

Python Arrays

Introduction:

In this article, we are discussing Arrays in Python. The Array is used in every programming language, like C, C++, Java, Python, R, JavaScript, etc. By using an array, we can store more than one data. The Array is a process of memory allocation. It is performed as a dynamic memory allocation. We can declare an array like `x[100]`, storing 100 data in x. It is a container that can hold a fixed number of items, and these items should be the same type. An array is popular in most programming languages like C/C++, JavaScript, etc.

The Array is an idea of storing multiple items of the same type together, making it easier to calculate the position of each element by simply adding an offset to the base value. A combination of the arrays could save a lot of time by reducing the overall size of the code. It is used to store multiple values in a single variable. If you have a list of items that are stored in their corresponding variables like this:

1. `car1 = "Lamborghini"`
2. `car2 = "Bugatti"`
3. `car3 = "Koenigsegg"`

If you want to loop through cars and find a specific one, you can use the Array. You can use an array to store more than one item in a specific variable.

The Array can be handled in Python by a module named **Array**. It is useful when we must manipulate only specific data values. The following are the terms to understand the concept of an array:

Element - Each item stored in an array is called an element.

Index - The location of an element in an array has a numerical index, which is used to identify the element's position. The index value is very much important in an Array.

Array Representation:

An array can be declared in various ways and in different languages. The important points that should be considered are as follows:

1. The index starts with 0.
2. We can easily find any elements within this Array using the Index value.
3. The length of the Array defines the capacity to store the elements. It is written like `x[100]`, which means the length of array x is specified by 100.

Array operations

Some of the basic operations supported by an array are as follows:

- **Traverse** - It prints all the elements one by one.

- **Insertion** - It adds an element at the given index.
- **Deletion** - It deletes an element at the given index.
- **Search** - It searches an element using the given index or by the value.
- **Update** - It updates an element at the given index.

The Array can be created in Python by importing the array module to the python program.

1. from array **import** *
2. arrayName = array(typecode, [initializers])

Accessing array elements

We can access the array elements using the respective indices of those elements.

Program code:

Here we give an example of how we access the elements of an array using its index value in Python. The code is given below -

1. **import** array as arr
2. a = arr.array('i', [2, 4, 5, 6])
3. **print**("First element is:", a[0])
4. **print**("Second element is:", a[1])
5. **print**("Third element is:", a[2])
6. **print**("Forth element is:", a[3])
7. **print**("last element is:", a[-1])
8. **print**("Second last element is:", a[-2])
9. **print**("Third last element is:", a[-3])
10. **print**("Forth last element is:", a[-4])
11. **print**(a[0], a[1], a[2], a[3], a[-1],a[-2],a[-3],a[-4])

Output:

Now we compile the above code in python, and after successful compilation, we run it. Then the output is given below -

```
First element is: 2
Second element is: 4
Third element is: 5
Forth element is: 6
last element is: 6
Second last element is: 5
Third last element is: 4
Forth last element is: 2
2 4 5 6 6 5 4 2
```

Explanation:

In the above example, we have imported an array, defined a variable named "a" that holds the elements of an array, and printed the elements by accessing elements through the indices of an

array. Here we can easily find out the array element by using the array index like a[0], a[1], a[-1], and so on.

How to change or add elements?

Arrays are mutable, and their elements can be changed similarly to lists.

Program code:

Here in this example, we can change or add or replace any element from the Array in Python. The code is given below -

```
1. import array as arr
2. numbers = arr.array('i', [1, 2, 3, 5, 7, 10])
3.
4. # changing first element 1 by the value 0.
5. numbers[0] = 0
6. print(numbers)      # Output: array('i', [0, 2, 3, 5, 7, 10])
7.
8. # changing last element 10 by the value 8.
9. numbers[5] = 8
10. print(numbers)     # Output: array('i', [0, 2, 3, 5, 7, 10])
11.
12. # replace the value of 3rd to 5th element by 4, 6 and 8
13. numbers[2:5] = arr.array('i', [4, 6, 8])
14. print(numbers)     # Output: array('i', [0, 2, 4, 6, 8, 10])
```

Output:

Now we compile the above code in python, and after successful compilation, we run it. Then the output is given below -

```
array('i', [0, 2, 3, 5, 7, 10])
array('i', [0, 2, 3, 5, 7, 8])
array('i', [0, 2, 4, 6, 8, 8])
```

Explanation:

In the above example, we have imported an array and defined a variable named "numbers," which holds the value of an array. If we want to change or add the elements in an array, we can do it by defining the index of an array where we want to change or add the elements. Here we just mentioned the index number of elements you want to change and declared the new value by which you want to replace the old elements.

Why use Arrays in Python?

A combination of arrays saves a lot of time. The Array can reduce the overall size of the code. Using an array, we can solve a problem quickly in any language. The Array is used for dynamic memory allocation.

How to Delete Elements from an Array?

The elements can be deleted from an array using Python's **del** statement. If we want to delete any value from the Array, we can use the indices of a particular element.

1. **import** array as arr
2. number = arr.array('i', [1, 2, 3, 3, 4])
3. del number[2] # removing third element
4. print(number) # Output: array('i', [1, 2, 3, 4])

Output:

Now we compile the above code in python, and after successful compilation, we run it. Then the output is given below -

```
array('i', [10, 20, 40, 60])
```

Explanation: In the above example, we have imported an array and defined a variable named as "number" which stores the values of an array. Here, by using del statement, we are removing the third element [3] of the given array.

Finding the length of an array

Advertisement

The length of an array is defined as the number of elements present in an array. It returns an integer value that is equal to the total number of the elements present in that array.

Syntax

By using the syntax below, we can easily find the length of the given Array. The syntax is -

1. len(array_name)

Array Concatenation

We can easily concatenate any two arrays using the + symbol.

Example 1:

1. a=arr.array('d',[1.1 , 2.1 ,3.1,2.6,7.8])
2. b=arr.array('d',[3.7,8.6])
3. c=arr.array('d')
4. c=a+b
5. print("Array c = ",c)

Output:

Now we compile the above code in python, and after successful compilation, we run it. Then the output is given below -

```
Array c= array('d', [1.1, 2.1, 3.1, 2.6, 7.8, 3.7, 8.6])
```

Explanation

In the above example, we have defined variables named as "a, b, c" that hold the values of an array.

Example 2:

1. **import** array as arr
2. `x = arr.array('i', [4, 7, 19, 22])` # Initialize the array elements
3. `print("First element:", x[0])`
4. `print("Second element:", x[1])`
5. `print("Second last element:", x[-1])`

Output:

First element: 4

Second element: 7

Second last element: 22

Now we compile the above code in python, and after successful compilation, we run it. Then the output is given below -

Explanation:

In the above example, first, we imported an array and defined a variable named "x," which holds the value of an array. Then, we printed the elements of an array using the index value of this array element.