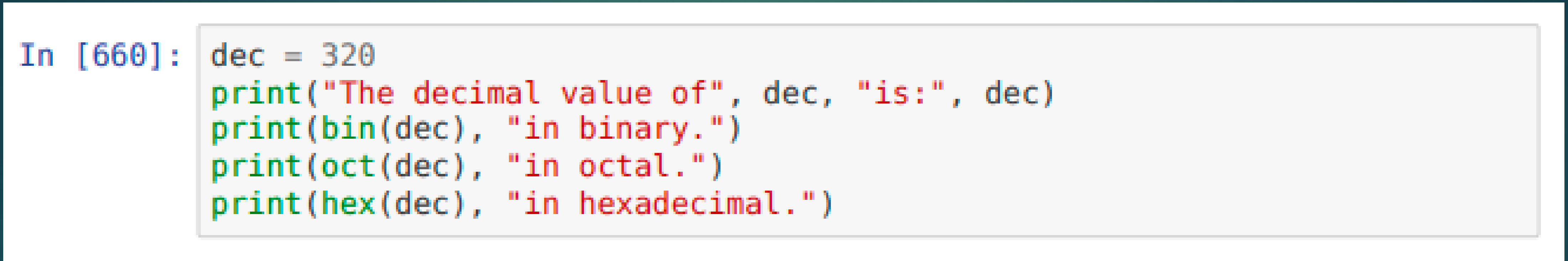
Python for Physics
Lab -3

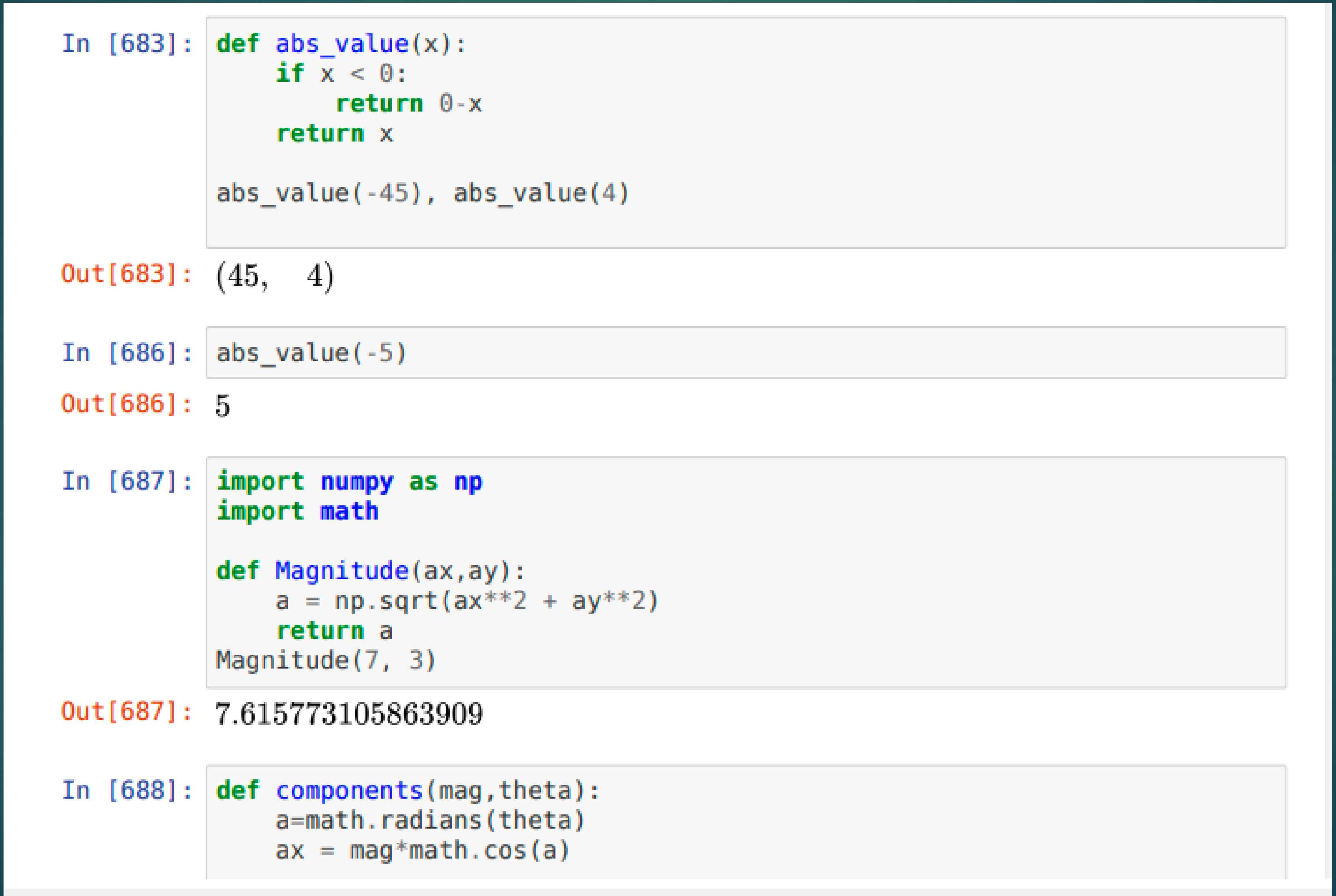
Binary



In [660]:

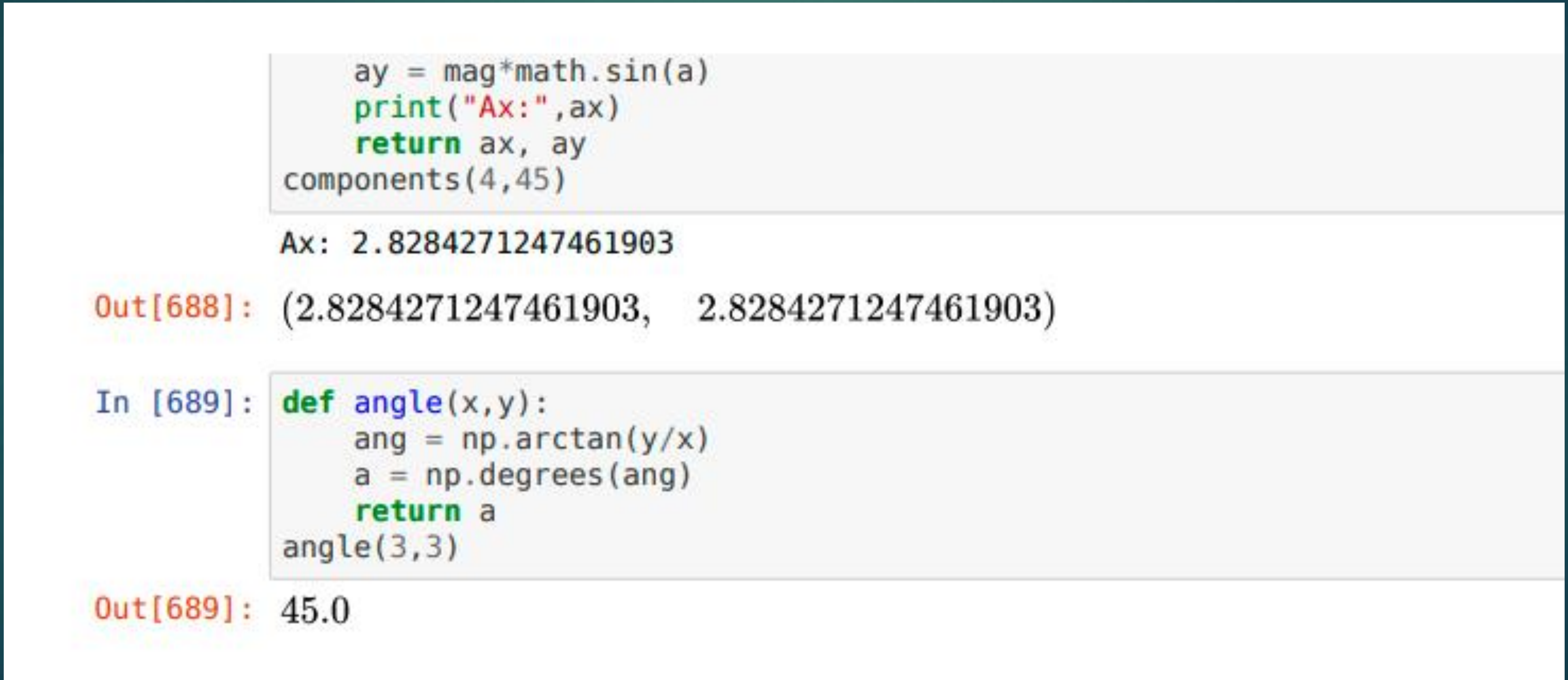
```
dec = 320
print("The decimal value of", dec, "is:", dec)
print(bin(dec), "in binary.")
print(oct(dec), "in octal.")
print(hex(dec), "in hexadecimal.")
```

