**Python Basics**IDLESypderjupiterEclipsepycharmPython FundamentalsThis PC --> Properties --> Advance system Settings --> Environment variable --> System Variable --> Path --> C:\python35

**Python Key Tokens** **Keywords -->** Reserved words which has its own meaning Python terminal --> help() --> keywords --> 33 keywords **Constants** The values will not change in run time Numerical Non-Numerical Natural/Integers Characters Real/Decimals/Float Strings Complex Boolean

**DataTypes --->** Type of the data which is stored Also named as Labels 2 types **1. Primitive datatype** Where we can store a single value Integer --> Float Boolean String Complex **2. Non primitive datatype** Where we can store multiple values List Tuple Dictionary Set

**Identifiers -->** name of a variable var **Variables** The value will change in runtime int a; datatype identifier; in Python a=10 --> based on context a="10" --> String called as literals  **Comments** # --> single line comment ''' --> multiple line comments

**Output statements:** display the text or the values on the console print() 2 types **formatted print** C-style print --> %d,%f,%s --> we can print anywhere in the statements nm="Athma" Exp=12.5 print("My name is %s and have the Experience of %f years"%(nm,Exp))  **unformatted print** print(""Myname is ",nm," and i have the experience of ",exp," years

**condition statements bare if**if(condn): statement

**if else**

if(condn): statement1else: statement2

**nested if**

if(condn1): if(cond2): statement1 else: satatement2 statement3else: if(cond3): statement3

**ladder if -->** if(cond1): statement1elif(cond2): statement2elif(condn3) statement3else: statement4

**Loops**

**for loop**

for i in 1,2,3,4 statement

for i in range() statement

for i in list statement

initializationconditionincrement/decrementwhile loopintitwhile(condition): statement incr/decrintitwhile(condition): statement incr/decrelse: statement

program0:

print("Objects before initialize the literals",dir())a=10b="wipro"print("Objects after initialize the literals",dir())print("The a is ",type(a), " and b is ",type(b))print("I am with ",b," from past ",a," years") #Unformatted printprint("I am with %s from past %.2f years"%(b,a)) #Formatted or C style

program 3:

#a=eval(input("Enter the value for A \n"))#if(a==10):# print("a is equals to 10")#else:# print("a is not equals to 10")#print("We are out of if-else block")a=eval(input("Enter the value for A \n"))#"Sunil"if(a=="wipro"): print("yes, it is wipro")else: print("It is not wipro")

…………………………………………….

Program4

#Nested if#greater in 3 numbersa=int(input("Enter the value for A \n")) #1b=int(input("Enter the value for B \n")) #20c=int(input("Enter the Value for C \n")) #2if a>b: if a>c: print(a," is greater") else: print(c," is greater")else: if b>c: print(b," is greater") else: print(c," is greater")

Program5

# Ladder if# Vowels ch=str(input("Enter an alphabet \n")) #pif(ch=='a'): print(ch," is a Vowel")elif(ch=='e'): print(ch," is a vowel")elif(ch=='i'): print(ch," is a vowel")elif(ch=='o'): print(ch," is a vowel")elif(ch=='u'): print(ch," is a vowel")else: print(ch," is not a vowel")

Program6

#using or operatorch=str(input("Enter an Alphabet \n")) if ch=='a' or ch=='e' or ch=='i' or ch=='o' or ch=='u': print(ch," is Vowel")else: print(ch," is not Vowel")

…………………….

#And Operator#greater in 3 numbersa=int(input("Enter the value for A \n"))b=int(input("Enter the value for B \n"))c=int(input("Enter the Value for C \n"))if a>b and a>c: print(a," is greater") elif b>c: print(b," is greater") else: print(c," is greater")

…………………..

#x=int(input("Enter the value \n"))y=input("Enter the value \n")if not y: print(" True Block Y = ",y)else: print("False block")

Prog9

#For Loop#for i in 1,2,3,4,5,6,7,8,9,10:# print("i is ",i)#for z in range(1,10,1): #range(start:stop(<10 ==9):step)# print("z is ",z)#for z in range(10): # start:0 stop:9 step:1# print("z is ",z)#for q in range(10,100,10):# print("q is ",q)for p in range(100,-1,-10): print("p is ",p)

#While loop

Prog10

''' cnt=0 #initializationwhile (cnt<=10): #condition print("Value is ",cnt) cnt+=1 #incrementation'''i=1while (i<5): print("Value is ",i) i=i+1else: print("The Value of i is ",i)

