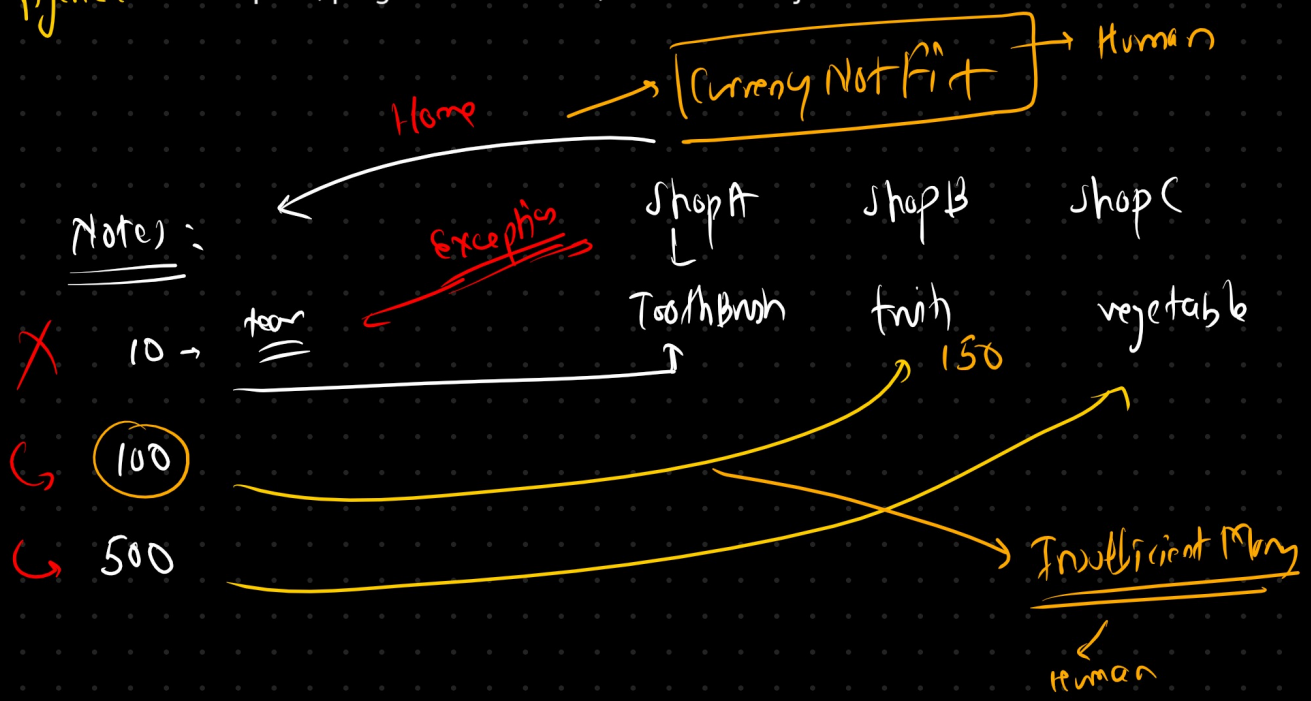
26-04-2025

Agenda – Exception, program termination , Classes and objects

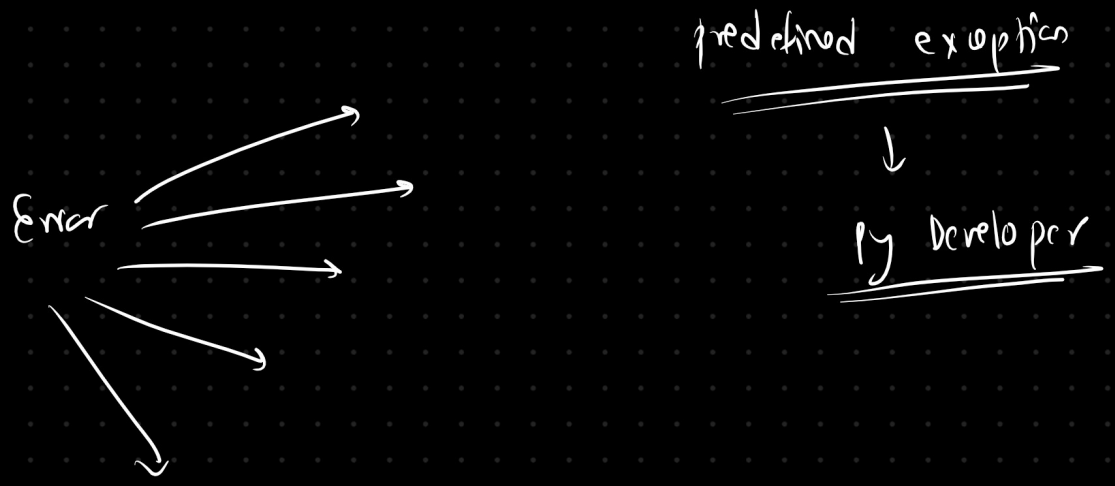


Exception Handling

Try, Except, else, finally

try :

except :



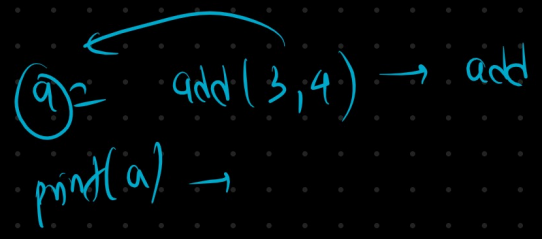
try-Except

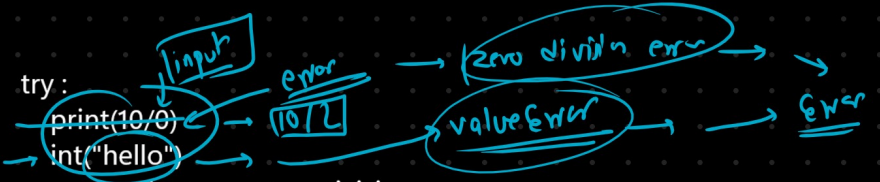
try:
    Usual code
except:
    print("Hello")

```
try :
    for i in range(5):
        a= input("Enter a number")
        a = int(a)
        print(a)
except ValueError:
    print("Issue with the program")

print("Hello world")
```

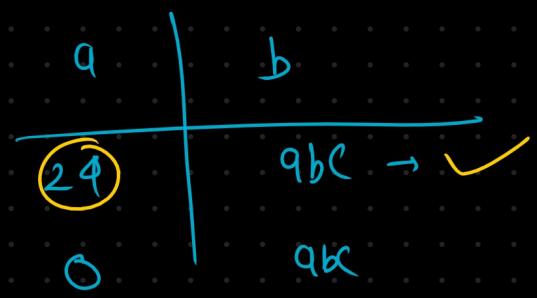


```
def add ( a,b ):
    c = a+b
    return c
```

