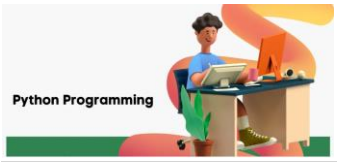
LECTURE FIVE

List

It may happen that you have to read, store, process, and finally, print hundreds, perhaps even thousands of numbers. Do you need to create a separate variable for each value? Will you have to spend long hours writing statements like the one below?

```
var1 = int(input())
var2 = int(input())
var3 = int(input())
:
:
var1000 = int(input())
```

- Python offers a range of compound data types often referred to as sequences. List is one of the most frequently used.
- List is an ordered sequence of items. It is one of the most used data type in Python and is very flexible.
- List is a collection which is ordered. Allows duplicate members.
- Lists are mutable, meaning their elements can be changed.
- Declaring a list is pretty straight forward. Items separated by are enclosed within **brackets** [ ].
- All the items in a list do not need to be of the same type. It can have any number of items and they may be of different types (integer, float, string etc.).
- A list can also have another list as an item. This is called a nested list (Next lecture).
- We can access item in list using **indexing** and a **range** of characters using **slicing** with the **bracket** operator. **Index** starts from **0**. Trying to access a character out of index range will raise an **IndexError**. The index must be an **integer**. Python allows **negative** indexing for its sequences.



Example: Create List

```
list1 = []                  # empty list
list2 = [1, 2, 3]           # list of integers
list3 = [1, "Hello", 3.4]   # list with mixed data types
```

Example: List Indexing

```
mylist = ['TDS', 'medium', 597, 99.99, True, 'Learn']
```

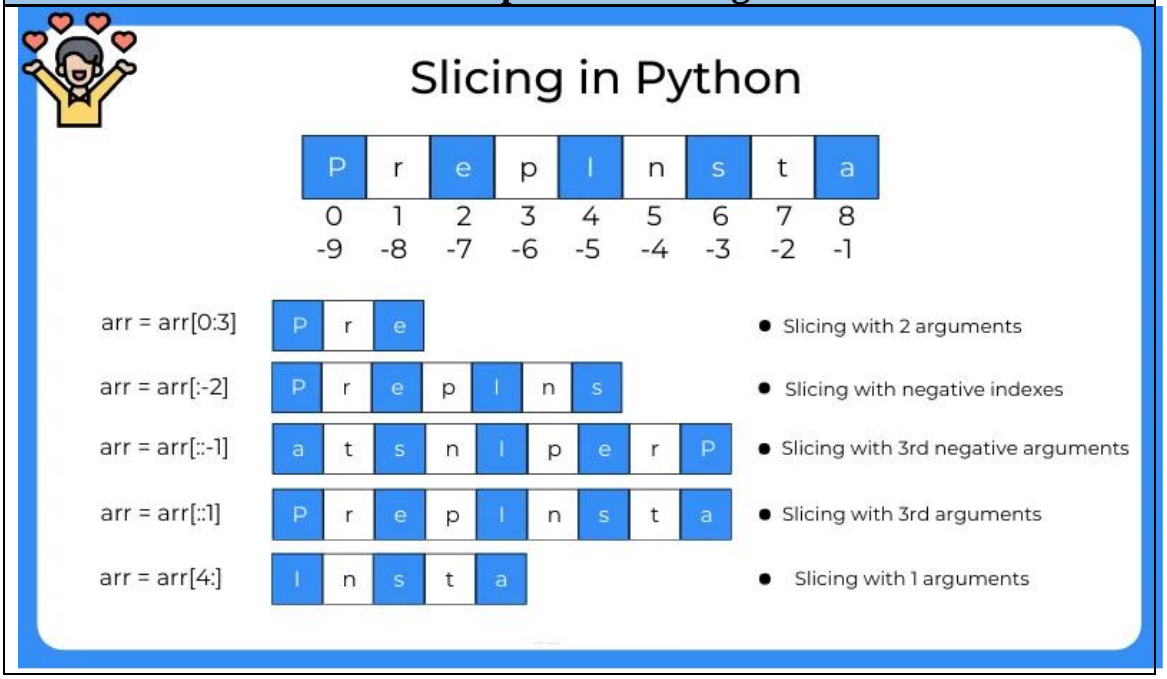
Indexing

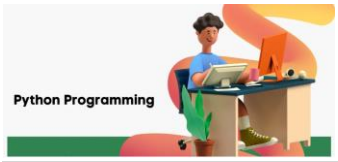
```
mylist[0] = TDS
mylist[1] = medium
mylist[2] = 597
mylist[3] = 99.99
mylist[4] = True
mylist[5] = Learn
```

Reverse -Indexing

```
mylist[-6] = TDS
mylist[-5] = medium
mylist[-4] = 597
mylist[-3] = 99.99
mylist[-2] = True
mylist[-1] = Learn
```

Example: List Slicing





Change Elements of a list

Lists are mutable, meaning their elements can be changed. We can use the assignment operator (=) to change an item or a range of items.

