In [2]:

```
5 * 5
```

Out[2]:

```
25
```

In [3]:

```
45
```

Out[3]:

```
45
```

In [4]:

```
complex(45)
```

Out[4]:

```
(45+0j)
```

Input Functions

In [5]:

```
name = input('Enter Name: ')
```

```
Enter Name: Sumit Kumar
```

In [6]:

```
name
```

Out[6]:

```
'Sumit Kumar'
```

In [7]:

```
import random
print(dir(random))
```

```
['BPF', 'LOG4', 'NV_MAGICCONST', 'RECIP_BPF', 'Random', 'SG_MAGICCONS
T', 'SystemRandom', 'TWOPI', '_BuiltinMethodType', '_MethodType', '_Se
quence', '_Set', '__all__', '__builtins__', '__cached__', '__doc__',
'__file__', '__loader__', '__name__', '__package__', '__spec__', '_aco
s', '_bisect', '_ceil', '_cos', '_e', '_exp', '_inst', '_itertools',
'_log', '_os', '_pi', '_random', '_sha512', '_sin', '_sqrt', '_test',
'_test_generator', '_urandom', '_warn', 'betavariate', 'choice', 'choi
ces', 'expovariate', 'gammavariate', 'gauss', 'getrandbits', 'getstat
e', 'lognormvariate', 'normalvariate', 'paretovariate', 'randint', 'ra
ndom', 'randrange', 'sample', 'seed', 'setstate', 'shuffle', 'triangul
ar', 'uniform', 'vonmisesvariate', 'weibullvariate']
```

Ranges

In [8]:

```
1, 2, 3, 4 ,5, 6 ,7, 8, 9, 10
```

Out[8]:

```
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

In [10]:

```
for i in range(25):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
```

In [11]:

```
range(0, 5)
```

Out[11]:

```
range(0, 5)
```

In [12]:

```
list(range(0,5))
```

Out[12]:

```
[0, 1, 2, 3, 4]
```

In [14]:

```
list(range(-5,5+1))
```

Out[14]:

```
[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
```

In [16]:

```
range(0, 20, 2) #range(start, stop, [interval/step])
```

Out[16]:

```
range(0, 20, 2)
```

In [18]:

```
list(range(0, 20+2, 2))
```

Out[18]:

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

In [19]:

```
help('random.randrange')
```

```
Help on method randrange in random:

random.randrange = randrange(start, stop=None, step=1, _int=<class 'in
t'>) method of random.Random instance
    Choose a random item from range(start, stop[, step]).

    This fixes the problem with randint() which includes the
    endpoint; in Python this is usually not what you want.
```

In [20]:

```
# endpoint = True, stop = exactly same,and false then stop - 1
```

In [21]:

```
random.randrange(0,5)
```

Out[21]:

```
3
```

In [22]:

```
random.randrange(-5, 5)
```

Out[22]:

```
-4
```

In [26]:

```
random.randrange(1234, 5678)
```

Out[26]:

```
1654
```

In [27]:

```
low = 'abcdefghijklmnopqrstuvwxyz'
upp = low.upper()
num = '0123456789'
spc = '!@#%&*'

pas = low + upp + num + spc
```

In [28]:

```
pas
```

Out[28]:

```
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*'
```

In [37]:

```
# 5 digit password
''.join(random.sample(pas,3))
```

Out[37]:

```
'DZX'
```

In [33]:

```
random.sample(pas, 10)
```

Out[33]:

```
['o', 'J', 'D', 'l', '%', 'N', '4', 'k', 'L', 'F']
```

Arithmetic Ops

In [38]:

```
# add
x,y = 5, 4
x + y
```

Out[38]:

```
9
```

In [40]:

```
# sub
x - y, y - x
```

Out[40]:

```
(1, -1)
```

In [41]:

```
# mul
x * y, y * x
```

Out[41]:

```
(20, 20)
```

In [43]:

```
# div - (float value)
x / y, y/ x
```

Out[43]:

```
(1.25, 0.8)
```

In [44]:

```
# intger division(floor division)
x // y, y //x
```

Out[44]:

```
(1, 0)
```

In [45]:

```
# exponent
x **y, y **x
```

Out[45]:

```
(625, 1024)
```

In [46]:

```
# remainder
x % y, y % x
```

Out[46]:

```
(1, 4)
```

In [47]:

```
100 % 3
```

Out[47]:

```
1
```

g-codes

In [53]:

```
print('G -' + str(random.randint(123456, 987654)) + ' is your One Time Google ve
```

```
G -755320 is your One Time Google verification Code.
```

In [57]:

```python
import numpy, pandas
code = []
for i in numpy.arange(10000):
    gen = 'G -' + str(numpy.random.randint(123456, 987654))
    code.append(gen)
gcodes = numpy.array(code).reshape(1000,10)
df_gcodes = pandas.DataFrame(data = gcodes)
```

In [58]:

