Introduction to DataScience

Python functions

```
x=1
y=2
x+y
```

```
3
```

```
def add_numbers(x,y):
    return x+y
add_numbers(1,2)
```

```
3
```

```
def add_numbers(x,y,z=None):
    if(z==None):
        return x+y
    else:
        return x+y+z
print(add_numbers(1,2,))
print(add_numbers(1,2,3))
```

```
3
6
```

```
def add_numbers(x,y,z=None,flag=False):
        if(flag):
            print("Flag is True")
        if(z==None):
            return x+y
        else:
            return x+y+z
print(add_numbers(1,2,flag=True))
```

```
Flag is True
3
```

Python Types and Sequences

```
type ('This is a string')
```

```
str
```

```
type(None)
```

```
NoneType
```

```
type(111)
```

```
int
```

```
type(13.31)
```

```
float
```

```
type (add_numbers)
```

```
function
```

```
x=(1,'a',2,'b')
type(x)
```

```
tuple
```

```
x=[1,'a',2,'b']
type(x)
```

```
list
```

```
x.append(3.3)
print(x)
```

```
[1, 'a', 2, 'b', 3.3]
```

```
for item in x:
    print (item)
```

```
1
a
2
b
3.3
```

```
i=0
while (i!=len(x)):
    print(x[i])
    i=i+1
```

```
1
a
2
b
3.3
```

```
[1]*3
```

```
[1, 1, 1]
```

```
[1,2]+[3,4]
```

```
[1, 2, 3, 4]
```

```
1 in [2]
```

```
False
```

```
x='This is a string'
print(x[0])
print(x[0:1])
print(x[0:2])
```

```
T
T
Th
```

```
x[-2]
```

```
'n'
```

```
x[-4:-2]
```

```
'ri'
```

```
firstname='Krushnal'
lastname='Sony'
print(firstname + '' +lastname)
print(firstname*3)
print('Krush'in firstname)
```

```
KrushnalSony
KrushnalKrushnalKrushnal
True
```

```
'Krush' +str(2)
```

```
'Krush2'
```

```
x={'Krushnal Sony':'kru@gmail.com','abxsd':'asd@gmail.com'}
x['Krushnal Sony']
```

```
'kru@gmail.com'
```

```
for name in x:
    print (x[name])
```

```
kru@gmail.com
```

```
for email in x.values():
    print (email)
```

```
kru@gmail.com
asd@gmail.com
```

```
for name,email in x.items():
    print(name)
    print(email)
```

```
Krushnal Sony
kru@gmail.com
abxsd
asd@gmail.com
```

```
x=('krushnal','Sony','kru@gmail.com')
fname,lname,email=x
```

```
lname
```

```
'Sony'
```

```
fname
```

```
'krushnal'
```

More on Strings

```
sales_record={'price':3.24,
              num_items:4
              'person':'Krush'}
sales_statement='{} bought items(s) at price of {} each for total of{}'
print(sales_statement.format(sales_record['person'],
                             sales_record['num_items'],
                             sales_record['price'],
                              sales_record['num_items'],sales_record['price']))
```

```
  File "<ipython-input-81-935fed4fa280>", line 3
    'person':'Krush'}
           ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

Date and time

```
import datetime as dt
import time as tm
```

```
import time as tm
```

```
tm.time()
```

```
1587195900.0925856
```

```
dtnow=dt.datetime.fromtimestamp(tm.time())
dtnow
```

```
datetime.datetime(2020, 4, 18, 13, 16, 3, 796401)
```

```
dtnow.year,dtnow.month,dtnow.day,dtnow.hour,dtnow.minute
```

```
(2020, 4, 18, 13, 16)
```

```
delta=dt.timedelta(days=100)
delta
```

```
datetime.timedelta(days=100)
```

```
today=dt.date.today()
```

```
today-delta
```

```
datetime.date(2020, 1, 9)
```

```
today>today-delta
```

```
True
```

Objects and Map()

```
class Person:
    departmemt='School of Information'
    def set_name (self,new_name):
        self.name=new_name
    def set_location(self,new_location):
        self.location=new_location
```

```
person=Person()
person.set_name('Sony')
person.set_location('Ahmedabad')
print(''.format(person.name,person.location))
```

```
store1=[11,12.21,14,15]
store2=[16,10.21,17,13]
cheapest=map(min,store1,store2)
cheapest
```

```
.
```

```
<map at 0x1835eae79c8>
```

```
for item in cheapest :
    print(item)
```

```
11
10.21
14
13
```

Lambda and List Comprehension

```
my_function=lambda a,b,c:a+b
```

```
my_function(1,2,3)
```

```
3
```

```
my_list = []
for number in range(0,100):
    if number % 2==0:
        my_list.append(number)
my_list
```

```
[0,
 2,
 4,
 6,
 8,
 10,
 12,
 14,
 16,
 18,
 20,
 22,
 24,
 26,
 28,
 30,
 32,
 34,
 36,
 38,
 40,
 42,
 44,
 46,
 48,
 50,
 52,
 54,
 56,
 58,
 60,
 62,
 64,
 66,
 68,
 70,
 72,
 74,
 76,
 78,
 80,
 82,
 84,
 86,
 88,
 90,
 92,
 94,
 96,
 98]
```

NumPy

```
import numpy as np

mylist=[1,2,3]
x=np.array(mylist)
```

```
x=np.array(mylist)
x
```

```
array([1, 2, 3])
```

```
y=np.array([4,5,6])
y
```

```
array([4, 5, 6])
```

```
m=np.array([[1,2,3],[4,5,6]])
m
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

```
m.shape
```

```
(2, 3)
```

```
n=np.arange(0,30,2)
n
```

```
array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28])
```

```
n= n.reshape(3,5)
n
```

```
array([[ 0,  2,  4,  6,  8],
       [10, 12, 14, 16, 18],
       [20, 22, 24, 26, 28]])
```

```
o=np.linspace(0,4,9)
o
```

```
array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. ])
```

```
o.resize(3,3)
o
```

```
array([[0. , 0.5, 1. ],
       [1.5, 2. , 2.5],
       [3. , 3.5, 4. ]])
```

```
np.ones((3,2))
```

```
array([[1., 1.],
       [1., 1.],
       [1., 1.]])
```

```
np.zeros((2,3))
```

```
array([[0., 0., 0.],
```

```
np.eye(3)
```

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

```
np.array([1,2,3]*3)
```

```
array([1, 2, 3, 1, 2, 3, 1, 2, 3])
```

```
p=np.ones([2,3],int)
p
```

```
array([[1, 1, 1],
       [1, 1, 1]])
```

```
np.hstack([p,2*p])
```

```
array([[1, 1, 1, 2, 2, 2],
       [1, 1, 1, 2, 2, 2]])
```

Operations

```
x+y
```

```
array([5, 7, 9])
```

```
x*y
```

```
array([ 4, 10, 18])
```

```
x**2
```

```
array([1, 4, 9], dtype=int32)
```

```
x.dot(y)
```

```
32
```

```
z=np.array([y,y**2])
z
```

```
array([[ 4,  5,  6],
       [16, 25, 36]])
```

```
z.T
```

```
array([[ 4, 16],
       [ 5, 25],
       [ 6, 36]])
```

```
z.T.shape
```

```
(3, 2)
```

```
z.dtype
```

```
dtype('int32')
```

```
a=np.array([-4,-2,1,3,5])
```

```
a.sum()
```

```
3
```

```
a.max()
```

```
5
```

```
a.min()
```

```
-4
```

```
a.mean()
```

```
0.6
```

```
a.std()
```

```
3.2619012860600183
```

```
a.argmax()
```

```
4
```

```
a.argmin()
```

```
0
```

Indexing / Slicing

```
s=np.arange(13)**2
s
```

```
array([  0,   1,   4,   9,  16,  25,  36,  49,  64,  81, 100, 121, 144],
      dtype=int32)
```

```
s[0],s[4],s[:3]
```

```
(0, 16, array([0, 1, 4], dtype=int32))
```

```
s[:-4]
```

```
array([ 0,  1,  4,  9, 16, 25, 36, 49, 64], dtype=int32)
```

```
s[-5::-2]
```

```
array([64, 36, 16,  4,  0], dtype=int32)
```

```
r=np.arange(36)
r.resize((6,6))
r
```

```
array([[ 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, 25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34, 35]])
```

```
r[2,2]
```

```
14
```

```
r[:2,:-1]
```

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

```
r[-1,::2]
```

```
array([30, 32, 34])
```

```
r[r>30]
```

```
array([31, 32, 33, 34, 35])
```

```
r[3,3:6]
```

```
array([21, 22, 23])
```

```
r[r>30]=30
r
```

```
array([[ 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, 25, 26, 27, 28, 29],
       [30, 30, 30, 30, 30, 30]])
```

```
r2=r[:3,:3]
r2
```

```
array([[ 0,  1,  2],
       [ 6,  7,  8],
       [12, 13, 14]])
```

```
r2[:]=0
r2
```

```
array([[0, 0, 0],
       [0, 0, 0],
       [0, 0, 0]])
```

```
r
```

```
array([[ 0,  0,  0,  3,  4,  5],
       [ 0,  0,  0,  9, 10, 11],
       [ 0,  0,  0, 15, 16, 17],
       [18, 19, 20, 21, 22, 23],
       [24, 25, 26, 27, 28, 29],
       [30, 30, 30, 30, 30, 30]])
```

Iterating over arrays

```
test=np.random.randint(0,10,(4,3))
test
```

```
array([[2, 9, 7],
       [7, 2, 5],
       [2, 1, 7],
       [4, 0, 3]])
```

```
for row in test:
    print(row)
```

```
[2 9 7]
[7 2 5]
[2 1 7]
[4 0 3]
```

```
for i in range(len(test)):
    print(test[i])
```

```
[2 9 7]
[7 2 5]
[2 1 7]
[4 0 3]
```

```
for i ,row in enumerate(test):
    print('row',i,'is,row')
```

```
row 0 is,row
row 1 is,row
row 2 is,row
row 3 is,row
```

```
test2=test**2
test2
```

```
array([[ 4, 81, 49],
       [49,  4, 25],
       [ 4,  1, 49],
       [16,  0,  9]], dtype=int32)
```

```
for i,j in zip(test,test2):
    print(i,'+',j,'=',i+j)
```

```
[2 9 7] + [ 4 81 49] = [ 6 90 56]
[7 2 5] + [49  4 25] = [56  6 30]
[2 1 7] + [ 4  1 49] = [ 6  2 56]
[4 0 3] + [16  0  9] = [20  0 12]
```