2. Shell Scripting – if block

Difference between blocks and loops:

Blocks are a set of instructions for one time execution whereas loops are for repetition of a set of instructions based on some criteria. The number of times of execution of loops completely depends on the type of loop used. There are different types of loops and it can be used based on the requirement.

Relational operators:

Relational operators are used in conditions to check inside if block or while loop. Operators are different for integers, real numbers and strings.

1. Relational operators for integers (input and output will be integers):

   -gt → greater than

   -ge → greater than or equal to

   -lt → less than

   -le → less than or equal to

   -eq → equal to

   -ne → not equal to

2. Relational operators for strings:

   -z → to check if string length is equal to zero

   -n → to check if string length is greater than zero

   = → to check if two strings are equal

   != → to check if two strings are unequal

3. Relational operators for real numbers:

   - input and output will be real numbers
   - used along with bc.

     Ex: echo “2.5 > 10” | bc
   - output will be either 0 or 1

   > → greater than

   < → less than

   >= → greater than or equal to

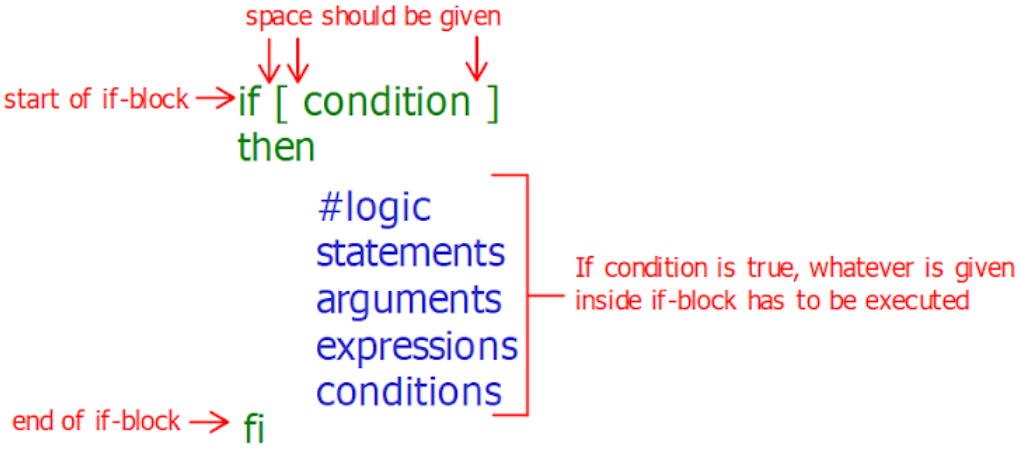
   <= → less than or equal to

   == → equal to

!= → not equal to

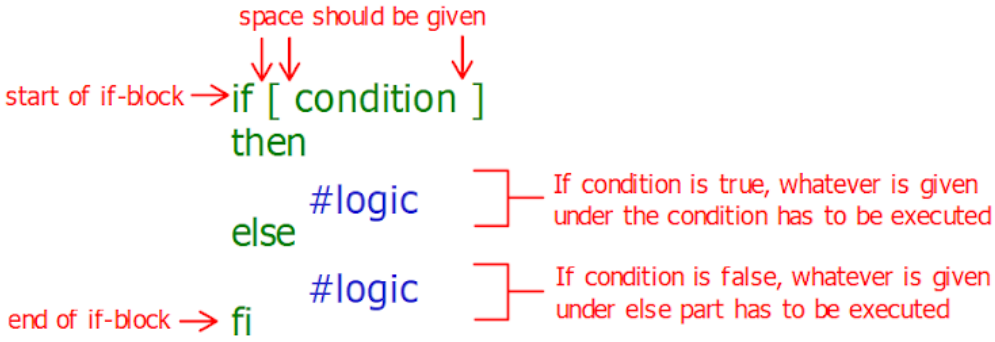
if block:

The following if-block can be used only if there is one condition to be checked and executing the block based on it. The logic part of if block can have anything to be executed based on the requirement; it can be some statements, evaluation of expressions, checking another condition or even some arguments to the same block.



