Week2

Renato Erazo

8 de octubre de 2020

## Loop while

z <- 5  
while(z >=3 && z <= 10)  
{  
 print(z)  
 coin <- rbinom(1,1,0.5)  
   
 if(coin == 1){ #Random walk  
 z <- z+1  
   
 }else{  
 z <- z-1   
   
 }  
}

## [1] 5  
## [1] 6  
## [1] 5  
## [1] 4  
## [1] 3

## Functions

add2 <- function(x,y){  
 x+y  
}  
add2(4,3)

## [1] 7

# Valores ,mayores a 10 en un vector

above10 <- function(x){  
 use <- x > 10  
 x[use]  
}  
  
above <- function(x,n = 10){ #Default value 10  
 use <- x > n  
 x[use]  
}  
above(1:20,12)

## [1] 13 14 15 16 17 18 19 20

# The next function calculate the mean of columns in dataset o dataFrame

columMean <- function(x, removeNA = TRUE){  
 cn <- ncol(x)  
 means <- numeric(cn)  
 for (i in 1:cn)  
 {  
 means[i]<- mean(x[,i],na.rm = removeNA)  
 }  
 means  
}  
  
columMean(airquality)

## [1] 42.129310 185.931507 9.957516 77.882353 6.993464 15.803922

## Scoping

# -------- Binding --- the function search() show the available library

search()

## [1] ".GlobalEnv" "package:stats" "package:graphics"   
## [4] "package:grDevices" "package:utils" "package:datasets"   
## [7] "package:methods" "Autoloads" "package:base"

library("ggplot2")  
search()

## [1] ".GlobalEnv" "package:ggplot2" "package:stats"   
## [4] "package:graphics" "package:grDevices" "package:utils"   
## [7] "package:datasets" "package:methods" "Autoloads"   
## [10] "package:base"

# ---- Prueba de scoping

prueba <- function(x){  
 x <- x +1  
 x  
}  
valor <- 5  
prueba(valor)

## [1] 6

valor

## [1] 5

# ---- Enviroments, workspace, global, parent

x <- 1 # Global Segun yo  
y <- 2  
  
funcionAmbiente <- function(z){  
 x <- 3  
 print(c("-> En la función x", x))  
 print(c("-> En la función y", y))  
   
 anidada <- function(w){  
 print(c("--> Función Anidada suma x,y,z,w", x+y+z+w))  
 }  
   
 anidada(5)  
   
 x + y + z  
}  
funcionAmbiente(4)

## [1] "-> En la función x" "3"   
## [1] "-> En la función y" "2"   
## [1] "--> Función Anidada suma x,y,z,w" "14"

## [1] 9

print(c("> Fuera de la función luego de ejecutarla x", x))

## [1] "> Fuera de la función luego de ejecutarla x"  
## [2] "1"

# Lexcial scoping

make.power <- function(n){  
 pow <- function(x){  
 x\*n  
 }  
 pow  
}  
  
cube <- make.power(3)  
square <- make.power(2)  
otro <- make.power(4)  
cube(3)

## [1] 9

square(2)

## [1] 4

otro(2)

## [1] 8

# Explorar las variables de ambiente

ls(environment(cube))

## [1] "n" "pow"

get("n",environment(cube))

## [1] 3

# ----- Forware functions

y <- 10  
f <- function(x){  
 y <- 2  
 y^2 + g(x)  
}  
  
g <- function(x){  
 x\*y  
}  
  
f(3)

## [1] 34