

* Local and global variables:-

Local variables : [LV]

- Variables that created (=) inside a fn are called as LV.
- LV can be used only inside that fn in which they are created.

Saturday

DATE 18 06 2022

P271.py

```
def square():
    x = int(input('x :')) # x & s are Lv of
                           # square fn
    s = x * * 2
    print(f'square : {s}')
# end : square

def cube():
    y = int(input('y :')) # y & c are Lv of cube
                           # fn
    c = y * * 3
    print(f'cube : {c}')
# end : cube

def main():
    square()
    cube()
# end : main
```

main()

* Global variables :- [GV]

① Variables that are created outside of all fns
are called as global variables.

② All fns of that pgm can use GV and can modify
contents of GV by specifying following syntax:

```
global var1, var2, ..., varN
```

classmate

PAGE

--	--	--

P272.PY

```

def india():
    global city
    city = 'New Delhi'
# end: india

def japan():
    global city
    city = 'Tokyo'
# end: japan

def main():
    global city
    india()
    print(f'Capital of India: {city}')
    japan()
    print(f'Capital of Japan : {city}')
# end: main

```

city = None
main()

o/p

Capital of India: New Delhi
Capital of Japan: Tokyo

india

city

[None]

[None] -> japan

city

main

city

None

'New Delhi'

'Tokyo'

P273.py

```

def square():
    global a
    for i in range(len(a)): # 0 1 2
        a[i] = a[i] ** 2
    # end: for
# end: square
def main():
    global a
    print(a)
    square()
    print(a)
# end: main

```

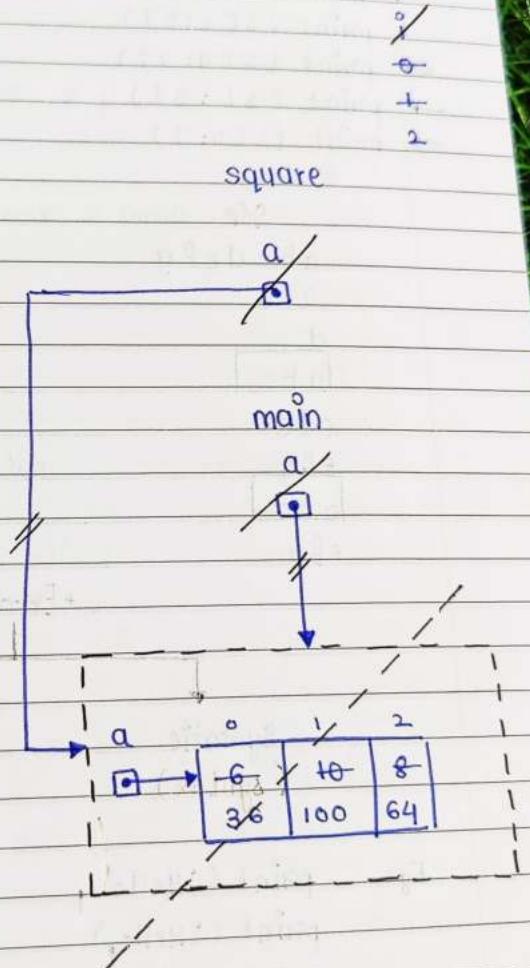
a = [6, 10, 8]

main()

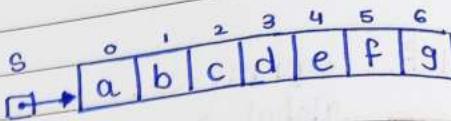
O/P

[6, 10, 8]

[36, 100, 64]



P274.PY



```

s = 'abcdefg'
print(s)
print(s[0])
print(s[3])
→ print(s[0:3]) # [si: ej]
print(s[2:5])
⇒ print(s[4:7])
→ print(s[:3])
⇒ print(s[4:])
    
```

o/p

abcdefg

q

d

abc

cde

efg

abc

efg

* Errors *

Syntactic

(syntax)

Exceptions

(Data / Info.)

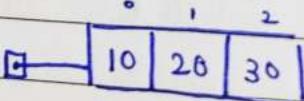
Eg:

```

print('Hello')
print('Hello')
z = x > y
      ↑
    
```

```

z = 10/0
l = [10, 20, 30]
      ↑
    
```



print(l[5])

Exception Handling :-

try raise except Python
↓ ↓ ↓
try. throw catch Java

Syntax :

try :

if condition :

* raise Error # throw

end: if

end: try

except Error : # catch

end: except

next st

if condition is False.