if-else block:

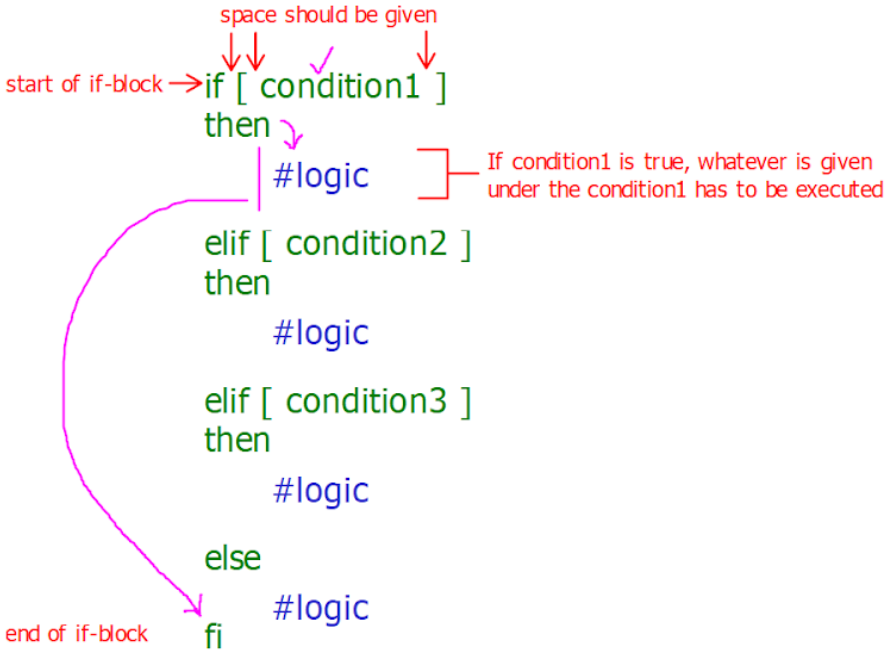
In the above if-block if the condition is false, there will be neither output nor error printed on screen, which is logically wrong. So, always if-block will be correct by adding else part to it as given below:



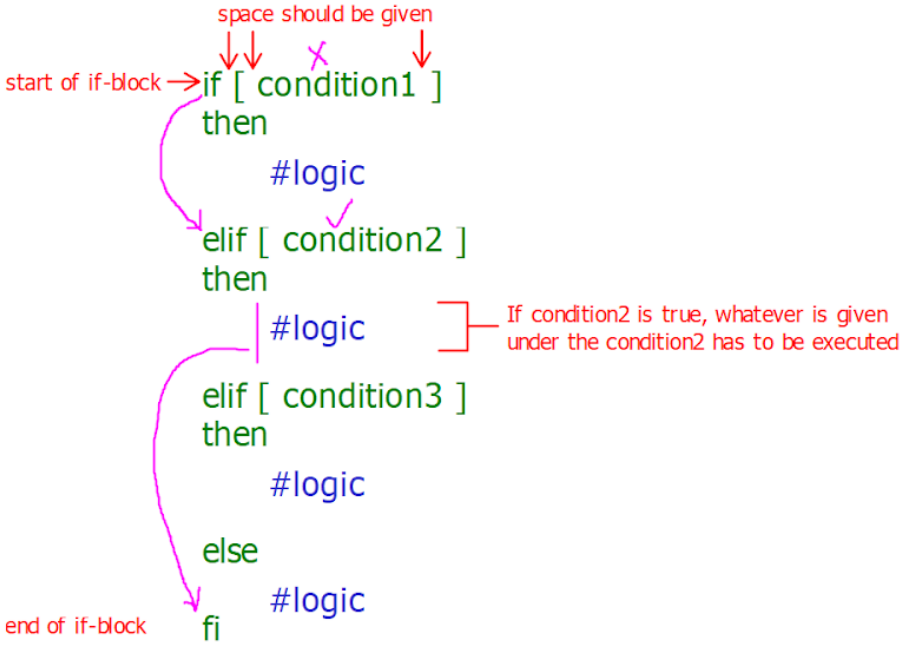
if-elif-else block:

