

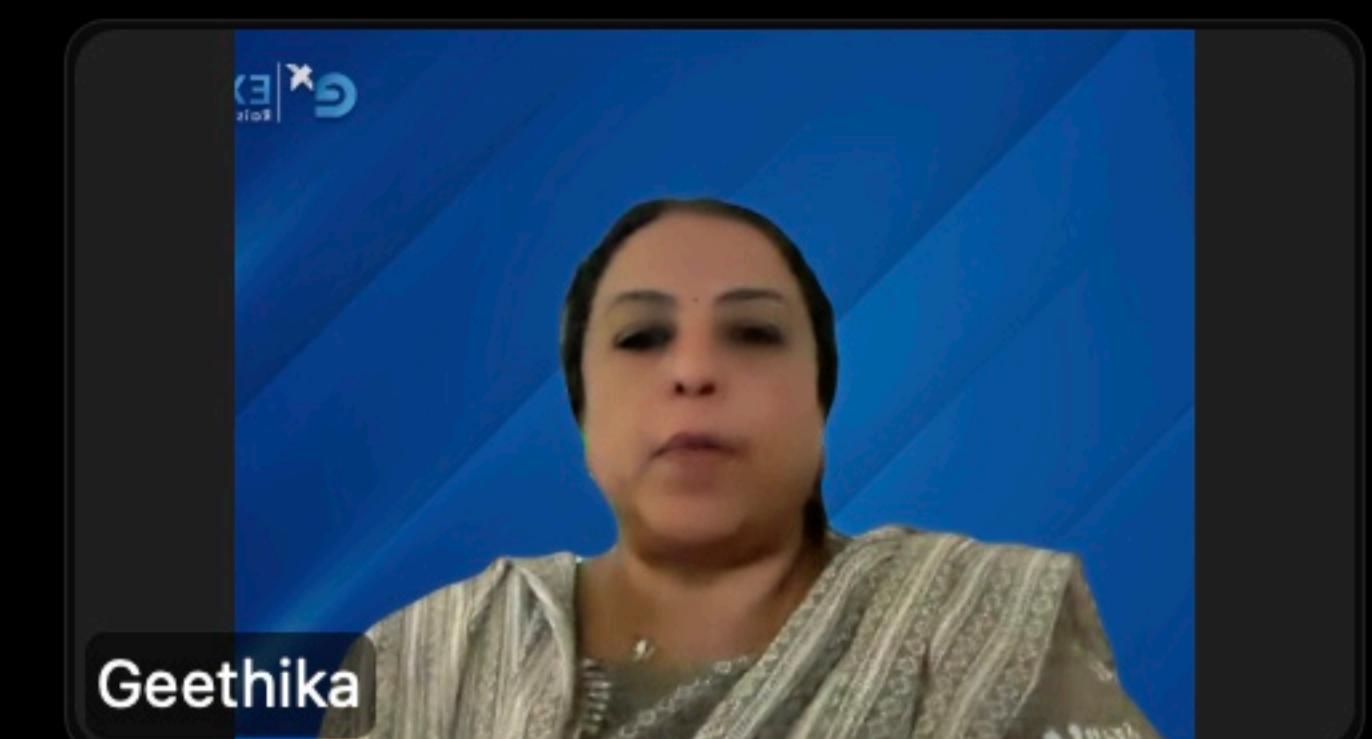
You are screen sharing zm Stop share

# Core python

Numbers → int, float  
 Boolean → bool

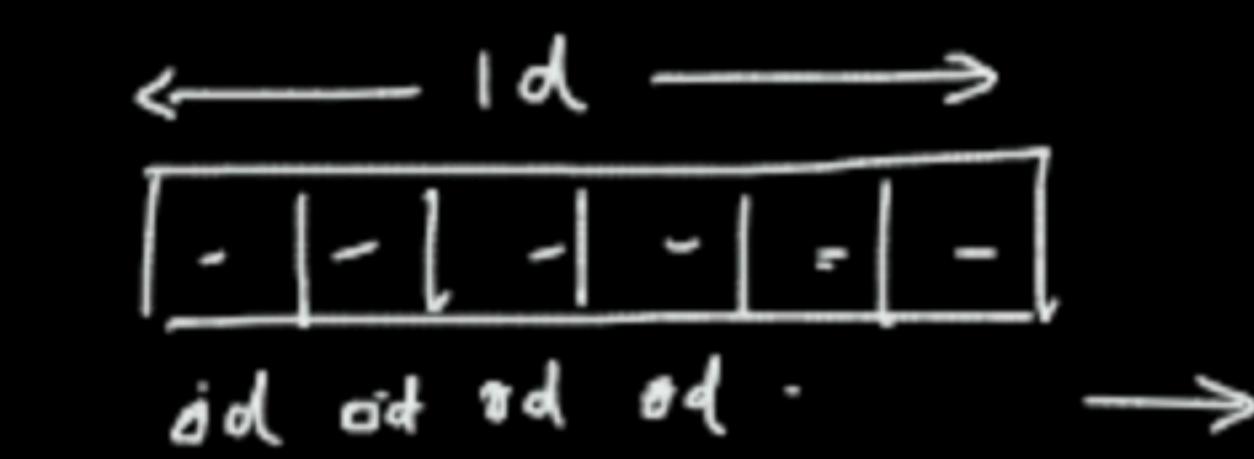
} Basic types

$x = 5$  → 0d → zero dimension  
 $y = 12.3$



Collections →

- Sequences → list, tuple, str
- Mapping → dict.



2d or higher dimension — external libraries

- pandas → 1d and 2d data
- numpy

↳ High dimension

→ Numeric fns ✓

← 2d →

c1	c2	c3	c4
-	-	-	-
-	-	-	-
-	-	-	-

1d 1d 1d 1d      3x4

↑

$$\begin{bmatrix} 7 & 8 & 9 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}_{2 \times 3}$$

2x2/3

You are screen sharing



Stop share



Modules, Packages & Libraries → Collection of Packages  
 ↳ Collection of modules  
 ↳ Collection of Variables, Fns and Classes

