Welcome

Let's begin

Some Basics Arithmetic Operations

Basic arithmetic operations are built into the Python langauge. Here are some examples. In particular, note that exponentiation is done with the ** operator.

In [6]:

```
a = 12
b = 2
c = 5
d = -3

print("Addition:",a+b)
print("Subtraction:",a-b)
print("Multiplication:",a*b)
print("Division :",a/b)
print("Power :",a**b)
print("Modulus/Remainder:",a%c)
print("Absolute:",abs(d))
```

```
Addition: 14
Subtraction: 10
Multiplication: 24
Division : 6.0
Power : 144
Modulus/Remainder: 2
Absolute: 3
```

Python Libraries

The Python language has only very basic operations. Most math functions are in various math libraries. The `numpy` library is convenient library. This next cell shows how to import `numpy` with the prefix `np`, then use it to call a common mathematical functions.

In [7]:

```
import numpy as np

# Mathematical constants

print(np.pi)
print(np.e)

# Trignometric Functions

angle = np.pi/4

print(np.sin(angle))
print(np.cos(angle))
print(np.tan(angle))
```

```
3.141592653589793
2.718281828459045
0.7071067811865476
0.7071067811865476
0.9999999999999999
```

Working with Lists

In [8]:
```
xList = [1, 2, 3, 4]
xList
```

Out[8]:
```
[1, 2, 3, 4]
```

Concatentation is the operation of joining one list to another.

In [10]:
```
# Concatenation
xlist = [1,2,3,4]
ylist = [5,6,7,8]

xlist + ylist
```

Out[10]:
```
[1, 2, 3, 4, 5, 6, 7, 8]
```

Sum a list of numbers

In [11]:
```
np.sum(xlist)
# np.sum(y)
```

Out[11]:
```
10
```

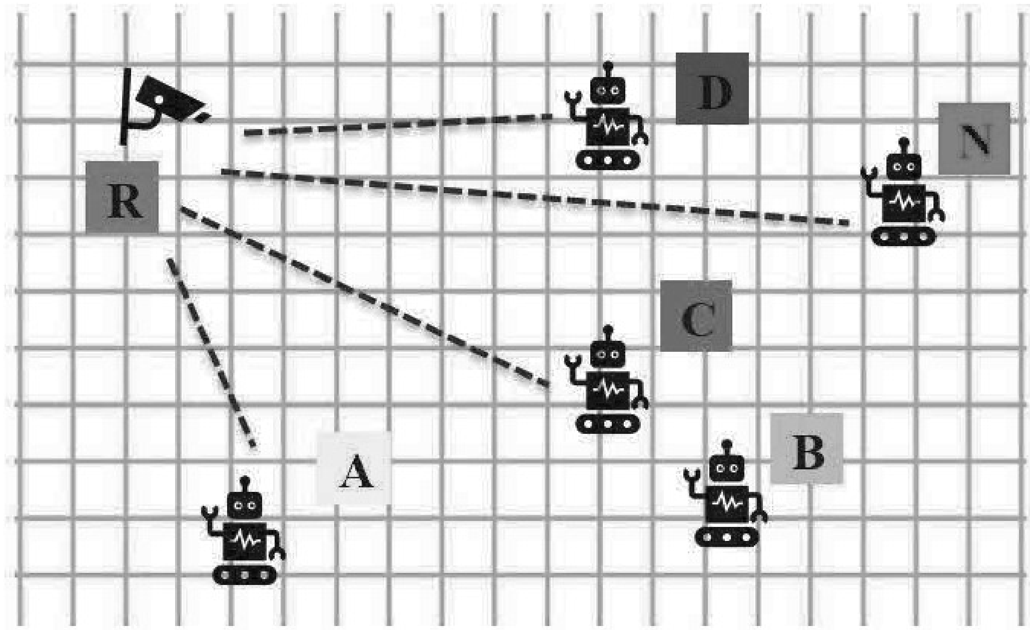
An element-by-element operation between two lists may be performed with

