19-9-2020

why do we need to give importance to the python?

---

programming lang and scripting

c/c++,java java script.perl....

codind design(html)example web browers

<html>.....</html> to design the web browers

1.python =coding +scripting

2.desktop with 3D animations

3.games

4.comapring with the programming languages ..python have very less coding

1989...python..guido van rossum...

ABC ....failure

2014 ..python

why we need to learn

.....

less coding

easy compilation

3D outputs

open source anyone download

high performance

comparative between java and python

Python download and install

1 8=

>>> called prompt

jvm,no compalation ,no API,no packages { ,}main method .....

installation and download

difference between java and python instructions

.....

where we are using this?

- ✓ 1.funcational programming (from c)
- ✓ 2.OOPS(OBJECT ORIENTED PROGAMMING LANG) (FROM C++)
- ✓ 3.Scripting lang (from perl,shell and module 3)
- ✓ 4.php,java ,.....

1+2+3+4=python

- Desktop applications
- Web applications ....Django
- database application
- networks
- games(3d)
- IoT
- data Analysis
- AI
- Machine learning

21-09-2020

Difference BETWEEN Javascript and python

Login page ……user name and password …button .design and validation

User name ..email

Password A@2dddd

Button ..click…action

Python design, validation, and action with dynamic outputs ….Dynamic Typed language

It won't take data types

Example: int, float …..

Syntax

Note: depends on versions symbols will be changed

1. # is used for comment section

Java comment section is //

C comment section is /*

Python comment section is # or $

Example :

X=10 # declaration

2. { and } there is no symbol in python

Instead of this we are using: colon

3. There is no; semi colon for ending the Statement in the python

Example :

X=10;y=10.2 (correct)

X=10;(correct statement)

Its expecting next python instruction

How we are reading the values

X=10 #reading the variable of x

Y=10.2 #reading the variable of y

Print(x) #printing the x value

10

Print(y) #printing the y value

10.2

Print(x,y) #printing the x,y value

10,10.2

Ex:

X=2;y=3;z=4;c=2.4;z=4J

Hear semi colon will divide the any type of value

4. Assignment variable

A=5 #integer

B=4.4 #float

C=1+6J #complex

Note we can assign a value without any data type

A=b=c=10 #multiple assignment (=) with same values

Print(a)

10

Print(b)

10

Print(c)

10

A,b,c=1,2.4,9 #multiple assignment with different values

The assignment operator(=) can assign the same and different values will assign to the compiler

1.no need to give(data type) in the user interface

2.while in the runtime(compilation) it will take as int, float…..etc

A user no need to give any data type instructions to the python code but runtime (python compilation) it will assign the data type without user interaction

Print(a,b,c)

1,2.4,9

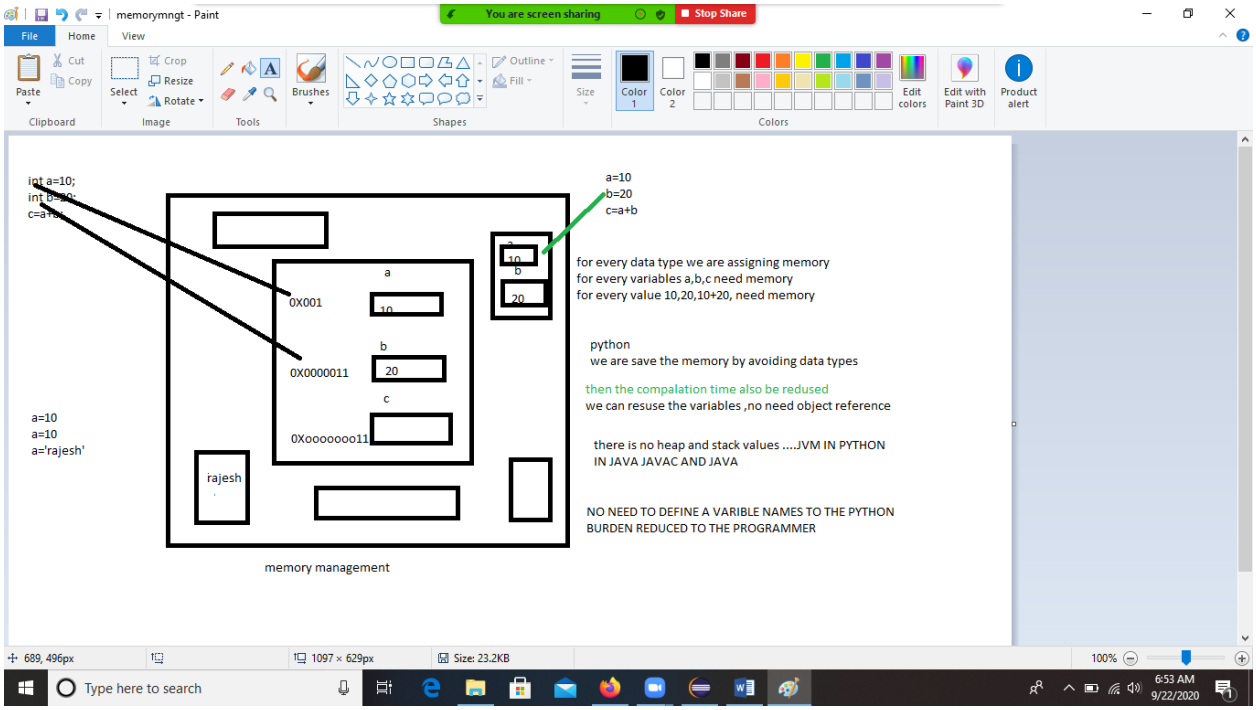
Expressions

Ex:

x=10

x=(y=X+1) is not allowed = no need to assignment operators

memory management



HOW many ways we can start the python

1.command prompt

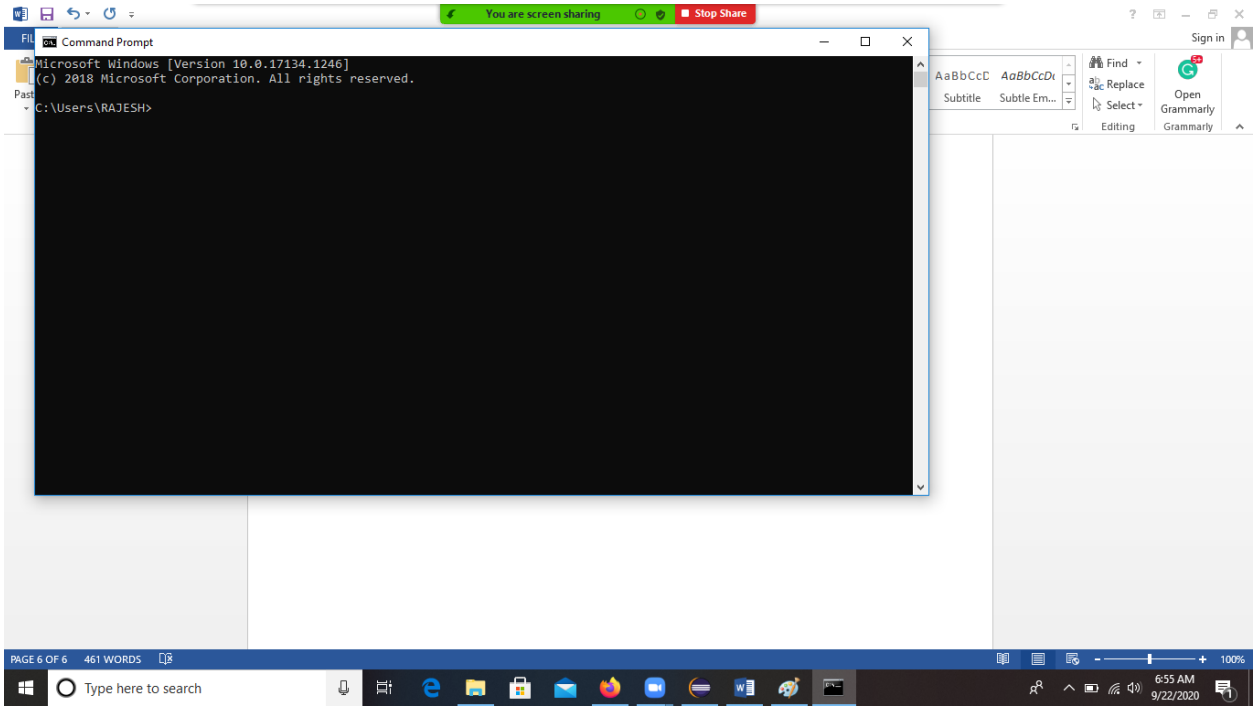
2.IDLE

4 WAYS TO SHOW THE PYHTON

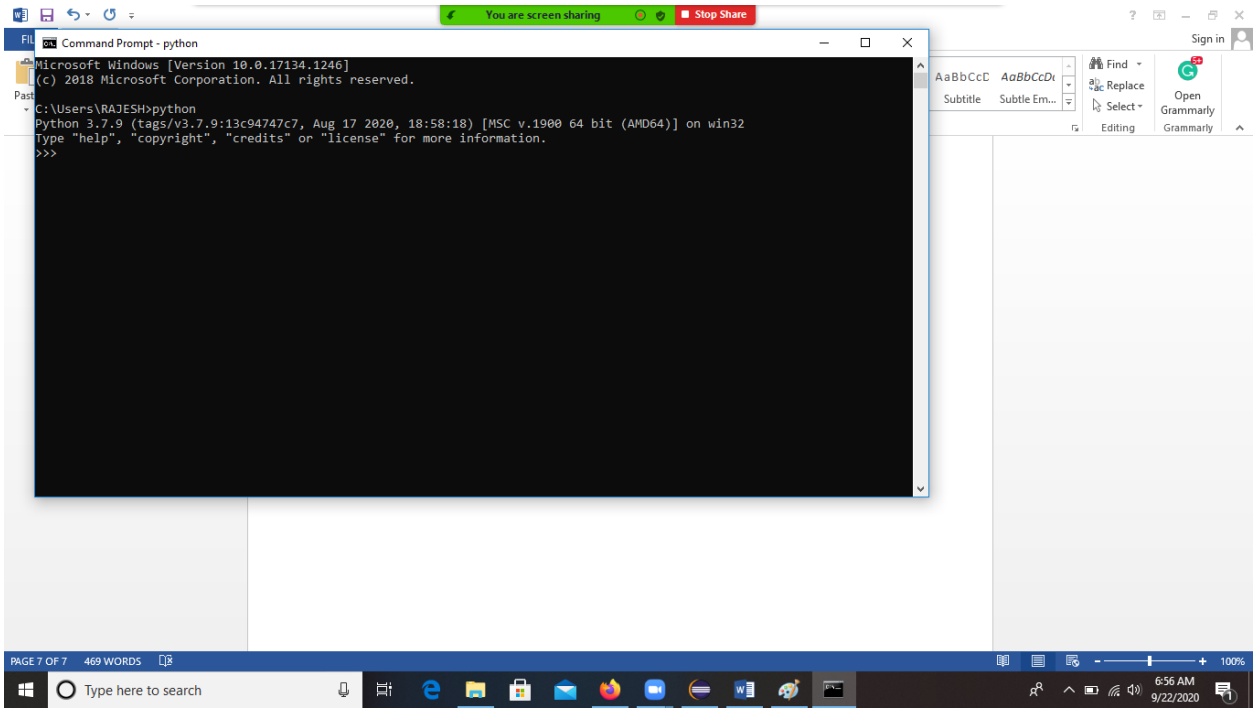
1.METHOD BY USING COMMAND PROMPT

STEP 1: TYPE CMD IN THE SEARCH

A new command prompt opened



Then type python



Method 2 by using note pad with command prompt

Select the folder path the choose note pad and type your program

And save the file as .py

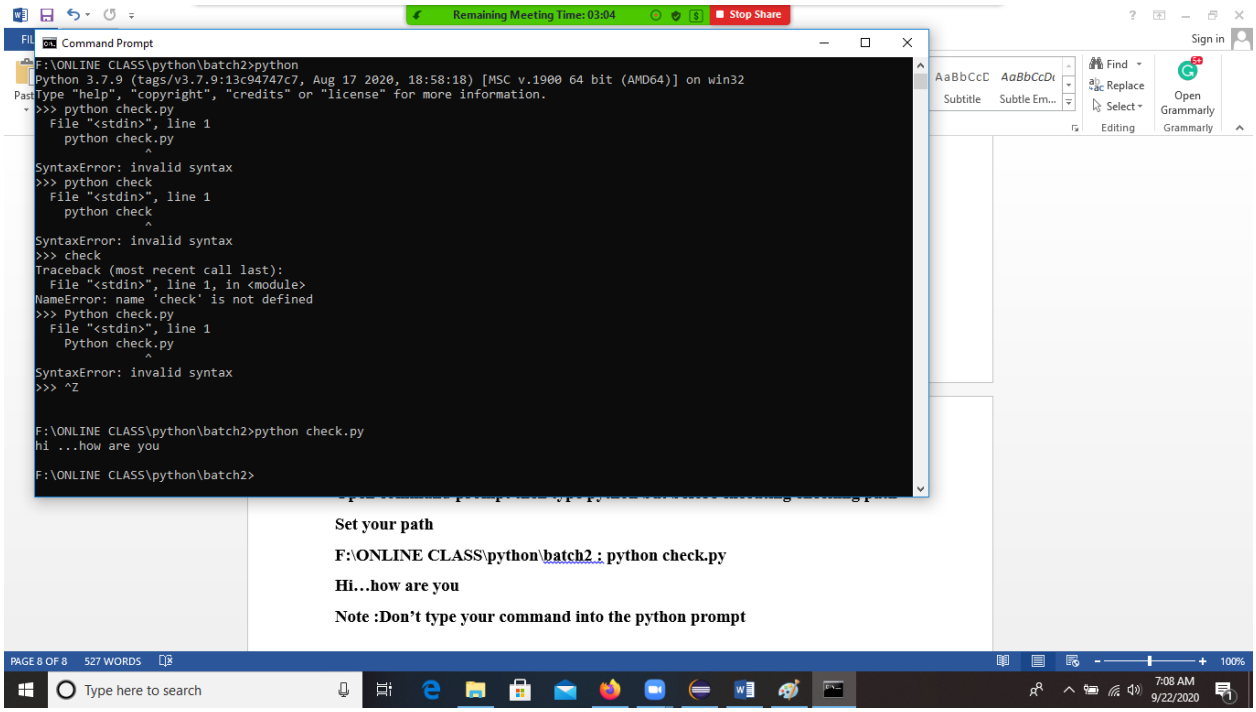
Open the command prompt then type python but before executing checking the path

Set your path

F:\ONLINE CLASS\python\batch2 : python check.py

Hi…how are you

Note: Don't type your command into the python prompt



E: change the directory

cd.. reduce the sub folders

cd add the folder

how to exit from the python command prompt ?

cntrl+z to exit in python

how to exit from the command prompt ?

exit command exit( )

22-09-2020

Numbers

Integer example X=12

Plain Integer Example: 12

Long

Yes 9999999999999999=Long

99999999999999999999999999999999999999999999999999999999=java

999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999L

Memory 8gb

Songs 8gb+8gb+8gb=24+imag+v+au+….

Python has a virtual space for Long

999999999999L

Floating point

X=1.2

Real numbers

o.23344

Complex numbers

X=1+2J

Real….print(x.real)

Imag…..print(x.imag)

Conjugate…print(x.conjugate)

Immutable number

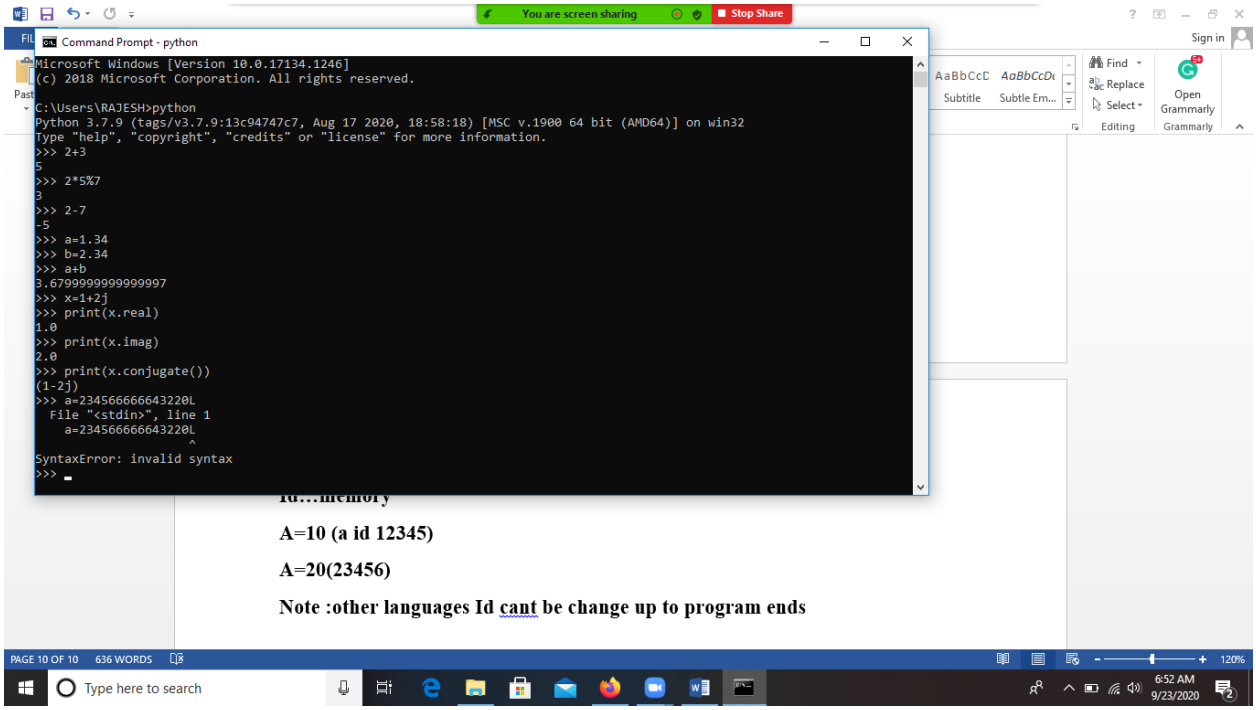
If you want replace the number the location of an identity(user or memory) will change

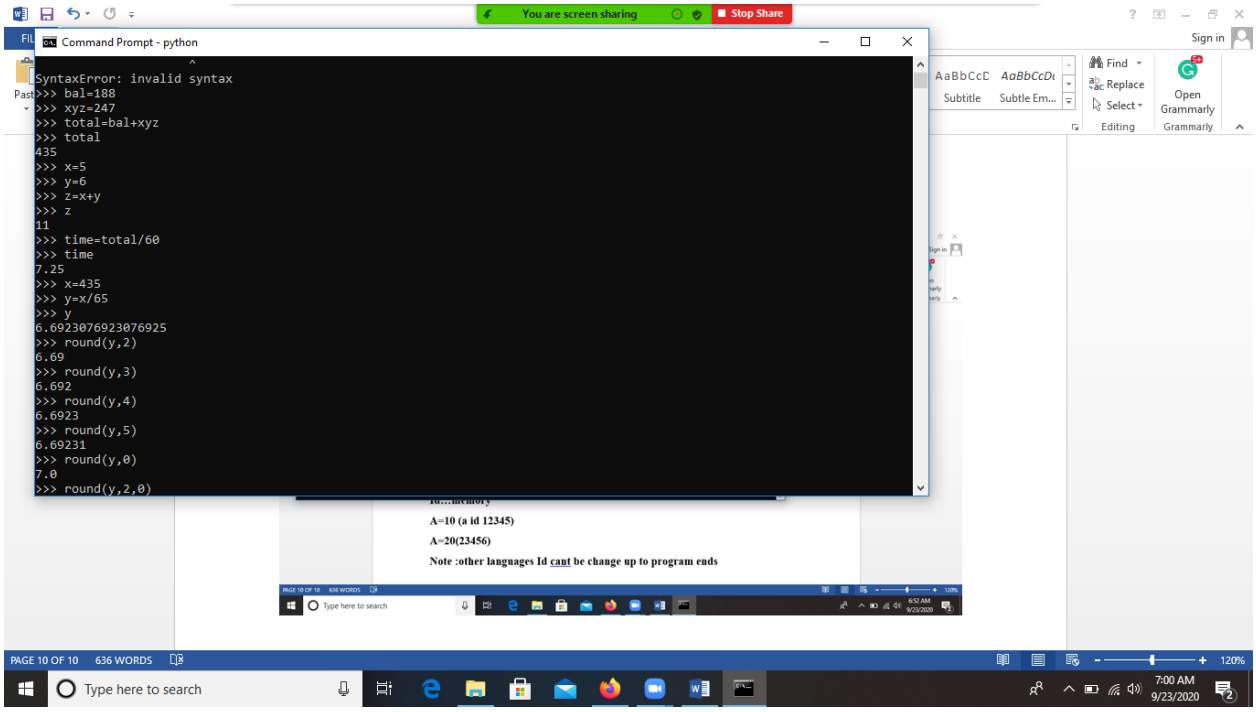
Id…memory

A=10 (a id 12345)

A=20(23456)

Note: other languages Id cant be change up to program ends



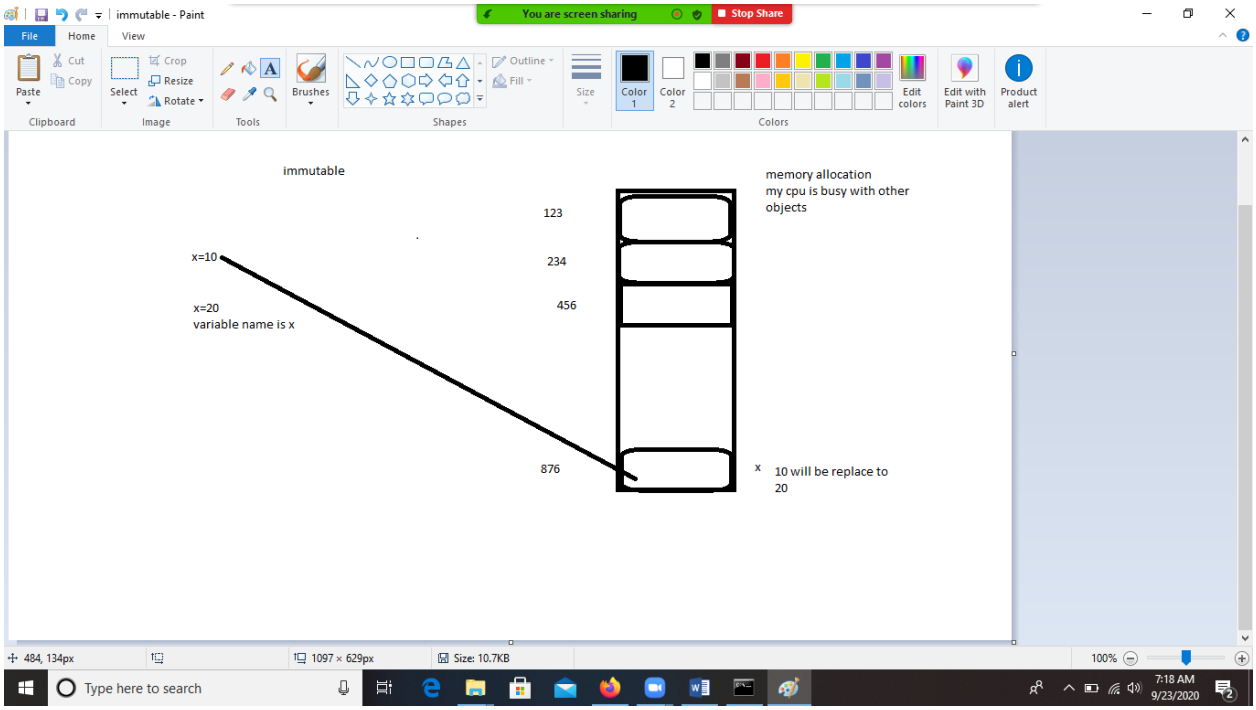


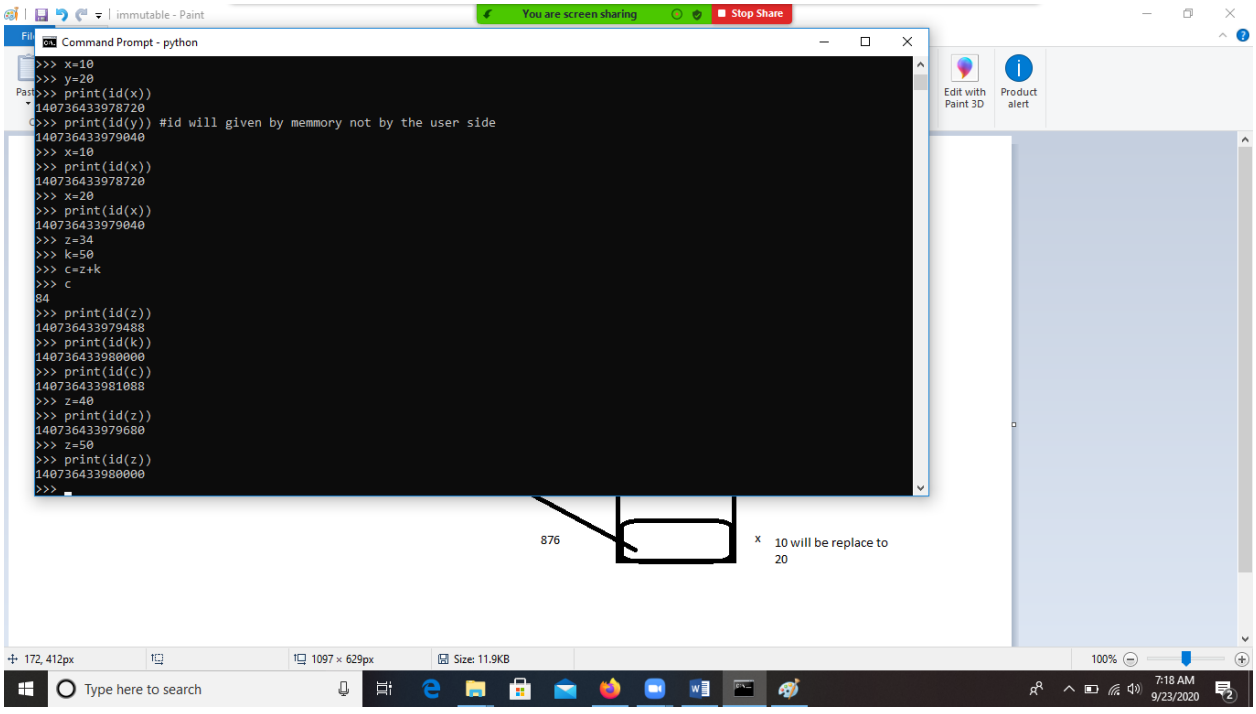
X=10 replace 20

1234

Y=20(234)

X=20


immutable - Paint
You are screen sharing
Stop Share
File
Home
View
immutable
memory allocation
my cpu is busy with other
objects
123
234
456
x=10
x=20
variable name is x
876
x
10 will be replace to
20
484, 134px
1097 × 629px
Size: 10.7KB
100%
Type here to search
7:18 AM
9/23/2020


immutable - Paint
You are screen sharing
Stop Share
Command Prompt - python
>>> x=10
>>> y=20
>>> print(id(x))
140736433978720
>>> print(id(y)) #id will given by memmory not by the user side
140736433979040
>>> x=10
>>> print(id(x))
140736433978720
>>> x=20
>>> print(id(x))
140736433979040
>>> z=34
>>> k=50
>>> c=z+k
>>> c
84
>>> print(id(z))
140736433979488
>>> print(id(k))
140736433980000
>>> print(id(c))
140736433981088
>>> z=40
>>> print(id(z))
140736433979680
>>> z=50
>>> print(id(z))
140736433980000
>>>
876
x
10 will be replace to
20
172, 412px
1097 × 629px
Size: 11.9KB
100%
Type here to search
7:18 AM
9/23/2020

Exercise

(1) Find out an area of a triangle whose base is 15 meter and height is 22 meter. The mathematical equation for an area of a triangle is:
Area = ½*Base*Height

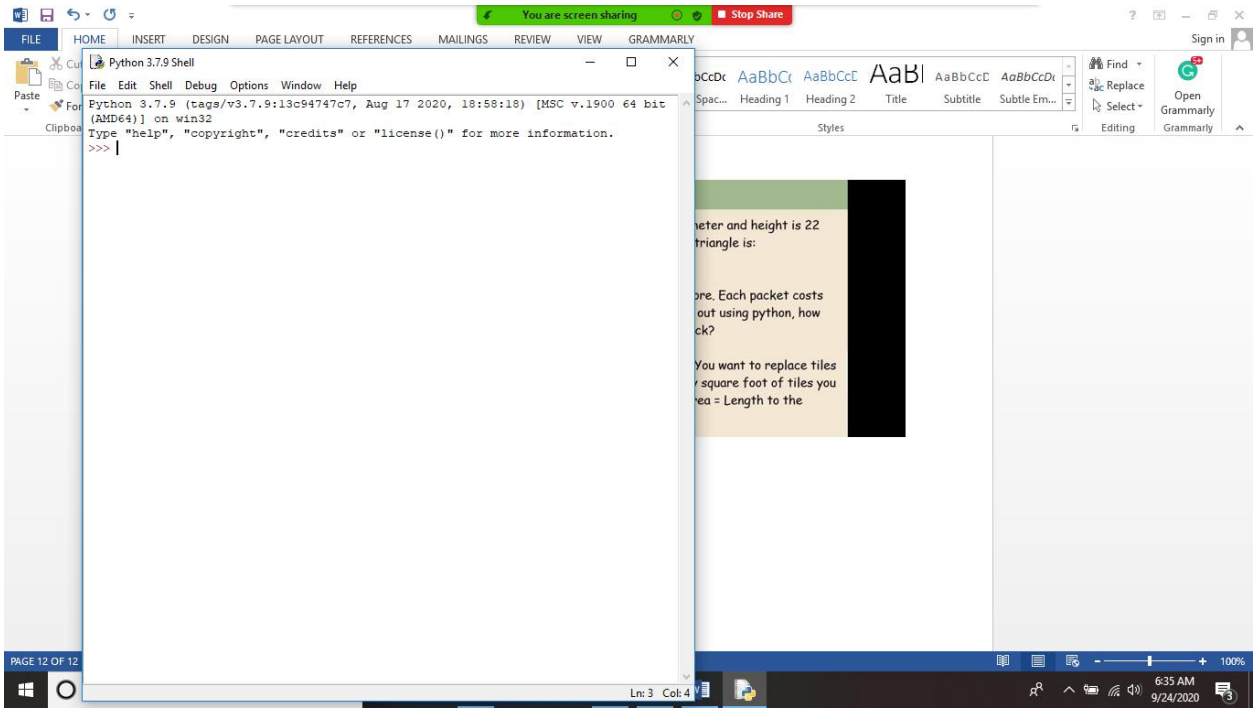
(2) You bought 9 packets of potato chips from a store. Each packet costs 1.49 dollar and you gave shopkeeper 20 dollar. Find out using python, how many dollars is the shopkeeper going to give you back?

