

Множества

План урока

- Понятие множества
- Устройство множества
- Свойства и методы множества
- Примеры

МНОЖЕСТВО

Это набор **уникальных** элементов



МНОЖЕСТВО

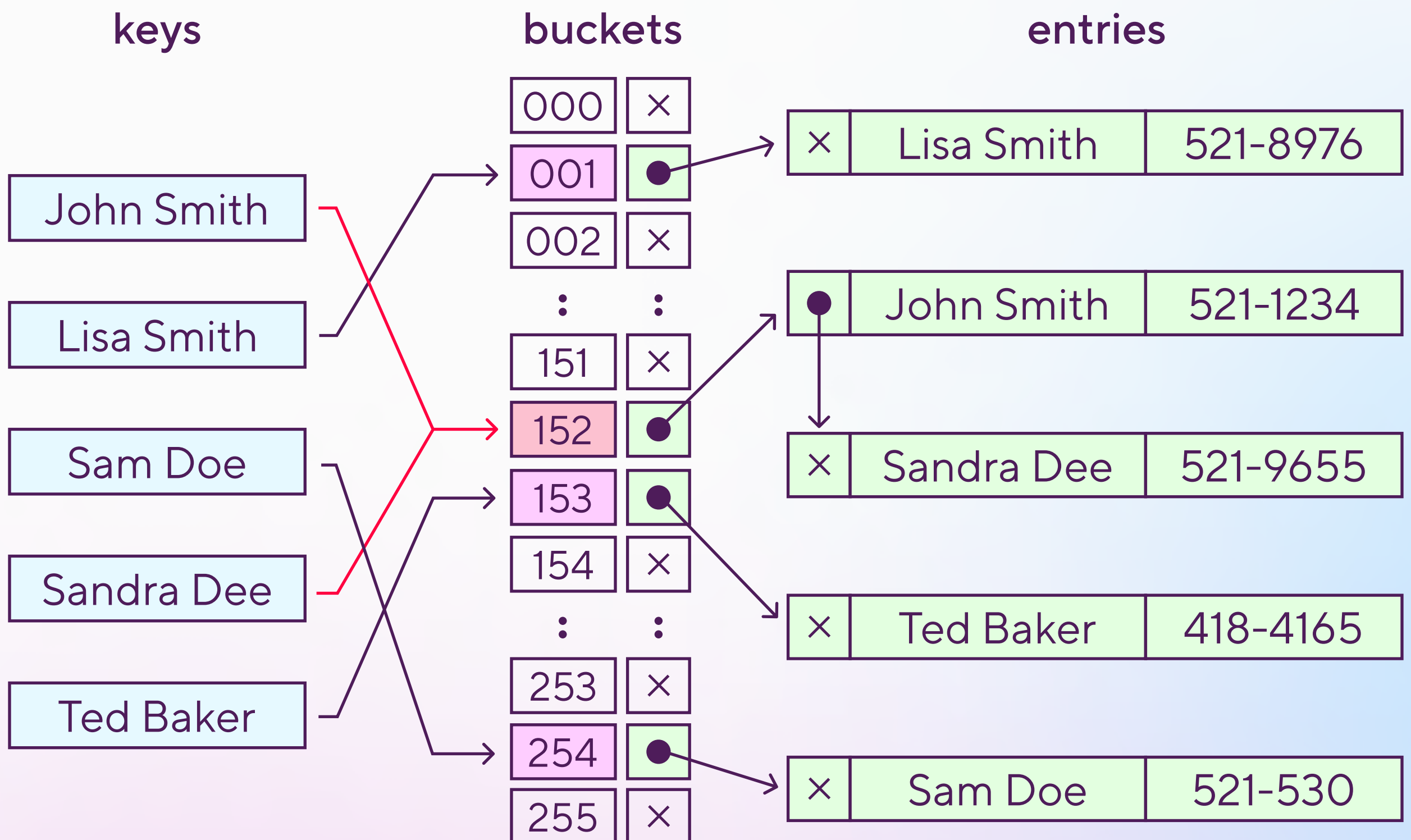
Дискретное множество

это набор отдельных элементов

В Python класс **set** используется для представления дискретных множеств с конечным набором элементов

Множество

Множество в Python - это изменяемый и итерируемый тип данных, построенный на **хеш-таблицах**