In [12]:
```
print(np.add(xlist,ylist))
print(np.dot(xlist,ylist))
```

```
[ 6  8 10 12]
70
```

Example 1

Consider an interactive and cognitive environment (ICE) in which a smart camera is monitoring robot movement from one location to another. Let a robot be at location A for some time instant and then moves to point B and eventually reaches at point C and so on and so forth shown in the Fig. Calculates a distance between reference point R (4,0) of a camera and A, B, and C and N number of locations



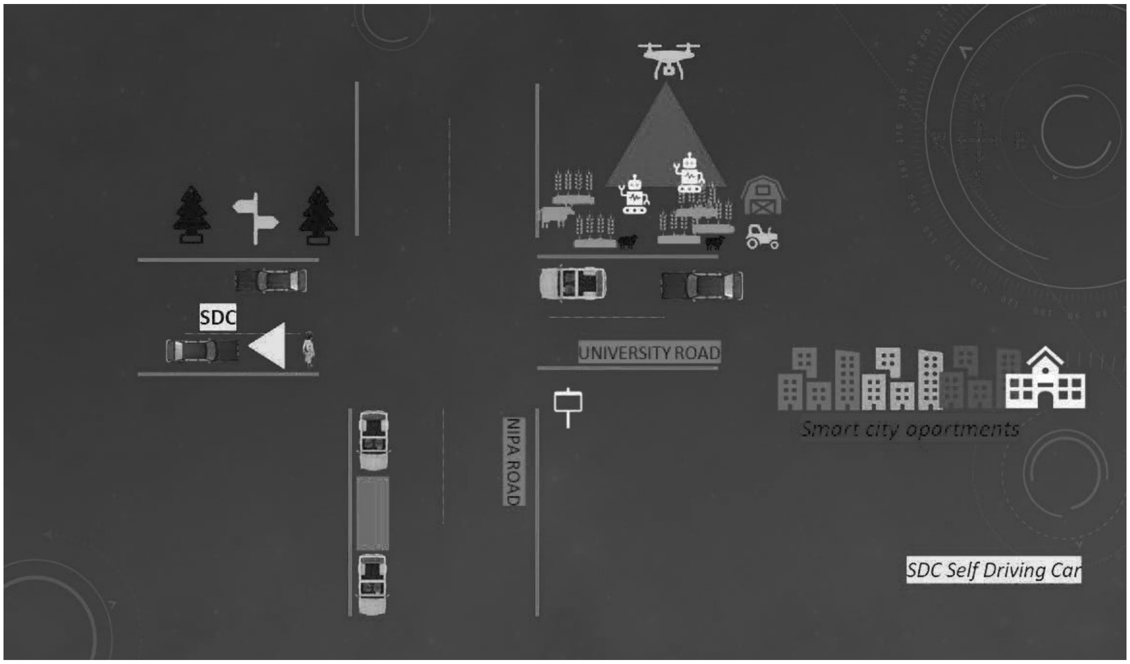
In [17]:

```
#import math

orig = [4, 0]
user1 = [6, 6]
user2 = [9, 5]
user3 = [16, 2]
user4 = [5, 9]
user5 = [10, 8]

distance1 = math.sqrt( ((user1[0]-orig[0])**2)+((user1[1]-orig[1])**2) )
distance2 = math.sqrt( ((user2[0]-orig[0])**2)+((user2[1]-orig[1])**2) )
distance3 = math.sqrt( ((user3[0]-orig[0])**2)+((user3[1]-orig[1])**2) )
distance4 = math.sqrt( ((user4[0]-orig[0])**2)+((user4[1]-orig[1])**2) )
distance5 = math.sqrt( ((user5[0]-orig[0])**2)+((user5[1]-orig[1])**2) )
print("Distance between user1 and tower: {:.3f}".format(distance1)+"cm")
print("Distance between user2 and tower: {:.3f}".format(distance2)+"cm")
print("Distance between user3 and tower: {:.3f}".format(distance3)+"cm")
print("Distance between user4 and tower: {:.3f}".format(distance4)+"cm")
print("Distance between user5 and tower: {:.3f}".format(distance5)+"cm")
```

```
Distance between user1 and tower: 6.325cm
Distance between user2 and tower: 7.071cm
Distance between user3 and tower: 12.166cm
Distance between user4 and tower: 9.055cm
Distance between user5 and tower: 10.000cm
```

Example 2

Consider a scenario shown in the figure below in which a Self-driving car (SDC) is moving with an acceleration of 120 m/sec^2 and having mass equals 160 kg. Calculate the force acting on an SDC.



