**Array. Dynamic array**

Модуль array визначає масиви в python. Від списків масиви відрізняються обмеженням на тип даних та розмір кожного елемента.

| **Код типу** | **Тип в C** | **Тип в python** | **Мінімальний розмір в байтах** |
| --- | --- | --- | --- |
| 'b' | signed char | int | 1 |
| 'B' | unsigned char | int | 1 |
| 'h' | signed short | int | 2 |
| 'H' | unsigned short | int | 2 |
| 'i' | signed int | int | 2 |
| 'I' | unsigned int | int | 2 |
| 'l' | signed long | int | 4 |
| 'L' | unsigned long | int | 4 |
| 'q' | signed long long | int | 8 |
| 'Q' | unsigned long long | int | 8 |
| 'f' | float | float | 4 |
| 'd' | double | float | 8 |

Розмір та тип елемента в масиві визначається при його створенні та може приймати наступні значення:

Клас array.array(TypeCode [, ініціалізатор]) - новий масив, елементи якого обмежені TypeCode, і ініціалізатор, який має бути списком, об’єктом, підтримуючим інтерфейс буфера, або ітерований об’єкт.

array.typecodes - стрічка, де є всі типи в масиві.

Масиви можна змінювати. Масиви підтримують всі методи списків (індексація, зрізи, множення, ітерація) та інші.

**Деякі методи масивів (array) в python**

*array.itemsize* – розмір в байтах одного елементу в масиві;

*array.append(х)* – додавання елементу в кінець масиву;

*array.count(х)* – повертає кількість входжень х в масив;

*array.extend(iter)* – додавання елементів із об’єкту в масив;

*array.index(х)* - номер першого входження x в масив;

*array.insert(n, х)* - включити новий пункт зі значенням х в масиві перед номером n. Від’ємні значення беруться відносно кінця масиву;

*array.pop(i)* – видаляє і-й елемент з масиву та повертає його. За замовчанням видаляється останній елемент;

*array.remove(х)* -видаляє перше входження х з масиву;

*array.reverse()* – обернений порядок елементів в масиві;

*array.tobytes()* – перетворення в байти;

*array.tofile(f)* - запис масиву в відкритий файл;

*array.tolist()* - перетворення масиву в список.

**Dynamic array**

Динамічні масиви – продовження масивів. Вони можуть змінювати розмір під час роботи програми. Це дає перевагу у випадку, коли кінцева кількість елементів, які мають зберігатись у масиві, невідома.

Динамічні масиви переважно ініціалізуються з кількістю елементів вдвічі більшу за початкову. За рахунок цього є можливість додавати нові елементи. Розмір зберігається, поки його вистачає для елементів, які додаються.

Коли масив збільшується в розмірі, то спочатку початковий масив копіюється, а потім в новому місці створюєтесь новий масив з вдвічі більшим розміром і зберігає початковий масив як першу половину нового.

|  |  |
| --- | --- |
| **Операція** | **Швидкість виконання** |
| Індексація | Θ(1) |
| Додати/видалити на початку | Θ(n) |
| Додати/видалити в кінці | Θ(1) |
| Додати/видалити в середині | Θ(n) |
| Витрачене місце (в середньому) | Θ(n) |