**List as Stacks and Queues**

**Stacks - LIFO data structure**

= stack = []  
= append() and pop()

=stack = [int(x) for x in input().split()] = Четене на числа на един ред

**Queue - FIFO data structure**

from collections import deque  
= queue = deque()

= append(), appendleft(), pop() and popleft()

= print(" ".join(map(str, queue))) - За принтиране трябва да е STR

**Tuples**

They are immutable, nonunique and ordered  
= tuple = tuple()

= tuple = tuple(map(int, input().split()))  
= tuple = [int(x) for x in input().split(' ')]

**Sets**

They are mutable, unique and unordered  
= set = set()

= set = set(int(x) for x in input().split(' ')) - Добавяне на числа в сет  
= [set.add(x) for x in input().split()] - - Добавяне на стойности в сет  
= [set.add(int(x)) for x in command[2:]] - Добавяне на числа от определена позиция  
= [print(s) for s in set]  
= ', '.join([str(x) for x in set]) = Ако е число, трябва да се превърне в Стринг, иначе дава грешка  
= add() and remove()  
= intersection() - Проверява дали се съдържа информация от един сет в друг  
= union() - Обединява два сета  
= symmetric\_difference() - Показва всички стойности, които не са общи в двата сета  
= difference() - Показва стойностите, които не са общи с първия сет

**Dictionaries**

= dict = {}  
= [print(f"{key} - {value} times") for key, value in dict()]  
= append()  
= sorted()

Matrix

= matrix = [[0 for j in range(2)] for i in range(3)] - Creating a matrix with zeros

= matrix = [[j for j in range(1, 4)] for i in range(3)] - Creating a matrix with numbers

= matrix = [[1, 2, 3], [4, 5, 6]] - Flattening a matrix  
= flattened = [num for sublist in matrix for num in sublist]

**Други**

= end=" " - Прехвърля на нов ред  
= f'{x:.2f}' - Закръгля до две числа  
= CTLR + WIN = Показва информация за функциите