List --> mutable,heterogeniuous,ordered set of elements Mutable --> modify Heterogenious --> different types of data ordered --> in sequential --> indexes [] --> index will starts with 0 names=["wipro","TCS","Infosys"] lst=[10,3.142,True,"Wipro"] lst[0] --> 10 lst[3] --> "wipro" List functions --> append() --> add the value at the end of list insert(arg1(index),arg2(Value)) --> insert the values in given index number pop() --> it will remove the last element of the list pop(index) --> remove the value based on index remove() --> remove based on values index() --> it will return the index number of given value sort() --> sort the elements in ascending order reverse() --> mirror display of the list count() --> display the repeaded element in the list extend() --> merge the lists………………………..

sum=0num=12345strng=str(num)for z in strng: sum=sum+int(z)print(sum)

………………………

dt1=[1,2,3,4,5,6,7,9,10]'''for i in dt1: if ((i%2)==0) : print(i," is Even") else: print(i," is Odd")for i in dt1: print("i = ",i," and index of i is",dt1.index(i))strng="wipro" #collections of characters --> list --> ['w','i','p','r','o']for z in strng: if z=='a' or z=='e' or z=='i' or z=='o' or z=='u': print(z," is the Vowel") else: print(z," is not a Vowel")# String will be conidered as list with set of characters

…………………………..

dt=["wipro","technologies","Sandton City", "SA"]print("The length of the List is ",len(dt))print("List is ",dt)dt.append(200100)print("List After append ",dt)dt.insert(3,"Johansberg")print("List After insert ",dt)dt.pop()print("List After poping the last element ",dt)dt.pop(3)print("List After poping the 3rd index value ",dt)print(" index number of given list valu ",dt.index("SA"))dt.reverse()print(" Reverse the list elements ",dt)dt.remove("wipro")print("deletes the the values from the list",dt)……………………….

dt1=[1,20,30,50,0,1,10,5,15,25]mx=max(dt1)mn=min(dt1)ln=len(dt1)print("Max,Min,Length is ",mx,mn,ln)print("1 repeated of ",dt1.count(1)," times")dt1.sort()print("The sorted list is ", dt1)………………………………

# data appending to empty listdata=[]for i in range(5): #range(0,<5,1) ap=eval(input("Enter the values to append \n")) data.append(ap)print("The list values of data ", data )# data retriew from the listfor i in data: print("The index ",data.index(i),"has the data",i)#print(data.count(1))

……………………

strng="Technologies"print("Length of strng is ",len(strng))#for ele in strng:# print("alphabet is ",ele)#for ele in strng[2:8:2]:# print(ele,end=",")for ele in strng[:7:]: print(ele,end=" ")

………………………..

num=[50,9,8,17,3,5,16,10]'''# Wrong patternfor n in len(num): print(n)'''# Right Patternfor n in range(len(num)): print(n)for n in range(len(num)): print(num[n])strng="Mahesh"for i in range(len(strng)): print(i)a=10for i in range(a): print i…………………………………

num=[50,9,8,17,3,5,16,10]#for n in range(len(num)):# if (num[n]%2)==0 :# print(num[n]," is Even")# else:# print(num[n]," is Odd")for n in num: if (n%2)==0 : print(n," is Even") else: print(n," is Odd")

……………………………………………….

from Sunil Kumar Aralimara Channappa to everyone: 3:30 PM

'''strng=eval(input("Enter a string \n"))for val in strng: #print("type is ",type(val)) if(val >='a' and val <='z') or (val>='A' and val<='Z'): print(val," is Alphabet") elif( val >='0' and val<='9' ): print(val, " is the digit") else: print(val ," is Special character")'''strng=eval(input("Enter a string \n"))for val in strng: #print("type is ",type(val)) if(ord(val) >=65 and ord(val) <=90) or (ord(val)>=97 and ord(val)<=122): print(val," is Alphabet") elif( ord(val) >=48 and ord(val) <=57 ): print(val, " is the digit") else: print(val ," is Special character")

…………………………..

