Conditional Statement:

In R we have two Conditional Statement. One is If else Statement and another one is Nested If Else Statement. It works like any other programming language.

The If else statements are a very important part of R programming. In R there are a lot of powerful packages for data manipulation. It produces a logical value and carries out the next statement only when the logical values become TRUE.

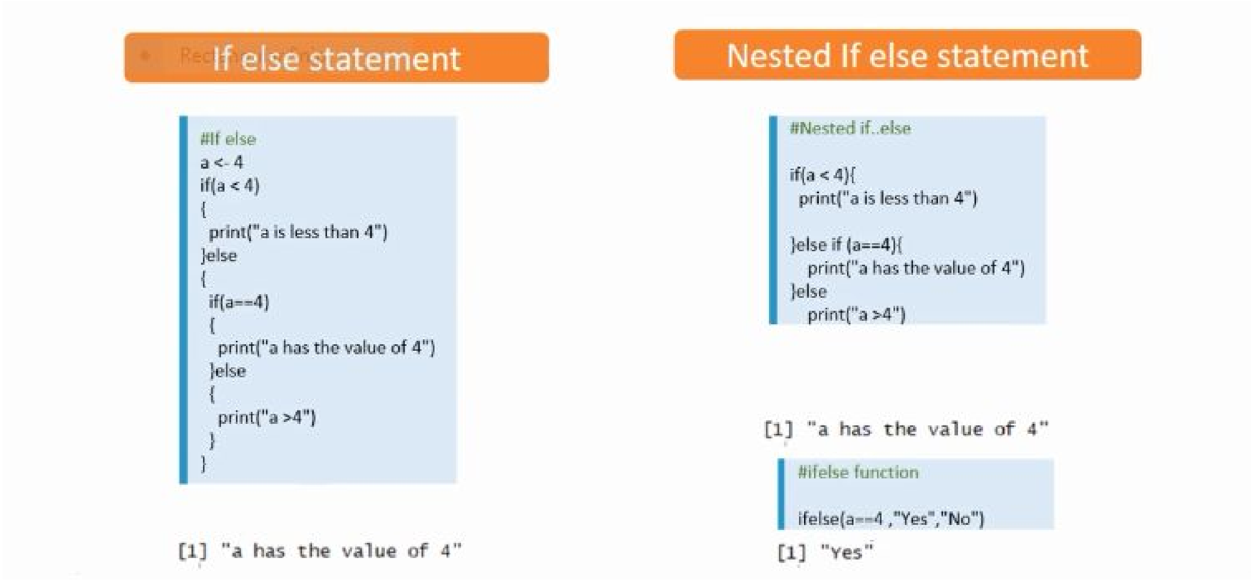
If statement,

If the condition is true you can try to access the condition and if it doesn’t make then it goes to else Statement.

As you can check, whether a is less than 4 or not?

If it is less than 4 then it is satisfied and then enters into the block and print a is less than 4 and else again check whether it is a=4?

If it is a=4 then print whether a has the value of 4 otherwise go to another statement that’s how it goes on and on. So, if statement you have the condition and you write that conditions it's just checked and if it’s satisfying and it enters into the block and does whatever you order to do and otherwise it goes to else statement.



If else statement

```
#If else
a <- 4
if(a < 4)
{
 print("a is less than 4")
}else
{
 if(a==4)
 {
  print("a has the value of 4")
 }else
 {
  print("a >4")
 }
}
```

```
[1] "a has the value of 4"
```

Nested If else statement

```
#Nested if..else

if(a < 4){
 print("a is less than 4")

}else if (a==4){
  print("a has the value of 4")
}else
  print("a >4")
```

```
[1] "a has the value of 4"
```

```
#ifelse function

ifelse(a==4 ,"Yes","No")
```

```
[1] "Yes"
```

