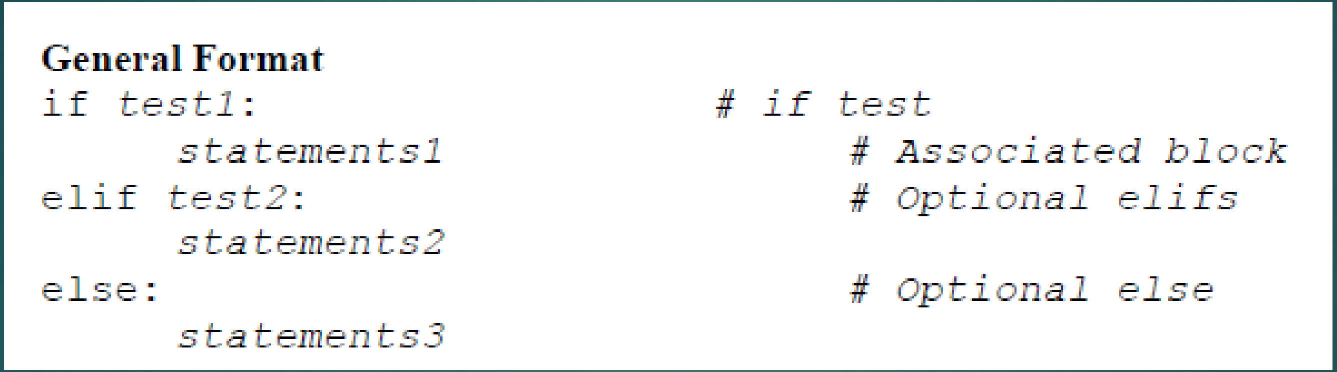
Lecture 03

Decision Making and Conditional Operator

Decision Making Structures

Normally, the program flows along line by line in the order in which it appears in source code. But, it is sometimes required to execute a particular portion of code only if certain condition is true; or false i.e. you have to make decision in the program.



General Format

```
if test1:                   # if test
      statements1                 # Associated block
elif test2:                       # Optional elifs
      statements2
else:                             # Optional else
      statements3
```

- The indentation (blank whitespace all the way to the left of the two nested statements here) is the factor that defines which code block lies within the condition statement.

- Python doesn‘t care how indents can be inserted (either spaces or tabs may be used), or how much a statement can be indented (any number of spaces or tabs can be used). In fact, the indentation of one nested block can be totally different from that of another.

- The syntax rule is only that for a given single nested block, all of its statements must be indented the same distance to the right. If this is not the case, a syntax error will appear, and code will not run until its indentation is repaired to be consistent. Python almost forces programmers to produce uniform, regular, and readable code.

- The one new syntax component in Python is the colon character (:). All Python compound statements that have other statements nested inside them—follow the same general pattern of a header line terminated in a colon, followed by a nested block of code usually indented underneath the header line.

Python Conditions and If statements

Python supports the usual logical conditions from mathematics:

Equals: a == b
Not Equals: a != b
Less than: a < b
Less than or equal to: a <= b
Greater than: a > b
Greater than or equal to: a >= b

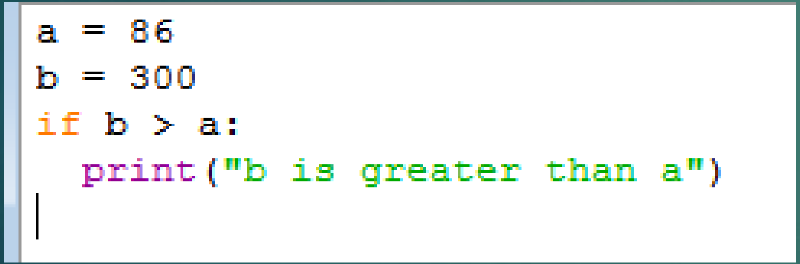
These conditions can be used in several ways, most commonly in "if statements" and loops.

An "if statement" is written by using the if keyword.

Indentation

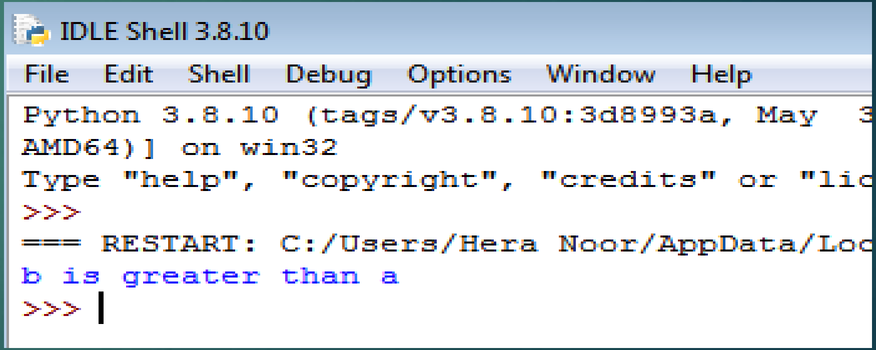
Python relies on indentation (whitespace at the beginning of a line) to define scope in the code. Other programming languages often use curly-brackets for this purpose.

CODE



```
a = 86
b = 300
if b > a:
  print("b is greater than a")
```

OUTPUT

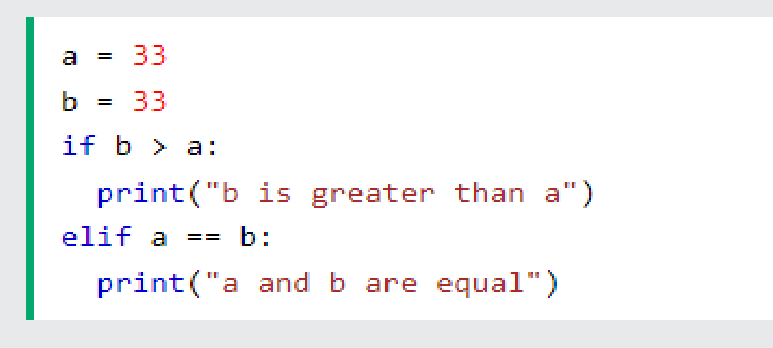


```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3
AMD64)] on win32
Type "help", "copyright", "credits" or "lic
>>>
=== RESTART: C:/Users/Hera Noor/AppData/Loc
b is greater than a
>>> 
```

Elif

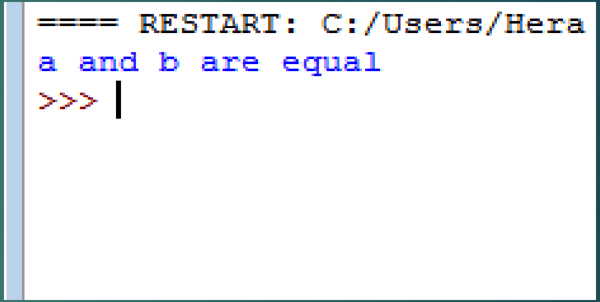
The **‘elif’** keyword is pythons way of saying "if the previous conditions were not true, then try this condition".

CODE



```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

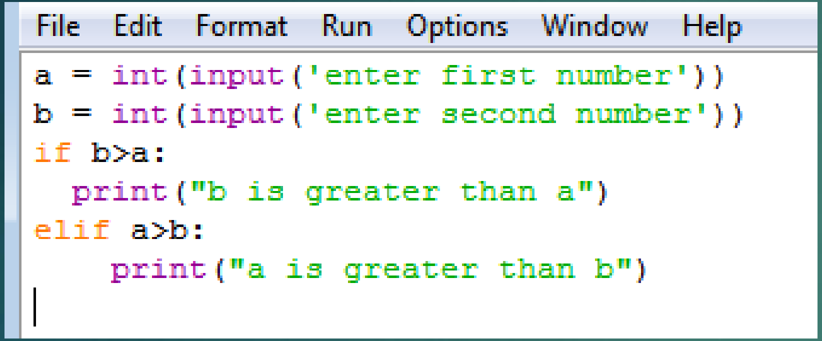
OUTPUT



