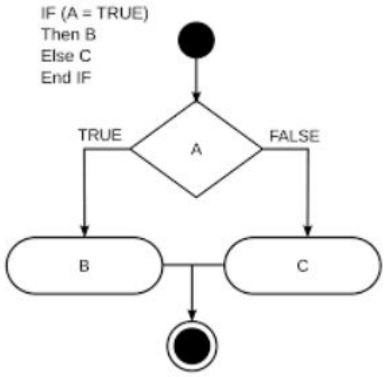
If..else Conditional Statements

Contents

Introduction

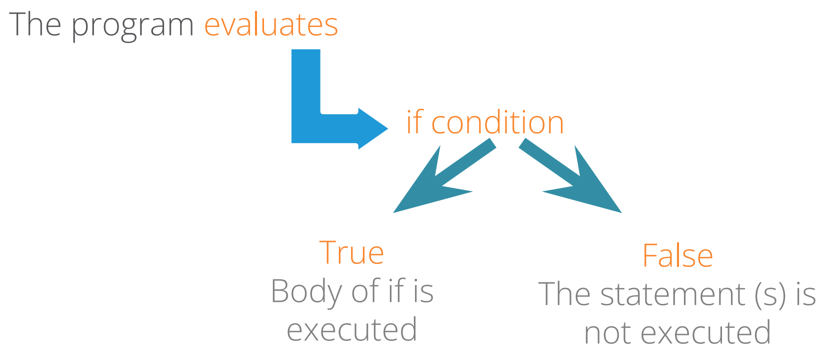
- Conditional Statements are useful in data science programming when we want to execute a code only if a certain condition is satisfied.
- Condition is referred as a logical expression where a decision is made to execute some code based on a Boolean (true or false) condition
- The most simplest form is `if` statement. `else` and `else if` statements can also be used depending on the number of conditions to be checked.



if Statement

Syntax:

```
  if  (condition)
{
        statement(s)
}
```



if Statement

If the condition is TRUE, the statement gets executed. But if it's FALSE, nothing happens.

```
# Check if x is a positive number or not
```

```
x<-5
if(x > 0){
   print("Positive number")
}
```

```
[1] "Positive number"
```

- ❑ Comparison operator '>' is used in condition.
- ❑ Here, x >0 returns TRUE; hence the print statement is executed

```
# Print the value of the object if it is of a character type
```

```
x<-"Hello there"
if(is.character(x)) {
  print(x)
}
```

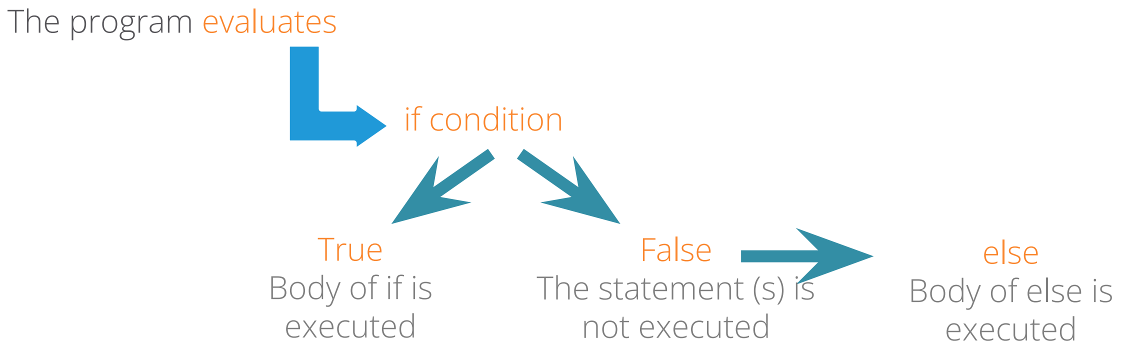
```
[1] "Hello there"
```

- ❑ is.character() returns TRUE or FALSE depending on whether the argument is of character type or not.
- ❑ Here x is a character, hence the condition returns TRUE and "Hello there" is printed

if..else Statement

Syntax:

```
if condition{
    statement(s)
} else {
        statement(s)
}
```



* The **`else`** statement is optional and there could be at most only one **`else`** statement following **`if`.**

if..else Statement

`if..else` Statements are used when you want to execute one block of code when a condition is true, and another block of code when it is false.

```
# Check if x is a positive number or not and print the status
x<- -5
if(x > 0){
   print("Positive number")
} else {
   print("Negative number")
}
```

This condition evaluates to false; hence it will skip the body of if and the body of else will be executed.

```
[1] "Negative number"
```

* There is an easier way to use **`if..else`** statement specifically for vectors. You can use **`ifelse()`** function instead.

if..else Statement

```
# Take input as a number from user
# Print the sum of natural numbers up to that number.
```

```
num=as.integer(readline(prompt="Enter Integer Value: "))
if(num < 0) {
  print("Enter a positive number")
} else {
  sum=(num * (num + 1)) / 2;

  print(paste("The sum is", sum))
}
```

```
[1] "The sum is 15"
```

