**https://www.youtube.com/watch?v=pJ3IPRqiD2M**

**str1="Hi"**

**str2="How are you ?"**

**print(str1+ "\n" +str2);**

**a=1**

**b=2**

**c=a+b**

**print("The sum(a,b) is " +str(c))**

**a=input()**

**b=input()**

**print("The sum is " +str(int(a)+int(b)))**

**(or)**

**a=int(input())**

**b=int(input())**

**print("The sum is " +str(a+b))**

**Adding 2 numbers in a single line:**

**sum=int(input())+int(input())**

**print(sum) (or)**

**print(int(input())+int(input()))**

**You can reinitialise the variable values:**

**a=1**

**b=2**

**c=a+b**

**print(c)**

**a=2**

**b=3**

**c=a+b**

**print(c)**

**o/p**

**3**

**5**

**num1 = 15**

**num2 = 12**

**# Adding two nos**

**sum = num1 + num2**

**# printing values**

**print("Sum of {0} and {1} is {2}".format(num1, num2, sum))**

**number1 = int(input("First number: "))**

**number2 = int(input("Second number: "))**

**# Adding two numbers**

**# User might also enter float numbers**

**sum = (number1) + (number2)**

**# Display the sum**

**# will print value in float**

**print("The sum of {0} and {1} is {2}".format(number1, number2, sum))**

**FIBONACCI:**

**def fib(n):**

**a=0;**

**b=1;**

**if n<0:**

**print("Incorrect input")**

**elif n==0:**

**return a;**

**elif n==1:**

**return b;**

**else:**

**for i in range(2,n):**

**c=a+b**

**a=b**

**b=c**

**return b**

**print(fib(0))**

**(or)**

**def fib(n):**

**if n<0:**

**print("Incorrect input");**

**elif n==1:**

**return 0;**

**elif n==2:**

**return 1;**

**else:**

**return fib(n-1)+fib(n-2)**

**print(fib(9))**

**Lists: (collection of dissimilar items)**

**Lists retain the inserted order**

**Lists are mutable datatypes**

**that is , you can change the values using index.**

**They use stack operation.**

**Strings:**

**Same as lists ,but strings are immutable.**

**But with additional functions like replace you can modify .**

**Strings are iterable.**

**ie, it can be used inside a loop.**

**list1=['a','c','c','e','n','t','u','r','e']**

**eg,**

**for character in list1:**

**print(character)**

**o/p:**

**a**

**c**

**e**

**n**

**t**

**u**

**r**

**e**

**list1.pop()**

**list1.extend(list2)**

**diff btw insert and append:**

**insert-> it will ask for the position.**

**same way for string .**

**Integers cant be iterable**

**number=12345**

**for c in number:**

**print(c)**

**o/p: error ,bcz int is not iterable**

**so for that,**

**number=12345**

**for c in str(number):**

**print(c)**

**o/p 12345**

**Tuple:**

**Same like list, but immutable**

**tup1=(1,2,3)**

**Dictionary:**

**It is a key value pair.**

**Key is an identifier**

**eg, student inf**

**dict1={"name":"hussain","age":25}**

**get func:**

**dict1.get('name')**

**keys func:**

**dict1.keys()**

**value func**

**dict1.values()**

**appending cannot be done in dictionary**

**so dict1.append(dict2)**

**Print each character of accenture 2 times:**

**string1="ACCENTURE"**

**list1=[]**

**for i in string1:**

**list1=list1+list(2\*i)**

**print(list1)**

**['A', 'A', 'C', 'C', 'C', 'C', 'E', 'E', 'N', 'N', 'T', 'T', 'U', 'U', 'R', 'R', 'E', 'E']**

**You can use extend funtion also inplace of '+'**

**(or)**

**string1="accenture"**

**for i in string1:**

**print(i\*2)**

**s="ACCENTURE"**

**li=[]**

**for i in s:**

**n=i\*2**

**li.append(n)**

**print("".join(li))**

**string1="ACCENTURE"**

**list1=[]**

**for i in string1:**

**if i not in list1[]**

**list1=list1+list(2\*i)**

**print(list1)**

**['A', 'A', 'C', 'C', 'E', 'E', 'N', 'N', 'T', 'T', 'U', 'U', 'R', 'R']**

**1.**

**Write a program that examines three variables?x, y, and z? and prints the largest odd number among them.**

**If none of them are odd, it should print a message to that effect.**

**x=int(input("Enter 1st number"))**

**y=int(input("Enter 2nd number"))**

**z=int(input("Enter 3rd number"))**

**list1=[]**

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

**list1.append(x)**

**if y%2!=0:**

**list1.append(y)**

**if z%2!=0:**

**list1.append(z)**

**largest=0**

**for i in list1:**

**if i>largest:**

**largest=i**

**if largest>0:**

**print(largest)**

**else:**

**print("not an odd number")**

**li=[]**

**x=int(input("First no"))**

**y=int(input("second no"))**

**z=int(input("third no"))**

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

**li.append(x)**

**if y%2!=0:**

**li.append(y)**

**if z%2!=0:**

**li.append(z)**

**print(li)**

**print(max(li))**

**output:**

**------**

**Enter 1st number2**

**Enter 2nd number4**

**Enter 3rd number6**

**not an odd number**

**Enter 1st number 2**

**Enter 2nd number 5**

**Enter 3rd number 3**

**5**

**2.**

**def right\_justify(s):**

**print("%70s" %s)**

**right\_justify('cigna')**

**output:**

**------**

**cigna**

**3.**

**#Write a program that asks the user to input 10 integers, and then prints the largest odd number that was entered.**

**# If no odd number was entered, it should print a message to that effect.**

**numlist=[]**

**largest=0**

**n = int(input("Enter number of elements"))**

**# take list as input from user and put it in an empty list**

**for i in range(0,n):**

**element=int(input())**

**numlist.append(element)**

**# to traverse through the list and find the largest among them**

**for j in numlist:**

**if j%2!=0 and j>largest: # give largest as 0 initially**

**largest=j # if j is odd and largest when traversed using for loop - j will be largest**

**if largest==0 : #if thr is no odd number then largest will be 0 only - which makes it a list of even numbers - so print no odd numbers - else just print the largest**