If more than one condition needs to be included in one if-block and if the block has to be executed based on the different conditions, then each condition can be added using elif. Required space should be given before and after every condition and [ ]. If one condition is satisfied, then the rest of the conditions or else if present will be neglected. Below given is the example of if-block with 3 conditions:

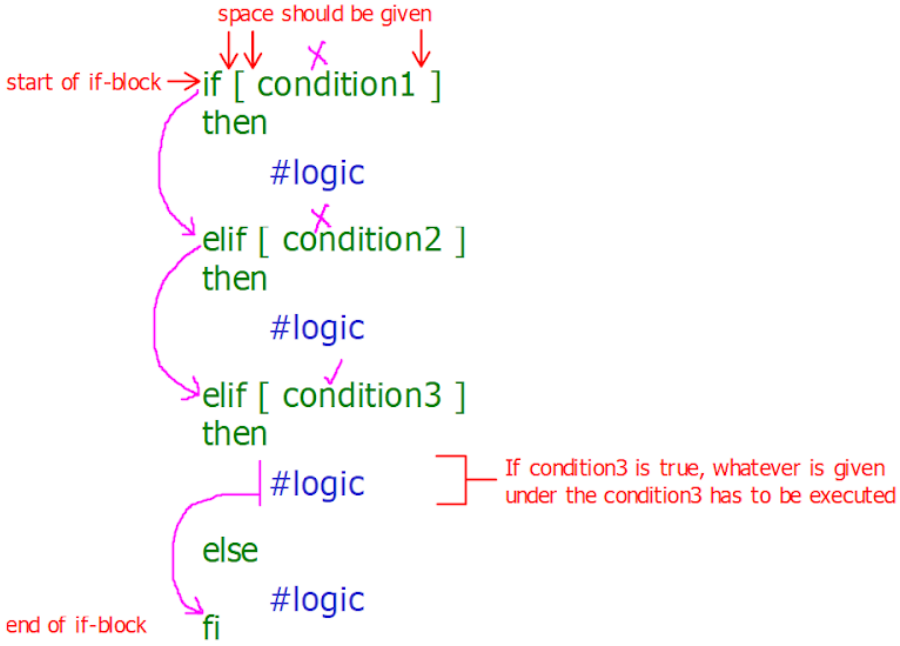
1. If condition1 is true:



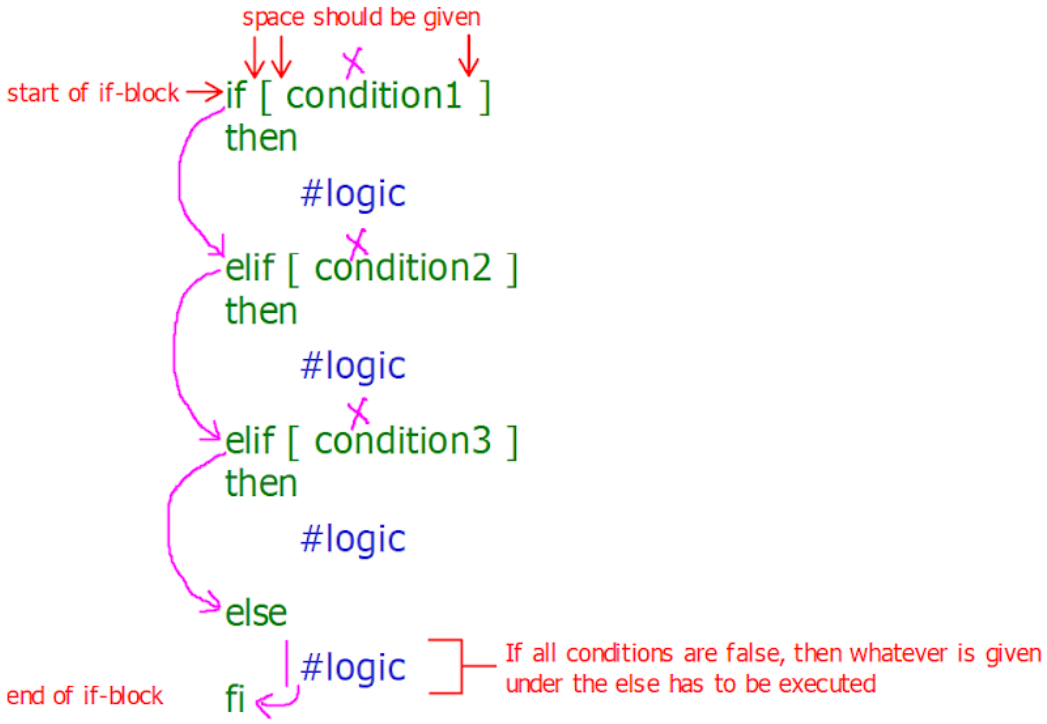
2. If conditon1 is false and if condition2 is true:



