```
In [1]: 1 + 1

Out[1]: 2

In [2]: 2-1

Out[2]: 1

In [3]: 3*4

Out[3]: 12

In [4]: 8/4

Out[4]: 2.0

In [5]: 8/6

Out[5]: 1.3333333333333333

In [6]: 8//7

Out[6]: 1

In [7]: 8 + 9 - 7

Out[7]: 10

In [8]: 5 + 5 * 5

Out[8]: 30

In [9]: (5 + 5) * 5

Out[9]: 50

In [10]: 2 * 2 * 2 * 2 * 2

Out[10]: 32

In [11]: 2 ** 5

Out[11]: 32
```

In [12]:

```
a,b,c,d,e = 15, 7.8, 'nit', 8+9j, True

print(a)
print(b)
print(c)
print(d)
print(e)
```

```
15
7.8
nit
(8+9j)
True
```

In [13]:

```
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'complex'>
<class 'bool'>
```

In [14]:

```
type(c)
```

Out[14]:

```
str
```

In [15]:

```
'Janhavi'
```

Out[15]:

```
'Janhavi'
```

In [16]:

```
print('Janhavi')
```

```
Janhavi
```

In [17]:

```
"Janhavi Landge"
```

Out[17]:

```
'Janhavi Landge'
```

In [18]:

```
s1 = "Janhavi Landge"
s1
```

Out[18]:

```
'Janhavi Landge'
```

In [19]:

```
a = 2
b = 3

a + b
```

Out[19]:

```
5
```

In [20]:
```
c = a + b
c
```
Out[20]:
```
5
```

In [21]:
```
a = 3
b = 'hi'
type(b)
```
Out[21]:
```
str
```

In [22]:
```
print('Datascience it\'s"Technology"')
```
```
Datascience it's"Technology"
```

In [23]:
```
print ('Datascience it','Technology"')
```
```
Datascience it Technology"
```

In [24]:
```
'VNIT' + 'VNIT'
```
Out[24]:
```
'VNITVNIT'
```

In [25]:
```
  'VNIT'  ' VNIT'
```
Out[25]:
```
'VNIT VNIT'
```

In [26]:
```
5 * "VNIT"
```
Out[26]:
```
'VNITVNITVNITVNITVNIT'
```

In [27]:
```
print('c:\VNIT')
```
```
c:\VNIT
```

In [28]:
```
print(r'c:\VNIT')
```
```
c:\VNIT
```

In [29]:
```
2
```
Out[29]:
```
2
```

In [30]:
```
x = 2
x
```
Out[30]:
```
2
```

In [31]:
```
x + 3
```
Out[31]:
```
5
```

```
In [32]: y =3
         y

Out[32]: 3

In [33]: x + y

Out[33]: 5

In [34]: x = 9
         x

Out[34]: 9

In [35]: x + y

Out[35]: 12

In [36]: x + 10

Out[36]: 19

In [37]: y

Out[37]: 3

In [38]: _ + y

Out[38]: 6

In [39]: _-8

Out[39]: -2

In [40]: name = 'mit'

In [41]: name

Out[41]: 'mit'

In [42]: name + 'technology'

Out[42]: 'mittechnology'

In [43]: name + ' technology'

Out[43]: 'mit technology'

In [44]: name

Out[44]: 'mit'
```

```
In [45]: len(name)

Out[45]: 3
```

```
In [46]: name[0]

Out[46]: 'm'
```

```
In [47]: name[-1]

Out[47]: 't'
```

```
In [48]: name[-2]

Out[48]: 'i'
```

slicing

```
In [49]: name

Out[49]: 'mit'
```

```
In [50]: name[0:1]

Out[50]: 'm'
```

```
In [51]: name[0:2]

Out[51]: 'mi'
```

```
In [52]: name[1:4]

Out[52]: 'it'
```

```
In [53]: name

Out[53]: 'mit'
```

```
In [54]: name[1:]

Out[54]: 'it'
```

```
In [55]: name[:4]

Out[55]: 'mit'
```

```
In [56]: name[3:9]

Out[56]: ''
```

```
In [57]: name1 = 'fine'
         name1

Out[57]: 'fine'

In [58]: name1[0:1]

Out[58]: 'f'

In [59]: name1

Out[59]: 'fine'

In [60]: name1[1:]

Out[60]: 'ine'

In [61]: 'd' + name1[1:]

Out[61]: 'dine'

In [62]: len(name1)

Out[62]: 4
```

List

```
In [63]: nums = [10,20,30]
         nums

Out[63]: [10, 20, 30]

In [64]: nums[0]

Out[64]: 10

In [65]: nums[-1]

Out[65]: 30

In [66]: nums[1:]

Out[66]: [20, 30]

In [67]: nums[:1]

Out[67]: [10]

In [68]: num1 = ['hi', 'hello']
In [69]: num1

Out[69]: ['hi', 'hello']

In [70]: num1

Out[70]: ['hi', 'hello']

In [71]: num2 = ['hi', 8.9, 34]
         num2

Out[71]: ['hi', 8.9, 34]

In [72]: num3 = [nums, num1]

In [73]: num3

Out[73]: [[10, 20, 30], ['hi', 'hello']]

In [74]: num4 = [nums, num1, num2]

In [75]: num4

Out[75]: [[10, 20, 30], ['hi', 'hello'], ['hi', 8.9, 34]]

In [76]: nums

Out[76]: [10, 20, 30]

In [77]: nums.append(45)

In [78]: nums

Out[78]: [10, 20, 30, 45]

In [79]: nums.remove(45)

In [80]: nums

Out[80]: [10, 20, 30]

In [81]: nums.pop(1)

Out[81]: 20

In [82]: nums

Out[82]: [10, 30]
```

```
In [83]: nums.pop()

Out[83]: 30

In [84]: nums

Out[84]: [10]

In [85]: num1

Out[85]: ['hi', 'hello']

In [86]: num1.insert(2,'nit')

In [87]: num1

Out[87]: ['hi', 'hello', 'nit']

In [88]: num1.insert(0, 1)

In [89]: num1

Out[89]: [1, 'hi', 'hello', 'nit']

In [90]: num2

Out[90]: ['hi', 8.9, 34]

In [91]: del num2[2:]

In [92]: num2

Out[92]: ['hi', 8.9]

In [93]: num2.extend([29,15,20])

In [94]: num2

Out[94]: ['hi', 8.9, 29, 15, 20]

In [95]: num3

Out[95]: [[10], [1, 'hi', 'hello', 'nit']]

In [96]: num3.extend(['a', 5, 6.7])

In [97]: num3

Out[97]: [[10], [1, 'hi', 'hello', 'nit'], 'a', 5, 6.7]
```

```
In [98]: nums

Out[98]: [10]

In [99]: min(nums) #inbuild function

Out[99]: 10

In [100]: max(nums) #inbuild function

Out[100]: 10

In [101]: num1

Out[101]: [1, 'hi', 'hello', 'nit']

In [102]: sum(nums) #inbuild function

Out[102]: 10

In [103]: nums.sort() #sort method

In [104]: nums

Out[104]: [10]
```

Tuple

```
In [105]: tup = (15,25, 35)
          tup

Out[105]: (15, 25, 35)

In [106]: tup[0]

Out[106]: 15
```

SET

```
In [107]: S = {}

In [108]: s1 = {21,6,34,58,5}

In [109]: s1

Out[109]: {5, 6, 21, 34, 58}
```

```
In [110]: s3= {50,35,53,'nit', 53}
```

```
In [111]: s3
```

```
Out[111]: {35, 50, 53, 'nit'}
```

