable to figure out how to make the knitted PDF
# interactive to allow for user input :<
analyze_text <- function(sentence) {
  words <- unlist(strsplit(sentence, " "))
  num_words <- length(words)
  num_chars <- nchar(sentence)
  skill_level <- ifelse(num_words <= 5, "Beginner",
                        ifelse(num_words <= 10, "Intermediate",
```

```
                                ifelse(num_words <= 15, "Advanced")))

  summary <- paste(
    "Number of Words:", num_words,
    "\nNumber of Characters:", num_chars,
    "\nCommunication Skill Level:", skill_level
  )

  return(summary)
}

sentence <- readline(prompt = "Enter a sentence to analyze: ")
```

```
## Enter a sentence to analyze:
```

```
analysis_result <- analyze_text(sentence)

cat("\nAnalysis Result:\n")
```

```
##
## Analysis Result:
```

```
cat(analysis_result,"\n")
```

```
## Number of Words: 0
## Number of Characters: 0
## Communication Skill Level: Beginner
```