



R for Data Science

Bài 4: Cấu trúc lặp

Phòng LT & Mạng

<https://csc.edu.vn/lap-trinh-va-csd/R-Programming-Language-for-Data-Science> 190

2020



Giới thiệu



- ☐ Một cấu trúc lặp gồm một câu lệnh hay một khối lệnh sẽ thi hành lặp lại cho tới khi biểu thức điều kiện sai
- ☐ Có ba loại cấu trúc lặp trong R
 - repeat
 - while
 - for



Nội dung

1. repeat
2. while
3. for
4. break & next

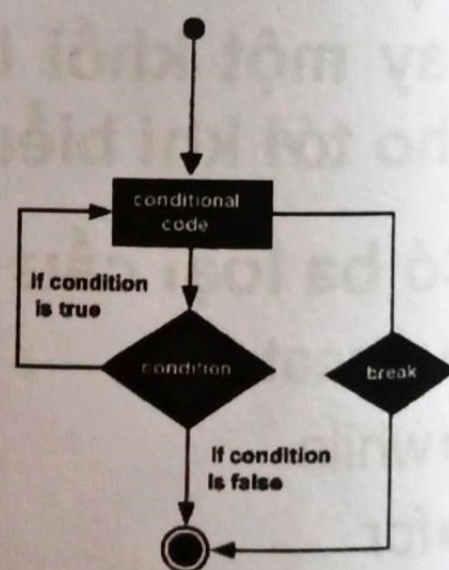


R programming language for Data Science

repeat

□ Lặp đi lặp lại các lệnh cho đến khi gặp một điều kiện dừng

```
repeat {
  commands
  if(condition) {
    break
  }
}
```



R programming language for Data Science

• Ví dụ

```
start <- 1
repeat {
  print(paste("Time: ", start))
  start <- start + 1
  if(start > 5){
    break
  }
}
```

```
[1] "Time: 1"
[1] "Time: 2"
[1] "Time: 3"
[1] "Time: 4"
[1] "Time: 5"
```

```
x <- 1
repeat{
  print(x)
  x <- x + 1
  if(x > 10){
    break
  }
}
```

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



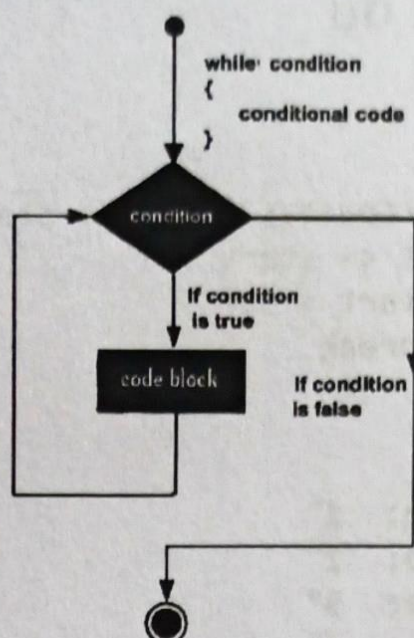
Nội dung

1. repeat
2. while
3. for
4. break & next



❑ Lặp đi lặp lại các lệnh khi còn thỏa điều kiện.
Vòng lặp dừng lại khi không thỏa điều kiện.

```
while (test_expression) {
  statement
}
```



• Ví dụ:

```
start <- 1
while(start < 6) {
  print(paste("Time: ", start))
  start <- start + 1
}
```

```
[1] "Time: 1"
[1] "Time: 2"
[1] "Time: 3"
[1] "Time: 4"
[1] "Time: 5"
```

```
i = 1
while(i==1){
  print("Hello")
  i <-
  readline(prompt="Continue or
not? 1: Continue, <> 1: Stop ")
  i <- as.numeric(i)
}
```

