In [ ]:

```
#Today Agenda :
                              1. Dictionary Comprehension
                              2. Set Comprehension
                              3. Tuple Comprehension
                              4. Packages and Modules
                              5. Regular Expressions
```

In [ ]:

```
#Dictionary Comprehension:
 Syntax:
        resD= {key:value for key,value in iterable if key,value satisfy condition}
```

In [1]:

```
#Example:
inputlist = [1,2,3,4,5,6,7,8,9,10]
resD = {}
for i in inputlist:
    resD[i]=pow(i,i)
print(resD)
```

```
{1: 1, 2: 4, 3: 27, 4: 256, 5: 3125, 6: 46656, 7: 823543, 8: 16777216, 9: 387420489, 10: 10000000000}
```

In [3]:

```
print({i:pow(i,i) for i in [1,2,3,4,5,6,7,8,9,10]},end=" ")
```

```
{1: 1, 2: 4, 3: 27, 4: 256, 5: 3125, 6: 46656, 7: 823543, 8: 16777216, 9: 387420489, 10: 10000000000}
```

In [4]:

```
#Example:2
iplteams = ["CSK","MI","RCB","RR","Punjab"]
captains = ["Dhoni","Rohit","Kohli","Smith","K L Rahul"]
res = {}
for key,value in zip(iplteams,captains):
    res[key]=value

print(res)
```

```
{'CSK': 'Dhoni', 'MI': 'Rohit', 'RCB': 'Kohli', 'RR': 'Smith', 'Punjab': 'K L Rahul'}
```

In [5]:
```
{key:value for key,value in zip(iplteams,captains)}
```

Out[5]:
```
{'CSK': 'Dhoni',
 'MI': 'Rohit',
 'RCB': 'Kohli',
 'RR': 'Smith',
 'Punjab': 'K L Rahul'}
```

In [7]:
```
print({iplteams:captains for iplteams,captains in zip(iplteams,captains)})
```

```
{'CSK': 'Dhoni', 'MI': 'Rohit', 'RCB': 'Kohli', 'RR': 'Smith', 'Punjab': 'K L Rahul'}
```

In [9]:
```
res={iplteams:captains for iplteams in ["csk","MI","RR","smith","punjab"]}
res
```

Out[9]:
```
{'csk': ['Dhoni', 'Rohit', 'Kohli', 'Smith', 'K L Rahul'],
 'MI': ['Dhoni', 'Rohit', 'Kohli', 'Smith', 'K L Rahul'],
 'RR': ['Dhoni', 'Rohit', 'Kohli', 'Smith', 'K L Rahul'],
 'smith': ['Dhoni', 'Rohit', 'Kohli', 'Smith', 'K L Rahul'],
 'punjab': ['Dhoni', 'Rohit', 'Kohli', 'Smith', 'K L Rahul']}
```

In [10]:
```
#Set Comprehensions:  #syntax:   {outputvariable for value in iterable if conditions}
s1 = {1,6,8,8,3,4,5,6,89,90,21,25}
s2 = set()
for i in s1:
    if i%3==0:
        if i%5==0:
            print(i,end=" ")
```

```
90
```

In [13]:
```
{i for i in {1,6,8,8,3,4,5,6,89,90,21,25} if i%3==0 if i%5==0 if i%6==0}
```

Out[13]:
```
{90}
```

In [19]:
```
#Genarator or Tuple Comprehension:
#Syntax:  (outvarible for value in iterable if conditions)
tuple(i for i in (5,6,7,8,9,21,24,30) if i>=3)
```

Out[19]:
```
(5, 6, 7, 8, 9, 21, 24, 30)
```

In [ ]:

```
#Packages and Modules?

                 Package-Subpackages-Modules
    # Package is collection of modules and sub packages its root directory
    #Module is a Python (.py ) file that containing python definitions,functions,instructions.
#Module Types:
           1. User Defined Modules
           2. Built-in Modules
#Packages Types:
           1. User Defined
           2. Built-in
```

In [26]:

```
#How to create User Defined Modules?

#How to use user defined Modules?

#import modulename
import calc
from calc import add,sub
from calc import *
print(calc.add(100,500))          #modulename.functionname(parameters)
print(calc.sub(900,500))
print(calc.mul(10,100))
print(calc.div(500,5))
```

```
600
400
1000
100.0
```

In [33]:

```
#Using factorial user defined module:
import factorial1                                  #Its imports entire module
from factorial1 import fact1,ispalindrome     #Its imports only specific functions in a module

from factorai1 import *                          #Its import all functions in a module
factorial1.fact1(6)
factorial1.ispalindrome(input("enter a String"))
```

```
enter a Stringpython
```

Out[33]:

```
False
```

In [39]:

```
#Built - in Modules:
import sys
print((sys.path),end=" ")
```

```
['C:\\Users\\Mission Impossible\\Desktop\\Python_Online_Programmingg\\Day-11', 'C:\\ProgramData\\Anaconda3\\p
ython37.zip', 'C:\\ProgramData\\Anaconda3\\DLLs', 'C:\\ProgramData\\Anaconda3\\lib', 'C:\\ProgramData\\Anacon
da3', '', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages
\\win32', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\win32\\lib', 'C:\\ProgramData\\Anaconda3\\lib\\sit
e-packages\\Pythonwin', 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\IPython\\extensions', 'C:\\Users\\Mi
ssion Impossible\\.ipython']
```

