SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 OF THE UGC ACT, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

III B.Tech.

Computer Science & Engineering

CSE304: PYTHON PROGRAMMING WITH WEB FRAMEWORKS

UNIT – II: Comprehensions

By
Mrs. S. KAMAKSHI, AP-III / CSE
School of Computing

Comprehension

- **Implicit definition for items to be part of the object**
- List Comprehension
- Tuple Comprehension
- Dictionary Comprehension
- Set Comprehension

List Comprehension

- L1 = [10, 20, 30, 40]
- L2 = [x+5 for x in L1]
- L3 = [x+5 for x in range(10,50,10)]
- s = "Think Merit, Think Transparency, Think SASTRA"
- L4 = [x for x in s.split(',')]
- L5 = [x.strip() for x in s.split(',')]
- L6 = [x.strip().upper() for x in s.split(',')]
- L7 = [x for x in range(50) if x%2 == 0]

Set Comprehension

- S1 = {10, 20, 30, 40}
- S2 = {x+5 for x in L1}
- S3 = {x+5 for x in range(10,50,10)}
- s = "Think Merit, Think Transparency, Think SASTRA"
- S4 = {x for x in s.split(',')}
- S5 = {x.strip() for x in s.split(',')}
- S6 = {x.strip().upper() for x in s.split(',')}
- S7 = {x for x in range(50) if x%2 == 0}

Dictionary Comprehension

- L1 = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']
- D1 = {x:ord(x) for x in L1}
- D2 = {x:x**2 for x in range(1,11)}
- L1 = [(x, ord(x)) for x in 'abcdefghijklmnopqrstuvwxyz']
- D3 = {x:y for (x, y) in L1}
- D3 = {x:y for x, y in L1}
- L1 = [['Joe', 48], ['Kim', 32], ['Raja', 25], ['Ajai', 60]]
- D4 = {a:b for [a,b] in L1}
- D4 = {a:b for a,b in L1}

Tuple Comprehension

- D1 = {'Joe': 48, 'Kim': 32, 'Raja': 25, 'Ajai': 60}
- T1 = tuple(D1.items())
- T2 = tuple((x.upper(), y) for x,y in D1.items())
- L1 = [1, 2, 3, 4]
- L2 = [10, 20]
- T3 = tuple((x, y, x*y) for x in L1 for y in L2)

Iterables and Iteration Protocol

- An object is considered as iterable if it is either a physically stored sequence, or an object that produces one result at a time in the context of an iteration tool like a for loop
- iterable objects include both physical sequences and virtual sequences computed on demand

Iteration Protocol

- The for loop first obtains an iterator from the iterable object by passing it to the **iter** built-in function; The iter function internally runs the __iter__ method of that object
- Then the **next()** built-in function is called which in turn calls the __next__() method of that iterable object.
- The next () built-in function also stops the iteration when **StopIteration** exception is raised and handles the exception.

Manual Iteration using __iter__() and __next__()

```
L = [1, 2, 3]
I = L.__iter__()     # Obtain an iterator object from an iterable
I.__next__()         # Call iterator's next to advance to next item
        1
I.__next__()
        2
I.__next__()
        3
I.__next__()
        ...error text omitted...
        StopIteration
```

Manual Iteration using iter() and next()


```
L = [1, 2, 3]
l = iter(L)     # Obtain an iterator object from an iterable
next(l)         # Call iterator's next to advance to next item
          1
next(l)
          2
next(l)
          3
next(l)
          ...error text omitted...
          StopIteration
```