(3) The bathroom of your home is an exact square. You want to replace tiles in it. Length of this bathroom is 5.5 feet. How many square foot of tiles you need to buy? Equation for an area of a square is: Area = Length to the power of 2.

2.method by using IDLE

STEP 1: CLICK ON WINDOW BUTTON→python

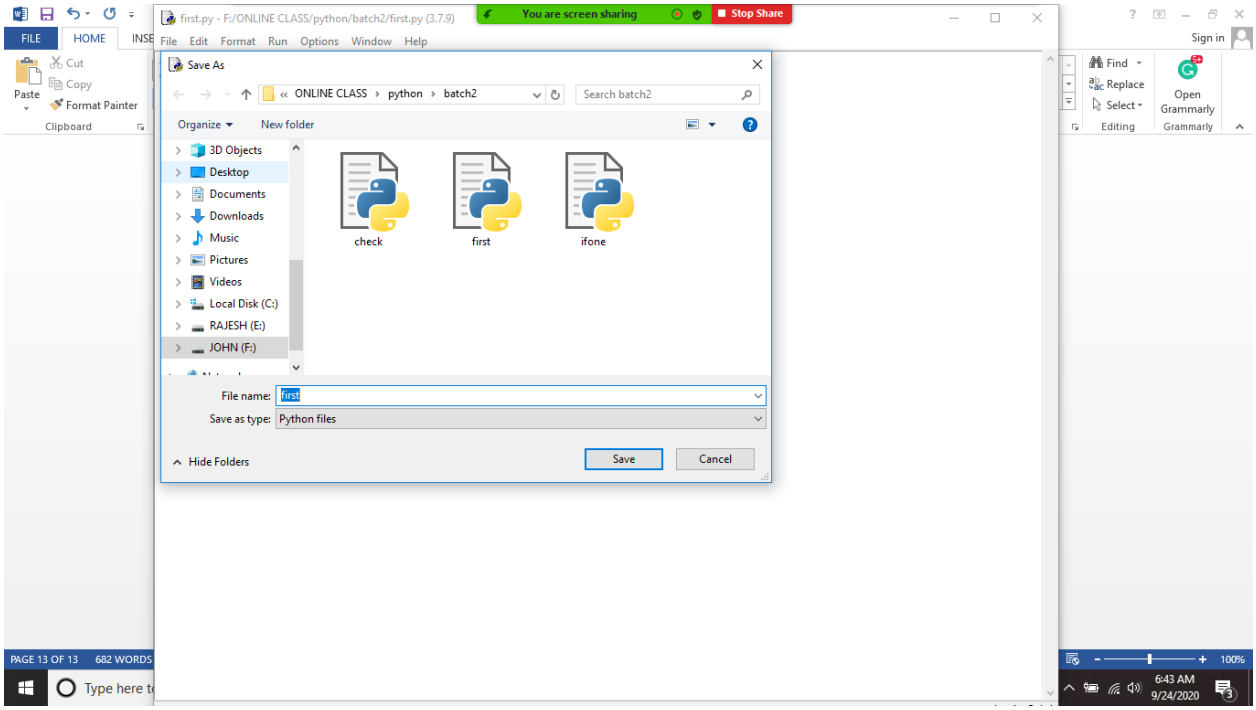
Choose IDLE new shell will be appear



Then check the configurations

Select new→choose the new file →new sheel window will appear then type your program

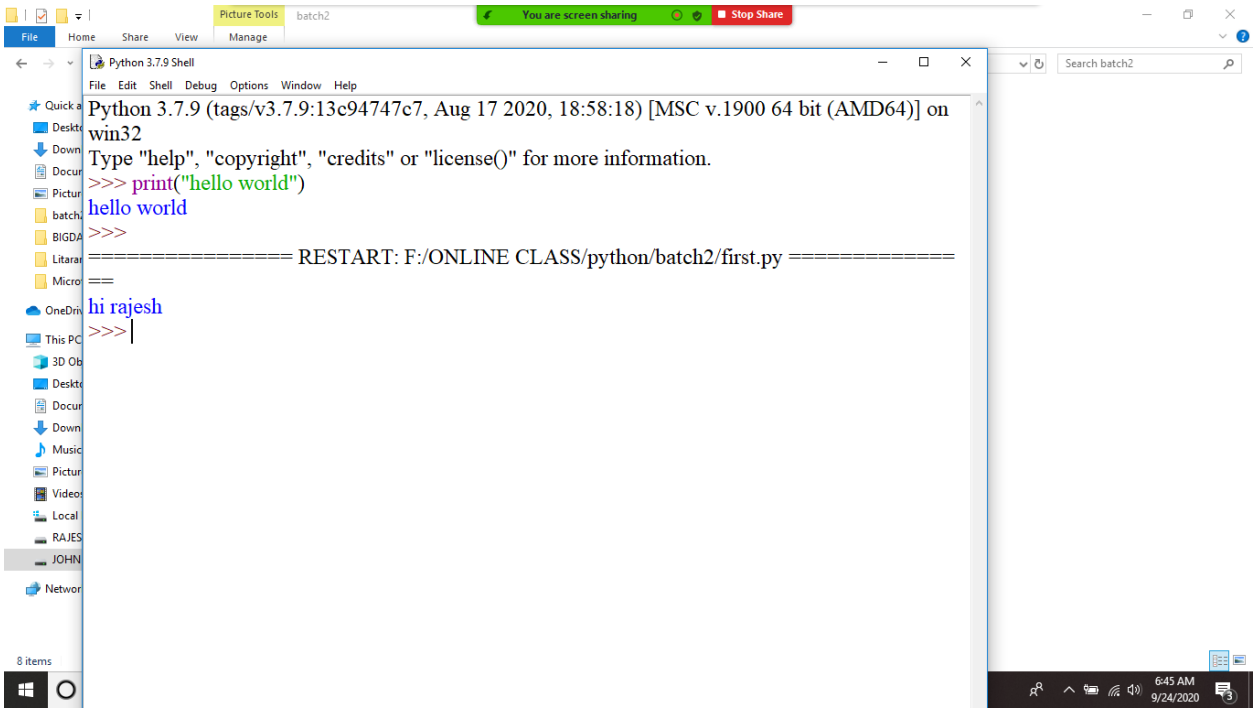
File →new →save →choose the directory path



Save the file name with .py

To execute the program press F5 or run→run module

Then out put will be appear in sheel



Conditional statements

IF_else:

Decision making is the most important aspect of almost all the programming languages.

If→the statement is used to test the specific condition. If the condition true, a block of code will be executed

Else: the remaining code will be executed.if the if statement goes wrong

Elif: for the nested if condition we are using elif .

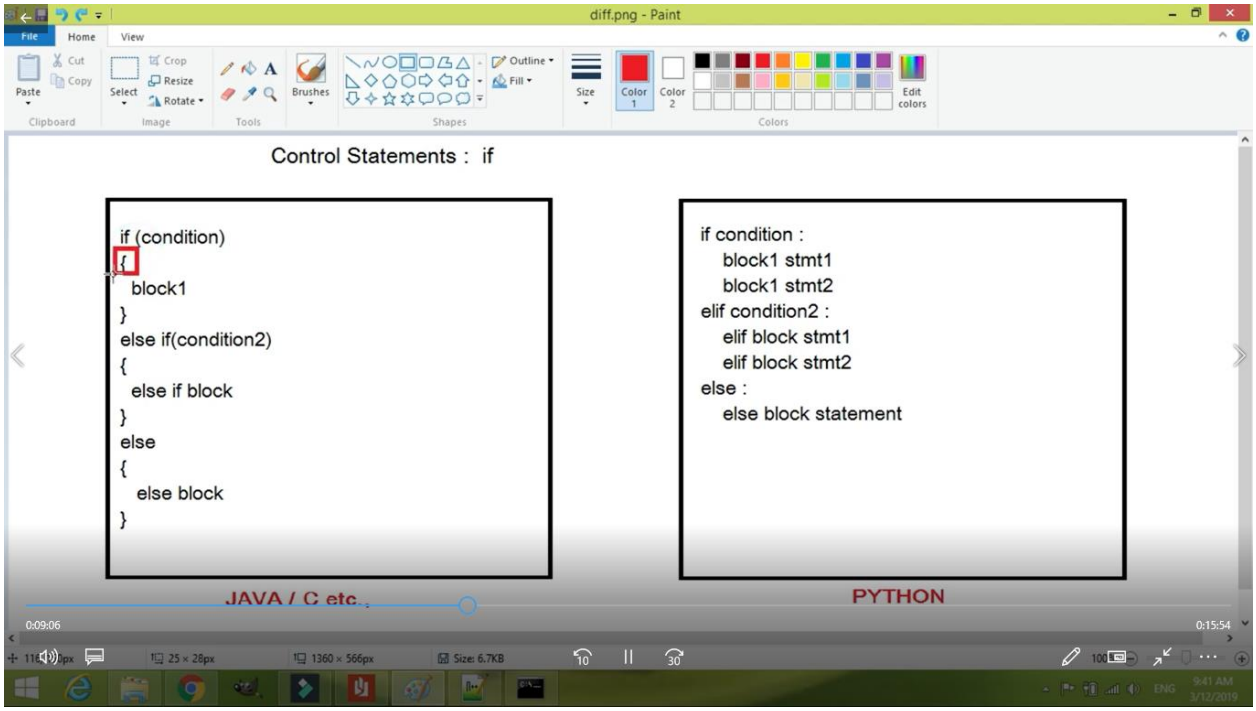
If (a=men)

If (age<60)

If(state==ap)

If (count==india)

…….



Syntax { }, : ( ), no nested if →elif

If condition:

Statement1

Statement2

elif :

Statement1

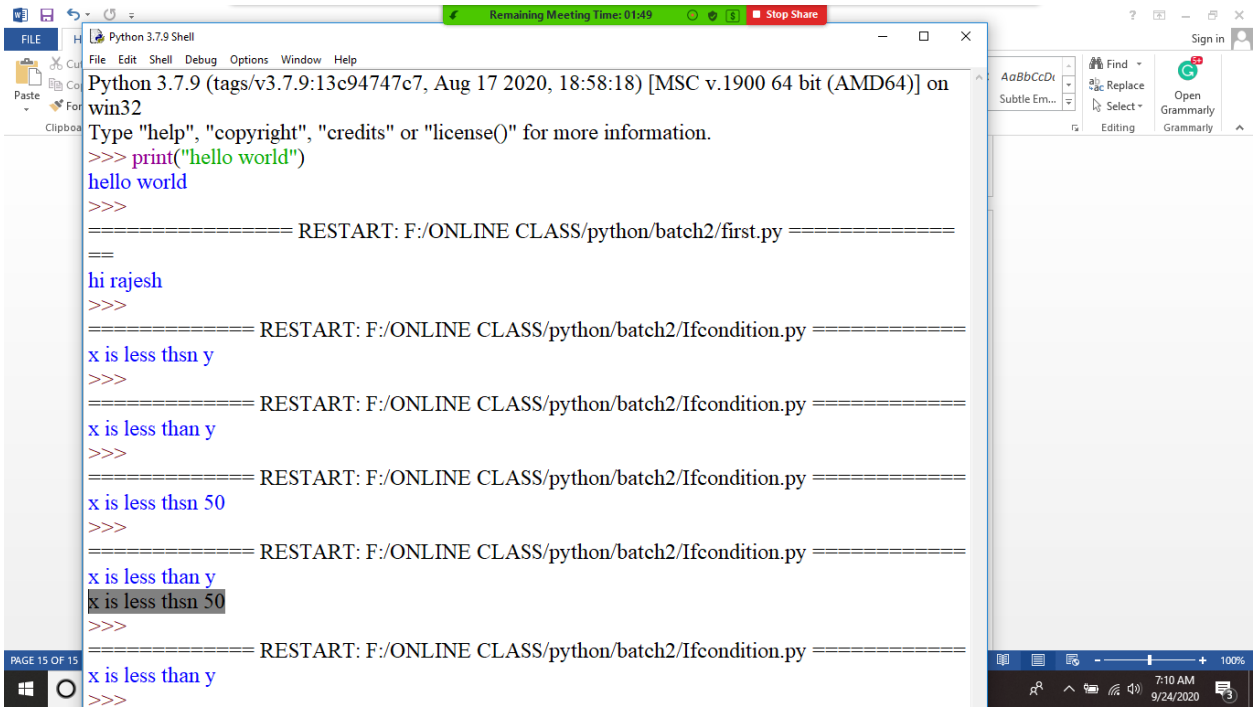
Statement2

Statement3

else:

Statement

```
# If-else condition for10 and 20
x=10;y=20 # variable declaration
if  x < y :
    print('x is less than y')
elif x==y:
    print('x is equal to y')
elif x<50:
    print('x is less thsn 50')
else :
    print('y greater than x')
    # the elif statement allows you to check multiple expressions for TRUE
    #and execute a block of code .
#diffrence between if-else java and python
```



Program 3 : to print largest of three numbers by using runtime environment

```
#largest of 3 numbers
#largest of 3 numbers
```

```
#a=10,b=30
a=  int(input ('enter a value::'))
b=int(input('enter bvalue::'))
c=int(input('enter c value::'))
if a > b and a > c:
   print(a,' is largest')
if b>a and b>c:
   print(b,'is largest')
if c>a and c>b:
   print(c,'  is largest')
```

output:

enter a value::2

enter bvalue::5

enter c value::1

5 is largest

Program 4: test runtime environment with variables

Case1:

```
#a=10,b=30
a=  input ('enter a value')
b=input('enter next value')
print(a,b)
print(a+b)
```

output:

enater a value : 10

enter next value : 30

1030→ concatenation

Python expecting data type for the user output

Case 2:After changing the int as data type for the runtime

```
#largest of 3 numbers
#a=10,b=30
a=  int(input ('enter a value'))
b=int(input('enter next value'))
print(a,b)
print(a+b)
enter a value10
enter next value30
10 30
40
```

Program 5: find whether the number is even or not by using python

Runtime numbers number : 4

Logic …..4 %2==0 if number%2==0:

True

8%2==0

9%2=1

13%2=1

```
number=int(input("enter a number::"))
if number%2==0:
   print(number,"is even number")
else:
      print(number,"is odd number")
```

output:

```
enter a number::10
10 is even
```

Program 6: wpp to find grades

85 to 95→ a

m>85 and m<=95…>A

m>75 and m<=85-→b

m>60 and m<=75-→c

m>50 and m<=60-→pass

else

fail

```
# marks grade
m=int(input("enter your marks::"))
if(m>85 and m<=95):
    print("congratrss! youscored first ranl")
elif(m>75 and m<=85):
    print('you scored B grade')
elif(m>60 and m<=75):
    print('you scored c grade')
elif(m>50 and m<=60):
    print('you scored D grade')
else:
    print('you failed .....')
```

for loop

0,1,2,3,4,…….100

2*1=2

2*2=4

2*3=6

……

2*20=40

The for loop in python is used to "iterate the statements or a part of a program several times .its is frequently used to traverse the data Structure like List, tuple, or directory

Syntax:

```
For iterating _var in sequence:
    Statement(s)
```

For sequence we are using a method called "range( )"

range(5)------→0 to 4

range(5,10)--→5,6,7,8,9

****Index starts with 0**

range(0,10,3)→ 0,0+3,3+3,6+3

0,3,6,9

range(1,8,2)→1,1+2,3+2,5+2

1,3,5,7

0 is starting index

10 is limit of the index

3 is increment value

Program 7: wpp to find multiplication table by using for

2*1=2

2*2=4

2*3=6

……

2*20=40

3,"*",a, "= ",3*2

, is the concatenation of print method

#table

```
x=int(input('enter your choice table::'))
for a in range(1,11):
    print (x,"*",a,"=",x*a)
```

output:

enter your choice table::3

3 * 0 = 0

3 * 1 = 3

3 * 2 = 6

3 * 3 = 9

3 * 4 = 12

3 * 5 = 15

3 * 6 = 18

3 * 7 = 21

3 * 8 = 24

3 * 9 = 27

3 * 10 = 30

26-9-2020

For loop methods or functions

Break: terminates from the loop

For I in range(0,10):

Print(I) 0,1,2,3,…..9

If(i<=3):

Break or continue:3 or 0,1,2,3 skip to be test

Print(I)…0,1,2,3

Continue: it skips the execution of the current iteration

Pass:

If (condition):python

Pass

If(condition) java/c

(

//empty

}

Program 8: wpp to test for loop methods

```
for i in range(10):
        if(i==3):
                break
        print(i);
```

out:

0

1

2

Case 2: continue

0

1

2

4

5

6

7

8

9

Case 3:pass

0

1

2

3

4

5

6

7

8

9

Case 4: if(1<=3)

Continue : 4,5,6,7,8,9

Break

Pass: 0,1,2,3……9

Program 9: nested for loop by using Python

		•		
	*		*	
*		*	*	

For

I=0 range o, num…..intilization

J 0,num-i-1…….>j values

Print(end=” ”)

J 0,i+1……concatenation

Print(“*”,end=””)

27-09-2020

Input :5,6,7,	**J=input**	
Gaps and *	**9,7,5,3,1…*,**,***,****	
Print	**Gaps and stars**	
print	**Lines**	

J=9…gaps

For I in range (1,10,2): # i=1,2,3…

Print(‘ ‘*j+i* ‘*’)

‘ ‘→gaps to be print

***J+i*→gaps to j and I relationship….+,-,%,*…**

‘*’….pattern to be print

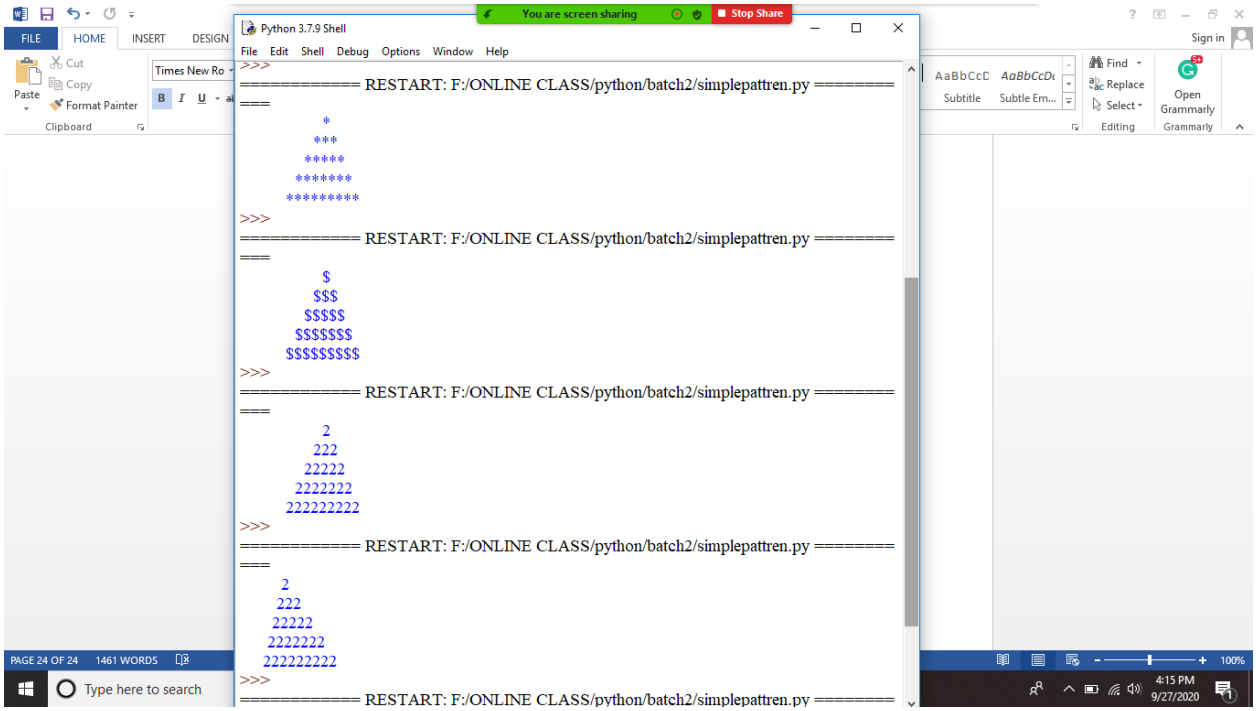
J=j-1-→decreasing

```
j=9
for i in range(1,10,2):
    print('  '*j+i*'2')
    j=j-1
```

Or

```
rows=int(input("enter rows::"))
for i in range(0,rows):  #rows
    for j in range(0,i+1):  #columns row and column relationship
        print("*",end='  ')
```

print(" ")



Python Literals

It can be defined as data that is given in a “variables or constant”

1.String literals

2.numaric literals

3.Boolean

4.Special

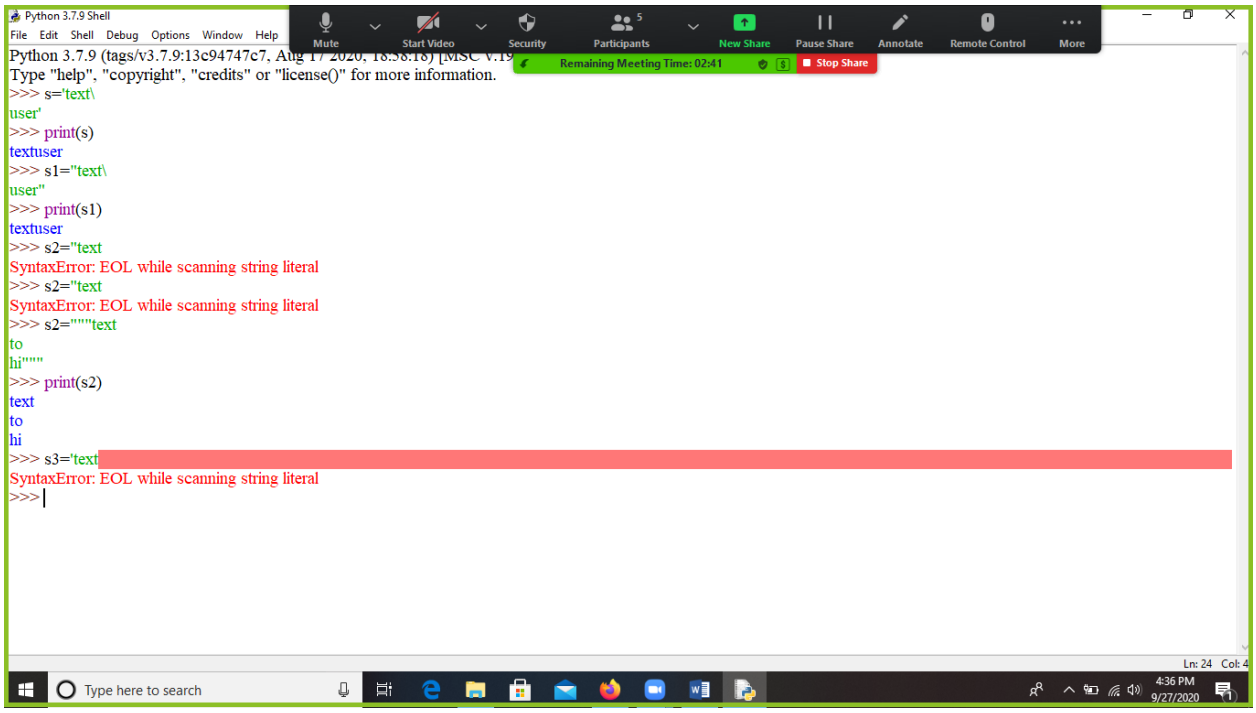
5.literal collections

1.String literals

It can be formed by enclosing a text in quotes

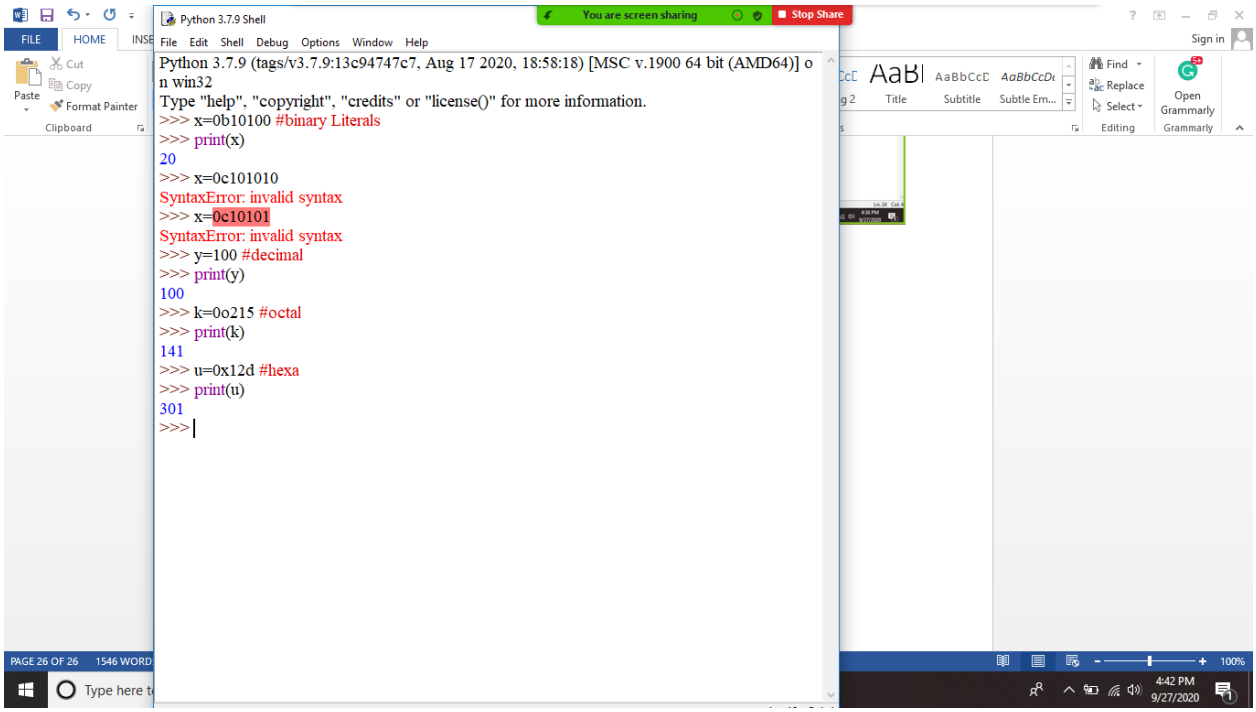
1.single:String that are treated within a single line

2.multi: a Piece of text that is written in multiple



2.numaric literals

3.