Множество

о о о	КОД
<pre>s = {3, 6, 9, -1, 2} for x in s: print(x)</pre>	

о о о	КОД
<pre>>>> 2 >>> 3 >>> 6 >>> 9 >>> -1</pre>	

о о о	КОД
<pre>s = {3, 6, 9, -1, 2} print(len(s))</pre>	

о о о	КОД
<pre>>>> 5</pre>	

Добавление элемента

о о о	КОД
<pre>s = {4, 23, 9, -4, 6, 8} s.add(11) print(s) s.add(6) print(s)</pre>	

о о о	КОД
<pre>>>> {4, 6, 23, 8, 9, 11, -4} >>> {4, 6, 23, 8, 9, 11, -4}</pre>	

Множество

Методы для **удаления** элементов:

- Remove
- Discard
- Pop

Remove

Метод **remove** удаляет элемент из множества или вызывает исключение `KeyError`, если элемента нет

Discard

Метод **discard** удаляет элемент из множества, если он присутствует, и ничего не делает, если элемент отсутствует

Remove

ooo

КОД

```
s = {4, 6, 23, 8, 9, 11, -4}
s.remove(6)
print(s)

try:
    s.remove(6)
except KeyError:
    print("there is no 6 in s")
```

ooo

ВЫВОД

```
>>> {4, 23, 8, 9, 11, -4}
>>> there is no 6 in s
```

Remove

ooo	КОД
<pre>s = {4, 23, 8, 9, 11, -4} s.discard(9) print(s) s.discard(9) print(s)</pre>	

ooo	ВЫВОД
<pre>>>> {4, 23, 8, 11, -4} >>> {4, 23, 8, 11, -4}</pre>	

Remove

о о о	КОД
<pre>s = {4, 23, 8, 11, -4} print(s.pop())</pre>	

о о о	ВЫВОД
<pre>>>> 4</pre>	

Оператор in

о о о	КОД
<pre>s = {4, 23, 8, 11, -4} print(23 in s) print(50 in s)</pre>	

о о о	ВЫВОД
<pre>>>> True >>> False</pre>	

Объединение множеств

о о о	КОД
<pre>a = {3, 4, 0, 11} b = {45, -3, 4, 11, 34} print(a b)</pre>	

о о о	ВЫВОД
<pre>>>> {0, 34, 3, 4, 11, 45, -3}</pre>	

Создаем
множества a и b

Объединение множеств

о о о	КОД
<pre>a = {3, 4, 0, 11} b = {45, -3, 4, 11, 34} print(a b) ←</pre>	

о о о	ВЫВОД
<pre>>>> {0, 34, 3, 4, 11, 45, -3}</pre>	

Объединение
множеств

Пересечение множеств

о о о	КОД
<pre>a = {3, 4, 0, 11} b = {45, -3, 4, 11, 34} print(a & b) ←</pre>	

о о о	ВЫВОД
<pre>>>> {11, 4}</pre>	

Пересечение
множеств

Сравнение множеств

о о о	КОД
<pre>a = {3, 4, 0, 11} b = {45, -3, 4, 11, 34} print(a < b)</pre>	

о о о	ВЫВОД
<pre>>>> False</pre>	

Сравнение множеств

о о о	КОД
<pre>a = {3, 4, 0, 11} b = {45, -3, 4, 11, 34} print(a <= b)</pre>	

о о о	ВЫВОД
<pre>>>> False</pre>	

Сравнение множеств

о о о	КОД
<pre>a = {3, 4, 0, 11} b = {3, 4, 0, 11} print(a == b)</pre>	

о о о	ВЫВОД
<pre>>>> True</pre>	

Вычитание множеств

о о о	КОД
<pre>a = {3, 4, 0, 11} b = {45, -3, 4, 11, 34} print(a - b)</pre>	

о о о	ВЫВОД
<pre>>>> {0, 3}</pre>	

python.org

Множество

ooo	КОД
	<pre>S = { 5, 'hello', ("Peter", 25), 5.8, 2 + 5j, True, None, }</pre>

Списки и множества нельзя использовать в качестве ключей, так как они **изменяемые** и **нехэшируемые** типы данных

Задача

Посчитать количество уникальных элементов в списке

о о о	КОД
<pre>numbers = [5, 9, 0, -3, 4, 5, 1, 4, 5, 0] unique_numbers = set(numbers) print(len(unique_numbers))</pre>	

о о о	ВЫВОД
<pre>>>> 6</pre>	

Задача

Даны две строки: “**hello**” и “**world**”

Необходимо найти множество их общих символов

о о о	КОД
<pre>s1 = "hello" s2 = "world" common_characters = set(s1) & set(s2) print(common_characters)</pre>	

о о о	ВЫВОД
<pre>>>> {'l', 'o'}</pre>	