```
==== RESTART: C:/Users/Hera
a and b are equal
>>> 
```

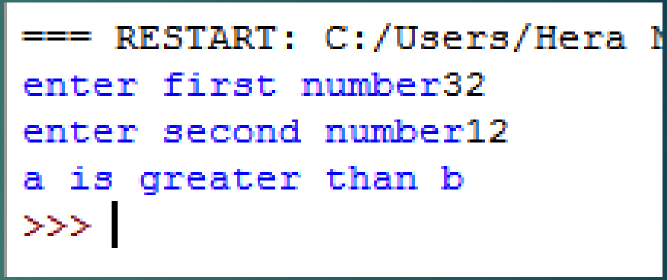
In this example a is equal to b, so the first condition is not true, but the elif condition is true, so we print to screen that "a and b are equal".

CODE



```
a = int(input('enter first number'))
b = int(input('enter second number'))
if b>a:
 print("b is greater than a")
elif a>b:
  print("a is greater than b")
```

OUTPUT



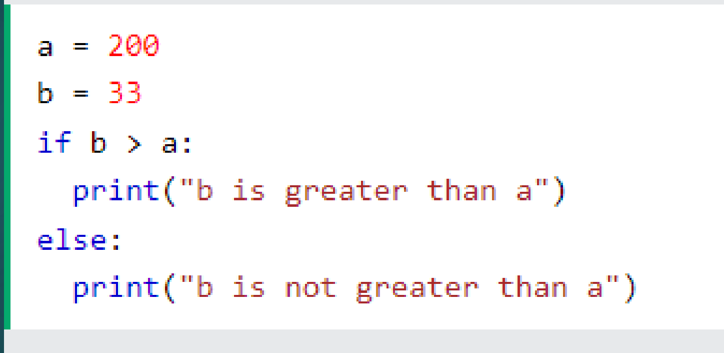
```
=== RESTART: C:/Users/Hera
enter first number32
enter second number12
a is greater than b
>>> 
```

In this example a and b are variables, so any number can be entered, the result will depend on variables and thus, we print to screen that “a is greater than b” or “b is greater than a” .

Else

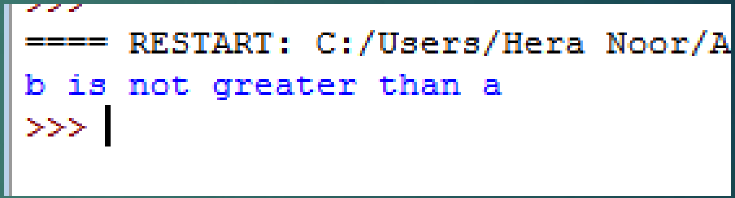
It is used as the final condition respective to ‘if’ operator. Note: no condition can be defined within else statement.

CODE



```
a = 200
b = 33
if b > a:
  print("b is greater than a")
else:
  print("b is not greater than a")
```

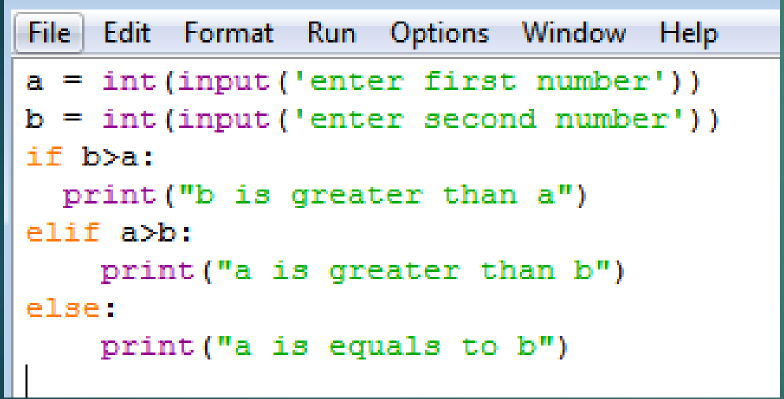
OUTPUT