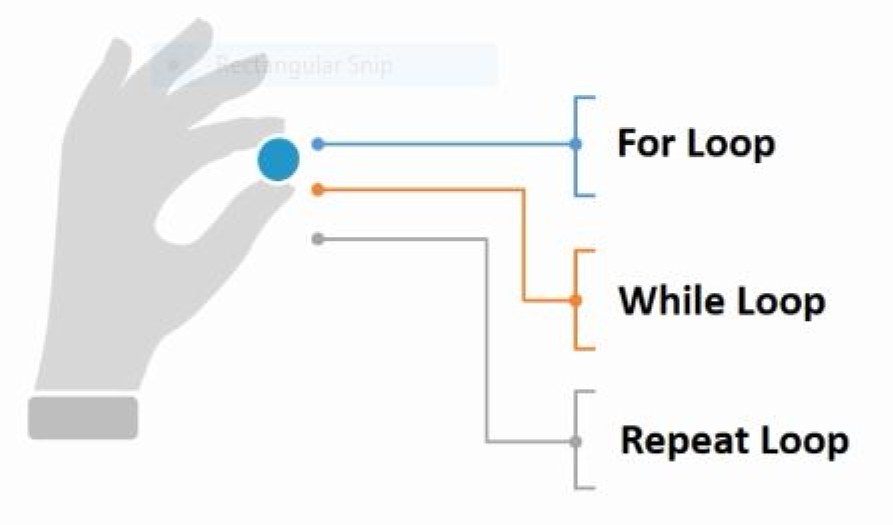
The Nested If-Else the same thing can be done. With a Nested statement you are saying that if a is less than equal to 4 then print a is less than equal to 4 which is exactly same as the previous one but here you don’t write multiple else rather you are writing else if a == 4 then print a is value of 4 otherwise else print a>=4.

Hurriedly, see the output. The output is also same. For If else statement a is the value of 4 and since here we had stored 4 for a.

Similar things happened for the Nested If Else Statement.

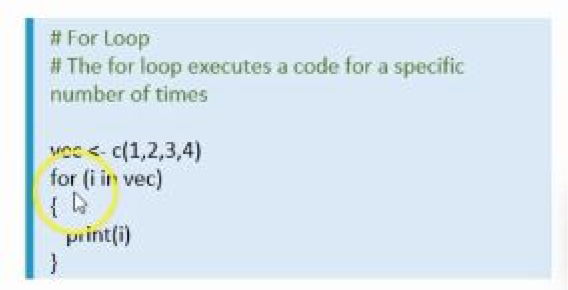
Same in another variant which is If else variant. You can use If else function here you can print, when a==4 then the output is Yes,otherwise it is No. it doesn’t have the If Else Statement level of control but it does your job checking two conditions whether this made then print whatever you want to say otherwise you print whatever is not made.

Loops-



- **For Loop in R:**

Suppose you are storing 1,2,3,4 in vector, using that combining function then you are seeing for then you are writing the condition that if “i” in a vector for 1 to 4 times you are asking to run that loop and you are asking to **print (i)**.



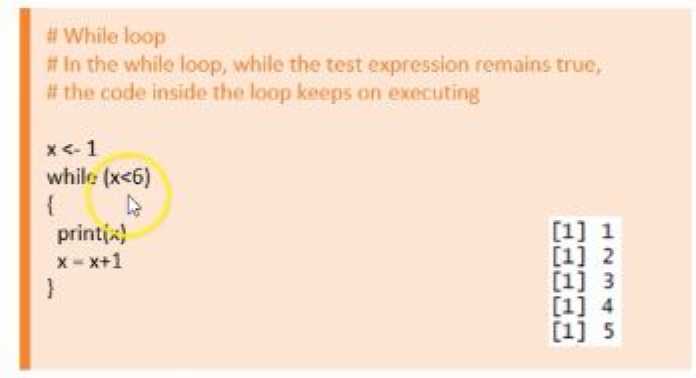
```
# For Loop
# The for loop executes a code for a specific
number of times

vec <- c(1,2,3,4)
for (i in vec)
{
  print(i)
}
```

so I will start from 1 then 2 then 3 then 4, that’s how the output you can see 1,2,3,4 and that’s how a For Loop also. It is exactly to similar to any other languages.

- **While Loop:**

While loop repeats a statement or group of statements when the given condition is true. It is testing the condition before executing the Loop body.Now in While Loop, while the test expression remains true, the code inside the loop keeps on executing.



```
# While loop
# In the while loop, while the test expression remains true,
# the code inside the loop keeps on executing

x <- 1
while (x<6)
{
 print(x)
 x = x+1
}
```

