



Python 3 для начинающих

Книги Самоучитель Курсы Категории ↓ Все статьи

Модуль copy - поверхностное и глубокое копирование объектов



Онлайн-тренажер Python 3 для начинающих

Теория без воды. Задачи с автоматической проверкой. Подсказки на русском языке. Работает в любом современном браузере.

НАЧАТЬ БЕСПЛАТНО

Операция присваивания не копирует объект, он лишь создаёт ссылку на объект. Для изменяемых коллекций, или для коллекций, содержащих изменяемые элементы, часто необходима такая копия, чтобы её можно было изменить, не изменяя оригинал. Данный модуль предоставляет общие (поверхностная и глубокая) операции копирования.

`copy.copy(x)` - возвращает поверхностную копию `x`.

`copy.deepcopy(x)` - возвращает полную копию `x`.

Исключение `copy.error` - возникает, если объект невозможно скопировать.

Разница между поверхностным и глубоким копированием существенна только для составных объектов, содержащих изменяемые объекты (например, список списков, или словарь, в качестве значений которого - списки или словари):

- **Поверхностная копия** создает новый составной объект, и затем (по мере возможности) вставляет в него ссылки на объекты, находящиеся в оригинале.
- **Глубокая копия** создает новый составной объект, и затем рекурсивно вставляет в него копии объектов, находящихся в оригинале.

```
>>> import copy
>>> test_1 = [1, 2, 3, [1, 2, 3]]
>>> test_copy = copy.copy(test_1)
>>> print(test_1, test_copy)
[1, 2, 3, [1, 2, 3]] [1, 2, 3, [1, 2, 3]]
>>> test_copy[3].append(4)
>>> print(test_1, test_copy)
[1, 2, 3, [1, 2, 3, 4]] [1, 2, 3, [1, 2, 3, 4]]
>>> test_1 = [1, 2, 3, [1, 2, 3]]
>>> test_deepcopy = copy.deepcopy(test_1)
>>> test_deepcopy[3].append(4)
>>> print(test_1, test_deepcopy)
[1, 2, 3, [1, 2, 3]] [1, 2, 3, [1, 2, 3, 4]]
```

Для операции глубокого копирования часто возникают две проблемы, которых нет у операции поверхностного копирования:

- Рекурсивные объекты (составные объекты, которые явно или неявно содержат ссылки на себя) могут стать причиной рекурсивного цикла;

Поиск ...

Свежее

- Модуль csv - чтение и запись CSV файлов
- Создаём сайт на Django, используя хорошие практики. Часть 1: создаём проект
- Онлайн-обучение Python: сравнение популярных программ

Категории

- Книги о Python
- GUI (графический интерфейс пользователя)
- Курсы Python
- Модули
- Новости мира Python
- NumPy
- Обработка данных
- Основы программирования
- Примеры программ
- Типы данных в Python
- Видео
- Python для Web
- Работа для Python-программистов

Полезные материалы

- Сделай свой вклад в развитие сайта!
- Самоучитель Python
- Карта сайта
- Отзывы на книги по Python
- Реклама на сайте

Мы в соцсетях

- Поскольку глубокая копия копирует всё, она может скопировать слишком много, например, административные структуры данных, которые должны быть разделяемы даже между копиями.

Функция `deepcopy` решает эти проблемы путем:

- Хранения "мето" словаря объектов, скопированных во время текущего прохода копирования;
- Позволения классам, определенным пользователем, переопределять операцию копирования или набор копируемых компонентов.

```
>>>
>>> r = [1, 2, 3]
>>> r.append(r)
>>> print(r)
[1, 2, 3, [...]]
>>> p = copy.deepcopy(r)
>>> print(p)
[1, 2, 3, [...]]
```

Этот модуль не копирует типы вроде [модулей](#), [классов](#), [функций](#), методов, следа в стеке, стековых кадров, [файлов](#), сокетов, окон, и подобных типов.

Поверхностная копия изменяемых объектов также может быть создана методом `.copy()` у [списков](#) (начиная с [Python 3.3](#)), присваиванием среза (`copied_list = original_list[:]`), методом `.copy()` [словарей](#) и [множеств](#).

Создавать копию неизменяемых объектов (таких, как, например, [строк](#)) необязательно (они же неизменяемые).

Для того, чтобы определить собственную реализацию копирования, класс может определить специальные методы `__copy__()` и `__deepcopy__()`. Первый вызывается для реализации операции поверхностного копирования; дополнительных аргументов не передается. Второй вызывается для реализации операции глубокого копирования; ему передается один аргумент, словарь мето. Если реализация `__deepcopy__()` нуждается в создании глубокой копии компонента, то он должен вызвать функцию `deepcopy()` с компонентом в качестве первого аргумента и словарем мето в качестве второго аргумента.

Ошибка в тексте?
Выделите ее мышкой!
И нажмите:



Для вставки кода на Python в комментарий заключайте его в теги `<pre><code class="python3">Ваш код</code></pre>`

Подпишись на обновления [по RSS](#) или по почте!

Ваш e-mail...

Подписаться!



Присоединиться к обсуждению...

войти с помощью

или через DISQUS 

Имя

 4

Поделиться

Лучшие Новые Старые



Максим Гугаев

6 лет назад

Максимально просто и понятно, спасибо.

3 3 Ответить 



Pandorin

9 лет назад

сурово

2 1 Ответить 

Подписаться

О защите персональных данных