1. **#Program Fibonacci Number upto user choice**

**Ex1**

def feb():

a, b = (0,1)

while b <= 100:

print b,

a, b = b, a+b

feb()

**Ex2**

def feb1(a,b):

while b <= 200:

print b,

a, b = b, a+b

feb1(0,1)

**Ex 3**

def feb2 (a,b):

print "Fibonacci Series of given input is"

while b <= n:

print b

a, b = b, a+b

n = input("Enter number upto your choice")

feb2(0,1)

output:

Enter number upto your choice100

Fibonacci Series of given input is

1

1

2

3

5

8

13

21

34

55

89

# using recursion

Def f(n):

If n == 0:

Return 0

Elif n == 1:

Return 1

Else:

Return (f(n-1)+f(n-2))

Output

>>>

>>> f(10)

55

>>>

1. **#Proam to Check Email-ID is Valid or not using with Regular expression**

import smtplib

import re

def check\_email(email):

while not re.match('\w+[.|\w]\w+@\w+[.]\w+[.|\w+]\w+',email):

print " Please Enter Valid Email ID "

email = raw\_input(" Enter email: ")

print " Given mail is correct"

email = raw\_input(" Enter customer\_email: ")

check\_email(email)

**Output:**

Enter customer\_email: k

Please Enter Valid Email ID

Enter email: abc@gmail.com

Given mail is correct

1. **#Program to print Number of Occurences of an element in a list**

**Ex-1**

l = [1,2,3,5,2,1,3,7,6,5,7,7]

d={}

for x in l:

if x in d:

d[x] = d[x]+1

else:

d[x] = 1

print d

**Out put:**

{1: 2, 2: 2, 3: 2, 5: 2, 6: 1, 7: 3}

**Ex-2**

def oc\_l(l):

d= {}

for x in l:

if x in d:

d[x] = d[x]+1

else:

d[x] = 1

print d

l = input("Enter list of values with repeated numbers:")

oc\_l(l)

**Output**

Enter list of values with repeated numbers:[1,2,3,1,2,3,1,2,3]

{1: 3, 2: 3, 3: 3}

1. **#Python Script to parse index.xml File**

**Index.xml**

**<data>**

**<items>**

**<item name="item1"></item>**

**<item name="item2"></item>**

**<item name="item3"></item>**

**<item name="item4"></item>**

**</items>**

**</data>**

from xml.dom import minidom

xmldoc = minidom.parse('index.xml')

itemlist = xmldoc.getElementsByTagName('item')

#print len(itemlist)

#print itemlist[0].attributes['name'].value

for s in itemlist :

print s.attributes['name'].value

**OutPut:**

item1

item2

item3

item4

1. **#Program to using with join() method**

**Ex-1**

sq = "@"

l = "naveen"

j = (sq.join(l))

print "The oput of given sequence ",j

**Output:**

n@a@v@e@e@n

**ex-2**

def fun\_join(l):

un = "\_"

ls = (un.join(l))

print ls

l= raw\_input("Enter string:")

fun\_join(l)

**Output**

Enter String : naveen

n\_a\_v\_e\_e\_n

1. **#Program takes lise and finding avg**

def l\_avg(l):

avg = float(sum(l))/float(len(l))

print "Average of list", avg

l = [1,2,3,4,5,6]

l\_avg(l)

**OUTPUT**

3.5

1. **#Program to find Sum,Max,Min value in list\**

def sum\_max\_min(l):

s,ma,mi = sum(l),max(l),min(l)

print "Sum of list ", s

print "Maximum of list ",ma

print "Minimum of list", mi

l = [1,2,3,4,5,6]

sum\_max\_min(l)

**Output**

Sum of list 21

Maximum of list 6

Minimum of list 1

1. **#Program to get squre of the given sequence**

seq = range(8)

s = map(None,seq,map(lambda x : x\*x, seq))

print s

**Output**

[(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36), (7, 49)]

