Take 2 inputs from the user and compare whether they are equal or not

In [1]:

```
a = int(input('Enter no: '))
b = int(input('Enter no: '))
print(a==b)
```

```
Enter no: 4
Enter no: 5
False
```

Logical Operators

AND : `TRUE` : to get True results

truth table of and

In [2]:

```
True and True , True and False, False and True, False and False
```

Out[2]:

```
(True, False, False, False)
```

In [3]:

```
1 & 1, 1 & 0, 0 & 1 , 0 & 0
```

Out[3]:

```
(1, 0, 0, 0)
```

OR : `FALSE` : both situtaions are false

In [4]:

```
True or False, False or True, False or False, True or True
```

Out[4]:

```
(True, True, False, True)
```

In [5]:

```
1|0   , 0|0,    1|1,    0|1
```

Out[5]:

```
(1, 0, 1, 1)
```

Take 3 inputs from the user and compare whether any of them are equal or not

In [8]:

```
a = int(input('Enter no: '))
b = int(input('Enter no: '))
c = int(input('Enter no: '))
any_no = a==b or b==c or c==a
print(any_no)
```

```
Enter no: 2
Enter no: 3
Enter no: 2
True
```

In [9]:

```
score1 = 35.4
score2 = 35.6
```

round off

In [10]:

```
round(score1), round(score2)
```

Out[10]:

```
(35, 36)
```

In [16]:

```
Eq(sqrt(x), x**0.5)
```

Out[16]:

$$\sqrt{x} = x^{0.5}$$

In [17]:

```
sqrt(8)
```

Out[17]:

$$2\sqrt{2}$$

In [18]:

```
8**0.5
```

Out[18]:

```
2.8284271247461903
```

In [20]:

```
8**(1/2)
```

Out[20]:

```
2.8284271247461903
```

In [21]:

```
import math
math.sqrt(8)
```

Out[21]:

```
2.8284271247461903
```

In [22]:

```
import cmath
num = eval(input('Enter number: '))
print('sqrt: ',cmath.sqrt(num))
```

```
Enter number: 7+8j
sqrt:  (2.9690188457413544+1.34724641634978j)
```

In [23]:

```
print(dir(cmath))
```

```
['__doc__', '__file__', '__loader__', '__name__', '__package__', '__sp
ec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atanh', 'cos', 'cos
h', 'e', 'exp', 'inf', 'infj', 'isclose', 'isfinite', 'isinf', 'isna
n', 'log', 'log10', 'nan', 'nanj', 'phase', 'pi', 'polar', 'rect', 'si
n', 'sinh', 'sqrt', 'tan', 'tanh', 'tau']
```

In [25]:

```
Eq(a*x**2 + b*x + c, 0)
```

Out[25]:

$$ax^2 + bx + c = 0$$

In [26]:

```
Dis = b**2 - 4*a*c
Dis
```

Out[26]:

$$-4ac + b^2$$

In [ ]:

```
Sol1 = (-b-cmath.sqrt(Dis))/(2*a)
Sol1 = (-b+cmath.sqrt(Dis))/(2*a)
```

In [30]:

```
r = eval(input('Enter number: '))
s = eval(input('Enter number: '))
t = eval(input('Enter number: '))

D = s**2 - 4*r*t

Sol1 = (-s-cmath.sqrt(D))/(2*r)
Sol2 = (-s+cmath.sqrt(D))/(2*r)
print('Solutions are', Sol1,'and', Sol2)
```

```
Enter number: 2
Enter number: 4
Enter number: 6
Solutions are (-1-1.4142135623730951j) and (-1+1.4142135623730951j)
```

