# 1. Comparison Operators (Relational Operators) OPERATORS (RELATIONAL OPERATORS)

```
>   MORE THAN
<   LESS THAN
>=  More than or equal to
<=  Less than or equal to
==  Equal to
!=  Not equal to
```

In [1]:
```
a=10
b=20
print(a>b)
print(a<b)
print(a>=b)
print(a<=b)
print(a==b)
print(a!=b)
```

```
False
True
False
True
False
True
```

In [2]:
```
a="ARMAN"
b="ARYAN"
print(a>b)
print(a<b)
print(a>=b)
print(a<=b)
print(a==b)
print(a!=b)
```

```
False
True
False
True
False
True
```

# 2. Logical Operators : AND,OR,NOT

### 1) Boolean Type Behaviour

```
and = If both arguments are true then only result is True else
result is False
or  = If atleast one argument is True then result is True.
not = complement
```

**2) Non-Boolean Type Behaviour**

```
0 means False
non-zero means True
empty-string means False
```

x and y ->If x is evaluates to False then return x otherwise return y.

x and y ->If x is evaluates to True then result is x otherwise y.

In [3]:
```
#Boolean Type
print(True and False)
print(True or False)
print(not False)
print(not True)
print(True and False and True and True)
```

```
False
True
True
False
False
```

In [7]:
```
#Non-Boolean Type
print(10 and 20)
print(10 or 20)
print(0 and 20)
print(0 or 20)
print(not 0)
print(not 20)
```

```
20
10
0
20
True
False
```

## 3. Ternary (Conditional) Operators

**Syntax** : **x=first value if condition else second value**

- If condition true then first value will be considered else second value will be considred

In [8]:
```
#eg
a=10
b=20
x=30 if a>b else 40
print(x)
```

```
40
```

## 4. Assignment Operators

-> We can use assignment operator to assign value to the variable

In [10]:
```
x=10
x+=10 # 10+10
print(x)
x-=10 # 20-10
print(x)
x*=10 # 10*10
print(x)
x/=10 # 100/10 (When we use / it gives float val)
print(x)
x//=10 # 10.0//10
print(x)
x**=10
print(x)
```

```
20
10
100
10.0
1.0
1.0
```

## 5. Membership Operators

-> To check whether the given object is pressent in the given collection.
(String,List,Tuple,dict,set)

```
  in     returns True if given object present in the specified
collection.
not in   returns Ture if given object is not present in the
specified collection
```

In [15]:
```
x="Hello learing python is very easy"
print("i" in x)
print("python" in x)
print('Python' in x)
print("d" in x)
print("d" not in x)
```

```
True
True
False
False
True
```

In [17]:
```
a=[1,2,3,4,5]
print(3 in a)
```

```
True
```

| Operators | Associativity |
|---|---|
| () Highest precedence | Left - Right |
| ** | Right - Left |
| +x , -x, ~x | Left - Right |
| *, /, //, % | Left - Right |
| +, - | Left - Right |
| <<, >> | Left - Right |
| & | Left - Right |
| ^ | Left - Right |
| \| | Left - Right |
| Is, is not, in, not in,<br><, <=, >, >=, ==, != | Left - Right |
| Not x | Left - Right |
| And | Left - Right |
| Or | Left - Right |
| If else | Left - Right |
| Lambda | Left - Right |
| =, +=, -=, *=, /= Lowest Precedence | Right - Left |

In [20]:
```
print(7 and 0 or 5 and 3 or 7/0)
```

3

**Step 1**: 7 and 0
**Rule: x and y → if x is False → return x, else return y.**
Here 7 is True → so result is 0.
Expression becomes: print(0 or 5 and 3 or 7/0)

**Step 2**: 5 and 3
5 is True → return 3.
Expression becomes: print(0 or 3 or 7/0)

**Step 3**: 0 or 3
**Rule: x or y → if x is True → return x, else return y.**
0 is False → so result is 3.
Expression becomes: print(3 or 7/0)

**Step 4**: 3 or 7/0

First operand is 3 (True), so or returns 3 without evaluating 7/0.

In [19]:
```
5 and False or 3/0
5 and True or 3/0
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[19], line 1
----> 1 5 and False or 3/0
      2 5 and True or 3/0

ZeroDivisionError: division by zero
```

In [21]:
```
"1" in "123" and "False" or True
```

Out[21]: 'False'

In [22]:
```
print(7*5**2/True*False)
```

```
0.0
```

In [23]:
```
3/True
```

Out[23]: 3.0

In [24]:
```
10*4+5**2**2/10
```

Out[24]: 102.5

In [25]:
```
print(5==5.0 or 10 and 5 or 5 == 5.0 and 7!= 7.0)
```

```
True
```

In [29]:
```
new = (1 and "True") and ("False" or TRAIN)
str ="This statement is "+ new
print("This is False " if "false" in new else "This is True")
```

```
This is True
```

## 6. ord() and chr() Functions

- `ord()` converts character to ASCII
- `chr()` converts ASCII to character Char to Ascii

In [30]:
```
print(ord("A"))
print(chr(65))
```

```
65
A
```

In [31]:
```
for i in range (ord("a"),ord("z")+1):
      print(chr(i))
```

```
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
```

In [32]:
```
for i in range(65,91):
    print(chr(i))
```

```
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
```

In [33]:
```python
# write python progam to conver given date in year month and days
a=int(input("ENTER DAYS"))
y=a//365
m=(a%365)//30
d=(a%365)%30

print(y ," year" , m , "months " ,d, "days")
```

```
27  year 4 months  25 days
```

In [37]:
```python
# 10 20 50 100 200 500 notes i have

a=int(input("ENTER AMOUNT"))
n500=a//500
n200=(a%500)//200
n100=((a%500)%200)//100
n50=(((a%500)%200)%100)//50
n20=((((a%500)%200)%100)%50)//20
n10=(((((a%500)%200)%100)%50)%20)//10

print( n500,"* 500 ",n200,"* 200 ",n100,"* 100 ",n50,"* 50 ",n20,"* 20 ",n10,"
```

```
2 * 500  1 * 200  1 * 100  0 * 50  2 * 20  0 * 10
```