In [3]:

```
from math import *
d = 10.0 # diameter
A = pi * d**2 / 4
print("diameter =", d)
print("area = ", A)
```

```
diameter = 10.0
area =  78.53981633974483
```

This is the program which displays the diameter and area of the circle by importing math module.

In [6]:

```
s = input("What is your name?")
print("HELLO ", s)
```

```
What is your name?Vasanthi
HELLO  Vasanthi
```

This is the simple program asks “what is your name” and after entering the name it prints as Hello “name”.

In [7]:

```
d = 10.0
```

In [10]:

```
x = int(input("Input an integer: "))
y = float(input("Input a float: "))
print(x, y)
```

```
Input an integer: 5
Input a float: 2.3
5 2.3
```

This is the program asks to input the integer and float values, gives the ouput as (integer, float).

In [1]:

```
from math import *
i = 0
while i<= 100:
    print(i, "\t\t" , sqrt(i))
    i = i + 1
print("READY!")
```

```
0               0.0
1               1.0
2               1.4142135623730951
3               1.7320508075688772
4               2.0
5               2.23606797749979
6               2.449489742783178
7               2.6457513110645907
8               2.8284271247461903
9               3.0
10              3.1622776601683795
11              3.3166247903554
12              3.4641016151377544
13              3.605551275463989
14              3.7416573867739413
15              3.872983346207417
16              4.0
17              4.123105625617661
18              4.242640687119285
19              4.358898943540674
20              4.47213595499958
21              4.58257569495584
22              4.69041575982343
23              4.795831523312719
24              4.898979485566356
25              5.0
26              5.0990195135927845
27              5.196152422706632
28              5.291502622129181
29              5.385164807134504
30              5.477225575051661
31              5.5677643628300215
32              5.656854249492381
33              5.744562646538029
34              5.830951894845301
35              5.916079783099616
36              6.0
37              6.082762530298219
38              6.164414002968976
39              6.244997998398398
40              6.324555320336759
41              6.4031242374328485
42              6.48074069840786
43              6.557438524302
44              6.6332495807108
45              6.708203932499369
46              6.782329983125268
47              6.855654600401044
48              6.928203230275509
49              7.0
50              7.0710678118654755
51              7.14142842854285
52              7.211102550927978
53              7.280109889280518
54              7.3484692283495345
55              7.416198487095663
56              7.483314773547883
57              7.54983443527075
58              7.615773105863909
59              7.681145747868608
60              7.745966692414834
61              7.810249675906654
62              7.874007874011811
63              7.937253933193772
64              8.0
65              8.06225774829855
66              8.12403840463596
67              8.18535277187245
68              8.246211251235321
69              8.306623862918075
70              8.366600265340756
71              8.426149773176359
72              8.48528137423857
73              8.54400374531753
74              8.602325267042627
75              8.660254037844387
76              8.717797887081348
77              8.774964387392123
78              8.831760866327848
79              8.888194417315589
80              8.94427190999916
81              9.0
82              9.055385138137417
83              9.1104335791443
84              9.16515138991168
85              9.219544457292887
86              9.273618495495704
87              9.327379053088816
88              9.38083151964686
89              9.433981132056603
90              9.486832980505138
91              9.539392014169456
92              9.591663046625438
93              9.643650760992955
94              9.695359714832659
95              9.746794344808963
96              9.797958971132712
97              9.848857801796104
98              9.899494936611665
99              9.9498743710662
100             10.0
READY!
```

Here we used while loop.This program prints the all the 100 numbers square roots .

In [3]:

```
s = input ("Input your name: ")
if s == "Tom":
    print("HELLO ", s)
```

```
Input your name: Tom
HELLO  Tom
```

Here We already given name So we have to enter the name given in the code and print Hello tom.

In [4]:

```
s = input ("Input your name: ")
if s == "Tom":
    print("Hello ", s)
else:
    print("Hello unknown")
```

```
Input your name: Vasanthi
Hello unknown
```

In this program,we used if else condtional statements used. We give the input name as given in code then prints hello Tom if not it prints Hello unknown. That means else block is exexuted.vales given in the code.

