Python

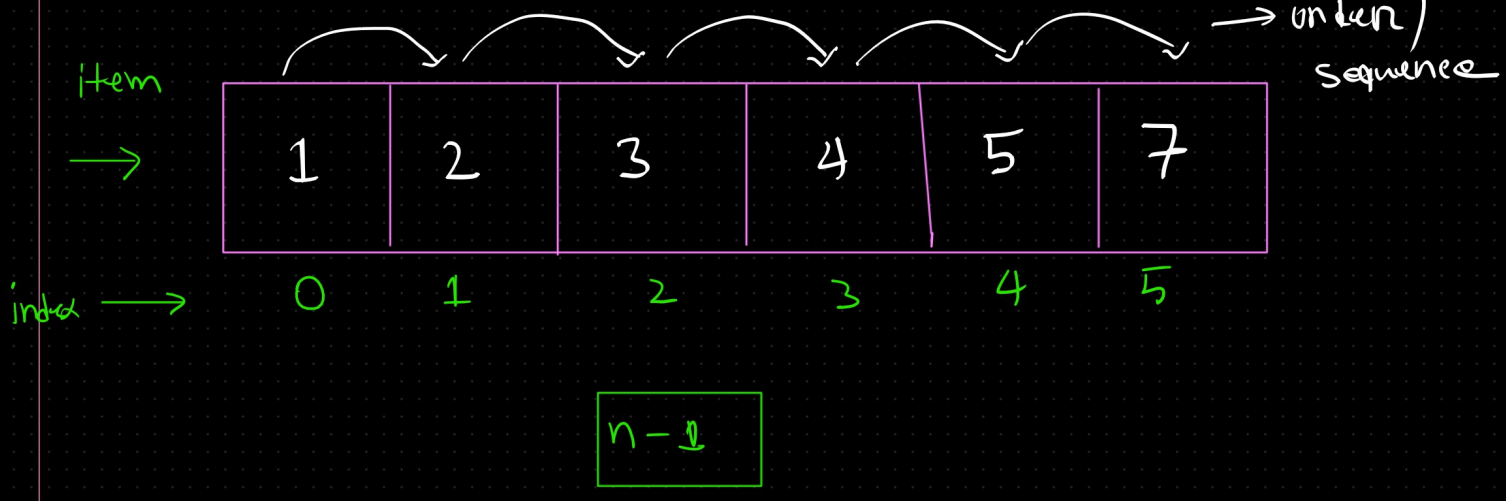
1. Python Basic
2. Data Type in Python
3. Python Indexing
4. Type casting and slicing
5. Operators in python
6. Relational Logical and identity Relationship
7. String formatting
8. conditional statement → if else
9. loops → for, while
10. Nested loops, pattern, Range function
11. Python Data Structure → Deepcopy
    - List / Array
    - Tuples
    - Dictionary
    - sets
12. Function in python
13. OOPS → Object oriented programming
14. Rest API
15. DataBase
16. python projects

Lists / Array

* Lists are ordered collection of data items.
* They store multiple items in a single variable
* List items are separated by commas and enclosed within square brackets → [ ]
* List are changeable / mutable – meaning we can alter them after creation.

Example:

```
lst1 = [1, 2, 3, 4, 5, 7]
lst2 = ["Bappy", "Alex", "Sunny", "krish"]
lst3 = ["Bappy", 1, 0.3, True]
```



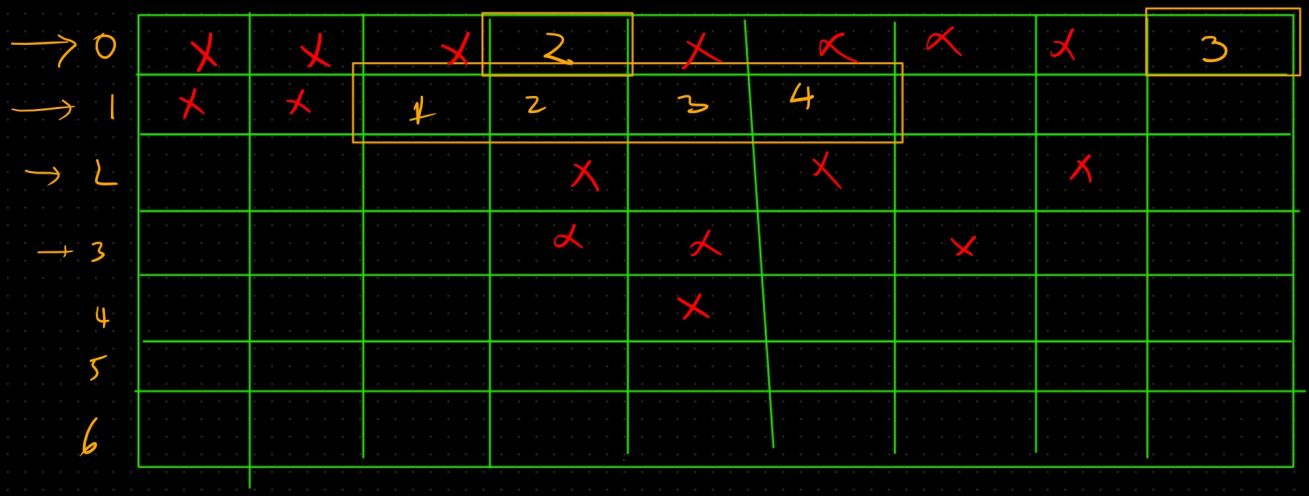
Variable:

$a = 3$ $\longrightarrow$ 0 $\boxed{3}$ memory loc

$b = 4$ 10 $\boxed{4}$ memory loc

RAM

→ 0	x	x	x	2	x	x	x	x	3
→ 1	x	x	1	2	3	4			
→ 2				x		x		x	
→ 3				x	x		x		
4					x				
5									
6									



$b = 3$
$a = 2$

lst = [1, 2, 3, 4]
0 1 2 3

Why list Data Structure Important?

Stocool →

class-1 = ["Bappy", "Alex", "Krish", "Suny"]

class-2 ⇒ ["price", "sandeep" - - -]

class-3 ⇒ [" — — — —]

print(class2) →

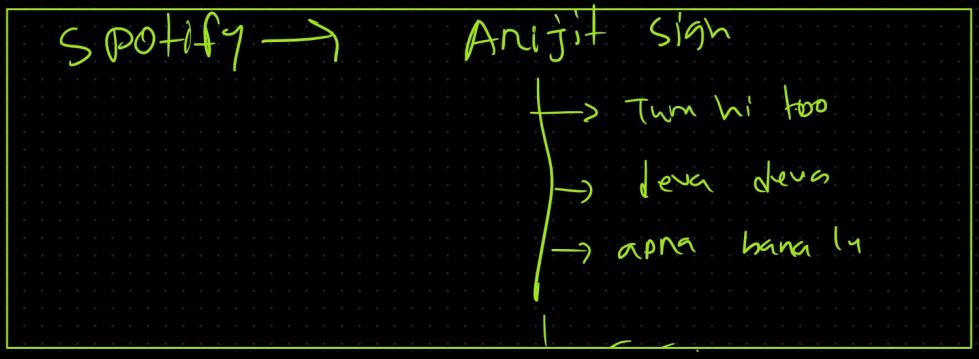
class-1_1 ⇒ "Bappy"

class-1_2 ⇒ "Alex"

class-1_3 ⇒ "kris"

print (calss-1)

print (class-1)



programmer → Arijit ⇒ ["Tum hi ho", "deva deva", apna bana lo"]

list ⇒ ["Bappy", "Bappy"]

0 1

colon ⇒ ["Yellow", "Green", "Red"]

if "Red" in colon:

print ("yes")

Indexing → [start : end : step]

List comprehension

Next → List method

- append()
- clear()
- copy()
- count()
- extend()
- Index()
- insert()
- pop()
- remove()
- Reverse()
- sort()