



ХОЧУ ПОМОЧЬ

Модуль weakref, слабые ссылки на объекты в Python

 mango-office.ru



РЕКЛАМА

Виртуальная АТС Расширенная

2 000 ₽

Узнать больше

Ст	АВИТО	/ Модуль weakref, слабые ссылки на объекты в Python
Ст		екты
Мо		раммисту Python создавать слабые ссылки на объекты.
Об		етчик ссылок на объект и предотвращает сбор мусора. Этот результат не всегда
же		ет присутствовать циклическая ссылка или когда кэш объектов должен быть удален, когда
тр		ка - это дескриптор объекта, который не препятствует его автоматической очистке.

Когда единственными оставшимися ссылками на объект являются [слабые ссылки](#), то сборщик мусора может уничтожить этот объект и повторно использовать его память для чего-то другого. Однако до тех пор, пока объект не будет уничтожен, слабая ссылка может вернуть исходный объект, даже если на него нет сильных ссылок.

Основное использование слабых ссылок - реализация кешей или [словарей](#), содержащих большие объекты, которые не должны долго находится в памяти.

Например, есть несколько больших [двоичных](#) объектов (фото или картинки), то можно связать какое то имя с каждым из них. Если использовать [словарь](#) Python для сопоставления имен с картинками или картинки с именами, то объекты картинок остались бы "живыми" только потому, что они находились в словарях как значения или ключи. Классы [weakref.WeakKeyDictionary\(\)](#) и [weakref.WeakValueDictionary](#), предоставляемые [модулем weakref](#), являются альтернативой, используя слабые ссылки для создания словарей, которые не поддерживают объекты "живыми" только потому, что они появляются в объектах словарей. Если, например объект той же картинки является значением в [WeakValueDictionary](#) и последними оставшимися ссылками на этот объект являются слабые ссылки, то сборщик мусора будет извлекать и возвращать объект картинки, а его соответствующие записи в словарях со слабыми ссылками удалит.

[Класс weakref.finalize\(\)](#) предоставляет прямой способ регистрации функции очистки, которая будет вызываться при сборке мусора. Это проще в использовании, чем установка функции обратного вызова на необработанной слабой ссылке, поскольку модуль автоматически гарантирует, что `finalize` остается "живым" до тех пор, пока объект не будет обработан сборщиком мусора.

Все это специфично для CPython. [Модуль weakref](#) работают по-разному в разных реализациях Python.

Поддержка слабых ссылок встроенными типами.

Большинство встроенных типов не поддерживают слабые ссылки, потому что слабый ссылочный механизм Python добавляет некоторые накладные расходы к каждому объекту. Простейшим способом проявления накладных расходов является то, что любой объект с поддержкой слабых ссылок нуждается в пространстве для дополнительного указателя для управления weakref и большинство встроенных объектов не резервируют место для этого указателя.

Если необходимо определить, имеет ли тип слабую ссылочную поддержку, вы можете проверить его атрибут `__weakrefoffset__`, который не будет равен нулю для типов, которые реализуют слабую ссылочную поддержку:

```
>>> int.__weakrefoffset__
0
>>> list.__weakrefoffset__
0
>>> class Foo(object):
...     pass
>>> Foo.__weakrefoffset__
24
>>> set.__weakrefoffset__
```

Вверх

```
192
>>> type.__weakrefoffset__
368
>>> str.__weakrefoffset__
0
РЕКЛАМА
:
```

Несколько встроенных типов, таких как [list](#) и [dict](#), напрямую не поддерживают слабые ссылки, но могут добавить поддержку через подклассы:

```
>>> dict.__weakrefoffset__
# 0
>>> class Dict(dict):
...     pass
...
>>> Dict.__weakrefoffset__
# 56

# этот объект поддерживает слабые ссылки
obj = Dict(red=1, green=2, blue=3)
```

Другие встроенные типы, такие как [tuple](#) и [int](#), не поддерживают слабые ссылки даже в подклассах.

```
>>> tuple.__weakrefoffset__
# 0
>>> class Tupl(tuple):
...     pass
...
>>> Tupl.__weakrefoffset__
# 0
```

Содержание раздела:
<ul style="list-style-type: none">• КРАТКИЙ ОБЗОР МАТЕРИАЛА.• Функция ref() модуля weakref• Функция проху() модуля weakref• Классы WeakKeyDictionary() и WeakValueDictionary() модуля weakref• Класс WeakMethod() модуля weakref• Класс finalize() модуля weakref• Вспомогательные функции модуля weakref• Сравнение метода del и класса finalize()