**print("there were no odd numbers")**

**else:**

**print(largest)**

**OUTPUT:**

**-------**

**Enter number of elements 10**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**1**

**`**

**9**

**6.**

**#Let s be a string that contains a sequence of decimal numbers separated by commas, e.g., s = '1.23,2.4,3.123'.**

**# Write a program that prints the sum of the numbers in s.**

**str1=input("enter a string of numbers separated by comas")**

**nums=[float(i) for i in str1.split(",")]**

**# creates a list of floats spilt by**

**print(sum(nums)) # sum() finds the sum of the elements in the list**

**s = '1.23,2.4,3.123'**

**var=[]**

**num=s.split(",")**

**for i in num:**

**var.append(float(i))**

**print(var)**

**print(sum(var))**

**print(max(var))**

**OUTPUT:**

**-------**

**enter a string of numbers separated by comas1.23,3.45,5.67**

**10.35**

**'''for i in range(1,11):**

**for j in range(1,i):**

**print('\*', end=" ")**

**print("\n")'''**

**======**

**'''for i in range(1,11):**

**for j in range(1,i):**

**print('\*', end=" ")**

**print("\n")'''**

**======**

**'''for i in range(1,6):**

**for j in range(1,6):**

**print('\*', end=" ")**

**print("\n")'''**

**====**

**n= int(input("Enter the number"))**

**for i in range(1,n):**

**for j in range(1,i+1):**

**print(j,end=" ")**

**print()**

**=====**

**'''i=1**

**while i<=5:**

**print(i)**

**i=i+1'''**

**=====**

**'''for i in range(1,6):**

**if i == 4:**

**break;**

**print(i)**

**print("You have reached the limit !")**

**print("Bye!!")'''**

**=====**

**'''for i in range(1,6):**

**if i == 4:**

**continue;**

**print(i)**

**print("You have reached the limit !")**

**print("Bye!!")'''**

**===**

**def sum(a,b):**

**c=a+b**

**return c**

**====\*\*\*\*\***

**def total():**

**res = sum(1, 3)**

**if res > 0:**

**print("Best of luck")**

**else:**

**print("Worst of luck")**

**total()**

**==**

**def sum\_samp(a,b):**

**c=a+b**

**d=a-b**

**return a,b,c,d**

**temp1,temp2,temp3,temp4=sum\_samp(1,2)**

**print(temp1)**

**print(temp2)**

**print(temp3)**

**print(temp4)**

**=====**

**str1="Nivetha"**

**li=[]**

**dict1={}**

**cnt=0**

**for i in str1:**

**if i in dict1:**

**dict1[i]=dict1[i]+1**

**else:**

**dict1[i]=1**

**print(dict1)**

**print(dict1['N'])**

**======**

**if j in dict1:**

**dict1[j] = dict1[j] + 1**

**else:**

**dict1[j]=1**

**print(dict1)**

**=>**

**ACCENTURE**

**{'A': 1, 'C': 2, 'E': 2, 'N': 1, 'T': 1, 'U': 1, 'R': 1}**

**A1C2E2N1T1U1R1**

**'''**

**#print the odd sum and even sum**

**a=int(input())**

**b=int(input())**

**odd\_sum=0**

**even\_sum=0**

**for i in range(a,b):**

**if(i % 2 == 0):**

**even\_sum=even\_sum + i**

**else:**

**odd\_sum=odd\_sum + i**

**print("EVEN: "+str(even\_sum))**

**print("ODD:" +str(odd\_sum))**

**'''**

**'''# print it in vertical form**

**a="Welcome"**

**for i in range():**

**print(i)'''**

**'''#print the length of the string**

**a="Novin"**

**cnt=0**

**for i in a:**

**cnt=cnt+1**

**print(cnt)**

**'''**

**'''#print the count of occurance of character**

**a="Accenture"**

**cnt=0**

**for i in a:**

**#cnt = 0**

**if(i=='e'):**

**cnt=cnt+1**

**print(cnt)'''**

**'''#insert range into a list**

**c=[]**

**for i in range(1,11):**

**if(i%2==0):**

**c.append(i)**

**print(c)'''**

**'''#count the digits and characters**

**lis=[1,2,'N','O',3]**

**char\_cnt=0**

**int\_cnt=0**

**for i in lis:**

**if(type(i)==str):**

**char\_cnt=char\_cnt+1**

**else:**

**int\_cnt=int\_cnt+1**

**print("Character: "+str(char\_cnt))**

**print("Integer: "+str(int\_cnt))'''**

**a = ['five', 5, 'six', 6, 'seven', 7, 'seven', 7, 'nine', 9]**

**dict1 = {}**

**for i in range(0, len(a), 2):**

**# print(a[i+1])**

**if a[i] in dict1:**

**l = dict1.get(a[i])**

**dict1[a[i]] = [l, a[i + 1]]**

**else:**

**dict1[a[i]] = a[i + 1]**

**print(str(dict1))**

**#string concatenation**

**----------------------**

**str1='hi'**

**str2='karishma'**

**print(str1+'\n'+str2)**

**#input statements**

**------------------**

**1)**

**a=int(input())**

**b=int(input())**

**c=a+b**

**print("the sum is"+ str(c))**

**input:2**

**3**

**output:5**

**2)**

**a=input()**

**b=input()**

**c=a+b**

**print("the sum is"+ str(c))**

**input:2**

**3**

**output:23**

**#to find sum with a single statement**

**-------------------------------------**

**print(int(input())+int(input()))**

**2**

**3**

**5**

**#integer is not iteratable - so we use type conversion - so we use str(number)- but if we want to print as a number give int(i)**

**--------------------------------------------------------------------------------------------------------------------------------**