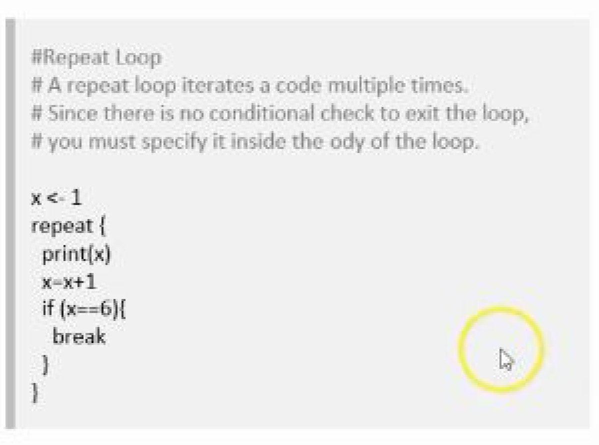
```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Till the point the condition is made which is X less than 6 this will just run. Here suppose we will start with X equal to 1 then we will check whether it is less than 6 then we print X and then before existing we will increase by 1 so it would print 1 and X becomes to again checks again print 2 then again increase to 3 and again goes till the point 5 and then once it becomes 5 then before existing it would increase that to 6 so it own entered into the while loop. That’s how a while loop works. It is pretty simple.

- **Repeat Loop:**

Now discuss the Repeat Loop. A Repeat Loop executes a sequence of statements multiple times and abbreviates the code that manages the Loop variable.

Earlier cases in For Loop and While Loop we are giving the condition before we even start the loop but in Repeat Loop we are not mentioning any condition. Here we just start X equal to 1(X=1) and let’s print X before existing you increase X/1 and then you give your condition that if X is == 6 (X==6) then break, break means nothing you just exist the loop. That’s how we jot down the condition in a Repeat Loop.



```
#Repeat Loop
# A repeat loop iterates a code multiple times.
# Since there is no conditional check to exit the loop,
# you must specify it inside the ody of the loop.

x <- 1
repeat {
 print(x)
 x=x+1
 if (x==6){
  break
 }
}
```

So you keep on doing all those conditions and then, in the end, you just check the condition at the end of the loop before existing and if it is satisfied, before it satisfied the condition then it gives the output 1,2,3,4,5 then you just increase that 1 and then it becomes X equal to 6(X=6) and finally you break from the loop.

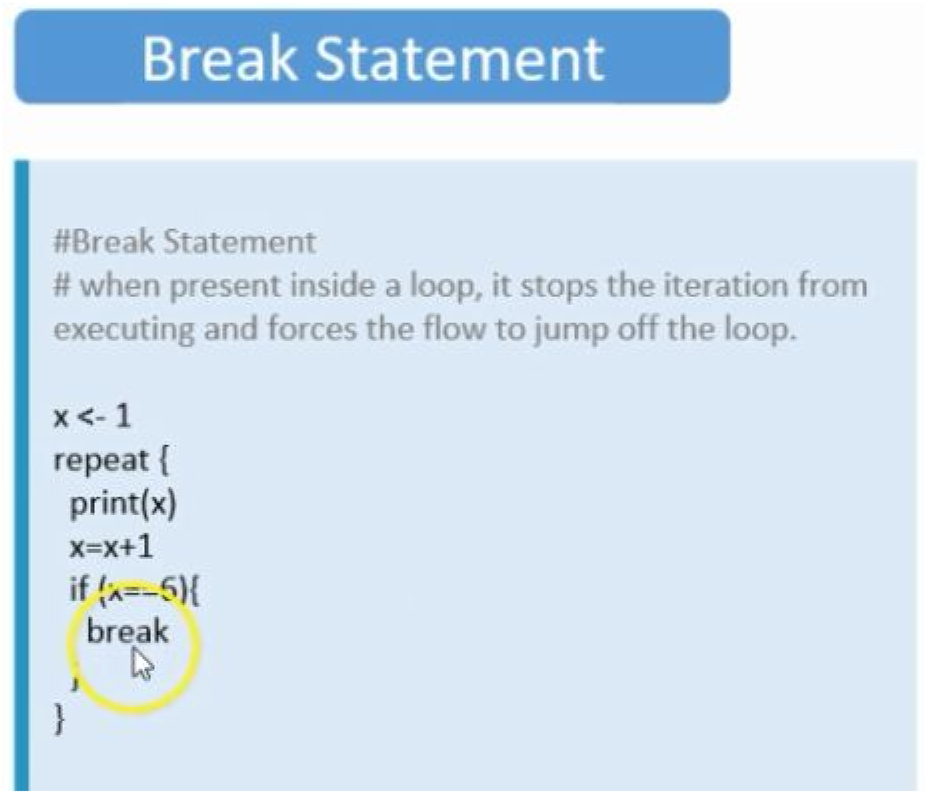
Loops in R:

Now we are discussing **Break Statement** and **Next Statement** as well.

Here will be some situation where we have to terminate the loop without executing all the statements. In this condition, we can use the Break Statement and Next statements. Just like the While and Repeat Loop, you can break out of a loop completely by using the break statement. Additionally, if you want to skip the current iteration and continue the loop then you can use the Next Statements.

Break Statement:

In a Repeat Loop if you actually want to break from that loop so based on a condition you can write that statement. Break Statement is used inside a loop, to stop the iteration and flow the control outside of the loop. It is also used to terminate a case in the switch statement. (covered in the next chapter) Break Statement can also be used inside the else branch of the if else statement. Just like the While and Repeat Loop, you can break out of a loop completely by using the break statement.



Break Statement

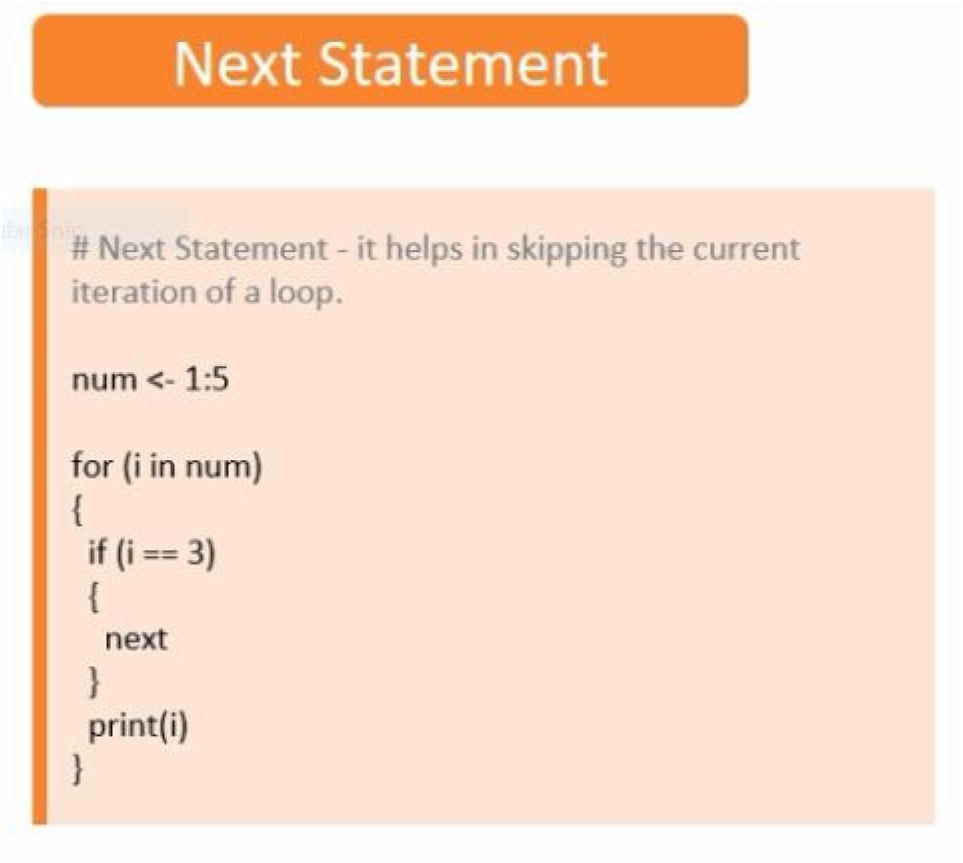
```
#Break Statement
# when present inside a loop, it stops the iteration from
executing and forces the flow to jump off the loop.