In [5]:

```
s = input ("Input your name: ")
if s == "Tom":
    print("Hello ", s)
elif s == "Carmen":
    print("I'm so glad to see you ", s)
elif s == "Sonia":
    print("I didn't expect you ",s)
else:
    print("Hello unknown")
```

```
Input your name: Carmen
I'm so glad to see you  Carmen
```

In this program we used if elif else conditional statements. If we give name as Tom it prints Hello tom, like that its prints the respective statements given in the code . If we didnt give the name as mentioned in the code it gives ouput as Hello unknown.

In [7]:

```
(x,y) = (5, 3)
coordinates = (x,y)
print(coordinates)
dimensions = (8, 5.0, 3.14)
print(dimensions)
print(dimensions[0])
print(dimensions[1])
print(dimensions[2])
```

```
(5, 3)
(8, 5.0, 3.14)
8
5.0
3.14
```

In this program we print coordinates give in the code. and print the dimensions values as given and print the 0th position value, 1st potision value and 2nd position vales respectively. That means if we give the index number it will print the value at that position.

In [4]:

```
a=[0,1,2]
print(a)
a.append(5)
a.append( "Zapzoo")
print(a)
```

```
[0, 1, 2]
[0, 1, 2, 5, 'Zapzoo']
```

In this program we used append which is a pre-defined method used to add a single item to certain collection types. It will print values in a and used append 5 and Zapzoo so it will print all values in a . That means 0,1,2,5, 'Zapzoo'

In [6]:

```
a=[0,1,2]
print(len(a))
```

```
3
```

In [ ]:

```
In this program it will print the length of the list. Here we have 3 numbers in the list so it will print 3.
```

In [7]:

```
mylist = ["black", "red", "orange"]
print(mylist[0])
print(mylist[1])
print(mylist[2])
```

```
black
red
orange
```

In [ ]:

```
In this program the list of colors are given and prints the colors according to their index numbers respectively.
```

In [9]:

```
r1 = range(11) # 0...10
print(r1) # [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
r2 = range(5,16) # 5...15
print(r2) # [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
r3 = range(4,21,2) # 4...20 step 2
print(r3) # [4, 6, 8, 10, 12, 14, 16, 18, 20]
r4 = range(15, 4, -5) # 15....5 step -5
print(r4)
```

```
range(0, 11)
range(5, 16)
range(4, 21, 2)
range(15, 4, -5)
```

In [ ]:

In [31]:

```
pwd
```

Out[31]:

```
'C:\\Users\\SAI VASANTHI'
```

In [2]:

```
import numpy as np
r5 = np.linspace(0,2,9)
print(r5)
```

```
[0.   0.25 0.5  0.75 1.   1.25 1.5  1.75 2.  ]
```

In [ ]:

```
In this program linspace function is used, which create an evenly spaced sequence in a specified interval.
In the code it is given 0 and 2 numbers and given 9 , because it will print 9 floating numbers between 0 and 2.
```

In [8]:

```
np.random.randint(1,50,size = (1,5))
```

Out[8]:

```
array([[15,  8, 30, 49, 43]])
```

In [10]:

```
r6 = np.logspace(2, 3, 9)
print(r6)
```

```
[ 100.          133.35214322  177.827941    237.13737057  316.22776602
  421.69650343  562.34132519  749.89420933 1000.        ]
```

In [4]:

In [ ]:

```
In this program logspace function is used, which return numbers which are evenly spaced on the log scale.
```

In [13]:

```
mynames = [ "Sam", "Pit", "Misch" ]
for n in mynames:
    print("HELLO ", n)
```

```
HELLO  Sam
HELLO  Pit
HELLO  Misch
```

In [ ]:

```
In this program list of names are given and print Hello followed by names.
```

In [14]:

```
from math import *
for i in range (0, 5):
    print(i, "\t", sqrt(i))
```

```
0        0.0
1        1.0
2        1.4142135623730951
3        1.7320508075688772
4        2.0
```

In [ ]:

```
In this program we use math module and given range 0 to 5. It prints square root values of that numbers.
```