Some Useful Examples:



```
In [683]: def abs_value(x):
              if x < 0:
                  return 0-x
              return x

          abs_value(-45), abs_value(4)

Out[683]: (45,   4)

In [686]: abs_value(-5)

Out[686]: 5

In [687]: import numpy as np
          import math

          def Magnitude(ax,ay):
              a = np.sqrt(ax**2 + ay**2)
              return a
          Magnitude(7, 3)

Out[687]: 7.615773105863909

In [688]: def components(mag,theta):
              a=math.radians(theta)
              ax = mag*math.cos(a)
```

(Continued)



```
    ay = mag*math.sin(a)
    print("Ax:",ax)
    return ax, ay
components(4,45)
```

```
Ax: 2.8284271247461903
```

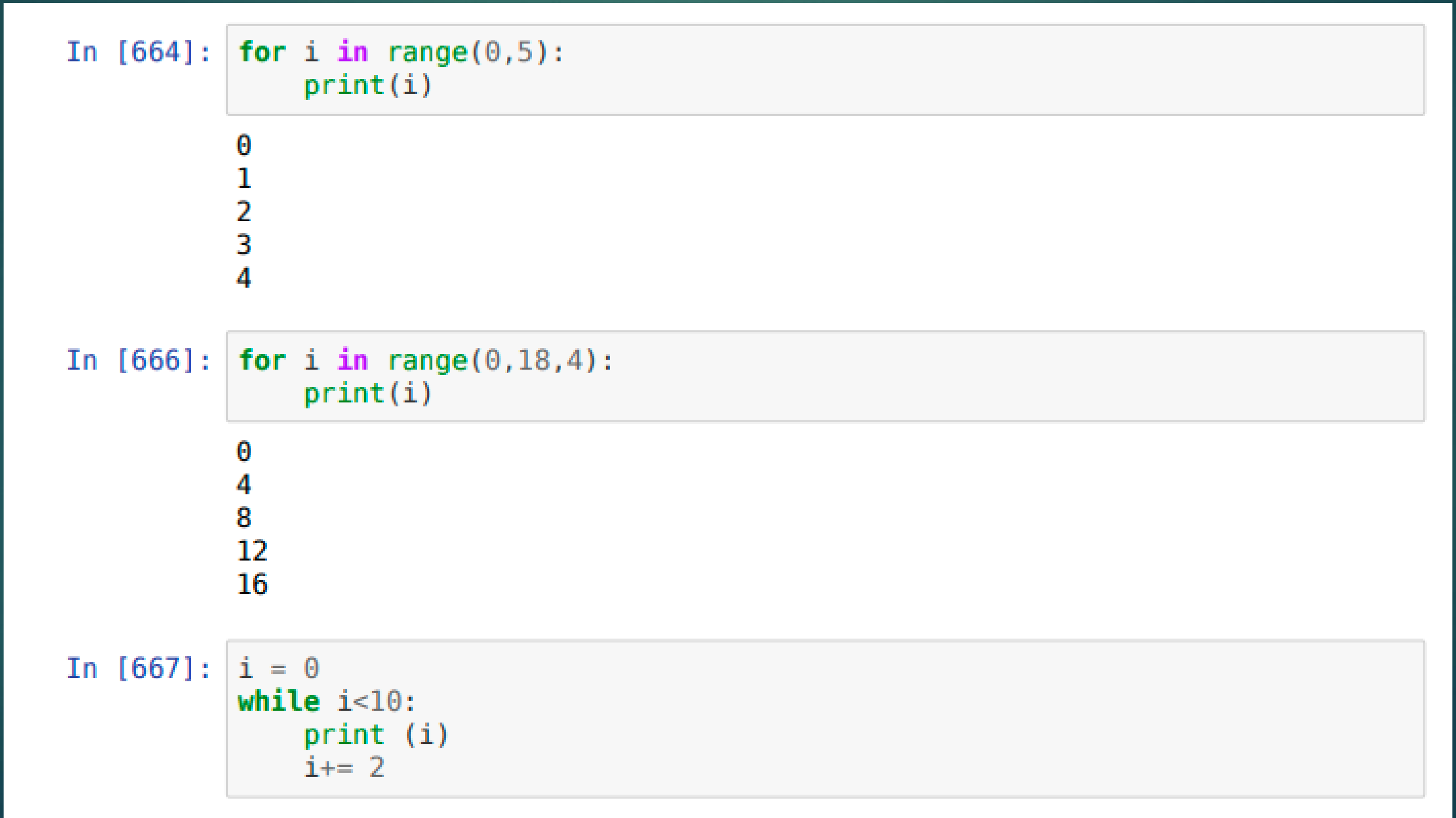
Out[688]: (2.8284271247461903,   2.8284271247461903)

In [689]:

```
def angle(x,y):
    ang = np.arctan(y/x)
    a = np.degrees(ang)
    return a
angle(3,3)
```

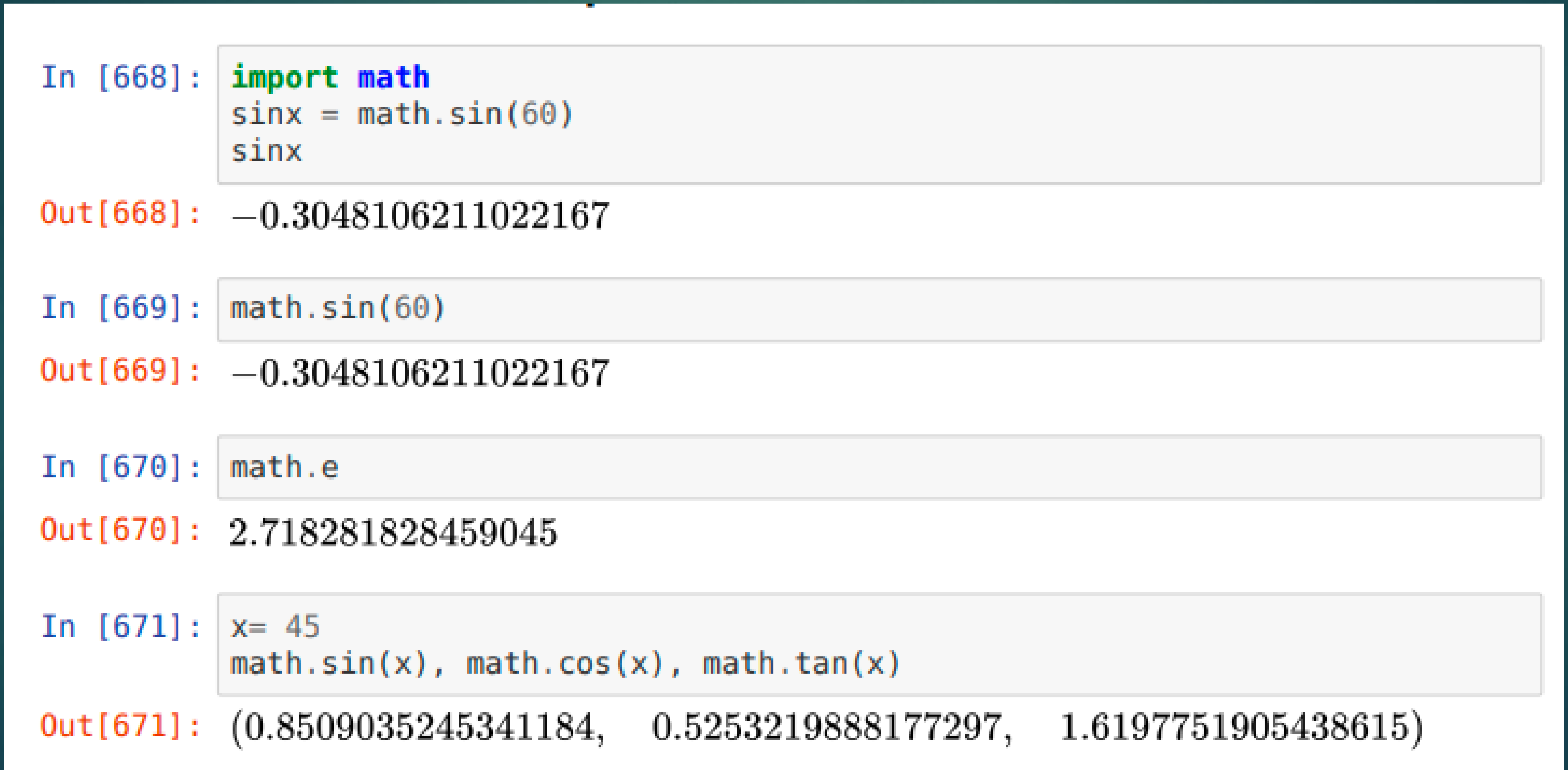
Out[689]: 45.0

Loops
range (start, stop,[,step])

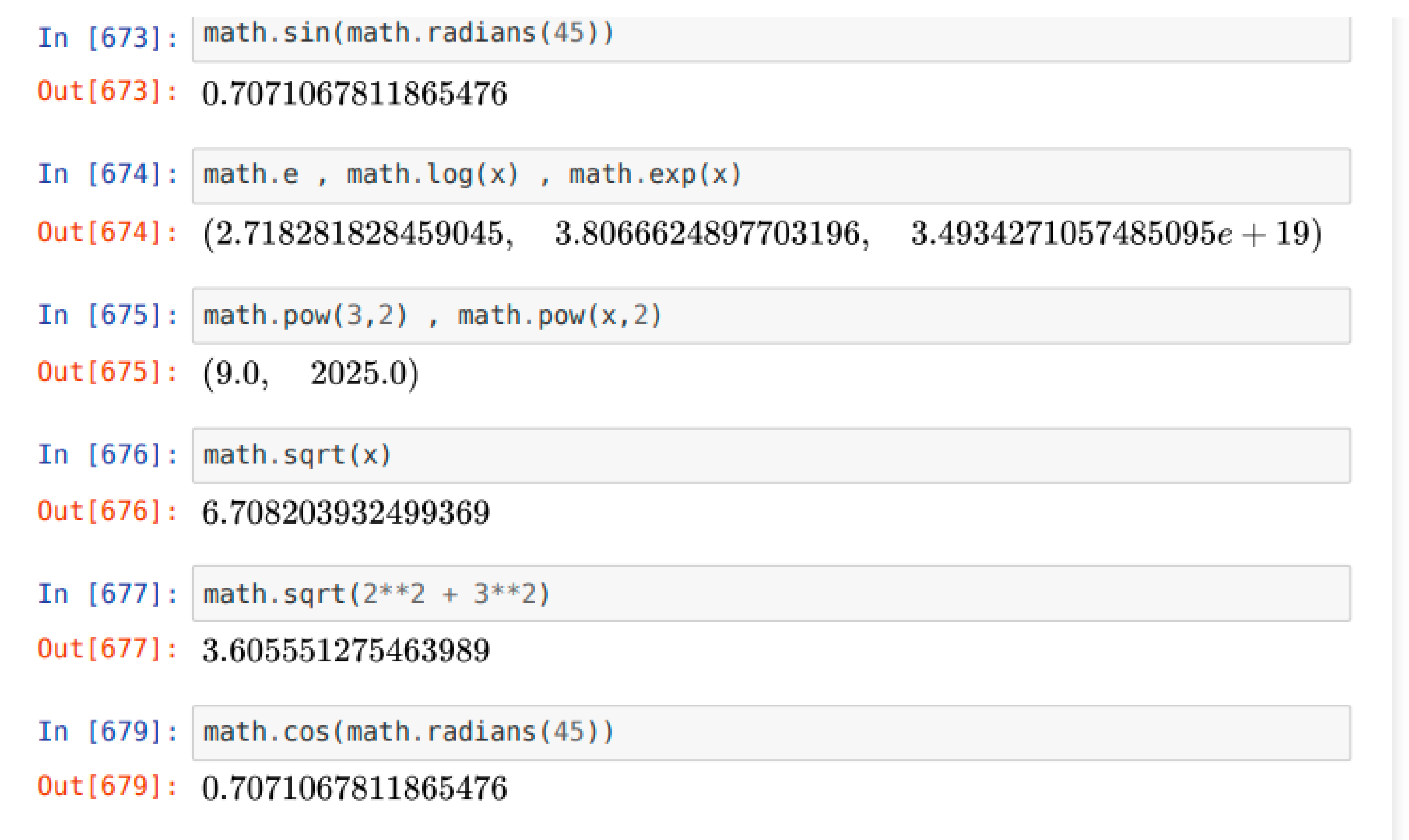


```
In [664]: for i in range(0,5):
              print(i)

          0
          1
          2
          3
          4

In [666]: for i in range(0,18,4):
              print(i)

          0
          4
          8
          12
          16

In [667]: i = 0
          while i<10:
              print (i)
              i+= 2
```

