BEINEX

Full Stack Internship Python

Python Sequences

- A sequence is a datatype that represents a group of elements.
- The purpose of any sequence is to store and process group elements.

Sequences

- Strings
- Lists
- Tuples
- Dictionaries

LIST

- A list is similar to an array that consists of a group of elements or items
- A list can store different types of elements

- Creating a list

  student = [556, “Mothi”, 84, 96, 84, 75, 84]

  Empty list = []

- Accessing Values in list:
- Use the square brackets for slicing along with the index or indices to obtain value available at that index.
- Example : student[0] or student[-7]

Looping on lists

- We can also display list by using for loop (or) while loop.
- The len( ) function useful to know the numbers of elements in the list.
  while loop retrieves starting from 0th to the last element i.e. n-1
- Ex-1: numbers = [1,2,3,4,5]

```
for i in numbers:
  print i
```

Output: 1 2 3 4 5

append()	Used for appending and adding elements to the end of the List.
copy()	It returns a shallow copy of a list
clear()	This method is used for removing all items from the list.
count()	These methods count the elements
extend()	Adds each element of the iterable to the end of the List
insert()	Inserts a given element at a given index in a list.
pop()	Removes and returns the last value from the List or the given index value.
remove()	Removes a given object from the List.
sort()	Sort a List in ascending, descending, or user-defined order
min()	Calculates the minimum of all the elements of the List
max()	Calculates the maximum of all the elements of the List

Python append()

Used for appending and adding elements to the List. It is used to add elements to the last position of the List in Python.

```
# Adds List Element as value of List.
List = ['Mathematics', 'chemistry', 1997, 2000]
List.append(20544)
print(List)
```

Output:

```
['Mathematics', 'chemistry', 1997, 2000, 20544]
```

Python insert()

Inserts an element at the specified position.

syntax : list.insert(position,element)

List = ['Mathematics', 'chemistry', 1997, 2000]

Insert at index 2 value 10087

List.insert(2, 10087)

print(List)

Output : ['Mathematics', 'chemistry', 10087, 1997, 2000, 20544]

Python extend()

Adds contents to List2 to the end of List1.

Syntax: List1.extend(List2)

List1 = [1, 2, 3] List2 = [2, 3, 4, 5]

```
# Add List2 to List1
List1.extend(List2)
print(List1)
# Add List1 to List2 now
List2.extend(List1)
print(List2)
```

Output:

```
[1, 2, 3, 2, 3, 4, 5]
[2, 3, 4, 5, 1, 2, 3, 2, 3, 4, 5]
```

Python sum()

Calculates the sum of all the elements of the List.

Syntax: sum(List)

```
List = [1, 2, 3, 4, 5]
print(sum(List))
```

Output:

15

The sum is calculated only for Numeric values, else wise throws TypeError.

Python count()

Calculates the total occurrence of a given element of the List.

Syntax: List.count(element)

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

print(List.count(1))
```

Output:

4

Python length - Calculates the total length of the List.

Syntax: len(list_name)

```
List = [1, 2, 3, 1, 2, 1, 2, 3, 2, 1]
print(len(List))
```

Output:

10

List Comprehensions

List comprehensions represent creation of new lists from an iterable object (like a list, set, tuple, dictionary or range) that satisfy a given condition.

List comprehensions contain very compact code usually a single statement that performs the task

We want to create a list with squares of integers from 1 to 100.

squares=[x**2 for x in range(1,11)]

TUPLE

- A Tuple is a python sequence which stores a group of elements or items.
- Tuples are similar to lists but the main difference is tuples are immutable whereas lists are mutable. Once we create a tuple we cannot modify its elements.
- Tuples are generally used to store data which should not be modified and retrieve that data on demand.

Creating Tuples

- We can create a tuple by writing elements separated by commas inside parentheses( ).
- Example : tup = ( )
- Accessing the elements from a tuple can be done using indexing or slicing.

Operations on tuple

len(t) - Return the length of tuple.

tup1+tup2 - Concatenation of two tuples.

cmp(tup1,tup2) - Compare elements of both tuples

max(tup) - Returns the maximum value in tuple.

min(tup) - Returns the minimum value in tuple.

tuple(list) - Convert list into tuple. tup.

count(x) - Returns how many times the element „x“ is found in tuple. **sorted(tup) -** Sorts the elements of tuple into ascending order. **sorted(tup,reverse=True) -** will sort in reverse order

Thank you