Advanced Data Structures

Besant Technologies

List Comprehension

- For creating list.
- Combining for loop iteration into a single line.
- Parts:
    - Output expression
    - Input sequence
    - Iterator
    - Optional if conditions

Examples:

- Without if:
    - x = [i for i in range(10)]
    - squares = [x**2 for x in range(10)]
    - multiplied = [item*3 for item in list1]

With if:

- numbers = [x for x in string if x.isdigit()]
- lst = [x ** 2 for x in range (1, 11) if x % 2 == 1]
- [i for i in range(8) if i%2==0 if i%3==0]
- ["Even" if i%2==0 else "Odd" for i in range(8)]
- [[i*j for j in range(1,11)] for i in range(7,9)]

Collections Module:

- Counter
- Defaultdict
- Ordereddict
- Namedtuple
- deque

Counter:

- It is a dict subclass.
- It counts the number of occurrences of elements.
- It forms a dictionary where elements are stored as keys and counts are stored as values
- Input can be any sequence type.
- Ex: count of elements in a list, count of chars in strings, number of words in a sentence etc.

defaultdict

- Dict like datatype
- Same as dictionary
- Except it will have default value.
- It will never show key error.
- It will assign the default value if any key doesn’t exists.

ordereddict

- Same as regular dict
- Except it maintains order of elements added.
- Ex: check the equality condition.

namedtuple:

- Creates a new tuple class with names as attributes.
- Syntax:
    - <class_name> = namedtuple(“Name”, “attributes separated by space”)
    - <object> = <class_name>(<values assigned to attributes>)

Ex:

- Dog = namedtuple(“Dog”, “name age breed”)
- D = Dog(name=”jks”, age=10, breed=”Lab”)

deque:

- List like datatype.
- Deque - Double ended queue
- Ex: d = deque(<sequence>)
- Iteration
- Indexing
- All list functions

Deque functions:

- append()
- appendleft()
- extend()
- extendleft()
- pop()
- popleft()
- clear()
- rotate()
- reversed()