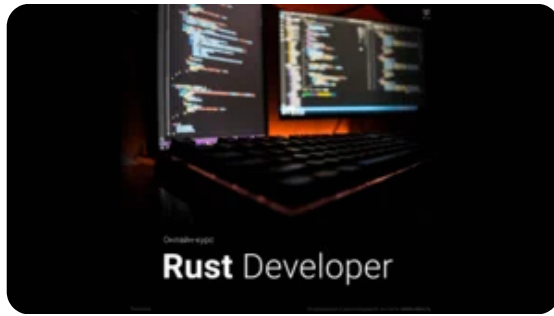


Контейнерные типы данных модуля collections



otus.ru

РЕКЛАМА

Продвинутый онлайн курс «Rust Developer»

Углублённый онлайн-курс по языку программирования Rust. Спец. цена для сдавших тест.

[Узнать больше](#)[Стандартная библиотека Python3.](#) / Контейнерные типы данных модуля collections

Модуль `collections` реализует специализированные типы данных контейнеров, предоставляя альтернативы встроенным контейнерам Python общего назначения `dict`, `list`, `set` и `tuple`.

[Класс ChainMap\(\) модуля collections в Python](#)

Класс `ChainMap()` предназначен для быстрого объединения нескольких словарей, чтобы их можно было рассматривать как единое целое. Такой контейнер объединяет словари и ищет ключи намного быстрее, чем создание нового словаря и выполнение объединения при помощи вызовов `dict.update()`.

[Класс deque\(\) модуля collections в Python](#)

Двусторонняя очередь `deque()` поддерживает поточно-ориентированные, эффективные по памяти операции добавления и извлечения элементов последовательности с любой стороны с примерно одинаковой производительностью $O(1)$ в любом направлении.

[Класс Counter\(\) модуля collections в Python](#)

Для удобных и быстрых подсчетов в последовательностях предусмотрен класс `collections.Counter()`. Это коллекция, в которой элементы хранятся в виде словарных ключей, а их счетчики хранятся в виде значений словаря.

[Класс defaultdict\(\) модуля collections в Python](#)

Класс `defaultdict()` модуля `collections` ничем не отличается от обычного словаря за исключением того, что по умолчанию всегда вызывается функция, которая возвращает значение по умолчанию для новых значений.

[Класс namedtuple\(\) модуля collections в Python](#)

Именованные кортежи присваивают имя каждому значению элемента в кортеже и тем самым создают более читаемый код. Они могут использоваться везде, где используются обычные кортежи и добавляют возможность доступа к полям по имени. Они легковесны и требуют не больше памяти, чем обычные кортежи.

[Класс OrderedDict\(\) модуля collections в Python](#)

Функция `OrderedDict()` модуля `collections` возвращает экземпляр подкласса `dict`, у которого есть методы, специализированные для изменения порядка словаря.

[Класс UserString\(\) модуля collections в Python](#)

Класс `collections.UserString()` был введен в Python, когда было невозможно наследоваться от встроенного типа `str` напрямую. Возможность прямого наследования от `str` означает, что нужда в `UserString` отпадает. Но этот класс по-прежнему доступен в стандартной библиотеке как для удобства, так и для обратн



[Класс UserDict модуля collections в Python](#)

Класс UserDict() модуля collections это удобная обертка для обычного объекта dict. Этот класс обеспечивает то же поведение, что и встроенный тип dict, с дополнительной возможностью предоставления доступа к базовому словарю через атрибут экземпляра UserDict.data.

[Класс UserList модуля collections в Python](#)

Класс UserList() модуля collections действует как обертка для объектов списка list. Это полезный базовый класс для собственных классов, подобных спискам, которые могут наследоваться от них и переопределять существующие методы или добавлять новые.

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

РЕКЛАМА · 16+ я

мы научим вас:

€

practicum.yandex.ru

Курс «Системный аналитик» с нуля от Яндекса

5,0 ★ Рейтинг организации ⓘ

За 8 мес. вы освоите инструменты и навыки, которые требуются системному аналитику.

О профессии Системный аналитик

Трудоустройство

Узнать больше

DOCS-Python.ru™, 2023 г.

(Внимание! При копировании материала ссылка на источник обязательна)

@docs_python_ru

https://docs-python.ru/standart-library/modul-collections-python/

2/2