04-12-23

List methods

Day 59

1) append ()

Syntax :

```
list.append (value)
```

Eg :

```
A = [10, 20, 30]
A.append (40)
print (A)  # [10, 20, 30, 40]
```

2) extend () Syntax : list.extend (list)

```
list1 = [10, 20, 30]
list2 = [40, 50]
list1.extend (list2)
print (list1)  # [10, 20, 30, 40, 50]
```

3) sort()

Syntax: sort(reverse = True/False, key = myFunc)

Eg: A = [2,3,1,4,7,6,5]

A.sort()

print(A) # [1,2,3,4,5,6,7]

4) insert()

Syntax: list.insert(index, object)

Eg: A = [1,2,3,5,6]

A.insert(3,4)

5) reverse()

Syntax: list.reverse()

Eg: A = [1,2,3,4,5]

A.reverse()

print(A)

6) count()

Syntax: list.count(x)

Eg: A = [1,3,3,5,6,3,3,7,3]

print(A.count(3)) # 5

7) len()

Syntax: len(list)

Eg:

A = [1,2,3]

print(len(A))

8) min()

Syntax: min(list)

A = [1,2,3,0]

print(mix(A)

9. max()

Syntax: max(list)

Eg: A = [1,2,3,0]
print(max(A))

10. index()

Syntax: list = index(value)

Eg: A = [1,2,3,4,5]
print(index(2))

List loop (Traversing a list)

⇒ Process of go through each element of a list sequentially

1) For loop

Syn: for <list_item> in <list>:
statements.

A = [10, 20, 30, 40, 50]
for i in A:
print(i)

2. for <index> in range(len(list)):
Statements.

Eg: A = [10, 20, 30, 40]
i = 0
for i in range(len(A)):
print(A[i])

j = 0
while j < (len(A)):
print(A[j])

3\. for i in []:
    print (i)

4\. A = [10, 20, [1, 2]]

len (A)   # 3

A [2] [0]   # [10, 20, [1, 2]]

(indices: 0 1 2 below the outer list; 0 1 above the inner list)

Mutability

→ changed (or) modified.

→ lists are mutable.

List membership.

in and not int operator.

Aliasing.

A = [1, 2, 3]   # A = [1, 2, 3]

B = A   # B ↗

print (B is A)   # True.

- The association of a variable with an object is called a reference.

Aliasing with mutable objects.

Possible / but Error prone

Aliasing with immutable objects

- String → immutable
- Not possible.