

Challenge-5

Guan Ziwen

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:

```
# Enter code here
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"
```

```
fn <- function(x){
  x <- 10
  return(sprintf("The value assigned to x inside the function is %d",x))
}
fn(x)
```

```
## [1] "The value assigned to 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:

```
# Enter code here
total <- 0
add_total<- function(x = 0){
  total <- x+total
  return(total)
}

add_total(7)
```

```
## [1] 7
```

```
add_total(15)
```

```
## [1] 22
```

```
add_total(5)
```

```
## [1] 27
```

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
add_total<- function(x){
  total <- x+total
  return(total)
}

add_total(7)
```

```
## [1] 107
```

```
add_total(50)
```

```
## [1] 157
```

```
add_total(100)
```

```
## [1] 257
```

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()
}

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:

```
# Enter code here
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

meme_fn <- function(text){
  image <- image_read("crying.jpg")
  image_annotate(image,text, size = 70, gravity = "south", color = "white")
  updated_image<-image_annotate(image,text, size = 70, gravity = "south", color = "white")
  image_write(updated_image, path = "meme.png", format = "png")
}
meme <- meme_fn("ME DOING R BE LIKE")
```

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

#The shorter the average word length, the higher the readability.

game <- function(sentence){
  nword <- lengths(strsplit(sentence, " "))
  ncharacter <- nchar(sentence)-nword+1
  avg_length <- ncharacter/nword
  print(paste0("You have used ",nword," words, and ",ncharacter," characters.
               You average word length is ",avg_length,"."))
  if(avg_length < 3){print("You are a genius!")}
  else if(avg_length<5){print("You can communicate well!")}
  else{print("Try harder!")}
}

game("The fox is red.")
```

```
## [1] "You have used 4 words, and 12 characters. \n
## [1] "You can communicate well!"
```

```
You average word length is 3."
```