3.Boolean Literals

#boolean

```
x=(1==True)
y=(2==False)
z=(3==True)
a=True+10
b=False+10
print("x is",x)  #True
print("y is",y) #false
print("z is",z) #true
print("a is",a) #11
print("b ia ",b)#2+10
```

output :

x is True

y is False

z is False

a is 11

b ia 10

4.Special Literals

Python contains None to specify that the field is not created

End of →None ……[lists]

X=None

Print(x)

None

5.Literal collections

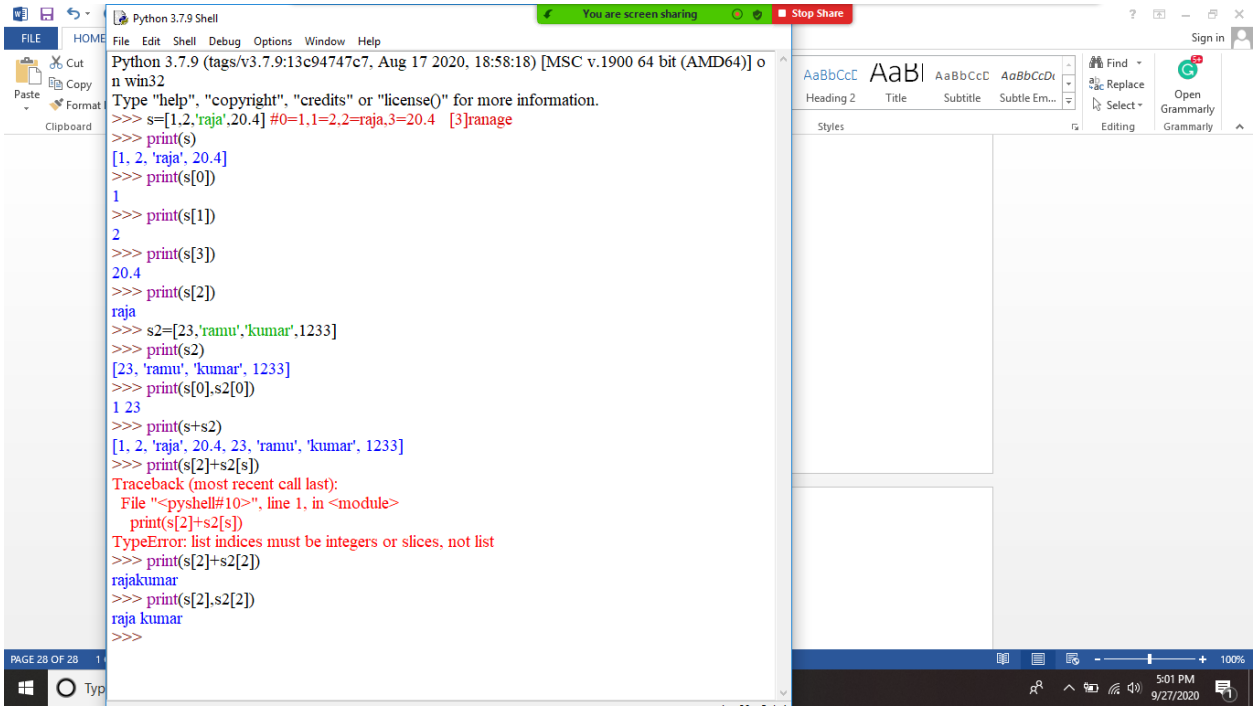
List: **is collection of items and different data types**

Value is stored in a list

Every values is separated by” , “

Enclosed with []

Example : s=[1,2,’raja’,20.4]



29-09-2020

How to compare two list

Rajesh_1=[121,"rajes",2000]

Rajesh_2=[121,"rajesh",2000]

Rajesh_1==Rajesh_2----→false

How:

[0]==[0]

[1]==[1]…>false

[2]==[2]

- ✓ **Index starts with "0"**
- ✓ **Storage of values from the memory**
- ✓ **The list are ordered**
- ✓ **The list are mutable**
- ✓ **The list can store the number of various elements**

Forward direction

0	1	2	3	4	5
raj	1	234	kumar	rani	200.23

-6	-5	-4	-3	-2	-1

Backward directions

Define list: write the python code (list)

List=[]→it goes to memory allocation

Example :

List=[“raj”,1,234,”kumar”,”rani”,200.23]

List[0]=raj…….

Size of list[5]

List[0:]

List[:]

List[2:4]

list[:3]

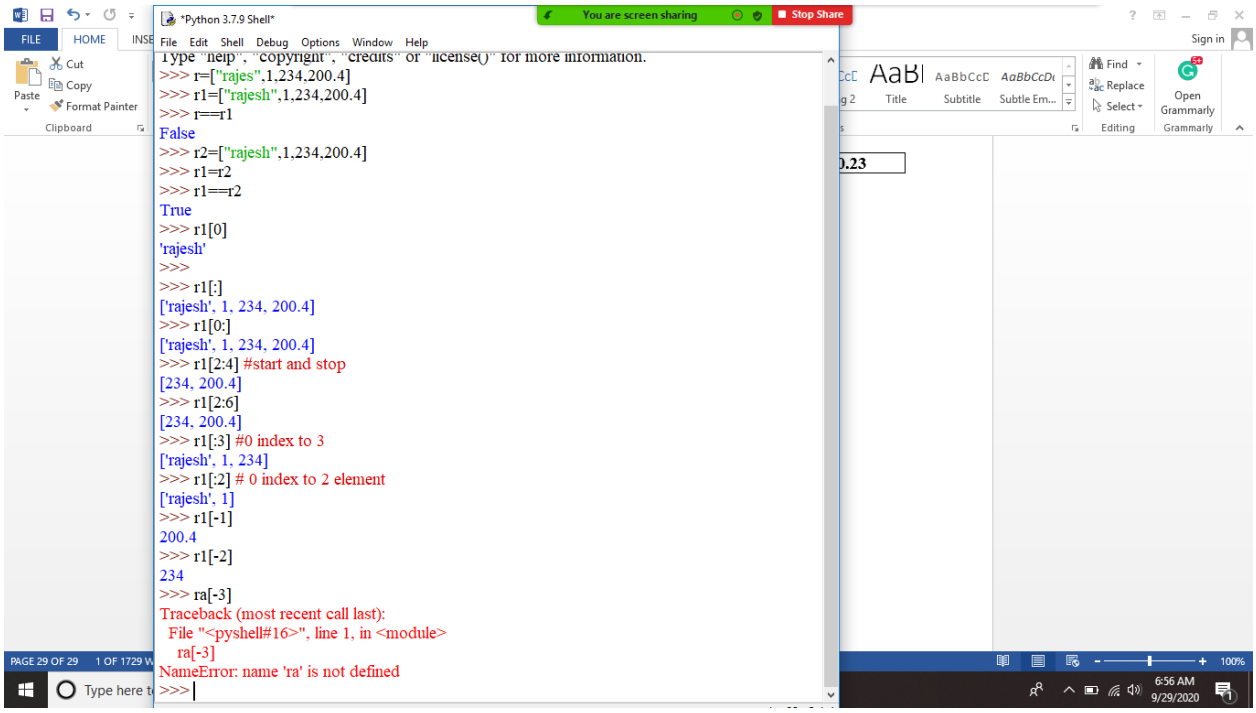
List[-1]

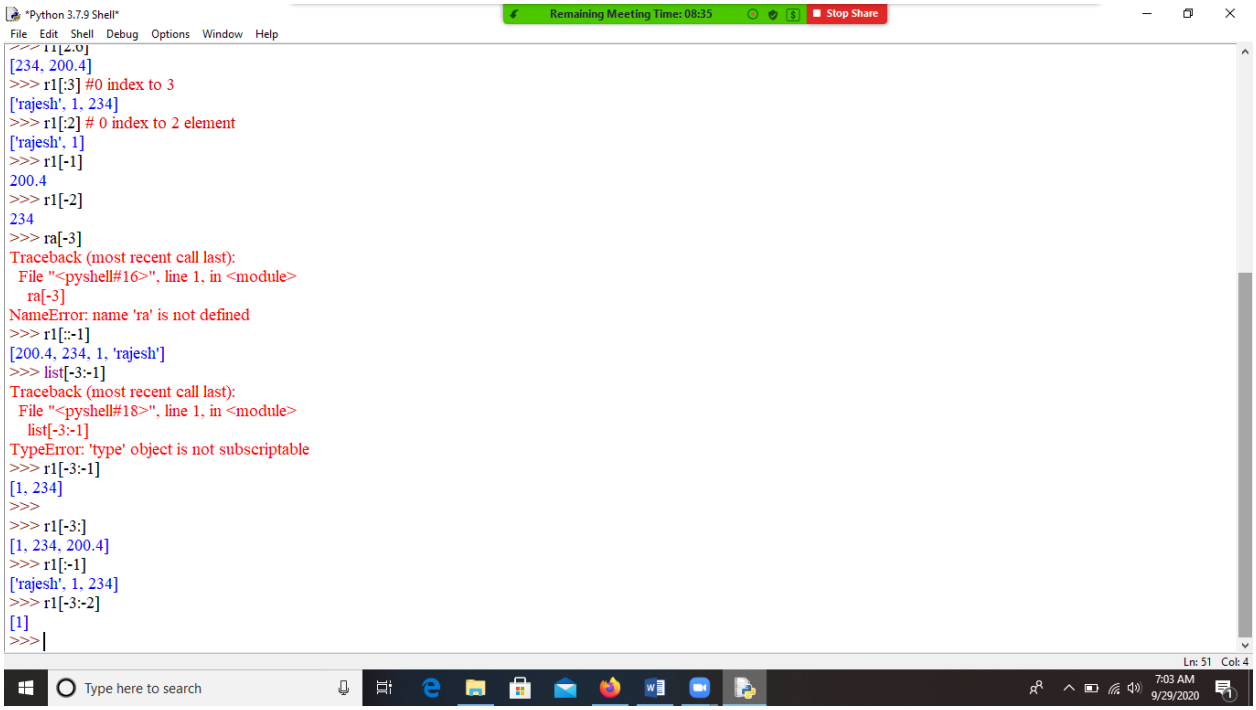
List_value[start:stop:step]

Start: starting index position

Stop: last index position

Step: skip the Nth element within start: stop





Methods in list

- ✓ Insert()
- ✓ Append()
- ✓ Delete()
- ✓ Remove()
- ✓ Reverse()
- ✓ Add
- ✓ Can we use it?

Program 11: Wpp to create empty list and fill the elements one by one

```
list=[] #empty list
n=int(input("enter new elemenmt::"))  #runtime list values
for i in range(0,n):
    list.append(input("eneter append value::")) #add value to list (upto end value)
    print("the index value::",i) # index value
for i in list:
        print(i,end=" ")
```

30-9-2020

Program 12 : wpp to remove elements from the List

remove()

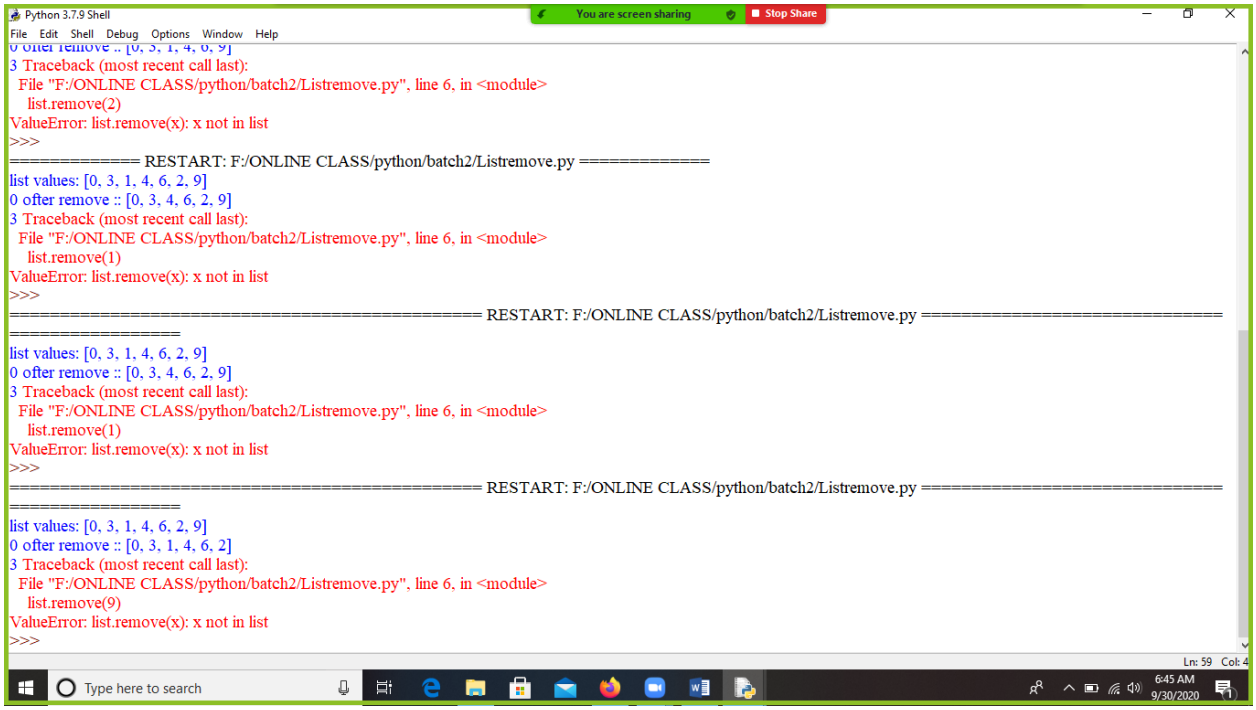
create the list

print the list

then remove the element

list.remove(index_value)

```
#remove elements from the lis
list=[0,3,1,4,6,2,9]
print("list values:",list)
for i in list:
    print(i,end=" ")
    list.remove(9)
    print("ofter remove ::",list)
for i in list:
        print(i,end=" ")
```



Python built in functions :

Cmp(list1,list2)→compares the elements of both the lists

Len(list)→length

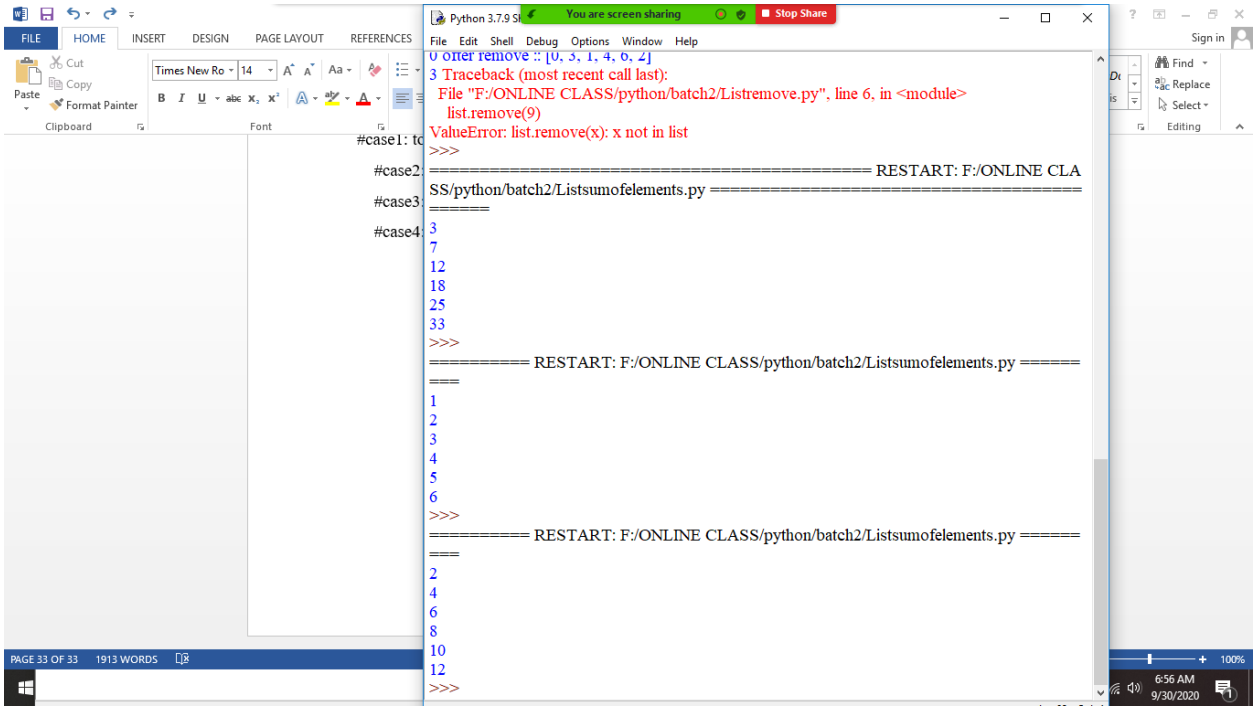
Max(list)→maximum element

Min(list(list)→minimun element

List(seq)→it returns any sequence to the list

#wpp to find the sum of the elements

```
s=[3,4,5,6,7,8]
total=0
for i in s:
    total=total+i # 0+3, 3+4,7+5,12+6,18+7,25+8
    print (total)
#case1: total=total+1
    #case2: total=total+2
    #case3: total=total-1
    #case4: total=total-2
```



Program 13 : wpp to remove duplicate elements

S1=[1,1,2,2,2,2,2,3,4,5,5,5,6,7,8,8]

S2[]

For I in lis1:

If I not in s2:

List.append(i)

Print(s2)

#how to remove duplicates

```
s1=[1,1,2,2,2,2,2,3,4,5,5,5,6,7,8,8]
s2=[]
for i in s1:    # 1 to 8
    if i not in s2: #not
        s2.append(i) # adding to the list
print(s2)
```

output:

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

Program 14 : Wpp X=[] empty list then print the values 0,2,4,6,8

X=[] empty list then print the values 0,2,4,6,8

For

……

X

[0,2,4,6,8]

```
x=[] # empty list
for k in range(0,9,2):
    x.append(k)
    print(k)
```

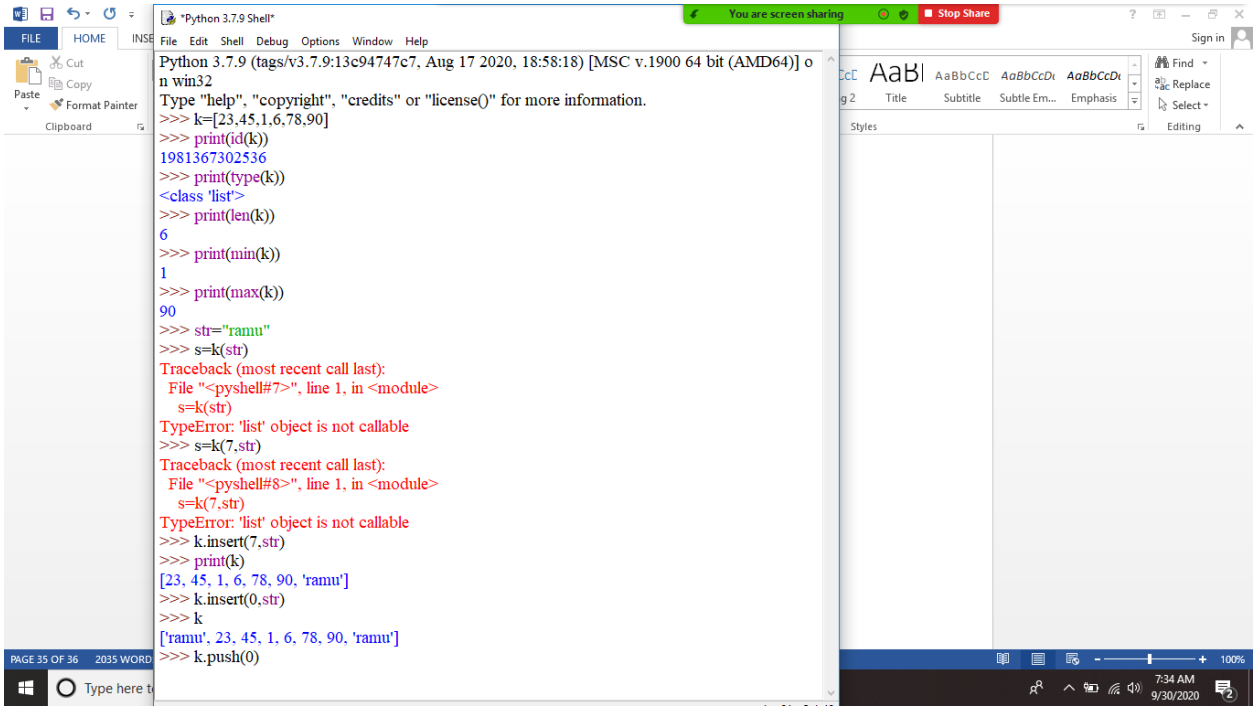
program 15 :wpp to compare the elements from lists

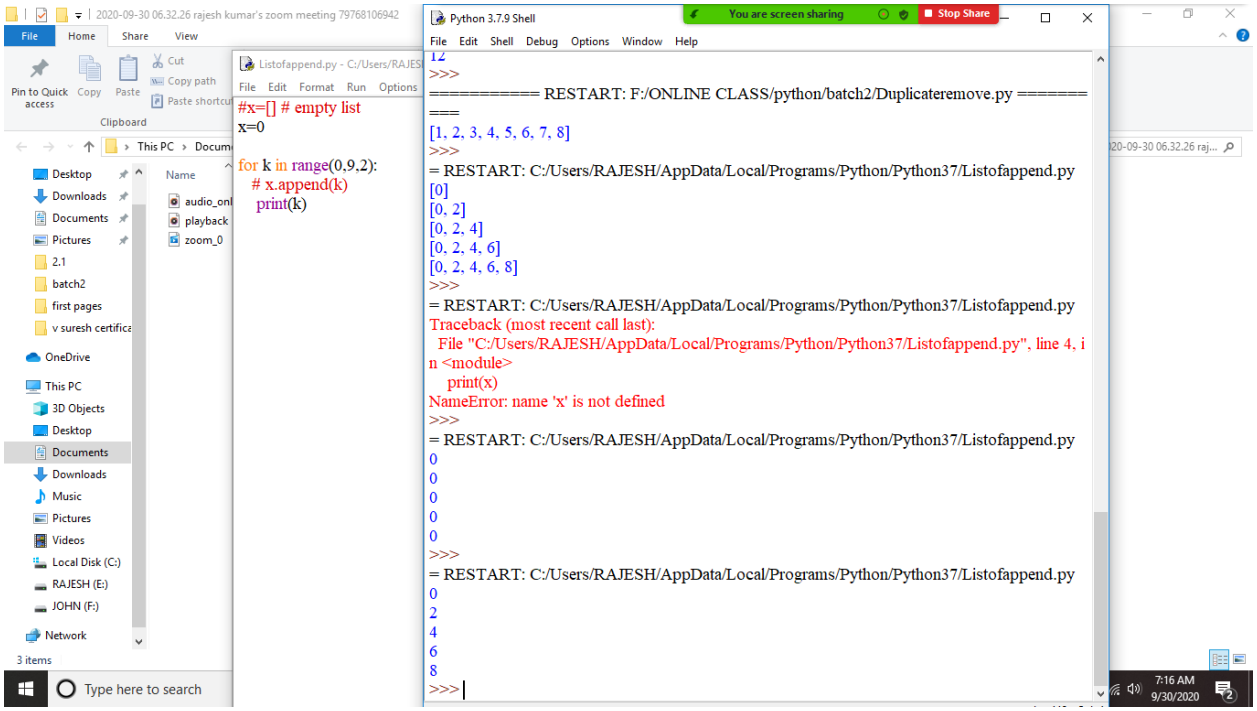
```
s3=[1,2,5]
s4=[1,5,7]
for x in s3:  # 1,2,5
    for k in s4: #1,5, 7
        if  x==k: # 1,5 it will the elements
            print(x)
```

output :

1,5

Built in methods





Reverse

k=[1,2,3]

>>> k.reverse()

>>> k

[3, 2, 1]

1-10-2020

<u>String</u>

String is a collection of Characters surrounded by single, double and triple quotes .

0 and 1---by computer

Character---→java or c ????

Python --→1 byte

No need to use the character in a string

Why: python doesn't need any data type.

When we need data type??

The python returns the output to the user then for user understand they need data types

X=int(input("enter a value::"))

10

X=10

In python, the string can be enclosed by the character or sequence of words in quotes

How to create Strings

X=" hello"

X[0]----------→

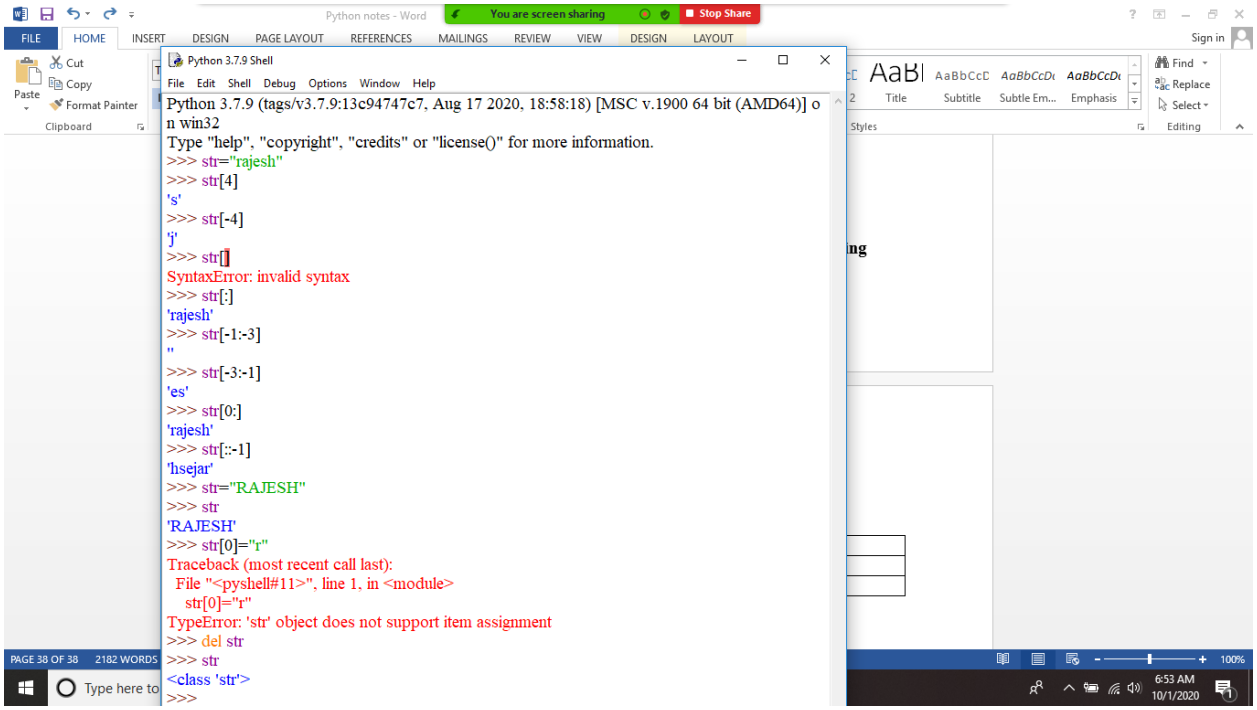
X[1]

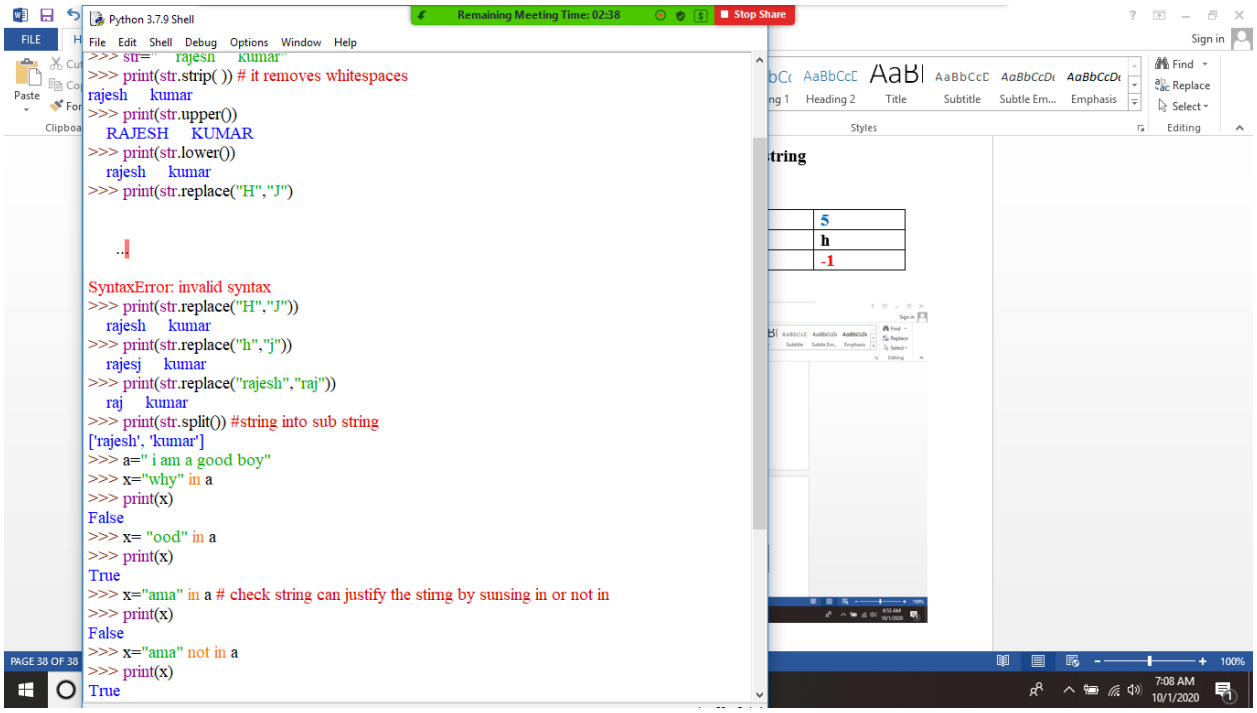
[]→slice operator used to access the individual characters to the string