Variables

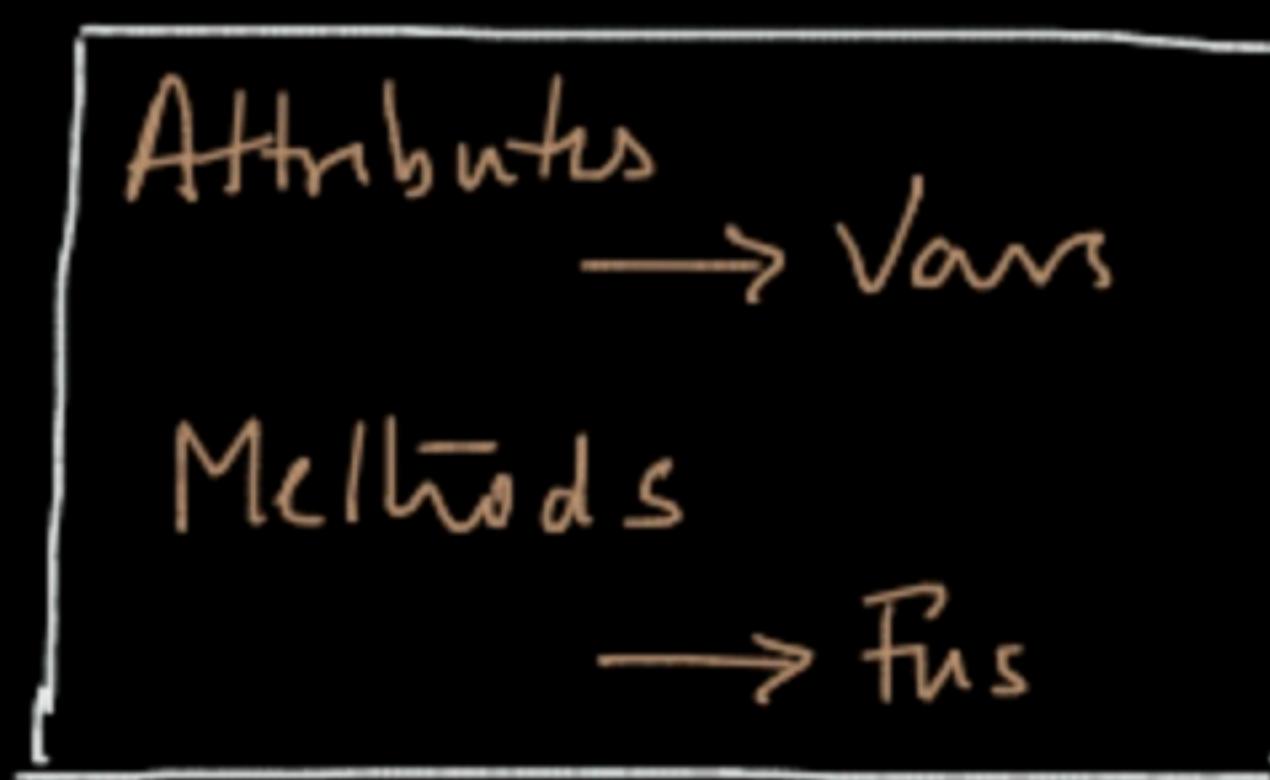
 $x = 58$  $\pi = 3.142$ 

Functions ✓

 $\text{myfn}(x,y) \rightarrow \text{val}$ 

{

→ Combination of Vars &amp; fns

My class → Title CaseClass Person ✓

→ datatype  
 → Template

Gender -  
 dob -  
 Name -  
 BG -

} Attributes

Methods

