1. Datatypes:

List:

List is one of the most frequently used & very versatile data type in python.

* How to create:

→ A list is created by placing all the items (elements) inside square brackets [], separated by commas.

→ It can have any no. of items, & they may be different types (int, float, string etc.).

Examples:

```
# empty list
mylist = []
```

```
#list of integers.
my-list = [1,2,3]
```

```
# list with mixed data
my-list = [1, "Hii", 3.4]
```

→ A list can also have another list as an item.
i.e. known as nested list.

```
# nested list.
my-list = ["mouse", [8,4,6], ['a']].
```

* Access elements from a list:

(i) List index

(ii) Negative index.

Indices starts with 0.

(i) List Index: We can use the index operator [] to access an item in a list.

→ list having 5 elements will have an index from 0 to 4

→ Trying to access indexes other than these will raise an Index Error

→ The index must be an integer. Not float or any other type this will result in Type Error.

E.g: (I) # list indexing.

list = ['H', 'I']

output : P
print (list[0])

output : I
print (list[1])

(II) # Nested List.

List = ["Hii", [1,2,3]]

Nested indexing
print (list[0],[1])
print (list[1],[2])

print (list[4.0]) → Error. (only Int.)

Output: i
3

Page:
Date: / /

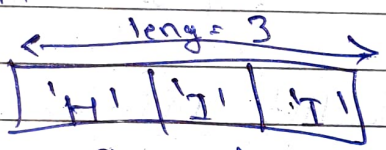
(iii) Negative Indexing:

→ Python allows -ve indexing for seq.
→ The index of -1 refers to the last item, -2 to the 2nd last item & so on.

Ex: # Negative Indexing:

```
List = ['H', 'I', 'I']
print (list [-1])
print (list [-3])
```

output: I
H



Index 0 1 2
-ve Index -3 -2 -1

* Slice List:

We can access a range of items in a list by using the slicing operator. : (colon).

Ex: # list slicing.
list = ['I', 'I', 'A', 'B', 'C']

elements 2nd to 4th
print (list [1:4])

output: ['I', 'B']

* Add or change elements:

Lists are mutable. We use assignment operator (=) to change an item.

Ex: # correct mistake in list.

```
odd = [1, 2, 3, 4]

# change the list item.
odd[0] = 10

print(odd)
```

Output: [10, 2, 3, 4]

* Add elements to list using append() or several elements by extend() method.

Ex. # appending & extending lists

```
odd = [1, 3, 4, 5]

odd.append(7)
print(odd)

odd.extend([1, 2, 3])
print odd
```

Output: [1, 3, 4, 5, 7]
[1, 3, 4, 5, 7, 1, 2, 3]

* Delete elements:

Delete one or more items from list using del. or even a list entirely

Ex: # Deleting list items.
list = ['H', 'E', 'L']

delete one item
del list[2]
print list

Output: ['H', 'L']

* Python list method:

append() => Add element
extend() => Add all element
insert() => Insert element
remove() => Remove element
pop() => Remove & return "
clear() => Remove all ele.
index() => Return the index
count() => return the count of no.
sort() => sort item in ascending order.
reverse() => Reverse the order
copy() => return a shallow copy.