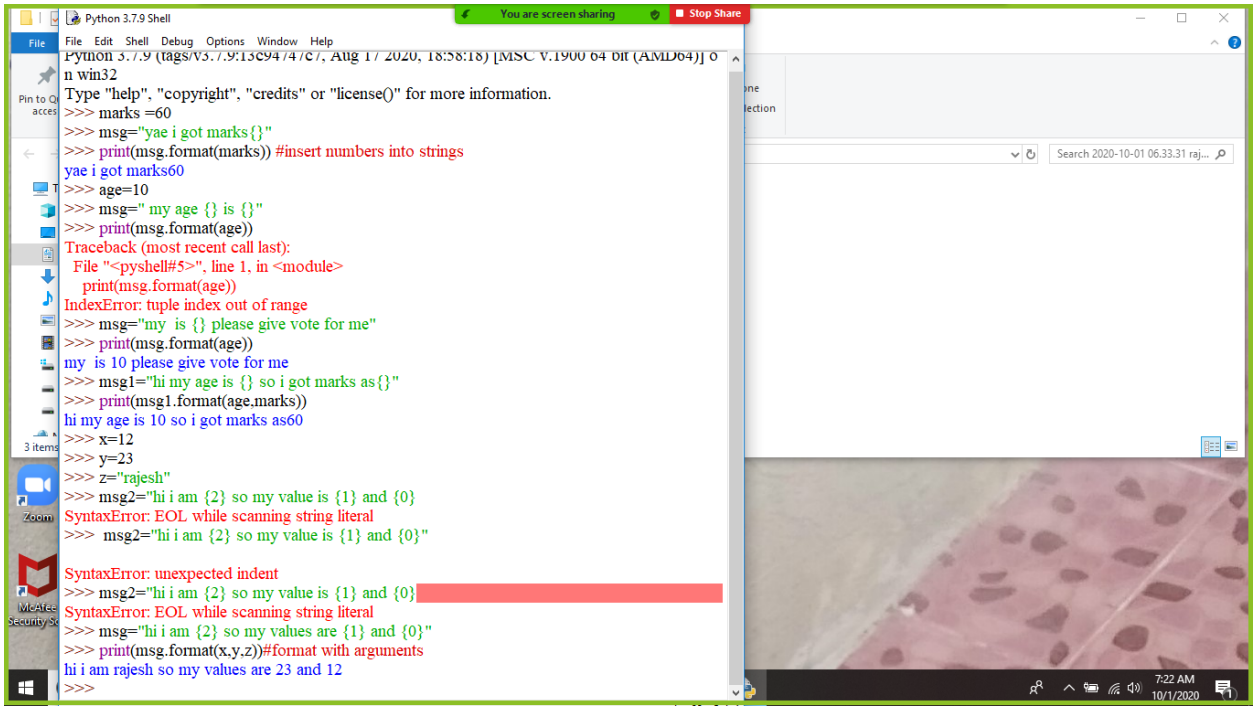
[:]→colon is used to find substring from the given main string

[::-1]→

0	1	2	3	4	5
r	a	j	e	s	h
-6	-5	-4	-3	-2	-1


Python notes - Word
You are screen sharing
Stop Share
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> str="rajesh"
>>> str[4]
's'
>>> str[-4]
'j'
>>> str[]
SyntaxError: invalid syntax
>>> str[:]
'rajesh'
>>> str[-1:-3]
''
>>> str[-3:-1]
'es'
>>> str[0:]
'rajesh'
>>> str[::-1]
'hsejar'
>>> str="RAJESH"
>>> str
'RAJESH'
>>> str[0]="r"
Traceback (most recent call last):
File "<pyshell#11>", line 1, in <module>
str[0]="r"
TypeError: 'str' object does not support item assignment
>>> del str
>>> str
<class 'str'>
>>>
PAGE 38 OF 38
2182 WORDS
Type here to
6:53 AM
10/1/2020


Python 3.7.9 Shell
Remaining Meeting Time: 02:38
Stop Share
File Edit Shell Debug Options Window Help
>>> print(str.strip( )) # it removes whitespaces
rajesh kumar
>>> print(str.upper())
RAJESH KUMAR
>>> print(str.lower())
rajesh kumar
>>> print(str.replace("H","J")
SyntaxError: invalid syntax
>>> print(str.replace("H","J"))
rajesh kumar
>>> print(str.replace("h","j"))
rajesj kumar
>>> print(str.replace("rajesh","raj"))
raj kumar
>>> print(str.split()) #string into sub string
['rajesh', 'kumar']
>>> a=" i am a good boy"
>>> x="why" in a
>>> print(x)
False
>>> x= "ood" in a
>>> print(x)
True
>>> x="ama" in a # check string can justify the stirng by sunsing in or not in
>>> print(x)
False
>>> x="ama" not in a
>>> print(x)
True
PAGE 38 OF 38
7:08 AM
10/1/2020



2-10-2020

Python tuple

Python tuple is used to store the sequence of immutable object's

A tuple can be created by using" ( )" and separated by ","

The tuple is only for reading (it can update or modify once the tuple is created)

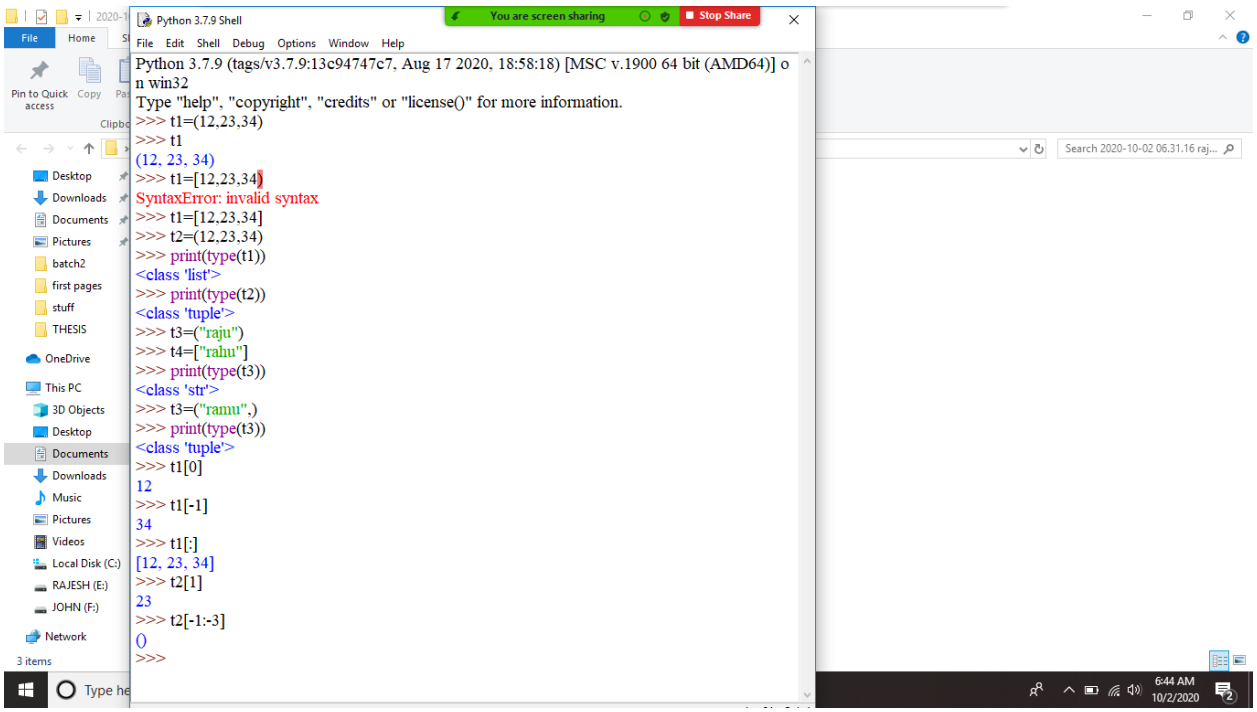
How to create tuple

T1=(101,"raju",22)

T2=("ramu","rani","vasu")

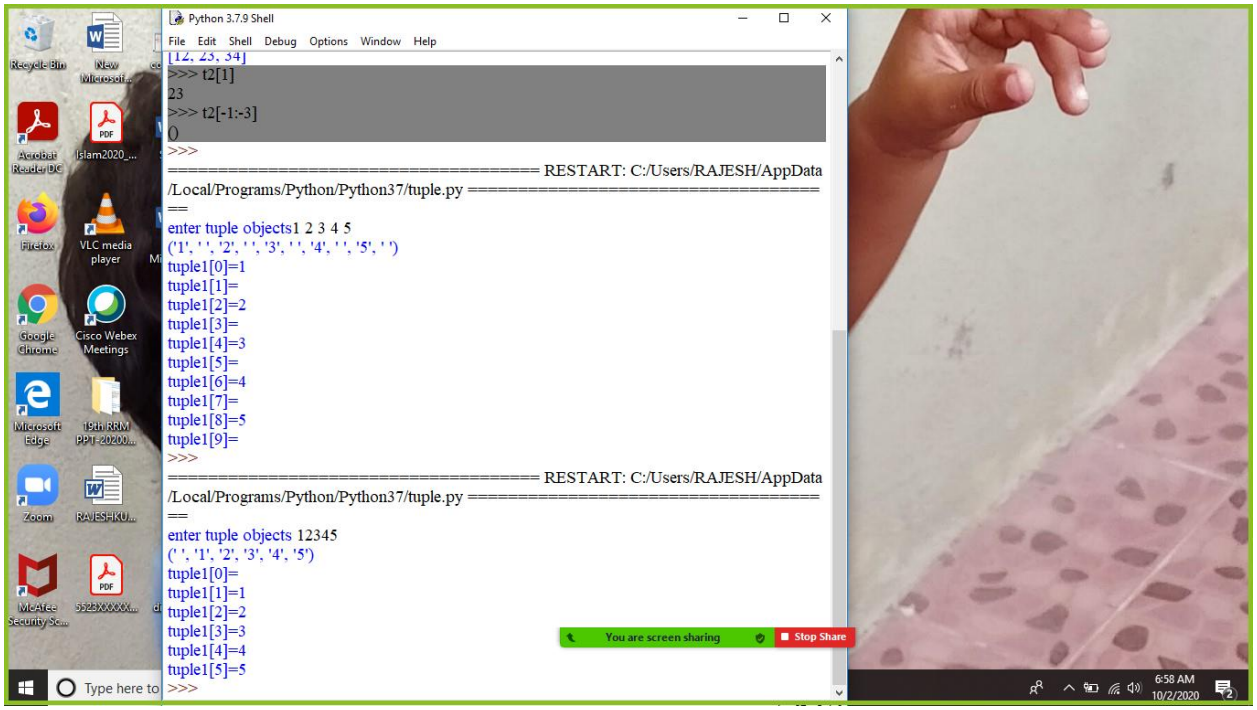
T3=10,20,"raju","rani"

T4=( ) #empty tuple



Program 17 : wpp and create empty tuple and insert the objects into empty tuple?

```
#empty tuple filling
tuple1=tuple(input("enter tuple objects::"))
print(tuple1)
count=0
for i in  tuple1:
    print ("tuple1[%d]=%s"%(count,i))
    count=count+1
```



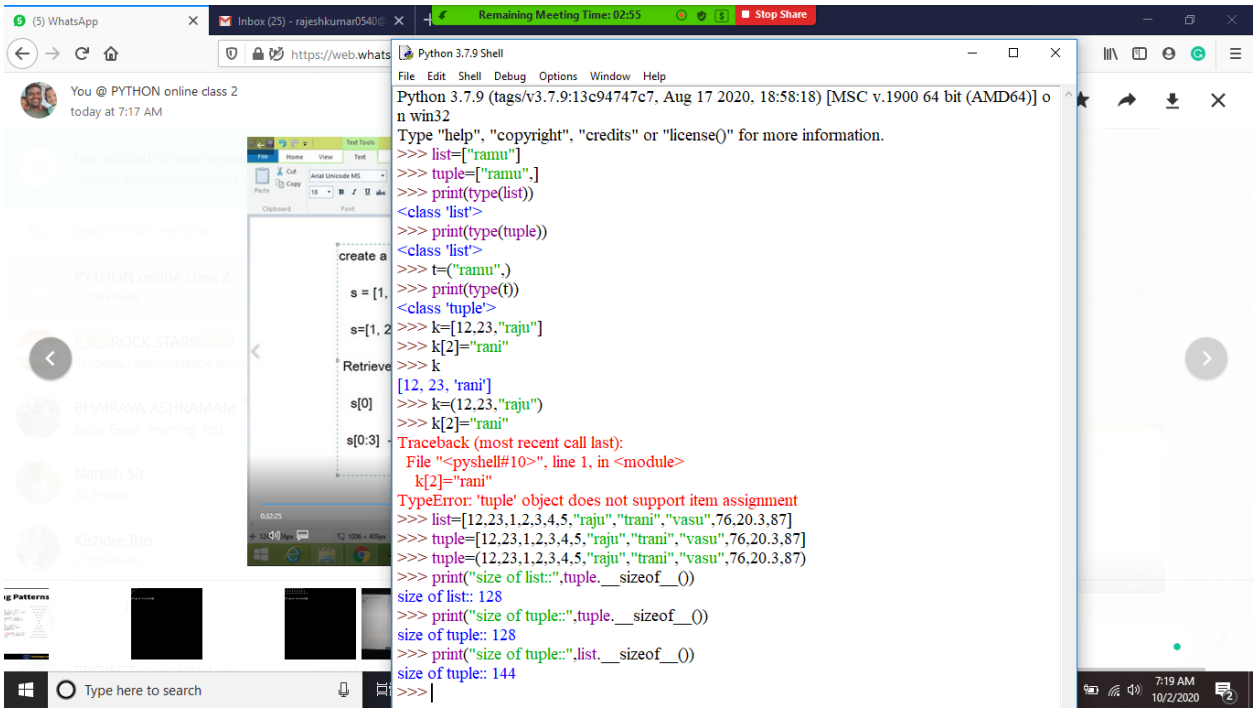
Difference List and tuple

List	Tuple
List is to store the elements in sequence	Tuple is to store the objects in sequence
List=[]	Tuple=()
All data types can be accessed	All data types can be accessed
list=["ramu"] print(type(list)) <class 'list'>	Tuple=("ramu",) Print(type(tuple)) <class 'tuple'>

Mutable :list are mutable
It can be modified
Read and write
k=[12,23,"raju"]
>>> k[2]="rani"
>>> k
	Immutable :Cant be modified
	Its read only
	k=(12,23,"raju")
	>>> k[2]="rani"
	Traceback (most recent call last):
	File "<pyshell#10>", line 1, in <module>
	k[2]="rani"
	TypeError: 'tuple' object does not support item assignment

Memory:list can store more memory
ist=[12,23,1,2,3,4,5,"raju","trani","vasu",76,20.3,87]
>>> >>>
tuple=(12,23,1,2,3,4,5,"raju","trani","vasu",76,20.3,87)
>>> print("size of list::",tuple.__sizeof__())
size of list:: 128
	It can store less memory
	tuple=[12,23,1,2,3,4,5,"raju","trani","vasu",76,20.3,87]
	
	>>> print("size of tuple::",list.__sizeof__())
	size of tuple:: 144

Function	



3-10-2020

Python Dictionary

- Is used to store the data in a **key-value** pair format
- Its deal with real life data arrangement
- It is mutable data structure

How to define dictionary

- ✓ Key must be a single element
- ✓ Value can be any type such as list,tuple or string
- ✓ It can be created by using multiple keypairs enclosed with curly brackets {} and each key is sepearated from its value bythe clolon(:)
- ✓ Dict() method to create the dictionary

Name =(name=”rajesh”,number=123,salary=20000.24)

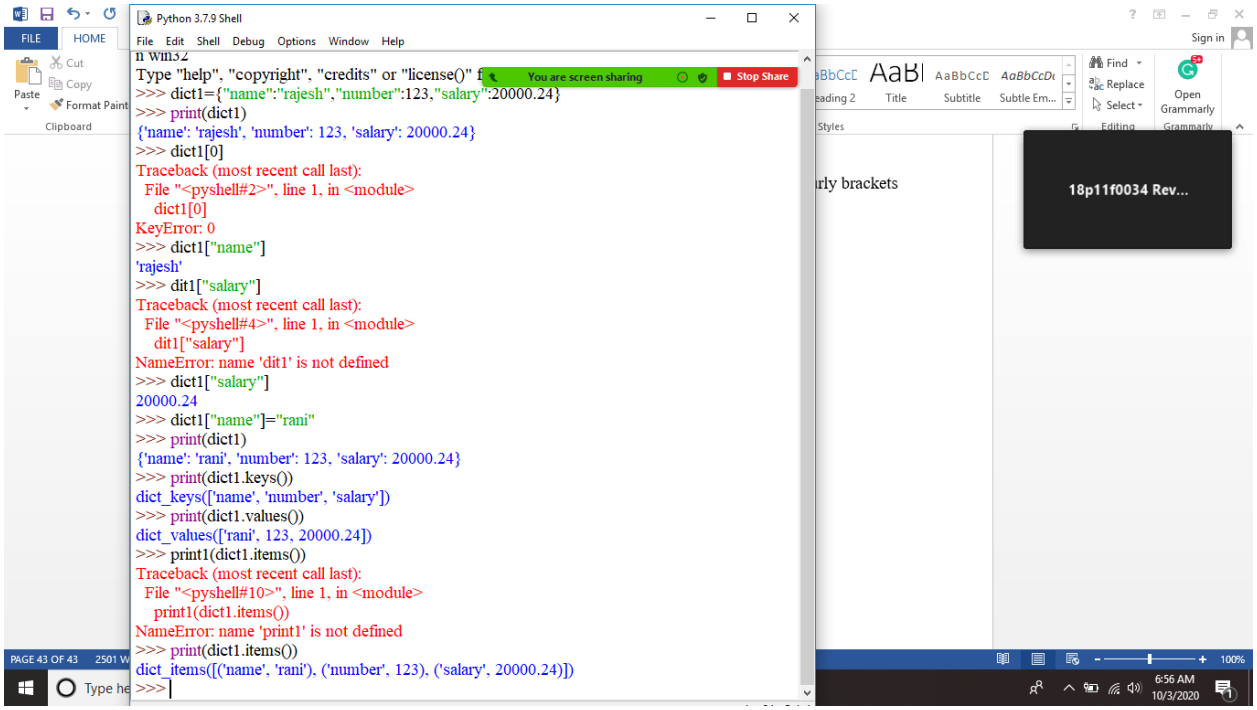
or

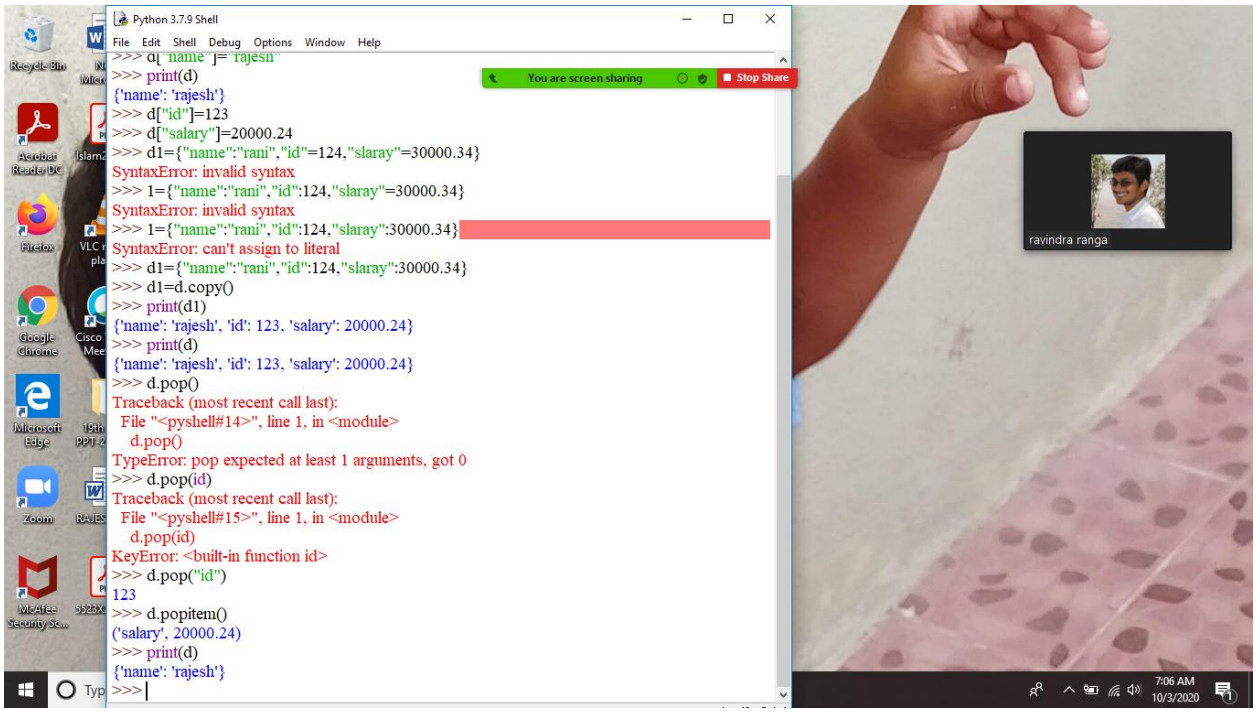
Name=dict(name=”rajesh”,number=123,salary=20000.24)

Or

Name={ “name”:”rajesh”,”number”:123,”salary”:20000.24}

But we cant sort because very value contains a key





dictionary	List
unordered	ordered
Accessing:keys	Index
Collection of key value pairs	Collection of elememnts
Preferred when you have unique keyvalue	Preferred for order data
No duplicates	Allow duplicates
Both are separated by ,	Both are separated by ,

Mutable:modifiable	Mutable:modifiable
It have very huge memory for key pairs	It have less memory compare to dictionary .

Case study:Store the indian cricketers data for 2020 and disply the stats of any player from the data by using sequence

101

Id name type numberofruns average ranking

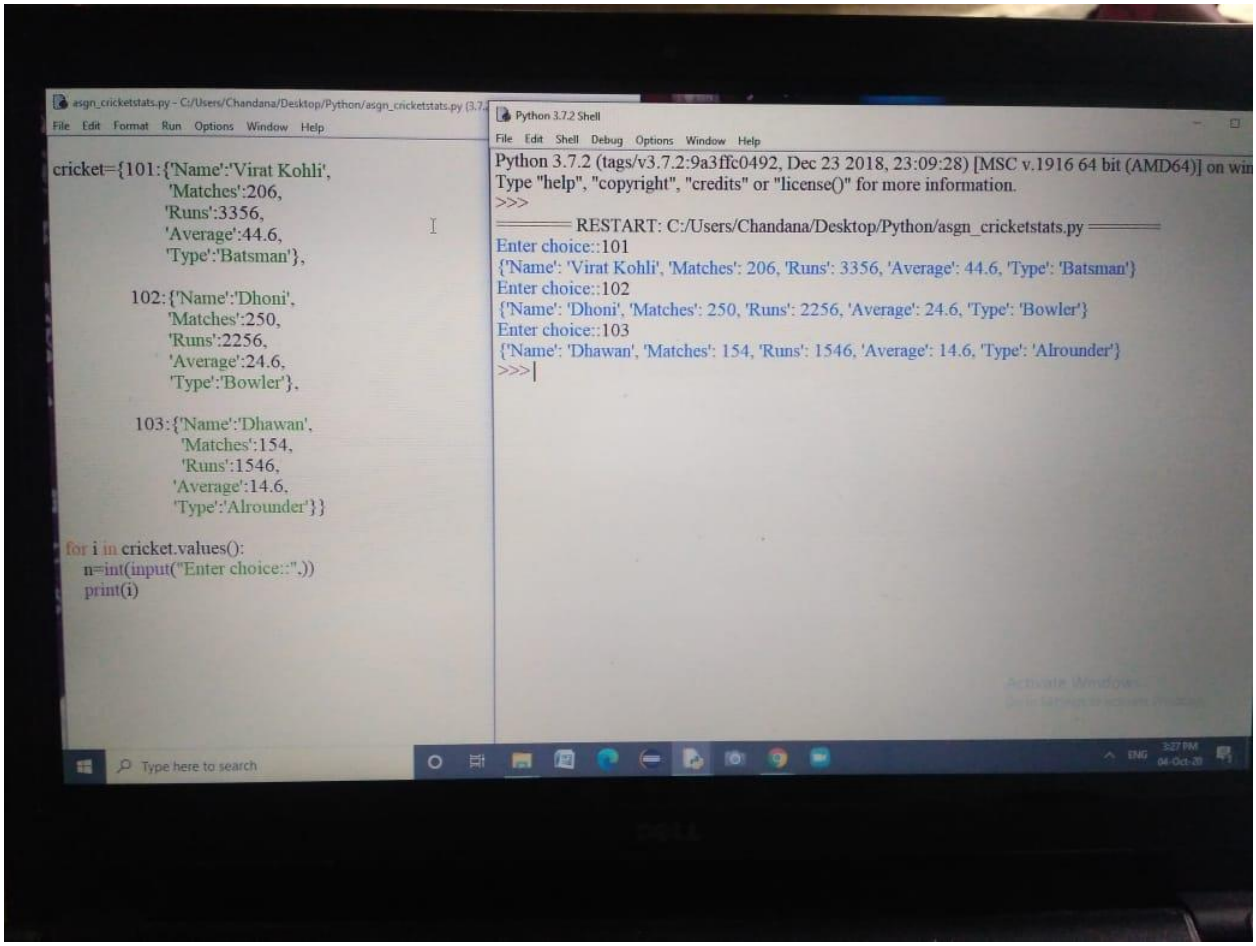
101 raj batsman 2000 30.45 1000

102

Id name type numberofruns average ranking

D_101

10



5-10-2020

Funcations

Funcations are the most important aspect of an applications

- ✓ A funcation can be defined as the organized block of "reusable" code which can be called whenever its required .

It helps to break the program in small sets

Synatax:

```
def my_name(parametrs):
    Funcation block
Return expression or funcation calling
```

- ➢ Def→define the funcation
- ➢ Parametrs or arguments can be used to define the variable refrence values .
- ➢ Function can be starts with ":"
- ➢ Funcation always return the statement or values

➢ User defined funcations are define by the user performance

➢ Built in funcation are pre-defined funcations

Advantage of Functions in Python

There are the following advantages of Python functions.

- Using functions, we can avoid rewriting the same logic/code again and again in a program.
- We can call Python functions multiple times in a program and anywhere in a program.
- We can track a large Python program easily when it is divided into multiple functions.
- Reusability is the main achievement of Python functions.
- However, Function calling is always overhead in a Python program.

a=20 b=10 print("the sum of",a+b) print("the division of",a/b)	def my_funcation(a,b): Print("the sum is",a+b) Print("the division",a/b) Print("the sub",b-a)

print("the mul of",a*b) a=200;b=100 print("the sum of",a+b) print("the division of",a/b) print("the mul of",a*b) a=3000;b=2000 print("the sum of",a+b) print("the division of",a/b) print("the mul of",a*b)	My_funcation(10,20) My_funcation(200,100) My_funcation(3000,1000)

Program 19: wpp to print hello for 5 times by using funcations

```
#hello for 5 times
def rajesh(name): # defining funcation
    print("hi how are you",name)  # statemnt or logic block


rajesh("kumar")       #calling funcation +return values
rajesh("kumar")
rajesh("kumar")
rajesh("kumar")
rajesh("kumar")
def new():
    print("i am fine",name)
new()
```

program 20: wpp to design birthday wishes to rani by using funaction within funcation

```
def main_new():
    print("happy birthday")
def name_change(name):
    main_new()
    main_new()
    print("happy birthday to youuuuu",name)

name_change("rani")
```

6-10-2020

program 21:wpp to find sum of two numbers bt using return key word

```
def sum()
    a=10
    b=20
    c=a+b
    return c
    print(sum()) # hear without return c it will show “none “
```

Arguments in function

- ✓ The arguments are types of information which can be passed into the function.
- ✓ The arguments are specified in the parentheses.
- ✓ We can pass any number of arguments, but they must be separate them with a comma.

Consider the following example, which contains a function that accepts a string as the argument.