Dictionary

car={"name":"I20","Version":"Sports","cost":"8.5L"}#print("dictionary is ", car)#print(" Name of the car ",car["name"])#print("Keys in car ",car.keys())#print("Values in car ",car.values())car.clear()print(" car dictionary after cleare the elemensts ",car)

…………..

car={"name":"I20","Version":"Sports","cost":"8.5L"}#print("dictionary is ", car)car.update(name="Verna")print("After update the name in car dict is",car)car1={"gear":"Auto"}car.update(car1)print("After update the dictionary car",car)car.update(color=["red","white","Blue","Gray"])print("After update the dictionary car dict with colors",car)

…………………………….

'''d={1:"one",2:"two"}d.update({3:"three"})print("After updating d with three",d)d.pop(1)print("After popping key 1, d is ",d)d1={}print("d1 is ",d1)d1=d.copy()print("d1 after copying the elements from d ",d1)'''d1=car.copy()print("d1 after copying the elements from car ",d1)'''

……………………………….

'''st1=set()print("set is ",st1," and its type is ",type(st1))st2=set([1,2,3,4,'a',5.2,False])print("st2 is : ",st2)st3={30,50,1,4,0}print("st3 is : ",st3)'''st4={1,2,31,2,4,59,"True"}print("st4 is : ",st4)…………………………………………………..

#Heterogeniuos and un ordered'''st1=set()print("set is ",st1," and its type is ",type(st1))st2=set([1,2,3,4,'a',5.2,False])print("st2 is : ",st2)st3={30,50,1,4,0}print("st3 is : ",st3)'''st4={1,2,31,2,4,59,"True"}print("st4 is : ",st4)

……………………………………………

st1={1,3,4,7,9,21}st1.add(12)print("st1 is ",st1)st2={3,21,5,47,98,101,12}# union --> combine both the setsst3=st1.union(st2)print("st3 is ",st3)# other way of combining 2 setsst4=st1|st2print("after union with | st4 is ",st4)#st1=st1.union(st2)#print("st1 is ",st1)#intersectionst5=st1.intersection(st2)print("after intersection st5 is ",st5)st6=st2&st1print("after intersection using & st6 is ",st6)#differencest7=st1.difference(st2)print("difference in st1 and st2 is ",st7)st8=st2 - st1print("difference in st1 and st2 using - is ",st8)#st1.clear()#print("st1 is ",st1)…………………………………………..

#jump statements'''for ch in "corona": if ch=="r": break print("current leter is ",ch)for ch in "corona": if ch=="o": continue print("current leter is ",ch)'''for ch in "corona": if ch=="o": pass print("current leter is ",ch)

……………………………………………

'''def cars(): print("I like to drive 4 wheelers")cars()'''def functionpara(a,b): print ("The values of a and b is ",a,b," respectively") print ("The address of a and b is ",id(a),id(b), " respectively")#x=10#y=20#print ("The address of x and y is ",id(x),id(y), " respectively")#functionpara(x,y)#functionpara(10,20)functionpara(int(input("Enter the Value \n")),int(input("Enter the Value \n")))…………………………………….

def arith(a,b): return a+b,a-b,a\*b,a/b#ans=arith(10,20)#print(ans)#print(arith(10,20))add,sub,mul,div=arith(20,10)print("add =",add," mul=",mul," div=",div," sub=",sub)

…………………………………..

def arith(a,b): #return is not mentioned with value it will return none return a+b,a-b,a\*b,a/badd,subt,mul,div=arith(20,10)print("addition of 10 and 20 is ",add)print("substraction of 10 and 20 is ",subt)print("multiplition of 10 and 20 is ",mul)print("division of 10 and 20 is ",div)a=20b=21def arith(a=10, b): #return is not mentioned with value it will return none global b return a+b,a-b,a\*b,a/badd,subt,mul,div=arith()print("addition of a and b is ",add)print("substraction of a and b is ",subt)print("multiplition of a and b is ",mul)print("division of a and b is ",div),,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

def arith(a=0,b=1): #return is not mentioned with value it will return none return a+b,a-b,a\*b,a/bx=10y=20add,subt,mul,div=arith(x)print("addition of 10 and 20 is ",add)print("substraction of 10 and 20 is ",subt)print("multiplition of 10 and 20 is ",mul)print("division of 10 and 20 is ",div)

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

def arith(a=0,b=1): #return is not mentioned with value it will return none return a+b,a-b,a\*b,a/bx=10y=20add,subt,mul,div=arith(x,y)print("addition of 10 and 20 is ",add)print("substraction of 10 and 20 is ",subt)print("multiplition of 10 and 20 is ",mul)print("division of 10 and 20 is ",div)

……………………………………………………………..

