```
In [2]:
   1 5 * 5
Out[2]:
25
In [3]:
  1 45
Out[3]:
45
In [4]:
   1 complex(45)
Out[4]:
 (45+0j)
Input Functions
In [5]:
  1 name = input('Enter Name: ')
Enter Name: Sumit Kumar
In [6]:
   1 name
Out[6]:
 'Sumit Kumar'
In [7]:
  1 import random
  2 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__', '_acos', '_bisect', '_ceil', '_cos', '_e', '_exp', '_inst', '_itertools', '_log', '_os', '_pi', '_random', '_sha512', '_sin', '_sqrt', '_test',
  _log', '_os', '
   _test_generator', '_urandom', '_warn', 'betavariate', 'choice', 'choi
ces', 'expovariate', 'gammavariate', 'gauss', 'getrandbits', 'getstat e', 'lognormvariate', 'normalvariate', 'paretovariate', 'randint', 'random', 'randrange', 'sample', 'seed', 'setstate', 'shuffle', 'triangul
ar', 'uniform', 'vonmisesvariate', 'weibullvariate']
```

Ranges

```
In [8]:
1 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]:
1 range(0, 5)
Out[11]:
range(0, 5)
In [12]:
1 | list(range(0,5))
Out[12]:
[0, 1, 2, 3, 4]
In [14]:
1 list(range(-5,5+1))
Out[14]:
[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
```

```
In [16]:
 1 range(0, 20, 2) #range(start, stop, [interval/step])
Out[16]:
range(0, 20, 2)
In [18]:
 1 list(range(0, 20+2, 2))
Out[18]:
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
In [19]:
 1 help('random.randrange')
Help on method randrange in random:
random.randrange = randrange(start, stop=None, step=1, int=<class 'in</pre>
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]:
 1 # endpoint = True, stop = exactly same, and false then stop - 1
In [21]:
 1 random.randrange(0,5)
Out[21]:
3
In [22]:
 1 random.randrange(-5, 5)
Out[22]:
-4
In [26]:
 1 random.randrange(1234, 5678)
Out[26]:
1654
```

```
In [27]:
 1 low = 'abcdefghijklmnopqrstuvwxyz'
 2 upp = low.upper()
 3 num = '0123456789'
 4 spc = '!@#%&*'
 6 pas = low + upp + num + spc
In [28]:
 1 pas
Out[28]:
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%&*'
In [37]:
 1 # 5 digit password
 2 ''.join(random.sample(pas,3))
Out[37]:
'DZX'
In [33]:
 1 random.sample(pas, 10)
Out[33]:
['O', 'J', 'D', 'l', '%', 'N', '4', 'k', 'L', 'F']
Arithmetic Ops
In [38]:
 1 # add
 2 x, y = 5, 4
 3 x + y
Out[38]:
In [40]:
 1 # sub
 2 x - y, y - x
Out[40]:
(1, -1)
```

```
In [41]:
 1  # mul
 2 x * y, y * x
Out[41]:
(20, 20)
In [43]:
1 # div - (float value)
2 x / y, y/ x
Out[43]:
(1.25, 0.8)
In [44]:
 1 # intger division(floor division)
 2 x // y, y //x
Out[44]:
(1, 0)
In [45]:
1 # exponent
 2 x **y, y **x
Out[45]:
(625, 1024)
In [46]:
1 # remainder
2 x % y, y % x
Out[46]:
(1, 4)
In [47]:
1 100 % 3
Out[47]:
1
g-codes
In [53]:
 1 print('G -' + str(random.randint(123456, 987654)) + ' is your One Time Google ve
G -755320 is your One Time Google verification Code.
```

In [57]:

```
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]:

```
1 df_gcodes
```

Out[58]:

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

1000 rows × 10 columns

In [59]:

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

Strings

replication, concatenation

```
In [67]:
 1 'word ' * 5
Out[67]:
'word word word word '
In [68]:
 1 x = 'word'
 2 x += 'game' # x+= means x = x + whatever
In [69]:
 1 x
Out[69]:
'word game'
len Function
In [70]:
1 len('word')
Out[70]:
4
In [73]:
1 len(code)
Out[73]:
10000
Comparison OPs
In [74]:
1 #greater than
 2 3 > 5
Out[74]:
False
In [75]:
1 # smaller than
 2 3 < 5
Out[75]:
True
```