Types of argumnets

1.required argumnets def fun(name)……fun(“rjaesh”)

2.key word argumnets (**)

3.default argumnets (age=22)or in java its called static

4.variablr –length argumnrts (*)

program 22: wpp to find emp details and salary by using funcations

emp details and salary

```
def emp_details(eno,ename,eadd,ephone):

    print(eno,ename,eadd,ephone)

def emp_salary(basic,ta,da):

    totalsalary=basic+ta+da

    print(totalsalary)

emp_details(121,"rajesh","tpt",9999)

emp_salary(1000,150.23,200.34)
```

program 23:wpp to check the argument is connecting to the return values

```
#argument return values

def func(name):

    msg="hi how are you::"+name #string declaration with argument

    return msg

name=input("enter the name::")

print(func(name))

# how to call string

#strings also returns
```

Output ;

```
enter the name::rani

hi how are you::rani
```

Program 24: wpp to check the default arguments

```
#defalut argumnets

def per_details(name,age=22):

    print(name,age)

    print("my name is",name,"age is ",age)

per_details("rajesh")
```

def fun(*name)→

for the large projects we may use number of arguments to be passed in advance

unkown arguments we can pass

its always , separated values in the funcation calling

for loop

program 25:wpp to check the unkown arguments and read the values

```
#unkown arguments

def rajesh(*names):

    print("type of arguments passed",type(names))

    print("argunehts are::")

    for i in names:

        print(i)

rajesh("rani","raju","vasu","devi","kani")
```

output:

```
type of arguments passed <class 'tuple'>

2421046457832

argunehts are::

rani

raju

vasu

devi

kani

123
```

Program 26: wpp to check the key word argumnents (**name)

- ** to allow user to display argumnets in random order
- The name of he argument treated as key word and matched
- If the same match is found ,the values of the argument are copied in the funcation definition

```
#keyword argumnets

def keyword(name,msg,name1):

    print("hi how are you",name,"and",msg,"and", name1)

keyword(name="ramu",msg="where are you",name1="rani")
```

Lambda funcation

It is anonymous funcation means →name less funcation

No need to define def :keyword rather we are using lambda keyword

It can accept any number of arguments ,but they can return only on evalue in the form expressions or sequence Syntax:

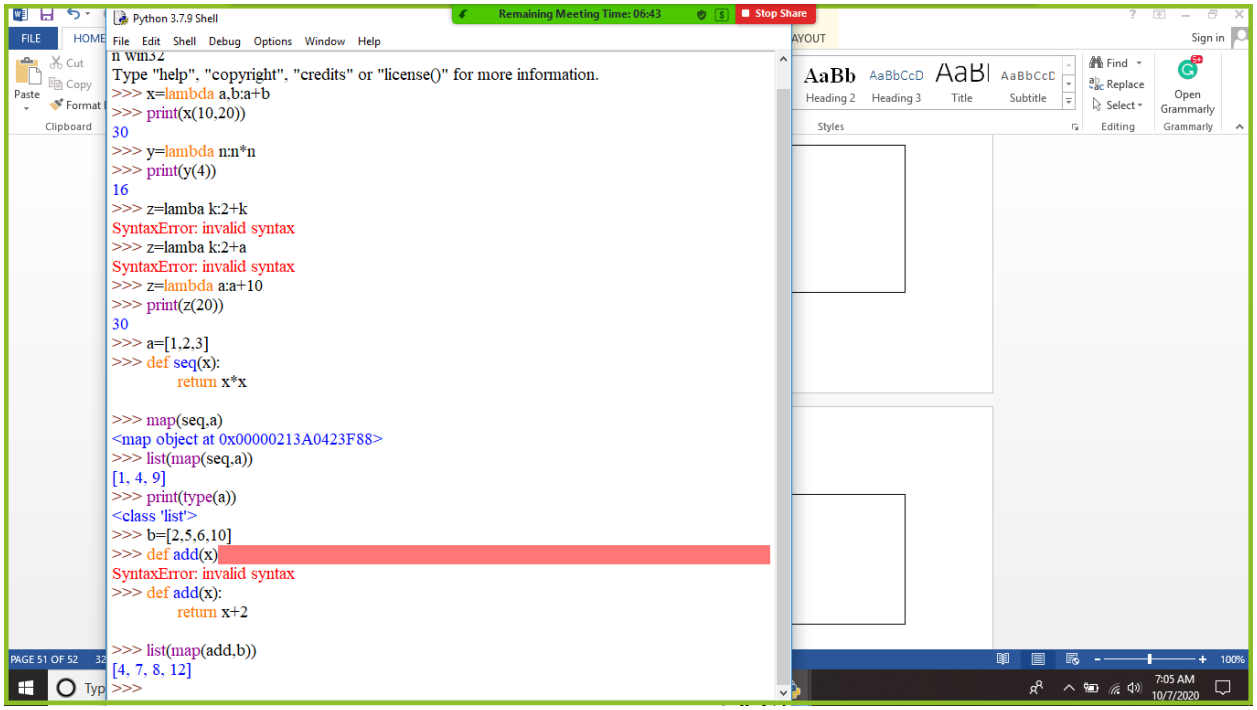
Lambda <input_values>:expressions

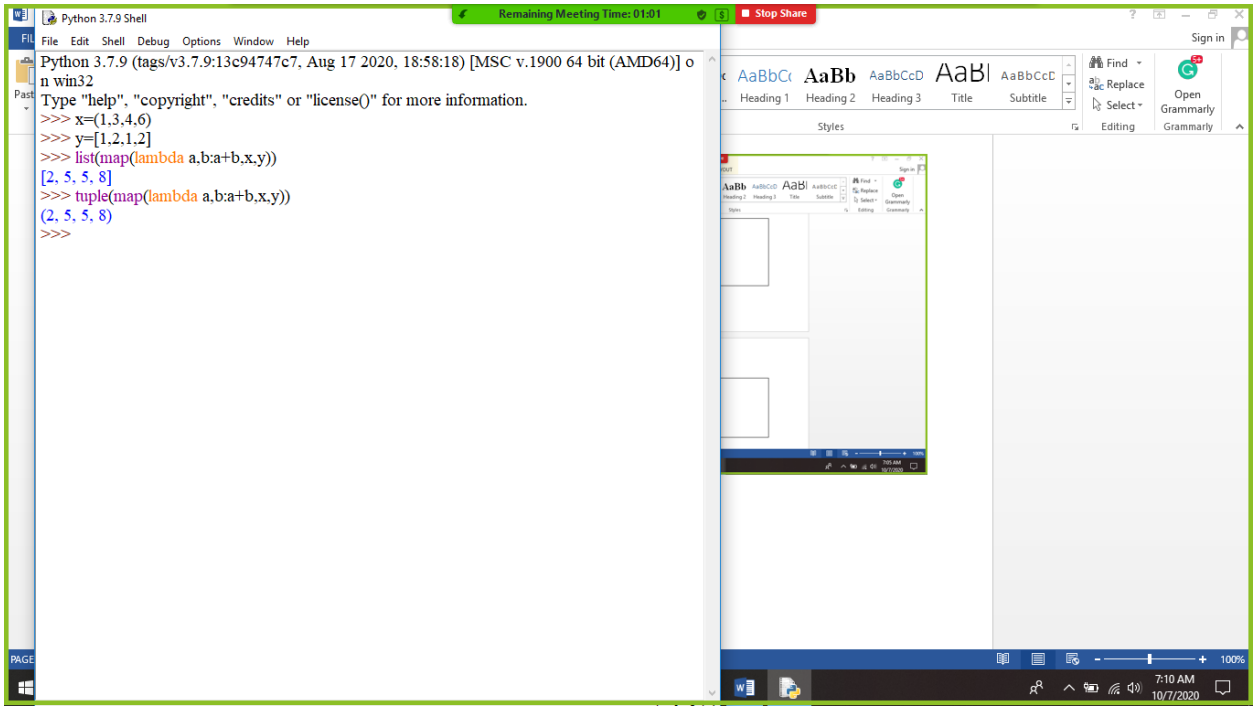
Funcation	Lambda
Example :1Def add(a,b) Print(a+b) Retutn c Add(20,10) Example 2:Def mul(a,b) Print(a*b) Return c Print(10,20)	Example 1:X=Lambda a,b:a+b Print(x) Print(x(10,20)) Example 2:x=lambda a,b:a*b Print(x(10.20))

Map() funcation in python accepts a funcation and a list

It gives a new list which contains all modified items return by the funcation

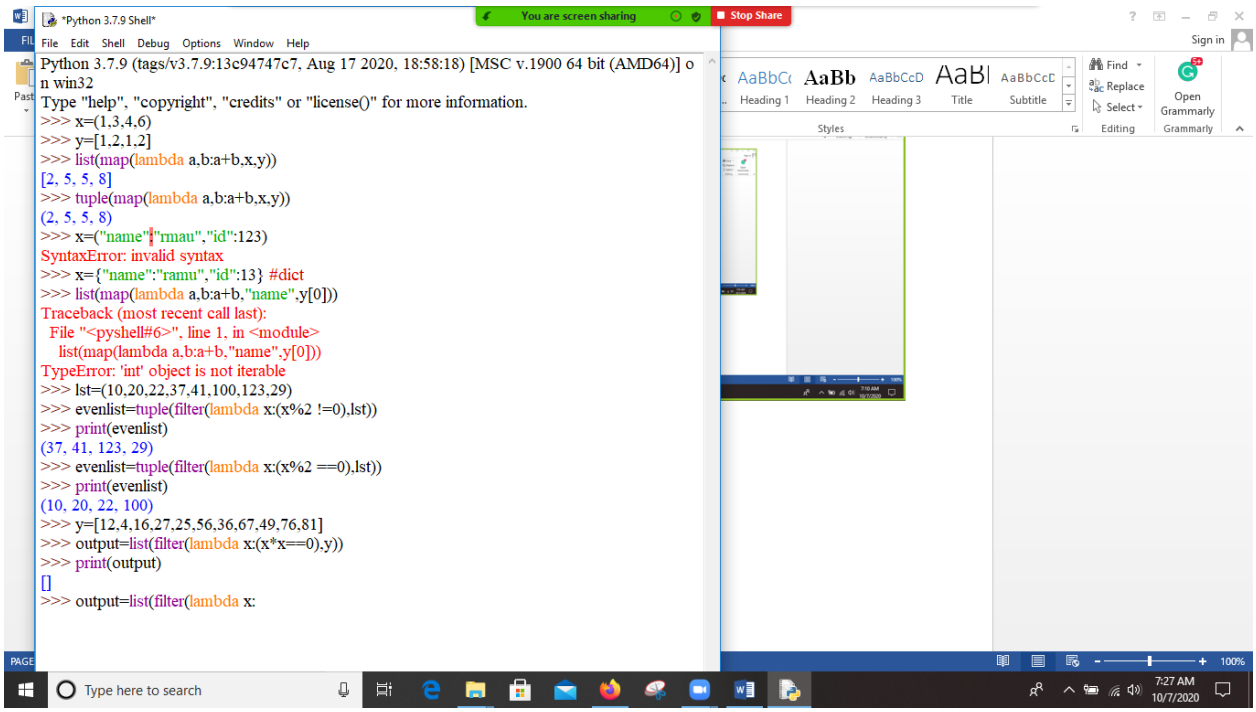
The size of the list or tuple can be equal while calling the map()





Filter()

- It accept a funcation and a list as an arguments .
- It provides an filter out all elemets of the sequence
- It returns new sequence



8-10-2020

Files in python:

File can read/write data to the text/csv/excel/.c/.java/.py

Files can read large number of data

How :

- ✓ Open a file
- ✓ Read or write a file ..>performing operations
- ✓ Close a file

Access modes :

r→it opens the file to read only

w→ it opens the file to write only

rb→read only in the binary format

wb→write only in the binary format

r+→read and write

w++→read and write

how to opea a file

- ✓ python provides a open() funcation that accept two arguments
- ✓ 1.file name 2.access mode in which the file is accessed
- ✓ The funcation can return the out put

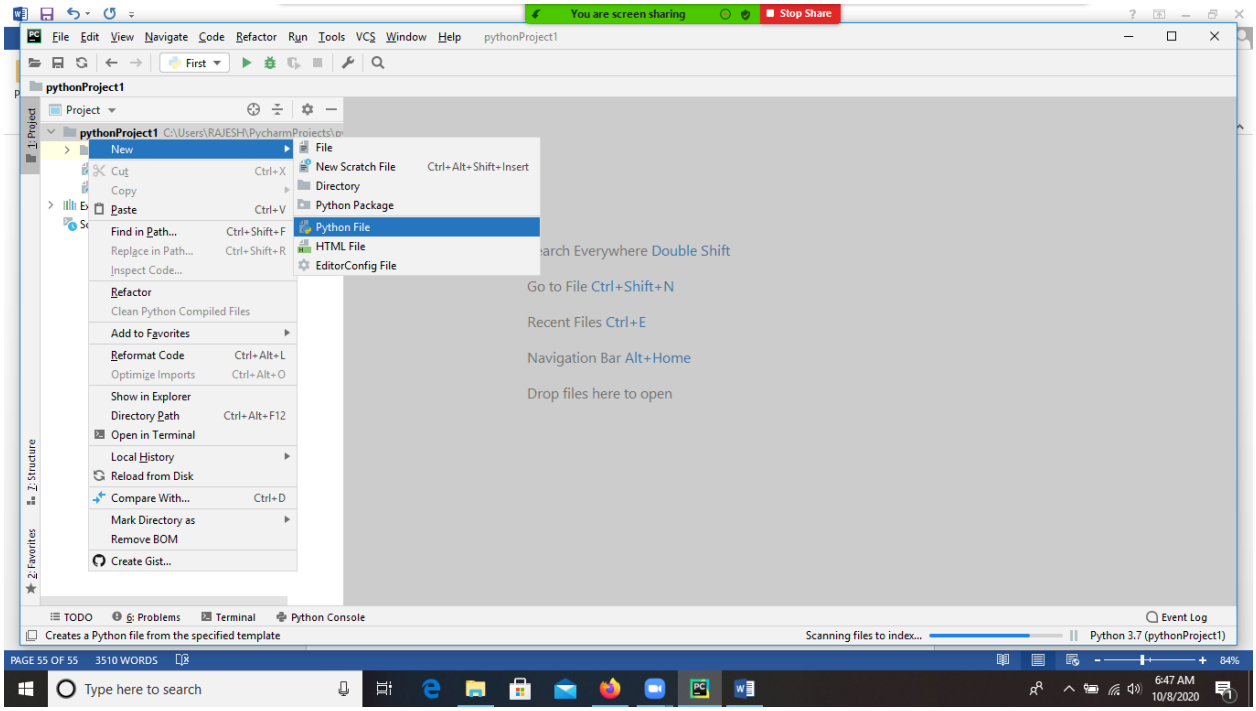
Syntax:

file_object=open(<file_name_path>,<access_mode>,<buffering>)

steps to open a file :

1.create .txt file in the specified directory

2.open payCham



3. we set a name for file as openfile.py

4. rajesh=open('raju.txt','r')

5.print(rajesh)

6.rajesh.close()

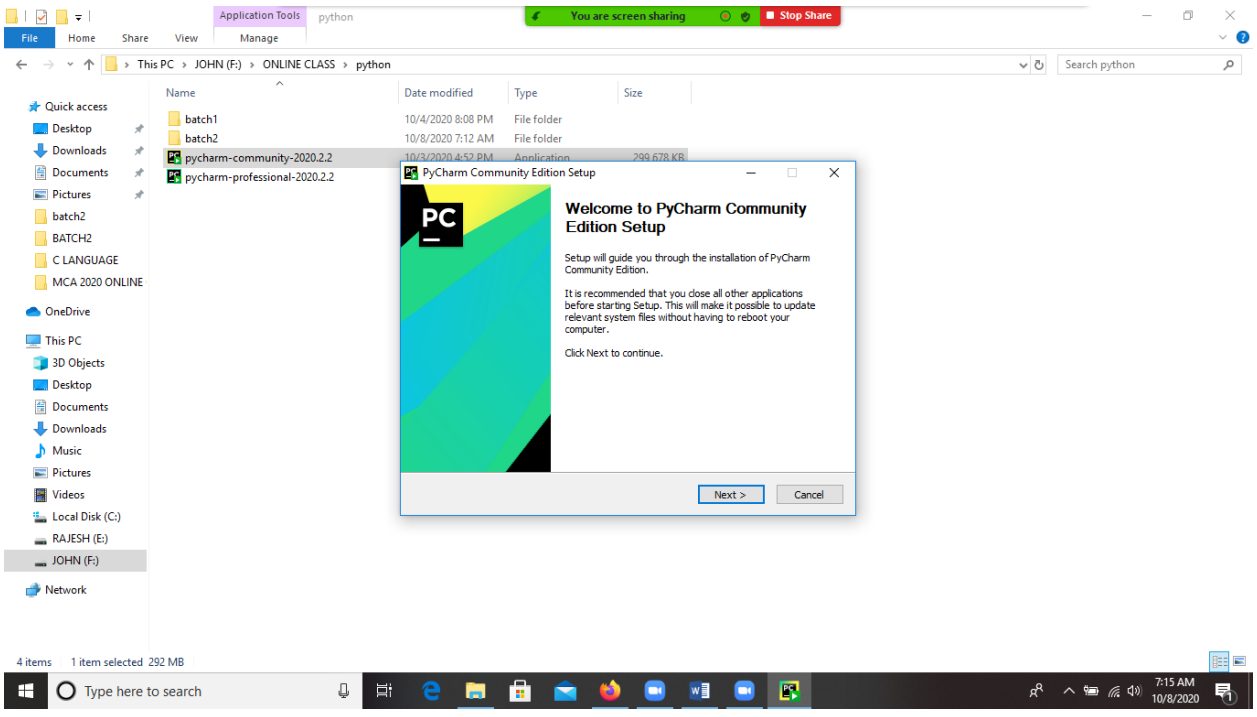
Output :

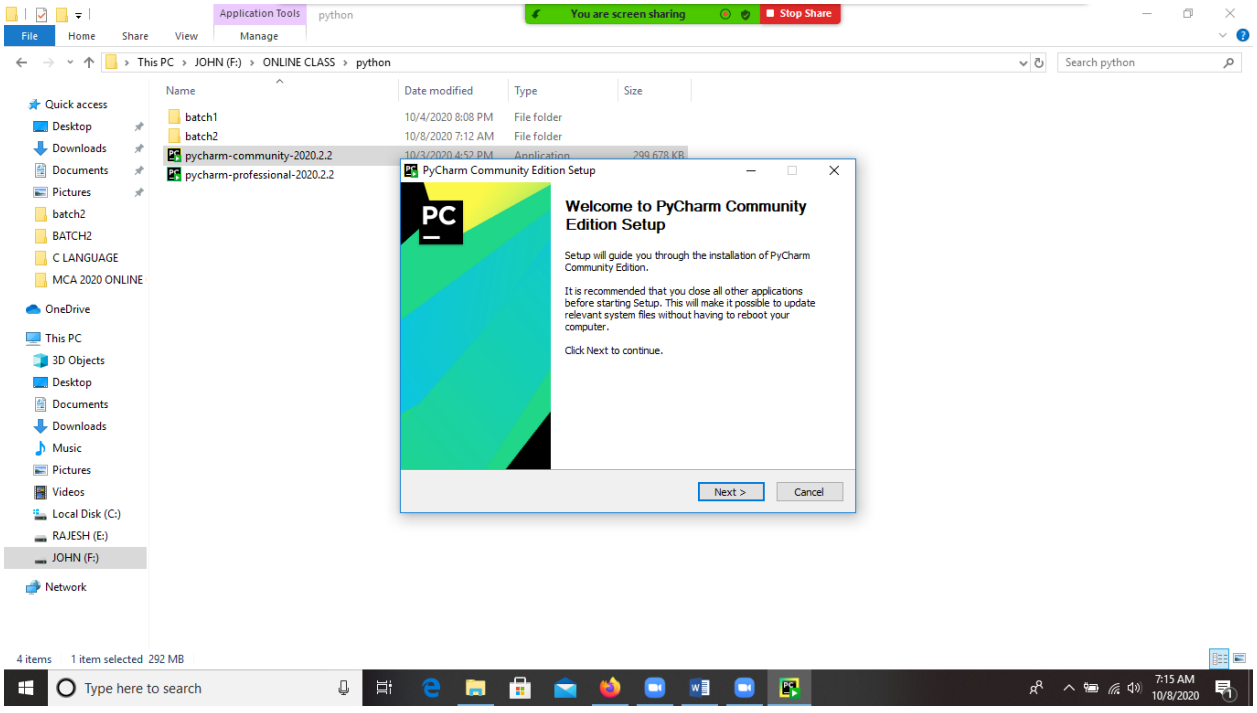
Program 29:wpp to read a file from path
#open file rajesh=open('raju.txt','r') data=rajesh.read() print(data) rajesh.close() *# read()-->gettting data from file without modifications*
Output: hi how are you and how old are you

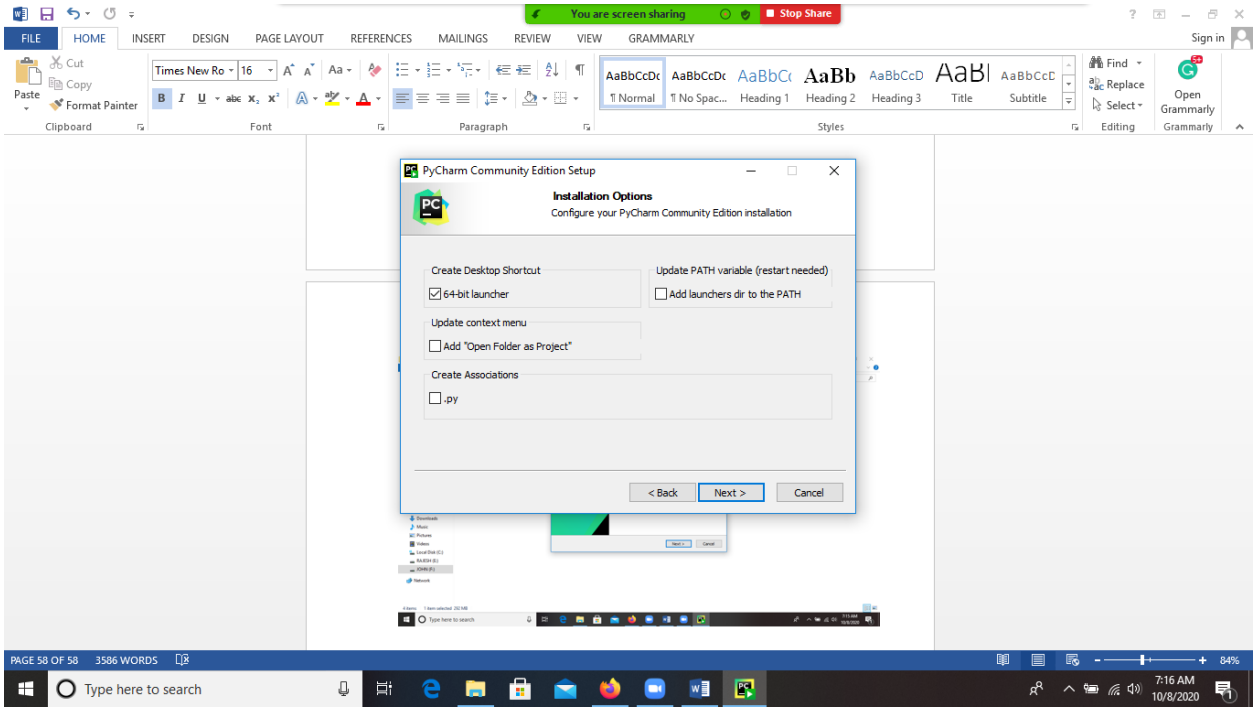
Program 30:wpp to read file from other directory
#open file rajesh=open('F:/ONLINE CLASS/python/batch2/dept.txt','r') data=rajesh.read() print(data) rajesh.close()
Output: DEPTNO,DNAME,LOC 10 ,ACCOUNTING,NEW YORK 20,RESEARCH,DALLAS 30,SALES,CHICAGO 40,ramu,india

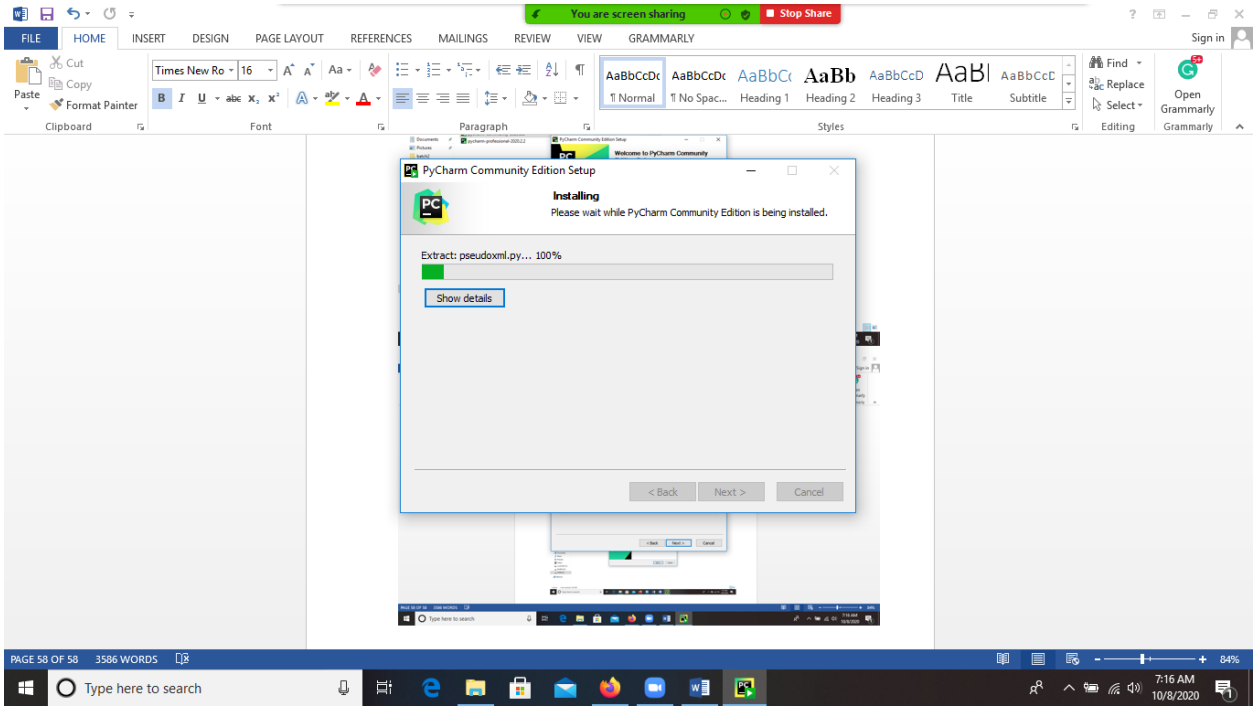
Paycharm download and installation :

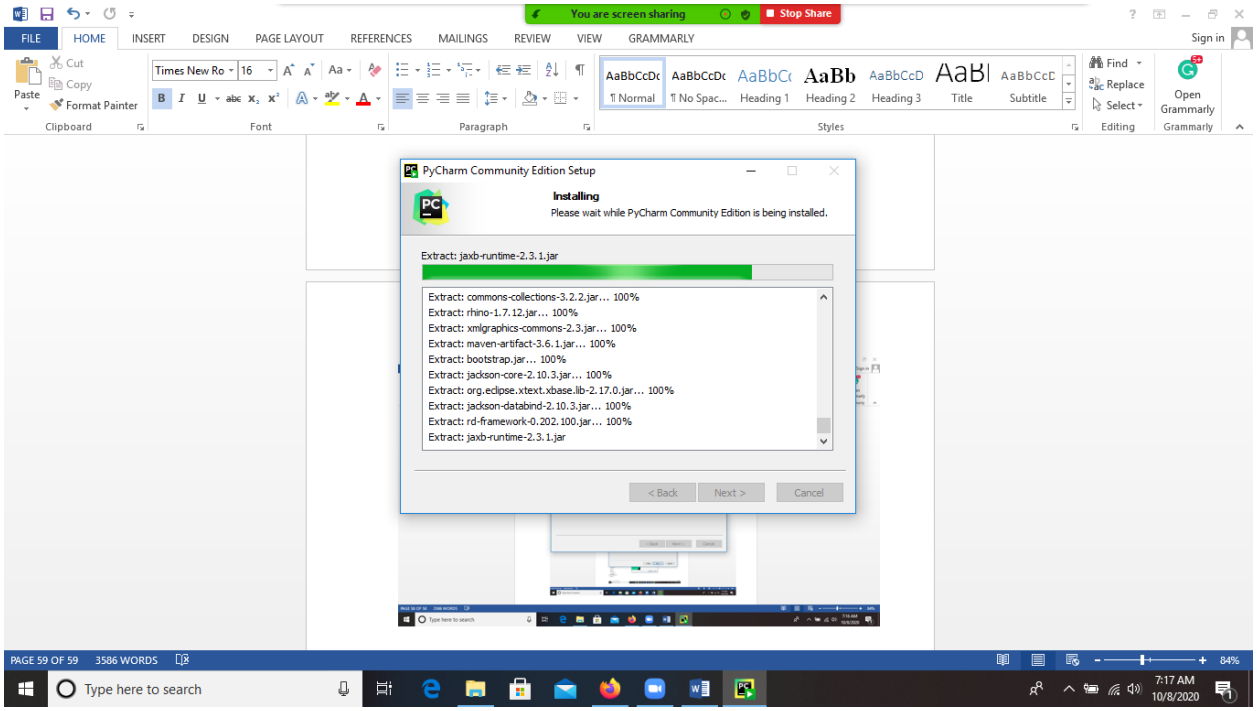
https://www.jetbrains.com/pycharm/download/#section=windows