```
>>>
==== RESTART: C:/Users/Hera Noor/A
b is not greater than a
>>> |
```

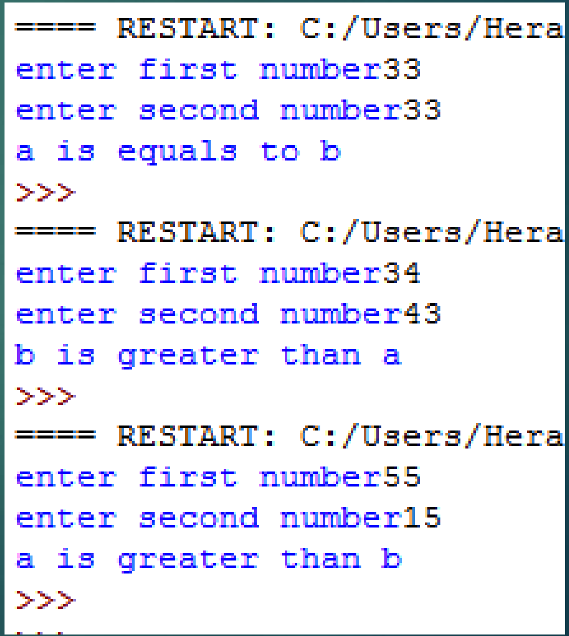
We cannot use ‘elif’ operator after ‘else’ operator, but multiple ‘elif’ and multiple ‘else’ operator can be used in a single code to define the condition properly.

CODE



```
a = int(input('enter first number'))
b = int(input('enter second number'))
if b>a:
  print("b is greater than a")
elif a>b:
    print("a is greater than b")
else:
    print("a is equals to b")
```

OUTPUT

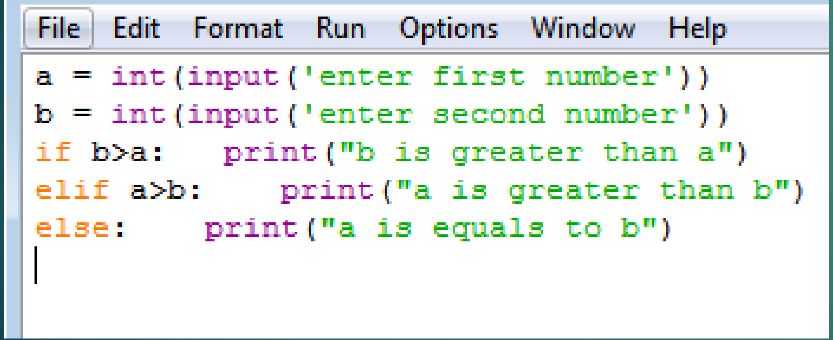


```
==== RESTART: C:/Users/Hera
enter first number33
enter second number33
a is equals to b
>>>
==== RESTART: C:/Users/Hera
enter first number34
enter second number43
b is greater than a
>>>
==== RESTART: C:/Users/Hera
enter first number55
enter second number15
a is greater than b
>>>
```

Ternary Operators or Shorthand ‘if’

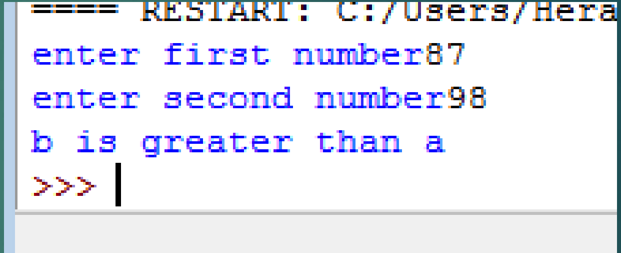
If you have only one statement to execute, you can put it on the same line as the if statement. This technique is known as Ternary Operators, or Conditional Expressions.

CODE