3. If condition1, condition2 are wrong and when condition3 is satisfied:



4. If none of the conditions are true, then:



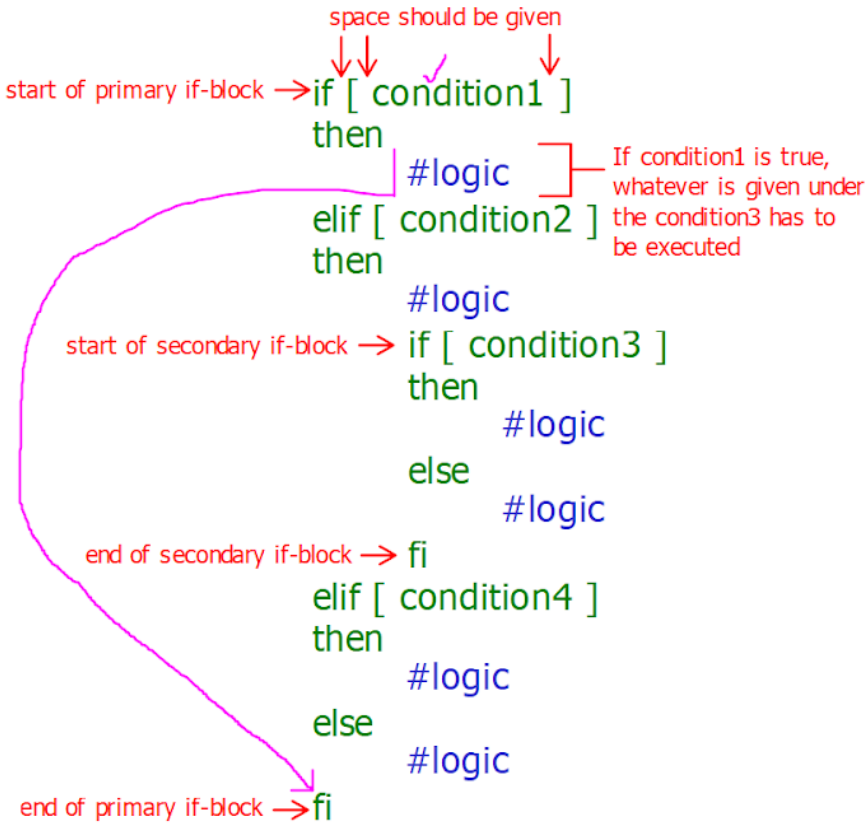
The above given is an example of flow of if-block with multiple conditions. The if-block can be further modified depending on the requirements and so there can be

loops inside blocks or blocks inside loops or loops inside loops or blocks inside blocks. When an if-block is used inside another if-block, that is called a nested if-block.