```
In [76]:
 1 # equals checing
2 3 == 5
Out[76]:
False
In [77]:
 1 # not equals
 2 3 != 5
Out[77]:
True
In [78]:
1 # greater than equal to
 2 3 >= 5
Out[78]:
False
In [79]:
1 # lesser than equal to
 2 3 <= 5
Out[79]:
True
In [80]:
 1 import calendar
 2 month = 2
 3 | year = 2021
 4 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]:
```

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

2021

		.Tar	ານລາ	٠v						Fel	riia	arv					M;	arcl	า	
Мо	Tu	We	Th	- <i>1</i> Fr	Sa	Su	ı	ol	Tu	We.	Th	Fr	Sa	Su	Мо	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	1	.5	16	17	18	19	20	21	15	16	17	18	19	20
21																				
	19	20	21	22	23	24	2	22	23	24	25	26	27	28	22	23	24	25	26	27
28																				
25	26	27	28	29	30	31									29	30	31			
	Anril							,								T	_			
Mo	April Mo Tu We Th Fr Sa Su				Cu	1	r_	ш.,	1	May	Бъ	C a	CII	Мо	ш.,	WO	June	₽ 	C a	
Su	тu	we	111	ГL	Sa	Su	Г	10	Ιu	we	111	ГL	ъa	Su	МО	тu	we	111	ГL	Sa
Su			1	2	3	4							1	2		1	2	3	4	5
6			-	_	•	-							_	_		_	_	J	-	J
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	1	0	11	12	13	14	15	16	14	15	16	17	18	19
20																				
19	20	21	22	23	24	25	1	. 7	18	19	20	21	22	23	21	22	23	24	25	26
27																				
26	27	28	29	30					25	26	27	28	29	30	28	29	30			
							3	31												
			T11] v	,						7.1	10110	- +				,	cont	- oml	oor	
Мо	Ф11		July Th		Sa	S11			Т11		ıgus Th			S11	Мо		Sept			Sa
	Tu				Sa	Su	ı													Sa
Mo Su	Tu				Sa 3															Sa 4
	Tu		Th	Fr										Su			We	Th	Fr	
Su	Tu		Th	Fr 2		4			Tu	We	Th	Fr		Su 1			We	Th 2	Fr	4
Su 5		We	Th	Fr 2	3	4		10 2	Tu 3	We	Th	Fr	Sa 7	Su 1 8	Мо	Tu	We	Th 2	Fr 3	4
Su 5 5 12 12	6	We	Th 1 8	Fr 2 9	3 10	4 11		10 2	Tu 3	We	Th	Fr	Sa	Su 1 8	Mo 6	Tu 7	We	Th 2 9	Fr 3 10	4 11
5 5 12 12 19	6	We 7 14	Th 1 8 15	Fr 2 9	3 10 17	4 11 18	P	2 9	Ти 3 10	We 4 11	Th 5	Fr 6 13	7 14	Su1815	Mo 6 13	Tu 7 14	We 1 8 15	Th 2 9 16	3 10 17	4 11 18
5 5 12 12 19	6	We 7 14	Th 1 8 15	Fr 2 9	3 10 17	4 11 18		2 9	Ти 3 10	We 4 11	Th 5	Fr 6 13	7 14	Su1815	Mo 6 13	Tu 7 14	We 1 8 15	Th 2 9 16	3 10 17	4 11 18
5 5 12 12 19 19	6 13 20	7 14 21	Th 1 8 15 22	Fr 2 9 16 23	3 10 17 24	4 11 18 25	<u></u>	2 9	3 10 17	We 4 11 18	Th 5 12 19	6 13 20	7 14 21	Su181522	6 13 20	Tu 7 14 21	We 1 8 15 22	Th 2 9 16 23	3 10 17	4 11 18
5 5 12 12 19 19	6 13 20	7 14 21	Th 1 8 15 22	Fr 2 9 16 23	3 10 17 24	4 11 18 25	-	2 9 .6	3 10 17 24	We 4 11 18	Th 5 12 19	6 13 20	7 14 21	Su181522	6 13 20	Tu 7 14 21	We 1 8 15 22	Th 2 9 16 23	3 10 17	4 11 18
