Name: Mostafa Nasser

Python for Data Science Day 1

In [4]:
```
#Importing Required Libraries
import numpy as np   #numerical python for mathematical operations on arrays
import collections   #for counting the frequency for each die
import itertools     #for joining multiple nested lists into one list (to make frequency works)
```

Write a dice game, where we throw 2 dice (i.e. 2 random numbers from 1 to 6). Display the dice readings + sum. For example Dice: [4, 5]. Sum: 9 If same number on both dice: display BINGO! Else if sum > 8, display WINNER! Otherwise, display LOOSER!

In [2]:
```
def check():
    dice1 = np.random.randint(1,7)
    dice2 = np.random.randint(1,7)
    print("Dice 1 Reading :",dice1)
    print("Dice 2 Reading :",dice2)
    if dice1 == dice2:
        print("BINGO!")
    elif dice1+dice2 > 8:
        print("WINNER!")
    else:
        print("LOOSER!")

check()
```

```
Dice 1 Reading : 4
Dice 2 Reading : 2
LOOSER!
```

LAB 2: Write a throwDice(numDice=1, numThrows=1) function. It should return a 2D array indicating the results. For example: throwDice(2, 3) should return something like [[1, 4], [3, 5], [5, 6]] (reading of the 2 dice when throwed 3 times) Defaults call throwDice() should return something like [[3]] If we throwDice(2, 1000) (2

dice for 1000 times), print out the frequency of the sum of each throw as a dict. For example: { 2: 50, 3: 44, 4: 37, ... } Does the above result comply with probability of sums of 2 dice?

In [5]:

```
def throwDice(numDice=1, numThrows=1):
    arr=[]
    for i in range(numThrows):
        x= list(np.random.randint(1,7,size=numDice))
        arr.append(x)
    return arr


#throwing 2 dices 1000 times
throws = throwDice(numDice=2, numThrows=1000)
freq = {}
for i in range(len(throws)):
    s = sum(throws[i])
    if s not in freq:
        freq[s] = 1
    else:
        freq[s] = freq[s] + 1
print(freq)
print(sum(freq.values()))
```

```
{4: 83, 6: 128, 5: 112, 10: 103, 7: 165, 8: 125, 9: 118, 12: 38, 2: 27, 11: 52, 3: 49}
1000
```