Инструкция del

Несмотря на свое название, инструкция del не удаляет объекты в буквальном смысле, она лишь открепляет ссылки, разрывая связь между именем и объектом. Удаление объекта произойдет как следствие работы сборщика мусора в отношении объектов, на которые больше не ссылается ни один другой объект.

- del variable - удаление переменной,
- del obj.attr - удаление атрибута,
- del data[k] - удаление элемента по индексу или ключу в случае словаря,
- del data[i:j] - удаление элементов по срезу.

Инструкция del состоит из ключевого слова del, за которым следует одна или несколько целевых ссылок, разделенных запятыми. Каждая целевая ссылка может быть переменной, ссылкой на атрибут, индексированным элементом или срезом и должна быть связана с каким-либо объектом. Если целевая ссылка является переменной, выполнение инструкции del означает разрыв связи между объектом и закрепленной за ним переменной. Если эта операция запрашивается, она всегда выполняется.

Во всех остальных случаях инструкция del определяет запрос к объекту на открепление одного или нескольких его атрибутов или элементов. Объект может отказать в выполнении этой операции по отношению к некоторым/всем атрибутам или элементам, бросая исключение при попытке отмены связывания в тех случаях, когда это запрещено.

Контейнерам разрешено сопровождать выполнение инструкции del побочными эффектами. Например, предположим что array - это словарь и инструкция del array[2] успешно выполнена, то это означает, что следующие обращения к элементу array[2] будут недопустимыми и приведут к возбуждению исключения KeyError. Такое поведение будет до тех пор, пока array[2] вновь не присвоим некоторое значение. Но если array - это список, то в результате выполнения инструкции del array[2] каждый последующий элемент сместится на одну позицию влево и последующие обращения к array[2] будут по прежнему действительными, хотя будут давать другое значение.

Пример:

```
a = [1, 'not', 10, 'in']
del a[2]
print(a)
# [1, 'not', 'in']
```

```
b = ['foo', 'array', 20]
x = b.append(a)
# удаляем из вложенного списка
del b[3][2]
print(b)
# ['foo', 'array', 20, [1, 'not']]

one = ['foo', 1, 2, 3]
two = [10, 20, 'foo', 30]
three = [11, 'foo', 22, 33]
# удаляем одновременно из всех списков
del one[0], two[2], three[1]
print(one,  two, three)
# [1, 2, 3] [10, 20, 30] [11, 22, 33]



x = {'year': "2020", 'month': "01", 'day': "01"}
y = {'artist': "Beethoven", 'title': 'Symphony No 5'}
del x['day'], y['artist']
print(x,  y)
# {'year': "2020", 'month': "01"} {'title': 'Symphony No 5'}
```