In [ ]:

```
# Types of Built-in Modules
  1. Os module (to perform the basic operations provided by operating system)
  2. sys module
  3. Math Module
  4. Statistics
  5. Random Module
```

In [51]:

```
#os module:
import os
#os.mkdir("python problems")
#os.mkdir("Data Science")
# to know the current dir
os.getcwd()
#change the dir

#os.chdir(r"C:\Users\Mission Impossible\Desktop\Python_Online_Programmingg\Day-11\Data Science")
os.getcwd()
os.chdir(r"C:\\Users\\Mission Impossible\\Desktop\\Python_Online_Programmingg\\Day-11")
os.getcwd()
```

Out[51]:

```
'C:\\Users\\Mission Impossible\\Desktop\\Python_Online_Programmingg\\Day-11'
```

In [56]:

```
#to remove a directory
os.rmdir("python problems")
#os.rmdir("Data Science")
#to view the sub files in a directory
os.listdir("C:\\Users\\Mission Impossible\\Desktop\\Python_Online_Programmingg\\Day-11")
```

Out[56]:

```
['.ipynb_checkpoints',
 'calc.py',
 'Day- 11 (Regular Expressions).ipynb',
 'factorial1.py',
 'Practise.ipynb',
 'Python Programming.pdf',
 'RE Rules.ipynb',
 '__pycache__']
```

In [ ]:

```
#Random Module:
this module can be used to generate the numbers in a range:
#Functions in Random:
1. random.random()- its generates the random float number between 0.0 and 1.0
2. random.randint()- its generates the integer numbers in a given range
3. random.randrange()
4. random.choice()
5. random.shuffle()
```

In [74]:

```
#random.random()
import random
print(random.random())
print(random.random()*100)
print(random.random()*1000)
```

```
0.558395032899074
80.4468519084465
593.9514207619735
```

In [85]:

```
#random.randint():
import random
print(random.randint(1,100))
print(random.randint(1,1000))
```

```
30
647
```

In [111]:

```
#random.randrange():  randrange(start,stop,step)
import random
random.randrange(0,10,3)
random.randrange(0,101,10)
```

Out[111]:

```
10
```

In [125]:

```
#random.choice():
import random
print(random.choice("python programming"))
print(random.choice([10,20,30,50,47,425,783]))
```

```
m
20
```

In [136]:

```
#random.shuffle():
numbers = [2455,2,3,4,67,12,9695,3537,3664,53,241,1]
random.shuffle(numbers)
numbers
```

Out[136]:

```
[2, 241, 3664, 3, 4, 12, 3537, 53, 2455, 67, 1, 9695]
```

In [ ]:

```
#Math Module:
This module can be used to perfom all mathimatical functions.
```

In [137]:

```
import math
math.pi
```

Out[137]:

```
3.141592653589793
```

In [138]:

```
math.e
```

Out[138]:

```
2.718281828459045
```

In [144]:

```
#angke conversions:
print(math.radians(30))
print(math.radians(60))
print(math.degrees(1.0471975511965976))
print(math.degrees(3))
```

```
0.5235987755982988
1.0471975511965976
59.99999999999999
171.88733853924697
```

In [148]:

```
print(math.sin(0.5))
print(math.cos(2.5))
print(math.tan(0.9))
```

Out[148]:

```
1.2601582175503392
```

In [149]:

```
math.log(10)
```

Out[149]:

```
2.302585092994046
```

In [154]:

```
math.pow(2,5)
math.sqrt(2)
math.sqrt(64)
math.ceil(4.589900)  #gives the number to the smallest interger,greater than or equal to given number
math.floor(4.56690)  #gives the largesr integer less than or equal to given number
```

Out[154]:

```
4
```

In [ ]:

```
#Statistics Module:
This module can be used to perform
mean()
median()
mode()
stdev() functions.
```

In [160]:
```
import statistics
print(statistics.mean([2,3,4,5,6,7,8,9]))
print(statistics.median([2,3,4,5,6,7,8,9])) #its returns the middile value of numeric data in a list
print(statistics.mode([2,3,4,5,6,7,8,9,2,3,5,5,5,5,5]))  #its returns the most common data point in the list
print(statistics.stdev([2,3,4,5,6,7,8,9]))
```

```
5.5
5.5
5
2.449489742783178
```

In [ ]:
```
#User Defined Packages:
          Package is a collection of subpackages and Modules
```

In [ ]:
```
#syntax:
from package import module
```

In [188]:
```
from UserPackage import usermodule
usermodule.isPrime(int(input("enter a numner")))
```

```
enter a numner121
```

Out[188]:
```
True
```

In [21]:
```
from UserPackage import UserModule
UserModule.isprime(103)
```

Out[21]:
```
True
```

In [ ]:

In [ ]: