→ Conditional Statements: if, else, elif, nested

→ Loops in Python: for, while, nested loops

→ Loop control statements: break, continue, pass

[Day 2]

Lists

→ allows us to store multiple items in a single variable
→ created by placing elements inside square brackets [].
→ Can store data of diff. data types in a list
→ Characteristics are: Ordered (allow maintain order of elem.)
Mutable (items can be changed after creation)
Allow duplicates (they contain dup. values)

Topics

↳ Accessing element : list[index]
↳ Negative index : list[-1]
↑ (last element)
↳ Slicing list : list[index1 : index n]
↑ (not included)

[Note]: If you omit the start index the slicing starts from the first element, similarly if you omit the last index, the slicing ends at the last element.

↳ list.append() – adds an item to the end of the list
list.insert() – adds/inserts item at specified index
list.pop() – returns & remove item present at given index
list.clear() – removes all items from list
list.sort() – sorts list
list.copy() – return shallow copy of the list

List comprehension [expression for item in list if condition==true]

It offers a concise way to create a new list based on the values of an existing list.

For loop:
```
for num in numbers:
    square_num.append(num*num)
print(square_num)
```

List comprehension:
```
square_num = [num*num for num in numbers]
print(square_num)
```

Conditionals in list comprehension: [num for num in range(1,10) if num%2==0]

Tuple

→ Created using ()
→ tuple() constructor
→ Characteristics are: Ordered – (maintain order of elements)
Immutable – (can't be changed after creation)
Allow duplicates – (can contain duplicate values)
→ Accessing by index tuple[index]
→ can iterate over items of tuple using for loop
→ Check if an item exists in tuple: print('item' in tuple)
→ We can't delete an individual items in tuple but tuple itself (del tuple)