

Numbers \rightarrow int, float } Basic types
Boolean \rightarrow bool }

$$y = 12.3$$

- $0d \rightarrow$ Zero dimension

Collectms

← 1d →

-	-	-	-	-	-
---	---	---	---	---	---

od ot rd od

2d or higher dimension — external libraries

- pandas 1d and 2d data
- numpy

- numpy -

↳ 4th dimension

→ Numeric fns ✓

← 2d →

c_1	c_2	c_3	c_4
—	—	—	—
—	—	—	—
—	—	—	—

 3×4 $2 \times 2 / 3$

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

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

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



Modules, Package & libraries → Collection of packages
→ collection of modules
→ collection of Variables, Fns and Classes

Variables

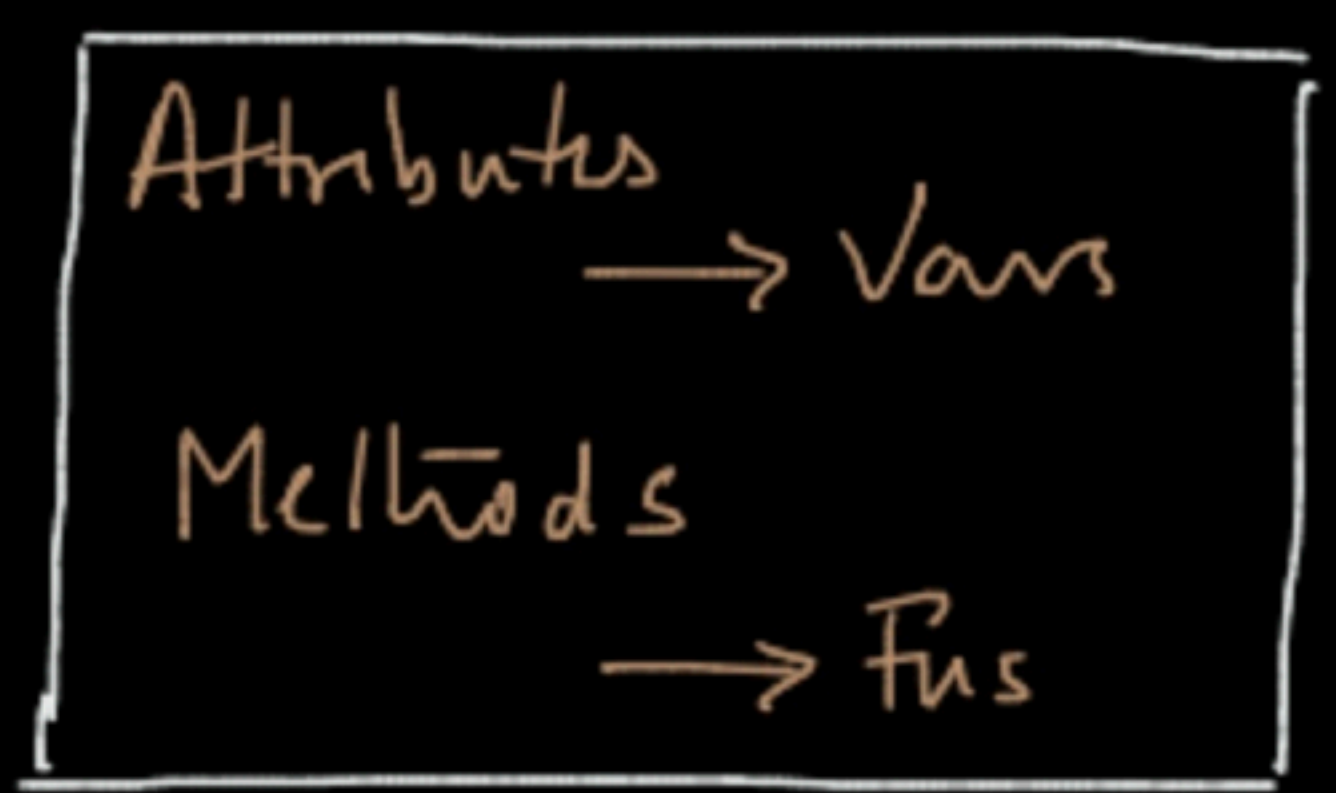
x = 58
P1 = 3.142

Functions ✓

myfn(x,y) → val

→ combination of vars & fns

MyClass → Title Case



Class Person ✓

→ datatype
→ Template

gender ✓
dob ✓
Name ✓
Bn ✓

} Attributes

Methods

talk ()
walk ()
laugh ()
sing ()

} Functionality

object
↑

type
↗

P1 = Person('M', 'Jan', '7m', 0+ve)

P2 = Person('F', 'Mar', '10y', 1+ve)

type(P1) → Person

type(P2) → Person

P2.sing ()

P1.talk ()

P1.dance () X

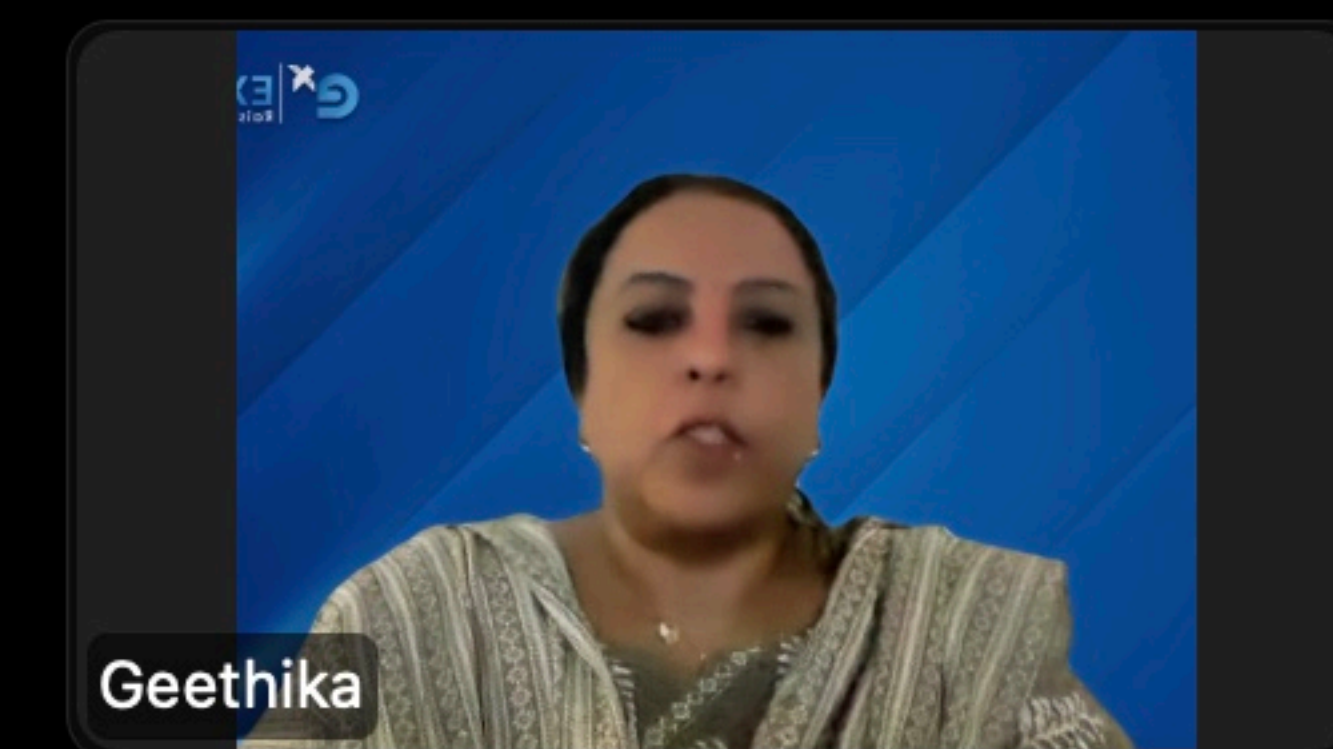
type(l1) ← l1 = [1,2,3] ✓

l2 = [7,8,9,10] ✓

list → append ()
→ extend ()
} lists

t1 = (1,2,3) tuple →

l1.append ()



Fns → standalone `avg()`
 `grades()`

Methods → tied to an object

\curvearrowright
Pl. talk()

Pandas → Package

Variable ✓

Functions ✓

Classes ✓ → Series (1d)
 → DataFrame (2d)

2BHK →

3BHK →

$\left. \begin{array}{l} \text{Bed} = 3 \\ \text{Hall} = 1 \\ \text{K} = 1 \\ \text{Bath} = 3 \end{array} \right\} -$

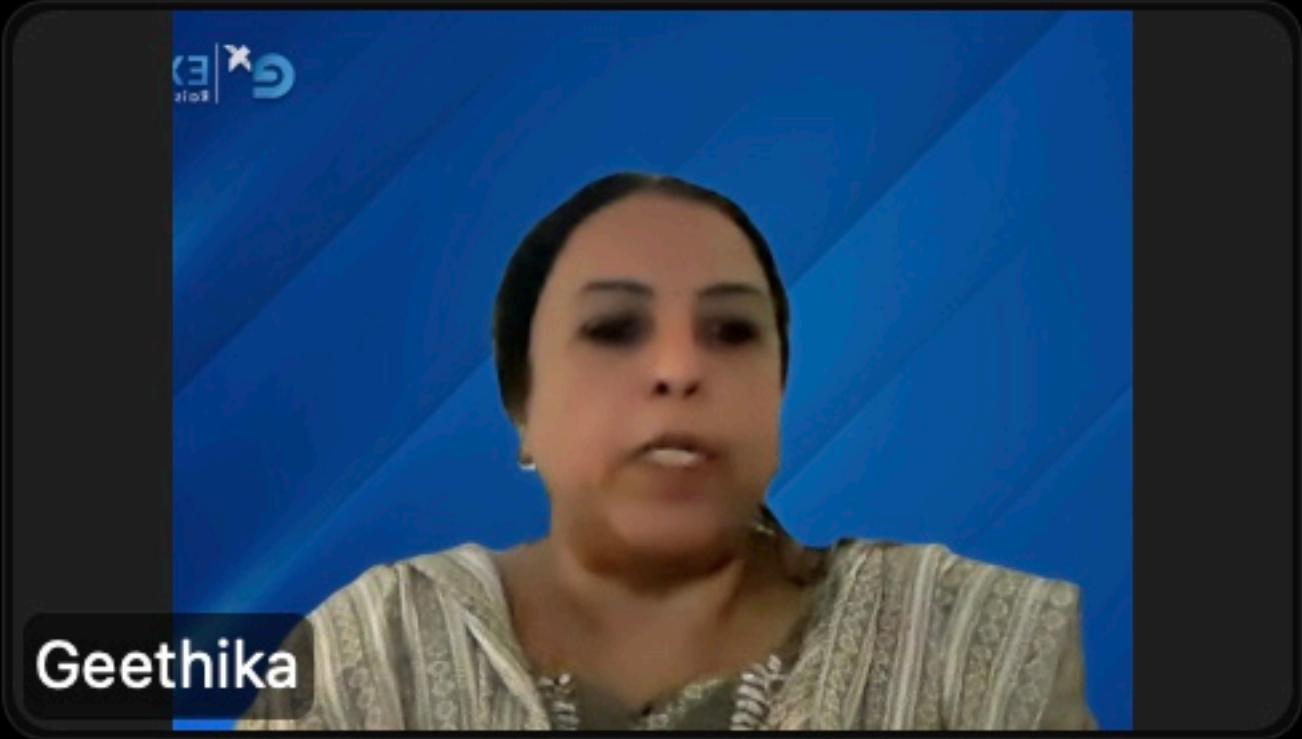
Villa → ✓

`h1 = Villa()`

`h2 = 3BHK()` ✓

`h3 = 3BHK()` ✓

\nearrow
`employee = DataFrame(-- , : ;)`
`student = DataFrame(data - - - -)`
 ✓



Module → math.py

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

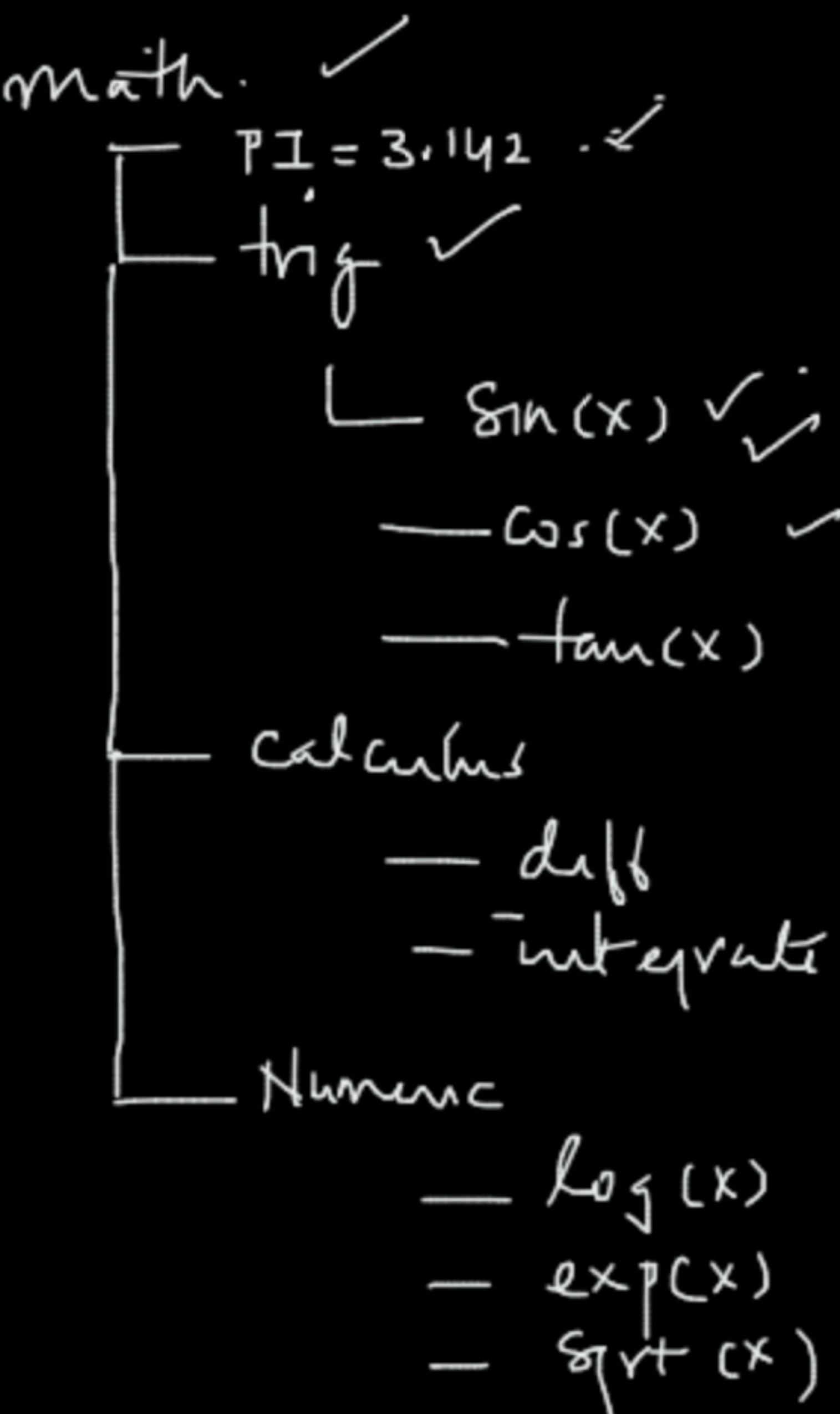
```
def sin(x):
    return
```

```
def exp(x):
    return
```

P1. | Pynb

```
import math
```

$y = \text{math.sin}(x) + \text{math.cos}(x)$
 $y = 5.4 + 6.3$



P1. Pgm

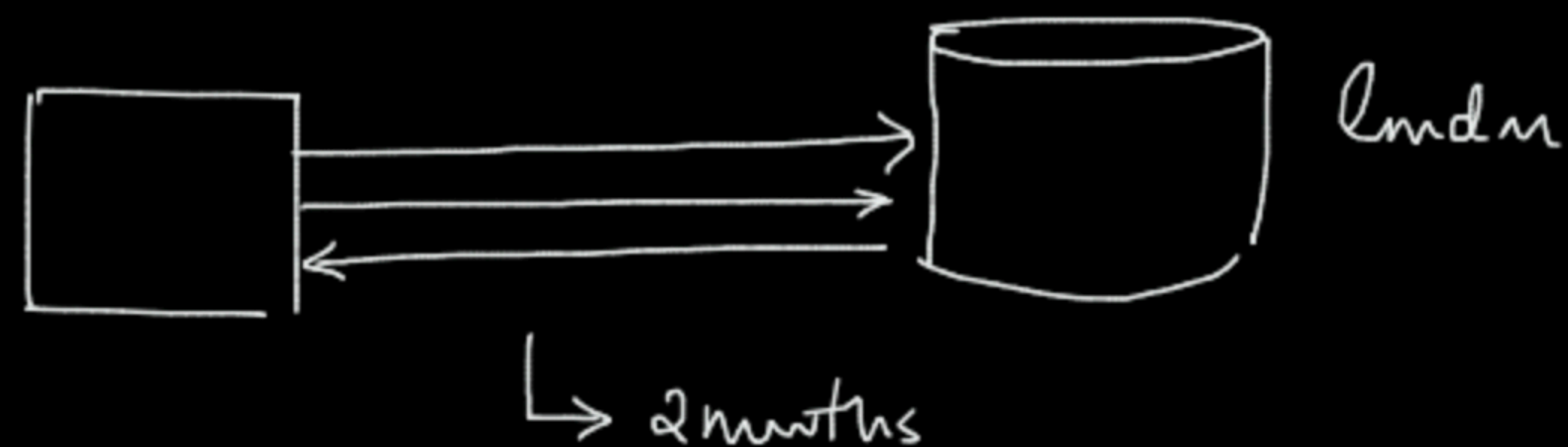
from math import trig

trig.sin(x) + trig.cos(x)

area = math.PI x rad x rad



Pgm →



py mysql → module

✓ Connect() ✓ ~~exit~~
fetch() ✓



import pymysql as ps

^{ps}
~~pymysql~~.Connect()

~~pymysql~~.fetch()
^{ps}



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 = ^{pd.}~~pandas~~.DataFrame(edata, columns=[ed, name, job, sal]) ✓

sdf = ^{pd.}~~pandas~~.DataFrame(— — — —)

edf →

ed	name	job	sal
—	—	—	—
—	—	—	—
—	—	—	—

type(edf) → DataFrame

edf.columns ✓

edf.head() → method

s1 → ^{pd.}~~pandas~~.Series(— — —) obj

s1.info()

sdf →

s1d	s1name	s1job	s1sal
—	—	—	—
—	—	—	—
—	—	—	—

type(sdf) → DataFrame

sdf.columns

sdf.join()

^{pd.}~~pandas~~.read_csv()

pd.~~pandas~~.merge()

		0	1 ✓	2	3	
		eid	ename	job	salary	→ columns
0	e1	100	Thomas	Mgr	3500	
1	e2	200	Mary	Pgm	4500	
2	e3	300	Kate	HR	3900	
3	e4	400	Jim	Mgr	4000	
4	e5	500	Jim	Pgm	7600	
5	e6	600	William	HR	2600	

↓
index.

Access columns →

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

edf.iloc[:, 1]
edf.iloc[:, :3]
edf.iloc[:, [0, 3]]

edf → [row, column]
↓ ↓
1 row 1 col 2
2 range 2 range [eid:job]
3 random 3 random [1, 3, 5]

Accessing Rows →

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

Exception: omit the cols