1. **#Program to sort dictonary based on Key and Value**

dict = {'a':1, 'b':3, 'd': 2, 'c':4}

key = sorted(dict.items(), key = lambda (k,v) : (k,v))

val = sorted(dict.items(), key = lambda (k,v) : (v,k))

print "Sorted based on Key's in Dictonary", key

print "Sorted based on Value's in Dictornary", val

**Output**

Sorted based on Key's in Dictonary [('a', 1), ('b', 3), ('c', 4), ('d', 2)]

Sorted based on Value's in Dictornary [('a', 1), ('d', 2), ('b', 3), ('c', 4)]

1. **#Program to Count the Word's in a fil**

with open("test.txt","w") as fp:

fp.write("Hai Naveen Kumar Welcome to Python World and Welcome to Django")

op = open("test.txt","r")

lines = op.readlines()

d={}

for i in lines:

x=i.split()

print x

for x in x:

if x in d:

d[x] = d[x]+1

else:

d[x] = 1

print "Count of the words in a file: ",d

**OutPut**

Count of the words in a file: {'and': 1, 'Hai': 1, 'Kumar': 1, 'Python': 1, 'Welcome': 2, 'Django': 1, 'to': 2, 'Naveen': 1, 'World': 1}

1. **#Program to create a generate objects and access elements in the objects without iterating over it**

def gener(n):

for i in n:

yield i

n = [1,2,3,4,5,6]

c = gener(n)

output:

>>> c.next()

1

>>>c.next()

2

1. **What are the different types of strings available in pythons? Write an example for each type.**

Python Supports three different types of strings.

Those are

1. Normal String
2. Unicode String
3. Raw String

Normal String : Which taken only strings.

Ex : ‘naveen’, “naveen”, “””hai naveen kumar”””

Unicode String: String starts with U,

Ex: U”naveen@kumar\_”

Raw String: String starts with R, The input will be any thing

Ex: R”r’^@[a-z, A-Z]”

1. **Does Python Supports Multiple Inheritance? If yes, Write an example to illustrate the same and Super Keyword?**

Python have limited support of multiple inheritance.

Ex:

#Program for multiple inheritance

class a():

def method1(self):

print "method1"

class b(a):

def method2(self):

print "method2"

class c(a,b):

def method3(self):

print "method3"

def method1(self):

print "method1 in c"

super(c,self).method1()

c = c()

c.method2()

c.method3()

c.method1()

1. **#Progrm to use listcomprehence and finding the odd prime numbers**

or

ls = [h for h in range(2,100) if h not in [ j for i in range(2,10) for j in range(2\*i,100,i)]]

print ls

**Output**

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

1. **What is a decorator and different types of decorators in python?**

A decorator is a template solution or already existing solution for a problem, we have two types of decorators and we call decorator by ‘@’ symbol. There are two types of decorators

1. Function Base
2. Class Base

Ex: To make a method as a static

@static method

Def fun():

State

State

Class decorator

@static method

Class test():

State..

State1..

1. **#program list as a stack (last in first out)**

def l():

list = [1,2,3]

list.append(5)

print list

list.pop()

print list

list.index(3)

print list

l()

**output:**

Actual list [1, 2, 3]

Adding One Element that is last in [1, 2, 3, 5]

Removing one element in list that is last in first our [1, 2, 3]

Index of the list [1, 2, 3]

1. **#Program to find factorial of given number**

def fac(num):

f = 1

while num>1:

f = f\*num

print f

num = num -1

print f

return num

fac(5)

Out put

5

20

60

120

120

**#using with recursive**

def fa(num):

while num<1:

return 1

else:

f = num\*fa(num-1) #recursive

print f

return f

fa(5)

Out Put:

1

2

6

24

120

1. **#program to find prime number**

def prim(num):

for x in range(1,num):

prime = True

for j in range(2,x):

if x % j == 0:

prime = False

if prime:

print x

prim(10)

output:

1

2

3

5

7

**def prim(num):**

**for x in range(1,num):**

**for j in range(2,x):**

**if x % j == 0:**

**break**

**else:**

**print x**

**prim(10)**

**output:**

**1**

**2**

**3**

**5**

**7**

**def prim(num):**

**for x in range(1,num):**

**if all (x % n != 0 for n in range(2,x)):**

**print x**

**prim(10)**

**Output:**

**1**

**2**

**3**

**5**

**7**

1. **# program to Count the words in a line and number of occurrences of words in lines and how many lines in a file**

fo = open("kumar.txt",'r')

lines = fo.readlines()

print "total lines",len(lines)

d = { }

for l in lines:

print "length of line",len(l)

x = l.split()

print "number of words in line",len(x)

for s in x:

if s in d:

d[s] = d[s]+1

else:

d[s] = 1

print "number of word occurrences in line",d

**"""**

**program for check dictonary values >=5 and**

**to print the same as dictonary to satisfy the**

**condition of >=5**

**input is d = {'a':[1,2,3,7,8,9],'b':[2,3,5,6,7]}**

**output is d = {'a':[7,8,9],'b':[5,6,7]}**

**"""**

**def di(d):**

**k = d.keys()**

**v = d.values()**

**l = v[0]**

**print l**

**l2 = v[1]**

**print l2**

**a = [x for x in l if x>=5]**

**b = [y for y in l2 if y>=5]**

**l =[a,b]**

**z = zip(k,l)**

**print dict(z)**

**d = {'a':[1,2,3,5,6],'b':[3,5,6,2]}**

**print d**

**di(d)**

**output**

**{'a': [1, 2, 3, 5, 6], 'b': [3, 5, 6, 2]}**

**[1, 2, 3, 5, 6]**

**[3, 5, 6, 2]**

**{'a': [5, 6], 'b': [5, 6]}**



**""" Program to print**

**EVEN\_ODD\_Prime**

**in a singe program**

**"""**

**def eve\_odd\_prim(n):**

**for x in range(n):**

**if x%2 == 0:**

**print x, "Even Number"**

**else:**

**print x, "Odd Number"**

**for y in range(2,x):**

**if x%y == 0:**

**break**

**else:**

**print x, "Prime number"**

**n = input("Enter some range of values:")**

**eop = eve\_odd\_prim(n)**

**output:**

**Enter some range of values:10**

**0 Even Number**

**0 Prime number**

**1 Odd Number**

**1 Prime number**

**2 Even Number**

**2 Prime number**

**3 Odd Number**

**3 Prime number**

**4 Even Number**

**5 Odd Number**

**5 Prime number**

**6 Even Number**

**7 Odd Number**

**7 Prime number**

**8 Even Number**

**9 Odd Number**

**>>>**



**"""**

**program to take two list and get in one dictionary as one list is key**

**and another is values**

**input: x = ['A','B','C','D'], y = ['P','Q','R','S']**

**create a function for above requirement**

**"""**

**def two\_list(x,y):**

**z = zip(x,y)**

**print dict(z)**

**x = input("Enter 1st list")**

**y = input("Enter 2nd list")**

**print x**

**print y**

**two\_list(x,y)**

**output:**

**Enter 1st list['A','B','C']**

**Enter 2nd list['P','Q','R']**

**['A', 'B', 'C']**

**['P', 'Q', 'R']**

**{'A': 'P', 'C': 'R', 'B': 'Q'}**

**>>>**



**"""**

**program to swap dictonary**

**as values, keys vise varse**

**"""**

**def swap(d):**

**k = d.keys()**

**v = d.values()**

**z = zip(v,k)**

**print dict(z)**

**z1 = zip(k,v)**

**print dict(z1)**

**d = input("Enter any list as a dictonar:")**

**swap(d)**

**output:**

**Enter any list as a dictonar:{'a':1,'b':2}**

**{1: 'a', 2: 'b'}**

**{'a': 1, 'b': 2}**



