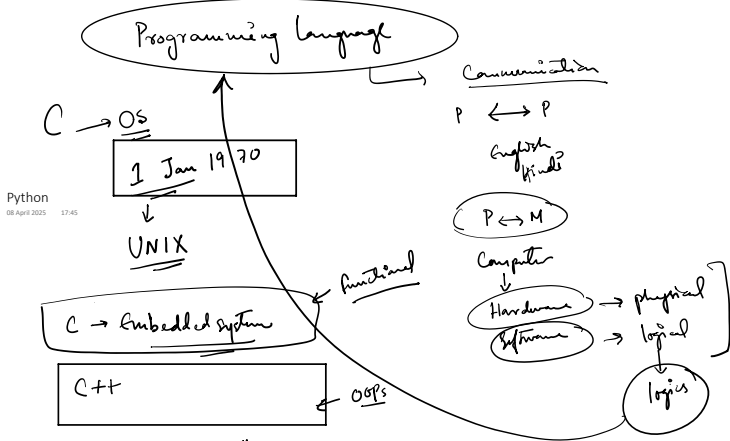


Introduction to Python :-



Python
08 April 2025 17:45

Java -> Fullstack

JS -> MERN

Python -> AI, ML, NN, DS, DL, NLP, DE, Web dev, Django, Flask, Tkinter, game, DAO, Kivy

History
Guido Van Rossum

late 1980's
1991
1994 v1
2000 v2
2008 v3

Why Python is called Python?

BBC comedy series

Monty Python's Flying Circus

Features ->

Easy to implement

Dynamic

Typed
Memory allocation

int a=45;

a=45

45

C language
#include <stdio.h>
#include <conio.h>
void main() {
clrscr();
printf("Hello World!");
getch();
}

sys. stdio
print("Hello world!")
main

Open Source :-

Source code -> Developer
Object code -> Executable
Windows -> Cloud source

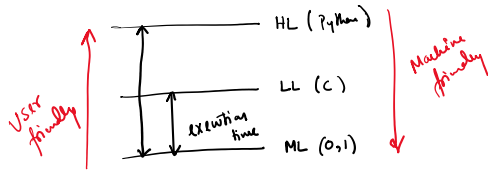
Linux
Android

Kernel
Shell

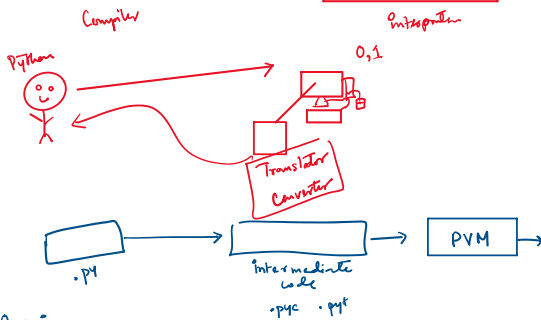
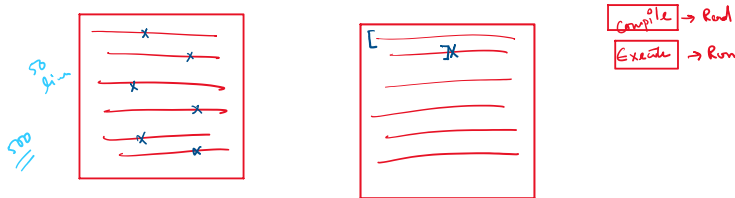
Windows
Mac OS
Unix
iOS
MIUI
One UI
Emui
Huawei

100
MUI
One V1
Exm?
function
class
type of
Python
BASIC
BASIC
BASIC

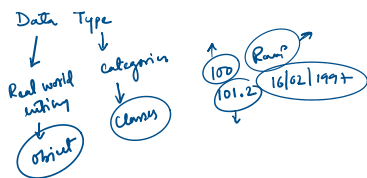
High level



Interpreted →



Ops =



a = 45



Human

functions = read(), write(), delete(),
edit(), think(), code(),
sleep() - - -

Attributes = name, age, designation



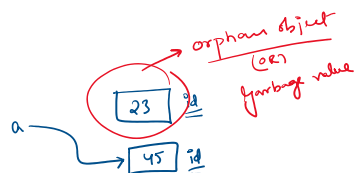
Data Types

a = 23

a = 45

a = 1

b = 1



Strings =

s = "Hello world!"



