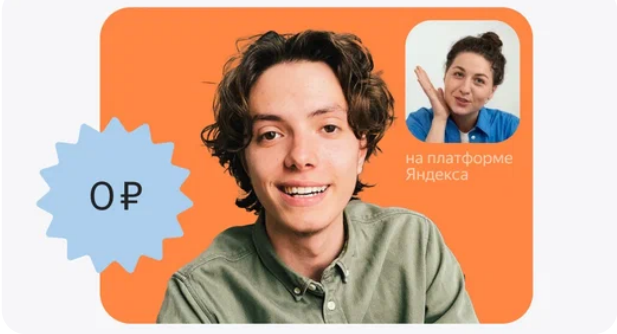


ХОЧУ ПОМОЧЬ  
ПРОЕКТУ

Модуль copyreg в Python, функция используемая при pickling



practicum.yandex.ru

РЕКЛАМА · 18+ Я

**Бесплатное занятие английским в Яндекс Практикуме**  
Полноценное занятие с преподавателем, а не презентация курсов

Узнать больше

Статья / Модуль copyreg в Python, функция используемая при pickling

Функция `copyreg.pickle()` используется при упаковке `pickling` объекта для определения функций, используемых при операции упаковки `pickling` определенных объектов. Функция `copyreg.pickle()` использует эту функции при сохранении объектов.

Модуль `copyreg` определяет конструктор объектов, которые не являются классами. Такие конструкторы используются для объектов, созданных с помощью функций или экземплярами классов.

Модуль определяет 2 функции.

*copyreg.constructor(object):*

Функция `copyreg.constructor()` объявляет объект `object` как допустимый конструктор.

Функция вызывает [исключение TypeError](#), если объект не может быть вызван и следовательно недействителен как конструктор.

*copyreg.pickle(type, function, constructor=None):*

Функция `copyreg.pickle()` объявляет, что функция `function` должна использоваться как функция "сокращения" для объектов типа `type`. Функция должна возвращать либо [строку](#), либо [кортеж](#), содержащий два или три элемента.

Необязательный параметр `constructor`, если он указан, то должен быть вызываемым объектом, который можно использовать для восстановления объекта при вызове с набором аргументов, возвращаемых функцией во время упаковки `pickling`. Если объект является классом или конструктор и не может быть вызван, то возникает [ошибка TypeError](#).

Смотрите [модуль pickle](#) для более подробной информации об интерфейсе функции `function` и конструктора `constructor`.

Обратите внимание, что атрибут `.dispatch_table` объекта выбора или [подкласса pickle.Pickler\(\)](#) также может использоваться для объявления функций "сокращения".

Пример использования:

```
import copyreg, pickle
class C(object):
    def __init__(self, a):
        self.a = a

def pickle_c(c):
    print("pickling a C instance...")
    return C, (c.a,)

copyreg.pickle(C, pickle_c)
c = C(1)
p = pickle.dumps(c)
pickling a C instance...
```



Вверх