5 5 12 12 19 19	6 13 20	7 14 21	Th 1 8 15 22	Fr 2 9 16 23	3 10 17 24	4 11 18 25	-	2 9 .6	3 10 17	We 4 11 18	Th 5 12 19	6 13 20	7 14 21	Su181522	6 13 20	Tu 7 14 21	We 1 8 15 22	Th 2 9 16 23	3 10 17	4 11 18
5 5 12 12 19 19 26 26	6 13 20 27	7 14 21 28	Th 1 8 15 22 29	Fr 2 9 16 23 30	3 10 17 24 31	4 11 18 25		2 9 .6 23	3 10 17 24 31	We 4 11 18 25	Th 5 12 19 26	6 13 20 27	7 14 21 28	1 8 15 22 29	Mo 6 13 20 27	Tu 7 14 21 28	We 1 8 15 22 29	Th 2 9 16 23 30	3 10 17 24	4 11 18 25
5 5 12 12 19 19 26 26	6 13 20 27	7 14 21 28	Th 1 8 15 22 29	Fr 2 9 16 23 30	3 10 17 24 31	4 11 18 25		2 9 .6 23	3 10 17 24 31	We 4 11 18 25	Th 5 12 19 26	6 13 20 27	7 14 21 28	1 8 15 22 29	Mo 6 13 20 27	Tu 7 14 21 28	We 1 8 15 22 29	Th 2 9 16 23 30	3 10 17 24	4 11 18 25
5 5 12 12 19 19 26 26	6 13 20 27	7 14 21 28	Th 1 8 15 22 29	Fr 2 9 16 23 30	3 10 17 24 31	4 11 18 25	-	2 9 .6 23	3 10 17 24 31	We 4 11 18 25	Th 5 12 19 26	6 13 20 27	7 14 21 28	1 8 15 22 29	Mo 6 13 20 27	Tu 7 14 21 28	We 1 8 15 22 29	Th 2 9 16 23 30	3 10 17 24	4 11 18 25
5 12 12 19 19 26 26	6 13 20 27	7 14 21 28	Th 1 8 15 22 29	Fr 2 9 16 23 30 er Fr	3 10 17 24 31	4 11 18 25		2 9 .6 .3 30	Tu 3 10 17 24 31	4 11 18 25 Nov	5 12 19 26 Th	6 13 20 27 Der Fr	7 14 21 28	Su18152229Su	Mo 6 13 20 27	Tu 7 14 21 28	We 1 8 15 22 29 Dec We	Th 2 9 16 23 30 ceml Th	3 10 17 24	4 11 18 25
5 12 12 19 19 26 26	6 13 20 27 Tu	7 14 21 28 Oct We	Th 1 8 15 22 29 Th	Fr 2 9 16 23 30 er Fr 1	3 10 17 24 31 Sa 2	4 11 18 25 Su 3		2 9 6 23 30	Tu 3 10 17 24 31 Tu 2	4 11 18 25 Nov We 3	Th 5 12 19 26 Teml Th 4	6 13 20 27 per fr 5	7 14 21 28 Sa 6	Su18152229Su7	Mo 6 13 20 27	Tu 7 14 21 28 Tu	We 1 8 15 22 29 Dec We 1	Th 2 9 16 23 30 Th 2	Fr 3 10 17 24 Der Fr 3	4 11 18 25 Sa 4
Su 5 5 12 12 19 19 26 26 Su 5 4	6 13 20 27 Tu	7 14 21 28 Oct We	Th 1 8 15 22 29 Th	Fr 2 9 16 23 30 er Fr 1	3 10 17 24 31 Sa 2	4 11 18 25 Su 3		2 9 6 23 30	Tu 3 10 17 24 31 Tu 2	4 11 18 25 Nov We 3	Th 5 12 19 26 Teml Th 4	6 13 20 27 per fr 5	7 14 21 28 Sa 6	Su18152229Su7	Mo 6 13 20 27	Tu 7 14 21 28 Tu	We 1 8 15 22 29 Dec We 1	Th 2 9 16 23 30 Th 2	Fr 3 10 17 24 Der Fr 3	4 11 18 25 Sa 4
Su 5 5 12 19 19 26 26 Su 5 4 12	6 13 20 27 Tu	7 14 21 28 Oct We	Th 1 8 15 22 29 Th 7	Fr 2 9 16 23 30 Fr Fr 8	3 10 17 24 31 Sa 2	4 11 18 25 Su 3		2 9 .6 23 30 10 1 8	Tu 3 10 17 24 31 Tu 2	4 11 18 25 Nov We 3 10	Th 5 12 19 26 Th 4 11	6 13 20 27 Der Fr 5 12	7 14 21 28 Sa 6	Su 1 8 15 22 29 Su 7 14	Mo 6 13 20 27	Tu 7 14 21 28 Tu 7	We 1 8 15 22 29 Dec We 1 8	Th 2 9 16 23 30 Th 2 9	Fr 3 10 17 24 Der Fr 3 10	4 11 18 25 Sa 4 11
