

# Модуль functools в Python, инструменты расширения функций

Яндекс Взгляд · Опрос

Выберите 1 или несколько ответов

Какие сервисы проверки истории автомобилей вы знаете?

Avtocod

ПроАвто/Auto.ru

Ни один из вариантов

Автотека/Авито

Avinfobot

1 из 3 вопросов

Продолжить

[Стандартная библиотека Python3.](#) / Модуль functools в Python, инструменты расширения функций

## Инструменты расширения функций и других вызываемых объектов

[Модуль functools](#) предназначен для [функций высшего порядка](#): функций, которые действуют или возвращают другие функции. В общем, любой вызываемый объект может рассматриваться как [функция](#) для целей этого модуля.

Модуль functools предоставляет инструменты для адаптации или расширения функций и других вызываемых объектов, не переписывая их полностью.

### [Способы использования модуля functools в Python](#)

Модуль functools предлагает множество полезных функций более высокого порядка, которые взаимодействуют с другими функциями или возвращают их. Эти функции можно использовать для реализации кэширования функций/методов, перегрузки, создания декораторов и в целом для того, чтобы сделать код немного бол

### [Декоратор @cached\\_property модуля functools в Python](#)

Декоратор @cached\_property модуля functools преобразует метод класса в свойство, значение которого вычисляется один раз, а затем кэшируется как обычный атрибут в течение срока службы экземпляра.

### [Функция cmp\\_to\\_key\(\) модуля functools в Python](#)

Класс `cmp\_to\_key()` модуля `functools` преобразует функцию сравнения старого стиля в ключевую функцию. Эта функция в основном используется в качестве инструмента перехода для программ, конвертируемых из Python 2.

### [Декоратор @cache\(\) модуля functools в Python, кеширующий декоратор](#)

Функция cache() модуля functools представляет собой простой легкий неограниченный кеш функций. Иногда называется "[\\*memoization\\*](#)". Возвращает то же самое, что и @lru\_cache(maxsize=None), создавая тонкую оболочку вокруг поиска по словарю для аргументов функции.

### [Декоратор @lru\\_cache\(\) модуля functools в Python](#)

Декоратор lru\_cache() модуля functools оборачивает функцию с переданными в нее аргументами и запоминает возвращаемый результат соответствующий этим аргументам.

Вверх

[Декоратор @total\\_ordering модуля functools в Python](#)

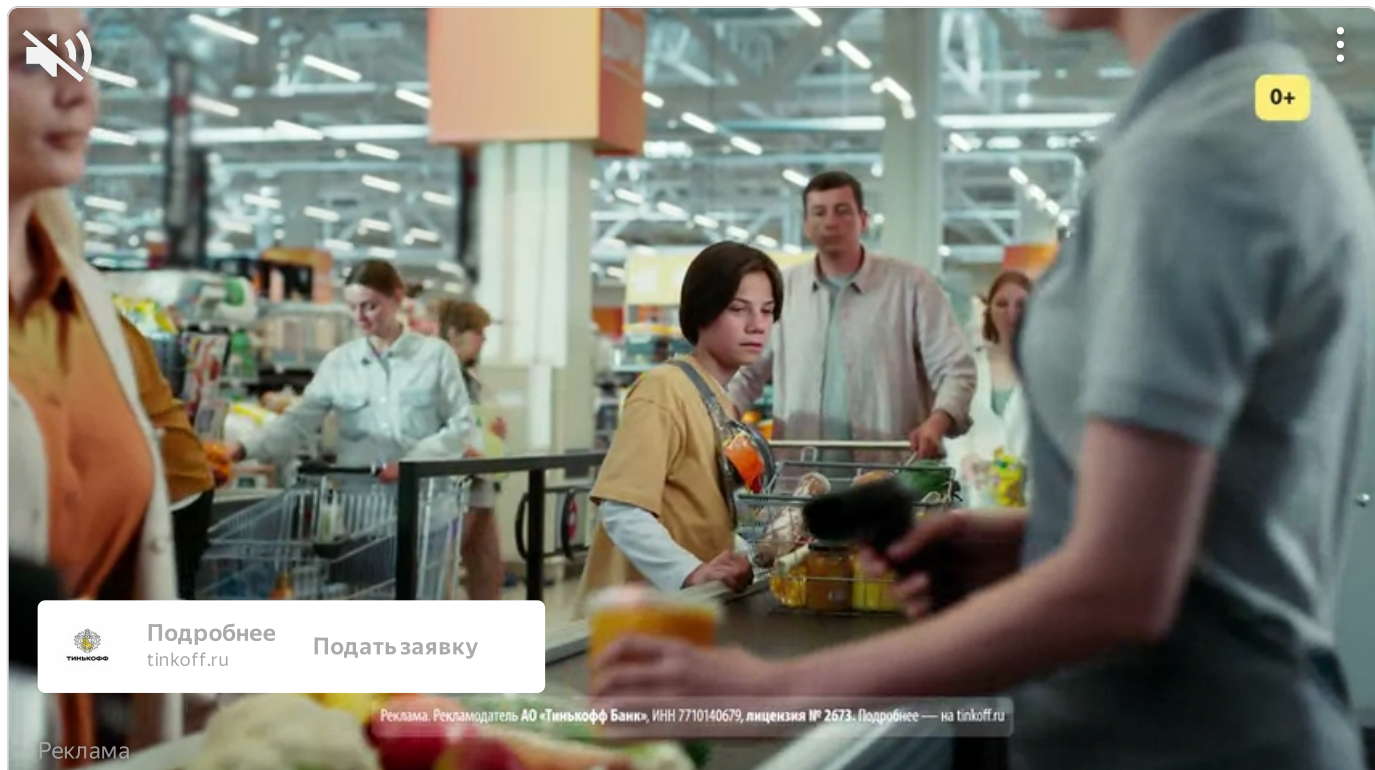
https://docs-python.ru/standart-library/modul-functools-python/