DATE

```
try:  
    if condition:  
        raise Error # throw until  
    # end: if  
  
# end: try  
except Error: # catch  
    #end: except  
  
next_st
```

P275.py

```
def main():  
    try:  
        age = int(input('enter age:'))  
        if age <= 0:  
            raise ValueError # throw  
        #end: if  
        print(f'you are {age} years old')  
    except ValueError: # catch  
        print(f'{age} is invalid\n'  
              f'age cannot be -ve')  
    # except  
# end: main
```

main()

Shot on OnePlus

By javed kadgaokar

11/08/22

11:07 am

o/p

```
age  
20  
enter age  
11 20
```

P276.f

def r

o/p

age

20

enter age : 20 ↴

U r 20 years old

age

-10

enter age : -10 ↴

-10 is invalid

age cannot be -ve.

P276.py

def main ():

try:

a = [10, 20, 30]

print ('before del: ', a)

i = int (input ('enter index [0..2]: '))

x = a.pop (i) # throw

print ('{} deleted from list'.format (x))

print ('after del: ', a)

except IndexError:

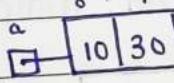
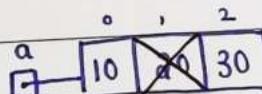
catch

print ('{} index does not exist'.format (i))

print ('plz i/p index between 0..2')

end: except

end: main



main()

i x
1 20

o/p

before del: [10, 20, 30]

enter index [0..2]: 1 ↴

20 deleted from list

after del: [10, 30]

a	0	1	2
i	10	20	30

i
12

before del : [10, 20, 30]
 enter index [0..2] : 12 ↴
 12 index does not exist
 plz i/p index between 0..2

[P2T 7 PY]

def main():

try:

x = int(input('enter x : '))

y = int(input('enter y : '))

z = x/y

print(f'quo: {z}')

except ZeroDivisionError: # catch ↴

print(f'Div by zero not allowed')

print(f'Plz i/p non-zero value for y')

end - except

end: main

[main()]

x

20

y

5

z

4.0

enter x : 20 ↴

enter y : 5 ↴

quo : 4.0

x	y
10	0

enter x: 10 ↴
 enter y: 0 ↴
 Div by zero not allowed
 Plz i/p non-zero value for y.

P2T8.py

```
def main():
    try:
        s = input('enter a no.: ')
        x = int(s)
        c = x ** 3
        print(f'cube : {c}')
    except ValueError:
        print(f'{s} is invalid no.\n'
              f'Plz i/p valid no.')
    # end: except
# end: main
```

main()

s	x	c
'10'	10	1000

enter a no.: 10 ↴
 cube : 1000

s
'4I'

enter a no.: 4T ↴
 4T is invalid no.
 Plz i/p a valid no.

[P279.py]

def main():

try:

l = ['raj', 'navi', 'amit']

print ('before del : ', l)

n = input ('enter name : ')

l.remove (n)

print ('after del : ', l)

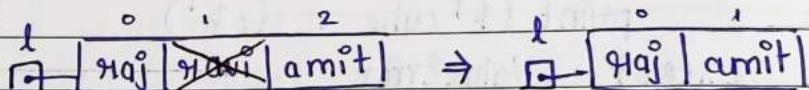
except ValueError:

print ('{} does not exist'.format(n))

#end: except

end: main

main()



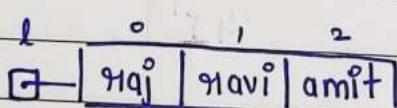
n
'navi'

/p

before del : ['raj', 'navi', 'amit']

enter name : navi ↴

after del : ['raj', 'amit']



n
'jim'

/p

before del : ['raj', 'navi', 'amit']

enter name : jim ↴

jim does not exist

single try and multiple except(catch) :

[p280.py]

def main:

try:

s = input('enter x : ')

x = int(s)

s = input('enter y : ')

y = int(s)

z = x/y

print(f' quo : {z}')

except ValueError:

print(f'{s} is invalid no.')

except ZeroDivisionError:

print(f'Plz i/p non-zero value for y')

end: try-except

end: main

[main ()]

s	x	y	z
'20'	20	5	4
'5'			

enter x : 20 ↴

enter y : 5 ↴

quo : 4

s
'2E'

x

enter x : 2E ↴
2E is invalid no.

s x y
'20' 20
5x

enter x : 20 ↴
enter y : 5x ↴
5x is invalid no.

s x y
'20' 20 0
'0'

enter x : 20 ↴
enter y : 0 ↴
Plz i/p non-zero value for y.

↓ # Nested try / except

def main ():

try:

s = input ('enter x : ')

x = int(s)

s = input ('enter y : ')

y = int(s)

try:

z = x / y

print (f'qno : {z}')

```

except ZeroDivisionError:
    print('Plz g/p non-zero value for y')
# end: inner try-except
except ValueError:
    print(f'{s} is invalid no.')
# end: outer try-except
# end: main

```

main()

S	x	y	z
'20'	20	5	4

'5'

enter x: 20 ↴

enter y: 5 ↴

quo: 4

S	x
---	---

'2E'

enter x: 2E ↴

2E is invalid no.

S	x	y
---	---	---

'20'

'5x'

enter x: 20 ↴

enter y: 5x ↴

5x is invalid no.

S	x	y
'20'	20	0
'0'		

enter x : 20 ↴
 enter y : 0 ↴
 Plz i/p non-zero value for y.

1 # try / except / finally

P282.py

```
def main():
    try:
        print('A')
        x = int(input('enter x: '))
        if x == 1: return
        if x == 0: z = 10/x
        print('B')
    except Exception:
        print('C')
    finally:
        print('D')
# end: main
```

main()

x	o/p
2	A
	enter x: 2 ↴
	0

x
1

o/p

A

enter x: 1 ↴
 Bx
0

o/p

A

enter x: 0 ↴
 C
 Dx
5E

o/p

ea A

enter x: 5E ↴

 C
 D

P283.py

```
t →
def series():
    yield 1
    yield 2
    yield 3
    # raise StopIteration
# end: series
```

```
def main():
    t = series()
    try:
        print(next(t))
        print(next(t))
```

```

print(next(t))
print(next(t))
except StopIteration:
    print('series ends')
#end: main

```

main()

o/p

1
2
3

series ends.

p284.py

```

def series():
    yield 1
    yield 2
    yield 3
    # raise StopIteration
#end: series

```

```

def main():
    t = series()
    for i in t: # i = next(t)
        print(i)
#end: for
#end: main

```

main()

```
for i in series():
    print(i)
#end: for
```

i	%/p
1	1
2	2
3	3

P285.py

```
def myrange(sv, ev):
    while sv < ev:
        yield sv
        sv += 1
    # end: while
    # raise StopIteration
# end: myrange
```

```
def main():
    t = myrange(5, 8)
    for i in t: # i = next(t)
        print(i)
    # end: for
# end: main
```

main()

%/p
5
6
7

```
for i in myrange(5, 8):
    print(i)
# end: for
```

P286.PY

```
def myrange(sv, ev):
    while sv < ev:
        yield sv
        sv += 1
    # end: while
    # raise StopIteration
# end: myrange
```

sv ev
 ⑤ 8
 ⑥
 ⑦
 8

```
def main():
    t = myrange(5, 8)
    try:
        while True:
            print(next(t))
        # end: while
    except StopIteration:
        pass
    # end: except
# end: main
```

main()

%p

5

6

7

P287.PY

```

def friends():
    for x in ['raj', 'ravi', 'amit']:
        yield x
    # end: for
    # raise StopIteration
# end: friends

```

```
def main():
```

```
    try:
```

```
        t = friends()
```

```
        print(next(t))
```

```
        print(next(t))
```

```
        print(next(t))
```

```
        print(next(t))
```

```
    except StopIteration:
```

```
        pass
```

```
# end: TF
```

```
# end: main
```

```
main()
```

%p
raj
ravi
amit

* Comprehension

✓ List

✓ Set

✓ Dictionary

✗ Tuple

Shot on OnePlus

By javed kadgaokar
CLASSMATE

PAGE

11/08/22

11:08 am

P288.py

```

"""
def friends():
    for x in ['raj', 'ravi', 'amit']:
        yield x
    # end: for
# end: friends
"""

def main():
    # t = friends()
    t = (x for x in ['raj', 'ravi', 'amit'])
    # Tuple Comprehension ✓ Generator
    try:
        print(next(t))
        print(next(t))
        print(next(t))
        print(next(t))
    except StopIteration:
        pass
    # end: TE
# end: main

```

main()

%p
raj
ravi
amit.