```
a = int(input('enter first number'))
b = int(input('enter second number'))
if b>a:   print("b is greater than a")
elif a>b:    print("a is greater than b")
else:    print("a is equals to b")
```

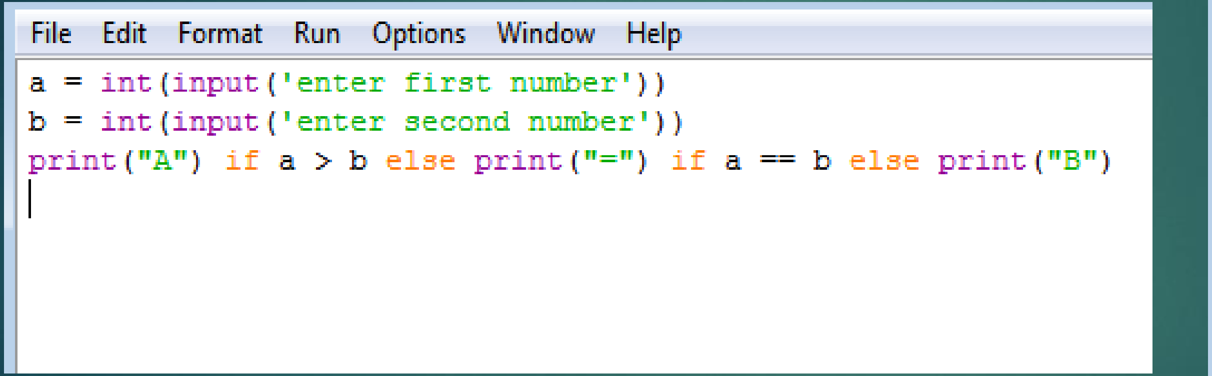
OUTPUT



```
==== RESTART: C:/Users/Hera
enter first number87
enter second number98
b is greater than a
>>> 
```

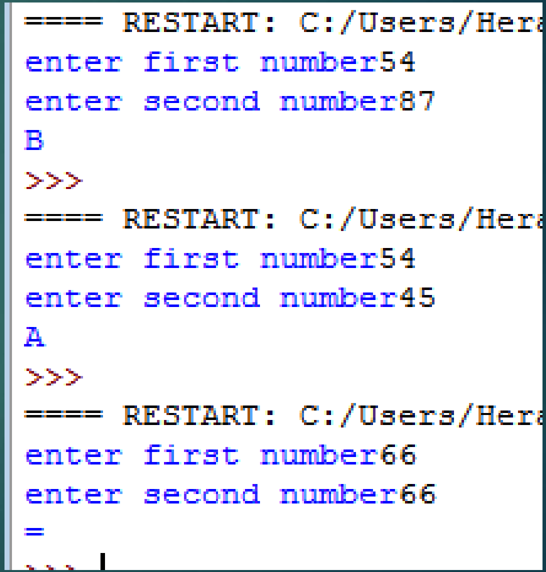
You can also have multiple else statements on the same line.

CODE