x <- 1
repeat {
  print(x)
  x=x+1
  if (x==6){
    break
  }
}
```

For an example:

Suppose you have 15 statements inside the loop and you want to exit from the loop when a certain condition is true otherwise it has to execute all of them. In this condition, you have to use the If Statement to check for the expression and place the Break Statement inside the If block. If the condition is true then the compiler will execute the break statement, and the break will exit the controller from the loop completely otherwise, it will execute all the statements.

Next Statement:
Next statement is useful when we want to skip the current iteration of a loop without terminating it. On encountering next, the R parser skips further evaluation and starts next iteration of the loop. Next Statement simulates the behavior of R switch. The next discontinues a particular iteration and jumps to the Next Cycle. In fact, it jumps to the evaluation of the condition holding the current loop. The Next Statements can also be used inside the else branch and of if else statements. Next Statement actually helps to skip from a current relationship of a loop.



Next Statement

```
# Next Statement - it helps in skipping the current
iteration of a loop.

num <- 1:5

for (i in num)
{
 if (i == 3)
 {
  next
 }
 print(i)
}
```

So, suppose in a For loop ideally it should go from 1 so suppose I equal to 1? (I=1) and in number, we have 5, so we are checking from one to five(1-5) and we are printing the value of i. But now you want whenever it reaches I equal to three(i=3) then you want to skip that loop. So, what would happen? So, it would print all of them but it own print if it is I equal to three (i=3). So, it would print one to then skip three then four five.

This brings an end to this post, I encourage you to re read the post to understand it completely if you haven’t and **THANK YOU**.