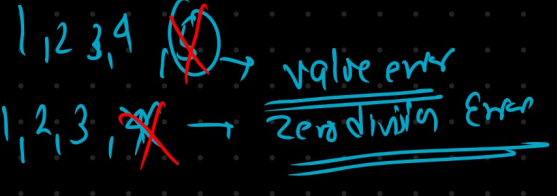


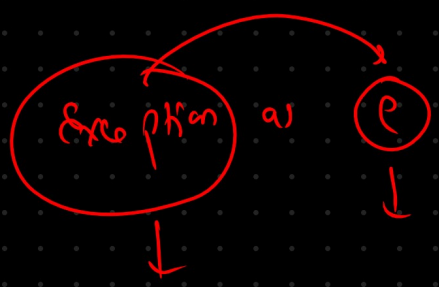


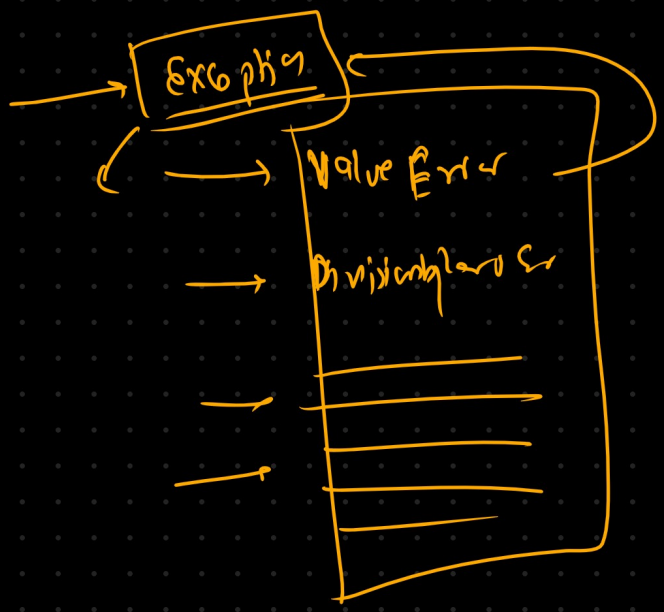

```
try :
    print(10/0)
    int("hello")
except (ValueError, ZeroDivisionError):
    print("Issue with the program")

print("Hello world")
```

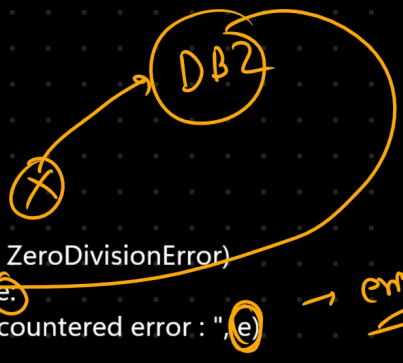



```
try :
    a = input("Enter a value : ")
    a = int(a)
    b = input("Enter a value to convert: ")
    print(10/a)
    int(b)
except (ValueError):
    print("Issue with the program")

print("Hello world")
```










```
try :
    a = 0
    a = int(a)
    b = "abc"
    print(10/a)
    int(b)
# except (ValueError, ZeroDivisionError)
except Exception as e:
    print("We have encountered error : ",e)

print("Hello world")
```

Block used to handle exception → try:

→ except Exception → parent of all exceptions

```
# input by user
try :
    a = 0
    a = int(a)
    b = "1"
    print(10/a)
    int(b)
except Exception as e:
    print("We have encountered error : ", e)
finally:
    print("Input section executed successfully")
```

try :
except :
finally

error
no-error
→ always
→ we dont do calc → cleanup

try: f.open()
read from .txt
open
except:
finally
f.close() → cleanup & print

Python always shows → where is error
→ try & except
→ not breaks the code
. error.

malva

harbs

data ← database.txt

xy ← abc.txt

music ← mp3.txt

mobile

static → a = (database.txt, abc.txt, music.txt)

```
def loaddata(a):
→ try: for i in a:
        f = open(i)
        f.readlines()
→ except
```

file not fou

→ momay

→ rename

→ deleted

load → run

exit

crash

No Yes

full

try:
except:

→ try: [ ] → no error

except:

→ else: → runs only when there is no error in try block

finally: → runs always

try

exception Exception: → 10/0 → ZDE

abc → int → value

cal

pre-defined error

user error

age < 0 or age > 100

age → to provid

raise ValueError("p

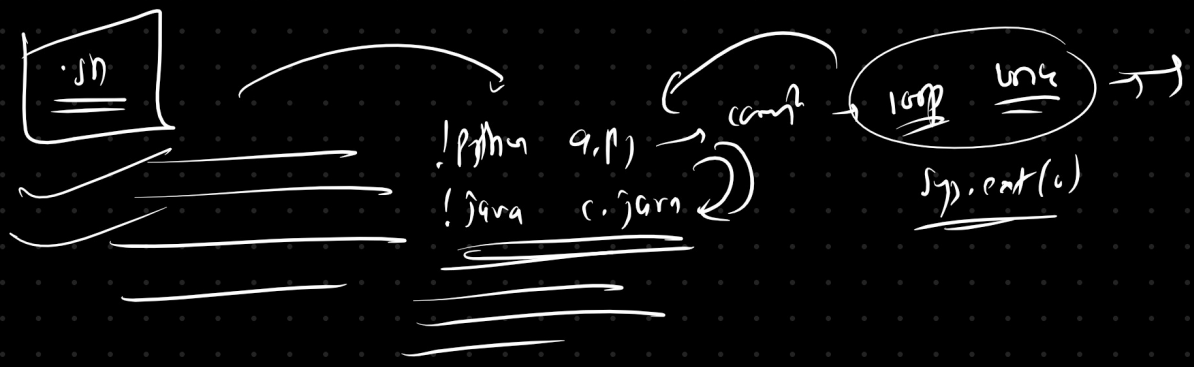
Python → 2

.py

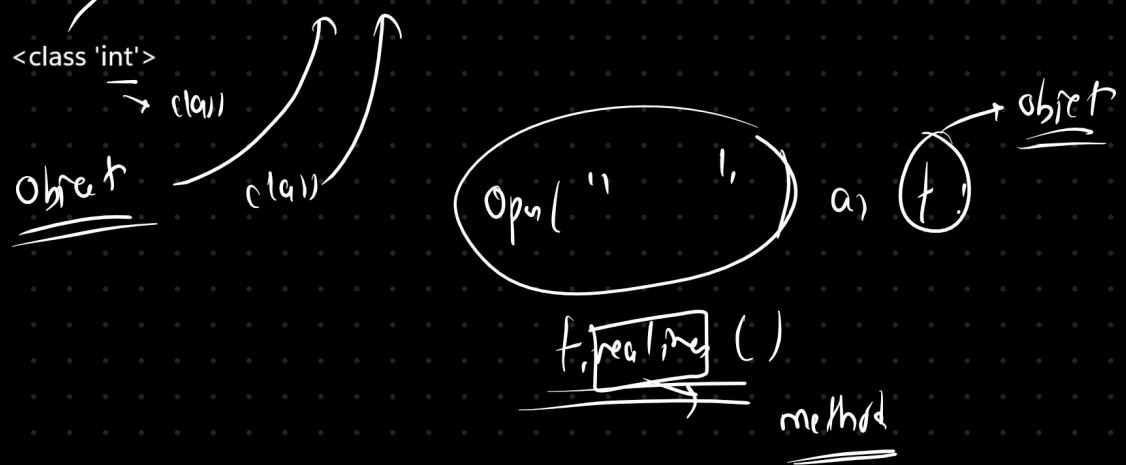
.ipynb → notebook

Python3 → python3

Python

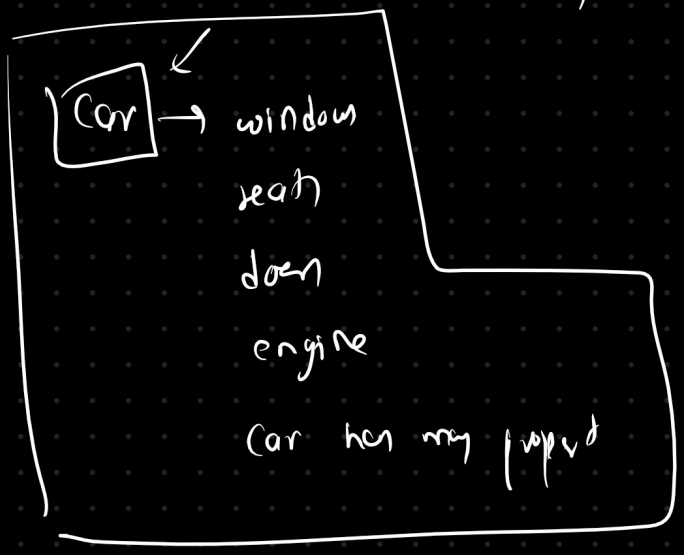


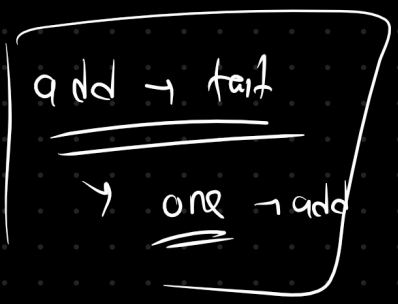
OOP → object oriented programming



def add() → standard

add()