Su 5 5 12 19 19 26 26 Mo Su 5 4 12 11	6 13 20 27 Tu	7 14 21 28 Oct We	Th 1 8 15 22 29 Th	Fr 2 9 16 23 30 Fr Fr 8	3 10 17 24 31 Sa 2	4 11 18 25 Su 3		2 9 .6 23 30 10 1 8	Tu 3 10 17 24 31 Tu 2	4 11 18 25 Nov We 3 10	Th 5 12 19 26 Th 4 11	6 13 20 27 Der Fr 5 12	7 14 21 28 Sa 6	Su 1 8 15 22 29 Su 7 14	Mo 6 13 20 27	Tu 7 14 21 28 Tu 7	We 1 8 15 22 29 Dec We 1 8	Th 2 9 16 23 30 Th 2 9	Fr 3 10 17 24 Der Fr 3 10	4 11 18 25 Sa 4 11
Su 5 12 19 19 26 26 Mo Su 5 4 12 11 19	6 13 20 27 Tu 5	7 14 21 28 Oct We 6 13	Th 1 8 15 22 29 Th 7 14	Fr 2 9 16 23 30 Fr 4 1 8 15	3 10 17 24 31 Sa 2 9	4 11 18 25 Su 3 10		2 9 .6 23 30 IO 1 8 .5	Tu 3 10 17 24 31 Tu 2 9 16	We 4 11 18 25 Nov We 3 10 17	Th 5 12 19 26 7emh Th 4 11	6 13 20 27 5 12	7 14 21 28 Sa 6 13	Su 1 8 15 22 29 Su 7 14 21	Mo 6 13 20 27 Mo 6 13	Tu 7 14 21 28 Tu 7	We 1 8 15 22 29 Dec We 1 8 15	Th 2 9 16 23 30 Th 2 9 16	Fr 3 10 17 24 24 10 17	4 11 18 25 Sa 4 11
Su 5 5 12 19 19 26 26 Su 5 4 12 11 19 18	6 13 20 27 Tu 5	7 14 21 28 Oct We 6 13	Th 1 8 15 22 29 Th 7 14	Fr 2 9 16 23 30 Fr 4 1 8 15	3 10 17 24 31 Sa 2 9	4 11 18 25 Su 3 10		2 9 .6 23 30 IO 1 8 .5	Tu 3 10 17 24 31 Tu 2 9 16	4 11 18 25 Nov We 3 10	Th 5 12 19 26 7emh Th 4 11	6 13 20 27 5 12	7 14 21 28 Sa 6 13	Su 1 8 15 22 29 Su 7 14 21	Mo 6 13 20 27 Mo 6 13	Tu 7 14 21 28 Tu 7	We 1 8 15 22 29 Dec We 1 8 15	Th 2 9 16 23 30 Th 2 9 16	Fr 3 10 17 24 24 10 17	4 11 18 25 Sa 4 11
Su 5 5 12 19 19 26 26 Su 5 4 12 11 19 18 26	6 13 20 27 Tu 5 12	7 14 21 28 Oct We 6 13 20	Th 1 8 15 22 29 Th 7 14 21	Fr 2 9 16 23 30 er Fr 1 8 15 22	3 10 17 24 31 Sa 2 9 16 23	4 11 18 25 Su 3 10 17 24		2 9 .6 23 30 10 1 8 .5 22	Tu 3 10 17 24 31 Tu 2 9 16 23	4 11 18 25 Nov We 3 10	Th 5 12 19 26 7emh Th 4 11	6 13 20 27 5 12	7 14 21 28 Sa 6 13	Su 1 8 15 22 29 Su 7 14 21	Mo 6 13 20 27 Mo 6 13 20	Tu 7 14 21 28 Tu 7 14 21	We 1 8 15 22 29 Dec We 1 8 15	Th 2 9 16 23 16 23	Fr 3 10 17 24 17 24	4 11 18 25 Sa 4 11

```
In [88]:

1   import datetime
2   current = datetime.datetime.now()
3   current

Out[88]:

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

In [87]:
1   current.hour, current.minute, current.second

Out[87]:
(16, 58, 43)

In [91]:
1   print(current)

2021-02-15 16:59:24.178053

In []:
1
```