Example: Change Elements of a list
x = [2, 4, 6, 8] x[0] = 1 print(x) x[1:4] = [3, 5, 7] print(x)

Output

[1, 4, 6, 8]
[1, 3, 5, 7]

Operations and Functions on List

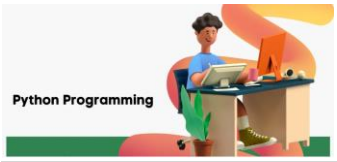
List have their own set of permissible operations. In general, lists can be:

- Concatenated (joined): The + operator does this in Python. Simply to combine two lists together.
- Replicated: The * operator can be used to repeats a list for the given number of times.
- python len function: it returns the number of items (length) in an object. use the len() to get the length of the given list.

Example: Concatenated list and Replicated and len
f = [1, 3, 5] k=f + [9, 7, 5] r= f * 3 print(k) print(r) print(len(r))

Ouput

[1, 3, 5, 9, 7, 5]
[1, 3, 5, 1, 3, 5, 1, 3, 5]
9



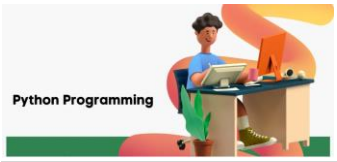
Common Python List Methods

There are numerous methods available with the list object, some of the commonly used methods are:

Method	Description
append()	Add an element to the end of the list
extend()	Add all elements of a list to the another list
insert()	Insert an item at the defined index
remove()	Removes an item from the list
pop()	Removes and returns an element at the given index
clear()	Removes all items from the list
index()	Returns the index of the first matched item
count()	Returns the count of the number of items passed as an argument
sort()	Sort items in a list in ascending order
reverse()	Reverse the order of items in the list

Example: Common Python List Methods

```
my_list=[3 ,2 ,1]
my_list.append(4)
print("After append():", my_list)
my_list.extend([6 ,5])
print("After extend():", my_list)
my_list.insert(10 ,2)
print("After insert():", my_list)
my_list.remove(2)
print("After remove():", my_list)
popped_element = my_list.pop(3)
print("Popped element:", popped_element)
my_list.clear()
print("After clear():", my_list)
my_list =[3 ,4 ,3 ,2 ,1]
print("Index of 3:", my_list.index(3))
print("Count of 3:", my_list.count(3))
my_list.sort()
print("After sort():", my_list)
```



```
my_list.reverse()
print("After reverse():", my_list)
```

Output

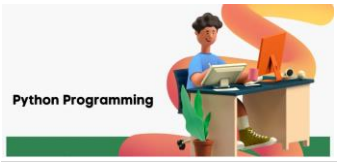
```
After append(): [1, 2, 3, 4]
After extend(): [1, 2, 3, 4, 5, 6]
After insert(): [1, 2, 10, 3, 4, 5, 6]
After remove(): [1, 10, 3, 4, 5, 6]
Popped element: 4
After clear(): []
Index of 3: 2
Count of 3: 2
After sort(): [1, 2, 3, 3, 4]
After reverse(): [4, 3, 3, 2, 1]
```

Example (1): Write Python Program to read list of numbers and print item of list using function called "create_list"

```
def create_list():
    z= []
    n = int(input("Enter length of list:"))
    for i in range(n):
        it = int(input("Enter item:"))
        z.append(it)
    return z
result_list = create_list()
print("Created list:", result_list)
```

Example (2)): Write Python Program to add item x at the defined index (y) of list [10, 5, 7, 2, 1] using function called "add_item_at_index"

```
def add_item_at_index(lst, item, ind):
    lst.insert(ind, item)
    return lst
original_list = [10, 5, 7, 2, 1]
x = int(input("Enter item: "))
y = int(input("Enter index: "))
updated_list = add_item_at_index(original_list, x, y)
print("Updated list:", updated_list)
```



Example (3): Write Python Program to read list and find the count of each item in list

```
n=int(input("enter lenght= "))
ls=[]
for i in range(n):
        x=input("enter item =")
        ls.append(x)
for i in ls:
        print(i,ls. count(i))
```

Example (4): Write Python Program to sum all the items in a list

```
n=int(input("enter lenght= "))
ls, s =[],0
for i in range(n):
        x=int(input("enter item ="))
        ls.append(x)
for i in ls:
        s+=i
print(s)
```

Example (5): Write Python Program to split the odd numbers and even numbers form one list to two lists using two functions

```
def split_odd_even(numbers):
   odd_numbers = []
   even_numbers = []
   for num in numbers:
      if num % 2 == 0:
         even_numbers.append(num)
      else:
         odd_numbers.append(num)
   return odd_numbers, even_numbers
n=int(input("enter length= "))
numbers_list = []
for i in range(n):
        x=int(input("enter item ="))
        numbers_list.append(x)
odd_nums, even_nums = split_odd_even(numbers_list)
print("Odd numbers:", odd_nums)
print("Even numbers:", even_nums)
```

Example (6): Write Python Program to get the largest number from a list.

```
n=int(input("enter length= "))
ls[]=
for i in range(n):
        x=int(input("enter item ="))
        ls.append(x)

max =ls[0]
for a in ls:
      if a > max:
         max = a
print("largest= ",max )
```

Key Points to Remember:

- Python allows negative indexing for list.

  The index of -1 refers to the last item, -2 to the second last item and so on.

- list object does support item assignment