```
a = int(input('enter first number'))
b = int(input('enter second number'))
print("A") if a > b else print("=") if a == b else print("B")
```

OUTPUT

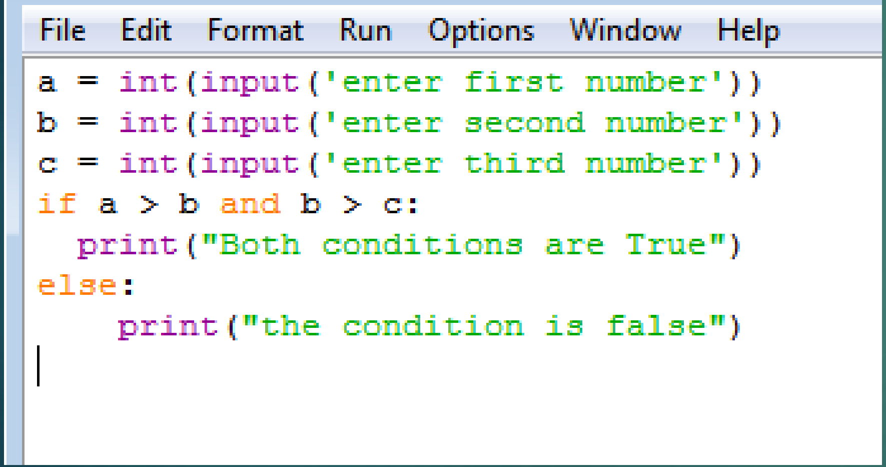


```
==== RESTART: C:/Users/Hera
enter first number54
enter second number87
B
>>>
==== RESTART: C:/Users/Hera
enter first number54
enter second number45
A
>>>
==== RESTART: C:/Users/Hera
enter first number66
enter second number66
=
>>>
```

And

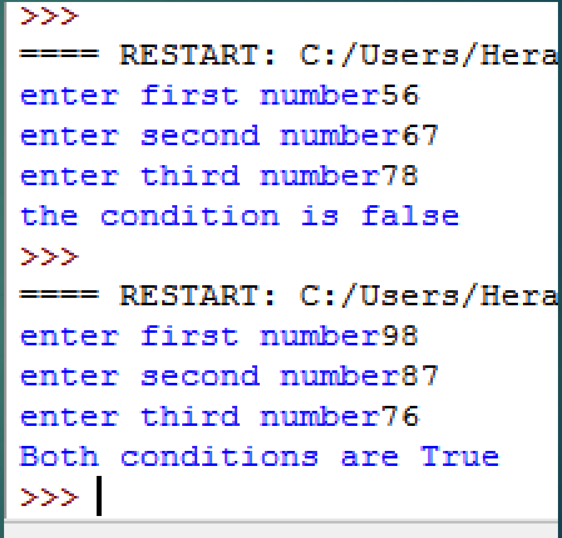
The and keyword is a logical operator, and is used to combine conditional statements

CODE



```
a = int(input('enter first number'))
b = int(input('enter second number'))
c = int(input('enter third number'))
if a > b and b > c:
  print("Both conditions are True")
else:
    print("the condition is false")
```

OUTPUT

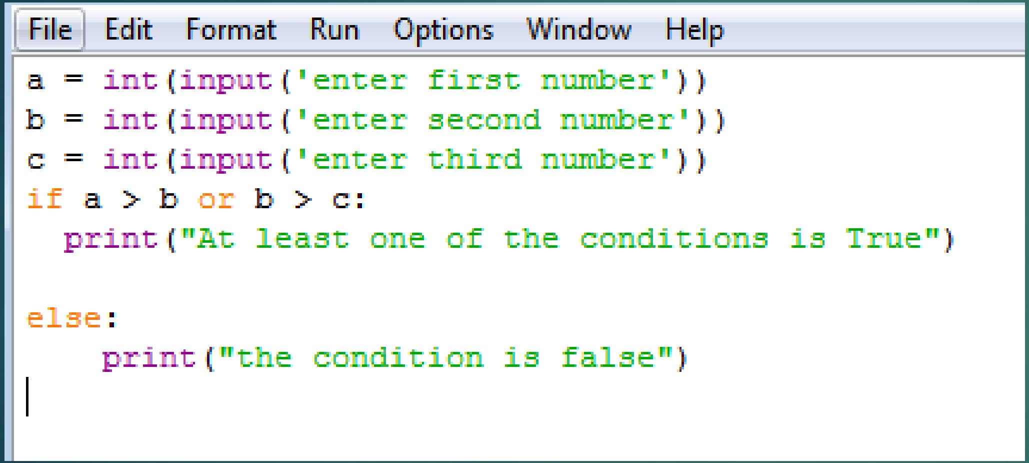