In [15]:

```
colours = [ "black", "brown", "red", "orange", "yellow",
"green", "blue", "violet", "grey","white" ]
cv = list (enumerate (colours))
for c in cv:
    print(c[0], "\t", c[1])
```

```
0        black
1        brown
2        red
3        orange
4        yellow
5        green
6        blue
7        violet
8        grey
9        white
```

In [ ]:

```
In this program the list enumurator function is used to give back a list of tuples,
that contain each an index and the color value as text.
```

In [21]:

```
def area(b, h):
#calculate area of a rectangle"""
    A = b * h
    return A
def perimeter(b, h):
#calulates perimeter of a rectangle"""
    P = 2 * (b+h)
    return P
# main program using defined functions
width = 5
height = 3
print("Area = ", area(width, height))
print("Perimeter = ", perimeter(width, height))
```

```
Area =  15
Perimeter =  16
```

In [ ]:

```
In this program we find the area and perimeter of a rectangle by using two functions area and perimeter.
These functions are called in the main function.
```

In [23]:

```
def area_and_perimeter (b, h):
    A = b * h
    P = 2 * (b+h)
    return A, P
# main program using defined function
ar, per = area_and_perimeter ( 4, 3)
print(ar)
print(per)
```

```
12
14
```

In [ ]:

```
In this program area and perimeter are calculated using using a single function
A function can return more than one value, here the return values are returned as a tuple.
```

In [25]:

```
def greeting():
    print("HELLO")
# main program using defined functions
greeting()
```

```
HELLO
```

In [ ]:

```
In this program we will define greetings and print Hello.
```

In [26]:

```
import numpy as np
# calculate 100 values for x and y without a for loop
x = np.linspace(0, 2* np.pi, 100)
y = np.sin(x)
print(x)
print(y)
```