**number=12345**

**for i in str(number):**

**print(int(i))**

**a=[1,2,3]**

**b=[3,4,5]**

**a.append(b) #it puts the elements into the list**

**print(a)**

**[1, 2, 3, [3, 4, 5]]**

**c=[1,2,3]**

**d=[3,4,5]**

**c.extend(d) # this will expand the list**

**print(c)**

**[1, 2, 3, 3, 4, 5]**

**#convert a string into a list - then print as a a c c c c like that - then remove duplicates**

**---------------------------------------------------------------------------------------------**

**str1='Accenture'**

**list1=[]**

**for i in str1:**

**list1.append(i)**

**list1.append(i)**

**print(list1)**

**['A', 'A', 'c', 'c', 'c', 'c', 'e', 'e', 'n', 'n', 't', 't', 'u', 'u', 'r', 'r', 'e', 'e']**

**list1=list(dict.fromkeys(list1)) # to remove duplicates**

**print(list1)**

**['A', 'c', 'e', 'n', 't', 'u', 'r']**

**# to remove one pair of duplicates**

**-----------------------------------**

**str1='Accenture'**

**list1=[]**

**for i in str1:**

**if i not in list1:**

**list1.append(i)**

**list1.append(i)**

**print(list1)**

**OUTPUT:**

**['A', 'A', 'c', 'c', 'e', 'e', 'n', 'n', 't', 't', 'u', 'u', 'r', 'r']**

**OR**

**str1='Accenture'**

**list1=[]**

**for i in str1:**

**if i not in list1:**

**list1.extend(i\*2)**

**print(list1)**

**OUTPUT:**

**['A', 'A', 'c', 'c', 'e', 'e', 'n', 'n', 't', 't', 'u', 'u', 'r', 'r']**

**#for loop**

**-----------**

**a=int(input("enter start number"))**

**b=int(input("enter end number"))**

**for i in range(a,b):**

**print(i)**

**output:**

**enter start number 1**

**enter end number 5**

**1**

**2**

**3**

**4**

**#sum of even numbers and odd numbers from 1 to 10**

**---------------------------------------------------**

**sumodd=0**

**sumeve=0**

**for i in range(1,11):**

**if i%2==0:**

**sumeve=sumeve+i**

**else:**

**sumodd=sumodd+i**

**print("sum of even numbers",sumeve)**

**print("sum of odd numbers",sumodd)**

**output:**

**sum of even numbers 30**

**sum of odd numbers 25**

**#strings are immutable**

**-----------------------**

**str1='hello'**

**print(str1[2])**

**str1[2]='o'**

**print(str1[2])**

**output:**

**str1[2]='o'**

**TypeError: 'str' object does not support item assignment**

**#print length of string**

**--------------------------**

**str1='hello'**

**count=0 #same can be done for list**

**for i in str1:**

**count=count+1**

**print(count)**

**#occurence of e**

**----------------**

**str1='ACCENTURE'**

**count=0**

**for i in str1:**

**if i=='E':**

**count=count+1**

**print(count)**

**output:**

**2**

**# the above pgm with modification**

**----------------------------------**

**for i in str1:**

**count=0 # here count is given in for loop - so again and again count is initialised to 0**

**if i=='E':**

**count=count+1**

**print(count)**

**#create a list runtime with even numbers less than 10**

**------------------------------------------------------**

**list1=[]**

**for i in range(0,11,2):**

**list1.append(i)**

**print(list1)**

**output:**

**[0,2,4,6,8]**

**OR**

**list1=[]**

**for i in range(1,11):**

**if i%2==0:**

**list1.append(i)**

**print(list1)**

**#number of characters and digits in a list**

**--------------------------------------------**

**list1=[1,'a',2,'b']**

**intcnt=0**

**charcnt=0**

**for i in list1:**

**if type(i)==int:**

**intcnt=intcnt+1**

**else:**

**charcnt=charcnt+1**

**print("int count",intcnt)**

**print("char count",charcnt)**

**output:**

**2**

**2**

**#to find number of characters and digits [here all elements are strings]**

**--------------------------------------------------------------------------**

**charcnt=0**

**digicnt=0**

**list1=['1','a','2','b','A']**

**for i in list1:**

**if ord(i)>=65 and ord(i)<=122:**

**charcnt=charcnt+1**

**else:**

**digicnt=digicnt+1**

**print(charcnt)**

**print(digicnt)**

**output:**

**3**

**2**

**LIST TO DICT:**

**--------------**

**list1=['five', 5, 'six', 6, 'seven', 7, 'seven', 7, 'nine', 9]**

**keys=[]**

**values=[]**

**dictnew={}**

**for i in list1:**

**if type(i)==str:**

**keys.append(i)**

**else:**

**values.append(i)**

**print(keys)**

**print(values)**

**for k in range (0,len(values)):**

**dictnew[keys[k]]=values[k]**

**print(dictnew)**

**output:**

**['five', 'six', 'seven', 'seven', 'nine']**

**[5, 6, 7, 7, 9]**

**{'five': 5, 'six': 6, 'seven': 7, 'nine': 9}**

**OR**

**list1=['five', 5, 'six', 6, 'seven', 7, 'seven', 7, 'nine', 9]**

**keys=[]**

**values=[]**

**dictnew={}**

**for i in list1:**

**if type(i)==str:**

**keys.append(i)**

**else:**

**values.append(i)**

**mn=zip(keys,values)**

**dictnew=dict(mn)**

**print(dictnew)**

**OR**

**list1=['five', 5, 'six', 6, 'seven', 7, 'seven', 7, 'nine', 9]**

**dictnew={}**

**keys=[]**

**values=[]**

**dictnew={}**

**for i in list1:**

**if type(i)==str:**

**keys.append(i)**

**else:**

**values.append(i)**

**print(keys)**

**print(values)**

**for j in range(0,len(keys)):**

**if keys[j] not in dictnew:**

**dictnew[keys[j]]=values[j]**

**else:**

**lis=[dictnew[keys[j]],values[j]]**

**dictnew[keys[j]]=lis**

**print(dictnew)**

**#print similar values for the same key**

**---------------------------------------**

**list1=['five', 5, 'six', 6, 'seven', 7, 'seven', 7, 'nine', 9]**

**dictnew={}**

**for k in range(0,len(list1),2):**

**if list1[k] in dictnew:**

**value=dictnew.get(list1[k])**

**dictnew[list1[k]]=[value,list1[k+1]]**

**else:**

**dictnew[list1[k]]=list1[k+1]**

**print(dictnew)**

**#take a list as input from user**

**---------------------------------**

**numlist=[]**

**n = int(input("Enter a number"))**

**for i in range(0,n):**

**element=int(input())**

**numlist.append(element)**

**print(numlist)**

**output:**

**------**

**Enter a number 10**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**1**

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

**#convert a STRING of decimal into a LIST and find sum**

**-------------------------------------------------------**

**str1=input("enter a string of numbers separated by comas")**

**nums=[float(i) for i in str1.split(",")] # creates a list of floats spilt by ,**

**print(nums)**

**output for print(nums):**

**------------------------**

**#enter a string of numbers separated by comas1.23,3.4**

**#[1.23, 3.4]**

**print(sum(nums)) # sum() finds the sum of the elements in the list**

**output:**

**------**

**enter a string of numbers separated by comas1.23,3.45,5.67**

**10.35**

**#PATTERNS:**

**-----------**

**for i in range(1,5):**

**for j in range(1,i):**

**print(j,end=" ")**

**print("\n")**

**OUTPUT:**

**------**

**1**

**1 2**

**1 2 3**

**#PATTERNS:**

**----------**

**for i in range(1,5):**

**for j in range(1,i):**

**print('\*',end=" ")**

**print("\n")**

**OUTPUT:**

**-------**

**\***

**\* \***

**\* \* \***

**#PATTERNS:**

**-----------**

**for i in range(1,5):**

**for j in range(1,5):**

**print('\*',end=" ")**

**print("\n")**

**OUTPUT:**

**------**

**\* \* \***

**\* \* \***

**\* \* \***

**WHILE LOOP:**

**------------**

**i=1**

**while(i<5):**

**print(i)**

**i=i+1 # IF THIS ISNT GIVEN - INFINITE LOOP WILL COME AS OUTPUT**

**#print 1 2 bye [USING BREAK STATEMNET]**

**----------------------------------------**

**for i in range(1,6):**

**if i==3:**

**break**

**print(i)**

**print("bye")**

**OR**

**for i in range(1,6):**

**if i==3:**

**print("you have reached the limit") # use case: enter passwrd wrong more than thrice - send error msg**

**print("bye")**

**break**

**else:**

**print(i)**

**OR**

**for i in range(1,6):**

**if i!=3:**

**print(i)**

**elif i == 3:**

**print("you have reached the limit")**

**print("bye")**

**break**

**output:**

**------**

**1**

**2**

**you have reached the limit**

**bye**

**#CONTINUE STATEMENT [SKIP ONE NUMBER AND PRINT THE REST]**

**---------------------------------------------------------**

**for i in range(1,6):**

**if i==3:**

**continue**

**print(i)**

**print("you have reached the limit")**

**print("bye")**

**OUTPUT:**

**------**

**1**

**2**

**4**

**5**

**you have reached the limit**

**bye**

**#FUNCTION:[WITHOUT ARGUMENTS]**

**-------------------------------**

**def sayhello():**

**print("hello")**

**sayhello()**

**OUTPUT:**

**------**

**hello**

**#FUNCTION:[WITH ARGUMENTS]**

**-------------------------------**

**def printint(a):**

**print("my value is",a)**

**printint(3)**

**OUTPUT:**

**------**

**3**

**#FUNCTION: [HERE WE RETURN THE VALUE]**

**---------------------------------------**

**def status(a,b):**

**sum1=a+b**

**return sum1**

**sum1=status(-1,-2)**

**if sum1<0:**

**print("fail")**

**else:**

**print("all the best")**

**OUTPUT:**

**------**

**FAIL**

**#FUNCTION:[WE RETURN TWO ARGUMENTS]**

**-------------------------------------**

**def sum(a,b):**

**c=a+b**

**return a,b**

**c,d=sum(1,2)**

**print(c)**

**print(d)**

**OUTPUT:**

**------**

**1**

**2**

**#FUNCTION:[function inside function]**

**--------------------------------------**

**def sum(a,b):**

**c=a+b**

**return c**

**def total():**

**tot=0**

**c=sum(5,5)**

**if c==10:**

**tot=c+1**

**print(tot)**

**total()**

**AND**

**def sum(a,b):**

**c=a+b**

**return a,b,c**

**f,d,e=sum(1,2)**

**print(f)**

**print(d)**

**print(e)**

**output:**

**-------**

**1**

**2**

**3**

**AND**

**def sum(a,b=2):**

**c=a+b**

**return a,b,c**

**f,d,e=sum(1,)**

**print(f)**

**print(d)**

**print(e)**

**OUTPUT:**

**------**

**1**

**2**

**3**

**#STRING to DICTIONARY [#input:string from user - output:count the charcetrs and print in dictionary]**

**------------------------------------------------------------------------------------------------------**

**string=input("Enter a string")**

**strdict={}**

**for i in string:**

**if i in strdict:**

**strdict[i]=strdict[i]+1 # in a dict u can update by giving dictname[value]=key - this will update it - if tht is nt present it will craete a new pair**

**else:**

**strdict[i]=1 # by giving this it will automatically create a dictionary with i as values and 1 as key**

**print(strdict)**

**OUTPUT:**

**-------**

**Enter a stringACCENTURE**

**{'A': 1, 'C': 2, 'E': 2, 'N': 1, 'T': 1, 'U': 1, 'R': 1}**

**#input:string from user - output:count the charcetrs and print in dictionary**

**string=input("Enter a string")**

**strdict={}**

**for i in string:**

**if i in strdict:**

**strdict[i]=strdict[i]+1 # in a dict u can update by giving dictname[value]=key - this will update it - if tht is nt present it will craete a new pair**

