PYTHON – ARRAYS

Course Unit 7:
Week 9

OBJECTIVES:

1. Discuss the python dictionaries and arrays.
2. Demonstrate programs using python dictionaries and arrays.
3. Develop a python program using dictionaries and arrays.

- **Python – Arrays**
- Accessing Array Items
- Add Array Items
- Remove Array Items
- Loop Arrays
- Copy Arrays
- Reverse Arrays
- Sort Arrays
- Join Arrays
- Array Methods

PYTHON – ARRAYS

Course Unit 7:
Week 9

PYTHON – ARRAYS

- Python's standard library has array module. The array class in it allows you to construct an array of three basic types, integer, float and Unicode characters.

Syntax

- The syntax of creating array is −

```
import array
obj = array.array(typecode[, initializer])
```

PYTHON – ARRAYS

Parameters

- typecode − The typecode character used to create the array.
- initializer − array initialized from the optional value, which must be a list, a bytes-like object, or iterable over elements of the appropriate type.

Return type

- The array() constructor returns an object of array.array class

PYTHON – ARRAYS

```
import array as arr

# creating an array with integer type
a = arr.array('i', [1, 2, 3])
print (type(a), a)

# creating an array with char type
a = arr.array('u', 'BAT')
print (type(a), a)

# creating an array with float type
a = arr.array('d', [1.1, 2.2, 3.3])
print (type(a), a)
```

Arrays are sequence types and behave very much like lists, except that the type of objects stored in them is constrained.

```
<class 'array.array'> array('i', [1, 2, 3])
<class 'array.array'> array('u', 'BAT')
<class 'array.array'> array('d', [1.1, 2.2, 3.3])
```

PYTHON – ARRAYS

Python array type is decided by a single character Typecode argument.

type code	Python data type	Byte size
'b'	signed integer	1
'B'	unsigned integer	1
'u'	Unicode character	2
'h'	signed integer	2
'H'	unsigned integer	2
'i'	signed integer	2
'I'	unsigned integer	2
'l'	signed integer	4
'L'	unsigned integer	4
'q'	signed integer	8
'Q'	unsigned integer	8
'f'	floating point	4
'd'	floating point	8

- Python – Arrays
- **Accessing Array Items**
- Add Array Items
- Remove Array Items
- Loop Arrays
- Copy Arrays
- Reverse Arrays
- Sort Arrays
- Join Arrays
- Array Methods

PYTHON – ARRAYS

Course Unit 7:
Week 9

ACCESSING ARRAY ITEMS

- Since the array object behaves very much like a sequence, you can perform indexing and slicing operation with it.

```
import array as arr
a = arr.array('i', [1, 2, 3])
#indexing
print (a[1])
#slicing
print (a[1:])
```

ACCESSING ARRAY ITEMS

- Changing Array Items-You can assign value to an item in the array just as you assign a value to item in a list.

```
import array as arr
a = arr.array('i', [1, 2, 3])
a[1] = 20
print (a[1])
```

```
import array as arr
a = arr.array('i', [1, 2, 3])
# assignment
a[1] = 'A'
```

- Python – Arrays
- Accessing Array Items
- **Add Array Items**
- Remove Array Items
- Loop Arrays
- Copy Arrays
- Reverse Arrays
- Sort Arrays
- Join Arrays
- Array Methods

PYTHON – ARRAYS

Course Unit 7:
Week 9

ADD ARRAY ITEMS

The append() Method

- The append() method adds a new element at the end of given array.

Syntax

- array.append(v)

```
import array as arr
a = arr.array('i', [1, 2, 3])
a.append(10)
print (a)
```

Parameters

- v − new value is added at the end of the array. The new value must be of the same type as datatype argument used while declaring array object.

ADD ARRAY ITEMS

The insert() Method

- The array class also defines insert() method. It is possible to insert a new element at the specified index.

Syntax

- array.insert(i, v)

```
import array as arr
a = arr.array('i', [1, 2, 3])
a.insert(1,20)
print (a)
```

Parameters

- i − The index at which new value is to be inserted.
- v − The value to be inserted. Must be of the arraytype.

