

## Exercise 6 : Control Structures in R

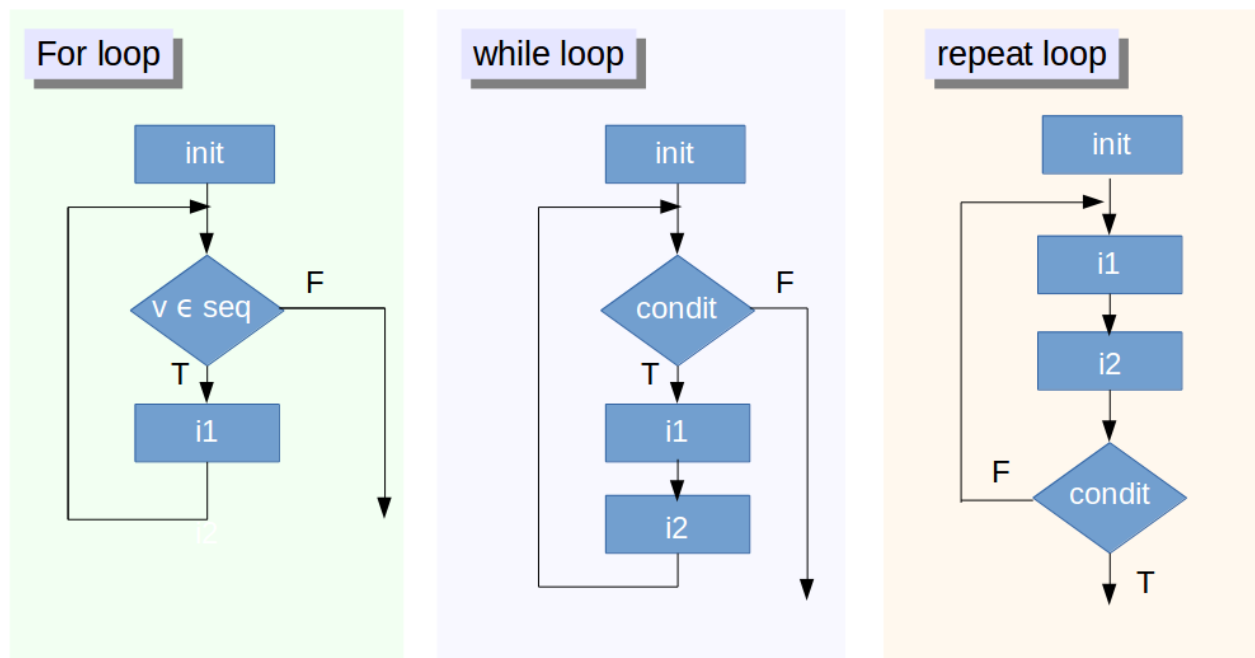
In this session of exercise, the following things are implemented in R

- For-loop
- If loop
- If-else-if loop
- While loop
- Break statement
- Next statement

Note the text shaded (which you will found as you go through the exercises) with grey are commands.

This is command

### Flow of For loop, while loop and repeat loop



**Note :** **Do not copy paste**, type the commands in REditor save and source the code.

### a) For Loop

1. Execute the following commands of for-loop. Here, everything inside curly braces { } are executed 100 times.

```
for(i in 1:100){  
  print("Hello world!")  
  print(i*i)  
}
```

2. Execute the following commands to run a for loop. Here we are implementing for-loop to square the values in the given vector. You can copy the code in R Notepad, save and Source the code from R Console.

```
## Implementing For loop To square the values in the given Vector
## Creating Vector of 10 values
forvector<-seq(from=2, to=22, by=2)
## Creating Empty vector to store the output
squaredVector=NULL
## Applying for loop
for (i in 1:length(forvector)){
  squaredVector[i]=forvector[i]^2
  print(squaredVector[i])
}
```

3. Execute the following commands to run a For-loop.  
Here we are implementing for-loop to print the mathematical table. Given the number by the user  
You can copy the code in R Notepad, save and Source the code from R Console.  
You can check the help pages of prompt and paste commands by using ?prompt() and ?paste() respectively

```
## For loop for printing mathematical table
## Taking input from user
num<-as.integer(readline(prompt="Enter a number= "))
for (i in 1:10){
  print(paste(num,'x',i,'=',num*i))
}
```

4. For loops may be nested, but when and why would we be using this? Suppose we wish to manipulate a bi-dimensional array by setting its elements to specific values; we might do something like this:.

```
# nested for: multiplication table
mymat = matrix(nrow=30, ncol=30) # create a 30 x 30 matrix (of 30 rows
and 30 columns)
for(i in 1:dim(mymat)[1]) # for each row
{
  for(j in 1:dim(mymat)[2]) # for each column
  {
    mymat[i,j] = i*j # assign values based on position: product of
two indexes
  }
}
```

## b) IF Statement

5. Write a R program to read the number from the command prompt and print if the entered number is negative.

```

num<-as.integer(readline("Enter a number: "))
if(num<=0){
msg<-sprintf("The entered number is %d\n", num)
print("It is Negative")
}

```

6. Write a R program to read the numbers from the command prompt and print if the entered numbers are equal

```

num1<-as.integer(readline("Enter a number: "))
num2<-as.integer(readline("Enter a number: "))
if(num1==num2){
msg1<-sprintf("First Number is %d \n", num1)
msg2<-sprintf("Second Number is %d \n", num2)
cat(msg1)
cat(msg2)
print("The entered numbers are equal")
}

```

### c) If else Statement

7. Write a R program to read the number from the command prompt and print if the entered number is negative or positive

```

num<-as.integer(readline("Enter a number: "))
if(num<=0){
msg<-sprintf("The entered number is %d\n", num)
print("It is Negative")
}
else{
print("The number entered is Positive")
}

```

8. Another example:

```

## Declaring a variable
a<-100;
## check the boolean condition
if( a < 20 ){
print("a is less than 20 \n")
}else{
{
print("a is not less than 20\n" )
}
}
msg<-sprintf("value of a is : %d\n", a)
cat(msg)

```

## d) If else if Statement

9. Run the following commands of if-else-if statements. Here we are checking the user entered number are equal or less than or greater than with the other number

```
var1<-as.integer(readline("Enter first number="))
var2<-as.integer(readline("Enter Second number="))
msg1<-sprintf("First number is %d \n", var1 )
msg2<-sprintf("Second number is %d \n", var2 )
cat(msg1)
cat(msg2)
if (var1 >var2)
{
    print("var1 is greater than var2")
}else if (var2 > var1)
{
    print("var2 is greater than var1")
}else
{
    print("var1 is equal to var2")
}
```

10. Another example:

```
a<-100;
## check the boolean condition
if( a == 10 ) {
    ## if condition is true then print the following
    print("Value of a is 10\n" );
}else if( a == 20 ) {
    ## if else if condition is true
    print("Value of a is 20\n" );
} else if( a == 30 ) {
    ## if else if condition is true
    print("Value of a is 30\n" );
} else {
    ## if none of the conditions is true
    print("None of the values is matching\n" );
}
sprintf("Exact value of a is %d \n", a );
```

## e) While loop in R

while() loop will execute a block of commands until the condition is no longer satisfied.

11. Execute while loop to print numbers.

```
x <- 1
while(x < 5) {
  x <- x+1;
  print(x);
}
```

### **f) Break in While loop in R**

12. Let's break the loop when x=3. Here, when x reaches 3, the loop is terminated.

```
x <- 1
while(x < 5) {x <- x+1; if (x == 3) break; print(x); }
```

Output:  
[1] 2

### **g) Next in While loop in R**

13. Let's skip one step when x=3. Here when x reaches 3, it will not print that value

```
x <- 1
while(x < 5) {x <- x+1; if (x == 3) next; print(x);}
```

Output:  
[1] 2  
[1] 4  
[1] 5