Dictionary

```
In [112]: # DICTIONARY DICTIONARY DICTIONARY
          data = {1:'apple', 2:'banana',4:'orange'}
          data
```

```
Out[112]: {1: 'apple', 2: 'banana', 4: 'orange'}
```

```
In [113]: data[4]
```

```
Out[113]: 'orange'
```

```
In [114]: data.get(2)
```

```
Out[114]: 'banana'
```

```
In [115]: data.get(3)
```

```
In [116]: print(data.get(3))
```

```
None
```

```
In [117]: data.get(1,'Not Fount')
```

```
Out[117]: 'apple'
```

```
In [118]: data.get(3,'Not Found')
```

```
Out[118]: 'Not Found'
```

```
In [119]: data
```

```
Out[119]: {1: 'apple', 2: 'banana', 4: 'orange'}
```

```
In [120]: data
```

```
Out[120]: {1: 'apple', 2: 'banana', 4: 'orange'}
```

```
In [121]: #list in the dictionary
          prog = {'python':['vscode', 'pycharm'], 'machine learning' : 'sklearn', 'datas
```

```
In [122]: prog

Out[122]: {'python': ['vscode', 'pycharm'],
           'machine learning': 'sklearn',
           'datascience': ['jupyter', 'spyder']}

In [123]: prog['python']

Out[123]: ['vscode', 'pycharm']

In [124]: prog['machine learning']

Out[124]: 'sklearn'

In [125]: prog['datascience']

Out[125]: ['jupyter', 'spyder']

In [126]: # variable address
          num = 5
          id(num)

Out[126]: 140733038052264

In [127]: name = 'nit'
          id(name) #Address will be different for both

Out[127]: 2234066311280

In [128]: a = 10
          id(a)

Out[128]: 140733038052424

In [129]: b = a

In [130]: id(b)

Out[130]: 140733038052424

In [131]: id(10)

Out[131]: 140733038052424

In [132]: k = 10
          id(k)

Out[132]: 140733038052424

In [133]: a = 20  # as we change the value of a then address will change
          id(a)

Out[133]: 140733038052744
```

```
In [134]: id(b)

Out[134]: 140733038052424

In [135]: PI = 3.14  #in math this is alway constant but python we can chang
          PI

Out[135]: 3.14

In [136]: PI = 3.18
          PI

Out[136]: 3.18

In [137]: type(PI)

Out[137]: float

In [138]: w = 2.5
          type(w)

Out[138]: float

In [139]: a

Out[139]: 20
```

Complex

```
In [140]: w2 = 2 + 3j
          type(w2)

Out[140]: complex

In [141]: 
          a = 5.6
          b = int(a)

In [142]: b

Out[142]: 5

In [143]: type(b)

Out[143]: int

In [144]: type(a)

Out[144]: float
```

In [145]:
```
k = float(b)
```

In [146]:
```
k
```

Out[146]:
```
5.0
```

In [147]:
```
print(a)
print(b)
print(k)
```

```
5.6
5
5.0
```

In [148]:
```
k1 = complex(b,k)
```

In [149]:
```
print(k1)
type(k1)
```

```
(5+5j)
```

Out[149]:
```
complex
```

In [150]:
```
b < k
```

Out[150]:
```
False
```

In [151]:
```
condition = b<k
condition
```

Out[151]:
```
False
```

In [152]:
```
type(condition)
```

Out[152]:
```
bool
```

In [153]:
```
int(True)
```

Out[153]:
```
1
```

In [154]:
```
int(False)
```

Out[154]:
```
0
```

In [155]:
```
l = [1,2,3,4]
print(l)
type(l)
```

```
[1, 2, 3, 4]
```

Out[155]:
```
list
```

In [156]:
```
s = {1,2,3,4}
s
```

Out[156]:
```
{1, 2, 3, 4}
```

In [157]:
```
type(s)
```

Out[157]:
```
set
```

In [158]:
```
s1 = {1,2,3,4,4,3,11} #duplicates are not allowed
s1
```

Out[158]:
```
{1, 2, 3, 4, 11}
```

In [159]:
```
t = (10,20,30)
t
```

Out[159]:
```
(10, 20, 30)
```

In [160]:
```
type(t)
```

Out[160]:
```
tuple
```

In [161]:
```
str = 'nit'
type(str)
```

Out[161]:
```
str
```

In [162]:
```
st = 'n'
type(st)
```

Out[162]:
```
str
```

range()

In [163]:
```
r = range(0,10)
r
```

Out[163]:
```
range(0, 10)
```

In [164]:
```
type(r)
```

Out[164]:
```
range
```

In [165]:
```
list(range(0,10))
```

Out[165]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [166]:
```
r1 = list(r)
r1
```

Out[166]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [167]:
```

even_number = list(range(2,10,2))
even_number
```