Types of the functions 1. Built in function pre-defined functions which are already exist in the python library input(),print(),help(),exit(),max(),min(),eval() dir() --> \_\_builtins\_\_ 2. User Defined function #Define the function def function\_Name(): print("") #call function function\_Name() Passing Parameters def fun(par1,par2): print("para1=",par1) fun(10,20) return statement return alows to return multiple values

…………………………………………

Data Structures List --> mutable,heterogeniuous,ordered set of elements Mutable --> modify Heterogenious --> different types of data ordered --> in sequential --> indexes [] --> index will starts with 0 names=["wipro","TCS","Infosys"] lst=[10,3.142,True,"Wipro"] lst[0] --> 10 lst[3] --> "wipro" List functions --> append() --> add the value at the end of list insert(arg1(index),arg2(Value)) --> insert the values in given index number pop() --> it will remove the last element of the list pop(index) --> remove the value based on index remove() --> remove based on values index() --> it will return the index number of given value sort() --> sort the elements in ascending order reverse() --> mirror display of the list count() --> display the repeaded element in the list extend() --> merge the lists

len()max()min()del() Tuple Immutable,heterogenious,ordered set of data tuple=() tuple=(1,"Tech",True,1.234) Dictionary mutable,heterogenious unordered set of data No indeces --> is the key-Value pair dictionary ==> {Key1:Value1,Key2:Value2...} Copy() clear() Keys() Values() update() pop() Set Mutable,heterogenious,un-ordered st={} --> empty dictionary st=set() --> empty set allow duplicate values

……………………..

Clear Copy StringFunctions/Modules/Packages set of statements to achieve the perticular task Types of the functions 1. Built in function pre-defined functions which are already exist in the python library input(),print(),help(),exit(),max(),min(),eval() dir() --> \_\_builtins\_\_ 2. User Defined function #Define the function def function\_Name(): print("") #call function function\_Name() Passing Parameters def fun(par1,par2): print("para1=",par1) fun(10,20) return statement return alows to return multiple values

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

Types of the functions 1. Built in function pre-defined functions which are already exist in the python library input(),print(),help(),exit(),max(),min(),eval() dir() --> \_\_builtins\_\_ 2. User Defined function #Define the function def function\_Name(): print("") #call function function\_Name() Passing Parameters def fun(par1,par2): print("para1=",par1) fun(10,20) return statement return alows to return multiple values 3. Anonymous function No name for the functions --> keyword lambda to define anonymous functions lambda arg1,arg2,arg3: arg1+arg2

DAY2:

Modules --> Module is a python file with .py extn collections of the functions in python file Types in modules 2 Types Built in modules modules are pre defined with python lib help>modules How to use built in modules Create an aliases import keyword User defined modules .py functions

………………………………………….

#print("before import math" , dir())import math#print("After import math",dir())#When you import all functions from math module#from math import \*#print(dir(math))#from math import sqrt#print("square root of 4 is", sqrt(4))#When you have nt imported the functions#print("square root of 4 is", math.sqrt(4))#from math import sqrtfrom math import sqrt as SQRTprint("square root of 4 is", SQRT(4))#print("square root of 9 is", sqrt(9))#import math#from math import sqrt,pow,pi#from math import \*

…………………………………

def add(a,b): return a+bdef mul(a,b): return a\*bdef div(a,b): return a/bdef sub(a,b): return a-b

………………………………….

from sys import pathpath.append("D:/")import arithfrom arith import add,mul#print("After import math",dir())#print(dir(arith))x=10y=20print("addition of x and y is ",add(x,y))print("multiplication of x and y is ",mul(x,y))

…………….

<https://pypi.org/>

#comparedef greater(a,b): if (a>b): return a else: return bdef StrngComp(a,b): if a==b : print("same") else: print("Not same")……………………….

from sys import pathpath.append("D:/WCF")import arithimport Compare

……………………….

def add(a,b): return a+bdef mul(a,b): return a\*bdef div(a,b): return a/bdef sub(a,b): return a-b

……………………………..

from sys import pathpath.append("D:/")import WCF#print(dir(WCF))from arith import \*#print(dir())from Compare import \*print("greater in 2 numbers",greater(1,2))

………………………..

a="sunil@123"for i in a: if i.isalpha(): print(i," is alphabet") elif i.isdigit(): print(i," is digit") else: print(i, " is other character")……………………………………