1/3

Декоратор класса `total_ordering()` модуля `functools` оборачивает класс, который определяет один или несколько методов сравнения и добавляет остальные.

[Функция `partial\(\)` модуля `functools` в Python](#)

Функция `partial()` используется для частичного применения каких то аргументов к вызываемой функции `func`. Другими словами `partial()` "замораживает" некоторую часть аргументов и/или ключевых слов, в результате чего создается новый объект с упрощенной записью аргументов вызываемой функции.



[Класс `partialmethod\(\)` модуля `functools` в Python](#)

Класс `partialmethod()` модуля `functools` возвращает новый дескриптор метода `func`, который ведет себя как функция `functools.partial()`, за исключением того, что он предназначен для использования в качестве определения метода, а не для прямого вызова.

[Функция `reduce\(\)` модуля `functools` в Python](#)

Функция `reduce()` модуля `functools` кумулятивно применяет функцию `function` к элементам итерируемой `iterable` последовательности, сводя её к единственному значению.

[Декоратор `@singledispatch` модуля `functools` в Python](#)

Декоратор `@singledispatch` модуля `functools` создает из обычной функции - универсальную функцию одиночной диспетчеризации.

[Декоратор `@singledispatchmethod` модуля `functools` в Python](#)

Декоратор `@singledispatchmethod()` модуля `functools` создает из обычного метода класса - универсальный метод одиночной диспетчеризации.

[Декоратор `@update\_wrapper\(\)` модуля `functools` в Python](#)

Декоратор `@update_wrapper()` модуля `functools` обновляет функцию-обертку, чтобы она выглядела как исходная функция. Другими словами дополняет декоратор, данными из некоторых атрибутов оборачиваемой функции.

[Декоратор `@wraps\(\)` модуля `functools` в Python](#)

Декоратор `@wraps()` модуля `functools` это удобная функция для вызова `@functools.update_wrapper()` в качестве декоратора при определении функции-обертки.

ХОЧУ ПОМОЧЬ  
ПРОЕКТУ

мегамакет  
megamarket.ru

Вверх

Сковорода антипригарная глубокая Moulin Villa Ferdinand FD-26-

Ул, индукция, 26 см. Ferdinand FU-26-UL (англ.) - фердинанд  
фд-26-дм. Рекламодатель ООО «МАРКЕТПЛЕЙС» (агрегатор),  
ОГРН 1167746803180, юридический адрес: 105082, г. Москва,  
ул. Спартаковская площадь, д. 16/15, стр. 6. Сюджа  
предоставляется продавцом АО "МУЛИН ВИЛЛА", 107140, город  
Москва, Верхняя Красносельская ул, д. 3 стр. 1. ОГРН  
5157746128953. Предложение действительно с 01.09.2023г.  
по 14.09.2023г. Количество товара ограничено. Кешбэк –  
возможность получения части стоимости товара бонусными  
единицами в Программе лояльности агрегатора или в Программе  
«СберСпасибо». megamarket.ru. Реклама.

DOCS-Python.ru™, 2023 г.

(Внимание! При копировании материала ссылка на источник обязательна)

@docs\_python\_ru

Вверх