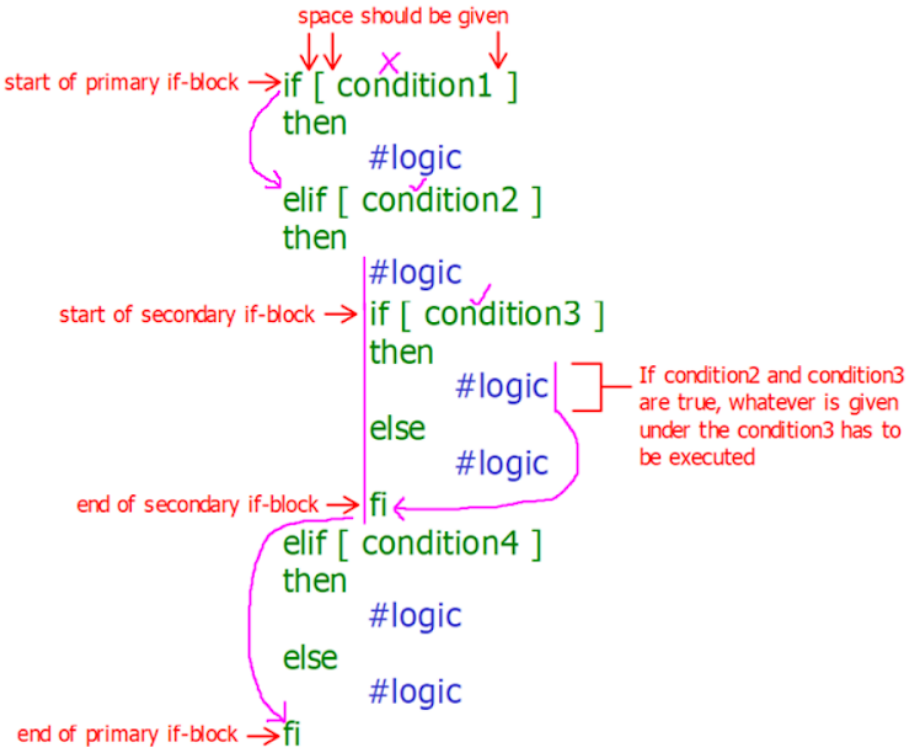
Nested if-block:

One or more if-blocks can be inside an if-block to check one condition under another or to check some condition under the else part of the if-block based on requirement.

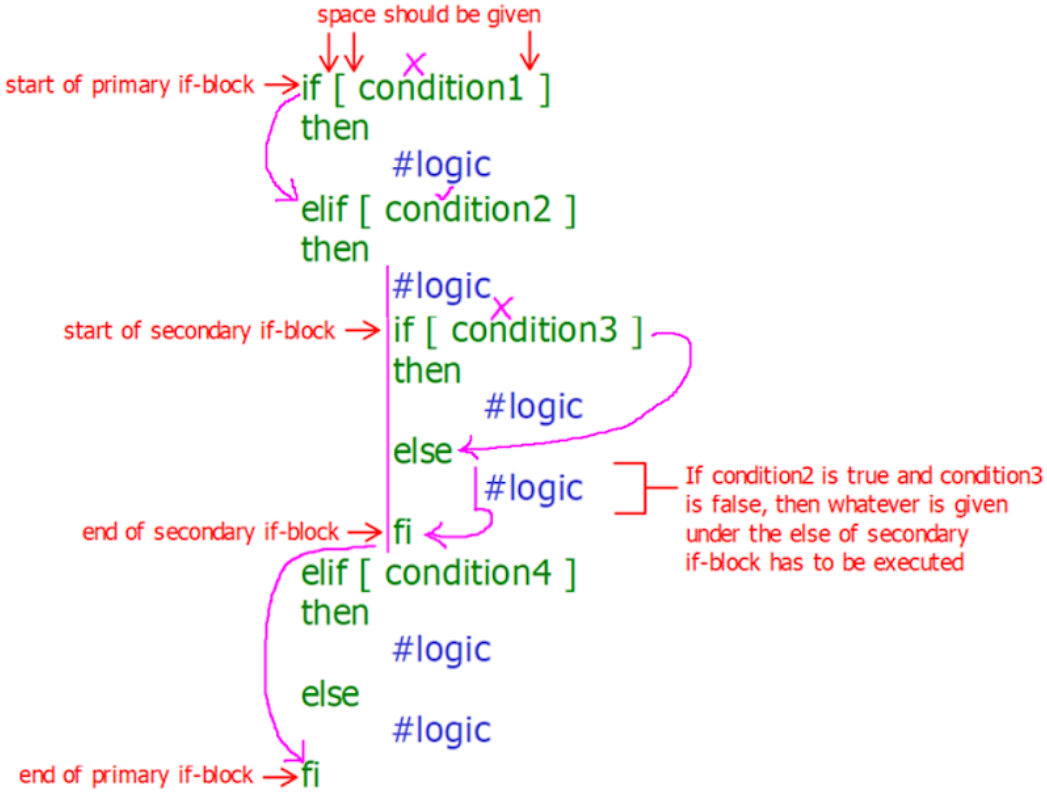
1. If condition1 is true:



