control structure : (theory or practical)

1.if-else :
it is control structure in which if the condition is true then if block will execute otherwise else block will exe cute.
IF & ELSE is used to perform if-else control structure

syntax :
```
if(condition){
 #statements
}else{
 #statements
}
```

example :
```
a <- 3
if(a>4){
 print("a is greater than 4")
}else{
 print("a is less than 4")
}
```

output :
a is less than 4

2.for loop :
it is a loop which is used to iterate or execute set of statement for a specified no of times.
for keyword is used to perform.

sytnax :
```
for(var in seq ){
 #statements
}
```

example :
```
for(i in 1:10){
 print(i)
}
```

output :
1
2
3
4
5
6
7
it will go to 10

3.while loop:
it is a loop , which is used to repeat or execute set of statements until the given condition is statisfied.
while keyword is used to perform.

syntax :
```
while(condition){
```

```
 #statements
}
```

example :

```
a <- 1
while(a<=10){
 print(a)
 a<- a+1
}
```

output :

1
2
3
4
5
6
7
it will go to 10

4.repeat loop :
it is also a loop , which is used to repeat or execute set of statements for infinitely.
to stop this we have condition in the loop.
repeat keyword is used to perform.

syntax :

```
repeat{
 #statements
 if(condition){
  break #break the loop
 }
}
```

example :

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

output :

1
2
3
4
5
6
7
it will go to 10

5.break :
it is keyword which is used to break or to stop the loop or iteration of loop when the given the condition is

satisfied.
it is used in the loops

syntax :
```
for(var in seq){
 if(condition){
  break
 }
}
```

example :
```
for(i in 1:10){
 if(i ==5){
  break
 }
 print(i)
}
```

output :
1
2
3
4

6.next :
it is also keyword which is used to skip the current loop or iteration.
it is used in the loops

syntax :
```
for(var in seq){
 if(condition){
  next
 }
}
```

example:
```
for(i in 1:10){
 if(i == 5){
  next
 }
 print(i)
}
```

output :
1
2
3
4
6
7
8
9
10

7.switch :

it is used when we have multiple conditions or cases.
switch keyword is used to perform.

syntax :
switch(condition,case1 , case1 , casen)

example :
switch(1 , "sunday" , "monday", "tuesday","thursday","friday","saturday")

ouput :
sunday

8.apply() :
it is a built-in function in R
it is used to apply functions on the rows and cols of array & matrix or vectors

syntax :
apply(x , MARGIN , FUN)

parameters :
x : array or matrix
MARGIN : an integer vector indicating which margin should be remained after applying function(1 for rows , 2 for cols)
FUN : the function to apply

example :
matrix <- matrix(1:12 , nrow = 4 , ncol = 3 , brow = TRUE)

matrix and their sum
1 2 3 = 6
4 5 6 = 15
7 8 9 = 24
10 11 12 = 33

apply(matrix , MARGIN = 1 , FUN = sum)

output :
6 15 24 33

9.lapply():
it is a built-in function in R.
it is used to apply function to each element in the list returns a list as an output.
it is commonly used when you to keep the output in the form of list data structure.

syntax :
lapply(x , FUN)

parameters :
x : input list
FUN : function to apply

example :
list <- list(a = c(1,2,3) , b = c(1,2,3) , c = (1, 2,3))
lapply(list , FUN = sum)

output :
6
6
6

10.sapply():
it is a built-in function in R.
it is used to simplify the output of lapply() into a vector or matrix when possible.
it is used to apply the function to each elements in the list.

syntax :
sapply(x , FUN)

example :
list <- list(a = c(1,2,3) , b = c(1,2,3) , c = (1, 2,3))
sapply(list , FUN = sum)

output :
a b c
6 6 6