**"""**

**program for calculation**

**"""**

**class calc():**

**def \_\_init\_\_(self,a,b):**

**self.a = a**

**self.b = b**

**def add(self):**

**return self.a+self.b**

**def sub(self):**

**return self.a-self.b**

**def mul(self):**

**return self.a\*self.b**

**def div(self):**

**return self.a/self.b**

**a = input("enter a value:")**

**b = input("enter b value:")**

**c = calc(a,b)**

**print "the sum of two numbers",c.add()**

**print "The subtraction of two numbers",c.sub()**

**print "The multiplication of two numbers",c.mul()**

**print "The division of two numbers",c.div()**

**output:**

**enter a value:10**

**enter b value:20**

**the sum of two numbers 30**

**The subtraction of two numbers -10**

**The multiplication of two numbers 200**

**The division of two numbers 0**

1. **Map,Filter,Reduce**

def f(x): return x % 2 != 0 and x % 3 != 0

\

>>> filter(f, range(2, 25))

[5, 7, 11, 13, 17, 19, 23]

>>> def cube(x): return x\*x\*x

>>> map(cube, range(1, 11))

[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]

>>> def add(x,y): return x+y

>>> reduce(add, range(1, 11))

55

1. **MAP,FILTER, REDUCE EXAMPLES**

Def f\_m\_r(seq):

Def prim(x):

Return x % 2 !=0 and x % 3 !=0

Return filter(prim, seq)

Def cube(x):

Return x\*x\*x

Return map(cube, seq)

Def sum(x, y):

Return x + y

Return reduce(summ, seq)

Seq = range(1,10)

f\_m\_r(seq)

1. **Phone number check**

customer\_number = raw\_input(" Enter customer\_number(Mobile/Landline): ")

while not re.match(r'^\d{10}$',customer\_number):

print " Please Enter Valid Number "

customer\_number = raw\_input(" Enter customer\_number(Mobile/Landline): ")

1. Word searching in file

customer\_id = raw\_input(" Enter Customer ID: ")

with open("customer.txt", "r") as f:

lines = f.readlines()

for i in range(len(lines)):

x = lines[i].find(customer\_id)

if x == 0:

values = lines[i].split(',')

print " Customer Data \n customer\_id,customer\_name,customer\_address,customer\_email,customer\_number,customer\_created\_date\n",values

break

else:

print " Customer ID is not available "

1. How to create a directory using with python script

Import os

os.mkdir(“path”)

1. Multidimensional Array in python?

You're technically trying to index an uninitialized array. You have to first initialize the outer list with lists before adding items:

# Creates a list containing 5 lists initialized to 0

Matrix = [[0 for x in xrange(5)] for x in xrange(5)]

>>> matrix

[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]

You can now add items to the list:

Matrix[0][0] = 1

Matrix[4][0] = 5

print Matrix[0][0] # prints 1

print Matrix[4][0] # prints 5

ex 2:

a =[]

a.append([])

a[0].append('a1')

>>> a[0].append('a2')

>>> a

[['a1', 'a2']]

Ex3:

a =[[]] \* 3

>>> a[0].append('test')

>>> a

[['test'], ['test'], ['test']]

Ex4:

a = [[],[]]

>>> a[0].append('a1')

>>> a[1].append('a2')

>>> a[0].append('a2')

>>> a[1].append('a3')

>>> a

[['a1', 'a2'], ['a2', 'a3']]

Ex5:



"""

I given input like l = '1 2 3 4 5 6 7 8 9 10 11 12 13 14'

and I need output like ['hp', 'hp', 'hp', 4, 'hp', 6, 'hp', 8, 9, 10, 'hp', 12, 'hp', 14]

