

# Коллекции данных. Словари. Множества.

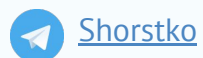
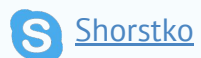


**ЕЛЕНА НИКИТИНА**



## Елена Никитина

Руководитель проектов ГК «Геоскан»





# План занятия

1. [Словари](#)
2. [Операции над словарями](#)
3. [Множества](#)
4. [Операции над множествами](#)



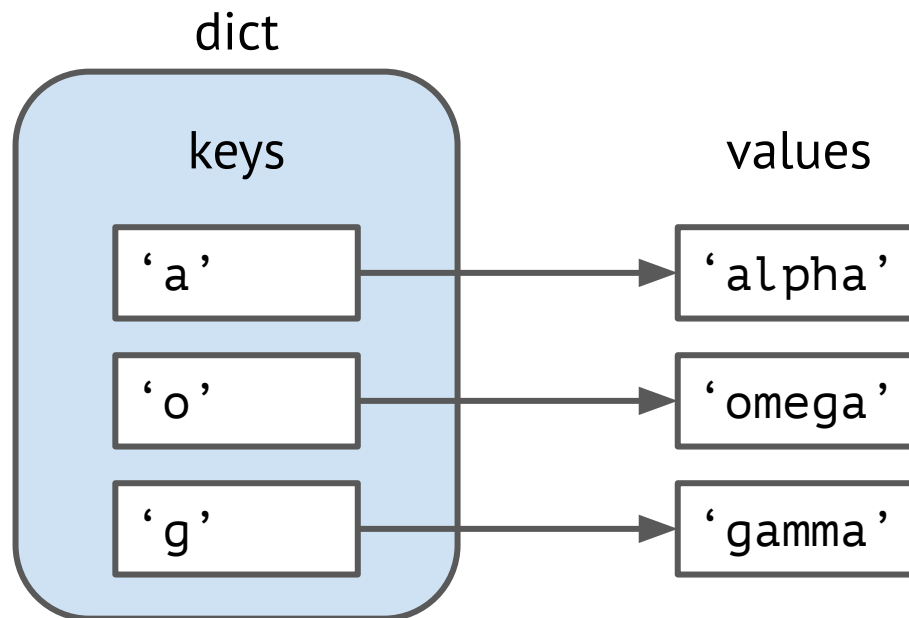
# Словари

# Словари

Словари (dictionaries) — неупорядоченные коллекции произвольных объектов с доступом по ключу. Значением словаря может быть любой тип данных.

Элементы в словарях хранятся в формате *key:value*.

```
dict = {'a': 'alpha', 'o': 'omega', 'g': 'gamma'}
```



# Словари

Словарь инициализируется при помощи функции **dict()** или **{ }**.

```
telephones_dict = dict()  
capitals_dict = { }
```

# Словари

Ключами могут быть любые неизменяемые типы данных: *strings*, *booleans*, *integers*, *floats*, *tuple*.

**key           :   value**

```
telephones_dict = {88009991122: 'Оля', 89008882211: 'Коля'}  
capitals_dict = {'Russia': 'Moscow', 'Ukraine': 'Kiev'}
```

Все ключи в словаре должны быть уникальными.

# Словари

Получение значения из словаря по ключу:

```
capitals_dict = {'Russia': 'Moscow', 'Ukraine': 'Kiev'}  
print(capitals_dict['Russia'])
```

Итерация по словарю:

```
for country, capital in capitals_dict.items():  
    print(country, '->', capital)
```

Добавление нового элемента в словарь:

```
capitals_dict['France'] = 'Paris'
```





# Операции над словарями

# Операции над словарями

- ***del(dict[key])*** удаляет элемент из списка по ключу;
- ***.keys()*** позволяет получить все ключи словаря;
- ***.values()*** позволяет получить все значения словаря;
- ***.items()*** позволяет получить ключи и значения словаря;
- ***.get(key)*** “безопасно” возвращает значение по ключу (при отсутствии ключа ошибка не возникает);
- ***.setdefault(key, default)*** позволяет получить значение по ключу, автоматически добавляет элемент в словарь, если его нет.



# Множества



# Множества

Множества (sets) — «контейнер», содержащий неповторяющиеся элементы в случайном порядке.

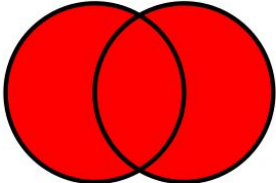
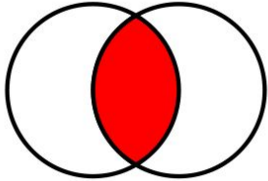
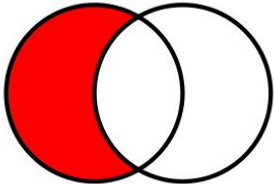
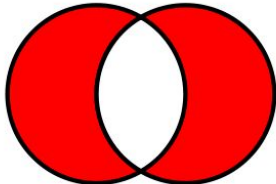
Множество инициализируется при помощи **set()**. Обычно создается из списков.

Реализуют теорию множеств в Python ([см. круги Эйлера](#)).



# Операции над множествами

# Операции над множествами

Объединение(union) Логическое “ИЛИ” (or)		<pre>set_a = {1, 2, 3} set_b = {3, 4, 5} set_c = set_a   set_b # {1, 2, 3, 4, 5}</pre>
Пересечение(intersection) Логическое “И” (and)		<pre>set_c = set_a &amp; set_b # {3}</pre>
Разность (difference)		<pre>set_c = set_a - set_b # {1, 2}</pre>
Симметричная разность (symmetric difference)		<pre>set_c = set_a ^ set_b # {1, 2, 4, 5}</pre>

# Операции над множествами

- **.add(el)** добавляет элемент в множество;
- **.update(set)** соединяет множество с другим множеством/списком;
- **.discard(el)** удаляет элемент из множества по его значению;
- **.union(set)** объединяет множества (логическое “ИЛИ”);
- **.intersection(set)** — пересечение множеств (логическое “И”);
- **.difference(set)** возвращает элементы одного множества, которые не принадлежат другому множеству (разность множеств);
- **.symmetric\_difference(set)** возвращает элементы, которые встречаются в одном множестве, но не встречаются в обоих.



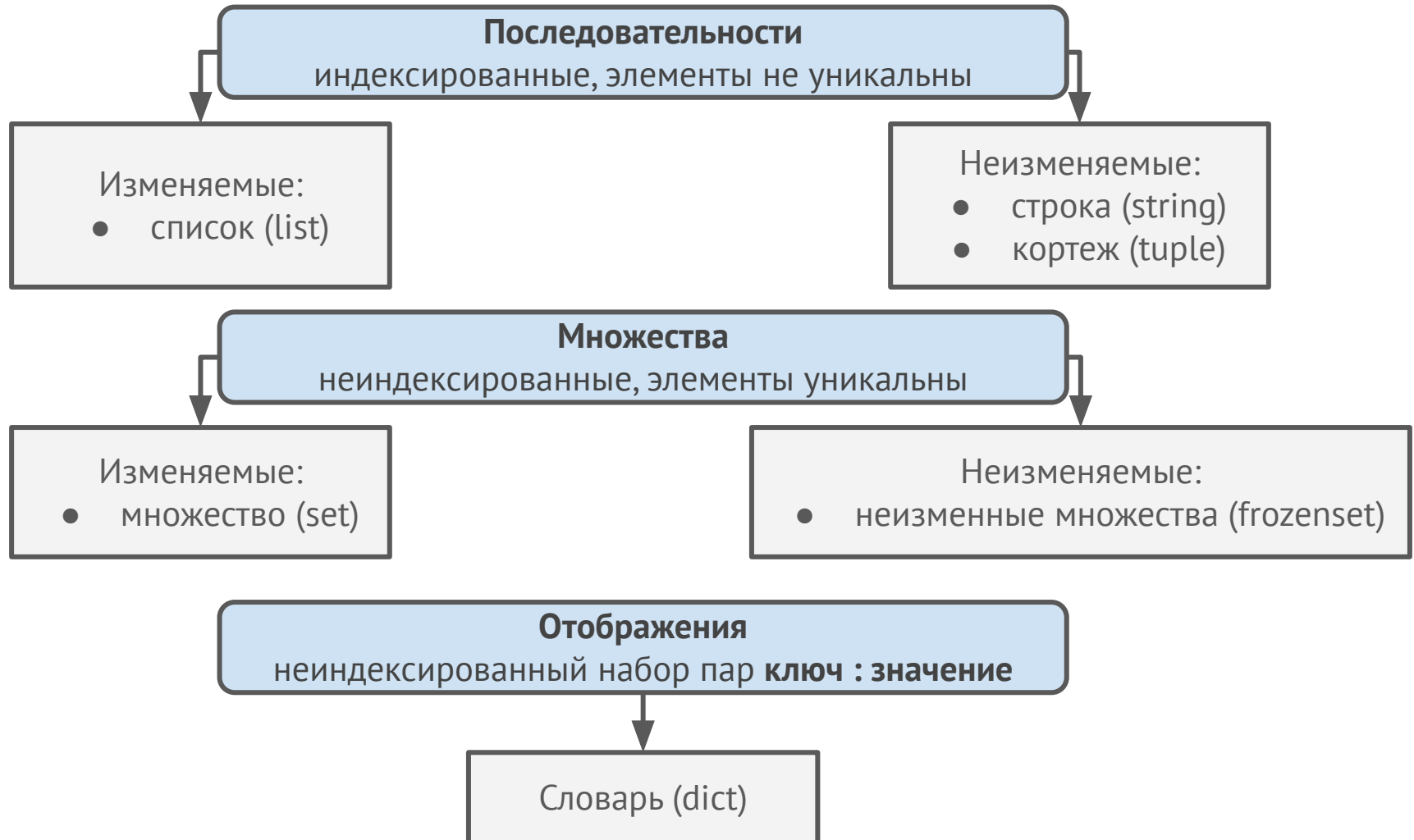
# Статические множества

Статические множества (*frozenset*) — неизменяемое множество.

Над *frozenset* доступны все операции, применяемые к множествам, за исключением тех, которые их изменяют.



# Классификация коллекций





# Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задаём в чате Slack!
- Задачи можно сдавать по частям.
- Зачёт по домашней работе проставляется после того, как приняты **все задачи**.



**Задавайте вопросы и напишите отзыв о лекции!**

**ЕЛЕНА НИКИТИНА**