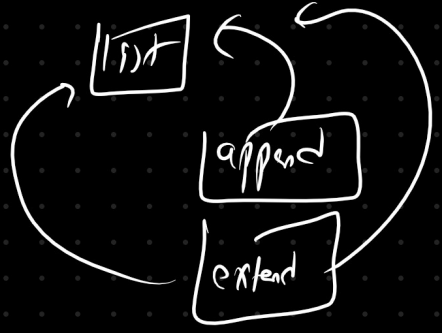


a = 3
a = "abc"

a = ( )
→ limit ?
→ complex

list.append() — 2+ or 1

list.extend()
→ (1a))

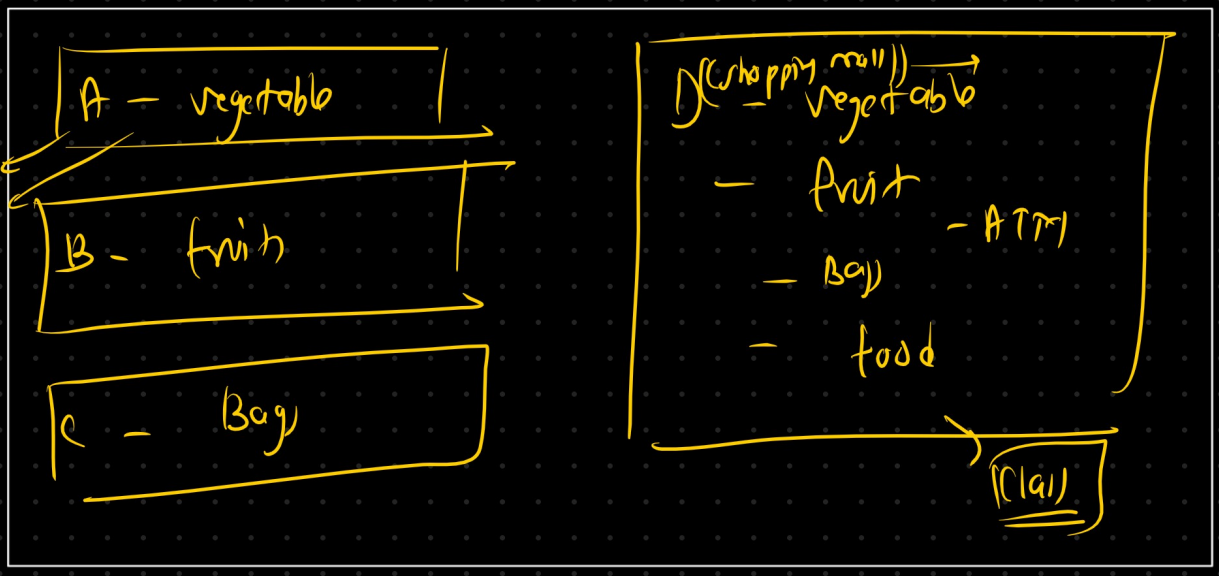


C(a)) → collection of function

function → method

add() → function

a.add()
→ method

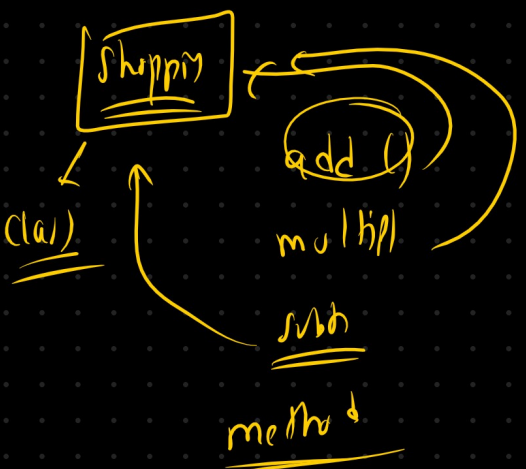


def add ()

→ standard.

↓

functions



method

Function inside a (lass) is called a method