P289.py

```

def square():
    for x in range(10,13):
        yield x ** 2
    # end: for
    # StopIteration (raise)
# end: square

```

Shot on OnePlus

By javed kadgaokar

classmate

11/08/22

11:08 am

```

def main():
    t = square()
    try:
        print(next(t))
        print(next(t))
        print(next(t))
        print(next(t))
    except StopIteration:
        pass
# end: TF
# end: main

```

o/p
100
121
144

main()

p290.py

```

def square():
    for x in range(10, 13): # 10 11 12
        yield x ** 2
# end: for
# stopIteration (raise)
# end: square
***
```

def main():

```

# t = square()
t = (x ** 2 for x in range(10, 13))
```

Generator Comprehension

try:

```

print(next(t))
print(next(t))
print(next(t))
print(next(t))
```

DATE

```
except StopIteration:  
    pass  
# end: TE  
# end: main
```

o/p

100

121

144

main()

* Default Parameters:

(Object Oriented Features :-)

→ def function_name (parameter = value):

default
param

default
value

p291.py

```
def greet (msg = 'Hi'):  
    print ('{} {} Raj'.format(msg))
```

greet

end: greet

x | msg
'Hello'

def main ():

(simple)

greet ('Hello')

x | msg
'Hi'

greet ()

end: main

main()

main

o/p

Hello Raj
Hi Raj

P292.py

```

def greet (msg = 'Hi', name = 'Raj'):
    print (msg, name)
# end: greet
def main():
    greet ('Hello', 'Amit')
    greet ('Hello')
    greet ()
    # greet (, 'Amit') # error
    greet (name = 'Amit')
# end: main

```

main()

o/p

greet

x	msg	name
	'Hello'	'Amit'

Hello Amit

Hello Raj

Hi Raj

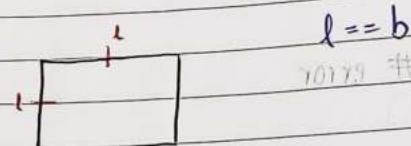
Hi Amit.

x	msg	name
	'Hello'	'Raj'

x	msg	name
	'Hi'	'Raj'

x	msg	name
	'Hi'	'Amit'

main

◎ $a = l \cdot b$ ◎ $a = l \cdot l$ 

P293-PY

```

def area(l, b = None):
    if b is None: # b == None
        print(f"area of square {l*l}")
    else:
        print(f"area of rect {l*b}")
    # end: if-else
# end: area
def main():
    area(20, 8) # rect
    area(10) # sq
# end: main

```

main()

area

main

x	l	b
	20	8

x	l	b
	10	None

%p
area of rect 160
area of square 100

* Rule :-

A parameter can have default value only when all params on R.H.S of its have default values.

P294.py

```
"""
def t1(a = 10, b): pass # error
"""

"""
def t2 (a= 10, b= 20, c) : pass # error
"""

"""
def t3 (a= 10, b ,c = 20): pass # error
"""

"""
def t4 (a , b= 10): pass

def t5 (a , b = 10 , c = 20): pass

def t6 (a , b , c = 20): pass

def t7 (a= 10 , b= 20 , c = 30): pass

# No %p is generated.
```

DATE

* class : data
+
function

* class className:
member 1 = value
member 2 = value
:
:
member N = value

#end: className

[P295.py]

```
class Student: # Class Declaration
    name = 'Roj'
    roll = 101
    per = 70.77
#end: student
def main():
    s1 = Student() # Class Definition
    print(s1.name)
    print(s1.roll)
    print(s1.per)
    print(f'Name: {s1.name}')
    print(f'Roll: {s1.roll}')
    print(f'per: {s1.per}')
#end: main
```

main()

S1	name	Raj
	roll	101
	per	70.77

o/p

Raj

101

70.77

Name : Raj

Roll : 101

Per : 70.77

P296.PY

class Staff:

name = "

job = "

end: staff

def main():

S1 = Staff()

S1.name = 'Ravi'

S1.job = 'officer'

print(S1.name, S1.job)

S2 = Staff()

S2.name, S2.job = 'Amit', 'Clerk'

print(S2.name, S2.job)

end: main

SI | name
| job

Ravi

officer

S2 | name
| job

Amit

Clerk

%/P
Ravi officer
Amit Clerk

[P297-PY]

```
class Bike:  
    company = ''  
    cost = 0  
#end: Bike  
def main():  
    b1 = Bike()  
    b1.company = input('enter company: ')  
    b1.cost = int(input('enter cost: '))  
    print(f'Company: {b1.company}\n' +  
          f'cost: {b1.cost} Rs')  
# end: main  
main()
```

b1	company	Yamaha
	cost	90000

enter company : Yamaha ↵

enter cost : 90000 ↵

Company : Yamaha

Cost : 90000 Rs.

P298.py

class Person:

name = 'Amit'

fruits = ['Apple', 'Mango', 'Kiwi']

end : Person

def main():

p1 = Person()

print(f'{p1.name} likes :')

for x in p1.fruits: # Apple Mango Kiwi

print(x)

end : for

end : main

main()

%p

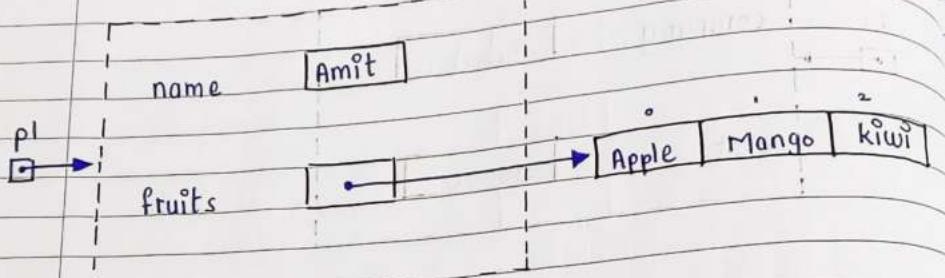
Amit likes :

Apple

Mango

Kiwi

DATE



[P299.py]

```
class Person:  
    name = ""  
    fruits = [None] * 3  
#end: Person  
def main():  
    p1 = Person()  
    p1.name = input('enter name : ')  
    print('enter fruits : ')  
    for i in range(len(p1.fruits)):  
        p1.fruits[i] = input()  
    #end for  
    print(f'{p1.name} likes : ')  
    for x in p1.fruits:  
        print(x)  
#end: for  
#end: main
```

[main()]

name	Raj				
fruits	•	→	Grapes	Apple	Cherry

enter name : Raj ↴

enter fruits :

Grape ↴ ←

Apple ↴ ←

Cherry ↴ ←

Raj likes :

Grapes

Apple

Cherry

me or self

| p300.py

class Demo :

def india(me):

print('Capital of India : New Delhi')

end : india

def japan(me):

print('Capital of Japan : Tokyo')

end : japan

end : Demo

def main():

```

d1 = Demo()
Demo.india(d1)
d1.india()
Demo.japan(d1)
d1.japan()
#end: main

```

main()



%p
Capital of India: New Delhi
Capital of India: New Delhi
Capital of Japan: Tokyo
Capital of Japan: Tokyo

p301.py

7

class Demo:

def square(me, x):

print(f'square: {x ** 2}')

#end: square

def add(me, a, b):

print(f'add: {a+b}')

#end: add

end: Demo.

def main():

d1 = Demo()

d1.square(7)

d1.add(10, 20)

end: main

a / 10 b / 20

main()

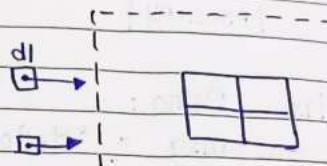
square
add:

* Access

① prev

① p

%
square : 49
add : 30



* Access Specifiers: private public protected (Inheritance)

- ① private (variable): It can be used only inside the class and it cannot be used outside the class.
- ② public (variable): It can be used inside as well as outside the class.

Program

Class

~~private~~

✓ private

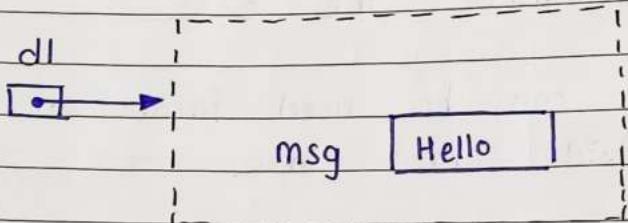
✓ public

✓ public

P302.PY

```
class Demo:  
    msg = "Hello" # public  
# end: Demo  
def main():  
    d1 = Demo()  
    print(d1.msg)  
# end: main
```

main()

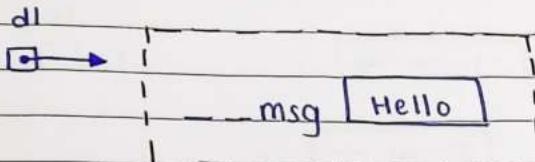


P303.PY

P283

```
class Demo:  
    msg = "Hello" # private  
# end: Demo  
def main():  
    d1 = Demo()  
    ""  
    print(d1.msg) # error  
    ""  
# end: main
```

main()



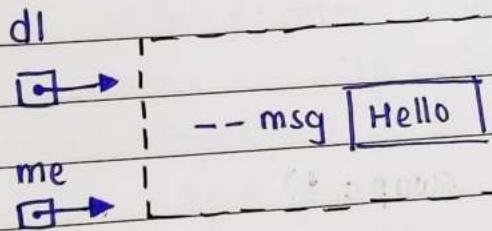
p304.py

```

class Demo:
    _msg = 'Hello'
    def output(me):
        print(me._msg)
# end: output
# end: Demo
def main():
    dl = Demo()
    dl.output()
# end: main

```

main()

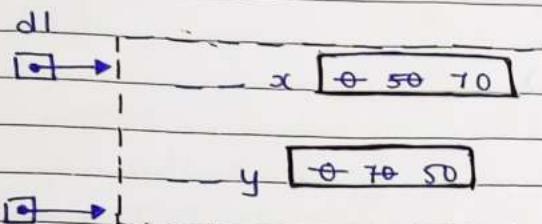


- ① private data of a class can be accessed (used) only through fn / methods of that class.

p304.py

```
class Data:  
    x = 0  
    y = 0  
    def input(me):  
        print('enter 2 no.s :')  
        me.x = int(input())  
        me.y = int(input())  
    #end: input  
    def output(me):  
        print(f'x : {me.x}\t'  
              f'y : {me.y}')  
    #end: output  
    def swap(me):  
        t = me.x  
        me.x = me.y  
        me.y = t  
    #end: swap  
#end: Data  
def main():  
    d1 = Data()  
    d1.input()  
    print('before swap :')  
    d1.output()  
    d1.swap()  
    print('after swap :')  
    d1.output()  
#end: main
```

Main()



enter 2 nos :

50 ↴

70 ↴

before swap :

x : 50 y : 70

after swap :

x : 70 y : 50

P306.py

class Square :

 __side = 0

 __area = 0

 def input(me) :

 print('enter side : ', end='')

 me.__side = int(input())

 # end: input

 def output(me) :

 print(f' side : {me.__side} \t')

 f' Area : {me.__area}')

 # end: output

end: square

def main() :

 s1 = Square()

 s1.input()

 s1.output()

 s2 = Square()

Shot on OnePlus

By javed kadgaokar

classmate

s2.input()
s2.output()
end: main

[main()]

s1 | side θ 8

| area θ 64

s2 | side θ 10

| area θ 100

enter side:

8 ↴

side : 8

Area : 64

enter side:

10 ↴

Side : 10

Area : 100

- ① input and output fn are member fn / method of class square.

P307-PY

```

class Square:
    side = 0
    area = 0
    def input(me):
        print('enter side : ', end=' ')
        me.side = int(input())
        me.area = me.side * * 2
    # end: input
    def output(me):
        print(f' side : {me.side} \t'
              f' Area : {me.area}')
    # end: output
# end: Square
def main():
    s1 = Square()
    s2 = square()
    t = s1
    t.input()
    s1.output()
    t = s2
    s2.input()
    t.output()
# end: main

```

main()

```

enter side : 10 ↴
side : 10   Area : 100
enter side : 7 ↴
side : 7   Area : 49

```

DATE