In [18]:

```
a = 120   #m/s**2
m = 160   #kg

F = m*a
print("Force applied is :",F,"N")
```

```
Force applied is : 19200 N
```

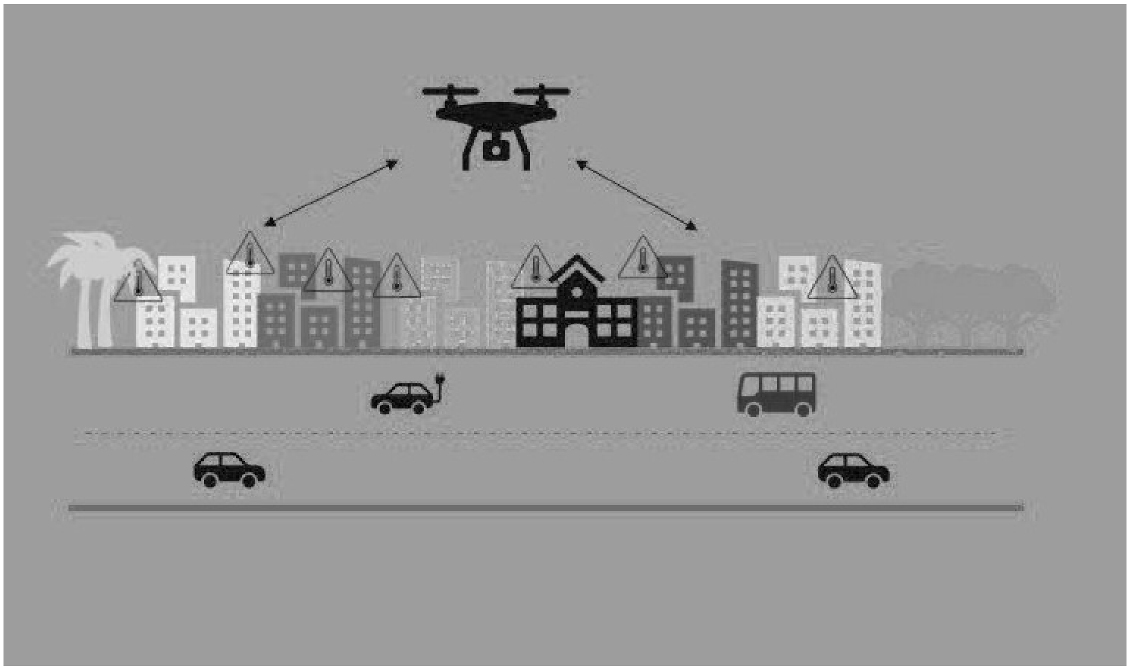
In [63]:

```
import numpy as np
radius = 5

area_t = np.pi*(radius**2)

if(area_t<20):
 print("Area :{:.2f}".format(area_t)+"m2")
 print("Descend the UAV")
else:
  print("Area :{:.2f}".format(area_t)+"m2")
  print("Ascend the UAV")
```

```
Area :78.54m2
Ascend the UAV
```

Example 3

Consider the following scenario where the UAV receives temperature data from the installed sensors in a residential area. Assume that there are nine sensors installed that are measuring temperature in centigrade. Calculate the average temperature in F.



In [24]:

```
sensor_val = [37,36.5,38,33,32.8,40,30,52,35] #in degree celsius

#v = len(sensor_val)

average_sensor_value = (np.sum(sensor_val))/len(sensor_val)

average_sensor_value_K = average_sensor_value*1.8 +32

print("Average sensor valuess in Farenhiet :",average_sensor_value_K,"F")


# Can you try to convert AVG sensor temprature to Kelvin ?
```

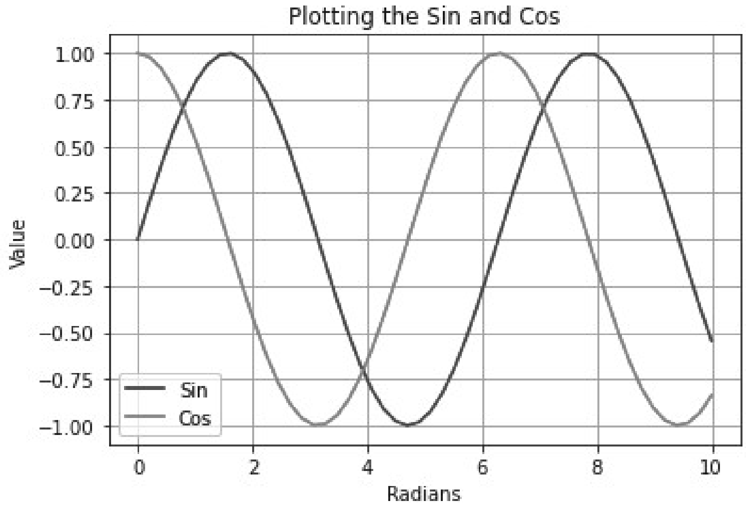
```
Average sensor valuess in Farenhiet : 98.86 F
```

Plotting with Matplotlib

Importing the `matplotlib.pyplot` library gives IPython notebooks plotting functionality very similar to Matlab's. Here are some examples using functions from the

In [25]:

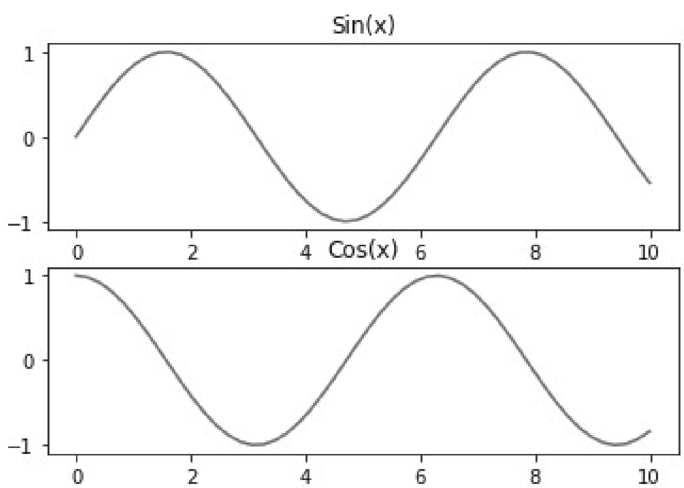
```
%matplotlib inline

import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0,10)
y = np.sin(x)
z = np.cos(x)

plt.plot(x,y,'b',x,z,'r')
plt.xlabel('Radians');
plt.ylabel('Value');
plt.title('Plotting the Sin and Cos')
plt.legend(['Sin','Cos'])
plt.grid()
```



In [27]:

```
plt.subplot(2,1,1)
plt.plot(x,y)
plt.title('Sin(x)')

plt.subplot(2,1,2)
plt.plot(x,z)
plt.title('Cos(x)')
```

Out[27]: `Text(0.5, 1.0, 'Cos(x)')`



Example 4

Consider the following scenario where the cognitive mobile users communicate with the Basestation (BS) in the CR-IoT network. Assume that there are three active mobile users [user1, user2, and user3], and the radio devices of each mobile user stay for a certain period and then get drained. Suppose that user one portable battery lasts on average, three years with a standard deviation of 0.5, user two battery lasts on average, six years with a standard deviation of 0.8, and user three battery lasts on average, five years with a standard deviation of 0.7.

Determine the following: