Arrays

In this lesson, we learn about arrays and how they are used in Python!

We'll cover the following

Introduction

In Python, an array is just an ordered sequence of *homogeneous* elements. In other words, an array can only hold elements of one datatype. Python arrays are basically just wrappers for C arrays. The type is constrained and specified at the time of creation.

Initializing Arrays

Python arrays are initialized using the array library:

```
import array
new_array = array.array('type', [list])
```

Here `type` defines the data type of array and `list` is a python list containing homogenous elements.

Consider the example below:

```
1  import array
2
3  # type: 'd' (float), initializer list: [1, 2, 3]
4  new_array = array.array('d', [1, 2, 3])
5  print(new_array[0])
```

Output 0.152s

```
1.0
```

Line 4 creates an array. Here `d` indicates that the array is of type `float`.

Types of Arrays

There are several types of arrays in Python; refer to the table below for a complete list.

Type code	C Type	Python Type	Minimum Size in Bytes
`'c'`	character	char	1
`'b'`	signed char	int	1
`'B'`	unsigned char	int	1
`'u'`	Py_UNICODE	Unicode character	2 or 4 depending on Unicode build
`'h'`	signed short	int	2
`'H'`	unsigned short	int	2
`'i'`	signed int	int	2
`'I'`	unsigned int	long	4
`'l'`	signed long	int	4
`'L'`	unsigned long	long	4
`'f'`	float	float	4
`'d'`	double	float	8

Array slicing

Array slicing is done in exactly the same way as list slicing is done. Look at the following example:

```
import array

initializer_list = [2, 5, 43, 5, 10, 52, 29, 5]
number_array = array.array('i', initializer_list)

print(number_array[1:5])  # 2nd to 5th
print(number_array[:-5])  # beginning to 3rd
print(number_array[5:])  # 6th to end
print(number_array[:])   # beginning to end

```

```
Output                                                    0.160s

 array('i', [5, 43, 5, 10])
 array('i', [2, 5, 43])
 array('i', [52, 29, 5])
 array('i', [2, 5, 43, 5, 10, 52, 29, 5])
```

Changing or adding array elements

Arrays are mutable; their elements can be changed in the same way as list elements. Have a look at the following coding widget.

```
 1  import array
 2  integers = array.array('i', [1, 2, 3, 5, 7, 10])
 3
 4  # changing first element
 5  integers[0] = 0
 6  print(integers)  # array('i', [0, 2, 3, 5, 7, 10])
 7
 8  # changing 3rd to 5th element
 9  integers[2:5] = array.array('i', [4, 6, 8])
10  print(integers)  # Output: array('i', [0, 2, 4, 6, 8, 10])
11
```

```
Output                                                    0.176s

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

Just as with lists, we can add one item to the end of an array using the `append()` method or add several items using the `extend()` method.

```
 1  import array
 2
 3  numbers = array.array('i', [1, 2, 3])
 4
 5  numbers.append(4)
 6  print(numbers)  # array('i', [1, 2, 3, 4])
 7
 8  # extend() appends iterable to the end of the array
 9  numbers.extend([5, 6, 7])
10  print(numbers)  # array('i', [1, 2, 3, 4, 5, 6, 7])
11
```

```
Output                                                    0.237s

 array('i', [1, 2, 3, 4])
 array('i', [1, 2, 3, 4, 5, 6, 7])
```

You can concatenate two arrays using + operator

```
 1  import array
 2
 3  odd = array.array('i', [1, 3, 5])
 4  even = array.array('i', [2, 4, 6])
 5
 6  integers = array.array('i')   # creating empty array of integer
 7  integers = odd + even
 8
 9  print(integers)
10
```

```
Output                                                    0.159s

 array('i', [1, 3, 5, 2, 4, 6])
```

How do you remove/delete elements?

To delete one or more items from an array, use the `del` statement as with lists.

```
 1  import array
 2
 3  integer_array = array.array('i', [1, 2, 3, 3, 4])
 4
 5  del integer_array[2]  # removing third element
 6  print(integer_array)  # Output: array('i', [1, 2, 3, 4])
 7
 8  del integer_array  # deleting entire array
 9  print(integer_array)  # Error: array is not defined
10
```

```
Output                                                    0.164s

 array('i', [1, 2, 3, 4])

Traceback (most recent call last):
  File "main.py", line 9, in <module>
    print(integer_array)  # Error: array is not defined
NameError: name 'integer_array' is not defined
```

We can use the `remove(val)` method to remove the given item or `pop(index)` to remove an item at the given index. The `remove(val)` method removes the first element that is equal to `val` in the array.

> **Note:** An error is thrown if the index exceeds the size of the array or element is not found in the array.

```
import array

integer_array = array.array('i', [10, 11, 12, 12, 13])

integer_array.remove(12)
print(integer_array)   # array('i', [10, 11, 12, 13])

print(integer_array.pop(2))   # Output: 12
print(integer_array)   # array('i', [10, 11, 13])

```

Output 0.700s

```
array('i', [10, 11, 12, 13])
12
array('i', [10, 11, 13])
```

In the next lesson, we'll be looking at the differences between arrays and lists in Python

Back

Lists

Next

Lists vs Arrays in Python

Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/arrays__introduction-to-lists__data-structures-for-coding-interviews-in-python)