- ❑ readline() interactively reading a line from terminal.
- ❑ prompt= takes a string printed when prompting the user for input

Here the input is 5, hence the sum of 5 natural number is 15

ifelse()

`ifelse()` Function is a vector equivalent form of the **`if..else`** statement

Syntax:

```
ifelse(condition,x,y)
```

- If the condition is TRUE it returns x else y
- **`ifelse()`** function can take a vector as an input and results into a vector

```
#Create a vector & put a condition to check & print even or odd depending
#on result.
```

```
a<-c(6,1,5,14)
ifelse(a%%2==0,"even","odd")
```

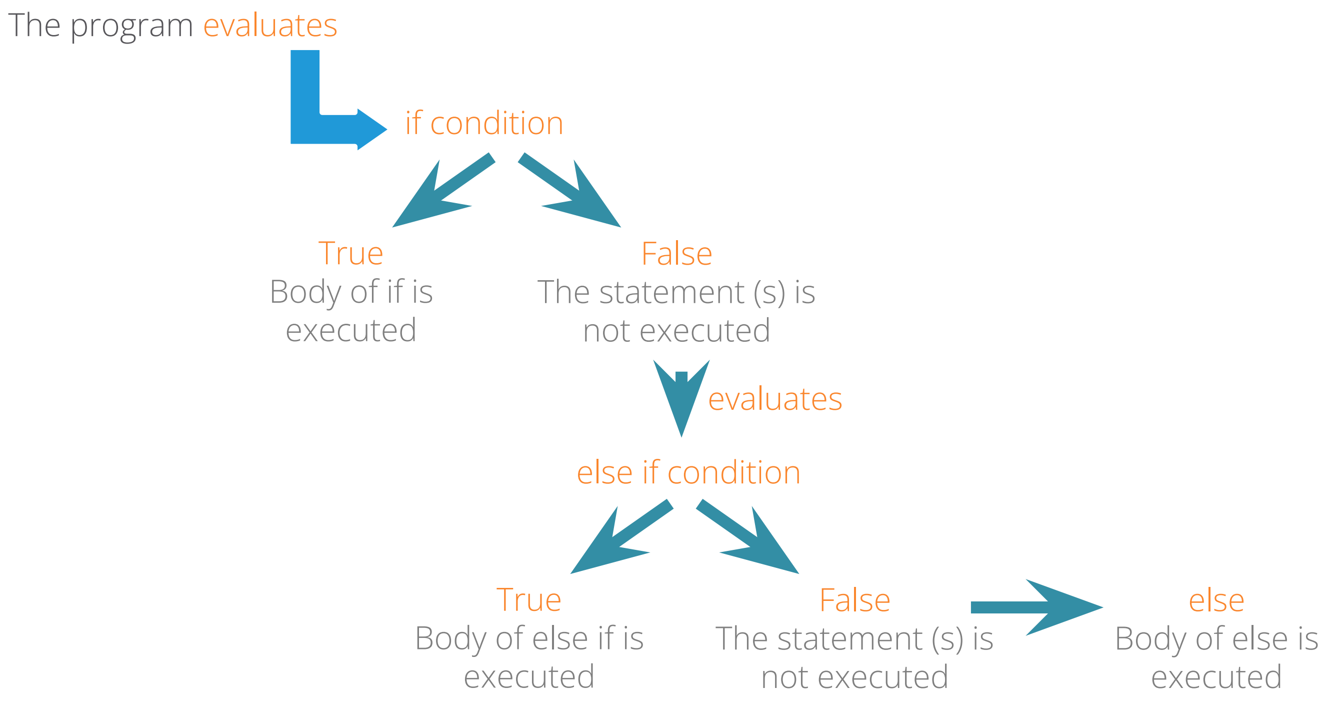
```
[1] "even" "odd" "odd" "even"
```

Nested if..else Statement

Syntax:

```
if (condition1) {
      statement(s)
} else if (condition2) {
      statement(s)
} else if (condition3) {
      statement(s)
} else {
      statement(s)
   }
```

Nested if..else Statement



Only one statement gets executed depending on the conditions.

Nested if..else Statement

Nested **if..else** Statements are used when there are several conditions. Any number of conditions can be added to **else if** statement after an **if** statement and before an **else** statement.

```
# check whether x is positive, negative or zero and print the status
```

```
x <- 0
if(x < 0){
   print("Negative number")
}else if(x>0) {
   print("Positive number")
}else{
   print("Zero")
}
```

```
[1] "Zero"
```

The if condition and else if condition evaluates to FALSE so it executes the statements in body of else.

Quick Recap

In this session, we learnt all the **`if..else`** conditional statements with the help of examples. Here is a quick recap:

if statement	• If the condition is TRUE, the statement gets executed. But if it's FALSE, nothing happens.
if..else statement	• Used when you want to execute a statement when the condition returns FALSE. **`ifelse()`** is the vector equivalent of the if ..else statement
Nested if..else statement	• Used when there are several conditions • Multiple conditions can be given using **`else if`** statement