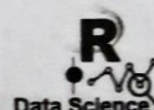
```
[1] "Hello"
Continue or not? 1: Continue, <> 1: Stop 1
[1] "Hello"
Continue or not? 1: Continue, <> 1: Stop 1
[1] "Hello"
Continue or not? 1: Continue, <> 1: Stop 1
[1] "Hello"
Continue or not? 1: Continue, <> 1: Stop 0
```



1. repeat
2. while
3. for
4. break & next



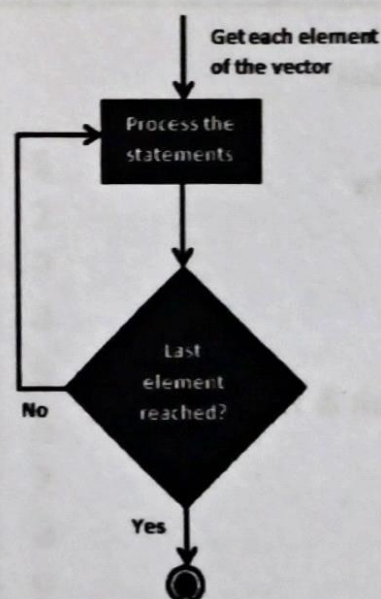
for



❑ Thực hiện công việc lặp khi biết số lần lặp

❑ Lặp các phần tử trong một danh sách

```
for (value in vector) {  
  statements  
}
```



● Ví dụ:

```
v <- LETTERS[1:10]
for (c in v){
  print(c)
}
```

```
[1] "A"
```

```
[1] "B"
```

```
[1] "C"
```

```
[1] "D"
```

```
[1] "E"
```

```
[1] "F"
```

```
[1] "G"
```

```
[1] "H"
```

```
[1] "I"
```

```
[1] "J"
```

```
colors <- c("red", "green", "yellow", "blue", "black", "white")
```

```
for ( color in colors) {
```

```
  print(color)
```

```
}
```

```
[1] "red"
```

```
[1] "green"
```

```
[1] "yellow"
```

```
[1] "blue"
```

```
[1] "black"
```

```
[1] "white"
```



Nội dung

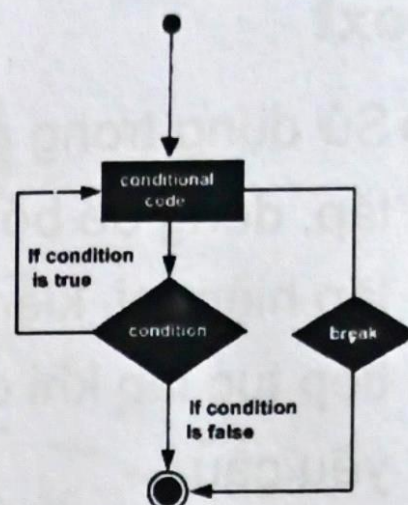
1. repeat
2. while
3. for
4. break & next



break & next

□ break

- Sử dụng trong cấu trúc lặp, dùng để thoát khỏi vòng lặp khi thỏa yêu cầu



break & next

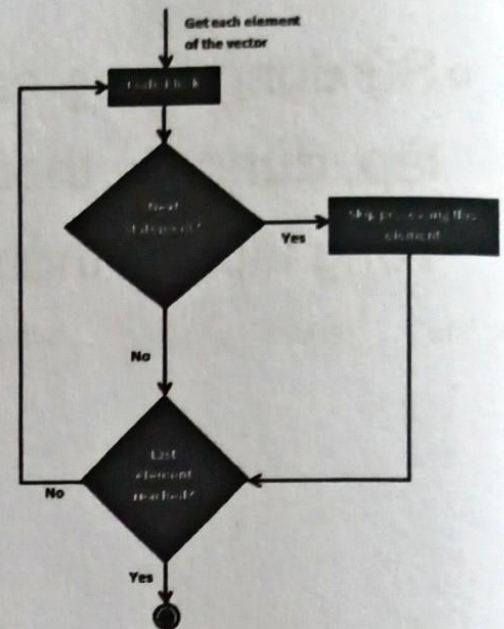
- Ví dụ:

<code>x <- 1</code>	<code>[1] 1</code>
<code>repeat{</code>	<code>[1] 2</code>
<code>print(x)</code>	<code>[1] 3</code>
<code>x <- x + 1</code>	<code>[1] 4</code>
<code>if(x > 10){</code>	<code>[1] 5</code>
<code>break</code>	<code>[1] 6</code>
<code>}</code>	<code>[1] 7</code>
<code>}</code>	<code>[1] 8</code>
	<code>[1] 9</code>
	<code>[1] 10</code>



□ next

- Sử dụng trong cấu trúc lặp, dùng để bỏ qua lần lặp hiện tại, kiểm tra và tiếp tục lặp khi còn thỏa yêu cầu



break & next

• Ví dụ

```

v <- LETTERS[1:10]
for ( i in v) {
  if (i == "E"){
    next
  }
  print(i)
}
    
```

```

[1] "A"
[1] "B"
[1] "C"
[1] "D"
[1] "F"
[1] "G"
[1] "H"
[1] "I"
[1] "J"
    
```





Chapter 4: Cấu trúc lặp

Exercise 1: Bạn muốn dừng lại hay chơi tiếp

- Chương trình yêu cầu người dùng nhập 1 để chơi tiếp (Continue) hoặc nhập khác 1 để dừng lại (Stop)
- Chương trình tiếp tục thực hiện nếu người dùng nhập 1 và dừng lại khi người dùng nhập khác 1.

Exercise 2: Bình phương vector

- Tạo một vector có n số ngẫu nhiên có giá trị trong khoảng từ 1..10 (n do người dùng nhập)
- Chương trình tính và in ra bình phương của các số ngẫu nhiên đã được tạo ra ở trên bằng cách sử dụng loop

Exercise 3: Bảng cửu chương

- Yêu cầu 1: Chương trình cho phép người dùng nhập vào một số. In ra cửu chương của số đó.
- Yêu cầu 2: Chương trình cho phép người dùng nhập vào số bắt đầu đến số kết thúc. In ra bảng cửu chương của các số trong khoảng từ số bắt đầu đến số kết thúc

Exercise 4: Tính giá trị

- Chương trình nhập vào một số n
- Chương trình tìm, tính và in ra tổng các số nguyên tố nằm trong khoảng 1..n

Exercise 5: Đoán số

- Chương trình tạo ra một số ngẫu nhiên nằm trong khoảng từ 1 đến 10.
- Cho phép người dùng nhập vào 1 số trong khoảng 1 đến 10.
- Nếu số người dùng nhập > số ngẫu nhiên => thông báo: Your number > Our number
- Nếu số người dùng nhập < số ngẫu nhiên => thông báo: Your number < Our number
- Nếu số người dùng nhập = số ngẫu nhiên => thông báo: Your number = Our number => You win!!! => Dừng trò chơi
- Chương trình chỉ cho người dùng đoán tối đa là 3 lần, nếu sau 3 lần không đoán ra số thì dừng trò chơi => thông báo: You lose!!!

Gợi ý:

Exercise 1: Bạn muốn dừng lại hay chơi tiếp

```
In [42]: continue <- 1
```

```
In [43]: repeat{
  continue <- as.integer(readline(prompt = "Input 1 (Continue), Other number (Stop): "))
  message(continue)
  if (continue != 1){
    break
  }
}
```

```
Input 1 (Continue), Other number (Stop): 1
```

```
1
```

```
Input 1 (Continue), Other number (Stop): 1
```

```
1
```

```
Input 1 (Continue), Other number (Stop): 1
```

```
1
```

```
Input 1 (Continue), Other number (Stop): 1
```

```
1
```

```
Input 1 (Continue), Other number (Stop): 0
```

```
0
```

Exercise 2: Bình phương vector

```
In [11]: n <- as.integer(readline(prompt = "Please input n: "))
```

```
Please input n: 10
```

```
In [12]: vector_random = round(runif(n,1,10), 0)
print(vector_random)
```

```
[1] 7 6 4 3 9 2 3 4 4 7
```




```
In [13]: # x <- vector_random * vector_random
# x
```

```
In [14]: square = c()
```

```
In [15]: for (value in vector_random){
  square <- c(square, value^2)
}
```

```
In [16]: print(square)
```

```
[1] 49 36 16  9 81  4  9 16 16 49
```

Exercise 3: Bảng cửu chương

```
In [17]: # bang cuu chuong
n <- as.integer(readline(prompt = "Input n:"))
s <- ""
danh_sach <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
for (i in danh_sach){
  s <- paste(s, "\n", n, "x", i, "=", n*i)
}
cat(s)
```

Input n:7

```
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
```



```
In [1]: # bang cuu chuong tu a den b
begin <- as.integer(readline(prompt = "Input begin:"))
end <- as.integer(readline(prompt = "Input end:"))
s <- ""
danh_sach <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
danh_sach_begin_end <- c(begin : end)
for (i in danh_sach){
  for (j in danh_sach_begin_end){
    s <- paste(s, "\t", j, "x", i, "=", j*i)
  }
  s <- paste(s, "\n")
}
cat(s)
```

Input begin:3

Input end:6

3 x 1 = 3	4 x 1 = 4	5 x 1 = 5	6 x 1 = 6
3 x 2 = 6	4 x 2 = 8	5 x 2 = 10	6 x 2 = 12
3 x 3 = 9	4 x 3 = 12	5 x 3 = 15	6 x 3 = 18
3 x 4 = 12	4 x 4 = 16	5 x 4 = 20	6 x 4 = 24
3 x 5 = 15	4 x 5 = 20	5 x 5 = 25	6 x 5 = 30
3 x 6 = 18	4 x 6 = 24	5 x 6 = 30	6 x 6 = 36
3 x 7 = 21	4 x 7 = 28	5 x 7 = 35	6 x 7 = 42
3 x 8 = 24	4 x 8 = 32	5 x 8 = 40	6 x 8 = 48
3 x 9 = 27	4 x 9 = 36	5 x 9 = 45	6 x 9 = 54

Exercise 4: Tính giá trị

- Chương trình nhập vào một số n
- Chương trình tìm, tính và in ra tổng các số nguyên tố nằm trong khoảng 1..n



```
In [20]: # tong cac so nguyen to <=n
n <- as.integer(readline(prompt = "Input n: "))
```

```
begin = 1
s = ""
sum_primes = 0
while (begin<=n){
  # test prime
  i=1
  count <- 0
  while (i<=begin){
    if(begin%%i == 0){
      count <- count + 1
    }
    i <- i + 1
  }
  if(count==2){
    sum_primes <- sum_primes + begin
    s <- paste(s, begin, "+")
  }
  begin<-begin+1
}
s<-paste("Sum of primes:",s, "=", sum_primes)
cat(s)
```

Input n: 10

Sum of primes: 2 + 3 + 5 + 7 + = 17

Exercise 5: Đoán số

```
In [38]: random_number = round(runif(1,1,10), 0)
```

```
In [39]: random_number
```

2

```
In [40]: times <- 1
flag = FALSE
```



```
In [41]: while (times<=3){
          your_number <- as.integer(readline(prompt = "Your number (1..10): "))
          if(your_number == random_number){
            message("You win!!!")
            flag = TRUE
            break
          }
          else if(your_number > random_number){
            message("Your number > Our number")
          }
          else{
            message("Your number < Our number")
          }
          times <- times + 1
        }
        if(flag == FALSE){
          message("You lose!!!")
        }
```

Your number: 7

Your number > Our number

Your number: 5

Your number > Our number

Your number: 3

Your number > Our number

You lose!!!