- Data Structure
(Linear)
- Similar data type

c = "Hello world!"

S = Hello World:

— Similar data type

0	1	2	3	4	5	6	7	8	9	10	11
H	E	L	L	O		W	O	R	L	D	!
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

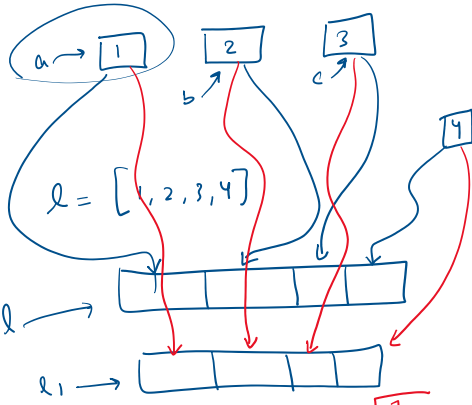
S = "Hello World!"

s[start: stop : step]
direction

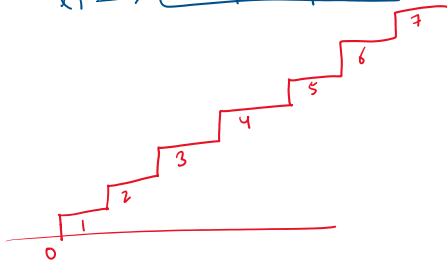
```
print(s[2::5]) #Lo, ew!
print(s[:-5:2]) #LL, (!)
print(repr(s[:4:-3])) #!r_, !r L
print(repr(s[-3:4:2])) #Lo L,
```



a = 1
b = 2
c = 3

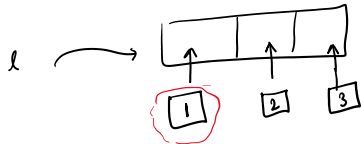


Step = 2

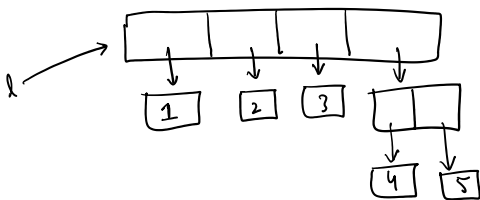


4 list

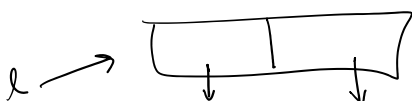
l = [1, 2, 3]

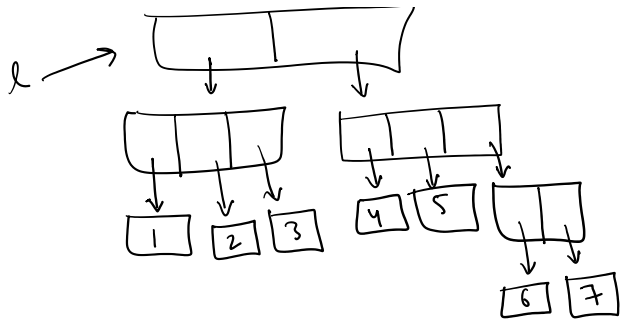


l = [1, 2, 3, [4, 5]]



l = [[1, 2, 3], [4, 5, [6, 7]]]

