Logical Operators

Logical operators are used to combine two or more Boolean expressions or conditions

Python support the following logical operators
These logical operators are represented using keywords

<table>
<tr><th>Operator</th><th>Description</th></tr>
<tr><td>and</td><td>Truth table of and operation
<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 any operand is False, the complete expression returns False.
If first operand is True, it evaluates second operand and returns result of second operand
If first operand is False, it will not evaluates second operand and return returns result first operand

>>> True and True
True
>>> True and False
False
>>> False and True
False
>>> False and False
False
>>> True and False and True
False

A-Z → 65 – 90
a-z → 97 – 122
0-9 → 48 – 57
>>> "a">"A"
True
>>> "A"<"a"</td></tr>
</table>

<table>
<tr><td></td><td>True
>>> "a"<"A"
False

<b>Example:</b>
write a program to find input character is alphabet,digit or special character

ch=input("enter any character")
print("alphabet") if ch>='A' and ch<='Z' or ch>='a' and ch<='z' else print("digit") if ch>='0' and ch<='9' else print("special character")

<b>Output:</b>
========== RESTART: F:/python6pmaug/test27.py =========
enter any characterA
alphabet

========== RESTART: F:/python6pmaug/test27.py =========
enter any character*
special character

========== RESTART: F:/python6pmaug/test27.py =========
enter any character6
digit</td></tr>
<tr><td>or</td><td>Truth table of or operator
<table>
<tr><td>Opr1</td><td>Opr2</td><td>Opr1 or Opr2</td></tr>
<tr><td>True</td><td>True</td><td>True</td></tr>
<tr><td>True</td><td>False</td><td>True</td></tr>
<tr><td>False</td><td>True</td><td>True</td></tr>
<tr><td>False</td><td>False</td><td>False</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

If any operand is True, the complete expression is True.
If opr1 is True, it returns result of opr1 without evaluating opr2
If opr1 is False, it evaluates opr2 and returns result of</td></tr>
</table>

opr2

Example:

```
>>> True or False
True
>>> False or True
True
>>> True or True
True
>>> False or False
False
>>> 10>5 or 10<5
True
>>> 10<5 or 10>5
True
>>> 100 and 200
200
>>> 100 and 200 and 300
300
>>> 100 and 0
0
>>> "Python" and "Django"
'Django'
>>> "A" and "B" and "C"
'C'
>>> 100 and 200 or 300
200
>>> 100 or 200 and 300
100
>>> 100 and 0 or 300
300
```

Example:

```
# write a program to find input character is vowel or not

ch=input("enter any character")
ch=ch.lower()
print(ch,"is vowel") if ch=='a' or ch=='e' or ch=='i' or
ch=='o' or ch=='u' else print(ch,"not vowel")
```

Output:

```
enter any characterA
a is vowel

========== RESTART: F:/python6pmaug/test28.py =========
enter any charactera
a is vowel
```

Example:

```
# write a program to input name and 2 subject marks
# and find result (PASS/FAIL)

name=input("Enter name")
sub1=int(input("Subject1"))
sub2=int(input("Subject2"))
print(name,"PASS") if sub1>=40 and sub2>=40 else print(name,"FAIL")
```

Output:

```
Enter namenaresh
Subject130
Subject299
naresh FAIL

========== RESTART: F:/python6pmaug/test29.py =========
Enter namesuresh
Subject178
Subject240
suresh PASS
```

not

not operator is unary operator
This operator is used with one operand

Truth table not operator

Opr1	**not opr1**
True	False
False	True