Out[167]:
```
[2, 4, 6, 8]
```

In [168]:
```
odd_number = list(range(1,9,2))
odd_number
```

Out[168]:
```
[1, 3, 5, 7]
```

In [169]:
```
odd_number = list(range(1,8,2))
odd_number
```

Out[169]:
```
[1, 3, 5, 7]
```

In [170]:
```
d= {1:'one', 2:'two', 3:'three'}
d
```

Out[170]:
```
{1: 'one', 2: 'two', 3: 'three'}
```

In [171]:
```
type(d)
```

Out[171]:
```
dict
```

In [172]:
```
d.keys()
```

Out[172]:
```
dict_keys([1, 2, 3])
```

In [173]:
```
d.values()
```

Out[173]:
```
dict_values(['one', 'two', 'three'])
```

In [174]:
```
d[2]
```

Out[174]:
```
'two'
```

In [175]:
```
d.get(2)
```

Out[175]:
```
'two'
```

Arithmetic operator

In [176]:
```
x1, y1 =10, 5
```

In [177]:
```
x1   + y1
```

Out[177]:
```
15
```

```
In [178]: x1 - y1

Out[178]: 5

In [179]: x1 * y1

Out[179]: 50

In [180]: x1 / y1

Out[180]: 2.0

In [181]: x1 // y1

Out[181]: 2

In [182]: x1 % y1

Out[182]: 0

In [183]: x1 ** y1

Out[183]: 100000

In [184]: 2 ** 3

Out[184]: 8
```

Assignment operator

```
In [185]: x = 2

In [186]: x = x + 2

In [187]: x

Out[187]: 4

In [188]: x += 2

In [189]: x

Out[189]: 6

In [190]: x+=2

In [191]: x

Out[191]: 8
```

```
In [192]: x*=2
```

```
In [193]: x
```

```
Out[193]: 16
```

```
In [194]: x-=2
```

```
In [195]: x
```

```
Out[195]: 14
```

```
In [196]: x /=2
```

```
In [197]: x
```

```
Out[197]: 7.0
```

```
In [198]: a, b = 5,6
```

```
In [199]: a
```

```
Out[199]: 5
```

```
In [200]: b
```

```
Out[200]: 6
```

Unary Operator

```
In [201]: n = 7 #negattion
```

```
In [202]: m = -(n)
```

```
In [203]: m
```

```
Out[203]: -7
```

```
In [204]: n
```

```
Out[204]: 7
```

```
In [205]: -n
```

```
Out[205]: -7
```

Relational operator

```
In [206]: a = 5
          b = 7

In [207]: a == b

Out[207]: False

In [208]: a<b

Out[208]: True

In [209]: a>b

Out[209]: False

In [210]: a == b

Out[210]: False

In [211]: a = 10

In [212]: a != b

Out[212]: True

In [213]: b = 10

In [214]: a == b

Out[214]: True

In [215]: a>=b

Out[215]: True

In [216]: a<=b

Out[216]: True

In [217]: a>b

Out[217]: False

In [218]: a<b

Out[218]: False

In [219]: b = 7
In [220]: a!=b

Out[220]: True
```

Logical Operator

```
In [221]: a = 5
          b = 4
```

```
In [222]: a < 8 and b < 5

Out[222]: True
```

```
In [223]: a < 8 and b < 2

Out[223]: False
```

```
In [224]: a>8 or b<2

Out[224]: False
```

```
In [225]: x = False
          x

Out[225]: False
```

```
In [226]: x = not x
          x

Out[226]: True
```

```
In [227]: x

Out[227]: True
```

```
In [228]: not x

Out[228]: False
```

```
In [229]: x

Out[229]: True
```

```
In [230]: not x

Out[230]: False
```

```
In [231]: 25

Out[231]: 25
```

```
In [232]: bin(25)

Out[232]: '0b11001'

In [233]: 0b11001

Out[233]: 25

In [234]: int(0b11001)

Out[234]: 25

In [235]: bin(35)

Out[235]: '0b100011'

In [236]: int(0b100011)

Out[236]: 35

In [237]: bin(20)

Out[237]: '0b10100'

In [238]: int(0b10100)

Out[238]: 20

In [239]: 0b1111

Out[239]: 15

In [240]: oct(15)

Out[240]: '0o17'

In [241]: 0o17

Out[241]: 15

In [242]: hex(9)

Out[242]: '0x9'

In [243]: 0xf

Out[243]: 15

In [244]: hex(10)

Out[244]: '0xa'
```