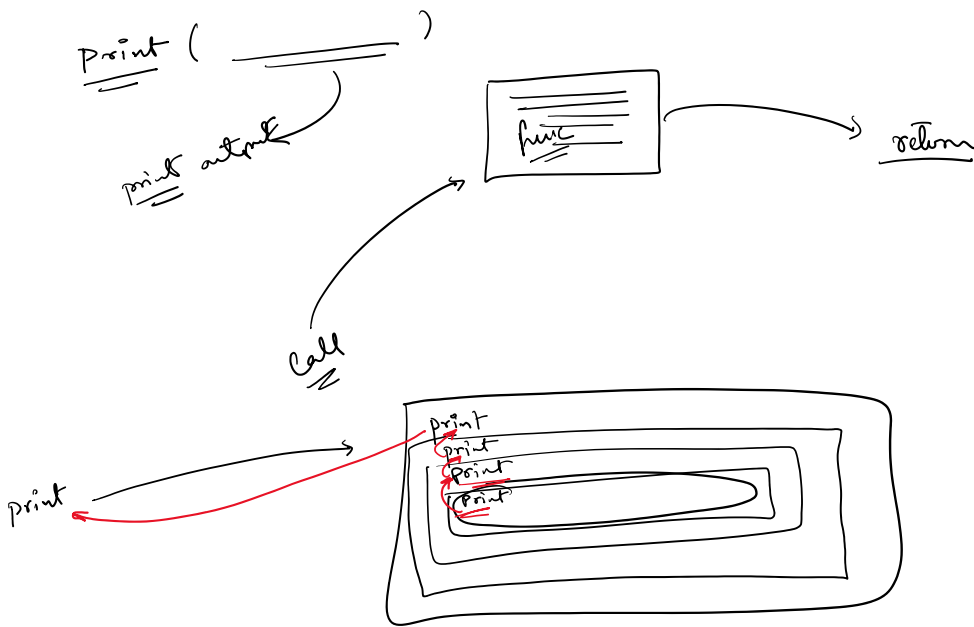




4/ Properties of list :-

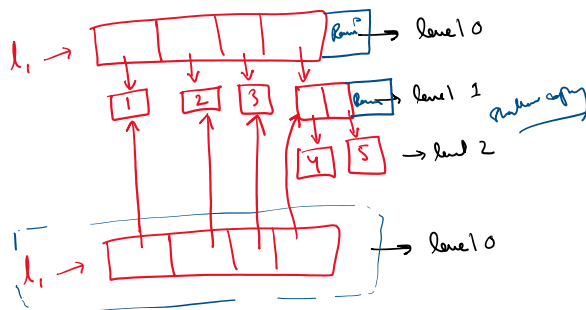
- ① Sequential \rightarrow indexing (sliding)
- ② Ordered
- ③ Mutable (can be changed (inplace))

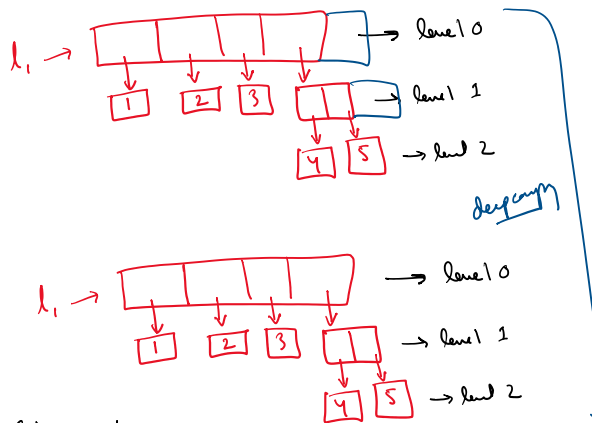
Methods of list :-



Ravi Rana

$l_1 = [1, 2, 3, [4, 5]]$





`append()` \leftarrow
`extend()` \leftarrow
 Copy \rightarrow shallow and deep copy
`clear()` \rightarrow

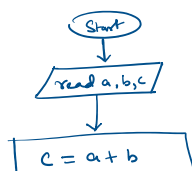
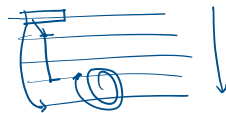
$$\begin{array}{r} 1100 \rightarrow 12 \\ \& 0111 \\ \hline 0100 \rightarrow 4 \end{array}$$