Strings isAlpha() --> isdigit() upper() lower() split() strip()

#i="Talent@123"'''upr=i.upper()print("Convert the string to upper ",upr)lwr=upr.lower()print("Convert the string to lower ",lwr)for ch in i: if ch.isdigit(): print(ch," is number") elif ch.isalpha(): if ch.isupper(): print(ch," is Alphabet and uppercase") else: print(ch," is Alphabet and lowercase") else: print(ch," is a special character")

…………………….

strng=" Talent "print(" The value before strip ",strng)strng1=strng.strip()print(" The value after strip ",strng1)……………………..

stng="we.are.learning.Python"stng1=stng.split(".")print(stng1)

…………………………..

from sys import argvprint("The length of command line arguments ",len(argv))print("The file name is ",argv[0])a=int(argv[1])b=int(argv[2])c=a+bprint("c =",c)

……………………

from sys import argvprint("The length of command line arguments ",len(argv))print("The file name is ",argv[0])print("type of 1st index is ",type(argv[1]))a=int(argv[1])b=int(argv[2])c=a+bprint("c =",c)

…………………

Command Line Arguments Arguments will be passed before executing the program in command prompt Arguments will considered or red in string format it will be in list 0 --> file name read the values from command line we use sys module and argv sys --> argv created a py file change the directory path to the file path in command line execute the python file by passing the arguments python cmdEx.py arg1,arg2.....argn

…………………………………….

a=100 # Global variabledef funName(): #a=10 # Local variables b=20 print(a)print("Value of a before calling function",a)funName()print("Value of a function calling function",a)……………………

x=100def myfunc(): global x x=75 print("inside the function",x)print("before calling function",x) myfunc()print("After calling function",x)

……………

x=100def myfunc(): global x x=75 print("inside the function",x)print("before calling function",x) myfunc()print("After calling function",x)……………………

x=25def fun(): global x x=200 return xprint("x value before function call ",x)a=fun()print("x value after function call ",x)import Externalprint("Val1 is ",External.Var1)print("Val2 is ",External.Var2)

……………………………..

try: a=eval(input("a is \n")) b=eval(input("b is \n")) div=a/b if div>1: print ("div ",div)except (NameError, TypeError, ValueError) as e: print(e)except ZeroDivisionError as z: print(z)#except SyntaxError as s:# print(s)else: print(div)finally: print("Program execution completed"),,,,,,,,,,,,,,,,,,,,,,,,,,

try: a=eval(input("a is \n")) b=eval(input("b is \n")) div=a/b if div>1: print ("div ",div)except (NameError, TypeError, ValueError) as e: print(e)except ZeroDivisionError as z: print(z)#except SyntaxError as s:# print(s)else: print(div)finally: print("Program execution completed"),,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

try: a=eval(input("a is \n")) b=eval(input("b is \n")) div=a/bexcept ZeroDivisionError as z: print("Error is ", z)#except NameError:# print("Sorry.. it is NameError")#except TypeError:# print("Sorry.. it is TypeError")else: print(div)finally: print("The program executed ")

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

try: a=eval(input("a is \n")) b=eval(input("b is \n")) div=a/bexcept ZeroDivisionError: print("Sorry.. the value is devided by zero")except NameError: print("Sorry.. it is NameError")except TypeError: print("Sorry.. it is TypeError")else: print(div)finally: print(" The program executed ")

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

<https://rollbar.com/blog/throwing-exceptions-in-python/#:~:text=The%20syntax%20error%20exception%20occurs,where%20the%20syntax%20error%20happened>.

try: a=eval(input("a is \n")) b=eval(input("b is \n")) div=a/bexcept (NameError, ZeroDivisionError, TypeError) as z: print("Error is ", z)else: print(div)finally: print("The program executed ")

,,,,,,,,,,,,,,,,

try: fl=open("tech1.txt",'r')except FileNotFoundError as e: print("Error is ", e)else: print(fl.read()) fl.close()finally: print("Done !!!")

………………………..

Scope of a Variable --> Local, Global and Externalglobal keywordif(): a=10 b=20else: c=1print(a) Exception Handling Try --> try catch --> except --> else finally --> finally,,,,,,,,,,,,,,,,,

fl=open('tech.txt','a')print("The name of the file ", fl.name)print("The mode of the file opened with ",fl.mode)fl.close()