Challenge-5

Wong Zi Xin

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

my_function <- function(){
  x <- 10
  print(paste("Inside function: x=", x))
}

my_function()
```

```
## [1] "Inside function: x= 10"
```

```
print(paste("Outside function: x=", x))
```

```
## [1] "Outside 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:

```
# Enter code here
total <- 0

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

add_total(5)
add_total(10)
add_total(7)

print(total)
```

```
## [1] 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:

```
# Enter code here
global <- 100

add_100 <- function(x) {
  global <- global + x
  return(global)
}

add_100(5)
```

```
## [1] 105
```

```
add_100(10)
```

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

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(paste("Inside function: x=", x))
  }
  print(paste("Outside function: x=", x))
  inner_function()
}

outer_function()
```

```
## [1] "Outside function: x= 5"
## [1] "Inside function: x= 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:

```
# Enter code here
mirror_url <- "https://cran.rstudio.com/"
options(repos = mirror_url)

install.packages("magick")
```

```
##
## The downloaded binary packages are in
##  /var/folders/bm/tyrx6fqs2jqdgysxknljvnq40000gn/T//RtmpbJmueM/downloaded_packages
```

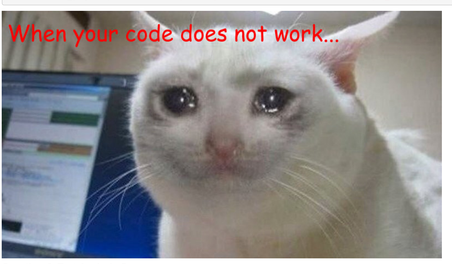
```
library(magick)
```

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

```
generate_meme <- function(text, output_path) {
  image_path <- "/Users/zixinwong/Downloads/crying.jpg"
  meme <- image_read(image_path)

  meme <- image_annotate(
    meme,
    text = text,
    color = "red",
    size = 40,
    location = "+10+10",
    font = "Comic Sans"
  )

  image_write(meme, path = output_path)

  print(paste("Meme generated and saved to:", output_path))
}

text_to_overlay <- "When your code does not work..."
output_image_path <- "output_meme.jpg"

generate_meme(text_to_overlay, output_image_path)
```

```
## [1] "Meme generated and saved to: output_meme.jpg"
```

```
knitr::include_graphics("/Users/zixinwong/Desktop/NUS/Academics/CNM/NM2207/Week-5/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:

```
# Enter code here
text_analysis_game <- function(user_input) {
  number_of_characters <- nchar(user_input)
  words <- strsplit(user_input, "\\s+")
  number_of_words <- length(words[[1]])
  avg_word_length <- number_of_characters / number_of_words

  result <- list(
    number_of_characters = number_of_characters,
    number_of_words = number_of_words,
    avg_word_length = avg_word_length
  )

  skill_level <- "Beginner"
  if (avg_word_length > 5) {
    skill_level <- "Intermediate"
  }
  if (avg_word_length > 10) {
    skill_level <- "Advanced"
  }

  result$skill_level <- skill_level

  return(result)
}

user_input <- "I live in Singapore."
result <- text_analysis_game(user_input)

cat("Statistics for your sentence:\n")
```

```
## Statistics for your sentence:
```

```
cat(paste("Number of characters:", result$number_of_characters), "\n")
```

```
## Number of characters: 20
```

```
cat(paste("Number of words:", result$number_of_words), "\n")
```

```
## Number of words: 4
```

```
cat(paste("Average word length:", round(result$avg_word_length, 2), "characters"), "\n")
```

```
## Average word length: 5 characters
```

```
cat(paste("Communication Skill Level:", result$skill_level), "\n")
```

```
## Communication Skill Level: Beginner
```