In [245]:
```
0xa
```

Out[245]:
```
10
```

In [246]:
```
hex(25)
```

Out[246]:
```
'0x19'
```

In [247]:
```
0x19
```

Out[247]:
```
25
```

In [248]:
```
0x15
```

Out[248]:
```
21
```

Swap Variable

In [249]:
```
a = 5
b = 6
```

In [250]:
```
a = b
b = a
```

In [251]:
```
a,b = b,a
```

In [252]:
```
print(a)
print(b)
```

```
6
6
```

In [253]:
```
a1 = 7
b1 = 8
```

In [254]:
```
temp = a1
a1 = b1
b1 = temp
```

In [255]:
```
print(a1)
print(b1)
```

```
8
7
```

In [256]:
```
a2 = 5
b2 = 6
```

In [257]:

```
a2 = a2 + b2    #11
b2 = a2 - b2    #5
a2 = a2 - b2    #6
```

In [258]:

```
print(a2)
print(b2)
```

```
6
5
```

In [259]:

```
print(bin(11))
print(0b1011)
```

```
0b1011
11
```

XOR Swapping

In [260]:

```
a2 = a2 ^ b2
b2 = a2 ^ b2
a2 = a2 ^ b2
```

In [261]:

```
print(a2)
print(b2)
```

```
5
6
```

In [262]:

```
a2 , b2 = b2, a2
```

In [263]:

```
print(a2)
print(b2)
```

```
6
5
```

Bitwise Operator

In [264]:

```
print(bin(12))
print(bin(13))
```

```
0b1100
0b1101
```

In [265]:

```
~45
```

Out[265]:

```
-46
```

```
In [266]: ~6

Out[266]: -7

In [267]: ~-6

Out[267]: 5

In [268]: ~-1

Out[268]: 0

In [269]: 12 & 13

Out[269]: 12

In [270]: 1 & 1

Out[270]: 1

In [271]: 1 | 0

Out[271]: 1

In [272]: 1 & 0

Out[272]: 0

In [273]: 12 | 13

Out[273]: 13

In [274]: 35 & 40

Out[274]: 32

In [275]: 35 | 40

Out[275]: 43

In [276]: 12 ^ 13

Out[276]: 1

In [277]: 25 ^ 30

Out[277]: 7

In [278]: bin(25)

Out[278]: '0b11001'
```

```
In [279]: bin(30)
```

```
Out[279]: '0b11110'
```

```
In [280]: int(0b000111)
```

```
Out[280]: 7
```

Bitwise Left Shift Opertor

```
In [281]: 20<<4
```

```
Out[281]: 320
```

Bitwise Right Shift Operator

```
In [282]: 10>>2
```

```
Out[282]: 2
```

```
In [283]: bin(20)
```

```
Out[283]: '0b10100'
```

```
In [284]: 20>>4
```

```
Out[284]: 1
```

import math module

```
In [285]: import math
```

```
In [286]: x = math.sqrt(25)
          x
```

```
Out[286]: 5.0
```

```
In [287]: x1 = math.sqrt(25)
          x1
```

```
Out[287]: 5.0
```

```
In [288]: print(math.floor(2.9))
```

```
2
```

In [289]:

```
print(math.ceil(2.9))
```

```
3
```

In [290]:

```
print(math.pow(3,2))
```

```
9.0
```

In [291]:

```
print(math.pi)
```

```
3.141592653589793
```

In [292]:

```
print(math.e)
```

```
2.718281828459045
```

In [293]:

```
import math as m
m.sqrt(10)
```

Out[293]:

```
3.1622776601683795
```

In [294]:

```
from math import sqrt,pow
pow(2,3)
```

Out[294]:

```
8.0
```

In [295]:

```
round(pow(2,3))
```

Out[295]:

```
8
```

In [3]:

```
x = input()
y = input()
z = x + y
print(z)
```

```
5
8
58
```

In [4]:

```
x1 = input('Enter the 1st number')
y1 = input('Enter the 2nd number')
z1 = x1 + y1
print(z1)
```

```
Enter the 1st number6
Enter the 2nd number8
68
```

In [5]:

```
type(x1)
type(y1)
```

Out[5]:

```
str
```

In [6]:

```
x1 = input('Enter the 1st number')
a1 = int(x1)
y1 = input('Enter the 2nd number')
b1 = int(y1)
z1 = a1 + b1
print(z1)
```

```
Enter the 1st number5
Enter the 2nd number6
11
```

In [7]:

```
x2 = int(input('Enter the 1st number'))
y2 = int(input('Enter the 2nd number'))
z2 = x2 + y2
z2
```

```
Enter the 1st number6
Enter the 2nd number7
```

Out[7]:

```
13
```

In [8]:

```
ch = input('enter a char')
print(ch)
```

```
enter a charJanhavi
Janhavi
```

In [9]:

```
print(ch[0])
```

```
J
```

In [10]:

```
print(ch[1])
```

```
a
```

In [11]:

```
print(ch[-1])
```

```
i
```

EVAL

In [21]:

```
result = eval(input('enter an expr'))
print(result)
```

```
enter an expr2 + 5-(4)
3
```