In [31]:

```
r,s,t
```

Out[31]:

```
(2, 4, 6)
```

In [35]:

```
ex1 = Eq(2*x**2 + 4*x + 6, 0)
ex1
```

Out[35]:

$$2x^2 + 4x + 6 = 0$$

In [36]:

```
solve(ex1)
```

Out[36]:

```
[-1 - sqrt(2)*I, -1 + sqrt(2)*I]
```

In [37]:

```
-1 - sqrt(2)*I
```

Out[37]:

$$-\sqrt{2}I - 1$$

In [38]:

```
-1 + sqrt(2)*I
```

Out[38]:

$$\sqrt{2}I - 1$$

Assignment Operators

In [39]:

```
word = 'ice'
word += ' cream'
word
```

Out[39]:

```
'ice cream'
```

In [40]:

```
x1 = 5
x1 += 5
x1
```

Out[40]:

```
10
```

In [41]:

```
x1 *= 5
x1
```

Out[41]:

```
50
```

In [42]:

```
x1 /= 5
x1
```

Out[42]:

```
10.0
```

In [43]:

```
x1 %= 3
x1
```

Out[43]:

```
1.0
```

membership Operators

```
### len('Intelligence')
```

In [50]:

```
'g' in 'Intelligence'
```

Out[50]:

```
True
```

In [51]:

```
files = ['pdf','txt','csv']
```

In [52]:

```
'jpg' in files
```

Out[52]:

```
False
```

In [53]:

```
'gif' not in files
```

Out[53]:

```
True
```

Identity Operators

In [54]:

```
id(files)
```

Out[54]:

```
140232373237152
```

Ways of printing

In [55]:

```
x1 = 4
x2 = 5
```

In [56]:

```
print('x1 = {} and x2 = {}'.format(x1,x2))
```

```
x1 = 4 and x2 = 5
```

In [57]:

```
print('x1 = %d and x2 = %d' %(x1,x2))
```

```
x1 = 4 and x2 = 5
```

In [58]:

```
print('x1 = %f and x2 = %f' %(x1,x2))
```

```
x1 = 4.000000 and x2 = 5.000000
```

In [59]:

```
print('x1 = %s and x2 = %s' %(x1,x2))
```

```
x1 = 4 and x2 = 5
```

In [63]:

```
print('x1 =',x1 ,'and x2 =',x2)
```

```
x1 = 4 and x2 = 5
```

In [64]:

```
import calendar
```

In [65]:

```
calendar.isleap(2021)
```

Out[65]:

```
False
```

In [66]:

```
calendar.isleap(2000)
```

Out[66]:

```
True
```

In [67]:

```
calendar.leapdays(2000, 2020)
```

Out[67]:

```
5
```

In [68]:

```
print(dir(calendar))
```

```
['Calendar', 'EPOCH', 'FRIDAY', 'February', 'HTMLCalendar', 'IllegalMo
nthError', 'IllegalWeekdayError', 'January', 'LocaleHTMLCalendar', 'Lo
caleTextCalendar', 'MONDAY', 'SATURDAY', 'SUNDAY', 'THURSDAY', 'TUESDA
Y', 'TextCalendar', 'WEDNESDAY', '_EPOCH_ORD', '__all__', '__builtins_
_', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__
package__', '__spec__', '_colwidth', '_locale', '_localized_day', '_lo
calized_month', '_spacing', 'c', 'calendar', 'datetime', 'day_abbr',
'day_name', 'different_locale', 'error', 'firstweekday', 'format', 'fo
rmatstring', 'isleap', 'leapdays', 'main', 'mdays', 'month', 'month_ab
br', 'month_name', 'monthcalendar', 'monthlen', 'monthrange', 'nextmon
th', 'prcal', 'prevmonth', 'prmonth', 'prweek', 'repeat', 'setfirstwee
kday', 'sys', 'timegm', 'week', 'weekday', 'weekheader']
```

In [69]:

```python
print('this is \nis a new line')
```

```
this is 
is a new line
```

In [70]:

```python
calendar.month(2021, 2)
```

Out[70]:

```
'   February 2021\nMo Tu We Th Fr Sa Su\n 1  2  3  4  5  6  7\n 8  9 1
0 11 12 13 14\n15 16 17 18 19 20 21\n22 23 24 25 26 27 28\n'
```

In [71]:

```python
print('   February 2021\nMo Tu We Th Fr Sa Su\n 1  2  3  4  5  6  7\n 8  9 10 11
```

```
   February 2021
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

In [ ]:

```python

```