**Array**

array\_1 = arr.array("i", [3, 6, 9, 12])

import array

array\_1 = array.array("i", [3, 6, 9, 12])

for i in range (0, 4):

print("array\_1[", i, "]", array\_1[i])

or

Integer:

a = [0] \* (6)

for i in range (0, 6):

print("a[", i, "]", a[i])

String:

strs = ['']\*size

or

strs = ["" for x in range(size)]

| Type Code |  | Python Type | Minimum Sizes in Bytes |
| --- | --- | --- | --- |
|  |  |  |  |
| ‘b’ |  | int | 1 |
| ‘B’ |  | int | 1 |
| ‘u’ |  | unicode characters | 2 |
| ‘h’ |  | int | 2 |
| ‘H’ |  | int | 2 |
| ‘i’ |  | int | 2 |
| ‘I’ |  | long | 2 |
| ‘l’ |  | int | 4 |
| ‘L’ |  | long | 4 |
| ‘f’ |  | float | 4 |
| ‘d’ |  | float | 8 |

**List**

A list is a data structure that's built into Python and holds a collection of items. Lists have a number of important characteristics:

* List items are enclosed in square brackets, like this [item1, item2, item3].
* Lists are ordered – i.e. the items in the list appear in a specific order. This enables us to use an index to access to any item.
* Lists are mutable, which means you can add or remove items after a list's creation.
* List elements do not need to be unique. Item duplication is possible, as each element has its own distinct place and can be accessed separately through the index.
* Elements can be of different data types: you can combine strings, integers, and objects in the same list.

Lists are very easily created in Python:

|  |
| --- |
| list = [3, 6, 9, 12]  print(list)  print(type(list)) |

Or

>>> temp\_list = []

>>> print temp\_list

[]

>>>

>>> temp\_list.append("one")

>>> temp\_list.append("two")

>>> print temp\_list

['one', 'two']

>>>

>>> temp\_list.append("three")

>>> print temp\_list

['one', 'two', 'three']

>>>

**The differences between Array and List**

* Arrays need to be declared. Lists don't, since they are built into Python. In the examples above, you saw that lists are created by simply enclosing a sequence of elements into square brackets. Creating an array, on the other hand, requires a specific function from either the array module (i.e., array.array()) or NumPy package (i.e., numpy.array()). Because of this, lists are used more often than arrays.
* Arrays can store data very compactly and are more efficient for storing large amounts of data.
* Arrays are great for numerical operations; lists cannot directly handle math operations. For example, you can divide each element of an array by the same number with just one line of code. If you try the same with a list, you'll get an error.

**Tuple**

A tuple is a collection of objects which ordered and immutable. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.

Creating a tuple is as simple as putting different comma-separated values. Optionally you can put these comma-separated values between parentheses also. For example −

tup1 = ('physics', 'chemistry', 1997, 2000);

tup2 = (1, 2, 3, 4, 5 );

tup3 = "a", "b", "c", "d";

The empty tuple is written as two parentheses containing nothing:

tup1 = ();

**Dictionary:**

In Python, curly braces are used to define a data structure called a dictionary (a key/value mapping). Dictionaries in Python are data structures that store key-value pairs. You can use them like associative arrays. Curly braces are used when declaring dictionaries:

d = {'One': 1, 'Two' : 2, 'Three' : 3 }

print d['Two'] # prints "2"