$$\begin{array}{r} 1100 \\ (8K) 1001 \\ \hline 1101 \rightarrow 13 \end{array}$$



Control flow

- ① Conditional Statements
- ② Iteration Statements
- ③ Break, continue

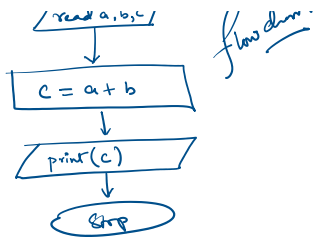


flowchart

pseudo code

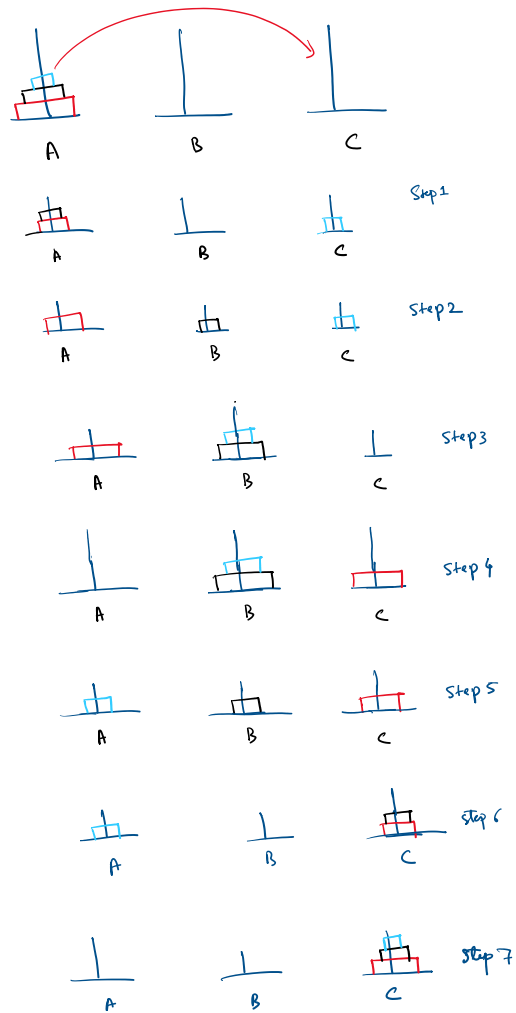
```

Start
read a, b, c
c = a + b
print c
  
```

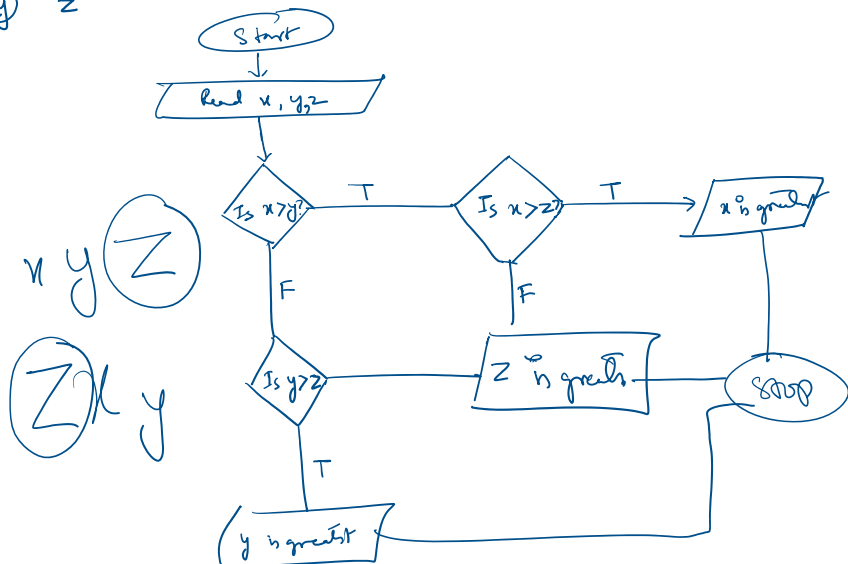


```

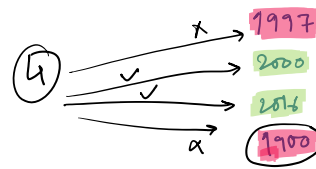
read a, b, c
c = a + b
print c
stop
  
```



N_1 N_2 N_3
 (x) (y) z

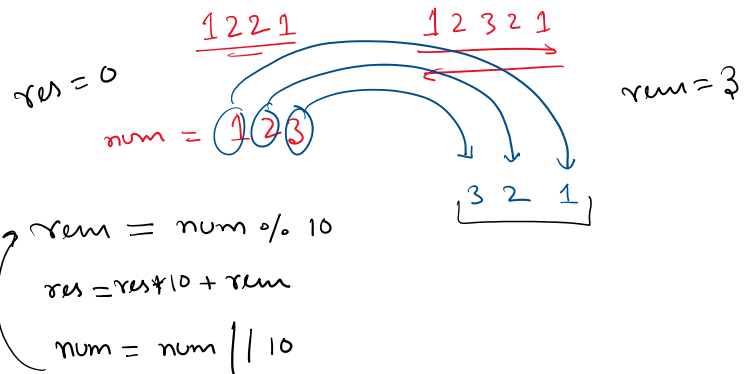


deep



$$\begin{matrix} \checkmark 2000 \\ \times 1900 \end{matrix} \rightarrow \text{year} \% 400 == 0$$

