

Laborator 04

Petculescu Mihai-Silviu

Laborator 04

Petculescu Mihai-Silviu

Funcții

Structuri de control

if-else

switch

for

while

repeat

break and next

Funcții

```
> f = function(){  
  # 0 functie fara parametri  
}  
> class(f)  
[1] "function"  
> f()  
NULL
```

```
# Functie cu parametri  
> f = function(mesaj){ nchar(mesaj) }  
> l_mesaj = f("Curs de statistica si probabilitati")  
> l_mesaj  
[1] 35  
> f()  
Error in nchar(mesaj) : argument "mesaj" is missing, with no default  
  
# Functie cu parametri default  
> f = function(mesaj = "Text default"){ nchar(mesaj) }  
> f()  
[1] 12
```

Să presupunem că Jack Sparrow este convins că poate prezice cât aur va găsi pe o insulă folosind următoarea ecuație: $ab - 324c + \log(a)$, unde a este aria insulei (în m^2), b este numărul de copaci de pe insulă iar c reprezintă cât de beat este pe o scală de la 1 la 10. Creați o funcție numită `Jacks.Moneycare` primește ca argumente a , b , c și întoarce valoarea prezisă.

```
> Jacks.Money = function(a, b, c){ a*b - 324*c + log(a) }  
> Jacks.Money(a = 1000, b = 30, c = 7)  
# Jacks.Money(1000, 30, 7)  
[1] 27738.91
```

Structuri de control

if-else

```
> x = runif(1, 0, 10)
> x
[1] 3.166125
> if(x > 3) y = 10 else y = 0
> y
[1] 10
```

switch

```
> nr1 = 20
> nr2 = 30
> operator = "*"
> switch(operator,
  "+" = print(paste("Suma celor 2 numere: ", nr1 + nr2)),
  "-" = print(paste("Diferenta celor 2 numere: ", nr1 - nr2)),
  "*" = print(paste("Inmultirea celor 2 numere: ", nr1 * nr2)),
  "^" = print(paste("Ridicarea la putere a celor 2 numere: ", nr1 ^ nr2)),
  "/" = print(paste("Impartirea celor 2 numere: ", nr1 / nr2)),
  "%/%" = print(paste("Catul impartirii celor 2 numere: ", nr1 %/% nr2)),
  "%%" = print(paste("Restul impartirii celor 2 numere: ", nr1 %% nr2))
)
[1] "Inmultirea celor 2 numere: 600"
```

for

```
> x = c("a", "b", "c", "d")
> for(i in 1:4) print(x[i])
# for(i in seq_along(x)) print(x[i])
# for(letter in x) print(letter)
[1] "a"
[1] "b"
[1] "c"
[1] "d"
```

```
> x = matrix(1:6, 2, 3)
> for(i in seq_len(nrow(x)))
  for(j in seq_len(ncol(x)))
    print(x[i, j])
[1] 1
[1] 3
[1] 5
[1] 2
[1] 4
[1] 6
```

Construiți următoarele matrice de dimensiune 10×10: $M_{i,j} = \frac{1}{\sqrt{|i-j|+1}}$ și $N_{i,j} = \frac{i}{j^2}$. Puteți construi matricea M și matricea N fără a folosi bucle `for`? (Hint: ce face comanda `outer`?)

```
# cu functia for
> M = matrix(rep(0,100), 10, 10)
> N = matrix(rep(0,100), 10, 10)
> for(i in 1:10)
  for(j in 1:10){
    M[i,j] = 1/sqrt(abs(i-j) + 1)
    N[i,j] = i/(j*j)
  }
```

```
# cu functia outer
> x = 1:10
> y = 1:10
> M = outer(x, y, function(x,y){ 1/sqrt(abs(x - y)+1) })
> N = outer(x, y, function(x,y){ x/(y*y)})
```

```
# Primele 3 zecimale (fara rotunjire)
> floor(M*1000)/1000
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] 1.000 0.707 0.577 0.500 0.447 0.408 0.377 0.353 0.333 0.316
[2,] 0.707 1.000 0.707 0.577 0.500 0.447 0.408 0.377 0.353 0.333
[3,] 0.577 0.707 1.000 0.707 0.577 0.500 0.447 0.408 0.377 0.353
[4,] 0.500 0.577 0.707 1.000 0.707 0.577 0.500 0.447 0.408 0.377
[5,] 0.447 0.500 0.577 0.707 1.000 0.707 0.577 0.500 0.447 0.408
[6,] 0.408 0.447 0.500 0.577 0.707 1.000 0.707 0.577 0.500 0.447
[7,] 0.377 0.408 0.447 0.500 0.577 0.707 1.000 0.707 0.577 0.500
[8,] 0.353 0.377 0.408 0.447 0.500 0.577 0.707 1.000 0.707 0.577
[9,] 0.333 0.353 0.377 0.408 0.447 0.500 0.577 0.707 1.000 0.707
[10,] 0.316 0.333 0.353 0.377 0.408 0.447 0.500 0.577 0.707 1.000

> floor(N*1000)/1000
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]      1 0.25 0.111 0.062 0.04 0.027 0.020 0.015 0.012 0.01
[2,]      2 0.50 0.222 0.125 0.08 0.055 0.040 0.031 0.024 0.02
[3,]      3 0.75 0.333 0.187 0.12 0.083 0.061 0.046 0.037 0.03
[4,]      4 1.00 0.444 0.250 0.16 0.111 0.081 0.062 0.049 0.04
[5,]      5 1.25 0.555 0.312 0.20 0.138 0.102 0.078 0.061 0.05
[6,]      6 1.50 0.666 0.375 0.24 0.166 0.122 0.093 0.074 0.06
[7,]      7 1.75 0.777 0.437 0.28 0.194 0.142 0.109 0.086 0.07
[8,]      8 2.00 0.888 0.500 0.32 0.222 0.163 0.125 0.098 0.08
[9,]      9 2.25 1.000 0.562 0.36 0.250 0.183 0.140 0.111 0.09
[10,]     10 2.50 1.111 0.625 0.40 0.277 0.204 0.156 0.123 0.10
```

while

```
> z = 5
> set.seed(123)
> while(z >= 3 && z <= 10){
  coin = rbinom(1, 1, 0.5)
  if(coin == 1) z = z + 1
  else z = z - 1
}
> print(z)
[1] 11
```

repeat

```
> x = 12345
> s = 0
> repeat{
  if(x > 0){
    s = s + x %% 10;
    x = x %/% 10
  } else break
}
> s
[1] 15
```

break and next

```
> for(i in 1:20) {
  if(i %% 3 == 0) next
  if(i > 16) break
  print(i)
}
[1] 1
[1] 2
[1] 4
[1] 5
[1] 7
[1] 8
[1] 10
[1] 11
[1] 13
[1] 14
[1] 16
```