



ХОЧУ ПОМОЧЬ
ПРОЕКТУ

eco-e.ru

РЕКЛАМА

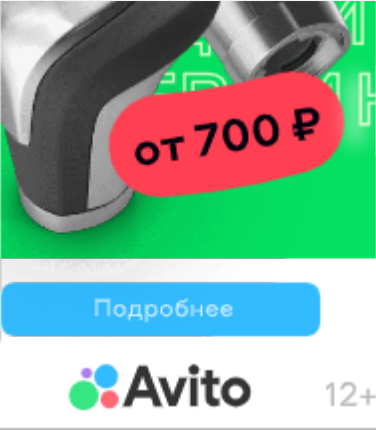


Аргус07 за 50100р? Выгодней еЛайт01

47 900 Р

Узнать больше

Ст
Мо
це
Ма
об
си



/ Массивы array в Python

бъекта, который может компактно представлять массив значений базовых типов: [символы](#), [и запятой](#).

и ведут себя очень похоже на [списки](#), за исключением того, что тип хранящихся в них
вадается во время создания объекта с помощью [кода типа](#), который представляет собой один
типов и соответствующие им коды смотрите в разделе "[Коды типов массива array\(\)](#)".

Фактическое представление значений array определяется архитектурой машины, строго говоря реализацией языка C. К фактическому размеру можно получить доступ через атрибут [array.itemsize](#).

Модуль array определяет следующий тип:

array.array(typecode[, initializer]):

Класс array.array() создает новый массив, элементы которого ограничены [typecode](#) и инициализируется из необязательного значения initializer, которое должно быть [списком](#), [байтоподобным объектом](#) или [итерируемым](#) по элементам объектом соответствующего типа.

Если задан [список](#) или [строка](#), то инициализатор передается методу [array.fromlist\(\)](#), [array.frombytes\(\)](#) или [array.fromunicode\(\)](#) нового массива для добавления начальных элементов в массив. В противном случае итеративный инициализатор передается в метод [array.extended\(\)](#).

Создание экземпляра [array.array\(\)](#) вызывает [событие аудита](#) array.__new__ с аргументами typecode, initializer.

Смотрите какие методы определяет [класс array.array\(\)](#) в разделе "[Методы и свойства класса array\(\)](#)".

Объекты класса array.array() поддерживают [общие операции с последовательностями](#), такие как индексация, срез, объединение и т. д. При [использовании среза](#), назначенное значение должно быть объектом массива с тем же [typecode](#). Во всех остальных случаях вызывается [исключение TypeError](#).

Объекты массива также реализуют интерфейс буфера и могут использоваться везде, где поддерживаются [байтообразные](#) объекты.

Когда объект класса array.array() [выводится на печать](#) или [преобразуется в строку](#), он представляется как array(typecode, initializer).

- Инициализатор initializer опускается, если массив пустой.
- Если тип [typecode](#) равен 'u', то это строка.
- Во всех остальных случаях это список чисел.

Строка гарантированно может быть преобразована обратно в массив с тем же типом и значением, используя [функцию eval\(\)](#) если класс array() был импортирован с использованием конструкции from array import array.

Примеры использования:

```
>>> from array import array
>>> arr = array('l')
>>> arr
# 1')
```

```
>>> arr = array('u', 'hello \u2641')
>>> arr
# array('u', 'hello ð')

>>> РЕКЛАМА arr = array('b', b'is array')
>>> arr
# array('b', [105, 115, 32, 97, 114, 114, 97, 121])

>>> arr = array('l', [1, 2, 3, 4, 5])
>>> arr
# array('l', [1, 2, 3, 4, 5])

>>> arr = array('d', [1.0, 2.0, 3.14])
>>> arr
# array('d', [1.0, 2.0, 3.14])

>>> str(arr)
# "array('d', [1.0, 2.0, 3.14])"

>>> print(arr)
# array('d', [1.0, 2.0, 3.14])
```

Смотрите так же [модуль struct](#), при помощи которого можно осуществлять упаковку и распаковку разнородных двоичных данных.

Содержание раздела:

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- [Типы данных массива array\(\), модуль array](#)
- [Методы и элементы данных класса array\(\)](#)
- [Массив array\(\), извлечение и сохранение в файл](#)
- [Операции с объектом array\(\) модуля array](#)