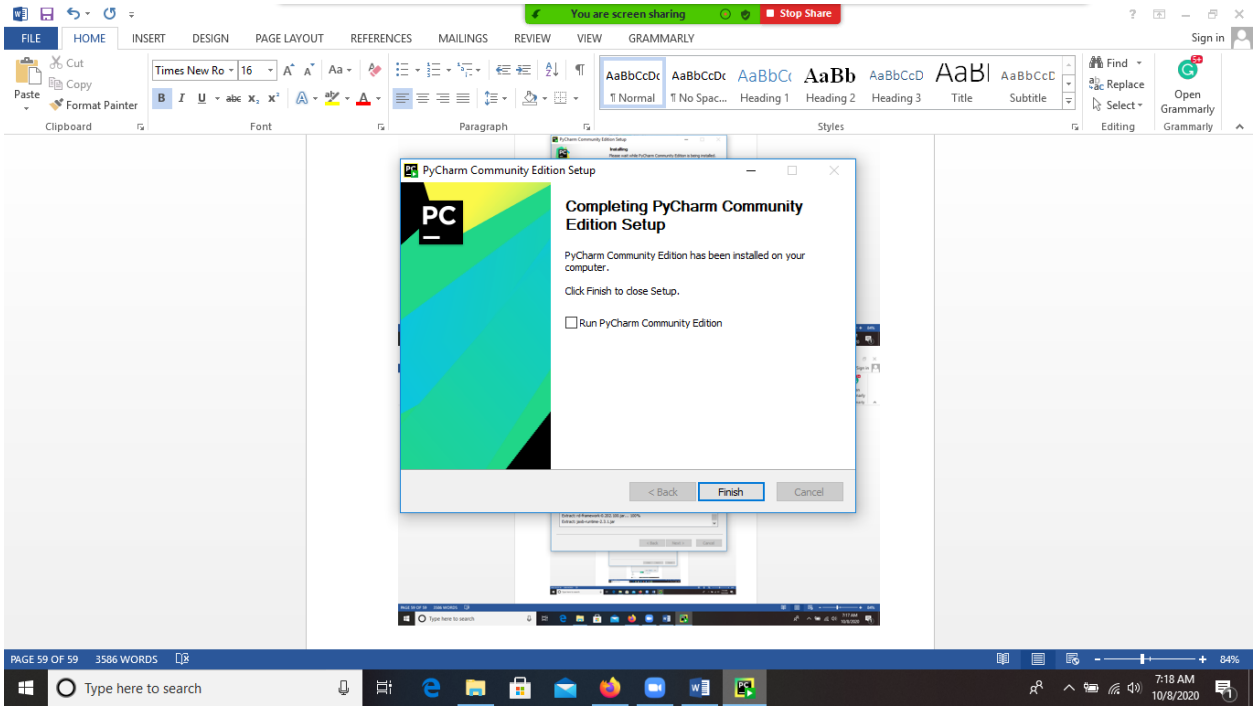



You are screen sharing
Stop Share
File Home Share View Application Tools Manage python
This PC › JOHN (F:) › ONLINE CLASS › python
Search python
Quick access
Desktop
Downloads
Documents
Pictures
batch2
BATCH2
C LANGUAGE
MCA 2020 ONLINE
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
RAJESH (E:)
JOHN (F:)
Network
Name
Date modified
Type
Size
batch1
10/4/2020 8:08 PM
File folder
batch2
10/8/2020 7:12 AM
File folder
pycharm-community-2020.2.2
pycharm-professional-2020.2.2
PyCharm Community Edition Setup
Welcome to PyCharm Community Edition Setup
Setup will guide you through the installation of PyCharm Community Edition.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel
4 items
1 item selected 292 MB
Type here to search
7:15 AM
10/8/2020

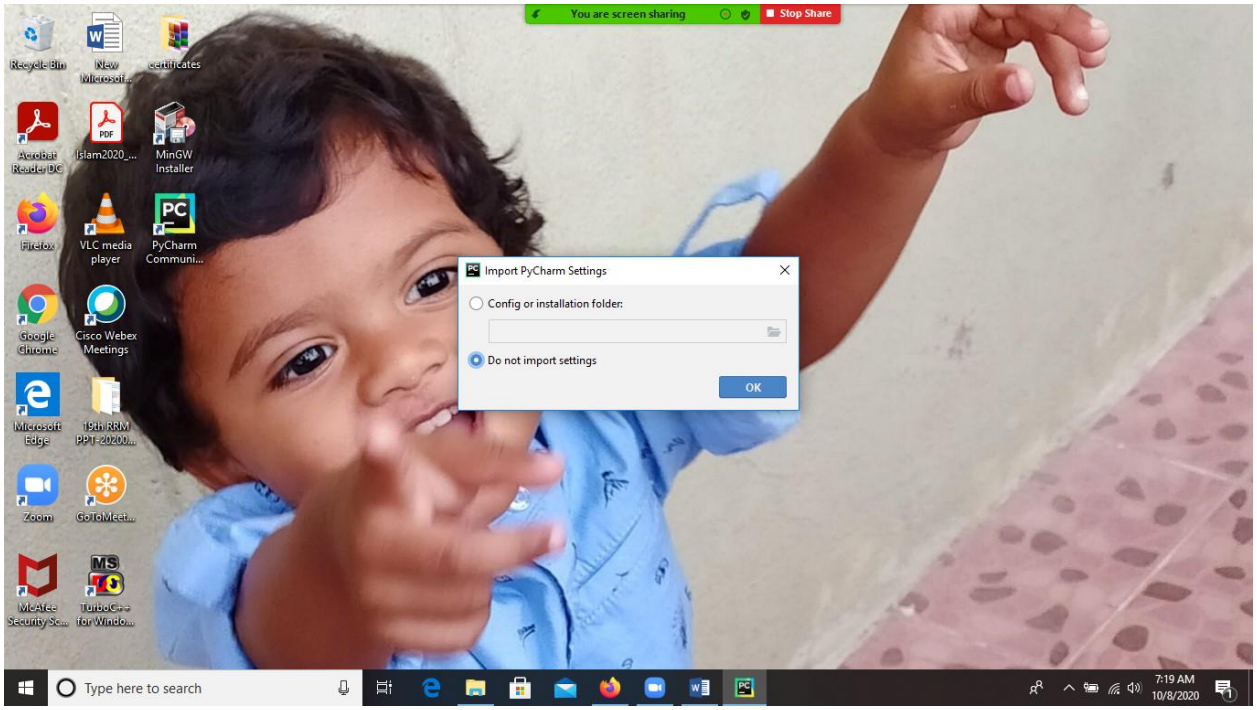

You are screen sharing
Stop Share
FILE HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW GRAMMARLY
Sign in
Paste
Cut
Copy
Format Painter
Clipboard
Times New Ro
16
Font
Paragraph
Normal
No Spac...
Heading 1
Heading 2
Heading 3
Title
Subtitle
Styles
Find
Replace
Select
Editing
Open Grammarly
Grammarly
PyCharm Community Edition Setup
Installation Options
Configure your PyCharm Community Edition installation
Create Desktop Shortcut
64-bit launcher
Update PATH variable (restart needed)
Add launchers dir to the PATH
Update context menu
Add "Open Folder as Project"
Create Associations
.py
< Back
Next >
Cancel
PAGE 58 OF 58
3586 WORDS
84%
Type here to search
7:16 AM
10/8/2020

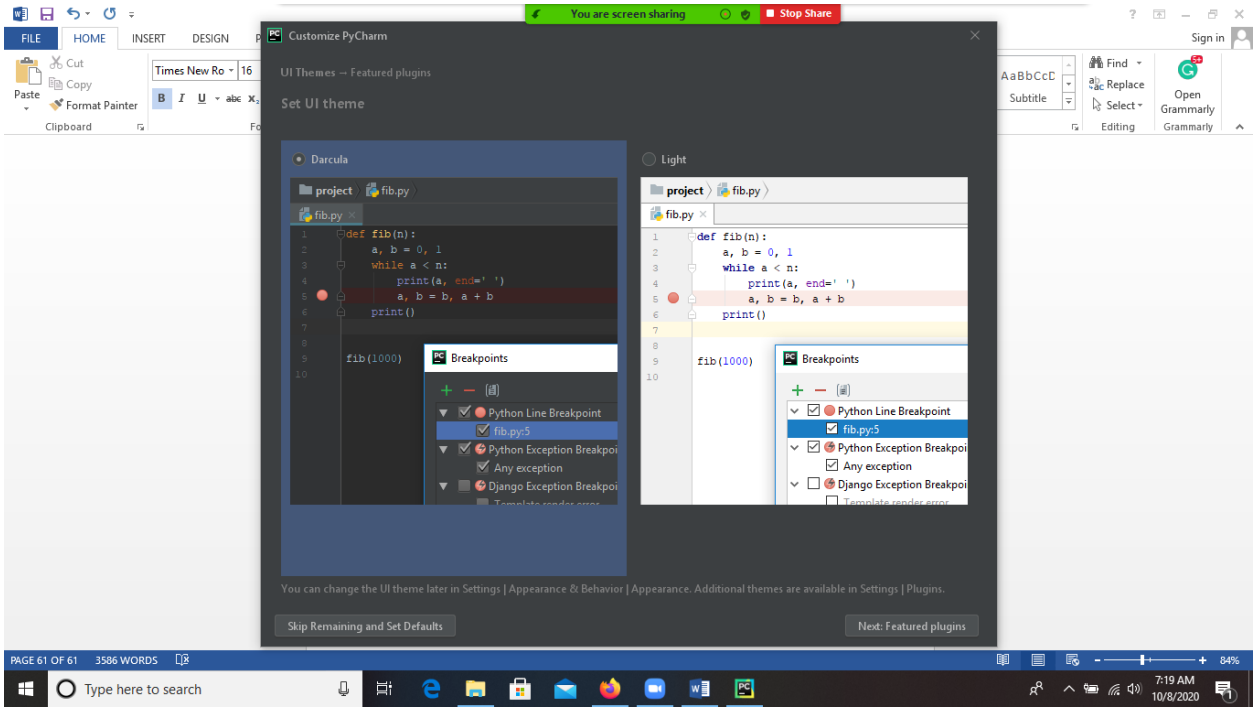

You are screen sharing
Stop Share
PyCharm Community Edition Setup
Installing
Please wait while PyCharm Community Edition is being installed.
Extract: pseudoxml.py… 100%
Show details
< Back
Next >
Cancel
PAGE 58 OF 58
3586 WORDS
84%
Type here to search
7:16 AM
10/8/2020

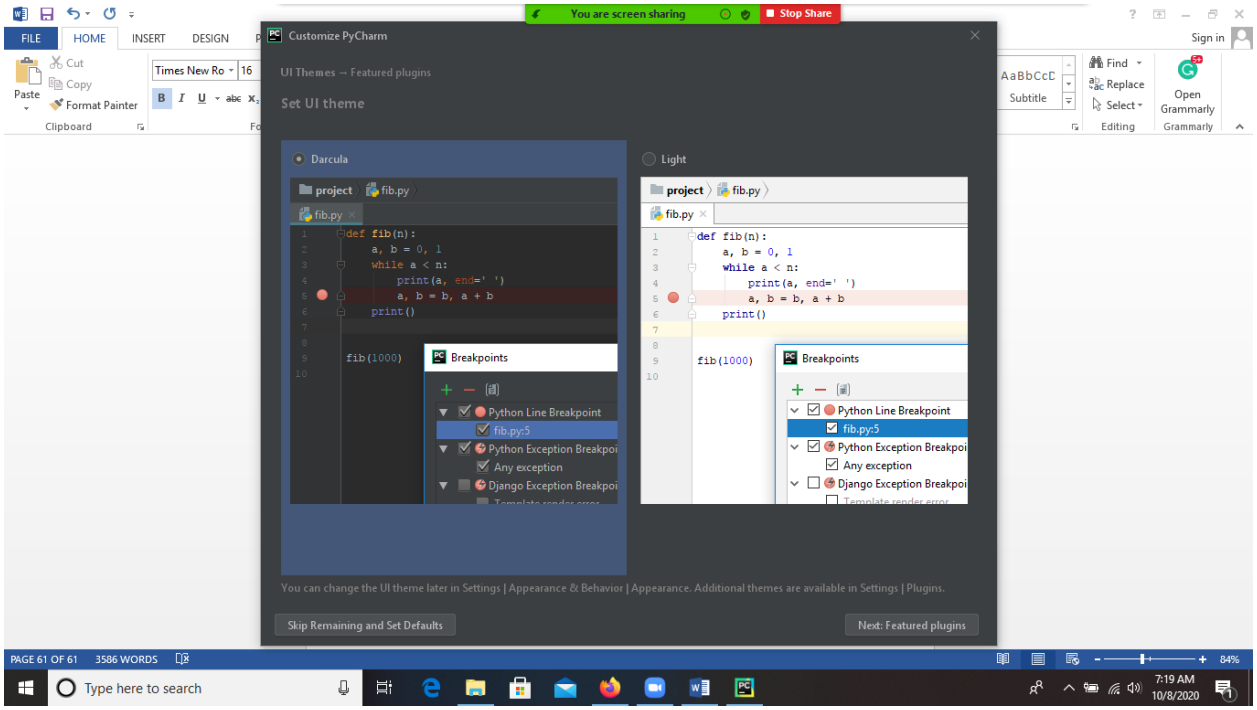

You are screen sharing
Stop Share
PyCharm Community Edition Setup
Installing
Please wait while PyCharm Community Edition is being installed.
Extract: jaxb-runtime-2.3.1.jar
Extract: commons-collections-3.2.2.jar… 100%
Extract: rhino-1.7.12.jar… 100%
Extract: xmlgraphics-commons-2.3.jar… 100%
Extract: maven-artifact-3.6.1.jar… 100%
Extract: bootstrap.jar… 100%
Extract: jackson-core-2.10.3.jar… 100%
Extract: org.eclipse.xtext.xbase.lib-2.17.0.jar… 100%
Extract: jackson-databind-2.10.3.jar… 100%
Extract: rd-framework-0.202.100.jar… 100%
Extract: jaxb-runtime-2.3.1.jar
< Back
Next >
Cancel
PAGE 59 OF 59
3586 WORDS
84%
Type here to search
7:17 AM
10/8/2020

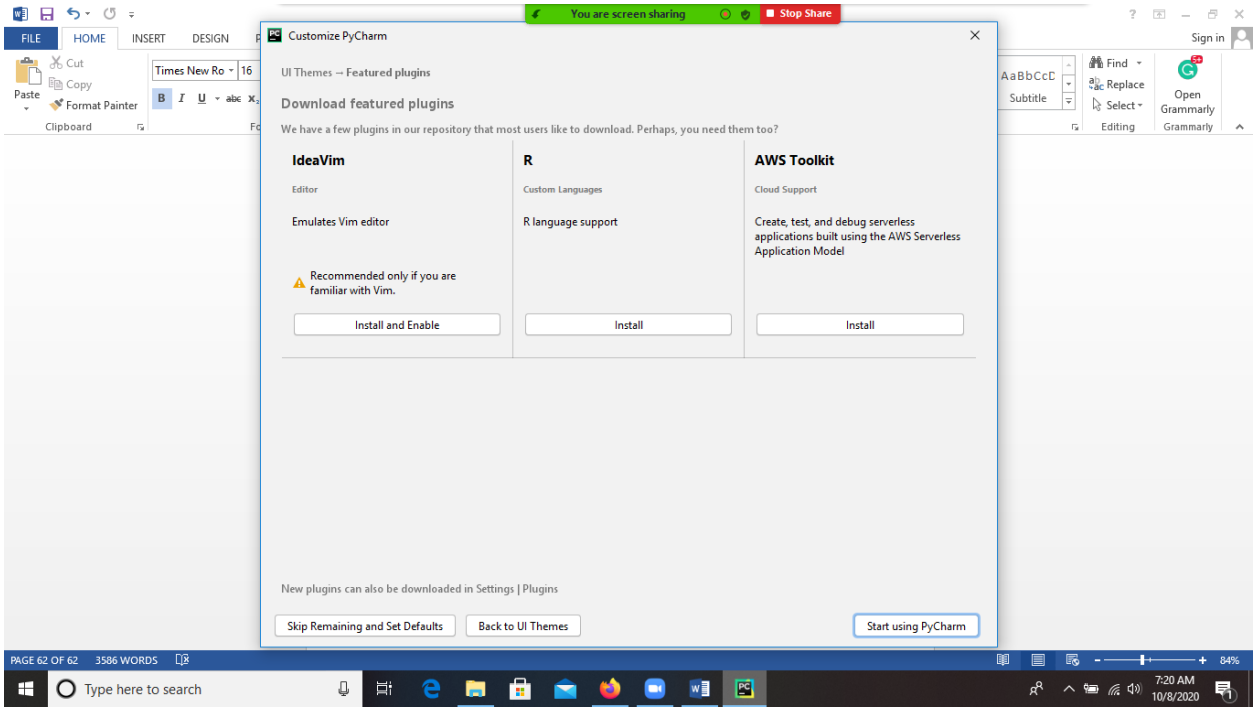

You are screen sharing
Stop Share
PyCharm Community Edition Setup
Completing PyCharm Community Edition Setup
PyCharm Community Edition has been installed on your computer.
Click Finish to close Setup.
Run PyCharm Community Edition
< Back
Finish
Cancel
PAGE 59 OF 59
3586 WORDS
Type here to search
7:18 AM
10/8/2020

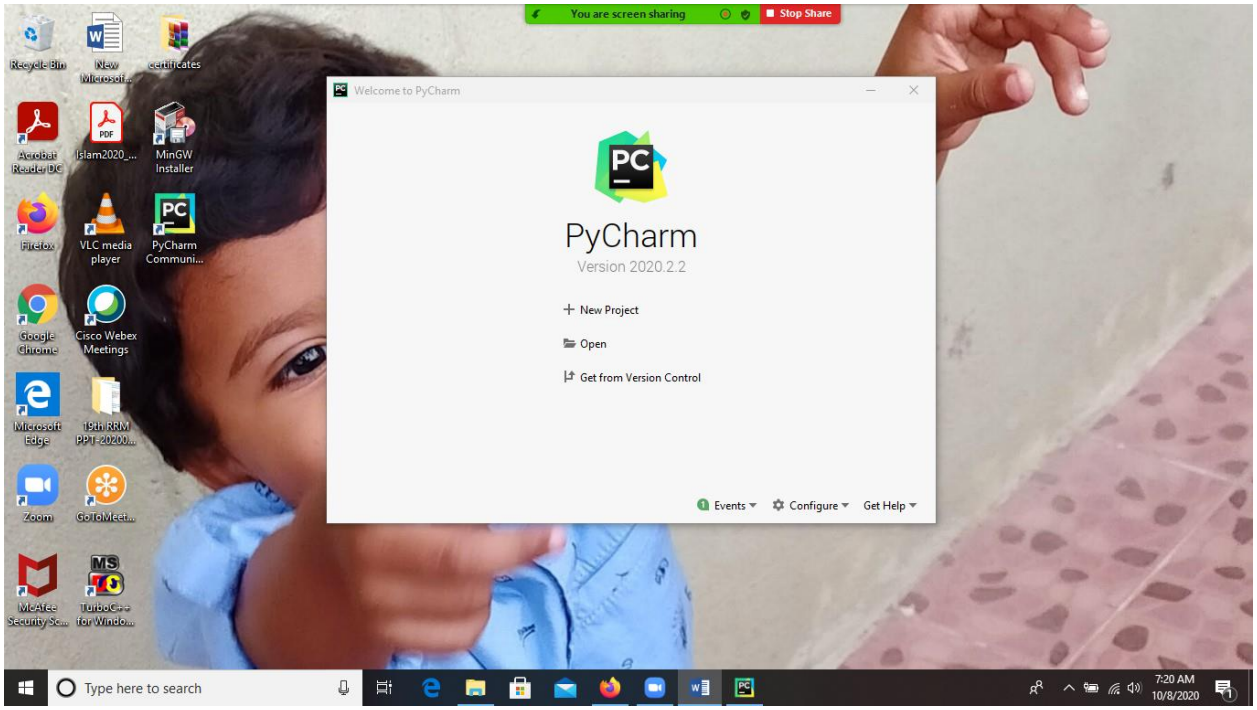

You are screen sharing
Stop Share
PyCharm
Community 2020.2
Type here to search
7:18 AM
10/8/2020

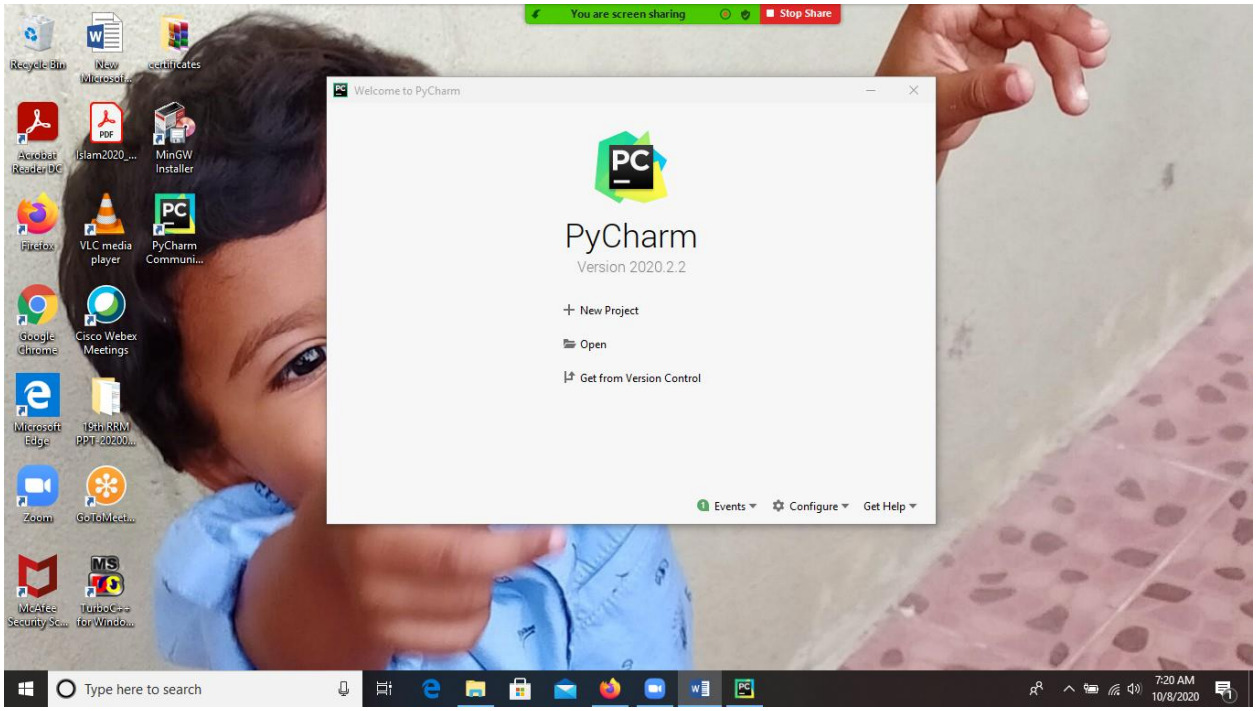

You are screen sharing
Stop Share
Import PyCharm Settings
Config or installation folder:
Do not import settings
OK
Type here to search
7:19 AM
10/8/2020

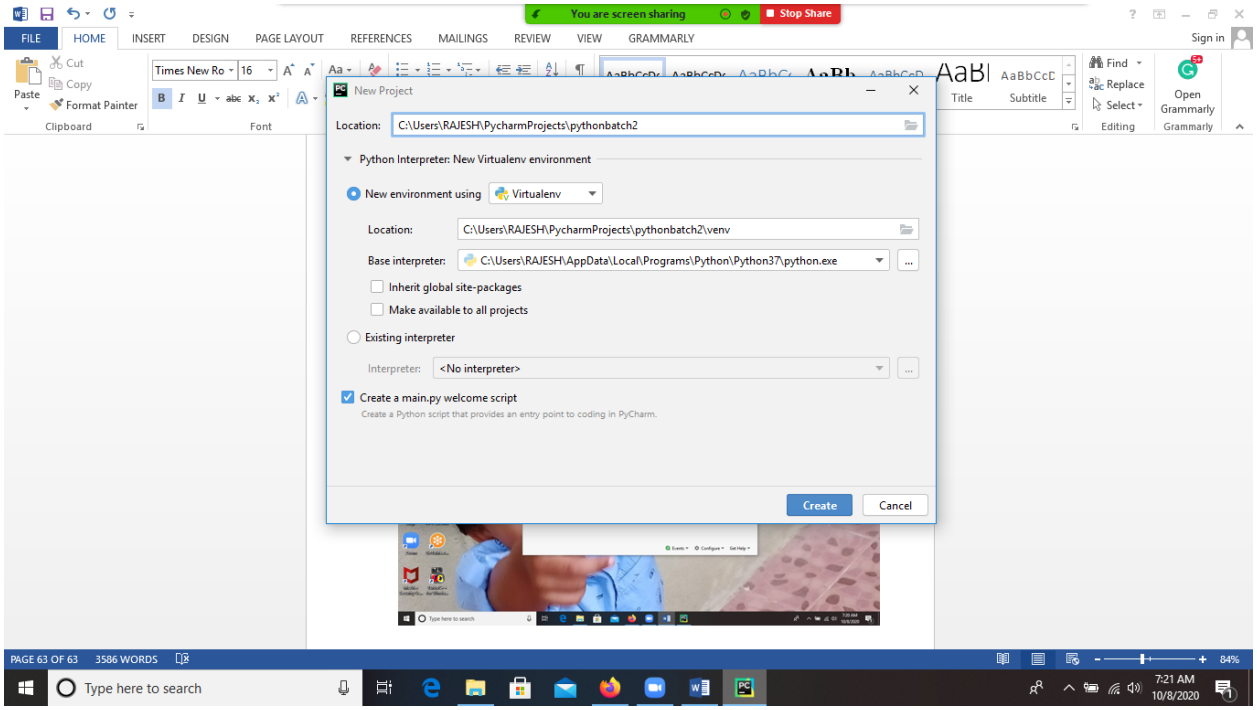

You are screen sharing
Stop Share
Customize PyCharm
UI Themes → Featured plugins
Set UI theme
Darcula
Light
Breakpoints
Python Line Breakpoint
fib.py:5
Python Exception Breakpoi
Any exception
Django Exception Breakpoi
You can change the UI theme later in Settings | Appearance & Behavior | Appearance. Additional themes are available in Settings | Plugins.
Skip Remaining and Set Defaults
Next: Featured plugins
PAGE 61 OF 61
3586 WORDS
Type here to search
7:19 AM
10/8/2020

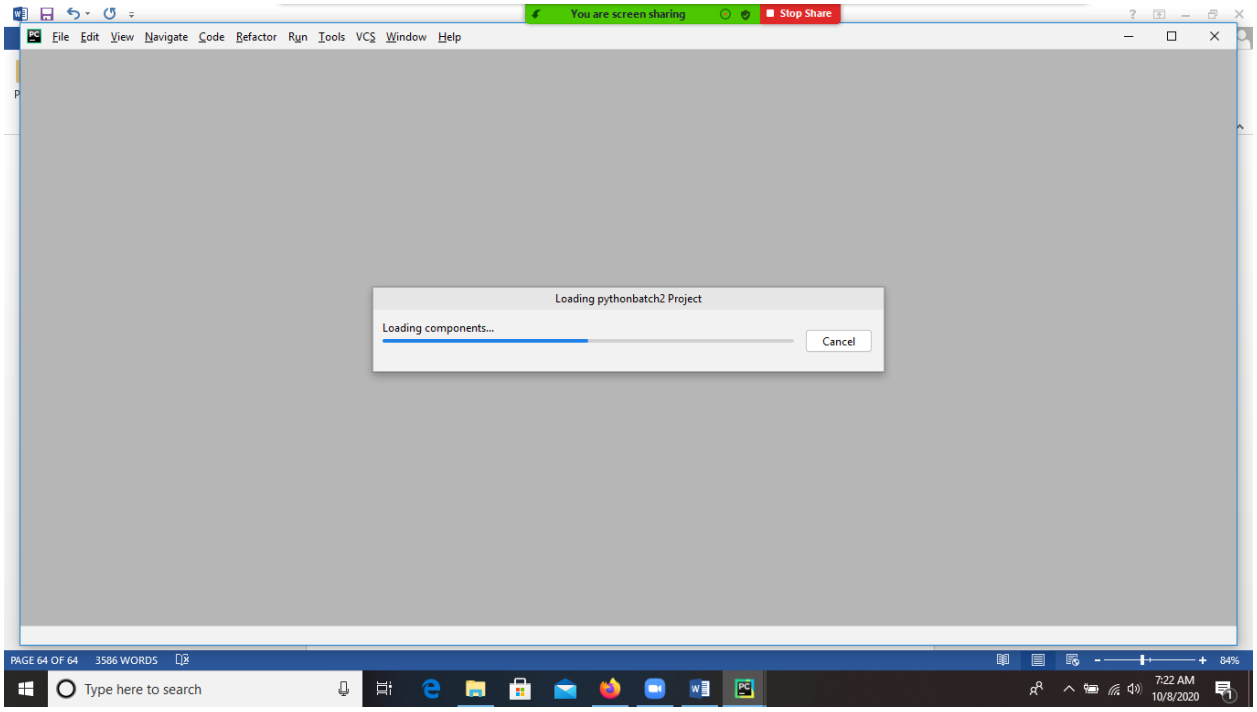

You are screen sharing
Stop Share
Customize PyCharm
UI Themes → Featured plugins
Set UI theme
Darcula
Light
Breakpoints
Python Line Breakpoint
fib.py:5
Any exception
You can change the UI theme later in Settings | Appearance & Behavior | Appearance. Additional themes are available in Settings | Plugins.
Skip Remaining and Set Defaults
Next: Featured plugins