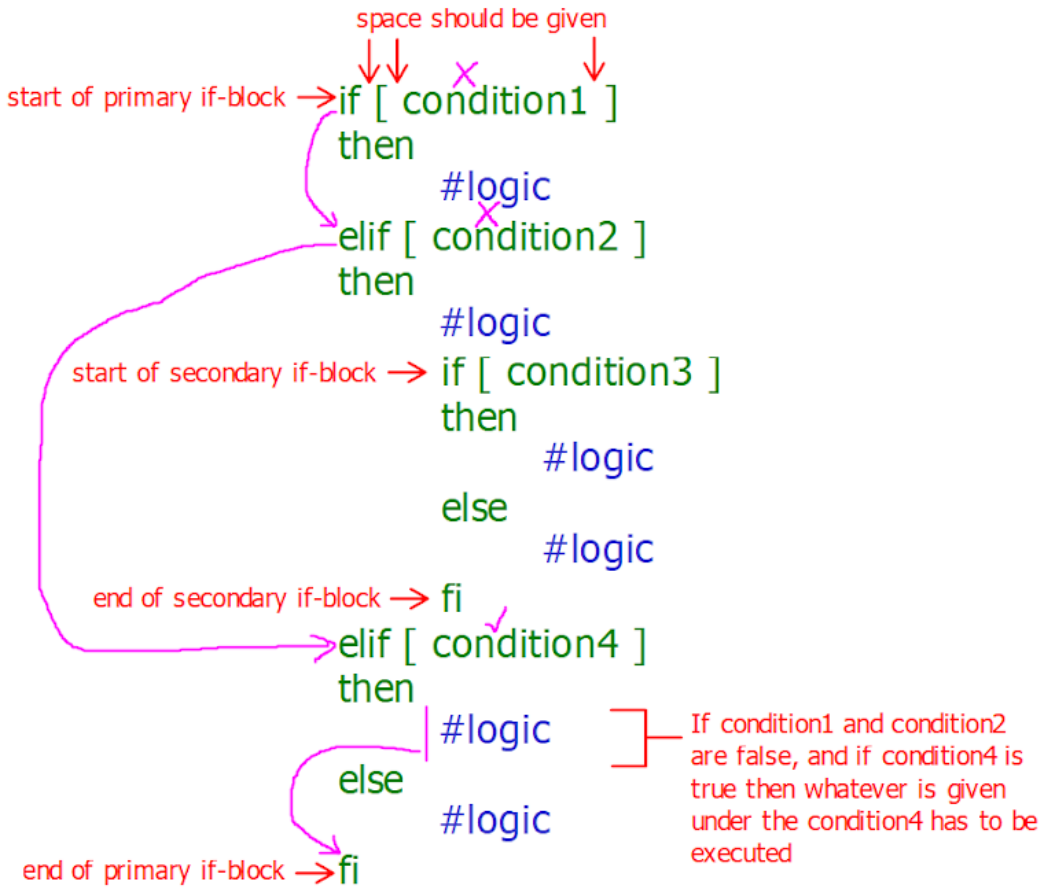
2. If condition2 is true, whatever is under condition2 has to be executed. If there are other conditions to check under condition2 it has to be checked with another if-block if it's meant to be executed once. Given below is the example if condition3 under condition2 is true:



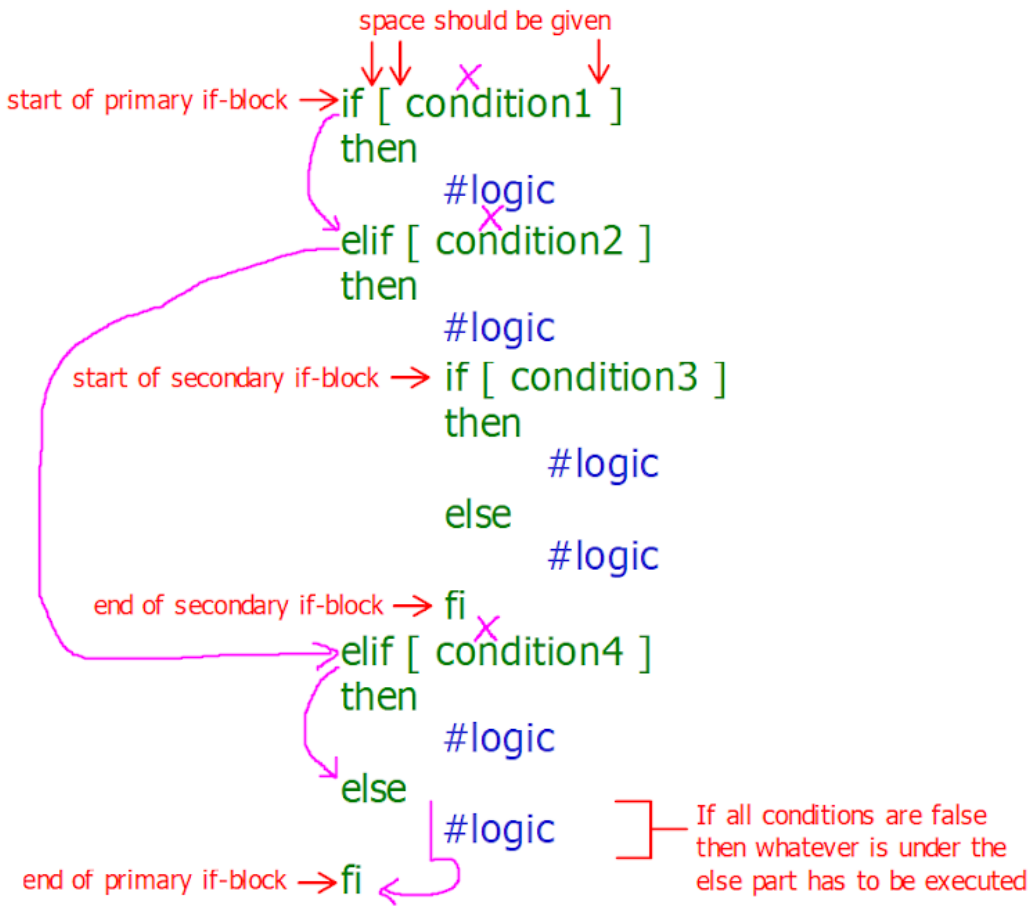
3. If condition2 is true but condition3 is false, then next condition will be checked or else part will be executed if its present:



4. If condition1 & condition2 are false and if condition4 is true:



5. If all conditions are false:


space should be given
start of primary if-block
if [condition1]
then
#logic
elif [condition2]
then
#logic
start of secondary if-block
if [condition3]
then
#logic
else
#logic
end of secondary if-block
fi
elif [condition4]
then
#logic
else
#logic
If all conditions are false
then whatever is under the
else part has to be executed
end of primary if-block
fi