

Trabalhando com estruturas de controle

Ivanildo Batista

7 de marco de 2021

Trabalhando com estruturas de controle IF, ELSE, FOR e WHILE

IF-ELSE

```
x=25#definido uma variavel
```

Se x for menor que 30, imprimir a frase em chaves

```
if (x<30)
{"Este numero e menor que 30"}
```

```
## [1] "Este numero e menor que 30"
```

ou

```
if (x<30)
  "Este numero e menor que 30"
```

```
## [1] "Este numero e menor que 30"
```

ELSE seria o "caso contrario" ou o "senao":

```
if(x<7){
  "Este numero e menor que 7"
}else{
  "Este numero nao e menor que 7"
}
```

```
## [1] "Este numero nao e menor que 7"
```

Comandos aninhados

```
x=7

if (x<7){
  "Este numero e menor que 7"
}else if (x==7){
  "Este numero e o 7"
}else{
  "Este numero nao e menor que 7"
}
```

```
## [1] "Este numero e o 7"
```

IFELSE

```
x=5
```

```
ifelse(x<6, "Correto!",NA) #vai imprimir a mensagem
```

```
## [1] "Correto!"
```

```
x = 9
```

```
ifelse(x<6, "Correto!", NA) #nao vai imprimir a mensagem
```

```
## [1] NA
```

IFELSE aninhados

```
x = c(5,6,4)
```

```
ifelse(x<5, "menor que 5",  
      ifelse(x==5,"Igual a 5",'Maior que 5'))
```

```
## [1] "Igual a 5" "Maior que 5" "menor que 5"
```

Dentro de funcoes

```
func1 = function(x,y){  
  ifelse(y<7, x+y, 'Nao encontrado')  
}
```

```
func1(4,2) #vai somar os valores, pois x e menor que 7
```

```
## [1] 6
```

```
func1(40,7) #nao vai somar os valores, pois x e maior que 7
```

```
## [1] "Nao encontrado"
```

Funcao rep()

```
rep(rnorm(10),5) #vai repetir os 10 valores gerados pela distribuicao normal 5x
```

```
## [1] 1.890728637 -0.637849211 -0.411073639 -0.755857569 -2.383960112
## [6] -0.690043158 -0.166062737 0.003247326 -0.839807997 1.201105134
## [11] 1.890728637 -0.637849211 -0.411073639 -0.755857569 -2.383960112
## [16] -0.690043158 -0.166062737 0.003247326 -0.839807997 1.201105134
## [21] 1.890728637 -0.637849211 -0.411073639 -0.755857569 -2.383960112
## [26] -0.690043158 -0.166062737 0.003247326 -0.839807997 1.201105134
## [31] 1.890728637 -0.637849211 -0.411073639 -0.755857569 -2.383960112
## [36] -0.690043158 -0.166062737 0.003247326 -0.839807997 1.201105134
## [41] 1.890728637 -0.637849211 -0.411073639 -0.755857569 -2.383960112
## [46] -0.690043158 -0.166062737 0.003247326 -0.839807997 1.201105134
```

Funcao repeat()

A partir de um valor inicial (x=1) vai fazer o processo varias vezes ate chegar no valor de parada, que no nosso caso e 99.

```
x=1
```

```
repeat{
  x=x+3
  if(x>99)
    break
  print(x)
}
```

```
## [1] 4
## [1] 7
## [1] 10
## [1] 13
## [1] 16
## [1] 19
## [1] 22
## [1] 25
## [1] 28
## [1] 31
## [1] 34
## [1] 37
## [1] 40
## [1] 43
## [1] 46
## [1] 49
## [1] 52
## [1] 55
## [1] 58
## [1] 61
## [1] 64
## [1] 67
## [1] 70
## [1] 73
## [1] 76
## [1] 79
## [1] 82
## [1] 85
## [1] 88
## [1] 91
## [1] 94
## [1] 97
```

LOOP FOR

```
for (i in 1:20){print(i)} #dados os valores dentro do intervalo, todos serao impressos
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
```

```
for (q in rnorm(10)){print(i)} #imprimir todos os 10 valores gerados pela distribuicao normal
```

```
## [1] 20
## [1] 20
## [1] 20
## [1] 20
## [1] 20
## [1] 20
## [1] 20
## [1] 20
## [1] 20
## [1] 20
```

Ignorando elementos de um *loop*

```
for(i in 1:20){
  if(i==13 | i==15)
    next
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 14
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
```

Imprimir todos os valores, mas com excecao dos valores 13 e 15.

Interromper *loop*

```
for(i in 1:22){
  if(i==13)
    break
  print(i)
} #dos elementos dos intervalos, quando chegar em 12 a impressao do valores ira parar
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
```

LOOP WHILE

```
x=1
```

```
while(x<5){
  x=x+1
  print(x)
} #imprimir valores menores iguais a 5
```

```
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5
```

Quando o *while* nao sera implementado

```
y = 6
```

```
while(y<5){  
  y=y+10  
  print(y)}
```

Nao vai funcionar, pois *y* deve ser menor que 5, porem e igual a 6 em um *loop while* sempre lembrar de colocar o mecanismo de interrompimento para o *loop* nao ficar sendo gerado infinitamente.