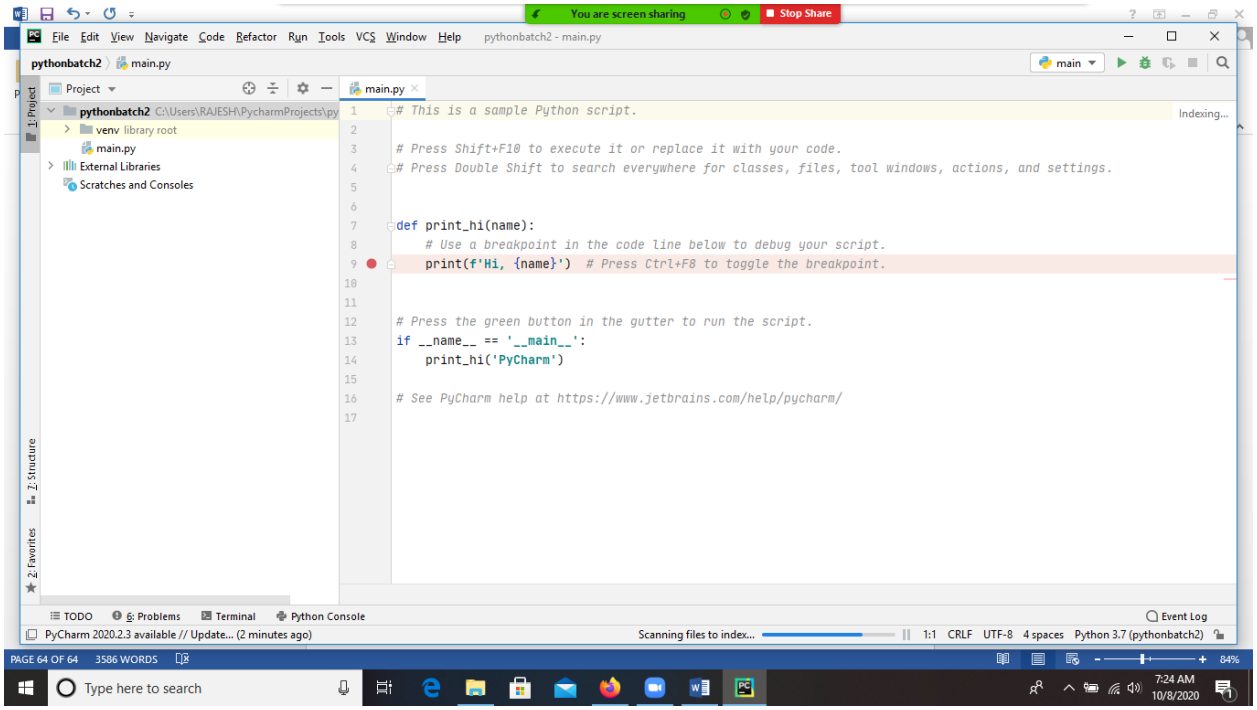

You are screen sharing
Stop Share
Customize PyCharm
UI Themes → Featured plugins
Download featured plugins
We have a few plugins in our repository that most users like to download. Perhaps, you need them too?
IdeaVim
Editor
Emulates Vim editor
Recommended only if you are familiar with Vim.
Install and Enable
R
Custom Languages
R language support
Install
AWS Toolkit
Cloud Support
Create, test, and debug serverless applications built using the AWS Serverless Application Model
Install
New plugins can also be downloaded in Settings | Plugins
Skip Remaining and Set Defaults
Back to UI Themes
Start using PyCharm


You are screen sharing
Stop Share
Welcome to PyCharm
PyCharm
Version 2020.2.2
New Project
Open
Get from Version Control
Events
Configure
Get Help
Type here to search
7:20 AM
10/8/2020


You are screen sharing
Stop Share
Welcome to PyCharm
PyCharm
Version 2020.2.2
New Project
Open
Get from Version Control
Events
Configure
Get Help
Type here to search
7:20 AM
10/8/2020


You are screen sharing
Stop Share
New Project
Location: C:\Users\RAJESH\PycharmProjects\pythonbatch2
Python Interpreter: New Virtualenv environment
New environment using Virtualenv
Location: C:\Users\RAJESH\PycharmProjects\pythonbatch2\venv
Base interpreter: C:\Users\RAJESH\AppData\Local\Programs\Python\Python37\python.exe
Inherit global site-packages
Make available to all projects
Existing interpreter
Interpreter: <No interpreter>
Create a main.py welcome script
Create a Python script that provides an entry point to coding in PyCharm.
Create
Cancel
PAGE 63 OF 63
3586 WORDS
Type here to search
7:21 AM
10/8/2020


You are screen sharing
Stop Share
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Loading pythonbatch2 Project
Loading components...
Cancel
PAGE 64 OF 64
3586 WORDS
Type here to search
7:22 AM
10/8/2020



9-10-2020

Reading Character data from file or text file

1.read()→to read total data from the file

2.read(n)→to read N charecters from the file

3.readline()→to read only one line

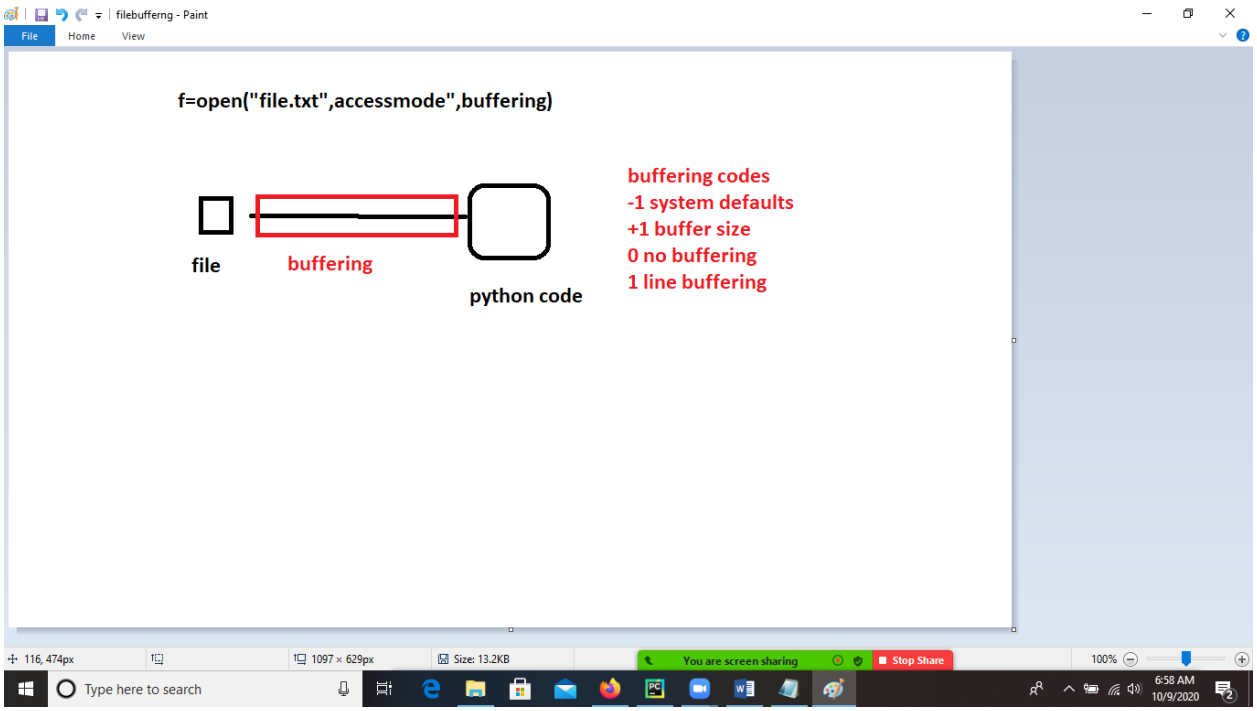
4.readlines()→to read all lines into a list

Prorgaram31: wpp to check the chareters byb using read() methods
f=open('readdata.txt','r') data=f.read(12) print(data) f.close()
#3 =raj *#7=rajesh* *#9rajesh* *How*
Note :ever line ends with new line called /n..it will take one dharecter value

Program 32: wpp to check the read(n) ,readline() methods
k=open('readdata.txt','r') output1=k.readlines() print(output1)

```
output2=k.readline()
print(output2)
```

['rajesh\n', 'how \n', 'fine \n', 'go to school']→output1
Rajesh→output2



Program33:wpp to write a new file in path

```
f=open('rajesh.txt','x')
#x write the file
f.write('i am good boy')
f.close()
print("ok data insertred go and check the path")
```

```
ok data insertred go and check the path
```

Program 34: wpp to read and write the file at a time

```
f=open('rajesh1.txt','x')
#x write the file
f.write('i am good boy')
f.close()
print("ok data insertred go and check the path")
f=open('rajesh.txt','r')
print(f.read())
```

```
ok data insertred go and check the path
I am a good boy
```

Program 35:wpp to remove the file

```
import os
os.remove('rajesh1.txt')
print('file removed')
```

Program 36: wpp to delete the directory or folder

```
import os

os.rmdir("ramu")
print("removed EMPTY FOLDER… go and check the floder ")
```

```
removed EMPTY FOLDER… go and check the floder
```

10-10-2020

ERRORS AND EXCEPTION HADLING

EXAMPLE : 5LINES OF CODE C/C++/JAVA/PYTHON

1

2

3→ERROR (IF WE APPLY EXCEPTION)

4

5

OUTPUT1:

ERROR

OUTPUT2

GOOD RESULTS

EXMP:2:

Bank

Insert atm

Pin enter

Transaction (error)(exception handling)

Remove Atm

Output:

Transaction failed and go and visit bank

Difference between error and exception handling

error	Excpetion handling
User /system/ide can create the error	It can covered the error
Impossiable to recover the error	Possiable to recover the error
Errors are unchecked type	It can be checked or unchecked
Its happen at runtime	It can happen at compile time
Caused by the environment on which application is running	Caused byapplication

Example :
A=10;
B=20
C=a/0
Print(c)
Output:
c=a/0
ZeroDivisionError: division by zero
	Example :
	`try:`
	`  a=10`
	`  b=20`
	`  c=a/0`
	`  print("the value of c",c)`
	`except:`
	`  print("this cant divided by zero")`
	output:
	`this cant divided by zero`

Exception hadling :

Exception hadling is an event ,which occurs during the execution of a program ,that checks the normal flow of programs

Synatax

Try-expect statement

Try:

#block of code

Expect Exception1:

#block of code

Expect Exception 2:

 # block of code

#other code

Program 37: wpp to check the exception at compile time

```
try:
    a=10
    b=20
    c=a/0
    print("the value of c",c)
except:
    print("this cant divided by zero")
```

```
this cant divided by zero
```

Program 38:wpp to check the try and except block

```
try:
    a=int(input("enter a value::"))
    b=int(input("enter a value::"))
    c=a/b
    print("the result is",c)
except Exception :
    print("this cant divide by zero")
    print(Exception)
else:
    print("i am else block")
```

Case 1:enter a value::20
enter a value::10
the result is 2.0
i am else block

Case 2:enter a value::10
enter a value::0
this cant divide by zero
<class 'Exception'>

Case 3:enter a value::0
enter a value::-1
the result is -0.0
i am else block

Program 39: wpp to check exception handling in list

```
try:
    list=[1,2,3]
    print(list(3))
except(TypeError):
    print("yes there is a typing error ..please check it ")
except(IndexError):
    print("yes there is a index problem..please check it ")
```

```
yes there is a typing error ..please check it
```

Program 40:wpp to check the excpetion handling in files

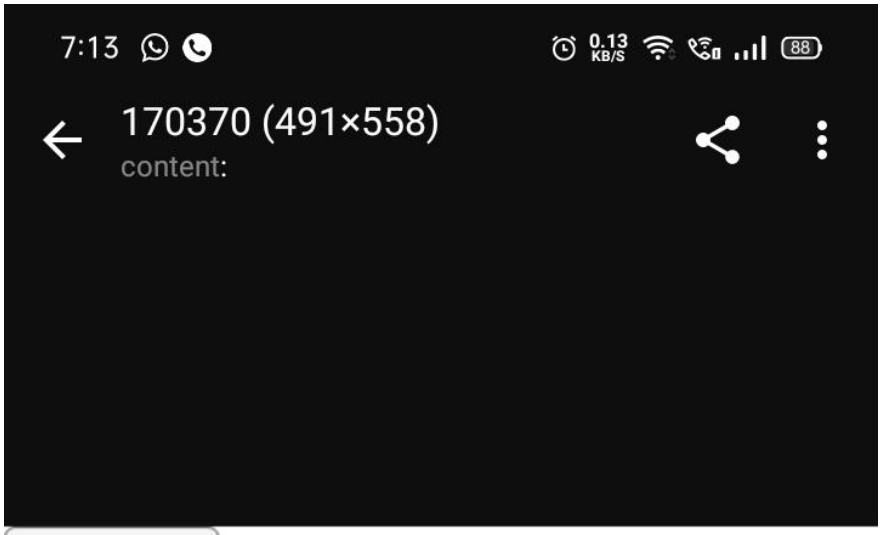
```
try:
    f=open('abc1.txt','r')
    print(f.read())
    print("yes readed")
except(NameError):
    print(" yes name error is there please change the name")
```

```
except(FileNotFoundError):
    print(" file is not found ")
```

```
hi howaare
yes readed
yes name error is there please change the name
file is not found
```

Exception	Cause of Error
AssertionError	Raised when an assert statement fails.
AttributeError	Raised when attribute assignment or reference fails.
EOFError	Raised when the input() function hits end-of-file condition.
FloatingPointError	Raised when a floating point operation fails.
GeneratorExit	Raise when a generator's close() method is called.
ImportError	Raised when the imported module is not found.
IndexError	Raised when the index of a sequence is out of range.
KeyError	Raised when a key is not found in a dictionary.
KeyboardInterrupt	Raised when the user hits the interrupt key (Ctrl+C or Delete).
MemoryError	Raised when an operation runs out of memory.
NameError	Raised when a variable is not found in local or global scope.
NotImplementedError	Raised by abstract methods.
OSError	Raised when system operation causes system related error.
OverflowError	Raised when the result of an arithmetic operation is too large to be represented.
ReferenceError	Raised when a weak reference proxy is used to access a garbage collected referent.
RuntimeError	Raised when an error does not fall under any other category.
StopIteration	Raised by next() function to indicate that there is no further item to be returned by iterator.
SyntaxError	Raised by parser when syntax error is encountered.
IndentationError	Raised when there is incorrect indentation.
TabError	Raised when indentation consists of inconsistent tabs and spaces.
SystemError	Raised when interpreter detects internal error.
SystemExit	Raised by sys.exit() function.
TypeError	Raised when a function or operation is applied to an object of incorrect type.
UnboundLocalError	Raised when a reference is made to a local variable in a function or method, but no value has been bound to that variable.
UnicodeError	Raised when a Unicode-related encoding or decoding error occurs.
UnicodeEncodeError	Raised when a Unicode-related error occurs during encoding.
UnicodeDecodeError	Raised when a Unicode-related error occurs during decoding.

UnicodeTranslateError	Raised when a Unicode-related error occurs during translating.
ValueError	Raised when a function gets an argument of correct type but improper value.
ZeroDivisionError	Raised when the second operand of division or modulo operation is zero.



try

{ Run this code }

except

{ Run this code if an exception occurs }

else

{ Run this code if no exception occurs }

finally

{ Always run this code }

12-10-2020

Object oriented programing through Python

We can create class and objects

Oops vs popl

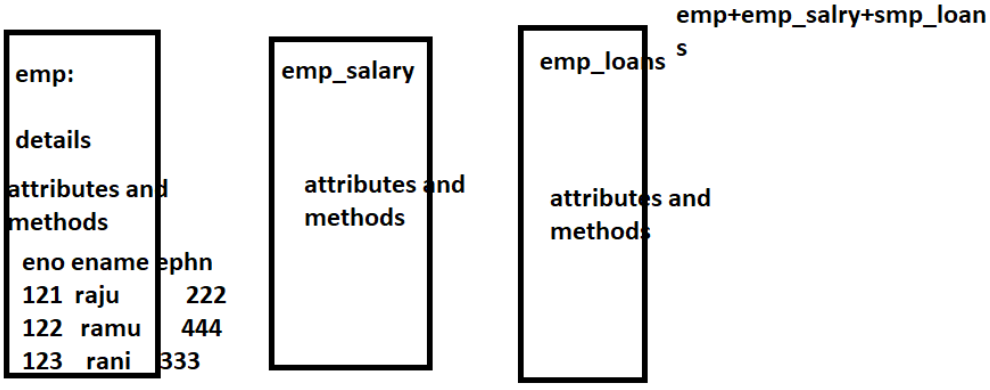
Principlas in oops

- ✓ Class
- ✓ Object
- ✓ Method
- ✓ Encapsulation
- ✓ Inheritance
- ✓ Polymorphism
- ✓ Data abstraction

Class :

The class can be defined as a collection of objects

The logical entity of attributes and methods



Class synatax

Class <userdefined _name>:

Variables/attributes

Statements …..

Example:

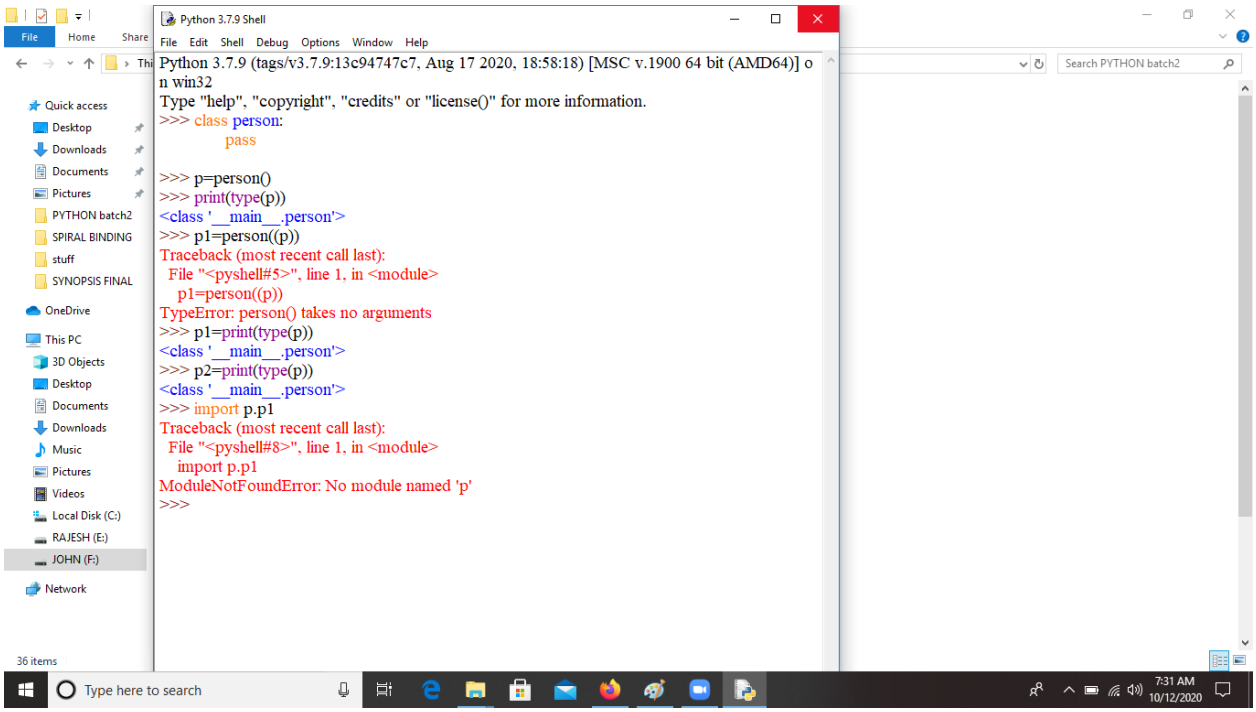
Class person: //class name

Pass

P=Person() //object reference

```
>>> class person:
        pass

>>> p=person()
>>> print(type(p))
<class '__main__.person'>
```



Object:object ia an entity that has state and behavior .

It may be any real word objects like pen,mouse,key board ,table ,car …etc

Syntax:

Class person:

Method_name:

Self.a=a

Self.b=b

Method :

The method is a funcation that is associated with an object

A method is not unique to class instance

But any object type can have methods

Syntax:

Class person: #class name

……

def __init__(self,eno,ename) #method with parameters /attributes

self.eno=eno  #instance objetcs

self.ename=ename

def disply(self): #method

print(self.eno,self.ename)

s=person(121,'ramu')

Program 41`:wpp to test the objects and methods

```
#zero attributes
class person:
    print("hi i am a class my name is person")
    a=10
    b=20
    name='raju'
    def disply(self):
        print(" hi i am a method ",self.a,self.name)
        c=self.a+self.b
        print(c)
s1=person()
s1.disply()
#class
#varlabels
#method
#instance objetcs
#object reference
#method calling
```

```
hi i am a class my name is person
 hi i am a method  10 raju
30
```

13-10-2020

Program 42: wpp to define class an methods and its arguments

```
class sum:
    def __init__(self,a,b): #constructor
        self.a=a
        self.b=b
        print("self is used as a reference variable which referes to the
current class objects")
    def output(self):  #method
            c=self.a+self.b
            print(c)
s1=sum(10,20) #object reference
s1.output()   #method calling
```

```
self is used as a reference variable which referes to the current class objects
30
```

Program 43:wpp to diplay employee details by using with out self arguments and delete the any object if you like

```
class Employee:
    id=10
    name="rani"
    def disp(self):
        print("my id is::%d\nmy name::%s"%(self.id,self.name))
e1=Employee()#
e1.disp()#
del e1.id
```

with self

```
class Employee:
    def __init__(self,id,name):
        self.id=id
        self.name=name

    def disp(self):
        print("my id is::%d\nmy name::%s"%(self.id,self.name))
e1=Employee(10,"rani")
e1.disp()
e2=Employee(101,"vani")
e2.disp()
```

Mini project:

Bank concept by using oops

Class

Methods

__init__

Accountopen 1000

Deposit 200

Withdraw 100

1000+200-100=

Switch case

Constructor :

A constructor is a special type of method which is used to intilize the instance memebers of the class

C++,java the constructor name same as class name .but in python we are using __init__(self) method

1.this method is called when the class is intilized

2.it accept the self keyword as a first argument

3.which allows the attributes or methods of a class

Two types of constructors

1.parametarized

2.default

Program 44:wpp to check with parametrs and without parametrs by using constructors

With parameters

```
class Student:
    def __init__(self,name):
        print("this is parametarized concept ")
        self.name=name
    def show(self):
        print("this is method")
        print("hello",self.name)
s=Student("rajesh")
s.show()
```

without(default)

```
class Student:
    name="rajesh"
    def __init__(self):
        print("this is withoutparametarized concept ")
        #self.name=name
```

```
    def show(self):
        print("this is method")
        print("hello",self.name)
s=Student()
s.show()
```

this is parametarized concept
this is method
hello rajesh

Program 45:wpp to check two constructors with in one class

```
class check:
    def __init__(self):
        print(" hi i am a first constructor")
    def __init__(self):
        print(" hi i am a second construct")
s=check()
#override


#the first constructor is not accessiable the "s" object.

#internally the object of the class will always call the last constructor
#if the class has multiple constructors
```

```
hi i am a second construct
```

Encapsulation:

Binding the variables and methods called encapsulation

A=10

Name=”raju”

Def disp(self):

……

S1=class()

Variables +method=encapusulation

It is used to restrict aceeess to members and vardiables

Code and data are wrapped together with in a single unit

15-10-2020

Built in funcations :

It defined in the class

Program 46:wpp to check built in funcations in the class

```
class student :
    #constructor definition with parameters
    def __init__(self,name,id,age):
        self.name=name
        self.id=id
        self.age=age
        #create the object  of the the class
s=student("ramu",102,23)
#print the attribute name of the objects
print(getattr(s,'name'))
#reset the value of attributr age of 30
setattr(s,"age",30)
#print modified attribute
print(getattr(s,'age'))
#print true values
print(hasattr(s,'id2'))
#delete  the attribute
delattr(s,'age')
print(getattr(s,"id"))
print(getattr(s,"name"))
#or
print(s.name)
print(s.id)
```

```
ramu
30
False
102
ramu
ramu
102
```

Program 47:wpp ti find built in class attributes

```
class Person:
    ' person class can dedines the person details like name,id ,age  '
    sname="vasu"
    def __init__(self,name,id,age):
        self.name=name
        self.id=id
        self.age=age
    def disp(self):
        print("name:%s,id is:%d,age is:%d"%(self.name,self.id,self.age))
p=Person("rani",101,24)
p.disp()
print(p.__doc__) # it contfains a string which has the class document
print(p.__dict__)#dictonary contains information about class name spaces
print(p.__module__)
#print(p.__bases__)
print(p.__name__) #itis used to access the class name
```

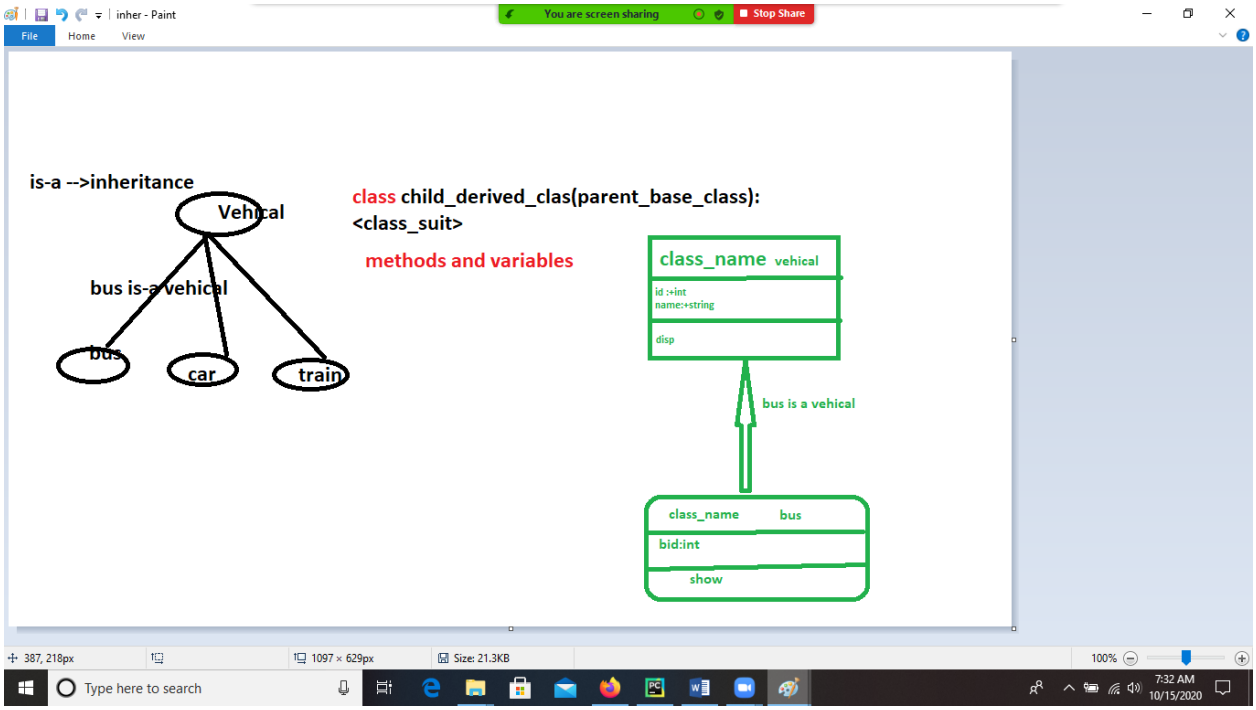
```
name:rani,id is:101,age is:24
 person class can dedines the person details like name,id ,age
{'name': 'rani', 'id': 101, 'age': 24}
```

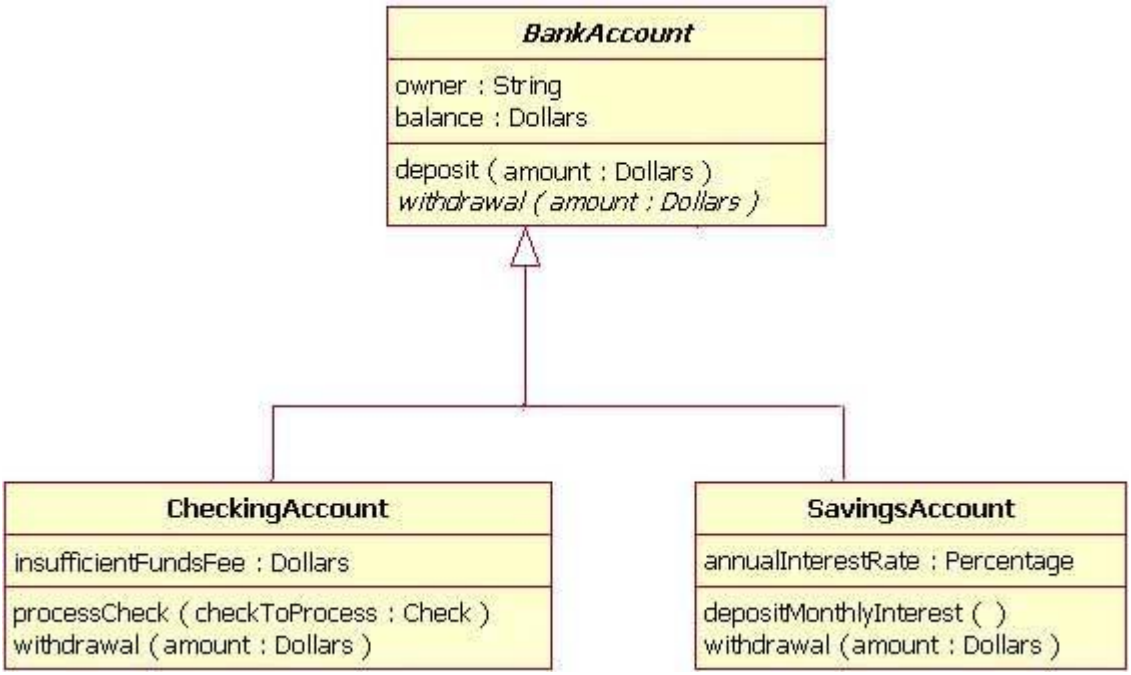
__main__

Inheritance :

OOPS

accessin properties from child class to parents class





16-10-2020

Types of inheritance

To access the multiple parents and child relationships

1.single inheritance

2.mulilevel inheritance

3.mulitiple inheritance

4.hierarchical ingheritance

1.single inheritance :

