Lecture#10 Python Programming

Python Array – Define, Create

What is Python Array?

A **Python Array** is a collection of common type of data structures having elements with same data type. It is used to store collections of data. In Python programming, an arrays are handled by the "array" module. If you create arrays using the array module, elements of the array must be of the same numeric type.

In this Python Array lecture, you will learn:

- **What is Python Array?**
- **When to use Array in Python?**
- **Syntax to Create an Array in Python**
- **How to create arrays in Python?**
- **How to access array elements?**
- **How to insert elements?**
- **How to modify elements?**
- **How to pop an element from Array in Python?**
- **How to delete elements?**
- **How to Search and get the index of a value in an Array**
- **How to Reverse an Array in Python**
- **Count the occurrence of a Value in Array**
- **Traverse an Array**
- **Summary:**

When to use Array in Python?

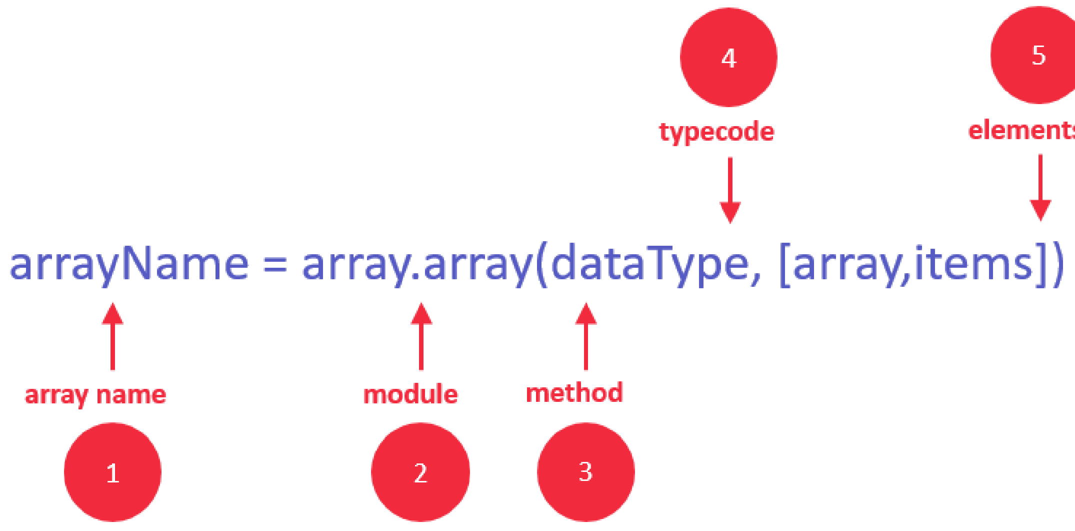
Python arrays are used when you need to use many variables which are of the same type. It can also be used to store a collection of data. The arrays are especially useful when you have to process the data dynamically. Python arrays are much faster than list as it uses less memory.

Syntax to Create an Array in Python

You can declare an array in Python while initializing it using the following syntax.

```
arrayName = array.array(type code for data type, [array,items])
```

The following image explains the syntax.



Array Syntax

1. **Identifier**: specify a name like usually, you do for variables
2. **Module**: Python has a special module for creating array in Python, called "array" – you must import it before using it
3. **Method**: the array module has a method for initializing the array. It takes two arguments, type code, and elements.
4. **Type Code**: specify the data type using the type codes available (see list below)
5. **Elements**: specify the array elements within the square brackets, for example [130,450,103]

How to create arrays in Python?

In Python, we use following syntax to create arrays:

```
Class array.array(type code[,initializer])
```

For Example

```
import array as myarray
abc = myarray.array('d', [2.5, 4.9, 6.7])
```

The above code creates an array having integer type. The letter ‘d’ is a type code.

Following tables show the type codes:

Type code	Python type	C Type	Min size(bytes)
‘u’	Unicode character	Py_UNICODE	2
‘b’	Int	Signed char	1
‘B’	Int	Unsigned char	1
‘h’	Int	Signed short	2
‘l’	Int	Signed long	4
‘L’	Int	Unsigned long	4
‘q’	Int	Signed long long	8
‘Q’	Int	Unsigned long long	8
‘H’	Int	Unsigned short	2
‘f’	Float	Float	4
‘d’	Float	Double	8
‘i’	Int	Signed int	2
‘I’	Int	Unsigned int	2

How to access array elements?