```
>>>
==== RESTART: C:/Users/Hera
enter first number56
enter second number67
enter third number78
the condition is false
>>>
==== RESTART: C:/Users/Hera
enter first number98
enter second number87
enter third number76
Both conditions are True
>>>
```

Or

The ‘or’ keyword is a logical operator, and is used to combine conditional statements.

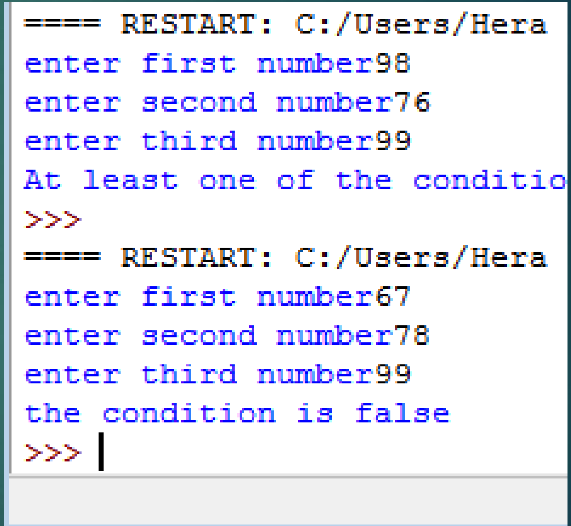
CODE



```
a = int(input('enter first number'))
b = int(input('enter second number'))
c = int(input('enter third number'))
if a > b or b > c:
  print("At least one of the conditions is True")

else:
    print("the condition is false")
```

OUTPUT

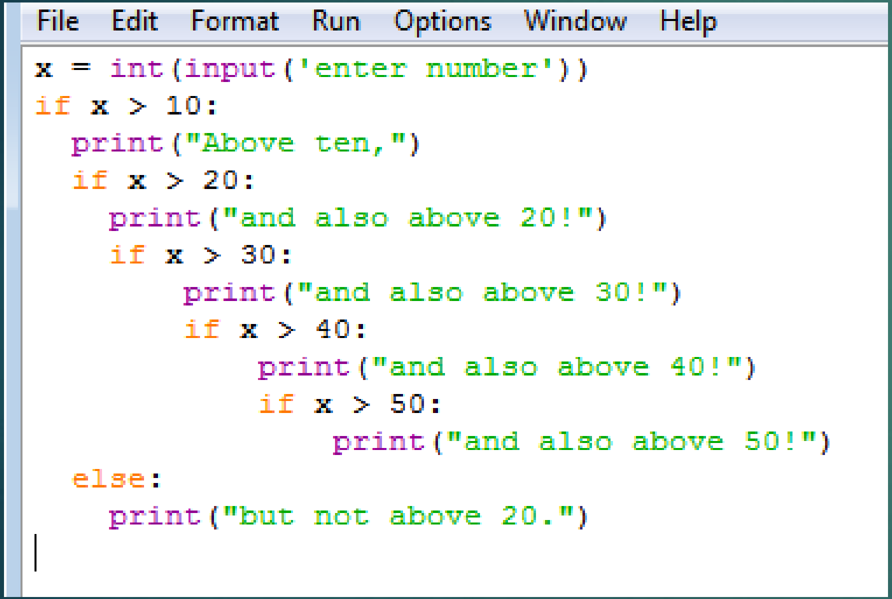


