Relational Operators

Relational operators are used to find relationship between two values (OR) relational operators are used to compare two values.
Relational operators are binary operators which required 2 operands
Relational operators after comparing values it return Boolean values (True/False). An expression which returns Boolean value is called logical expression or Boolean expression.
In python relational operators are used to compare numbers and sequences (strings, list, tuple, bytes,…)

Operator	Description
>	This operator return True, if opr1 is greater opr2 **#find max of two numbers** n1=int(input("Enter first number")) # 40 n2=int(input("Enter second number")) #20 print(n1,"is max") if n1>n2 else print(n2,"is max")
<	

>=	This operator returns True, if opr1>=opr2 else return False.
	this operator verify two conditions > and ==
	
	# write a program to find a person is elg to vote or not
	
	name=input("Enter name") # naresh
	age=int(input("Enter age")) # 17
	print(name,"is elg to vote") if age>=18 else print(name,"not elg to vote")

<=	

==	== operator is used for checking equality of the values it two operands are equal it returns True else False.
	
	#login application
	user=input("UserName:") # nit
	pwd=input("Password:") # nit123
	print("welcome") if user=="nit" and pwd=="n123" else print("invalid username or password")
	
	Output:

	========== RESTART: C:/Users/user/Desktop/python6pm/py12.py ========= UserName:nit Password:nit123 invalid username or password >>> ========== RESTART: C:/Users/user/Desktop/python6pm/py12.py ========= UserName:nit Password:n123 welcome >>> **# write a program to find input character is vowel or not** ch=input("Enter any character") print("vowel") if ch=='a' or ch=='e' or ch=='i' or ch=='o' or ch=='u' else print("not vowel") **Output:** Enter any charactera vowel >>> ========== RESTART: C:/Users/user/Desktop/python6pm/py13.py ========= Enter any characterx not vowel >>>
!=	

Conditional Operator

it is a ternary operator, it required 3 operands

Conditional operators are used to execute or evaluate operands based on condition/Boolean expression.

opr1 if opr2 else opr3

if opr2 is True, it evaluates opr1
if opr2 is False, it evaluates opr3

opr2 is a condition or boolean expression

Nested conditional operators
Using more than one conditional operator
opr1 if opr2 else opr3 if opr4 else opr5 if opr6 else opr7

Logical Operators
These operators are used to combine two conditions or Boolean expressions.
Logical operators are represented using keywords

<table>
<tr><th>Operator</th><th>Description</th></tr>
<tr><td>and</td><td>Truth table of and operator
<table>
<tr><td>Opr1</td><td>Opr2</td><td>Opr1 and Opr2</td></tr>
<tr><td>True</td><td>True</td><td>True</td></tr>
<tr><td>True</td><td>False</td><td>False</td></tr>
<tr><td>False</td><td>True</td><td>False</td></tr>
<tr><td>False</td><td>False</td><td>False</td></tr>
</table>
If first operand is True it evaluates second operand and return result of second operand
If first operand is False, it does not evaluates second operand it return result of first operand

Example:
>>> 100>20 and 100>10
True
>>> 100>20 and 100>200
False
>>> 20>100 and 100>10
False
>>> 100 and 200
200
>>> 100 and 200 and 300
300
>>> 100 and 0 and 300
0
>>> "java" and "python"
'python'
>>></td></tr>
<tr><td>or</td><td>Truth table of or operator</td></tr>
</table>

	Opr1 Opr2 Opr1 or Opr2
	True True True
	True False True
	False True True
	False False False
	
	If opr1 is True, it does not evaluate opr2, it returns value of opr1
	If opr1 is False, it evaluates opr2 and return result of opr2
	
	>>> True or False
	True
	>>> False or True
	True
	>>> 100>20 or 100<20
	True
	>>> 100 or 200 or 300
	100
	>>> 0 or 200
	200
	>>>

not	