You can access any array item by using its index.

The syntax is

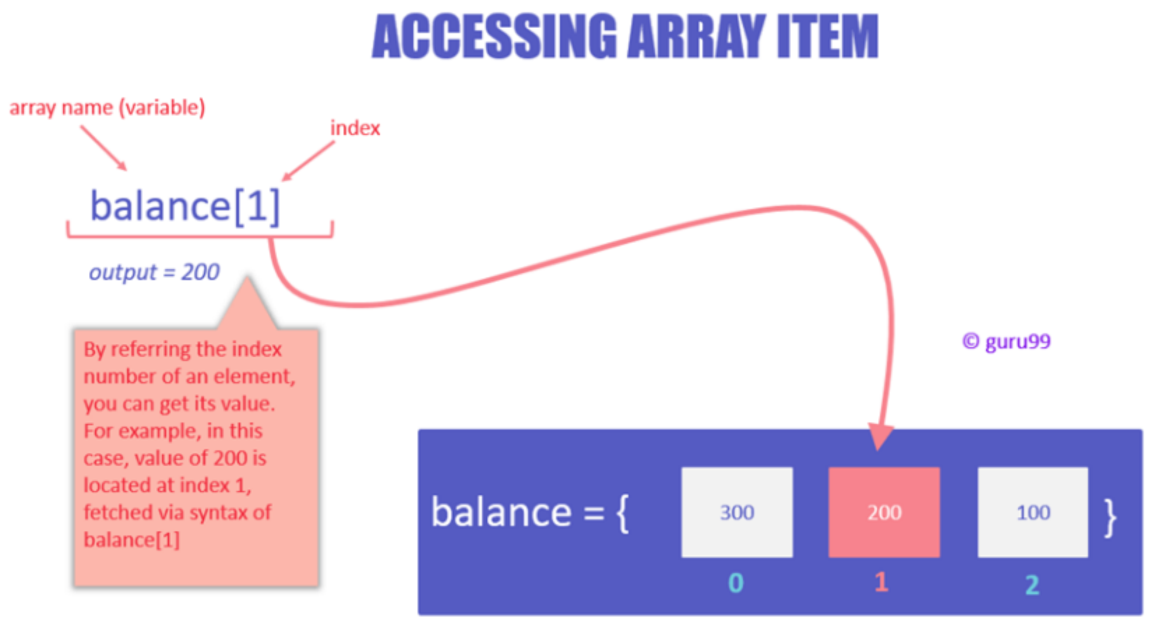
```
arrayName[indexNum]
```

For example,

```
import array
balance = array.array('i', [300,200,100])
print(balance[1])
```

Output:

```
200
```


ACCESSING ARRAY ITEM
array name (variable)
index
balance[1]
output = 200
By referring the index number of an element, you can get its value. For example, in this case, value of 200 is located at index 1, fetched via syntax of balance[1]
© guru99
balance = {
300
200
100
}
0
1
2

Accessing Array Item

Here, we have accessed the second value of the array using its index, which is 1. The output of this will be 200, which is basically the second value of the balanced array.

The array index starts with 0. You can also access the last element of an array using the -1 index.

Example:

```
import array as myarray
abc = myarray.array('d', [2.5, 4.9, 6.7])
print("Array first element is:",abc[0])
print("Array last element is:",abc[-1])
```

Output:

```
Array first element is: 2.5
Array last element is: 6.7
```

You can also access elements by using the ':' operator as shown in below Python arrays examples.

Example:

```
import array as myarray
abc= myarray.array('q',[3,9,6,5,20,13,19,22,30,25])
print(abc[1:4])
print(abc[7:10])
```

Output:

```
array('q', [9, 6, 5])
array('q', [22, 30, 25])
```

This operation is called a **slicing** operation.

How to insert elements?

Python array insert operation enables you to insert one or more items into an array at the beginning, end, or any given index of the array. This method expects two arguments index and value.

The syntax is

```
arrayName.insert(index, value)
```

Example:

Let’s add a new value right after the second item of the array. Currently, our balance array has three items 300, 200, and 100. Consider the second array item with a value of 200 and index 1.

In order to insert the new value right “after” index 1, you need to reference index 2 in your insert method, as shown in the below Python array example:

```
import array
balance = array.array('i', [300,200,100])
balance.insert(2, 150)
print(balance)
```

Output:

```
array('i', [300,200,150,100])
```

Example 2:

```
import array as myarr
a=myarr.array('b',[2,4,6,8,10,12,14,16,18,20])
a.insert(2,56)
print(a)
```

Output:

```
array('b', [2, 4, 56, 6, 8, 10, 12, 14, 16, 18, 20])
```

How to modify elements?

Arrays in Python are mutable. They can be modified by the following syntax:

```
Object_name[index]=value;
```

Example:

```
import array as myarr
a=myarr.array('b',[3,6,4,8,10,12,14,16,18,20])
a[0]=99
print(a)
```

Output:

```
array('b', [99, 6, 4, 8, 10, 12, 14, 16, 18, 20])
```

We can also perform concatenation operations on arrays in Python.

Example:

```
import array as myarr
first = myarr.array('b', [4, 6, 8])
second = myarr.array('b', [9, 12, 15])
numbers = myarr.array('b')
numbers = first + second
print(numbers)
```

Output

```
array('b', [4, 6, 8, 9, 12, 15])
```

The above Python array example code concates two variables called "first" and "second". The result is stored in a variable called "number".

The last line of code is used to print two arrays.

How to pop an element from Array in Python?

In Python, a developer can use pop() method to pop and element from Python array. Below is an example of pop() method in Python.

Python array pop Example:

```
import array as myarr
first = myarr.array('b', [20, 25, 30])
first.pop(2)
print(first)
```

Output

```
array('b', [20, 25])
```

You can also use the ‘del’ statement of Python.

Example

```
import array as myarr
no = myarr.array('b', [10, 4, 5, 5, 7])
del no[4]
print(no)
```

Output:

```
array('b', [10, 4, 5, 5])
```

How to delete elements?

With this operation, you can delete one item from an array by value. This method accepts only one argument, value. After running this method, the array items are re-arranged, and indices are re-assigned.

The syntax is

```
arrayName.remove(value)
```

Example:

Let’s remove the value of “3” from the array

```
import array as myarray
first = myarray.array('b', [2, 3, 4])
first.remove(3)
print(first)
```

Output:

```
array('b', [2, 4])
```

How to Search and get the index of a value in an Array

With this operation, you can search for an item in an array based on its value. This method accepts only one argument, value. It is a non-destructive method, which means it does not affect the array values.

The syntax is

```
arrayName.index(value)
```

Example:

Let’s find the value of “3” in the array. This method returns the index of the searched value.

```
import array as myarray
number = myarray.array('b', [2, 3, 4, 5, 6])
print(number.index(3))
```

Output:

```
1
```

This operation will return the index of the first occurrence of the mentioned element.

How to Reverse an Array in Python

This operation will reverse the entire array.

Syntax: array.reverse()

```
import array as myarray
number = myarray.array('b', [1,2, 3])
number.reverse()
print(number)
```

Output:

```
array('b', [3, 2, 1])
```

Count the occurrence of a Value in Array

You can also count the occurrence of elements in the array using the array.count(x) syntax.

Example:

```
import array as myarr
number = myarr.array('b', [2, 3, 5, 4,3,3,3])
print(number.count(3))
```

Output

```
4
```

Traverse an Array

You can traverse a Python array by using loops, like this one:

```
import array
balance = array.array('i', [300,200,100])
for x in balance:
        print(x)
```

Output:

```
200
300
100
```

Summary:

- An array is a common type of data structure wherein all elements must be of the same data type.
- Python programming, an array, can be handled by the "array" module.

- Python arrays are used when you need to use many variables which are of the same type.
- In Python, array elements are accessed via indices.
- Array elements can be inserted using an array.insert(i,x) syntax.
- In Python, arrays are mutable.
- In Python, a developer can use pop() method to pop and element from Python array.
- Python array can be converted to Unicode. To fulfill this need, the array must be a type ‘u’; otherwise, you will get “ValueError”.
- Python arrays are different from lists.
- You can access any array item by using its index.
- The array module of Python has separate functions for performing array operations.