```
==== RESTART: C:/Users/Hera
enter first number98
enter second number76
enter third number99
At least one of the conditio
>>>
==== RESTART: C:/Users/Hera
enter first number67
enter second number78
enter third number99
the condition is false
>>>
```

Nested 'if'

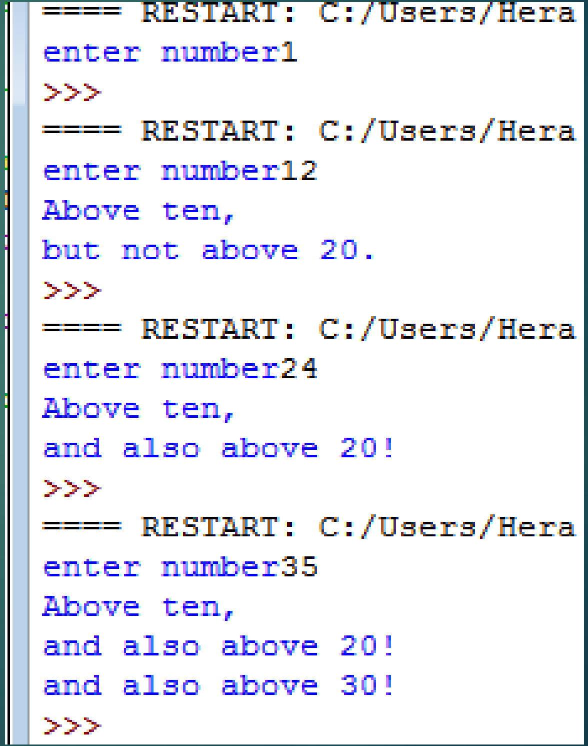
You can have 'if' statements inside 'if' statements, this is called nested 'if' statements.

CODE



```
x = int(input('enter number'))
if x > 10:
 print("Above ten,")
 if x > 20:
  print("and also above 20!")
  if x > 30:
    print("and also above 30!")
    if x > 40:
      print("and also above 40!")
      if x > 50:
        print("and also above 50!")
 else:
  print("but not above 20.")
```

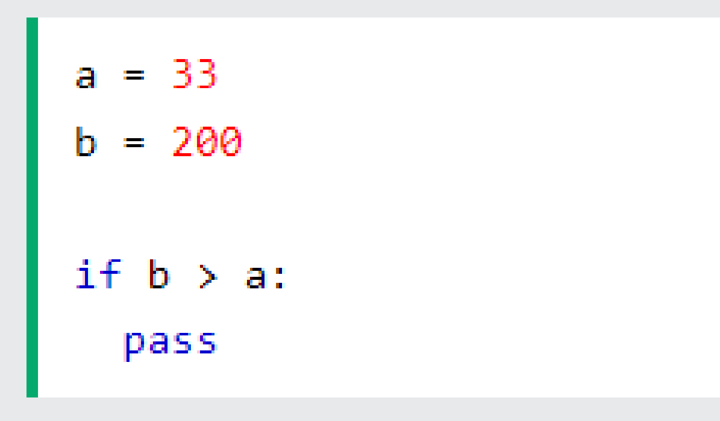
OUTPUT



```
==== RESTART: C:/Users/Hera
enter number1
>>>
==== RESTART: C:/Users/Hera
enter number12
Above ten,
but not above 20.
>>>
==== RESTART: C:/Users/Hera
enter number24
Above ten,
and also above 20!
>>>
==== RESTART: C:/Users/Hera
enter number35
Above ten,
and also above 20!
and also above 30!
>>>
```

Pass statement

‘if’ statements cannot be empty, but ‘if’ you for some reason have an ‘if’ statement with no content, put in the ‘pass statement’ to avoid getting an error.



```
a = 33
b = 200

if b > a:
  pass
```

Task – 4

1. Write a program that takes a positive integer as input from user and checks whether the number is even or odd, and displays an appropriate message on the screen.

2. Write a program that displays ―Kamran Akmal‖ on output, if score >30, Shoaib Akhtar, if 20<score <30, and Shahid Afridi if 10<score <20.

3. Write a program that takes password from user as input. Validate the password on the following criteria. Password length between 7 to 15 characters which contain at least one numeric digit and a special character is acceptable.

Thank you!