"""

l = '1 2 3 4 5 6 7 8 9 10 11 12 13 14'

print l

l2 = l.split()

print l2

l3 =[]

for x in l2:

l3.append(int(x))

print l3

l4 =[]

for i in l3:

for j in range(2,i):

if i%j == 0:

l4.append(i)

break

else:

a = 'hp'

l4.append(a)

print '\n',l4

output:

1 2 3 4 5 6 7 8 9 10 11 12 13 14

['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14']

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

Final output : **['hp', 'hp', 'hp', 4, 'hp', 6, 'hp', 8, 9, 10, 'hp', 12, 'hp', 14]**

**## another approach**

a = '1 2 3 4 5 6 7 8 9 10'

a1 = []

for a2 in range(1,len(a.split())+1):

a3 = int(a2)

for a4 in range(2,a2):

if a2 % a4 == 0:

a1.append(a2)

break

else:

z='hi'

a1.append(z)

print a1

output:

**['hi', 'hi', 'hi', 4, 'hi', 6, 'hi', 8, 9, 10]**



**"""**

**I have input like d = {(1,2) : {'a':{'b':1}}, (3,4):{'a':{'b':6}}, (5,6):{'a':{'b':1}}}**

**i need output like {1: [(1, 2), (5, 6)], 6: [(3, 4)]}**

**write program for above output**

**"""**

**d = {(1,2) : {'a':{'b':1}}, (3,4):{'a':{'b':6}}, (5,6):{'a':{'b':1}}}**

**key = d.keys()**

**d2 = {}**

**for x in range(len(key)):**

**print 'keys index = ',x**

**print 'keys index will be converted to values index, Then each key value=',d[key[x]]**

**print 'values of above dictonary=',d[key[x]].get('a',0)**

**print 'once again dividing above dictonary with values, and the values = ',d[key[x]].get('a',0).get('b',0)**

**v = d[key[x]].get('a',0).get('b',0)**

**if v !=0:**

**if d2.get(v,0) == 0:**

**d2[v]=[key[x]]**

**else:**

**d2[v] = d2[v] + [key[x]]**

**print d2**



**"""**

**write a program to updateing keys and changing keys to values values to keys**

**"""**

**d = {'a':1,'b':2,'d':3}**

**d['c']=d.pop('d')**

**print 'in dictonary d is replaced with c',d**

**d2 = {y:x for x,y in d.iteritems()}**

**print 'keys replaced with values ', d2**

**d3 = {x : y for x,y in d.iteritems()}**

**print 'values replaced with keys', d3**

**output:**

**in dictonary d is replaced with c {'a': 1, 'c': 3, 'b': 2}**

**keys replaced with values {1: 'a', 2: 'b', 3: 'c'}**

**values replaced with keys {'a': 1, 'c': 3, 'b': 2}**



**""" python program for sorting the list"""**

**data\_list = [-5, -23, 5, 0, 23, -6, 23, 67]**

**new\_list = []**

**while data\_list:**

**minimum = data\_list[0] # arbitrary number in list**

**for x in data\_list:**

**if x < minimum:**

**minimum = x**

**new\_list.append(minimum)**

**data\_list.remove(minimum)**

**print new\_list**

**[-23, -6, -5, 0, 5, 23, 23, 67]**



**""" list reverse"""**

**def reverse(data\_list):**

**length = len(data\_list)**

**s = length**

**new\_list = [None]\*length**

**for item in data\_list:**

**s = s - 1**

**new\_list[s] = item**

**return new\_list**



**""" write a python program for finding the file in a system"""**

**import os**

**from os.path import join**

**lookfor = "findfile.py"**

**for root, dirs, files in os.walk('D:\Python27'):**

**print "searching", root**

**if lookfor in files:**

**print "found: %s" % join(root, lookfor)**

**break**

**output:**

**searching D:\Python27\tcl\tcl8.5\tzdata\Atlantic**

**searching D:\Python27\tcl\tcl8.5\tzdata\Australia**

**searching D:\Python27\tcl\tcl8.5\tzdata\Brazil**

**searching D:\Python27\tcl\tcl8.5\tzdata\Canada**

**searching D:\Python27\tcl\tcl8.5\tzdata\Chile**

**searching D:\Python27\tcl\tcl8.5\tzdata\Etc**

**searching D:\Python27\tcl\tcl8.5\tzdata\Europe**

**searching D:\Python27\tcl\tcl8.5\tzdata\Indian**

**searching D:\Python27\tcl\tcl8.5\tzdata\Mexico**

**searching D:\Python27\tcl\tcl8.5\tzdata\Pacific**

**searching D:\Python27\tcl\tcl8.5\tzdata\SystemV**

**searching D:\Python27\tcl\tcl8.5\tzdata\US**

**searching D:\Python27\tcl\tix8.4.3**

**searching D:\Python27\tcl\tix8.4.3\bitmaps**

**searching D:\Python27\tcl\tix8.4.3\demos**

**searching D:\Python27\tcl\tix8.4.3\demos\bitmaps**

**searching D:\Python27\tcl\tix8.4.3\demos\samples**

**searching D:\Python27\tcl\tix8.4.3\pref**

**searching D:\Python27\tcl\tk8.5**

**searching D:\Python27\tcl\tk8.5\demos**

**searching D:\Python27\tcl\tk8.5\demos\images**

**searching D:\Python27\tcl\tk8.5\images**

**searching D:\Python27\tcl\tk8.5\msgs**

**searching D:\Python27\tcl\tk8.5\ttk**

**searching D:\Python27\test**

**found: D:\Python27\test\findfile.py**



**"""**

**For given sequence write a program to update string with 'india' where ever the**

**number divisible by 5**

**"""**

**l = '1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 25'**

**l2 = []**

**for i in range(1, len(l.split())+1):**

**j = int(i)**

**if j % 5 == 0:**

**s = 'india'**

**l2.append(s)**

**else:**

**l2.append(str(j))**

**l3 = ' '.join(l2)**

**print l3**

**output:**

**1 2 3 4 india 6 7 8 9 india 11 12 13 14 india 16 17 18 19 india 21 22 23 24 india 26 27 28 29 india 31**

1. **Program to sent mail**

**import smtplib**

**import re**

**class email\_alert():**

**def email(self):**

**from\_addr = 'nvk.grandhi@gmail.com'**

**to\_addrs = raw\_input("Enter email ID...")**

**while not re.match('\w+[.|\w]\w+@\w+[.]\w+[.|\w+]\w+',to\_addrs):**

**print "please enter valid email:"**

**to\_addrs=raw\_input("Enter Email ID:")**

**print "Sending mail in Progress Please wait for few seconds"**

**subject = 'NATIONAL LIBRARY'**

**text="this is a warning message from National library ::your cross the due date plz return the book"**

**message = 'Subject: %s\n\n%s' % (subject, text)**

**username = 'nvk.grandhi@gmail.com'**

**password = 'neevankumar'**

**server = smtplib.SMTP('smtp.gmail.com:587')**

**server.starttls()**

**server.login(username,password)**

**server.sendmail(from\_addr,to\_addrs, message)**

**server.quit()**

**print "email sended successfully"**

**em = email\_alert()**

**em.email()**

1. **Python program to open the default web browser**

**import webbrowser**

**new = 2 “””open in a new tab, if possible”””**

**“””open a public url, in this case, the webbrowser docs”””**

**url =** [**http://docs.python.org/library/webbroswer.html**](http://docs.python.org/library/webbroswer.html)

**webbrowser.open(url, new=new)**

**# below code is optional**

**“”” to open the html file on my own windows computer**

**url =”””** [**file://X://MiscDev/language\_links.html**](file://X://MiscDev/language_links.html)

**webbrowser.open(utl, new=new)**

**“””**

**Example two**

import webbrowser

url = '<http://www.python.org/>'

webbrowser.open\_new\_tab(url + 'tutorial/')

webbrowser.open\_new(url)

import webbrowser

print('OPENING FIREFOX...')

handle = webbrowser.get()

handle.open('[http://gmail.com](http://gmail.com/)')

handle.open\_new\_tab('[http://google.com](http://google.com/)')

1. **Adding multiple elements in list dynamically**

**def ade(a, l =[]):**

**l.append(a)**

**print l**

**output:**

**>>>**

**>>>**

**[1]**

**>>>**

**[1]**

**>>>[1.2.3.4]**

**[1.2.3.4]**

1. **Dividing list into sub lists**

**l = [1,2,3,4,5,6,7,8]**

**result = {}**

**idx = 0**

**for i in l:**

**group = idx/3**

**if group not in result:**

**result[group] = []**

**result[group].append(i)**

**idx += 1**

**result.values()**

**print result**

**output:**

**>>>**

**{0: [1,2,3], 1:[4,5,6]. 2:[7,8]}**

**l = [1,2,3,4,5,6,7,8]**

**for l1 in l[:2]:**

**print l1**

**## another approach**

l = [1,2,3,4,5,6,7,8,9,10]

>>> re = []

>>> for i in range(0, len(l), 3):

re.append(l[i:i+3])

>>> print re

[[1, 2, 3], [4, 5, 6], [7, 8, 9], [10]]

**##another**

 z = []

>>> for l3 in l2:

for x in l3:

if x % 5 == 0:

 z.append(x)

**##some other approaches**

l = range(10)

>>> l2 = [l[i:i+4] for i in range(0, len(l), 4)]

>>> l2

[[0, 1, 2, 3], [4, 5, 6, 7], [8, 9]]

>>> l2 = [l[i:i+5] for i in range(0, len(l), 5)]

>>> l2

[[0, 1, 2, 3, 4], [5, 6, 7, 8, 9]]

>>> for l3 in l2:

print l3

[0, 1, 2, 3, 4]

[5, 6, 7, 8, 9]

l = range(100+1)

>>> r = l[i:i+3]

>>> r

[6, 7, 8]

>>> re = [l[i:i+3] for i in range(0, len(l), 3)]

>>> re

[[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], [36, 37, 38], [39, 40, 41], [42, 43, 44], [45, 46, 47], [48, 49, 50], [51, 52, 53], [54, 55, 56], [57, 58, 59], [60, 61, 62], [63, 64, 65], [66, 67, 68], [69, 70, 71], [72, 73, 74], [75, 76, 77], [78, 79, 80], [81, 82, 83], [84, 85, 86], [87, 88, 89], [90, 91, 92], [93, 94, 95], [96, 97, 98], [99, 100]]

1. **#regular expression for find and replace or substitute special character in a file**

**import re**

**#create redata.txt file when executing the program and fill with some text and in between text add some like ‘----‘, special characters inside the file**

**f = open('redata.txt','r') #Open File**

**lines = f.readlines()**

**s = []**

**for line in lines:**

**string = re.findall(r'---',line)**

**print string**

**sub\_str = re.sub(r'---','', line)**

**print sub\_str**

**st = sub\_str.replace(" ","")**

**s.append(st)**

**print s**

1. **#Generator example**

**# generator**

**def gen(n):**

**for i in n:**

**yield(i)**

**n = input("Enter any sequence in list:")**

**c = gen(n)**

**output:**

**c.next()**

**1**

**c.next()**

**2**

1. **#How to find the length of string without using any built in function**

**l = 'naveen'**

**l1 = 0**

**for x in l:**

**l1+=1**

**le = l1**

**print le**

1. **palindrome without using built-in**

word = str(raw\_input('Is it palindromic?\n'))

def is\_palindrome(word):

   if len(word) <= 2 and word[0] == word[-1]:

       print 'True'

   elif word[0] == word[-1]:

       is\_palindrome(word[1:-1])

   else:

       print 'False'

is\_palindrome(word)