ADD ARRAY ITEMS

The extend() Method

- The extend() method in array class appends all the elements from another array of same typecode.

Syntax

- array.extend(x)

Parameters

- x – Object of array.array class

```
import array as arr
a = arr.array('i', [1, 2, 3, 4, 5])
b = arr.array('i', [6,7,8,9,10])
a.extend(b)
print (a)
```

- Python – Arrays
- Accessing Array Items
- Add Array Items
- **Remove Array Items**
- Loop Arrays
- Copy Arrays
- Reverse Arrays
- Sort Arrays
- Join Arrays
- Array Methods

PYTHON – ARRAYS

Course Unit 7:
Week 9

REMOVE ARRAY ITEMS

array.remove() Method

- The remove() method removes the first occurrence of a given value from the array

Syntax

- array.remove(v)

Parameters

- v − The value to be removed from the array

```
import array as arr
a = arr.array('i', [1, 2, 1, 4, 2])
a.remove(2)
print (a)
```

REMOVE ARRAY ITEMS

array.pop() Method

- The pop() method removes an element at the specified index from the array, and returns the removed element.

Syntax

- array.pop(i)

```
import array as arr
a = arr.array('i', [1, 2, 1, 4, 2])
a.pop(2)
print (a)
```

Parameters

- i − The index for the eminent to be removed. The method returns element at ith position after removal.

- Python – Arrays
- Accessing Array Items
- Add Array Items
- Remove Array Items
- **Loop Arrays**
- Copy Arrays
- Reverse Arrays
- Sort Arrays
- Join Arrays
- Array Methods

PYTHON – ARRAYS

Course Unit 7:
Week 9

LOOP ARRAYS

"for" Loop with Array

```
import array as arr
a = arr.array('d', [1, 2, 3])
for x in a:
    print (x)
```

```
import array as arr
a = arr.array('d', [1, 2, 3])
l = len(a)
for x in range(l):
    print (a[x])
```

"while" Loop with Array

```
import array as arr
a = arr.array('d', [1, 2, 3])
l = len(a)
idx =0
while idx<l:
    print (a[idx])
    idx+=1
```

- Python – Arrays
- Accessing Array Items
- Add Array Items
- Remove Array Items
- Loop Arrays
- **Copy Arrays**
- Reverse Arrays
- Sort Arrays
- Join Arrays
- Array Methods

PYTHON – ARRAYS

Course Unit 7:
Week 9

COPY ARRAYS

- To create another physical copy of an array, we use another module in Python library, named copy and use **deepcopy()** function in the module. A deep copy constructs a new compound object and then, recursively inserts copies into it of the objects found in the original.

```
import array, copy
a = arr.array('i', [1, 2, 3, 4, 5])
import copy
b = copy.deepcopy(a)
print (id(a), id(b))
a[2]=10
print (a,b)
```

- Python – Arrays
- Accessing Array Items
- Add Array Items
- Remove Array Items
- Loop Arrays
- Copy Arrays
- **Reverse Arrays**
- Sort Arrays
- Join Arrays
- Array Methods

PYTHON – ARRAYS

Course Unit 7:
Week 9

REVERSE ARRAYS

```
import array as arr
a = arr.array('i', [10,5,15,4,6,20,9])
b = arr.array('i')
for i in range(len(a)-1, -1, -1):
   b.append(a[i])
print (a, b)
```

The array class doesn't have any built-in method to reverse array. Hence, we have to use another array. An empty array "b“.

We traverse the numbers in array "a" in reverse order, and append each element to the "b" array. The array "b" now holds numbers from original array in reverse order.

REVERSE ARRAYS

```
from array import array as arr
a = arr.array('i', [10,5,15,4,6,20,9])
b = a.tolist()
b.reverse()
a = arr.array('i')
a.fromlist(b)
print (a)
```

Reverse the sequence of numbers in an array using the reverse() method in list class. List is a built-in type in Python. We have to first transfer the contents of an array to a list with tolist() method of array class. Call the reverse() method. If we now convert the list back to an array, we get the array with reversed order.

- Python – Arrays
- Accessing Array Items
- Add Array Items
- Remove Array Items
- Loop Arrays
- Copy Arrays
- Reverse Arrays
- **Sort Arrays**
- Join Arrays
- Array Methods

PYTHON – ARRAYS

Course Unit 7:
Week 9

SORT ARRAYS

The array class doesn't have any function/method to give a sorted arrangement of its elements. However, we can achieve it with one of the following approaches −

- Using a sorting algorithm
- Using the sort() method from List
- Using the built-in sorted() function

SORT ARRAYS

Using a Sorting Algorithm

- We shall implement the classical bubble sort algorithm to obtain the sorted array. To do it, we use two nested loops and swap the elements for rearranging in sorted order.

```
import array as arr
a = arr.array('i', [10,5,15,4,6,20,9])
for i in range(0, len(a)):
   for j in range(i+1, len(a)):
      if(a[i] > a[j]):
         temp = a[i];
         a[i] = a[j];
         a[j] = temp;
print (a)
```

SORT ARRAYS

Using the sort() Method from List

- Even though array doesn't have a sort() method, Python's built-in List class does have a sort method.

```
from array import array as arr
a = arr.array('i', [10,5,15,4,6,20,9])
b=a.tolist()
b.sort()
a = arr.array('i')
a.fromlist(b)
print (a)
```

SORT ARRAYS

Using the Builtin sorted() Function

- The third technique to sort an array is with the sorted() function, which is a built-in function.

The **syntax** of sorted() function is as follows −

- sorted(iterable, reverse=False)

The function returns a new list containing all items from the iterable in ascending order. Set reverse parameter to True to get a descending order of items.

- The sorted() function can be used along with any iterable. Python array is an iterable as it is an indexed collection. Hence, an array can be used as a parameter to sorted() function.

SORT ARRAYS

Using the Builtin sorted() Function

```
from array import array as arr
a = arr.array('i', [4, 5, 6, 9, 10, 15, 20])
sorted(a)
print (a)
```

- Python – Arrays
- Accessing Array Items
- Add Array Items
- Remove Array Items
- Loop Arrays
- Copy Arrays
- Reverse Arrays
- Sort Arrays
- **Join Arrays**
- Array Methods

PYTHON – ARRAYS

Course Unit 7:
Week 9

JOIN ARRAYS

- **First Method**

To join two arrays, we can do it by **appending** each item from one array to other.

```
import array as arr
a = arr.array('i', [10,5,15,4,6,20,9])
b = arr.array('i', [2,7,8,11,3,10])
for i in range(len(b)):
   a.append(b[i])
print (a, b)
```

JOIN ARRAYS

- **Second Method**

Using another method to join two arrays, first convert arrays to list objects. The list objects can be concatenated with the '+' operator. If "z" list is converted back to array, you get an array that represents the joined arrays

```
from array import array as arr
a = arr.array('i', [10,5,15,4,6,20,9])
b = arr.array('i', [2,7,8,11,3,10])
x=a.tolist()
y=b.tolist()
z=x+y
a=arr.array('i')
a.fromlist(z)
print (a)
```

JOIN ARRAYS

- **Third Method**

We can also use the extend() method from the List class to append elements from one list to another. Convert the array to a list and then call the extend() method to merge the two lists.

```
from array import array as arr
a = arr.array('i', [10,5,15,4,6,20,9])
b = arr.array('i', [2,7,8,11,3,10])
a.extend(b)
print (a)
```

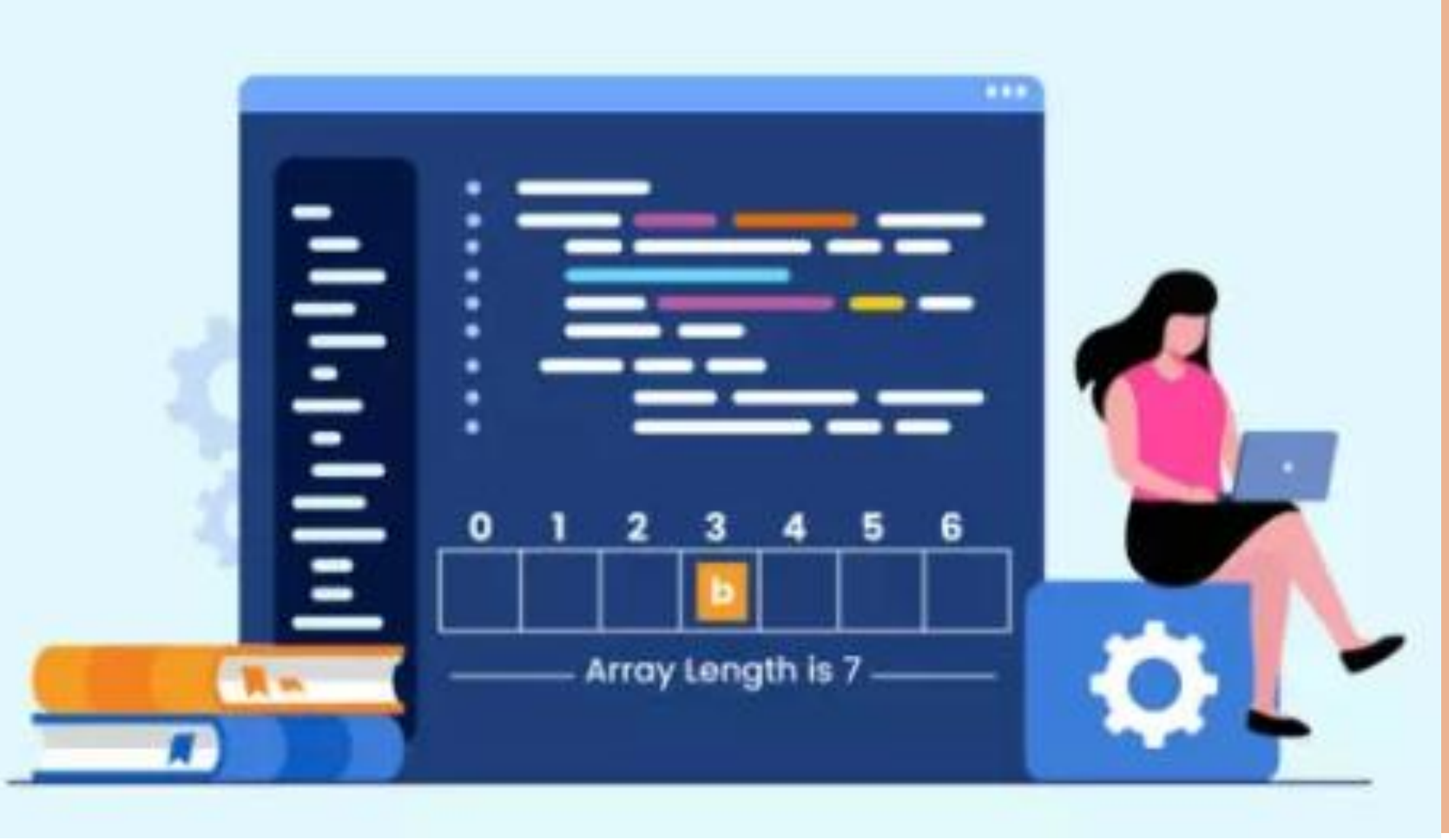
- Python – Arrays
- Accessing Array Items
- Add Array Items
- Remove Array Items
- Loop Arrays
- Copy Arrays
- Reverse Arrays
- Sort Arrays
- Join Arrays
- **Array Methods**

PYTHON – ARRAYS

Course Unit 7:
Week 9

ARRAY METHODS

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the first item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list



References

learn python programming. (2023). https://www.tutorialsteacher.com/python

python tutorial. (2022). https://www.w3resource.com/python/python-tutorial.php

python tutorial. (n.d.). https://www.tutorialspoint.com/python/index.htm

python tutorial. (n.d.). https://www.w3schools.com/python/default.asp