talk() .  
 Walk() .  
 Laugh()  
 Sing()

Functionality

object type  
 ↑ ↗  
 p1 = Person('M', 'Jan', '7m', 'o+ve')  
 p2 = Person('F', 'May', 'Many', 'A+ve')

type(p1) → Person  
 type(p2) → Person

p2.sing()

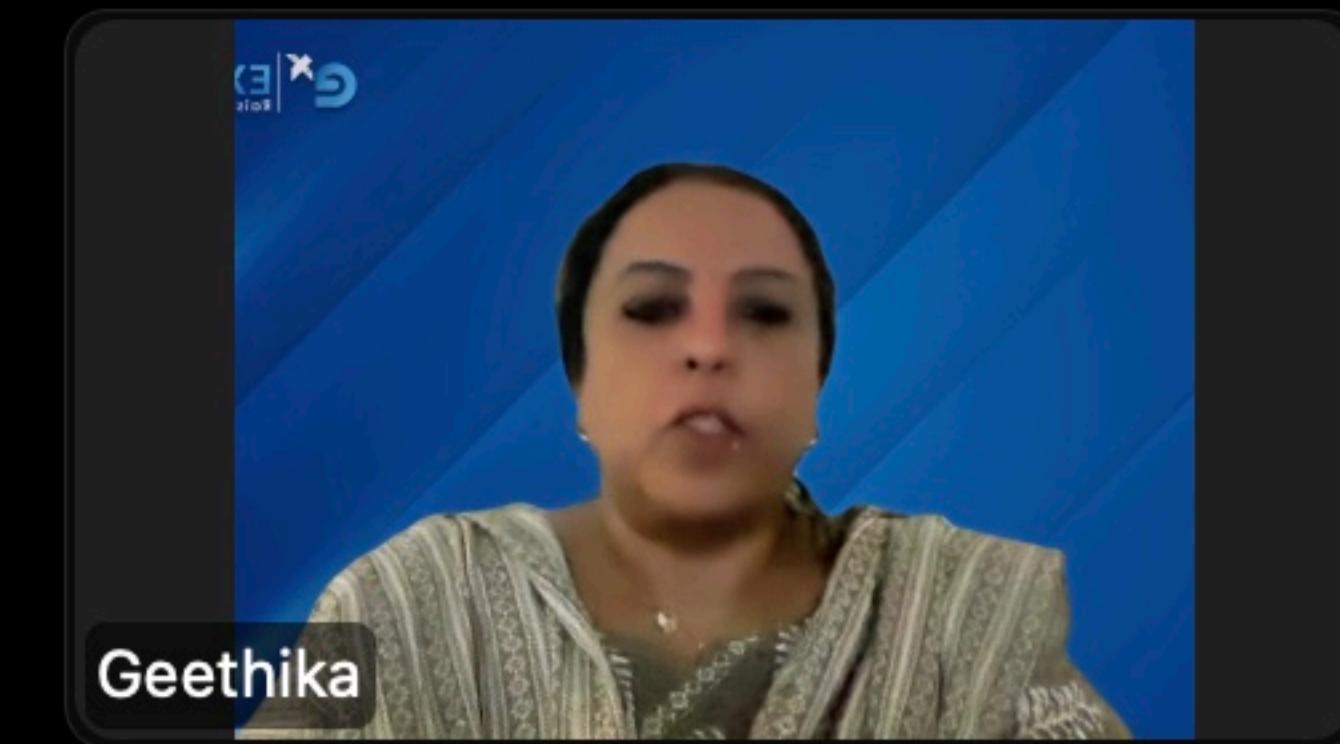
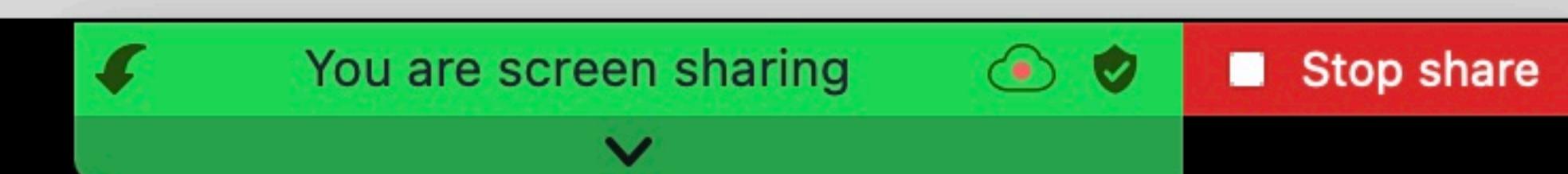
p1.talk()

p1.dance() X

type(l) ← l1 = [1, 2, 3] ✓  
 l2 = [7, 8, 9, 10] ✓  
 ↪ lists

l1.append()  
 t1 = (1, 2, 3) tuple → .





Fns → Standalone      `avg()`  
                              `gradus()`

Methods → tied to an object

Pl. talk()

Pandas → Package

Variable ✓

functions ✓

Classes ✓ → Series (1d)

→ Dataframe (2d)

2BHK →

3BHK →

$$\begin{aligned} \text{Bed} &= 3 \\ \text{Hall} &= 1 \\ K &= 1 \\ \text{Bal} &= 3 \end{aligned} \quad -$$

Villa → ✓

`h1 = Villa()`

`h2 = 3BHK()` ✓

`h3 = 3BHK()` ✓

`employee = DataFrame( - - ; ; ; )`

`student = DataFrame( data - - - - )`

You are screen sharing zm Stop share

Module → math.py

```
def log(x):
    =
    return -
```

```
def sin(x)
    =
    ret
```

```
def exp(x)
    =
    ret
```

P1. Python  
 →  
 →  
 → import math

 $y = \mathbf{math.sin(x)} + \mathbf{math.cos(x)}$ 

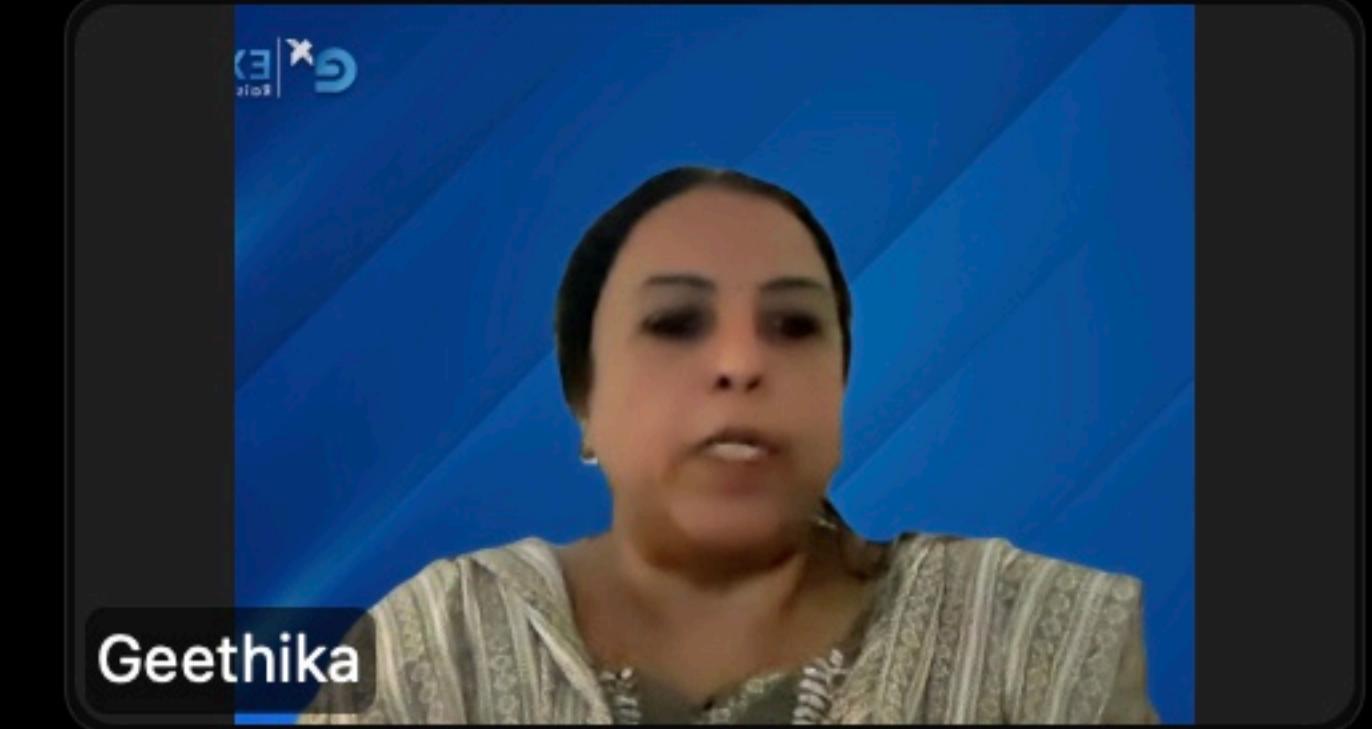
$$y = 5.4 + 6.3$$

P1. Pg 6

math.  
 └── π = 3.142  
 └── trig  
 └── sin(x)  
 └── cos(x)  
 └── tan(x)  
 └── calculus  
 └── diff  
 └── integrate  
 └── Numeric

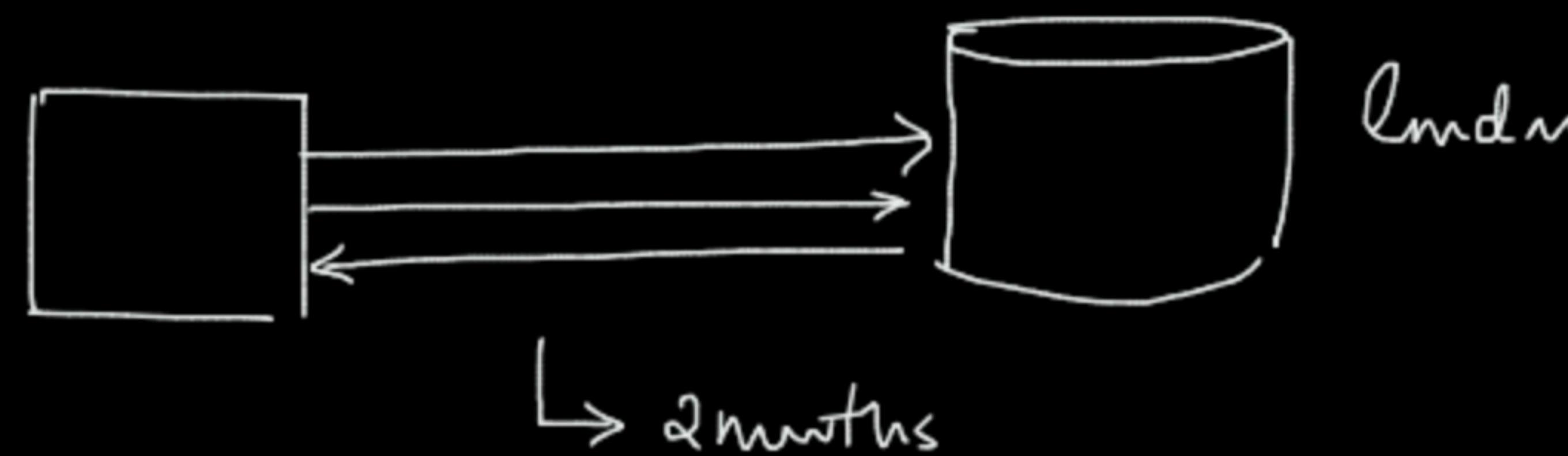
- log(x)
- exp(x)
- sqrt(x)

from math import trig  
trig.sin(x) + trig.cos(x)  
 area = math.pi × rad × rad

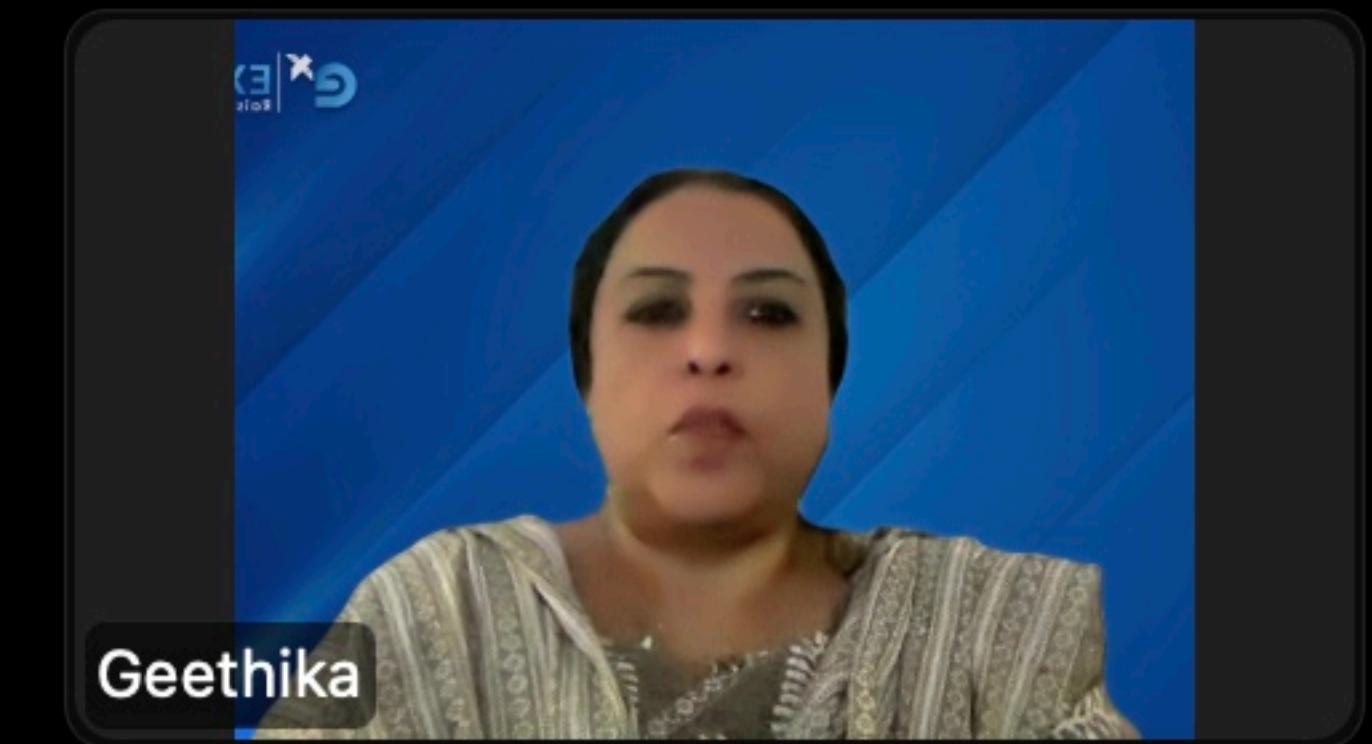


You are screen sharing

Pg n →

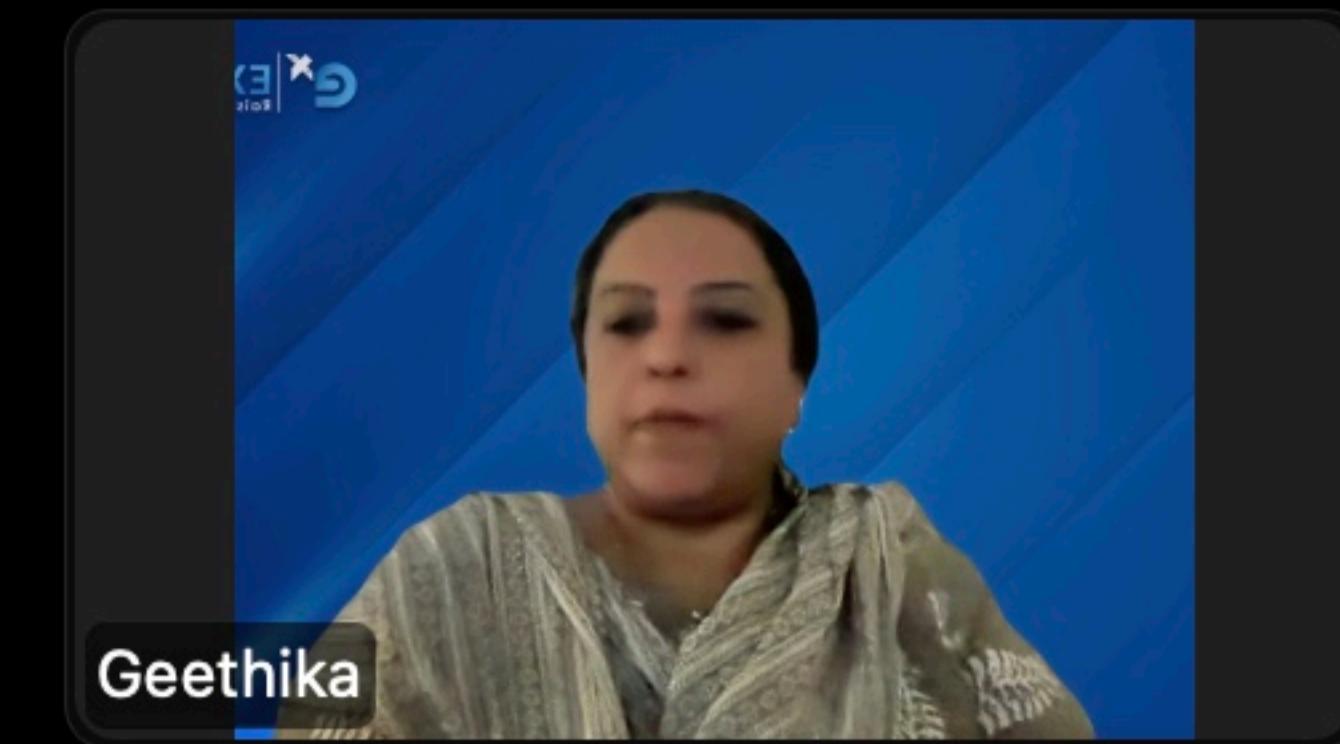
py mysql → module

✓ Connect() ✓ execute  
batch()

import pymysql as ps~~ps.~~  
~~pymysql~~.connect()~~pymysql~~.fetch()

ps

You are screen sharing



Pandas ✓

— read\_csv() ✓

— read\_html()

:

— merge().

Series →

data

shape

methods

head() —

tail() —

info() ↵

DataFrame ↵ →

data

columns ✓

shape

methods

head() ✓ ↵ } ↵

join() — } ↵

myfile.ipynb

import pandas as pd

edf = <sup>pd.</sup>pd.DataFrame( edata, columns=[id, name, job, sal] )sdf = <sup>pd.</sup>pandas.DataFrame( — — — — — )

ed	nu	an	co
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—

sdf →

sd	anne	Cms	nu
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—

type(edf) → DataFrame

edf.columns ✓

edf.head() → methods

↑ d.

sl → pandas.Series( --- ) ob

sl.info()

sdf.columns

sdf.join()

pd.  
Pandas.read\_csv()pd.  
Pandas.merge()

You are screen sharing

	0	1 ✓	2	3	
	eid	ename	job	salary	→ columns
0	e1	100	Thomas	Mgr	3500
1	e2	200	Mary	Pgmv	4500
2	e3	300	Kate	HR	3900
3	e4	400	Jim	Mgr	4100
4	e5	500	Jim	Pgmv	7600
5	e6	600	William	HR	2600

↓  
index.

Access columns →

edf[rm, col]  
 ↓      ↓  
 1 rm    1 col    2  
 2 range    2 range [ eid:job ]  
 3 random    3 random [ 1,3,5 ]

Accessing Rows →

edf.loc[rm, col]	edf.iloc[rm, col]
edf.loc['e4', :]	edf.iloc[3, :]
edf.loc['e2':'e5', :]	edf.iloc[1:5, :]
edf.loc[['e1','e4'], :]	edf.iloc[[0,3], :]

edf.loc[:, 'ename']

edf.iloc[:, 1]

Exception: omit the cols

edf.loc[:, 'eid':'job']

edf.iloc[:, :3]

edf.loc[:, ['eid', 'sal']]

edf.iloc[:, [0,3]]