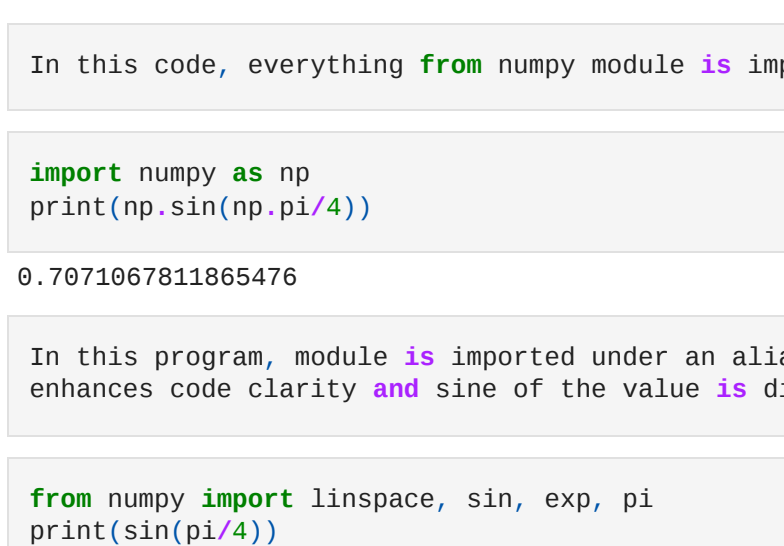
```
[0.         0.06346652 0.12693304 0.19039955 0.25386607 0.31733259
 0.38079911 0.44426563 0.50773215 0.57119866 0.63466518 0.6981317
 0.76159822 0.82506474 0.88853126 0.95199777 1.01546429 1.07893081
 1.14239733 1.20586385 1.26933037 1.33279688 1.3962634  1.45972992
 1.52319644 1.58666296 1.65012947 1.71359599 1.77706251 1.84052903
 1.90399555 1.96746207 2.03092858 2.0943951  2.15786162 2.22132814
 2.28479466 2.34826118 2.41172769 2.47519421 2.53866073 2.60212725
 2.66559377 2.72906028 2.7925268  2.85599332 2.91945984 2.98292636
 3.04639288 3.10985939 3.17332591 3.23679243 3.30025895 3.36372547
 3.42719199 3.4906585  3.55412502 3.61759154 3.68105806 3.74452458
 3.8079911  3.87145761 3.93492413 3.99839065 4.06185717 4.12532369
 4.1887902  4.25225672 4.31572324 4.37918976 4.44265628 4.5061228
 4.56958931 4.63305583 4.69652235 4.75998887 4.82345539 4.88692191
 4.95038842 5.01385494 5.07732146 5.14078798 5.2042545  5.26772102
 5.33118753 5.39465405 5.45812057 5.52158709 5.58505361 5.64852012
 5.71198664 5.77545316 5.83891968 5.9023862  5.96585272 6.02931923
 6.09278575 6.15625227 6.21971879 6.28318531]
[ 0.00000000e+00  6.34239197e-02  1.26592454e-01  1.89251244e-01
  2.51147987e-01  3.12033446e-01  3.71662456e-01  4.29794912e-01
  4.86196736e-01  5.40640817e-01  5.92907929e-01  6.42787610e-01
  6.90079011e-01  7.34591709e-01  7.76146464e-01  8.14575952e-01
  8.49725430e-01  8.81453363e-01  9.09631995e-01  9.34147860e-01
  9.54902241e-01  9.71811568e-01  9.84807753e-01  9.93838464e-01
  9.98867339e-01  9.99874128e-01  9.96854776e-01  9.89821442e-01
  9.78802446e-01  9.63842159e-01  9.45000819e-01  9.22354294e-01
  8.95993774e-01  8.66025404e-01  8.32569855e-01  7.95761841e-01
  7.55749574e-01  7.12694171e-01  6.66769001e-01  6.18158986e-01
  5.67059864e-01  5.13677392e-01  4.58226522e-01  4.00930535e-01
  3.42020143e-01  2.81732557e-01  2.20310533e-01  1.58001396e-01
  9.50560433e-02  3.17279335e-02 -3.17279335e-02 -9.50560433e-02
 -1.58001396e-01 -2.20310533e-01 -2.81732557e-01 -3.42020143e-01
 -4.00930535e-01 -4.58226522e-01 -5.13677392e-01 -5.67059864e-01
 -6.18158986e-01 -6.66769001e-01 -7.12694171e-01 -7.55749574e-01
 -7.95761841e-01 -8.32569855e-01 -8.66025404e-01 -8.95993774e-01
 -9.22354294e-01 -9.45000819e-01 -9.63842159e-01 -9.78802446e-01
 -9.89821442e-01 -9.96854776e-01 -9.99874128e-01 -9.98867339e-01
 -9.93838464e-01 -9.84807753e-01 -9.71811568e-01 -9.54902241e-01
 -9.34147860e-01 -9.09631995e-01 -8.81453363e-01 -8.49725430e-01
 -8.14575952e-01 -7.76146464e-01 -7.34591709e-01 -6.90079011e-01
 -6.42787610e-01 -5.92907929e-01 -5.40640817e-01 -4.86196736e-01
 -4.29794912e-01 -3.71662456e-01 -3.12033446e-01 -2.51147987e-01
 -1.89251244e-01 -1.26592454e-01 -6.34239197e-02 -2.44929360e-16]
```

In [ ]:

```
This program return the values of sine function ranging between 0 and 100 using numpy module and functions such as linspace,sin.
```

In [27]:

```
from numpy import linspace, sin, exp, pi
import matplotlib.pyplot as mp
# calculate 500 values for x and y without a for loop
x = linspace(0, 10*pi, 500)
y = sin(x) * exp(-x/10)
# make diagram
mp.plot(x,y)
mp.show()
```



In [ ]:

```
In this program, the calculated function values are diaplayed in a diagram
using Matplotlib which is the standard python plotting library.
```

In [28]:

```
from numpy import *
print(sin(pi/4))
```

```
0.7071067811865476
```

In [ ]:

```
In this code, everything from numpy module is imported and sine of the value is printed.
```

In [29]:

```
import numpy as np
print(np.sin(np.pi/4))
```

```
0.7071067811865476
```

In [ ]:

```
In this program, module is imported under an alias name np, which
enhances code clarity and sine of the value is displayed.
```

In [30]:

```
from numpy import linspace, sin, exp, pi
print(sin(pi/4))
```

```
0.7071067811865476
```

In [ ]:

```
Here functions in numpy module that are needed are only imported like
linspace,sin,exp and sine of the value is printed
```