Lists-1

(10 Assignments + 5 HW)

- ✓ 1. Introduction
- ✓ 2. Indexing
- ✓ 3. Append
- 4. Type
- 5. Negative Indexing
- 6. Operations
- 7. Problem Solving.
- 8. Up Next!
  - ↳ 2-3

Lists

=> Total 4 classes on lists

Announcement

=> Problem Solving Session This sunday 5th June

9 pm - 11 pm (*)

Optional

(Least solved A+HW

=> 5-6 problems)

=> Recording will be avl.

=> Any specific ques that you wish to be covered

Kindly DM on Slack/WA.

☆ Data Structure ?

Containers

	Bowl,	Plate,	Glass
Rice	✓	✓	✗
Lemon Juice	✓	✓	✗
Amras	✗	✗	✓

Depending on what data needs to be stored & how it is to be accessed ⇒ we have different D.S. data structures

✓ Python inbuilt D.S. ⇒

- List / Arrays.
- Set
- Tuples
- Dictionaries

⇒ Inbuilt Data types.

4 basic data types

- int
- float
- str
- bool

- Linked List
- Trees
- Graphs.

not provided in-built.

Adv DSA

Favorite Cricketer

Sachin 4 ever.

ODI matches.

463

Q) Write a python program that stores runs scored by Sachin in all ODI matches & finds min, max, sum, avg -- stats.

$$\left\{\begin{array}{l} s1 = \\ s2 = \\ s3 = \\ s4 = \\ s5 = \\ \vdots \\ s463 = \end{array}\right.$$

Not possible to handle these many variables.

$$\underline{s1} + \underline{s2} + \underline{s3} + \underline{s4} + \dots + \underline{s463}$$

$$\underset{\substack{\uparrow \\ \text{runs.}}}{s} = \underset{\substack{\uparrow \\ \text{square} \\ \text{bracket}}}{[} \underbrace{1,\ 100,\ 70,\ 24,\ 36,\ \dots,\ 59}_{463 \text{ elements.}} \underset{\substack{\uparrow \\ \text{Closing} \\ \text{square} \\ \text{bracket.}}}{]}$$

How to access the data? ⇒ using index

```
runs = [0,
    99,
    100,
    15,
    16,
    56,
    34]
print(runs)
```

runs =	[0,	99,	100,	15,	16,	54,	34]	
Match	1	2	3	4	5	6	7	
Position	0	1	2	3	4	5	6	−1

⇒ Index

N = 7 elements

print(runs[4]) # 16

$\boxed{\text{list\_name[index]}}$

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

len(l) # 8

l=	1	2	5	3	2	6	5	4
	0	1	2	3	4	5	6	7

Index for 6 is 5.

$\boxed{\text{len(l) − 1}}$

Append

```
l = [1, 3, 5, 7, 9]

l.append(17)
```

17 gets added to the end of list l.

