MDM_AI-ML_Python_Lec1

December 5, 2024

1 Introduction to Python

1.1 By Ajit Kumar

Date: Dec. 05, 2024

[1]:
```
print('Welcome to ICT Mumbai.')
```

```
Welcome to ICT Mumbai.
```

[2]:
```
3753465+8234261
```

[2]: 11987726

[3]:
```
36136*2534125
```

[3]: 91573141000

[4]:
```
57/12
```

[4]: 4.75

[6]:
```
57//12 # Returns quotient
```

[6]: 4

[8]:
```
57 % 12 # Returns remainder
```

[8]: 9

[10]:
```
6735**51 ## Raising power
```

[10]: 175880464131092932523290562993700513371081919080287265369751223913963172917345972223288032934897081660342019176629835776510448089765740358214072033655428868099823702664252778049558401107788085937 5

Defining Variables

[19]:
```
a = 63435
b = 76353.0
```

```
c = 6343
```

[16]:
```
(a+b)*c
```

[16]: 886675284

[17]:
```
my_name = 'Ajit Kumar'
affiliation = 'ICT Mumbai'
```

[21]:
```
type(a), type(b),type(my_name)
```

[21]: (int, float, str)

[23]:
```
P = 50000
r = 5.5 # Annual interest rate
t = 4 # No, of years
n = 4  # No of times per year interest is calculated
A = P*(1+r/(n*100))**(n*t)
print(A)
```

```
62210.526924725506
```

[28]:
```
x = float(input('Enter the principal amount'))
```

```
Enter the principal amount50000
```

[29]:
```
x
type(x)
```

[29]: float

[32]:
```
P = float(input('Enter the principal amount:'))
r = float(input('Enter the annual interest rate:'))
t = int(input('Enter the number of years:'))
n = int(input('Enter the number of times interest is calulated per year:'))
A = P*(1+r/(n*100))**(n*t)
print('The total return is ',A)
```

```
Enter the principal amount:50000
Enter the annual interest rate:4
Enter the number of years:5
Enter the number of times interest is calulated per year:12
The total return is  61049.82969710607
```

[34]:
```
print(f'The total return on investment of Rs.{P} is {A}')
```

```
The total return on investment of Rs.50000.0 is 61049.82969710607
```

[39]:
```
my_name+', '+affiliation
```

[39]:
```
'Ajit Kumar, ICT Mumbai'
```

[40]:
```
sin(1.5)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[40], line 1
----> 1 sin(1.5)

NameError: name 'sin' is not defined
```

[41]:
```
import math
```

[43]:
```
# help(math)
```

[44]:
```
dir(math)
```

[44]:
```
['__doc__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'acos',
 'acosh',
 'asin',
 'asinh',
 'atan',
 'atan2',
 'atanh',
 'cbrt',
 'ceil',
 'comb',
 'copysign',
 'cos',
 'cosh',
 'degrees',
 'dist',
 'e',
 'erf',
 'erfc',
 'exp',
 'exp2',
 'expm1',
 'fabs',
 'factorial',
 'floor',
```

```
 'fmod',
 'frexp',
 'fsum',
 'gamma',
 'gcd',
 'hypot',
 'inf',
 'isclose',
 'isfinite',
 'isinf',
 'isnan',
 'isqrt',
 'lcm',
 'ldexp',
 'lgamma',
 'log',
 'log10',
 'log1p',
 'log2',
 'modf',
 'nan',
 'nextafter',
 'perm',
 'pi',
 'pow',
 'prod',
 'radians',
 'remainder',
 'sin',
 'sinh',
 'sqrt',
 'tan',
 'tanh',
 'tau',
 'trunc',
 'ulp']
```

[45]:
```
math.sin(2.5)
```

[45]:
```
0.5984721441039564
```

[48]:
```
math.radians?
```

[49]:
```
math.sqrt(2)
```

[49]:
```
1.4142135623730951
```

[50]:
```
from math import sin, cos,sqrt, pi
```

[51]:
```
sqrt(3)
```

[51]:
```
1.7320508075688772
```

[54]:
```
math.remainder(23,7)
```

[54]:
```
2.0
```

[55]:
```
from math import remainder as rm
```

[56]:
```
rm(23,7)
```

[56]:
```
2.0
```

[57]:
```
from math import *
```

[58]:
```
gcd(72,32)
```

[58]:
```
8
```

[59]:
```
sqrt(-1)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[59], line 1
----> 1 sqrt(-1)

ValueError: math domain error
```

[60]:
```
import cmath
cmath.sqrt(-1)
```

[60]:
```
1j
```

[62]:
```
z = 3+5J
type(z)
```

[62]:
```
complex
```

[69]:
```
a, b, c = 2, -4, -2
disc = b**2-4*a*c
x1 = (-b+cmath.sqrt(disc))/(2*a)
x2 = (-b-cmath.sqrt(disc))/(2*a)
print(f'The roots are {x1} and {x2}')
```

```
The roots are (2.414213562373095+0j) and (-0.41421356237309515+0j)
```

