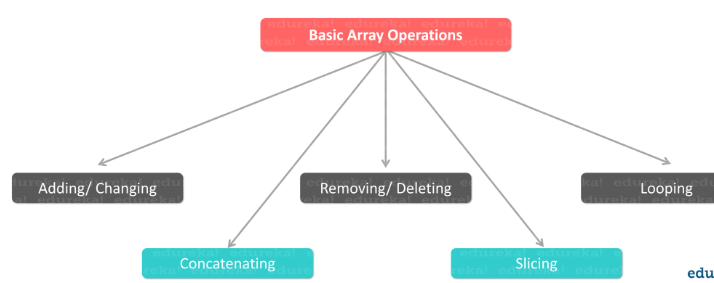
**Array using python**

Arrays are used to store multiple values in one individual variable. An array is a data structure that stores values of same data type. In Python, this is the main difference between arrays and lists. While python lists can contain values comparable to different data types, arrays in python can only contain values corresponding to same data type.

Python provides several different options for storing efficient, fixed-type data. Python has a built-in array module called “**array**” which is used to create arrays of uniform type. This was its main disadvantage

**Basic array operations**:

There are many operations that can be performed on arrays which are as follows

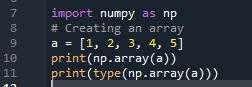


**np.array():**

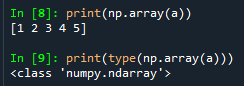
We can use [**np.array**](https://docs.scipy.org/doc/numpy/reference/generated/numpy.array.html)method to create arrays from python list

**Python code:**

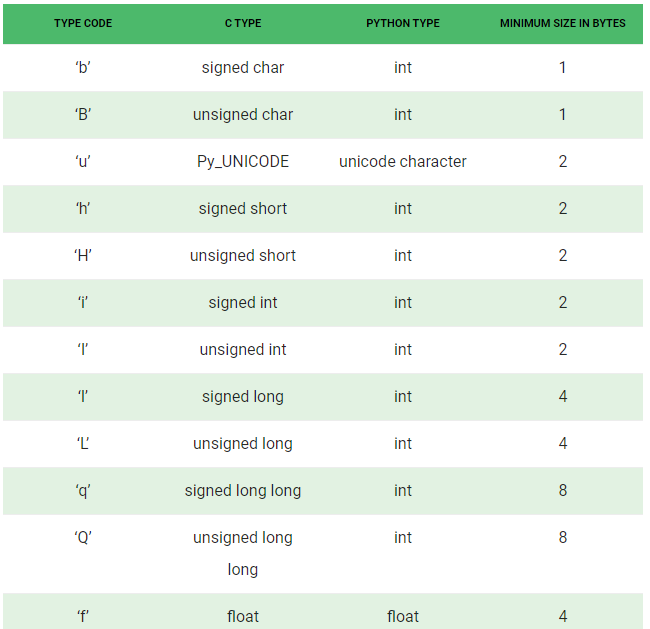
**Package:** import numpy as np



Output:



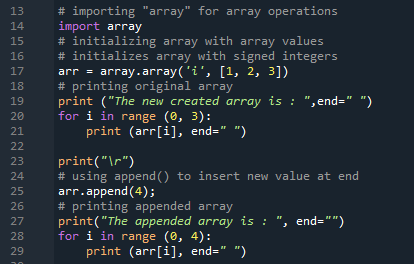
**array(data type, value list)** :- This function is used to**create**an array with data type and value list specified in its arguments. Some of the data types are mentioned in the table below.



**array.append()**:- This function is used to**add the value** mentioned in its arguments at the **end** of the array.

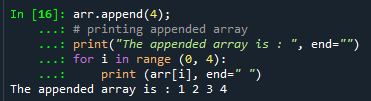
**Python code:**

Package: import array



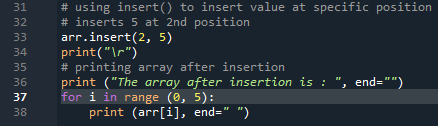
Output:

Using append() function to insert new values at end

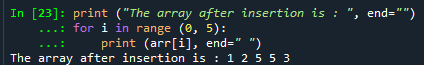


**array.insert()** :- This function is used to**add the value at the position**specified in its argument.

Python code: using insert() function to insert value at specific position

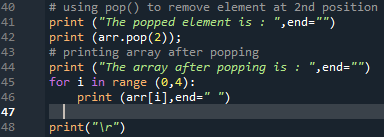


Output: The index position starts from ‘zero’ in python

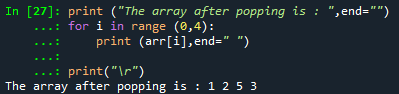


**array.pop()**:- This function**removes the element at the position**mentioned in its argument, and returns it.

**Python code:**

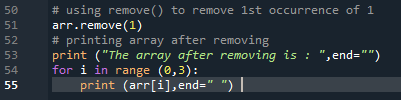


**Output: The below output shows array after popping**

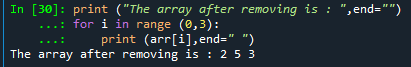


**array.remove()** :- This function is used to**remove the first occurrence**of the value mentioned in its arguments.

Python code:



Output: below output shows the array after removing the 1st index value



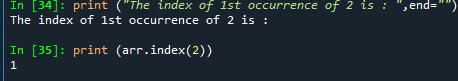
**array.index()** :- This function returns the**index of the first occurrence**of value mentioned in arguments.

Python code:



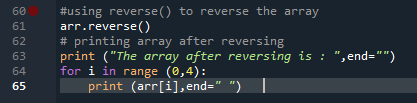
Output:

Below output shows the index of first occurrence of 2



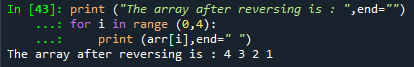
**array.reverse()** :- This function**reverses** the array.

Python code:



Output:

The below output shows array after reversing the values



**array.count(x):** This function return the number of occurrences of x in the array.

Python code:



Output:

The below output shows the value 2 is occurred one time in the array



**array**.**byteswap**(): This function is only supported for values which are 1, 2, 4, or 8 bytes in size; for other types of values, [RuntimeError](https://docs.python.org/3/library/exceptions.html#RuntimeError) is raised. It is helpful when reading data from a file written on a machine with a different byte order.

**array.extend(*iterable*):** This function isappend items from iterable to the end of the array. If iterable is another array, it must have absolutely the same type code; if not, [TypeError](https://docs.python.org/3/library/exceptions.html#TypeError) will be raised. If iterable is not an array, it must be iterable and its elements must be the right type to be appended to the array.