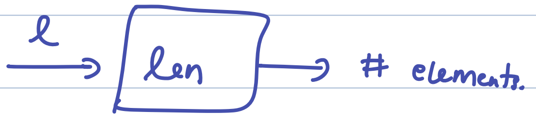
1	3	5	7	9	17

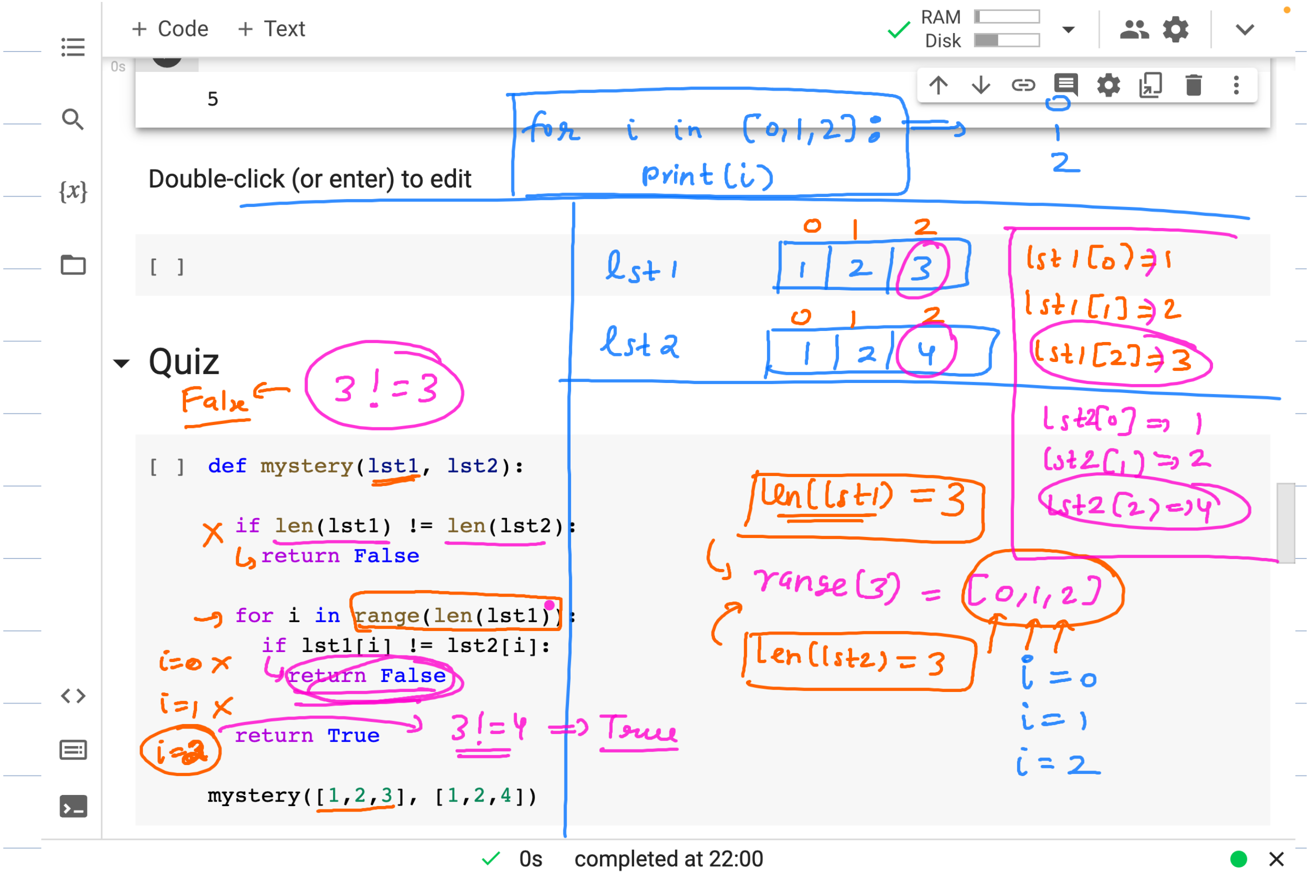
How to iterate?

```
runs= [1, 2, 4, 0, 5, 6, 7]

res = 0
  iterator                        list name
for   x   in   runs :
        res = res + x

print(res)
```

len (l) → returns no. of elements.

list name





```
5
```

Double-click (or enter) to edit

```
for i in [0,1,2]:
    print(i)
```

0
1
2

lst1 [1 | 2 | 3], indices 0 1 2

lst2 [1 | 2 | 4], indices 0 1 2

lst1[0] => 1
lst1[1] => 2
lst1[2] => 3
lst2[0] => 1
lst2[1] => 2
lst2[2] => 4

[ ]

Quiz

False ← 3 != 3

```
[ ] def mystery(lst1, lst2):

        if len(lst1) != len(lst2):
            return False

        for i in range(len(lst1)):
            if lst1[i] != lst2[i]:
                return False

        return True

    mystery([1,2,3], [1,2,4])
```

len(lst1) = 3
range(3) = [0,1,2]
len(lst2) = 3

i = 0
i = 1
i = 2

i=0 ×
i=1 ×
i=2

3 != 4 => True

✓ 0s completed at 22:00

$$l = [\overset{0}{1}, \overset{1}{2}, \overset{2}{4}]$$

len(l) => 3

for i in $\underbrace{\text{range}(\overset{3}{\text{len}(l)})}_{[0,1,2]}$:

print(i)

=> 0
1
2

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

$$l_2 = [1, 2, 4]$$

↓

return False

Checking whether 2 list are equal!

Break

Print last element

$$L = [\overset{0}{0}, \overset{1}{100}, \overset{2}{99}, \overset{3}{77}, \overset{4}{65}]$$

↑

print (len(l)) # 5

len(l) − 1

5 − 1 => 4.

print (l [len(l) − 1]) # 65

$\Downarrow$

5 − 1 = 4

```
                    0    1     2    3    4
                L = [0, 100, 99, 77, 65]
                     ↑    ↑     ↑    ↑    ↑
Negative Index      -5   -4    -3   -2   -1
```

List Operation

i) pop( ) => removes the last element

```
l = [ 4, 1, 2, 3]
l.pop()          # [4,1,2]
```

ii) pop (i) => removes the element at idx i

```
l.pop(0)         # [1,2]
```

Add at some position

list name

l . insert (index, val)

```
     0  1  2  3  4
l = [1, 2, 3, 5, 7]
```

```
l. insert (3, 4)     #   0  1  2  3
                        [1, 2, 3, 4, 5, 7]
```

index

QnA   Doubts