

PYTHON LIST
CLASS XI
(MODULE-1)
BY

Mrs. SUJATA PRADHAN,
PGT(SS), Computer Science,
AECS,ANUPURAM

- Python Datatypes
- Python List
- A **sequence** is an ordered collection of similar or different data types. Python consists of several data types which are capable of storing sequences, but the most common and reliable type is the **list**. A list in Python is used to store the sequence of various types of data. It also allows duplicate members. A list can be defined as a collection of values or items of same or different types. The items of a list are separated with comma (,) and enclosed with the square brackets [].
- Characteristics of Lists

- The list has following **characteristics**:
 - The lists are ordered or sequenced.
 - Each element of the list is accessed by index.
 - The lists are mutable types.
 - A list can store different types of elements.
-
- **Note:** Each element of a sequence is assigned a number i.e. its position or index. The first index is zero, the second index is one, and so on. We can access an item from a list using index. In Python, indices start from 0. So, a list having 5 elements will have an index from 0 to 4. Trying to access indices other than mentioned range will raise

an IndexError. The index must be an integer. We can't use float or other types, otherwise this will result in TypeError. Nested lists are accessed using nested indexing.



How to create a list?

In Python programming, a list is created by placing all the items (elements) inside square brackets [], separated by commas. It can have any number of items and they may be of different types (integer, float, string, tuples etc.).

empty list

```
list1 = []
```

list of integers

```
list2 = [1, 2, 3]
```

list of floats

```
list3 = [11.22, 9.9, 78.34, 12.0]
```

list with mixed data types

```
list4 = [1, "Hello", 3.4]
```

If we try to print the type of L1 and L2 using **type()** function then it will come out to be lists.

```
L1 = ["Sam", 102, "USA"]
```

```
L2 = [11, 20, 3, 47, 55, 96]
```

```
print(type(L1))
```

```
print(type(L2))
```

Output:

```
<class 'list'>
```

```
<class 'list'>
```

- List Operations

The most basic **data structure** in Python is the **sequence**. We can do certain operations like indexing, slicing, adding, multiplying, and checking for membership with sequences. In addition, Python has built-in functions for finding

the length of a sequence and for finding its largest and smallest elements.

List indexing

```
mylist = ['p', 'r', 'o', 'b', 'e']  
print(mylist[0])
```

Output: p

```
print(mylist[2])
```

Output: o

```
print(mylist[4])
```

Output: e

-

List Indexing & List Slices:

- Positive & Negative Index
- **Negative indexing**
Python allows negative indexing for its sequences. The index of -1 refers to the last item, -2 to the second last item and so on.

Negative indexing in lists

```
mylist = ['p','x','o','m','s']  
print(mylist[-1])
```

```
print(mylist[-5])
```

output:

s p

- Slices....
-

Mutability:

```
# Python code to test that lists are mutable  
color = ["red", "blue", "green"]  
print(color)
```

```
color[0] = "pink"  
color[-1] = "orange"  
print(color)
```

OUTPUT

```
['red', 'blue', 'green']  
['pink', 'blue', 'orange']
```

-

Mutability:

Lists are mutable. An item can be changed in a list by accessing it directly as part of the assignment statement. It is possible to add, delete, insert, and rearrange items in a list or dictionary. Hence, they are mutable objects.

- SUMMARY
 - List structure and characteristics
 - Basic List operations
 - Positive & Negative Indexing
 - List slicing
 - Mutability

Thank You