$$\begin{matrix} 2016 \\ 1997 \end{matrix} \rightarrow \text{Non century} \\ \text{year} \% 100 != 0 \text{ and } \text{year} \% 4 == 0$$



4 Dhaaba

5 veg
3 Non veg
(15) km

Food packets

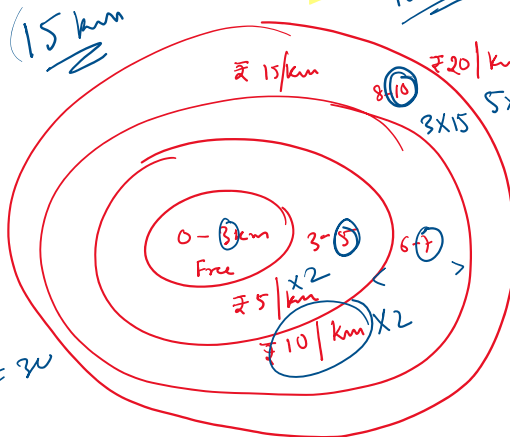
veg Non veg
₹ 250 ₹ 350

veg-qty =
non-veg-qty =
distance = 15 km

10 > km

Home Work

(15 km)



₹ 20/km
8(10) 3x15 5x20 16
15-3 free

if 12-2
2 * 5 = 10

10-2 = 8

2 * 10 = 20

8-2 = 6

2 * 15 = 30

6 * 20 = 120 + 30 + 2 * 10 = 180

d = 5 price = 20

if d > 8 and d <= 10:

$$d\text{-change} = (d-3) * 5 + (d-5) * 10 + (d-8) * 15$$

$$\text{price} = (d-8) * 15$$

d = 8

$$\text{price} = \text{price} + (d-5) * 10$$

210 + 30

$$\text{price} = \text{price} + (d - 5) * 10$$

30 + 30

$d = 5$

$$= 180$$

₹ 500
 ₹ 200
 ₹ 100
 ₹ 50
 ₹ 20
 ₹ 10
 ₹ 5
 ₹ 2
 ₹ 1

₹ 25677 ÷ 500 →
 177 ÷ 100 →
 77 ÷ 50 →
 27 ÷ 20 →
 7 ÷ 5 →
 2 ÷ 2 →

8 → 10

$$(3 * 0) + (2 * 5) + (2 * 10) + ((d - 7) * 15)$$

5 → 7

$$(3 * 0) + (2 * 5) + ((d - 5) * 10)$$

Iteration Statements

(i) While

Syntax :-

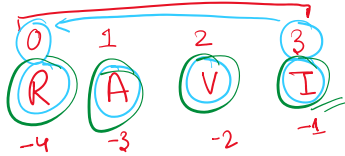
while condition :

→ Statement 1
 → ——— 2
 → ——— 3
 → ——— n

$$\begin{array}{rcl} 5 \times 1 & = & 5 \\ 5 \times 2 & = & 10 \\ | & & | \\ | & & | \\ | & & | \\ 5 \times 10 & = & 50 \end{array}$$

0 1 2 3

len(s)

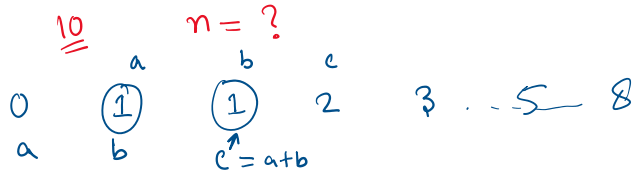


print(s[3])

print(s[2])

print(s[1])

print(s[0])



a = a + b
0 + 1

a = 1
0 1 2 3 5 8 13 21 ...

c = 1
a, b = 0, 1
while c <= n:

d = a + b
print(d)
a = b
b = d