**else:**

**strdict[i]=1 # by giving this it will automatically create a dictionary with i as values and 1 as key**

**print(strdict)**

**#print the occurence of the charceters in the string**

**--------------------------------------------------------**

**temp=''**

**for i in string:**

**temp=temp+str(i)+str(strdict.get(i)) # .get will get the values-**

**print(temp)**

**for i in range(1,10,2):**

**print(i)**

**=>**

**1**

**3**

**5**

**7**

**9**

**Process finished with exit code 0**

**-------------------------------------**

**a=int(input())**

**b=int(input())**

**for i in range(a,b,3):**

**print(i)**

**=>**

**33**

**50**

**33**

**36**

**39**

**42**

**45**

**48**

**Process finished with exit code 0**

**---------------------------------------**

**a=int(input())**

**b=int(input())**

**odd\_sum=0**

**even\_sum=0**

**for i in range(a,b):**

**if(i % 2 == 0):**

**even\_sum=even\_sum + i**

**else:**

**odd\_sum=odd\_sum + i**

**print("EVEN: "+str(even\_sum))**

**print("ODD:" +str(odd\_sum))**

**=>**

**1**

**11**

**EVEN: 30**

**ODD:25**

**Process finished with exit code 0**

**-------------------------------------------**

**a="Welcome"**

**for i in a:**

**print(i)**

**=>**

**W**

**e**

**l**

**c**

**o**

**m**

**e**

**Process finished with exit code 0**

**-------------------------------------------**

**# strings are immutable**

**a="Welcome"**

**'''for i in a:**

**print(i)'''**

**a[2]='l'**

**=>**

**Traceback (most recent call last):**

**File "C:/Users/a08019dirp\_c2e.03.06/PycharmProjects/day2/prac.py", line 22, in <module>**

**a[2]='l'**

**TypeError: 'str' object does not support item assignment**

**Process finished with exit code**

**-------------------------------------------**

**#print the length of the string**

**a="Novin"**

**cnt=0**

**for i in a:**

**cnt=cnt+1**

**print(cnt)**

**=>**

**5**

**Process finished with exit code 0**

**--------------------------------------------**

**#print the count of occurance of character**

**a="Accenture"**

**cnt=0**

**for i in a:**

**if(i=='e'):**

**cnt=cnt+1**

**print(cnt)**

**=>**

**2**

**Process finished with exit code 0**

**=======================**

**for i in a:**

**#cnt = 0**

**if(i=='e'):**

**cnt=cnt+1**

**=>**

**1**

**Process finished with exit code 0**

**---------------------------------------------**

**#insert range into a list**

**c=[]**

**for i in range(1,11):**

**if(i%2==0):**

**c.append(i)**

**print(c)**

**=>**

**[2, 4, 6, 8, 10]**

**Process finished with exit code 0**

**-------------------------------------------**

**#convert list to dictionary removing duplicates without using in-build funcz**

**a = ['five', 5, 'six', 6, 'seven', 7, 'seven', 7, 'nine', 9]**

**dict1 = {}**

**for i in range(0, len(a), 2):**

**# print(a[i+1])**

**if a[i] in dict1:**

**l = dict1.get(a[i])**

**dict1[a[i]] = [l, a[i + 1]]**

**else:**

**dict1[a[i]] = a[i + 1]**

**print(str(dict1))**

**=>{'five': 5, 'six': 6, 'seven': [7, 7], 'nine': 9}**

**Process finished with exit code 0**

**---------------------------------------------------------------------------------**

**def Isin(str\_one,str\_two):**

**if str\_one in str\_two or str\_two in str\_one:**

**return True**

**else:**

**return False**

**#print(str\_one,str\_two)**

**str\_one = input("Enter a name")**

**str\_two = input("Enter a name")**

**if(Isin(str\_one, str\_two)):**

**print("string contains other string")**

**else:**

**print("no substring")**

**a,b=1,2**

**c=a+b**

**print("the sum is"+str(c))**

**print(a+b)**

**a=int(input())**

**b=int(input())**

**c=a+b**

**print("the sum is:"+str(c))**

**print("the sum is:",int(input())+int(input()))**

**a=[1,2,3,4]**

**b=[3,4,5]**

**#a.append(b)**

**#a.extend(b)**

**print(a)**

**a="accenture"**

**a=set(a)**

**b=list(a)**

**c=list(a)**

**b.extend(c)**

**print(b)**

**list1=[]**

**a="accenture"**

**for c in a:**

**if c not in list1:**

**list1.extend(2\*c)**

**print(list1)**

**for i in range(1,10,2):**

**#print(i)**

**sum1=0**

**sum2=0**

**for i in range(1,11):**

**if i%2==0:**

**sum1=sum1+i**

**else:**

**sum2=sum2+i**

**print(sum1)**

**print(sum2)**

**str1="welcome"**

**count=0**

**print(len(str1))**

**for i in str1:**

**count=count+1**

**print(count)**

**str1="welcome"**

**count=0**

**for i in str1:**

**if i=="e":**

**count=count+1**

**print(count)**

**str1="welcome"**

**print(str1[:3])**

**print(str1[3:])**

**print(str1.split("l",1))**

**for i in range(1,11):**

**if i%2==0 and i<=10:**

**list1.append(i)**

**print(list1)**

**from string import \***

**'''count1=0**

**count2=0**

**list1=['1','A','2','B']**

**for i in list1:**

**if i.isdigit():**

**count1=count1+1**

**elif i.isalpha():**

**count2=count2+1**

**print(count1)**

**print(count2)'''**

**for i in list1:**

**if ord(i)>=65 and ord(i)<=90:**

**count1=count1+1**

**else:**

**count2=count2+1**

**print(count1)**

**print(count2)**

**count1=0**

**count2=0**

**list1=[1,'A',2,'B']**

**for i in list1:**

**if type(i)==int:**

**count1=count1+1**

**else:**

**count2=count2+1**

**print(count1)**

**print(count2)**

**import re**

**count1=0**

**count2=0**

**list1=[1,'A',2,'B']**

**str1=str(list1)**

**match=re.findall('[A-Z]',str1)**

**print(match)**

**for i in list1:**

**if i in match:**

**count1=count1+1**

**else:**

**count2=count2+1**

**print(count1)**

**print(count2)**

**count1=0**

**count2=0**

**list1=['1','A','2','B']**

**for i in list1:**

**if i>="A" and i<="Z":**

**count1=count1+1**

**else:**

**count2=count2+1**

**print(count1)**

**print(count2)**

**dict1={}**

**list2=[]**

**list3=[]**

**list1=["five",5,"six",6,"seven",7,"seven",7,"nine",9]**

**for i in list1:**

**if type(i)==int:**

**list2.append(i)**

**else:**

**list3.append(i)**

**print(list2)**

**print(list3)**

**for i in list2:**

**for j in list3:**

**dict1[i]=j**

**list3.remove(j)**

**break**

**print(dict1)**

**#dict1=dict(zip(list2,list3))**

**dict1={}**

**list2=[]**

**list3=[]**

**list1=["five",5,"six",6,"seven",7,"seven",7,"nine",9]**

**for i in list1:**

**if type(i)==int:**

**list2.append(i)**

**else:**

**list3.append(i)**

**print(list2)**

**print(list3)**

**j=0**

**for i in range(len(list2)):**

**if list2[i] in dict1:**

**flag=dict1.get(list2[i])**

**dict1[list2[i]]=[flag,list3[i]]**

**else:**

**dict1[list2[i]]=list3[i]**

**print(dict1)**

**dict1={}**

**list2=[]**

**list3=[]**

**list1=["five",5,"six",6,"seven",7,"seven",7,"nine",9]**

**for i in list1:**

**if type(i)==int:**

**list2.append(i)**

**else:**

**list3.append(i)**

**print(list2)**

**print(list3)**

**j=0**

**for i in range(len(list2)):**

**if list2[i] in dict1:**

**flag=[dict1[list2[i]],list3[i]]**

**dict1[list2[i]]=flag**

**else:**

**dict1[list2[i]]=list3[i]**

**print(dict1)**

**for i in range(1,10):**

**for j in range(1,i):**

**print("\*",end=" ")**

**print("\n")**

**for i in range(1,5):**

**for j in range(1,5):**

**print("\*",end=" ")**

**print("\n")**

**def inte(a):**

**return a**

**c=inte(12)**

**print(c)**

**def inte(a,b):**

**c=a+b**

**return a,b,c**

**def total():**

**c1,c2,c3=inte(10,5)**

**print(c1,c2,c3)**

**total()**

**def inte(a,b=0):**

**c=a+b**

**return a,b,c**

**def total():**

**c1,c2,c3=inte(10)**

**print(c1,c2,c3)**

**total()**

**dict1={}**

**count=1**

**string1="accenture"**

**for i in range(len(string1)):**

**if string1[i] not in dict1:**

**dict1[string1[i]]=count**

**else:**

**dict1[string1[i]]=count+1**

**print(dict1)**

**EXCEPTION HANDLING:**

**# Python code to illustrate**

**# working of try()**

**def divide(x, y):**

**try:**

**# Floor Division : Gives only Fractional Part as Answer**

**result = x // y**

**print("Yeah ! Your answer is :", result)**

**except ZeroDivisionError:**

**print("Sorry ! You are dividing by zero ")**

**# Look at parameters and note the working of Program**

**divide(3, 0)**

**Importing: (For reusability)**

**import math**

**print("The sqq root of num is",math.sqrt(2))**

**You can also import your own modules:**

**def addition(a,b):**

**return a+b**

**def subtraction(a,b):**

**return a-b**

**import dayone**

**print(dayone.addition(1,2))**

**print(dayone.subtraction(1,2))**

**Only to import the specific function: ( from module import functionname)**

**from dayone import addition**

**To import all the functions:**

**import dayone.\***

**If you import only the module in the beginning, then when you use it, reference it using the module.method name.**

**But if you do it in this way:**

**from module import functionname**

**Then, directly you can use the method name without the module reference.**

**Write a program to print 1 to 10 numbers using modules.**

**from importexample import func**

**#Write a program to print 1 to 10 numbers using modules.**

**a=int(input("Enter the first value"))**

**b=int(input("Enter the second value"))**

**func(a,b)**

**def func(a,b):**

**for i in range(a,b):**

**print(i)**

**RENAMING A MODULE:**

**import math**

**import math as m**

**then ,**

**m.sqrt(a)**

**Functions:**

**(reusable and ordganised)**

**ALways,python involves bottom up approach. ie,**

**the function can be called only after the def not above it.**

**Pass by reference:**

**If a list is passed inside a function and modified ,outside also it changes as modified.**

**Arguments:**

**1.required**

**2.keyword**

**3.variable length**

**4. default arg**

**default arg:**

**def tut(name,session="python"):**

**print (name)**

**print (session)**

**tut("Muthu","c")**

**tut("muthu")**

**o/p:**

**Muthu**

**c**

**muthu**

**python**

**Note:**

**Always, the default arguments will be present in the last position.**

**keyword arg:**

**Irrespective of the order you can call the arguments.**

**def fun(name,age=50):**

**print(name)**

**print(age)**

**fun(name="Muthu",age=70)**

**fun(name="raman")**

**fun(age=100,name="raj") #keyword argument**

**variable length arguments:**

**def calculator(function,\*arg):**

**if function=="add":**

**for i in arg:**

**calculator("add",5,10)**

**calculator("add",5,10,10,20)**

**REFER:**

**\*\*kwarg**

**def fun(name, \*\*details):**

**print(name)**

**print(details["age"])**

**fun("Muthu", age=50)**

**SCOPE OF VARIABLES:**

**global and local variables**

**GLobal:**

**x = 141**

**def foo():**

**x = 424 #local variable**

**print x**

**foo()**

**print x**

**OUTPUT**

**424**

**141**

**TO CHANGE THE GLOBAL VALUE:**

**x = 141**

**def foo():**

**global x**

**x = 424**

**print(x)**

**foo()**

**print(x)**

**OUTPUT**

**424**

**424**

**JOIN IN STRING:**

**str=['044','24959511','37968']**

**n="-".join(str)**

**print(n)**

**REVERSE A LIST:**

**list1=['H','e','l','l','o']**

**list1.reverse()**

**print(list1)**

**#if you print(list1.reverse())**

**#It will give none**

**print(list(reversed(list1))**

**o/p:**

**['o', 'l', 'l', 'e', 'H']**

**['H', 'e', 'l', 'l', 'o']**

**list1=['H','e','l','l','o']**

**#list1.reverse()**

**#print(list1)**

**#if you print(list1.reverse())**

**#It will give none**

**print(list(reversed(list1)))**

**print(list1)**

**['o', 'l', 'l', 'e', 'H']**

**['H', 'e', 'l', 'l', 'o']**

**That is, it is advisable to use reversed instead of reverse because it will retain the original list always unlike reverse.**

