


Оператор in вхождения/членства элемента в Python



mango-office.ru

Виртуальная АТС Расширенная

До 120 функций. Когда нужно распределять звонки по группам сотрудников и по типам...

Узнать больше

РЕКЛАМА

[Справочник по языку Python3.](#) / Оператор in вхождения/членства элемента в Python

Синтаксис:

```
x in s
x not in s
```

[Оператор in и отрицание not in](#) проверяет наличие элемента в последовательности. Выражение `x in s` принимает значение `True`, если `x` является членом `s`, и `False` в противном случае.

Выражение `x not in s` возвращает отрицание наличие элемента `x` в `s`.

Все [встроенные последовательности](#) и типы [множеств set/frozenset](#) поддерживают эту операцию, а также [словарь](#), для которого проверяется, имеет ли словарь данный ключ.

Для типов контейнеров, таких как `list`, `tuple`, `set`, `frozenset`, `dict` или [collections.deque](#), выражение `x in y` эквивалентно вызову `any(x is e or x == e for e in y)`

Для [строковых](#) и [байтовых](#) типов `x in y` имеет значение `True` тогда и только тогда, когда `x` является подстрокой `y`. Эквивалентный тест `y.find(x) != -1`. Пустые строки всегда считаются подстрокой любой другой строки, поэтому `'' in 'abc'` возвращает `True`.

Для определяемых пользователем классов, которые имеют метод `__contains__()`, выражение `x in y` возвращает `True`, если `y.__contains__(x)` возвращает истинное значение, и `False` в противном случае.

Для определяемых пользователем классов, которые НЕ определяют метод `__contains__()`, но определяют `__iter__()`, выражение `x in y` имеет значение `True`, если некоторое значение `z` создается при итерации по `y` и для которого выражение `x is z or x == z` истинно.

Наконец, используется протокол итерации старого стиля: если класс определяет метод `__getitem__()`, выражение `x in y` возвращает `True` тогда и только тогда, когда существует неотрицательный целочисленный индекс `i`, такой что `x is y[i]` or `x == y[i]` и ни один из младших целочисленных индексов не вызывает [исключение IndexError](#).

Оператор `not in` имеет значение обратной истинности `in`.

Примеры использования in, тестирование наличия элемента:

Вхождение/наличие элемента в список, кортеж множество

```
>>> x = [1, 2, 3, 4, 5, 6, 7, 8]
>>> 5 in x
# True
>>> 5 not in x
# False
>>> 0 in x
# False
>>> 0 not in x
# True
```

Вхождение/наличие подстроки в строке

```
>>> x = 'возвращает отрицание наличие элемента'
>>> 'отрицание' in x
```

```
# True
>>> 'наличие' not in x
# False


>>> x = 'абракадабра'
>>> 'б' in x
# True
>>> 'б' not in x
# False
>>> 'н' in x
# False
>>> 'у' not in x
# True
```


Вхождение/наличие ключа в словаре

```
>>> x = {'one':1, 'two':2, 'three':3, 'four':4}
>>> 'one' in x
# True
>>> 'one' not in x
# False
>>> 'five' in x
# False
>>> 'five' not in x
# True
```

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

РЕКЛАМА



 doravtosnab.ru

Вилочные погрузчики Lonking в наличии!

от 1 360 000 ₽

Официальный дилер.
Гарантия завода. Доставка.
Лизинг 0%. Звоните!

Запасные части

Сервис и ремонт

Контакты

Узнать больше