```python
df_gcodes
```

Out[58]:

	0	1	2	3	4	5	6	7	8	9
0	G -621827	G -378040	G -833637	G -530976	G -592976	G -138706	G -649036	G -208583	G -180745	G -275315
1	G -734040	G -336361	G -606830	G -885246	G -727671	G -166114	G -748423	G -330421	G -871932	G -692037
2	G -833243	G -883243	G -983254	G -732795	G -462528	G -337226	G -853583	G -845952	G -269457	G -491253
3	G -497505	G -353956	G -321886	G -151619	G -263237	G -508971	G -977754	G -696261	G -970299	G -226868
4	G -646465	G -447877	G -893745	G -272109	G -887206	G -427495	G -330347	G -287041	G -506778	G -347723
...	...	...	...	...	...	...	...	...	...	...
995	G -855243	G -721190	G -551258	G -174888	G -800382	G -222251	G -919876	G -979547	G -495847	G -575298
996	G -300370	G -278745	G -574918	G -615881	G -522420	G -720040	G -581091	G -630189	G -554821	G -569052
997	G -671853	G -314912	G -502196	G -767583	G -215041	G -236138	G -724366	G -159765	G -610359	G -611121
998	G -785110	G -186422	G -697507	G -955884	G -509847	G -521275	G -652326	G -590142	G -432877	G -767552
999	G -770737	G -575845	G -199137	G -928998	G -852040	G -888427	G -346798	G -755989	G -263809	G -467542

1000 rows × 10 columns

In [59]:

```python
df_gcodes.to_csv('sample_generated_gcodes.csv')
```

Strings

replication , concatenation

In [67]:

```
'word ' * 5
```

Out[67]:

```
'word word word word word '
```

In [68]:

```
x = 'word'
x += ' game' # x+= means x = x + whatever
```

In [69]:

```
x
```

Out[69]:

```
'word game'
```

len Function

In [70]:

```
len('word')
```

Out[70]:

```
4
```

In [73]:

```
len(code)
```

Out[73]:

```
10000
```

Comparison OPs

In [74]:

```
#greater than
3 > 5
```

Out[74]:

```
False
```

In [75]:

```
# smaller than
3 < 5
```

Out[75]:

```
True
```

In [76]:

```
# equals checing
3 == 5
```

Out[76]:

```
False
```

In [77]:

```
# not equals
3 != 5
```

Out[77]:

```
True
```

In [78]:

```
# greater than equal to
3 >= 5
```

Out[78]:

```
False
```

In [79]:

```
# lesser than equal to
3 <= 5
```

Out[79]:

```
True
```

In [80]:

```
import calendar
month = 2
year = 2021
print(calendar.month(year, month))
```

```
   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 [81]:

```
print(calendar.month(2022, 2))
```

```
   February 2022
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 [82]:

```
print(calendar.calendar(2021))
```

```
                                  2021

      January                   February                   March
Mo Tu We Th Fr Sa Su       Mo Tu We Th Fr Sa Su       Mo Tu We Th Fr Sa
Su
             1  2  3        1  2  3  4  5  6  7        1  2  3  4  5  6
7
 4  5  6  7  8  9 10        8  9 10 11 12 13 14        8  9 10 11 12 13
14
11 12 13 14 15 16 17       15 16 17 18 19 20 21       15 16 17 18 19 20
21
18 19 20 21 22 23 24       22 23 24 25 26 27 28       22 23 24 25 26 27
28
25 26 27 28 29 30 31                                  29 30 31

       April                      May                       June
Mo Tu We Th Fr Sa Su       Mo Tu We Th Fr Sa Su       Mo Tu We Th Fr Sa
Su
          1  2  3  4                       1  2           1  2  3  4  5
6
 5  6  7  8  9 10 11        3  4  5  6  7  8  9        7  8  9 10 11 12
13
12 13 14 15 16 17 18       10 11 12 13 14 15 16       14 15 16 17 18 19
20
19 20 21 22 23 24 25       17 18 19 20 21 22 23       21 22 23 24 25 26
27
26 27 28 29 30             24 25 26 27 28 29 30       28 29 30
                           31

        July                     August                  September
Mo Tu We Th Fr Sa Su       Mo Tu We Th Fr Sa Su       Mo Tu We Th Fr Sa
Su
          1  2  3  4                          1              1  2  3  4
5
 5  6  7  8  9 10 11        2  3  4  5  6  7  8        6  7  8  9 10 11
12
12 13 14 15 16 17 18        9 10 11 12 13 14 15       13 14 15 16 17 18
19
19 20 21 22 23 24 25       16 17 18 19 20 21 22       20 21 22 23 24 25
26
26 27 28 29 30 31          23 24 25 26 27 28 29       27 28 29 30
                           30 31

      October                   November                  December
Mo Tu We Th Fr Sa Su       Mo Tu We Th Fr Sa Su       Mo Tu We Th Fr Sa
Su
             1  2  3        1  2  3  4  5  6  7              1  2  3  4
5
 4  5  6  7  8  9 10        8  9 10 11 12 13 14        6  7  8  9 10 11
12
11 12 13 14 15 16 17       15 16 17 18 19 20 21       13 14 15 16 17 18
19
18 19 20 21 22 23 24       22 23 24 25 26 27 28       20 21 22 23 24 25
26
25 26 27 28 29 30 31       29 30                      27 28 29 30 31
```

In [88]:

```
import datetime
current  = datetime.datetime.now()
current
```

Out[88]:

```
datetime.datetime(2021, 2, 15, 16, 59, 24, 178053)
```

In [87]:

```
current.hour, current.minute, current.second
```

Out[87]:

```
(16, 58, 43)
```

In [91]:

```
print(current)
```

```
2021-02-15 16:59:24.178053
```

In [ ]:

```

```