**COMMAND LINE ARGUMENTS:**

**import sys**

**print("Name of the file" , sys.argv[0])**

**print("The first val from command line is",sys.argv[1])**

**(venv) C:\Users\A08019dirp\_c2e.03.06\PycharmProjects\pythontraining>python nivetha.py 1 2**

**Name of the file nivetha.py**

**The first val from command line is 1**

**Write a prog that accepts 3 arg. find the maximum out of them .**

**if there are no 3 args, then insufficient msg .**

**import sys**

**try:**

**if len(sys.argv) ==4:**

**if (int(sys.argv[1]) >int( sys.argv[2])) and int((sys.argv[3])):**

**print(sys.argv[1] ,"is the greatest")**

**elif (int(sys.argv[2]) > int(sys.argv[3])):**

**print(sys.argv[2] ,"is greatest")**

**else:**

**print(sys.argv[3] ,"is greatest")**

**else:**

**print("Type only 3 arguments")**

**except:**

**print('hi')**

**Bottom up concept:**

**def fun(a,b):**

**print(a)**

**print(b)**

**def fun(c):**

**print(c)**

**fun(10,4)**

**fun(9)**

**It'll show a error because of bottom up approach.**

**Lambda functions:**

**You cant do sum(a+b) or sum(int(a)) in the function definition.**

**Hence, we are going for lambda functions.**

**str1 = 'bbac'**

**str2 = 'bacb'**

**count = 0**

**if (len(str1) == len(str2)):**

**for i in range(0,len(str1)):**

**if str1[i] in str2 and str2[i] in str1:**

**count = count + 1**

**if count == len(str1):**

**print("anagram")**

**else:**

**print("Not")**

**string = input("enter a string")**

**dict1 = {}**

**for i in string:**

**dict1.update({i: string.count(i)})**

**print("dictionary : ", dict1)**

**dict1.pop('c')**

**print("dictionary : ", dict1)**

**dict1.popitem()**

**print("dictionary : ", dict1)**

**# How do you print duplicate characters from a string?**

**name = input("Enter a string")**

**print("duplicate characters")**

**dict1 = {}**

**for i in name:**

**cnt = name.count(i)**

**if (cnt > 1):**

**dict1.update({i: cnt})**

**print(dict1)**

**# How to find the maximum occurring character in given String**

**string = input("enter a string")**

**dict1 = {}**

**for i in string:**

**dict1.update({i: string.count(i)})**

**# print("dictionary : ", dict1)**

**max1 = 0**

**char = ''**

**for i,j in dict1.items():**

**if j > max1:**

**max1 = j**

**char = i**

**print("Max Occurring Char : ", char)**

**# print("Max Occurring Char : ", max(dict1.values()))**

**# How do you print the first non-repeated character from a string**

**name = input("Enter a name")**

**for i in name:**

**if (name.count(i) == 1):**

**print(i)**

**break**

**# How do you check if a string contains only digits**

**name = input("Enter a string")**

**if name.isdigit():**

**print("All are digits")**

**else:**

**print("contains other char")**

**# palindrome of number**

**number = int(input("Enter a number"))**

**str2 = ''**

**str\_num = str(number)**

**for i in range(len(str\_num) - 1, -1, -1):**

**str2 = str2 + str\_num[i]**

**if str\_num == str2:**

**print("Palindrome")**

**else:**

**print("Not a Palindrome")**

**# palindrome**

**number = int(input("Enter a number"))**

**temp = number**

**remainder = 0**

**while temp > 0:**

**num = temp % 10**

**remainder = remainder \* 10 + num**

**temp = temp // 10**

**if remainder == number:**

**print("Palindrome")**

**else:**

**print("Not a Palindrome")**

**string = input("Enter a String")**

**str1 = ''**

**for i in range(len(string) - 1, -1, -1):**

**str1 = str1 + string[i]**

**if string == str1:**

**print("Palindrome")**

**else:**

**print("Not a Palindrome")**

**# palindrome of number**

**number = int(input("Enter a number"))**

**str2 = ''**

**str\_num = str(number)**

**for i in range(len(str\_num) - 1, -1, -1):**

**str2 = str2 + str\_num[i]**

**if str\_num == str2:**

**print("Palindrome")**

**else:**

**print("Not a Palindrome")**

**# palindrome**

**number = int(input("Enter a number"))**

**temp = number**

**remainder = 0**

**while temp > 0:**

**num = temp % 10**

**remainder = remainder \* 10 + num**

**temp = temp // 10**

**if remainder == number:**

**print("Palindrome")**

**else:**

**print("Not a Palindrome")**

**# how to remove the duplicate character from String**

**string = input("Enter a String")**

**list1 = []**

**for i in string:**

**list1.append(i)**

**for i in list1:**

**if list1.count(i) > 1:**

**list1.remove(i)**

**str1 = ''**

**for i in range(0,len(list1)):**

**str1 = str1 + list1[i]**

**print(str1)**

**# reverse of a string**

**name = input("Enter a String")**

**str1 = name[::-1]**

**print("Reverse of a string : ", str1)**

**str2 = ''**

**for i in range(len(name) - 1, -1, -1):**

**str2 = str2 + name[i]**

**print("Reverse of string : ", str2)**

**# How do you count a number of vowels and consonants in a given string**

**name = input("Enter a string")**

**list1 = []**

**list2 = []**

**for i in name:**

**if i in ('a','e','i','o','u','A','E','I','O','U'):**

**list1.append(i)**

**else:**

**list2.append(i)**

**dict1 = {'vowels':len(list1),'consonants':len(list2)}**

**print("count : ",dict1)**

**TASKS:**

**1)**

**string = input("enter a string")**

**dict1 = {}**

**for i in string:**

**dict1.update({i: string.count(i)})**

**print("dictionary : ", dict1)**

**# dictionary to string**

**str1 = ''**

**for j in dict1:**

**str1 = str1+j+str(dict1[j])**

**print("String : ", str1)**

**2)**

**string = input("enter a string")**

**str1 = ''**

**for i in string:**

**str1 = str1 + i + str(string.count(i))**

**print(str1)**

**3)**

**import re**

**string = """Hakka and Bukka were brothers and warriors. The brothers wanted to build their own kingdom where people could live without fear. They collected a band of young men and trained them in warfare. They lived in a forest hideout on the banks of the river Tungabhadra in South India.**

**One day, the brothers were out on a hunt. Ferocious dogs accompanied them. They crossed the river and rode on. A couple of frightened rabbits ran out of the bushes. The dogs gave them chase with the two brothers closely behind on their horses.**

**It was a long chase. The rabbits were running for their life. The dogs were catching up. Suddenly, in a swift move, the rabbits turned and faced the dogs. Taken aback by the show of defiance, the barking dogs stepped back. Hakka called back the dogs. As the dogs turned back, the rabbits walked away.**

**Hakka looked around. They were on the other side of the Tungabhadra. It was a rocky land. The sun was blazing in the sky.**

**“Strange! I’ve never seen rabbits challenging dogs before!” said Bukka.**

**“That’s the quality of this land,” said a quiet voice, “Even rabbits give fight.”**

**Startled to hear a stranger speak, the two brothers turned.**

**They saw a holy man walking towards them. He was a picture of peace. At the same time, his eyes were blazing bright."""**

**print(len(string))**

**str1 = string.replace('.', "").replace(',', "").replace('!', "").replace('\"', "").replace('\n',"")**

**lst = str1.split(" ")**

**dict1 = {}**

**list2 = []**

**print(len(lst))**

**for i in lst:**

**dict1.update({i: lst.count(i)})**

**print("dictionary : ", dict1)**

**max1 = 0**

**char = ''**

**for i, j in dict1.items():**

**if j > max1:**

**max1 = j**

**char = i**

**print("Max Occurring Char : ", char)**

**print("Max Occurring : ", max(dict1.values()))**

**4)**

**lst = ['five plus three', 'seven minus two', 'two plus eight minus five', 'eight divide four']**

**dct = {'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5, 'six': 6, 'seven': 7, 'eight': 8, 'nine': 9, 'plus': '+', 'minus': '-', 'divide': '/'}**

**lst1 = []**

**for i in lst:**

**string = i.split(" ")**

**str2 = ''**

**for j in string:**

**if j in dct:**

**str2 = str2 + j.replace(j, str(dct[j]))**

**lst1.append(str2)**

**print(lst1)**

**lst2=[]**

**for k in lst1:**

**lst2.append(eval(k))**

**print(lst2)**

**5)**

**import re**

**string = """Hakka and Bukka were brothers and warriors. The brothers wanted to build their own kingdom where people**

**could live without fear. They collected a band of young men and trained them in warfare. They lived in a forest**

**hideout on the banks of the river Tungabhadra in South India. One day, the brothers were out on a hunt. Ferocious**

**dogs accompanied them. They crossed the river and rode on. A couple of frightened rabbits ran out of the bushes. The**

**dogs gave them chase with the two brothers closely behind on their horses. It was a long chase. The rabbits were**

**running for their life. The dogs were catching up. Suddenly, in a swift move, the rabbits turned and faced the dogs.**

**Taken aback by the show of defiance, the barking dogs stepped back. Hakka called back the dogs. As the dogs turned**

**back, the rabbits walked away. Hakka looked around. They were on the other side of the Tungabhadra. It was a rocky**

**land. The sun was blazing in the sky. "Strange! I’ve never seen rabbits challenging dogs before! " said Bukka.**

**"That’s the quality of this land, " said a quiet voice, "Even rabbits give fight. " Startled to hear a stranger**

**speak, the two brothers turned. They saw a holy man walking towards them. He was a picture of peace. At the same**

**time, his eyes were blazing bright. """**

**# print("Length Of String : ",len(string))**

**str1 = string.replace('.', "").replace(',', "").replace('!', "").replace('\"', "").replace('\n',"")**

**lst = str1.split(" ")**

**dict1 = {}**

**list2 = []**

**# print("Number Of Words : ",len(lst))**

**for i in lst:**

**dict1.update({i: lst.count(i)})**

**# print("dictionary : ", dict1)**

**mini = 100**

**char = ''**

**lst1 = []**

**lst2 = []**

**for i, j in dict1.items():**

**if j <= mini:**

**mini = j**

**char = i**

**# print("Min Occurring : ", min(dict1.values()))**

**# print("Min Occurring Word : ", char)**

**for k,l in dict1.items():**

**if dict1[k] == mini:**

**lst1.append(k)**

**lst2.append(l)**

**# print("Repeated word's list : ",lst1)**

**# print("Repeated word's count : ",lst2)**

**dict2 = {"Length Of String": len(string), "Number Of Words": len(lst), "Max Occurring": max(dict1.values()),**

**"Max Occurring Word": char, "Repeated word's list": lst1, "Repeated word's count": lst2}**

**print(dict2)**

**6)**

**# prime or not**

**num1 = int(input("enter a number"))**

**num2 = int(input("enter a number"))**

**list1 = []**

**for i in range(num1,num2):**

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

**if i != j and i % j == 0:**

**break**

**else:**

**if i not in list1:**

**list1.append(i)**

**print(list1)**

**LAMBDA FUNCTIONS:**

**\*) Map**

**\*)Filter**

**\*)Reduce**

**input=[1,2,3,4,5]**

**output=list(map(lambda a:a\*2,input)**

**print(output)**

**COMPREHENSION:**

**\*)list comprehension**

**for eg,printing even numbers in a range:**

**for i in range(1,50):**

**if i %2==0:**

**list1.append(i)**

**(or)**

**list1=[i for i in range(1,50) if i%2==0]**

**list1=[variable i/pseq condition]**

**list2=[i for i in range(1,50) if i%3==0]**

**print(list2)**

**[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48]**

**lis=[]**

**for i in range(1,100):**

**if i%3==0:**

**lis.append(3)**

**else:**

**lis.append(i)**

**print(lis)**

**s1="THis is a new batch in this year"**

**a=s1.split(" ")**

**for j in a:**

**if len(j)%2==0:**

**print(j)**

**s1="THis is a new batch in this year"**

**a=s1.split(" ")**

**lis=[]**

**lis.append(i[0])**

**print(lis)**

**s="nivetha is a girl"**

**li=[]**

**st = s.split(" ")**

**for i in st:**

**li.append(i[0])**

**print(li)**

**Dictionary comprehension:**

**{key:value loop condition}**

**eg,**

**11728204**

**2 times print each digit**

**dict1={i:i\*2 for i in String}**

**print(dict1)**

**Printing yes for leap years and vice versa:**

**dict1={i:('yes' if i%4==0 and i%100==0 else 'No') for i in range(2000,2019)}**

**print(dict1)**

**list1=[i for i in range(2,10) for j in range(2,i) if i%j==0]**

**list2=[item for item in range(2,10) if item not in list1]**

**print(list2)**

**list3=[x for x in range(2,10) if all(x%y!=0 for y in range(2,x))]**

**print(list3)**

**lis1=[2,5,8]**

**if lis1[0] in range(lis1[1],lis1[2]):**

**print("yes")**

**else:**

**print("no")**

**REMOVING DUPLICATES:**

**lis=[1,1,2,4,6,8,5]**

**lis1=[]**

**for i in lis:**

**if i not in lis1:**

**lis1.append(i)**

**print(lis1)**

**FILE HANDLING:**

**fh=open(filename,mode)**

**by default, read only mode.**

**you should explicitly close**

**(or)**

**open(<filename> mode) as fh**

**you nee not explicitly close**

**fh=open(filename,mode)**

**for i in fh:**

**print(i)**

**Readline()**

**fh.read() # you can also mention the buffer size**

**fh.read(100) #including the white spaces**

**fh.readline() # line by line output**

**fh.readlines()**

**Modes:**

**r,w,a**

**r -readonly**

**w- itll over write the contents if file exists**

**if not itll create a new file and write new contents**

**a- itll just append contents but not over writes**

**rb- We will get only bytes output**

**wb-**

**ab-**

**WRITE:**

**write() method**

**seek()**

**# seek(0,1)**

**tell() where youre currently pointing**

**flush() # it will flush the buffer**

**close()**

**with open('abc.txt') as fh1, open('test.txt') as fh2:**

**for line1, line2 in zip(fh1, fh2):**

**# line1 from abc.txt, line2 from test.txtg**

**print(line1+line2)**

**READ AND WRITE A CSV FILE AND A JSON FILE:**

**CSV:**

**First import the csv module**

**4 important packages:**

**reader**

**Dict reader**

**writer**

**Dict writer**

**with open(filename) as fh:**

**csvr=csv.reader(fh,delimiter)**

**for i in csvr:**

**print i[2]**

**Delimiter should be used for reader but not dict reader**

**Dict reader will give a dictionary object with which you can iterate through a dictionary .**

**eg,print(i['employeename'])**

**Dict writer:**

**You have to pass a dictionary .**

**JSON:**

**{ :**

**{ :**

**{**

**}**

**json has 4 methods:**

**load**

**loads**

**dump**

**dumps**

**'s' stands for string**

**load and loads are used for reading**

**dump and dumps are used for writing**

**with open(filename) as fh:**

**obj=jsonload(fh)read()**

**now to convert into dictionary,**

**plain dict into a file-> dump**

**dict converted to string into file=>dumps**

**If its a .json file, just use load and dump**

**Note:**

**module json**

**3 ways to solve a python program:**

**1)procedural**

**2)functional(lambda)**

**3)oops concept**

**classes-design /blueprint**

**objects- Instance of classes**

**functions are called as methods in oops**

**class oneplus:**

**def --init(self,ram,storage)**

**self.ram=ram**

**self.storage=storage**

**eg,**

**class nivetha:**

**a=12**

**def \_\_init\_\_(self,name,age):**

**self.name=name**

**self.age=age**

**n=nivetha('nivetha',22)**

**print(n.name)**

**print(n.age)**

**self is used to explicitly use the object name**

**SO, methods and variables called encapsulation**

**2 types of variables:**

**instance- different objects**

**class/static - outside the init method**

**class Student:**

**age=21**

**def St(self,mark):**

**self.mark=mark**

**print(mark)**

**p=Student()**

**p.St(750)**

**Example of default constructor :**

**class GeekforGeeks:**

**geek = ""**

**# default constructor**

**def \_\_init\_\_(self):**

**self.geek = "GeekforGeeks"**

**# a method for printing data members**

**def print\_Geek(self):**

**print(self.geek)**

**# creating object of the class**

**obj = GeekforGeeks()**

**# calling the instance method using the object obj**

**obj.print\_Geek()**

**Example of parameterized constructor :**

**filter\_none**

**edit**

**play\_arrow**

**brightness\_4**

**class Addition:**

**first = 0**

**second = 0**

**answer = 0**

**# parameterized constructor**

**def \_\_init\_\_(self, f, s):**

**self.first = f**

**self.second = s**

**def display(self):**

**print("First number = " + str(self.first))**

**print("Second number = " + str(self.second))**

**print("Addition of two numbers = " + str(self.answer))**

**def calculate(self):**

**self.answer = self.first + self.second**

**# creating object of the class**

**# this will invoke parameterized constructor**

**obj = Addition(1000, 2000)**

**# perform Addition**

**obj.calculate()**

**# display result**

**obj.display()**

**First number = 1000**

**Second number = 2000**

**Addition of two numbers = 3000**

**Polymorphism:**

**1)operator overloading**

**a=6**

**b=8**

**print(a+b)**

**(or)**

**a=6**

**b=7**

**print(a.\_\_add\_\_(b))**

**a=7**

**b=9**

**print(a.\_\_ge\_\_(b))**

**2)**

**Method overloading:**

**3)**

**Method overriding:**

**Abstraction:**

**In python ,there is no abstract keyword like in java.**

**Just import some modules to work with abstract classes.**

**from abc import abc ,abstract method**

**class A:**

**def add():**

**pass**

**class BC:**

**def add():**

**INTERVIEW QUESTION:**

**In a list, separate the integers and alphabets.**

**1)type**

**2)ord**

**3)regex**

**4)ascii**

**5)><**

**6)is digit, is alpha**

**7)range**

**8)isInstance**

**9)try except inside a for loop**

**Generators,iterators and decorators:**

**Iterators are iterable.**

**Generators are also iterators(to do tuple comprehension)**

**Decorsators are used as a wrapper.**

**To loop through generators, use --next()--**

**next(gen)**

**# A generator function that yields 1 for first time,**

**# 2 second time and 3 third time**

**def simpleGeneratorFun():**

**yield 1**

**yield 2**

**yield 3**

**# Driver code to check above generator function**

**for value in simpleGeneratorFun():**

**print(value)**

**(or)**

**# A Python program to demonstrate use of**

**# generator object with next()**

**# A generator function**

**def simpleGeneratorFun():**

**yield 1**

**yield 2**

**yield 3**

**# x is a generator object**

**x = simpleGeneratorFun()**

**# Iterating over the generator object using next**

**print(x.next()); # In Python 3, \_\_next\_\_()**

**print(x.next());**

**print(x.next());**

**Output :**

**1**

**2**

**3**

**stopiteration**

**FIBONACCI USING YIELD:**

**def fib(limit):**

**a, b = 0, 1**

**while a < limit:**

**yield a**

**a, b = b, a + b**

**# Create a generator object**

**x = fib(5)**

**print(next(x))**

**print(next(x))**

**print(next(x))**

**print(next(x))**

**FUNCTION DECORATOR:**

**def hello\_decorator(func):**

**def inner1():**

**print("Hello, this is before function execution")**

**func()**

**print("This is after function execution")**

**return inner1()**

**@hello\_decorator**

**def function\_to\_be\_used():**

**print("This is inside the function !!")**

**You can add any number of decorators on a function.**

**CLASS DECORATORS:**

**class A**

**def \_\_init\_\_():**

**--------**

**======**

**def \_\_call\_\_():**

**f()**

**@A**

**def some()**

**------**

**Exception :**

**process may be:**

**1.proper message**

**2.storage in log**

**3.close the db**

**try:**

**except:**

**finally:**

**User defined exceptions:**

**try:**

**in=input("Enter a value")**

**if in<0:**

**raise negativeerror(in)**

**except valueerror as val:**

**print(val)**

**except negativeeroor as neg:**

**print(in+ "is a negative number")**

**except stringargerror as s:**

**print(s)**

**except Exception as e:**

**print(str(e))**

**try:**

**i=input("Enter the value")**

**if i.isdigit():**

**raise ValueError**

**else:**

**print(i)**

**except Exception as e:**

**print("Exception base")**

**except ValueError :**

**print("Value error")**

**Order is also considered.**