



Программирование в лингвистике. Лекция 6

Множества. Методы множеств

Множества

`sett = {1, 2, 3}`

Множество – это набор элементов; как и списки, множества **изменяемые** и **итерируемые**.

- **Множества** – это объекты, которые работают в точности как множества в **математической теории множеств**.
- Множества **неупорядоченны**.
- У множеств **нет индексации**, поэтому во множестве могут находиться только **хешируемые** (неизменяемые) объекты.
- С этим же связано то, что во множестве могут быть только **уникальные** объекты. Если мы попытаемся добавить во множество объект, который там уже есть, **ничего не произойдет**.



Как создать множество?



Способ 1: Явно задать множество.

```
set_A = {1, 2, 3}
```

```
set_B = {"1", "2", "2"}
```

```
set_B
```

```
{'1', '2'}
```

```
set_C = {[1], [2], [3]}
```

TypeError

(Во множестве могут быть только **уникальные hashable объекты**)





Как создать множество?



Способ 2: Завести пустое и добавлять в него элементы.

```
set_A = set()
for i in range(1, 6):
    set_A.add(i)
set_A
```

{1, 2, 3, 4, 5}

Хотя множество заключено в фигурные скобки, пустое создается с помощью `set()`! Пустыми фигурными скобками мы создаем **словарь**.





Как создать множество?



Способ 3: Преобразование типов.

```
lst = [1, 1, 2, 2, 3]  
set_A = set(lst)  
set_A
```

```
{1, 2, 3}
```

```
string = "aabbcc"  
set_A = set(string)  
set_A
```

```
{'a', 'b', 'c'}
```





Как создать множество?



Способ 4: Генератор множеств (set comprehension).

```
lst = [0, 1, 2, 3, 4]
set_A = {x for x in lst if x}
set_A
```

```
{1, 2, 3, 4}
```





Методы множеств



Добавить элемент:

```
set_A = {1, 2}
set_A.add(3)
set_A
```

set.add(elem)

```
{1, 2, 3}
```

Удалить элемент:

```
set_A = {1, 2, 3}
set_A.discard(3)
set_A
```

set.discard(elem)

```
{1, 2}
```



Методы множеств

Метод	Результат	Пояснение
$A = \{1, 2\}; B = \{3, 4\}$ $A.union(B)$ или $A \cup B$	$\{1, 2, 3, 4\}$	Конъюнкция (объединение). Результат содержит уникальные элементы этих множеств.
$A = \{1, 2\}; B = \{2, 4\}$ $A.intersection(B)$ или $A \& B$	$\{2\}$	Дизъюнкция (пересечение). Результат содержит элементы, имеющиеся во всех этих множествах.
$A = \{1, 2\}; B = \{2, 4\}$ $A.difference(B)$ или $A - B$	$\{1\}$	Разность. Результат содержит элементы, имеющиеся только в первом множестве, но отсутствующие в остальных.
$A = \{1, 2\}; B = \{2, 4\}$ $A.symmetric_difference(B)$ или $A \wedge B$	$\{1, 4\}$	Симметрическая разность. Результат содержит элементы, уникальные для всех этих множеств.



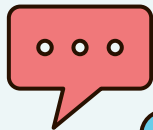
Методы множеств - проверки

Множества	Истина	Пояснение
$A = \{1, 2\}; B = \{1, 2\}$	$A == B$	Все элементы A принадлежат B и наоборот.
$A = \{1, 2\}; B = \{3, 4\}$	<code>A.isdisjoint(B)</code>	A и B не имеют общих элементов.
$A = \{1, 2\}; B = \{1, 2, 3\}$	<code>A.issubset(B)</code> $A \leq B$	Все элементы A принадлежат B.
$A = \{1, 2, 3\}; B = \{2, 3\}$	<code>A.issuperset(B)</code> $A \geq B$	Все элементы B принадлежат A.



Другие методы множеств

Метод	Результат	Пояснение
<code>{1, 2, 3}.remove(2)</code> <code>{1, 3}.remove(2)</code>	<code>{1, 3}</code> <code>KeyError</code>	Удаляет элемент из множества. <code>KeyError</code> , если не найдено.
<code>{1, 2, 3}.pop()</code>	Удален случайный элемент	Удаляет элемент с начала множества и возвращает его. Множества неупорядоченны, так что неизвестно, какой именно элемент будет удален.
<code>{1, 2, 3}.clear()</code>	<code>set()</code>	Очищает множество.



Спасибо за внимание!

admvereshchagina@gmail.com