Importing Module



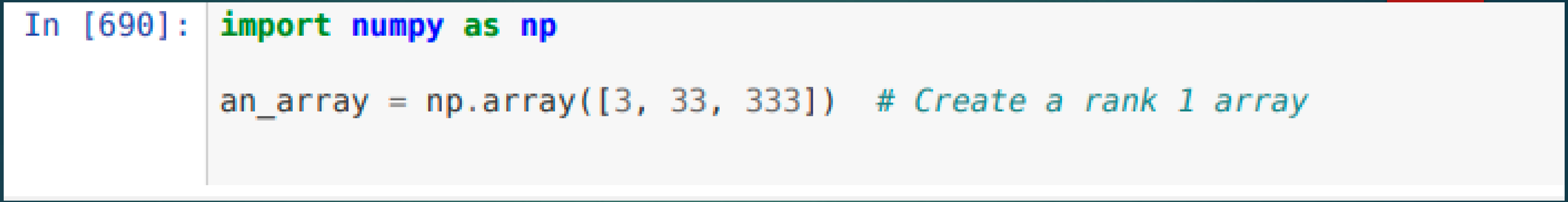
```
In [668]: import math
          sinx = math.sin(60)
          sinx

Out[668]: −0.3048106211022167

In [669]: math.sin(60)

Out[669]: −0.3048106211022167

In [670]: math.e

Out[670]: 2.718281828459045

In [671]: x= 45
          math.sin(x), math.cos(x), math.tan(x)

Out[671]: (0.8509035245341184,   0.5253219888177297,   1.6197751905438615)
```



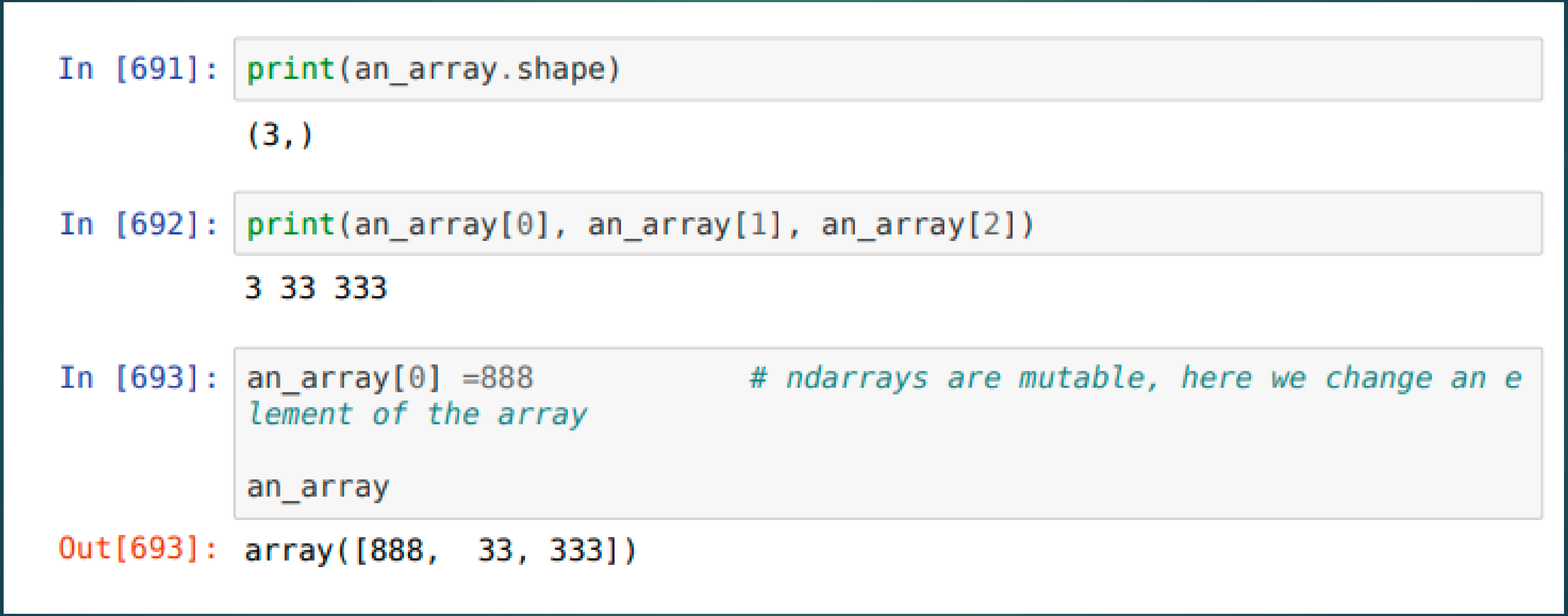
```
In [673]: math.sin(math.radians(45))
Out[673]: 0.7071067811865476

In [674]: math.e , math.log(x) , math.exp(x)
Out[674]: (2.718281828459045,    3.8066624897703196,    3.4934271057485095e + 19)

In [675]: math.pow(3,2) , math.pow(x,2)
Out[675]: (9.0,    2025.0)

In [676]: math.sqrt(x)
Out[676]: 6.708203932499369

In [677]: math.sqrt(2**2 + 3**2)
Out[677]: 3.605551275463989

In [679]: math.cos(math.radians(45))
Out[679]: 0.7071067811865476
```

Numpy Library

- Numerical Python, or "Numpy" for short, is a foundational package on which many of the most common data science packages are built. Numpy provides us with high performance multidimensional arrays which we can use as vectors or matrices. The key features of numpy are:
- ndarrays: n-dimensional arrays of the same data type which are fast and space-efficient. There are a number of built-in methods for ndarrays which allow for rapid processing of data without using loops (e.g., compute the mean).
- Broadcasting: a useful tool which defines implicit behavior between multi-dimensional arrays of different sizes.
- Vectorization: enables numeric operations on ndarrays.
- Input/Output : simplifies reading and writing of data from/to file.



```
In [690]: import numpy as np

          an_array = np.array([3, 33, 333])  # Create a rank 1 array
```



```
In [691]: print(an_array.shape)

          (3,)

In [692]: print(an_array[0], an_array[1], an_array[2])

          3 33 333

In [693]: an_array[0] =888            # ndarrays are mutable, here we change an e
          lement of the array

          an_array

Out[693]: array([888,  33, 333])
```