Mainly 3 Operators

1. Arithmatic

2. Relational

3. Logical

Other operators are - Assignment, Identity and Membership

1. Arithmatic Operators

Operator	Description	Example
+	Adds operands	a + b = 5
-	Subtracts right operand from left operand	a - b = 1
*	Multiplies both operands	a * b = 6
/	Quotient of division of left operand by right operand	a / b = 1.5 (float)
//	Quotient of division of left operand by right	a // b = 1 (int)
%	Remainder of division of left operand by right operand	a % b = 1
**	Left operand raised to the power of right operand	a ** b = 9

In [2]:
```
a = 11
b = 3
print("sum =", a + b)
print("difference =", a - b)
print("product =", a * b)
print("quotient =", a / b)
print("quotient (integer) =", a // b)
print("remainder =", a % b)
print("power =", a ** b)
```

```
sum = 14
difference = 8
product = 33
quotient = 3.6666666666666665
quotient (integer) = 3
remainder = 2
power = 1331
```

2. Relational Operators

Relational Operators check the relationship between two operands. They return True if the relationship is true and False if it is false.

Operator	Description	Example
==	Equal to	(a == b) is False
!=	Not equal to	(a != b) is True
>	Greater than	(a > b) is True
<	Less than	(a < b) is False
>=	Greater than or equal to	(a >= b) is True
<=	Less than or equal to	(a <= b) is False

In [11]:
```
a<b
```

Out[11]:
```
False
```

In [12]:
```
a>b
```

Out[12]:
```
True
```

In [13]:
```
a==b
```

Out[13]:
```
False
```

In [14]:
```
a!=b
```

Out[14]:
```
True
```

In [3]:
```
a = 3
b = 2
print("(a == b) :", a == b)
print("(a != b) :", a != b)
print("(a > b) :", a > b)
print("(a < b) :", a < b)
print("(a >= b) :", a >= b)
print("(a <= b) :", a <= b)
```

```
(a == b) : False
(a != b) : True
(a > b) : True
(a < b) : False
(a >= b) : True
(a <= b) : False
```

3. Logical Operators

and (&), or (/)

In [28]:
```
a=True
b=False
```

In [18]:
```
a&a #True&True is True
```

Out[18]:
```
True
```

In [19]:
```
a&b
```

Out[19]:
```
False
```

In [20]:
```
a|b
```

Out[20]:
```
True
```

In [21]:
```
b|a
```

Out[21]:
```
True
```

In [ ]:
```
#Why True Here? In the mix of operands, if any any operand has stored true in it the final result will always b
#In this case, among 'a' & 'b', 'a' has stared 'Ture'
```

In [29]:
```
a|a
```

Out[29]:
```
True
```

In [30]:
```
b|b
```

Out[30]:
```
False
```

Exp1	Operator	Exp1	Output (Boolean)
True	and	True	True
True	and	False	False
False	and	False	False
True	or	True	True
True	or	false	True
False	or	False	False

In [4]:
```
x = 10
y = 20
print(x == 10 and y == 20)
print(x == 3 or y == 20)
```

```
True
True
```

In [5]:
```
x = 10
y = 20
print(not(x == 10 and y == 20))
print(not(x == 3 or y == 20))
```

```
False
False
```