Relational Operators

Relational operators are used to for comparing values.
An expression created using relational operators is called Boolean expression. This expression returns Boolean value (True/False)

>	Greater Than
<	Less than
>=	Greater than or equal
<=	Less than or equal
==	Equal
!=	Not equal

In python relational operators are used for comparing scalar type and collection type.
All relational operators are binary operators and required 2 operands to perform operation.

>Greater Than Operator

This operator returns True if opr1 is greater opr2 else False

```
>>> 10>5
True
>>> 1.5>1.2
True
>>> "B">"A"
True
>>> "a">"A"
True
>>> 10>20
False
>>> 1.5>1.9
False
>>> "a">"b"
False
>>> 10>5>2
True
>>> 10>5>9
False
>>> 10>20>10
```

```
False
```

>= Greater than or equal

This operator performs two operations

1. Greater than
2. Equal

```
>>> 10>=5
True
>>> 10>=10
True
>>> 10>=20
False
>>> 'B'>='A'
True
>>> 'B'>='B'
True
```

< Less than

This operator is used to compare the value of operand1 is less than value of operand2. If value of operand1 is less than the value of operand2 it returns True else False

```
>>> 10<20
True
>>> 1.5<1.2
False
>>> 'A'<'B'
True
>>> False<True
True
>>> [10,20]<[40,50]
True
>>> [50,20]<[60,80]
True
>>> [50,20]<[20,90]
False
```

<= Less than or equal

This operator returns True if operand1 is less than or equal operand2 else return False.

```
>>>10<=5
False
>>>10<=20
True
>>>10<=10
True
>>>1.5<=1.8
True
>>>1.5<=1.5
True
>>> "B"<="A"
False
>>> 10>5<2
False
>>> 10>5<20
True
>>> 1<5<100
True
```

== equal operator

This operator is used for comparing object values or values

```
>>>10==10
True
>>>10==20
False
>>>1.5==1.5
True
>>>1.5==1.2
False
>>> True==False
False
>>> "A"=="A"
True
>>> "a"=="A"
False
>>> "naresh"=="NARESH"
```

```
False
>>> "naresh"=="naresh"
True
```

!= not equal operator

This operator returns True if operand1 is not equal to operand2 else False

```
>>> 10==10
True
>>> 10!=10
False
>>> 10==20
False
>>> 10!=20
True
```

Logical Operators

Logical operators are used for combining two or more conditions or Boolean expressions. In python logical operators are represented using 3 keywords.

1. and
2. or
3. not

“and”,”or” are called binary operators and required 2 operands to perform operation

“not” is a unary operator and required single operand or one operand.

and operator

“and” keyword represents logical and operator

Truth table of “and” operator

Opr1	**Opr2**	**Opr1 and Opr2**
True	True	True
True	False	False
False	True	False
False	False	False

```
>>> 10>5 and 20>10
True
```

```
>>> 10>5 and 20>40
False
>>> 20>40 and 10>5
False
>>> 20>40 and "A">"B"
False
>>> 10>2 and 20>10
True
>>> 10>30 and 20>10
False
>>> 10>2 and 10>20
False
```

Concept of Evaluation

In “and” operation if operand1 is True, it evaluates operand2 and returns result of operand2

If operand1 is False, it does not evaluate operand2 and returns result of operand1

POC (Proof of Concept)

```
>>> 100 and 200
200
>>> 100 and 0
0
>>> 0 and 100
0
>>> 5 and 2
2
>>> 1.5 and 1.3
1.3
>>> 0.0 and 1.6
0.0
>>> 100 and 200 and 300
300
>>> 100 and 0 and 200
0
>>> 0 and 100 and 200
0
```

or operator

“or” keyword represents logical operator in python
Truth table of or operator

Opr1	Opr2	Opr1 or Opr2
True	True	True
True	False	True
False	True	True
False	False	False

In **or** operation, if opr1 is True, it does not evaluate opr2 and return result of opr1
If opr1 is False, it evaluates opr2 and return result of opr2

```
>>> 10>2 or 20>10
True
>>> 10>20 or 20>10
True
>>> 10>20 or 20>30
False
>>> 100 or 200
100
>>> 0 or 200
200
```

“not” operator

“not” keyword represents logical operator in python
It is a unary operator and required one operand to perform operation

Truth table

Opr1	not Opr1
True	False
False	True

```
>>> not 10>2
False
>>> not 10>20
True
```

Precedence of logical operators

1. not
2. and
3. or

```
>>> 100 and 200 or 300
200
>>> 100 or 200 and 300
100
>>> 0 or 200 and 300
300
```

Conditional Operator

Conditional operator is a ternary operator and required 3 operands to perform operation.
Conditional operator is used for creating conditional expression.

Syntax: <variable-name>=opr1 if opr2 else opr3