



- NIDHI VAISHNAV

# Learning



# List

- It can be list of numbers (integer / float), strings, or list
- `someListName = []`
- `myList = []`
- `myList = [1,2,3,4]`
- `leagueList= ["Justice league", "Avengers"]`
- List can be multi dimensional (2D list)
- List index starts from 0

# List methods:

- *List.append(x)*
  - Add an item at the end of the list
- *List.extend(iterable)*
  - Extend the list by appending all the items from the iterable
- *List.insert(index, x)*
  - Insert item x at given position index.
- *List.remove(x)*
  - Remove the first item from the list whose value is x, error if x does not exist
- *List.pop([index])*
  - Remove item which is at the index and returns it
  - If no index is specified, it will (remove + return) the last item on the list



# List methods:

- *indexVal = List.index(x)*
  - Provides first index of x item
- *countVal = List.count(x)*
  - Returns number of times x appears on the list
- *List.sort()*
  - Sort the item of list in place
- *List.reverse()*
  - Reverse the elements of the lists
- *newList = List.copy()*
  - Return a shallow copy of list

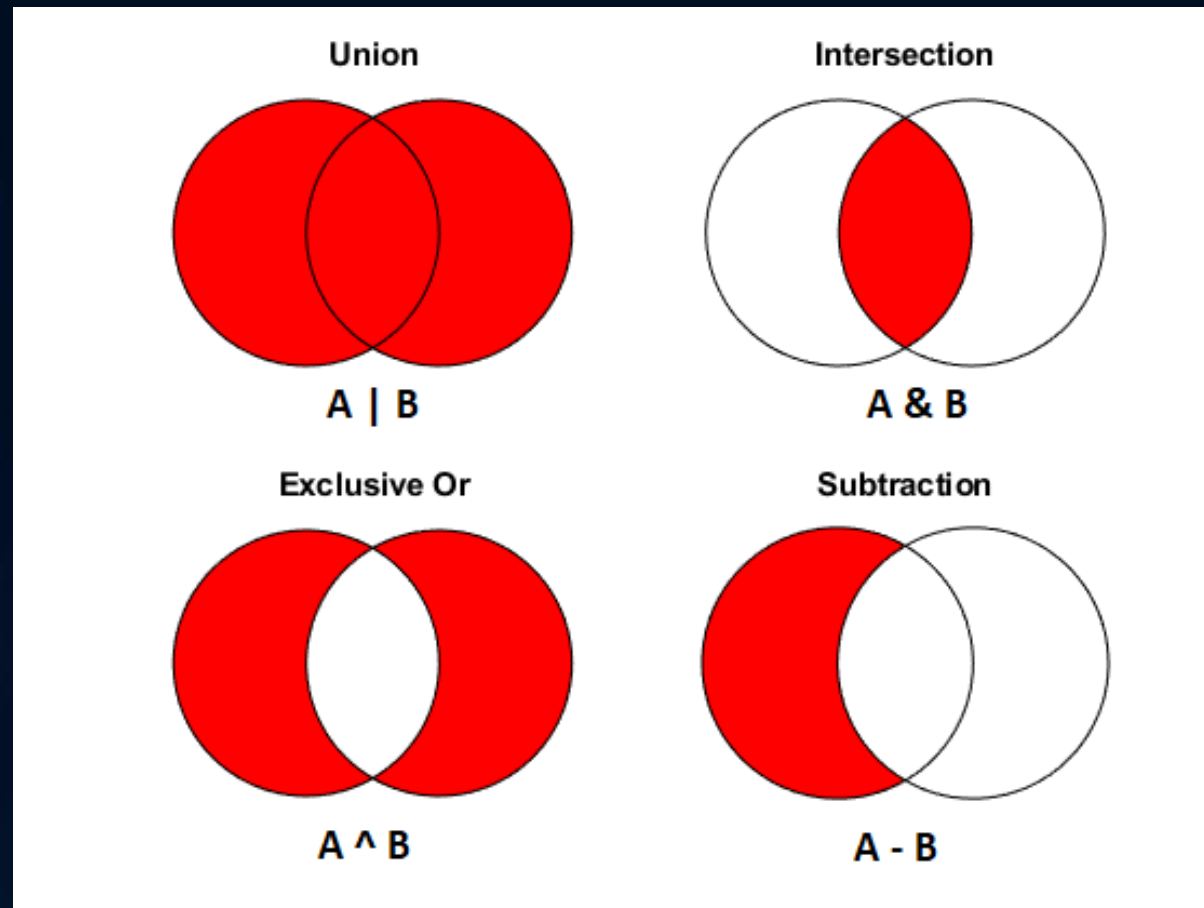
# List input

- `myList = [int(x) for x in input.split()]`
- `myList = list(map(int, input().split()))`

# Set

- Unordered collection with no duplicate elements
- `mySet = set()`

# Set operations





# Dictionary

- Associative memory (associative array)
- Key – value pair
  - Keys are unique
  - Key can be a string or number but not list
- `myDict = {}`
- `justiceDict = dict([("Batman", "Bruce Wyne"), ("Superman", "clark Kent"), ("wonder women", "Diana")])`

# Len()

- It gives length of list, set or dictionary

# Loops: For

- `for i in range(5):`
  - `#do something` – this task will be executed 5 times
- `for item in myList:`
  - `Print (item)`
- `For index, item in enumerate(myList):`
  - `Print (index, item)`

# Loop: while

- While condition:
  - #do something - this task will be executed till the condition remains true

i=5

- While i>0:
  - Print (i)



Tasks



# Basic operations:

- Insert 0 5
- Insert 1 10
- Insert 0 6
- Insert 2 5
- remove 6
- append 9
- append 1
- reverse
- Count 5
- Index 1
- Pop
- Extend [200,100]
- Sort
- copy

# Set operations

- User inputs 2 list of names
- Perform following operations:
  - Find common items
  - Combine all the items in 1 list, but do not repeat items
  - Find the items which are in list 1 but not in list 2
  - Find the items which are not common in both list

# Create super heroes dictionary

- Here, key is the name of super hero
- Value is his / her alliance to the super hero team

# Remove duplicate, find largest and smallest item from list

- Take list of numbers as user input
- Remove duplicates
  - Hint: set
- Find the smallest number
- Find the largest number

# Print multiplication table using while loop

- Take a number as user input
- Print it's multiplication table in the format:
  - $1 * 1 = 1$
  - $1 * 2 = 2$
  - :
  - :



# Triangle Pattern 1

- Take a number n as user input, and print following pattern in n lines

- \*

- \* \*

- \* \* \*

- \* \* \* \*

#for user input 4

# Triangle pattern 2

- Take a number n as user input, and print following pattern in n lines

- \* \* \* \*

- \* \* \*

- \* \*

- \*

#for user input 4

# Triangle pattern 4

- Take a number n as user input, and print following pattern in n lines

- \* \* \* \*

- \* \* \*

- \* \*

- \*

#for user input 4

# Triangle pattern 5

- Take a number n as user input, and print following pattern in n lines

- \*  
• \* \*  
• \* \* \*  
• \* \* \* \*

#for user input 4

# References:

- <https://docs.python.org/3/tutorial/datastructures.html>