Single parent to single child class

A ←

PROGRAM 48. wpp to check single inheritance :

```
class Parent:
        pid=121
        def fun(self):
            print("my parent id is",self.pid)
class child(Parent):
    cid=212
    def fun1(self):
        print("child id is::",self.cid,self.pid)
s=Parent()
s.fun()
c=child()
c.fun1()
```

my parent id is 121
child id is:: 212 121

Program 49:wpp to define family details by using multilevel inheritance and its constuctors

```
class grandfather:
    def __init__(self,gfname):
        self.gfname=gfname
        #intermidiate class
class father(grandfather):
    def __init__(self,fname,gfname):
        self.fname=fname
        self.gfname=gfname
        #derived class or child class
class son(father):
    def __init__(self,sname,fname,gfname):
        self.sname=sname
        self.gfname=gfname
        self.fname=fname
    def print_name(self):
        print("grandfather name is ::",self.gfname)
```

```
        print("father name is::",self.fname)
        print("son name is ::",self.sname)
s=son("ramu","vasu","ranga")
print(s.gfname,s.fname,s.sname)
s.print_name()
# define parent and child class relationships
#properitce variables and methods usage
```

ranga vasu ramu
grandfather name is :: ranga
father name is:: vasu
son name is :: ramu

Program:50 wpp to find the analysis of multiple inheritance

```
class Father:
    def fun1(self):
        print("i ama father")
class mother:
    def fun2(self):
        print("i am a mother")
class son(Father,mother):
    def fun3(self):
        print(" i am a son")
s=son()
s.fun1()
s.fun2()
s.fun3()
```

i ama father
i ama mother
i am a son

17-10-2020

HIERARCHICAL INGHERITANCE

parent to child

PROGRAM 51:WPP TO find hierarchical inheritance by the concept of college and students

```
class college:
    def cdetails(self,cname,code):
        self.cname=cname
        self.code=code
        print("your college name is::",self.cname)
        print("your college code is::", self.code)
class student1(college):
    def stu1(self,sname,sid,smarks):
        self.sname=sname
        self.sid=sid
        self.smarks=smarks
        print("the student name is ::",self.sname)
        print("the studnet id is ::",self.sid)
        print("the student marks are ::",self.smarks)
class student2(college):
    def stu2(self,sname,sid,smarks):
        self.sname=sname
        self.sid=sid
        self.smarks=smarks
        print("the student name is ::",self.sname)
        print("the studnet id is ::",self.sid)
        print("the student marks are ::",self.smarks)
class student3(college):
    def stu3(self,sname,sid,smarks):
        self.sname=sname
        self.sid=sid
        self.smarks=smarks
        print("the student name is ::",self.sname)
        print("the studnet id is ::",self.sid)
        print("the student marks are ::",self.smarks)
c=college()
s1=student1()
s1.stu1("raju",123,80)
s1.cdetails("PRK",12345) #HIERARCHICAL INHERITANCE

s2=student2()
s2.stu2("rani",123,87)
s2.cdetails("PRK",12345) ##HIERARCHICAL INHERITANCE

s3=student3()
s3.stu3("vasu",123,67)
s3.cdetails("PRK",12345) #HIERARCHICAL INHERITANCE
```

```
the student name is :: raju
the studnet id is :: 123
the student marks are :: 80
your college name is:: PRK
your college code is:: 12345
the student name is :: rani
the studnet id is :: 123
the student marks are :: 87
your college name is:: PRK
your college code is:: 12345
```

```
the student name is :: vasu
the studnet id is :: 123
the student marks are :: 67
your college name is:: PRK
your college code is:: 12345
```

ISSUBCLASS(SUB,SUP) IS USED TO CHECK THE relationships between the specified class

It returns true if the first class is the subclass

Other wise it returns false

Isinstance (obj,class) is used to check the relationships between the object and classes

If it is true it will return first parameter

Program 52:wpp to check the objetcs and classes and sub class relationships

```
class Parent:
    def food(self):
        print("my chid come and eat food")
class Child(Parent): #inheritance
    def play(self):
        print("no ma ..i wnat to play")
c=Child()
c.food()
c.play()
print(issubclass(Child,Parent))
print(issubclass(Parent,Child))
print(isinstance(c,Child))
print(isinstance(c,Parent))
```

```
my chid come and eat food
no ma ..i wnat to play
True
False
True
True
```

19-10-2020

Access modifiers

Is used to control the methods and variables

3 types

Public

Protected

Private

Program52:wpp to test the access modifiers in python

```
lass Bank1:
    def __init__(self,name,phone,pwd):
        self.pep=name     #public attribute
        self._mob=phone   #protected attribute
        self.__bank=pwd    #private attribute
        b= Bank1("sbi",88888,123)
        print(b.pep)
        print(b._mob)
        print(b._Bank1__bank)
```

Output:
Sbi
8888888
123

```
class Bank1:
    def __init__(self,name,phone,pwd):
        self.pep=name     #public attribute
        self._mob=phone   #protected attribute
        self.__bank=pwd    #private attribute
        #set and gets
    def setmob(self,phone):
            self._mob=phone
    def getmob(self):
            return self._mob
    def setbank(self,pwd):
        self.__bank=pwd
    def getbank(self):
        return self.__bank

b= Bank1("sbi",88888,123)
print(b.pep)
print(b.getmob()) #protected accessing
print(b.getbank()) #private acessing
print(b._mob)
print(b._Bank1__bank)
```

```
<__main__.Bank1 object at 0x00000209C2E35788>
sbi
88888
123
88888
```

```
123
```

Program53:wpp to test to accesss the access modifiers by using inheritance

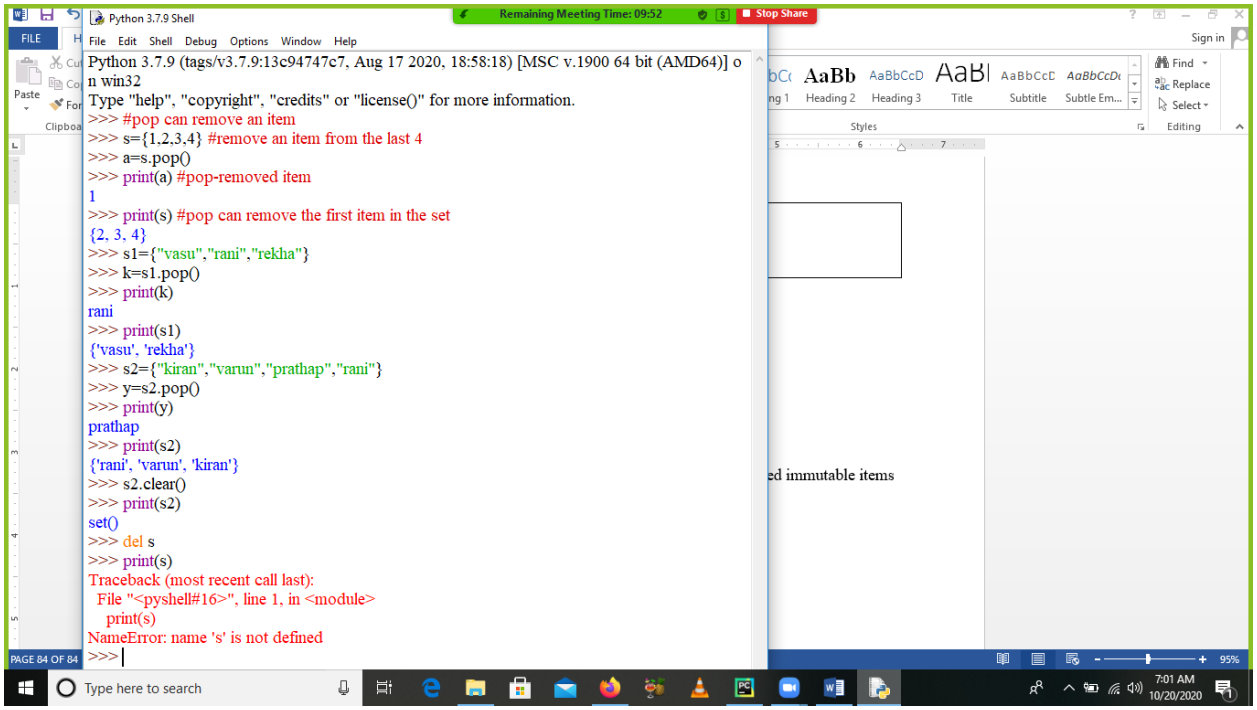
```
class test:
    name="rani"
    _phone=999
    __pwd=123
    def name1(self):
        print(self.name)
        print(self._phone)
        print(self.__pwd)
class test1(test):
    def name2(self):
        print(self.name)
        print(self._phone)
        print(self._test__pwd)
t=test()
t.name1()
t1=test1()
t1.name2()
```

```
rani
999
123
rani
999
123rani
999
123
rani
999
123
```

Python set:

- Unordered collection of element
- Each elemet must be unique
- It is a immutable or mutable :immutable …update
- There is no index to access: yes
- How to create sets?
- The set can be created by enclosing the comman separated immutable items with the curly braces {}
- Python provides the set() method

How to create the sets ??



Python set operations :

Set maths

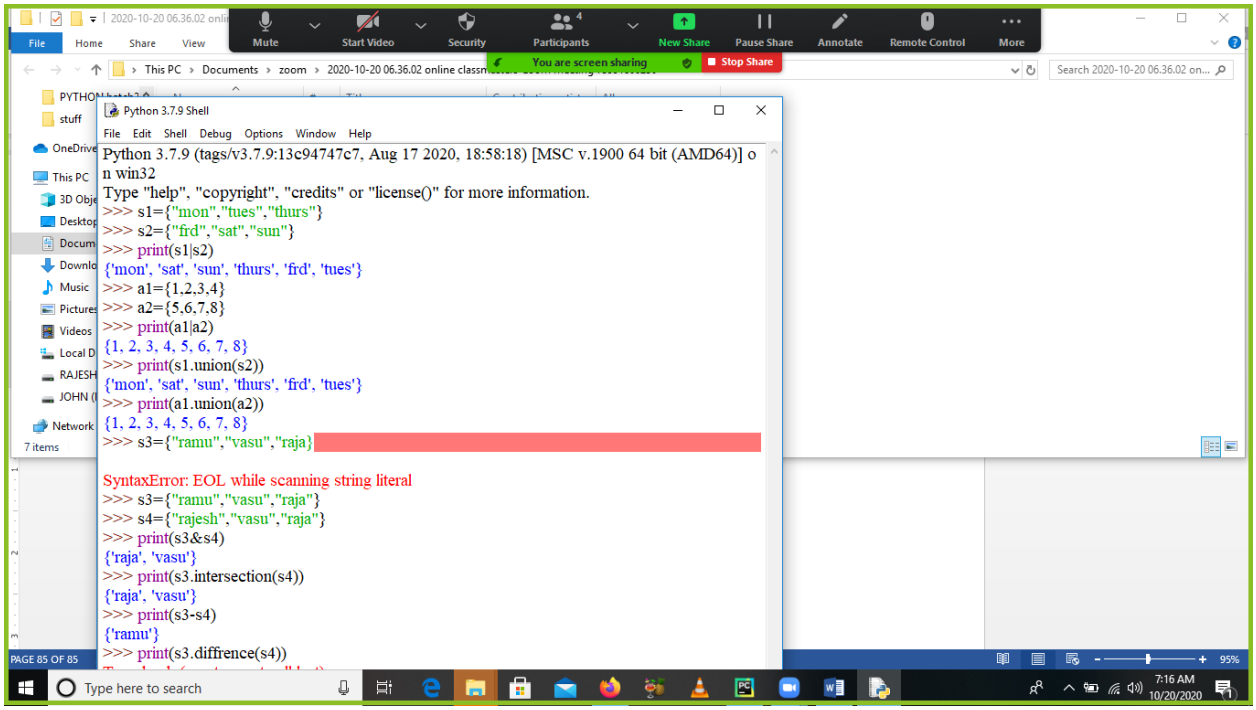
Union

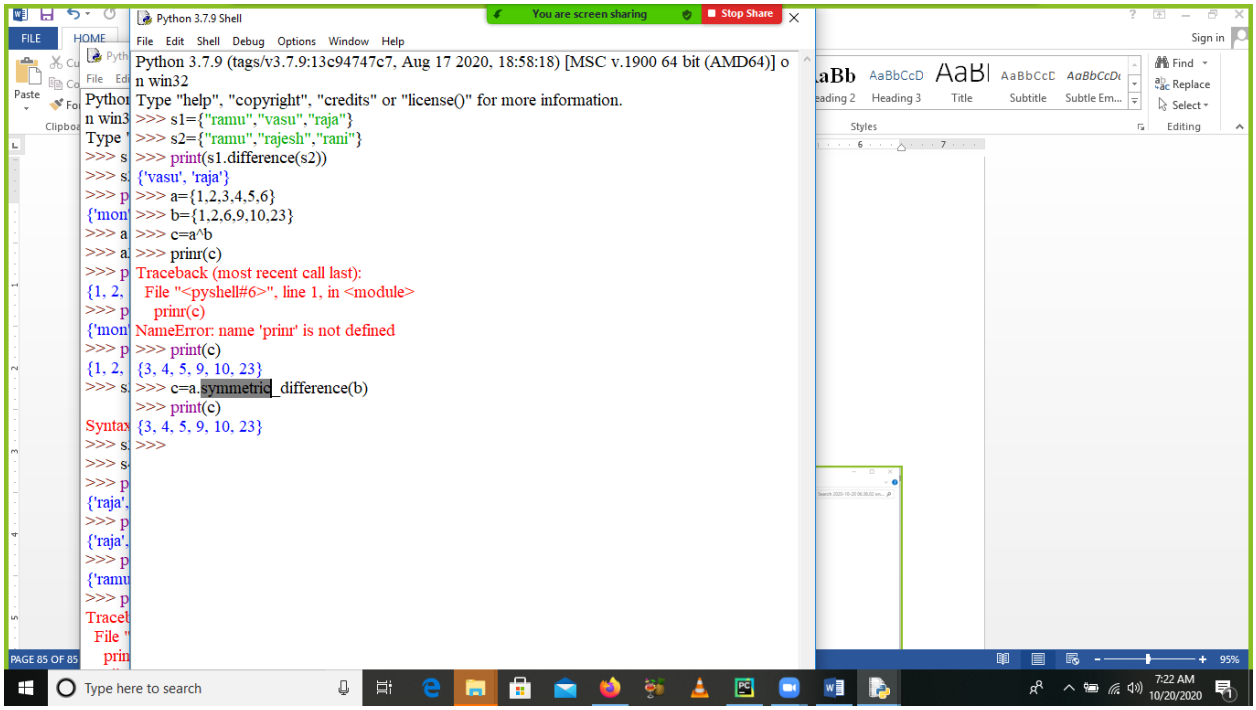
Intersection

Difference

Subtraction

symmetric





union	sub	sy	

list	tuple	Dictionary	set
order	order	order	unorder
Index	Index	index	No index

im			

21-10-2020

Program 54:wpp to print set results by using for loop

```
names={"hema","ganesh","devi","geetha"}
print(names)
print(type(names))
print(names.__sizeof__())
print("for loop results::")
for i in names:
    print(i)
```

```
{'hema', 'ganesh', 'devi', 'geetha'}
<class 'set'>
200
for loop results::
hema
ganesh
devi
geetha
```

Program 55:wpp to check immutable or mutable in set operations

```
set1={1,2,3.5,"geetha",45.8,1}
print(type(set1))
#immutable @no chnage
set1.add(12)
set1.add("ramu")
print(set1)
```

```
<class 'set'>
{1, 2, 3.5, 'geetha', 'ramu', 12, 45.8}
```

Imutable or not:

Program 56:wpp to comparision between sets

```
set1={1,2,3.5,"geetha",45.8,1}
set2={1,2,3}
set3={1,2,3}
print(set1>set2) #false
print(set1<set2) #false
print(set2<set3)
print(set2>set3)
print(set2==set3)
```

```
False
False
False
```

```
False
True
```

Frozen sets:

The frozen sets are immutable(no change) form of normal sets

The items of the frozen cant be changed and therefore it can be used as a key like dictionary

Frozen() is used to create the frozen set object

Iterable →is sequence is passed into method ,which convert frozen sets as a return type like for loop

Program 57:wpp to create frozen sets

```
fz=frozenset([1,2,3,4])
print(type(fz))
print("\nthe frozen set objects are....")
for i in fz:
    print(i)
    fz.add(6) #immutable-no chnage
```

```
<class 'frozenset'>

the frozen set objects are....
1
2
3
4
```

<class 'frozenset'>

the frozen set objects are....
1
Traceback (most recent call last):
File "C:/Users/RAJESH/PycharmProjects/pythonbatch2/frozencreattion.py", line 6, in <module>
fz.add(6)
AttributeError: 'frozenset' object has no attribute 'add'

Program 58:wpp to dictionaries in frozensets

```
dic={"name":"rajesh","id":123,"state":"ap","sal":12345}
print(type(dic))
fz=frozenset(dic)#key
print(type(fz))
for i in fz:
    print(i)
```

```
<class 'dict'>
<class 'frozenset'>
name
id
sal
state
```

Database connection

Any database + any highlevel language

Java jdbc,odbc(oracle,mysql,sql…..)

Python pip(is a package to manage the system which is used to install and packages /3.4) (oracle,mysql,sql…..)

Big data (spark,hadoop)amzon,azura,no sql

Communicating data base with python

DATA BASE NAME -→oracle -→import cx_Oracle (oracle 10g and 19c)

Step 1: Mysql--→how to install

https://dev.mysql.com/downloads/installer/

step 2:python install

step 3: pip install

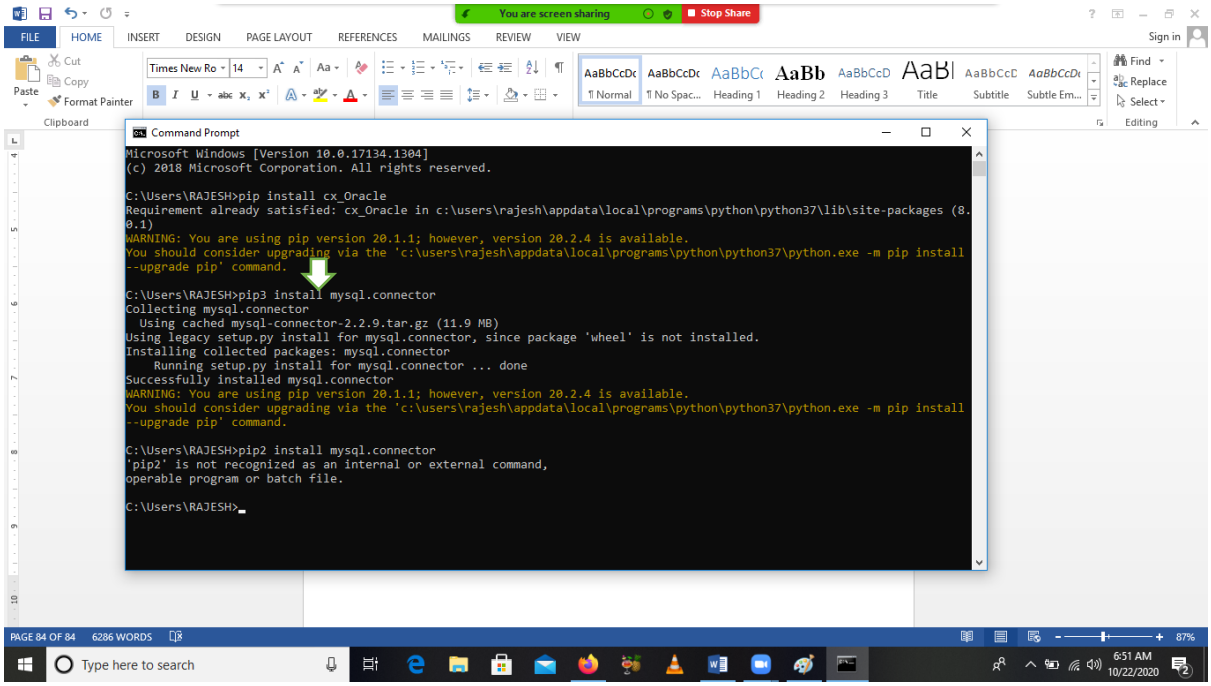
cmdprompt :pip install mysql.connector

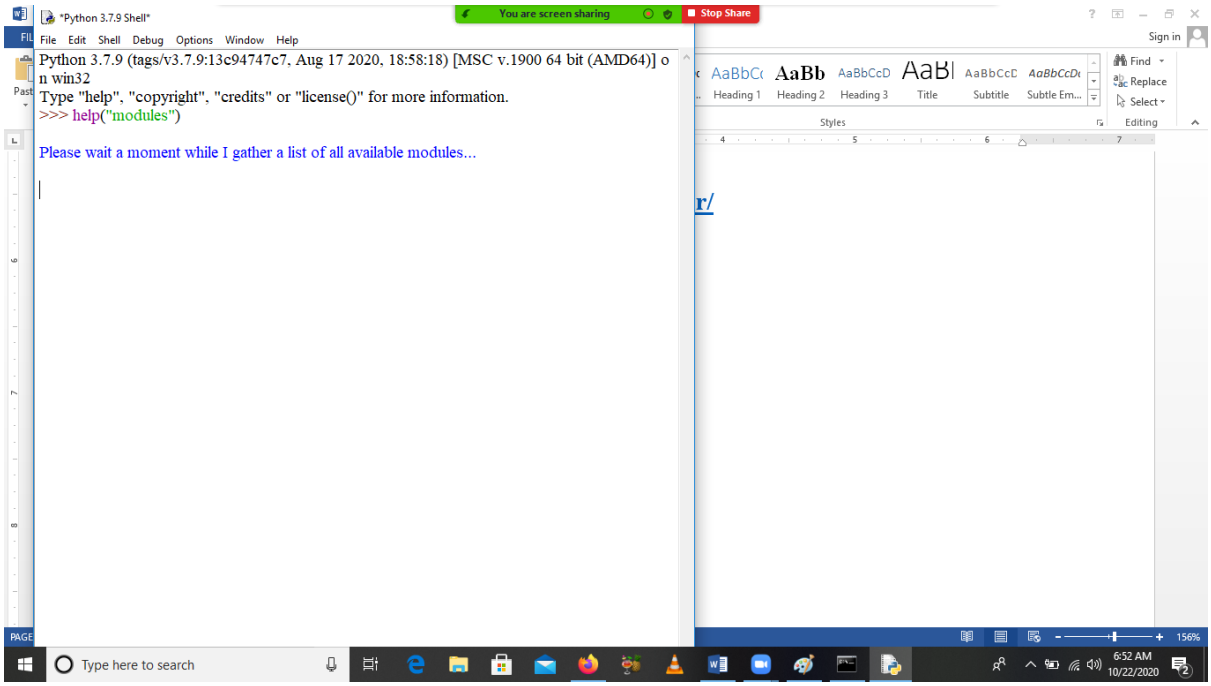
check help("modules")

…..

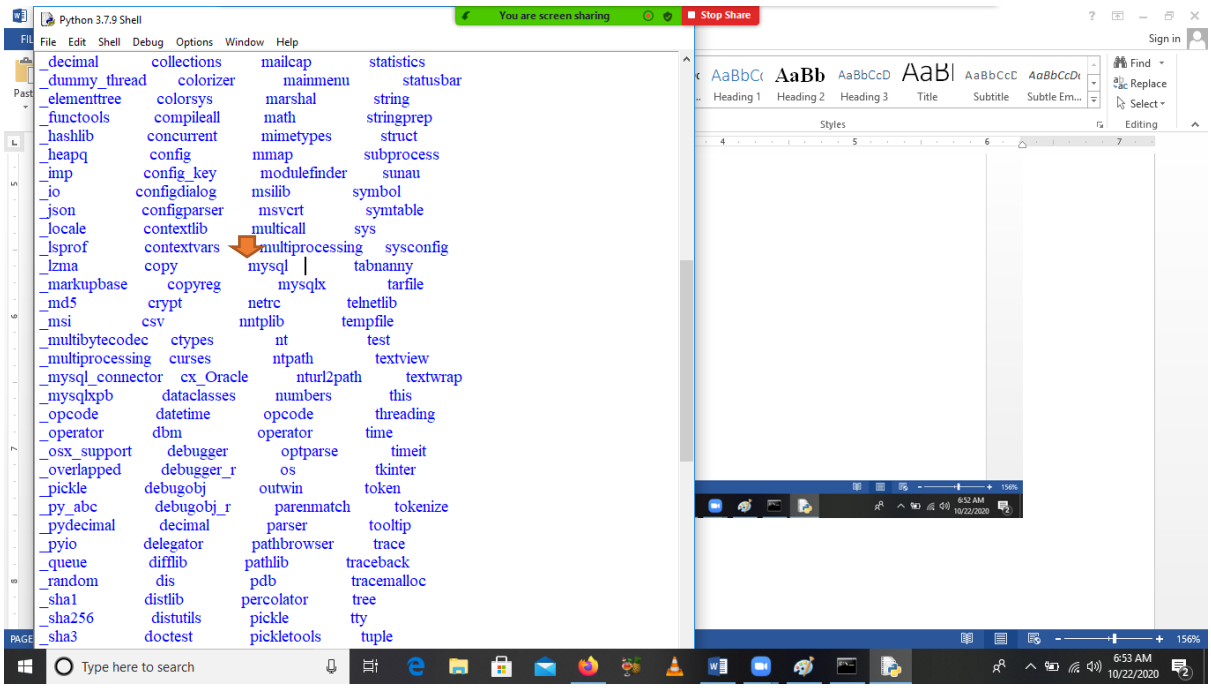
……

…….

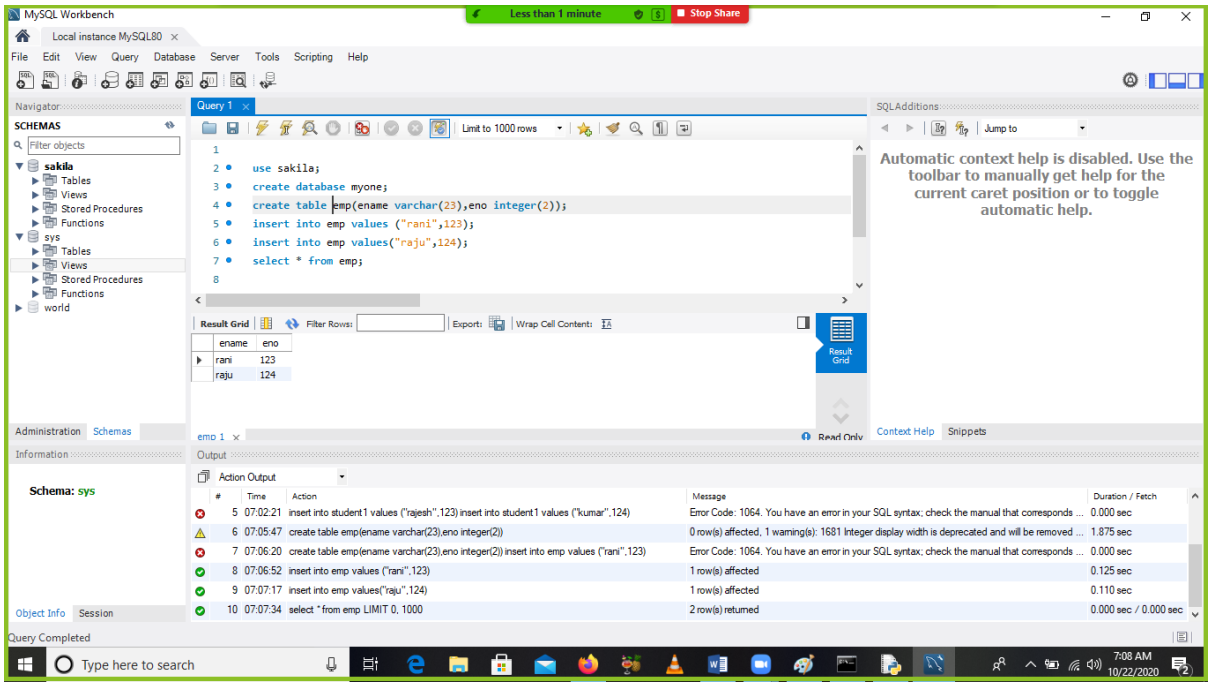




Checking python modules available or not



My sql work bench



Step4: writing python code

1. **Import the module** **import mysql.connector**
2. **Create the connection object**
3. **Create the cursor object**
4. **Execute the query**

2.Create the connection object

Connection-object=mysql.connector.connect(host name,username,password)

3.cursor :it will call the connection(2) with quries

3.1 mycur=connection-object.cursor();

4. mycur.execute("show databses ")

Program 59:wpp to check what are the data bases available in mysql byusing pip

```
import mysql.connector
cobj=mysql.connector.connect(host="localhost",user="root",password="rajesh")
mycur=cobj.cursor()
mycur.execute("show databases") #query
for i in mycur:
    print(i)
```

```
('information_schema',)
('mysql',)
('performance_schema',)
('sakila',)
('sys',)
('world',)
```

23-10-2020

Program 60:wpp to eastablish the connection between mysql and python and create a database or table through the python code

```
#module
import mysql.connector
#connection between python and mysql
mycon=mysql.connector.connect(host="localhost",user="root",password="rajesh",database="prk")
#cursor
mycur=mycon.cursor()
try:
    mycur.execute("create table email1(uname varchar(23),pwd varchar(12))")
    print("table created please check your data base")
except Exception as err:
    print(err)
mycur.close()
```

table created please check your data base

********************Thank you**********************

********************ALL THE BEST********************