[70]:
```
cmath.sin(2+3J)
```

[70]:
```
(9.15449914691143-4.168906959966565j)
```

[71]:
```
dir(math)
```

[71]:
```
['__doc__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'acos',
 'acosh',
 'asin',
 'asinh',
 'atan',
 'atan2',
 'atanh',
 'cbrt',
 'ceil',
 'comb',
 'copysign',
 'cos',
 'cosh',
 'degrees',
 'dist',
 'e',
 'erf',
 'erfc',
 'exp',
 'exp2',
 'expm1',
 'fabs',
 'factorial',
 'floor',
 'fmod',
 'frexp',
 'fsum',
 'gamma',
 'gcd',
 'hypot',
 'inf',
 'isclose',
 'isfinite',
 'isinf',
 'isnan',
 'isqrt',
```

```
 'lcm',
 'ldexp',
 'lgamma',
 'log',
 'log10',
 'log1p',
 'log2',
 'modf',
 'nan',
 'nextafter',
 'perm',
 'pi',
 'pow',
 'prod',
 'radians',
 'remainder',
 'sin',
 'sinh',
 'sqrt',
 'tan',
 'tanh',
 'tau',
 'trunc',
 'ulp']
```

1.2 Working with lists

[73]:
```
L =[23,89,12,94,100,735,425,28,53,279]
type(L)
```

[73]:
```
list
```

[74]:
```
L.pop?
```

[75]:
```
L.pop()
```

[75]:
```
279
```

[76]:
```
L
```

[76]:
```
[23, 89, 12, 94, 100, 735, 425, 28, 53]
```

[77]:
```
L[0]
```

[77]:
```
23
```

[78]:
```
L[7]
```

[78]:
```
28
```

[79]:
```
L[2:5]
```

[79]:
```
[12, 94, 100]
```

[80]:
```
L.append([1,2,3])
```

[81]:
```
L
```

[81]:
```
[23, 89, 12, 94, 100, 735, 425, 28, 53, [1, 2, 3]]
```

[82]:
```
L.pop()
```

[82]:
```
[1, 2, 3]
```

[83]:
```
L
```

[83]:
```
[23, 89, 12, 94, 100, 735, 425, 28, 53]
```

[84]:
```
L.extend([1,2,3])
```

[90]:
```
L
```

[90]:
```
[23, 89, 12, 94, 100, 735, 425, 28, 53, 1, 2, 3]
```

[91]:
```
L[2]=21
L
```

[91]:
```
[23, 89, 21, 94, 100, 735, 425, 28, 53, 1, 2, 3]
```

1.3 Use of Tuples

[86]:
```
T = (1,4,7,3,10,52,13,24)
```

[87]:
```
type(T)
```

[87]:
```
tuple
```

[88]:
```
T[4]
```

[88]:
```
10
```

[89]:
```
len(T)
```

[89]:
```
8
```

[92]:
```
T[2]=9
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[92], line 1
----> 1 T[2]=9

TypeError: 'tuple' object does not support item assignment
```

[95]:
```
A = {1,2,3,4}
B = {2,4,6,8}
```

[96]:
```
A.intersection(B)
```

[96]:
```
{2, 4}
```

1.4 Dealing with dictionary data type

[97]:
```
d = {}
```

[98]:
```
type(d)
```

[98]:
```
dict
```

[99]:
```
d['Name'] = 'Mr. XYZ'
```

[100]:
```
d
```

[100]:
```
{'Name': 'Mr. XYZ'}
```

[101]:
```
d['DOB'] = 'Januay 1, 2000'
```

[102]:
```
d
```

[102]:
```
{'Name': 'Mr. XYZ', 'DOB': 'Januay 1, 2000'}
```

[103]:
```
d= {'Name': 'Mr. XYZ', 'DOB': 'Januay 1, 2000','Roll':'23CHD62526'}
```

[104]:
```
d.keys()
```

[104]:
```
dict_keys(['Name', 'DOB', 'Roll'])
```

[106]:
```
d.values()
```

[106]:
```
dict_values(['Mr. XYZ', 'Januay 1, 2000', '23CHD62526'])
```

[107]:
```
d['DOB']
```

[107]:
```
'Januay 1, 2000'
```

[108]:
```
d['marks']={'Math':79,'Phy':87,'Chem':97,'Eng':76}
```

[109]:
```
d
```

[109]:
```
{'Name': 'Mr. XYZ',
 'DOB': 'Janaay 1, 2000',
 'Roll': '23CHD62526',
 'marks': {'Math': 79, 'Phy': 87, 'Chem': 97, 'Eng': 76}}
```

[110]:
```
d['marks']['Phy']
```

[110]:
```
87
```

[111]:
```
d.items()
```

[111]:
```
dict_items([('Name', 'Mr. XYZ'), ('DOB', 'Janaay 1, 2000'), ('Roll',
'23CHD62526'), ('marks', {'Math': 79, 'Phy': 87, 'Chem': 97, 'Eng': 76})])
```

1.5 User defined functions

[112]:
```
def Return_CI(P,r,n,t):
    A = P*(1+r/(n*100))**(n*t)
    return A
```

[113]:
```
Return_CI(20000,3.5,4,10)
```

[113]:
```
28338.176758622725
```

[114]:
```
from math import sqrt
def Heron(a,b,c):
    s =(a+b+c)/2
    A = sqrt(s*(s-a)*(s-b)*(s-c))
    return A
```

[117]:
```
def Return_CI(P,r,t,n=1):
    A = P*(1+r/(n*100))**(n*t)
    return A
```

[119]:
```
Return_CI(20000,3.5,10,4)
```

[119]:
```
28338.176758622725
```

[120]:
```
def Return_CI(P,r,n=1,t):
    A = P*(1+r/(n*100))**(n*t)
    return A
```

```
  Cell In[120], line 1
    def Return_CI(P,r,n=1,t):
```

```
SyntaxError: non-default argument follows default argument
```

[ ]: