Experiment No: 02

Basics Of Python

In [1]:
```
print('--------------------------------EXPERIEMENT-02-------------------
---------------')
print('NAME:  Pratyush Srivastava')
print('ROLL NO: 18SCSE1010128')
```

```
--------------------------------EXPERIEMENT-02-------------------------
--------
NAME:  Pratyush Srivastava
ROLL NO: 18SCSE1010128
```

In [2]:
```
!python --version
```

```
Python 3.8.3
```

In [3]:
```
x = 3
print(x, type(x))
```

```
3 <class 'int'>
```

In [4]:
```
print(x + 1)
print(x - 1)
print(x * 2)
print(x ** 2)
```

```
4
2
6
9
```

In [5]:
```
x += 1
print(x)
x *= 2
print(x)
```

```
4
8
```

In [6]:
```
y = 2.5
print(type(y))
print(y, y + 1, y * 2, y ** 2)
```

```
<class 'float'>
2.5 3.5 5.0 6.25
```

In [7]:
```
t, f = True, False
print(type(t))
```

```
<class 'bool'>
```

In [8]:
```
print(t and f)
print(t or f)
print(not t)
print(t != f)
```

```
False
True
False
True
```

In [9]:
```
hello = 'hello'
world = "world"
print(hello, len(hello))
```

```
hello 5
```

In [10]:
```
hw = hello + ' ' + world
```

```
print(hw)
```

```
hello world
```

In [11]:
```
hw12 = '{} {} {}'.format(hello, world, 12)
print(hw12)
```

```
hello world 12
```

In [12]:
```
s = "hello"
print(s.capitalize())
print(s.upper())
print(s.rjust(7))
print(s.center(7))
print(s.replace('l', '(ell)'))
print('  world '.strip())
```

```
Hello
HELLO
  hello
 hello
he(ell)(ell)o
world
```

In [13]:
```
xs = [3, 1, 2]
print(xs, xs[2])
print(xs[-1])
```

```
[3, 1, 2] 2
2
```

In [14]:
```
xs[2] = 'foo'
print(xs)
```

```
[3, 1, 'foo']
```

In [15]:
```
xs.append('bar')
print(xs)
```

```
[3, 1, 'foo', 'bar']
```

In [16]:
```
x = xs.pop()
print(x, xs)
```

```
bar [3, 1, 'foo']
```

In [17]:
```
nums = list(range(5))
print(nums)
print(nums[2:4])
print(nums[2:])
print(nums[:2])
print(nums[:])
print(nums[:-1])
nums[2:4] = [8, 9]
print(nums)
```

```
[0, 1, 2, 3, 4]
[2, 3]
[2, 3, 4]
[0, 1]
[0, 1, 2, 3, 4]
[0, 1, 2, 3]
[0, 1, 8, 9, 4]
```

In [18]:
```
animals = ['cat', 'dog', 'monkey']
for animal in animals:
    print(animal)
```

```
cat
dog
monkey
```

In [19]:
```
animals = ['cat', 'dog', 'monkey']
for idx, animal in enumerate(animals):
    print('#{}: {}'.format(idx + 1, animal))
```

```
#1: cat
```

```
#2: dog
#3: monkey
```

In [20]:
```
nums = [0, 1, 2, 3, 4]
squares = []
for x in nums:
    squares.append(x ** 2)
print(squares)
```

```
[0, 1, 4, 9, 16]
```

In [21]:
```
nums = [0, 1, 2, 3, 4]
squares = [x ** 2 for x in nums]
print(squares)
```

```
[0, 1, 4, 9, 16]
```

In [22]:
```
nums = [0, 1, 2, 3, 4]
even_squares = [x ** 2 for x in nums if x % 2 == 0]
print(even_squares)
```

```
[0, 4, 16]
```

In [23]:
```
d = {'cat': 'cute', 'dog': 'furry'}
print(d['cat'])
print('cat' in d)
```

```
cute
True
```

In [24]:
```
d['fish'] = 'wet'
print(d['fish'])
```

```
wet
```

In [25]:
```
print(d.get('monkey', 'N/A'))
print(d.get('fish', 'N/A'))
```

```
N/A
wet
```

In [26]:
```
del d['fish']
print(d.get('fish', 'N/A'))
```

```
N/A
```

In [27]:
```
d = {'person': 2, 'cat': 4, 'spider': 8}
for animal, legs in d.items():
    print('A {} has {} legs'.format(animal, legs))
```

```
A person has 2 legs
A cat has 4 legs
A spider has 8 legs
```

In [28]:
```
nums = [0, 1, 2, 3, 4]
even_num_to_square = {x: x ** 2 for x in nums if x % 2 == 0}
print(even_num_to_square)
```

```
{0: 0, 2: 4, 4: 16}
```

In [29]:
```
animals = {'cat', 'dog'}
print('cat' in animals)
print('fish' in animals)
```

```
True
False
```

In [30]:
```
animals.add('fish')
print('fish' in animals)
print(len(animals))
```

```
True
3
```

In [31]:
```
animals.add('cat')
print(len(animals))
```

```
animals.remove('cat')
print(len(animals))
```

```
3
2
```

In [32]:
```
animals = {'cat', 'dog', 'fish'}
for idx, animal in enumerate(animals):
    print('#{}: {}'.format(idx + 1, animal))
```

```
#1: fish
#2: cat
#3: dog
```

In [33]:
```
from math import sqrt
print({int(sqrt(x)) for x in range(30)})
```

```
{0, 1, 2, 3, 4, 5}
```

In [34]:
```
d = {(x, x + 1): x for x in range(10)}
t = (5, 6)
print(type(t))
print(d[t])
print(d[(1, 2)])
```

```
<class 'tuple'>
5
1
```

In [35]:
```
def sign(x):
    if x > 0:
        return 'positive'
    elif x < 0:
        return 'negative'
    else:
        return 'zero'
for x in [-1, 0, 1]:
    print(sign(x))
```

```
negative
zero
positive
```

In [36]:
```
def hello(name, loud=False):
    if loud:
        print('HELLO, {}'.format(name.upper()))
    else:
        print('Hello, {}!'.format(name))
hello('Bob')
hello('Fred', loud=True)
```

```
Hello, Bob!
HELLO, FRED
```

In [37]:
```
class Greeter:
    def __init__(self, name):
        self.name = name
    def greet(self, loud=False):
        if loud:
          print('HELLO, {}'.format(self.name.upper()))
        else:
          print('Hello, {}!'.format(self.name))
g = Greeter('Fred')
g.greet()
g.greet(loud=True)
```

```
Hello, Fred!
HELLO, FRED
```

In [38]:
```
import numpy as np
a = np.array([1, 2, 3])
print(type(a), a.shape, a[0], a[1], a[2])
a[0] = 5
print(a)
```

```
<class 'numpy.ndarray'> (3,) 1 2 3
[5 2 3]
```

In [39]:
```
b = np.array([[1,2,3],[4,5,6]])
print(b)
```

```
[[1 2 3]
 [4 5 6]]
```

In [40]:
```
print(b.shape)
print(b[0, 0], b[0, 1], b[1, 0])
```

```
(2, 3)
1 2 4
```

In [41]:
```
a = np.zeros((2,2))
print(a)
```

```
[[0. 0.]
 [0. 0.]]
```

In [42]:
```
b = np.ones((1,2))
print(b)
```

```
[[1. 1.]]
```

In [43]:
```
c = np.full((2,2), 7)
print(c)
```

```
[[7 7]
 [7 7]]
```

In [44]:
```
d = np.eye(2)
print(d)
```

```
[[1. 0.]
 [0. 1.]]
```

In [45]:
```
e = np.random.random((2,2))
print(e)
```

```
[[0.71031002 0.75384614]
 [0.62682118 0.69662839]]
```

In [46]:
```
import numpy as np

a = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])
b = a[:2, 1:3]
print(b)
```

```
[[2 3]
 [6 7]]
```

In [47]:
```
print(a[0, 1])
b[0, 0] = 77
print(a[0, 1])
```

```
2
77
```

In [48]:
```
a = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])
print(a)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

In [49]:
```
row_r1 = a[1, :]
row_r2 = a[1:2, :]
row_r3 = a[[1], :]
print(row_r1, row_r1.shape)
print(row_r2, row_r2.shape)
print(row_r3, row_r3.shape)
```

```
[5 6 7 8] (4,)
[[5 6 7 8]] (1, 4)
[[5 6 7 8]] (1, 4)
```

In [50]:
```
col_r1 = a[:, 1]
```

```
col_r2 = a[:, 1:2]
print(col_r1, col_r1.shape)
print()
print(col_r2, col_r2.shape)
```

```
[ 2  6 10] (3,)

[[ 2]
 [ 6]
 [10]] (3, 1)
```

In [51]:
```
a = np.array([[1,2], [3, 4], [5, 6]])

print(a[[0, 1, 2], [0, 1, 0]])

print(np.array([a[0, 0], a[1, 1], a[2, 0]]))
```

```
[1 4 5]
[1 4 5]
```

In [52]:
```
print(a[[0, 0], [1, 1]])
```

```
[2 2]
```

In [53]:
```
print(np.array([a[0, 1], a[0, 1]]))
```

```
[2 2]
```

In [54]:
```
a = np.array([[1,2,3], [4,5,6], [7,8,9], [10, 11, 12]])
print(a)
```

```
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
```

In [55]:
```
b = np.array([0, 2, 0, 1])
print(a[np.arange(4), b])
```

```
[ 1  6  7 11]
```

In [56]:
```
a[np.arange(4), b] += 10
print(a)
```

```
[[11  2  3]
 [ 4  5 16]
 [17  8  9]
 [10 21 12]]
```

In [57]:
```
import numpy as np
a = np.array([[1,2], [3, 4], [5, 6]])
bool_idx = (a > 2)
print(bool_idx)
```

```
[[False False]
 [ True  True]
 [ True  True]]
```

In [58]:
```
print(a[bool_idx])

print(a[a > 2])
```

```
[3 4 5 6]
[3 4 5 6]
```

In [59]:
```
x = np.array([1, 2])
y = np.array([1.0, 2.0])
z = np.array([1, 2], dtype=np.int64)
print(x.dtype, y.dtype, z.dtype)
```

```
int32 float64 int64
```

In [60]:
```
x = np.array([[1,2],[3,4]], dtype=np.float64)
y = np.array([[5,6],[7,8]], dtype=np.float64)

print(x + y)
print(np.add(x, y))
```

```
[[ 6.  8.]
 [10. 12.]]
[[ 6.  8.]
 [10. 12.]]
```

In [61]:
```
print(x - y)
print(np.subtract(x, y))
```

```
[[-4. -4.]
 [-4. -4.]]
[[-4. -4.]
 [-4. -4.]]
```

In [62]:
```
print(x * y)
print(np.multiply(x, y))
```

```
[[ 5. 12.]
 [21. 32.]]
[[ 5. 12.]
 [21. 32.]]
```

In [63]:
```
print(x / y)
print(np.divide(x, y))
```

```
[[0.2        0.33333333]
 [0.42857143 0.5       ]]
[[0.2        0.33333333]
 [0.42857143 0.5       ]]
```

In [64]:
```
print(np.sqrt(x))
```

```
[[1.         1.41421356]
 [1.73205081 2.        ]]
```

In [65]:
```
x = np.array([[1,2],[3,4]])
y = np.array([[5,6],[7,8]])
v = np.array([9,10])
```

```
w = np.array([11, 12])

print(v.dot(w))
print(np.dot(v, w))
```

```
219
219
```

In [66]:
```
print(v @ w)
```

```
219
```

In [67]:
```
print(x.dot(v))
print(np.dot(x, v))
print(x @ v)
```

```
[29 67]
[29 67]
[29 67]
```

In [68]:
```
print(x.dot(y))
print(np.dot(x, y))
print(x @ y)
```

```
[[19 22]
 [43 50]]
[[19 22]
 [43 50]]
[[19 22]
 [43 50]]
```

In [69]:
```
x = np.array([[1,2],[3,4]])
print(np.sum(x))
print(np.sum(x, axis=0))
print(np.sum(x, axis=1))
```

```
10
[4 6]
[3 7]
```

In [70]:
```
print(x)
print("transpose\n", x.T)
```

```
[[1 2]
 [3 4]]
transpose
 [[1 3]
 [2 4]]
```

In [71]:
```
v = np.array([[1,2,3]])
print(v )
print("transpose\n", v.T)
```

```
[[1 2 3]]
transpose
 [[1]
 [2]
 [3]]
```

In [72]:
```
x = np.array([[1,2,3], [4,5,6], [7,8,9], [10, 11, 12]])
v = np.array([1, 0, 1])
y = np.empty_like(x)

for i in range(4):
    y[i, :] = x[i, :] + v
print(y)
```

```
[[ 2  2  4]
 [ 5  5  7]
 [ 8  8 10]
 [11 11 13]]
```

In [73]:
```
vv = np.tile(v, (4, 1))
vv
```

Out[73]:
```
array([[1, 0, 1],
       [1, 0, 1],
```

```
                [1, 0, 1],
                [1, 0, 1]])
```

In [74]:
```
y = x + vv
print(y)
```

```
[[ 2  2  4]
 [ 5  5  7]
 [ 8  8 10]
 [11 11 13]]
```

In [75]:
```
import numpy as np

x = np.array([[1,2,3], [4,5,6], [7,8,9], [10, 11, 12]])
v = np.array([1, 0, 1])
y = x + v
print(y)
```

```
[[ 2  2  4]
 [ 5  5  7]
 [ 8  8 10]
 [11 11 13]]
```

In [76]:
```
v = np.array([1,2,3])
w = np.array([4,5])
print(np.reshape(v, (3, 1)) * w)
```

```
[[ 4  5]
 [ 8 10]
 [12 15]]
```

In [77]:
```
x = np.array([[1,2,3], [4,5,6]])

print(x + v)
```

```
[[2 4 6]
 [5 7 9]]
```

In [78]:
```
print((x.T + w).T)
```

```
[[ 5  6  7]
 [ 9 10 11]]
```

In [79]:
```
print(x + np.reshape(w, (2, 1)))
```

```
[[ 5  6  7]
 [ 9 10 11]]
```

In [80]:
```
print(x * 2)
```

```
[[ 2  4  6]
 [ 8 10 12]]
```

In [81]:
```
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)
print(quicksort([1, 41, 2, -3, 86, 8]))
```

```
[-3, 1, 2, 8, 41, 86]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [82]:
```
def a(b):
```

```
    if len(b) <= 1:
        return b
    p=b[len(b)//2]
    print(p)
    l=[x for x in b if x < p]
    m=[x for x in b if x == p]
    r=[x for x in b if x > p]
    print("left",l)
    print("middle",m)
    print("right",r)
    return a(l)+m+a(r)
print(a([2,1,9,4,0,5,-4]))
```

```
4
left [2, 1, 0, -4]
middle [4]
right [9, 5]
0
left [-4]
middle [0]
right [2, 1]
1
left []
middle [1]
right [2]
5
left []
middle [5]
right [9]
[-4, 0, 1, 2, 4, 5, 9]
```

In [83]:

```
animals = ['cat', 'dog', 'monkey']
for i in animals:
    print('#{}: {}'.format(i , animals))
```

```
#cat: ['cat', 'dog', 'monkey']
#dog: ['cat', 'dog', 'monkey']
#monkey: ['cat', 'dog', 'monkey']
```

In [84]:
```
from math import sqrt
print({int(sqrt(x)) for x in range(30)})
```

```
{0, 1, 2, 3, 4, 5}
```

In [85]:
```
def a(name,b=True):
    if b:
        print('Hello,{}'.format(name.lower()))
    else:
        print('Hello,{}'.format(name.capitalize()))
a('Anil')
a('Pradeep',b=False)
```

```
Hello,anil
Hello,Pradeep
```

In [86]:
```
class a:
    def __init__(self, name):
        self.name=name
    def b(self,loud=False):
        if loud:
            print('Hello,{}'.format(self.name.upper()))
        else:
            print('Hello,{}'.format(self.name))
g=a('Ayush')
g.b()
g.b(loud=True)
```

```
Hello,Ayush
Hello,AYUSH
```

In [87]:
```
import numpy as np
a=np.array([[1,2,3],[4,5,6],[7,8,9]])
print(a,a.shape,a[0,2],a.T)
v = np.array([1, 0, 1])
y = np.empty_like(a)
print(y)
for i in range(3):
```

```
    y[i, :] = a[i, :] + v
print(y)
```

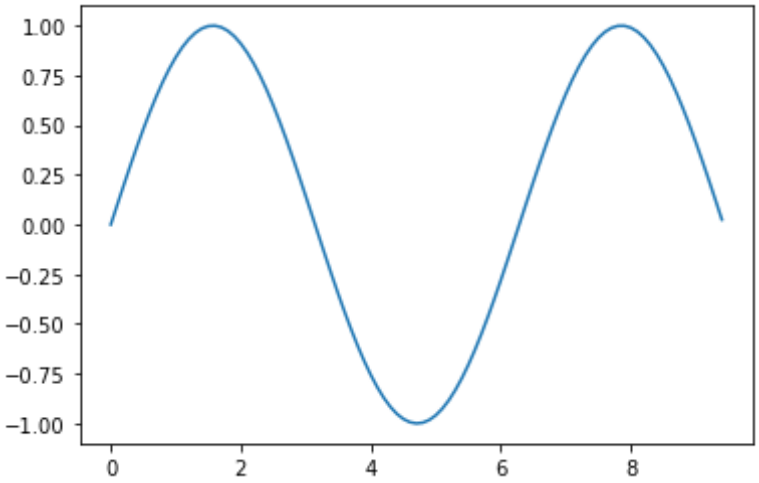
```
[[1 2 3]
 [4 5 6]
 [7 8 9]] (3, 3) 3 [[1 4 7]
 [2 5 8]
 [3 6 9]]
[[      1       0       1]
 [      0       0       0]
 [      0       0 4456543]]
[[ 2  2  4]
 [ 5  5  7]
 [ 8  8 10]]
```

In [88]:
```
import matplotlib.pyplot as plt
```

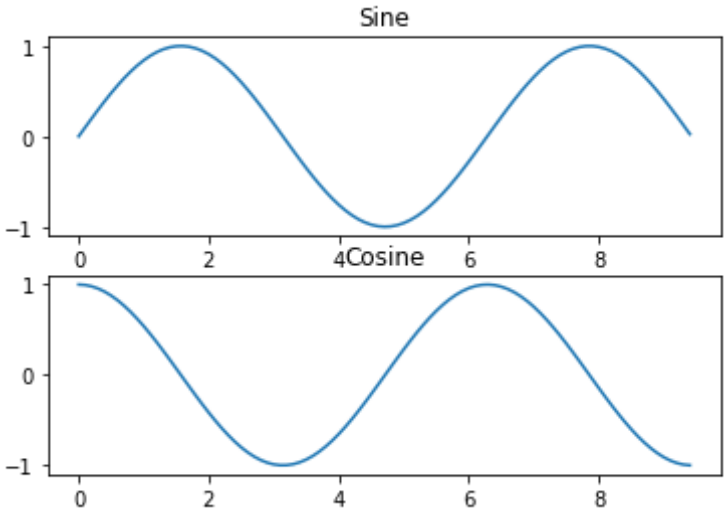
In [89]:
```
%matplotlib inline
```

In [90]:
```
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)

plt.plot(x, y)
```

Out[90]:
```
[<matplotlib.lines.Line2D at 0x1dc3577afa0>]
```

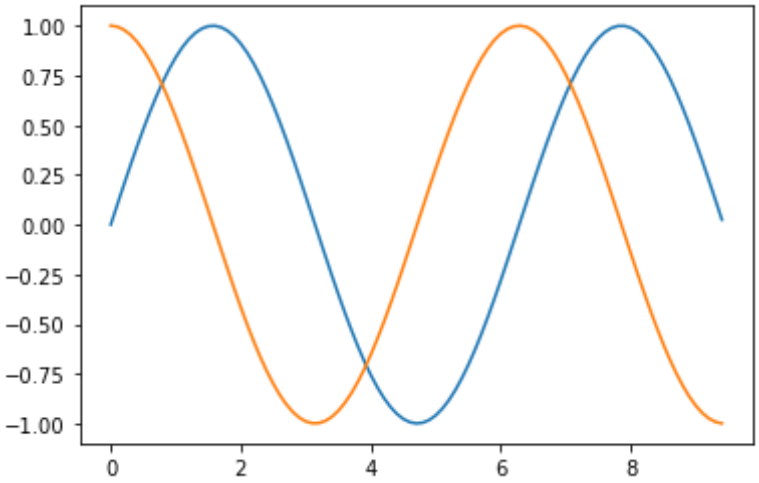


In [91]:
```
x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)

plt.subplot(2, 1, 1)

plt.plot(x, y_sin)
plt.title('Sine')

plt.subplot(2, 1, 2)
plt.plot(x, y_cos)
plt.title('Cosine')

plt.show()
```



In [92]:
```
import matplotlib.pyplot as plt
%matplotlib inline
x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)
plt.plot(x, y_sin)
plt.plot(x, y_cos)
```

Out[92]:
```
[<matplotlib.lines.Line2D at 0x1dc358c74c0>]
```



In [ ]: