08-02-2026

AGENDA

- time module
- fstring
- Object Oriented Programming - Polymorphism
- Module
- Package
- Decorators

$T_1 - T_2 = -30$ min

$T_2 - T_1 = 30$ min

Seconds

121 seconds ⟶

H:M:S

121 / 60 ⟶ 1 min ⟶ 2.xx

121 / 3600 ⟶ hour

60 min

60 x 60 ⟶

OOP (Polymorphism)

```
class TV:
    def start_tv(self):
        logic to start tv
```

```
class Fan:
    def start(self)
        logic to start fan
```

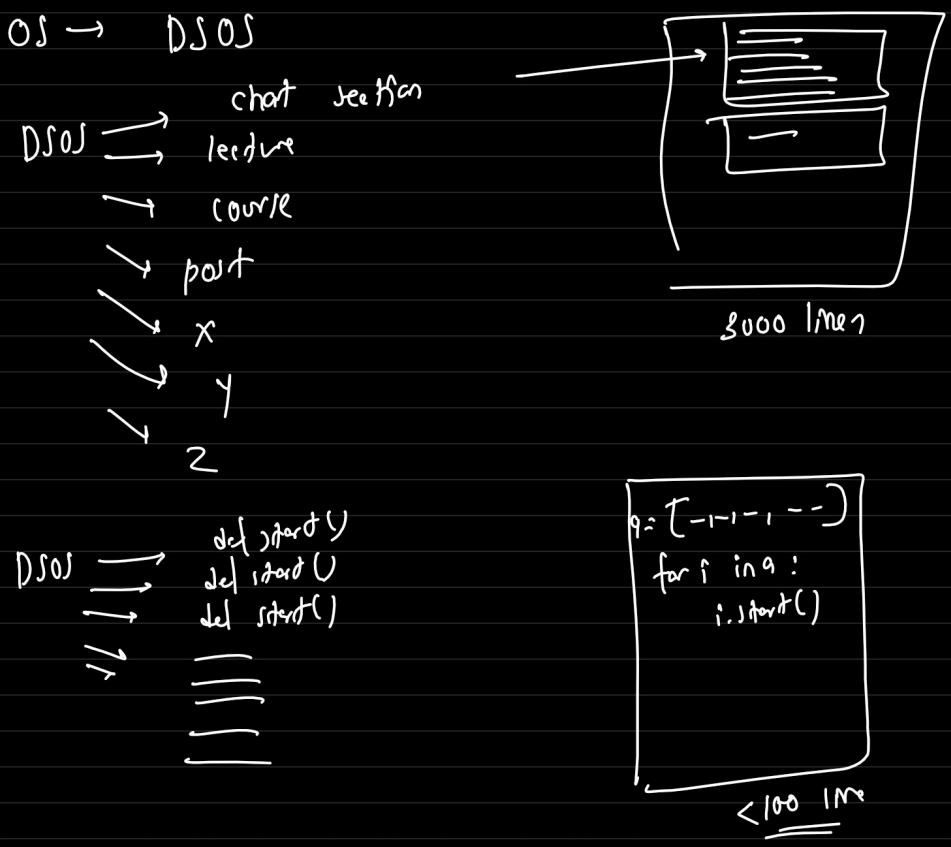
```
class Microwave:
    def start_microwave(self):
        logic to start mv
```

```
tv = TV()
tv.start_tv()

fan = Fan()
fan.start()

mx = Microwave()
mx.start_microwave()
```

start



Module

A python file containing code that can be imported.

9.py

Electronic opptics → 5 classes

mathematical functn (bmi, C→F) → 10 functions

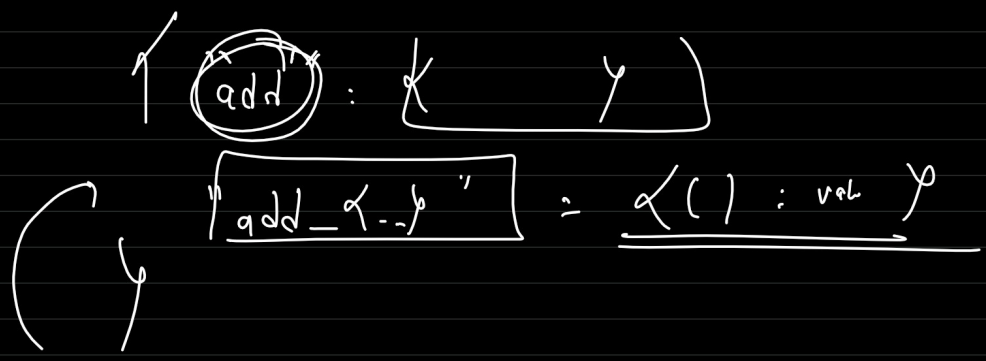
math.py → module → maths functn or classes

utils.py → open text, append text

electronic.py → 5 classes

OS → implemented OS level instructions } module

time → time(), perf_()



Modules → .py → code

to seperate complex & reusable code from the main python file.

calculator.py
temp.py

calculator.py

```
import time

def hello_world():
    print("hello world")

class Calulcator:
    def __init__(self):
        self.__history = []

    def add(self, *args):
        total = 0
        for i in args:
            total+=i

        t = time.time()

        self.__history.append({f"Add_{t}" :
{str(args) : total}})
        return total

    def get_history(self):
        print(self.__history)
```

a.py

temp.py

import os → link → os.py

os.getcwd()

c = calculator.Calculator()

c.add(1,2)

c.gethistory()

Package (folder of modules)

my_utils

Calculator.py
string.py
math.py
} 3 modules

temp.py

import calculator
import string
import math
} 3 modules

ML →
→ a.py → ML-1-2-3
→ b.py → utils

project (name)

step 1) create a .venv

project_name
.venv
package
module
main.py → flow

module → .py
package → collection of similar modules

< Python 3.3+ > __init__.py

package_name
→ __init__.py

modules — .py

package — folder
- .py
- .py
- .py

library — folder-1
- folder-2
- folder-3

PYPI — pip

File — Folder of file — Bundles of package

UDS-B2

100 students

1-33 → Task → Attendance system → AttSys
- .py
- .py
- .py

34-66 → Task → creating notes summary system → NSS
- .py
- .py
- .py

67-100 → Task → chatbot → cbot
- .py
- .py
- .py

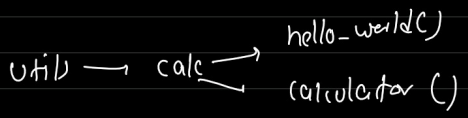
To manage Data Science class

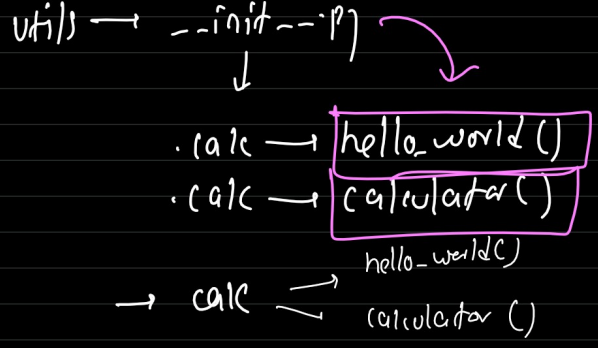
KNDS2 → library → PYPI → other

AttSys → a.py

NSS

cbot





→ calc → hello_world()
calculator()

temp.py

```
from utils import calc
```

```
from utils import calc
from utils import hello_world()
```

calc.py → 100 class
→ 40 funct

temp.py → calculator()
hello_world()

import * → 140

import calc()
import hello_world() → 2

import calc(), hello_world()

Assignment :

Create a Payment Processing System for a store and follow OOP.

- The system should enable store to add cash(online) in their system using CreditCard or UPI

- The system should also have functionality to refund to customers

- The system should have functionality to show balance

- The system should have functionality to check history

Terminal : uv run python main.py
Welcome to Payment system:
Press 1 for Add cash to the system
Press 2 to refund the money
Press 3 check balance
Press 4 to check the transcation
Press 5 to exit the system

if user press 1 : Want to use credit card or UPI : 1 of C, 2 UPI
if user press 2 : enter amount to refund : subtract the amount from system credit
if user press 3 :
if user press 4 : show the history but with good format in terminal

The code should be robust, running in while loop, should handle user input and type casting, should have exception handling + All OOPS concept + modules & package creation is (optional)