

Словари

План урока

- Абстрактная структура данных map
- Свойства и методы словарей в Python
- Примеры

MAP

это абстрактная структура данных,
хранящая пары ключ-значение

в Python она реализована через хэш-
таблицы, что позволяет быстро выполнять
операции поиска, модификации
и удаления данных по ключу

СЛОВАРЬ

это неупорядоченная изменяющаяся коллекция, доступ к элементам которой выполняется по ключу

словари и множества похожи, но имеют и некоторые различия

оба используют хэш-таблицы

Создание словаря

ооо

код

```
my_dict = {'key1': 'value1', 'key2': 'value2'}
```

Пары ключ-значение
разделяем запятыми
и заключаем всё
в фигурные скобки

Создание словаря

ооо

код

```
my_dict = {'key1': 'value1', 'key2': 'value2'}
```

Внутри каждой пары ключ
и значение разделены
двоеточием

Создание словаря

о о о	код
my_dict = {}	

Множество vs словарь

ооо

код

```
# создание множества
```

```
set1 = set()
```

```
# создание словаря
```

```
dict1 = {} или dict1 = ()
```

Цикл for в словаре

ooo | код

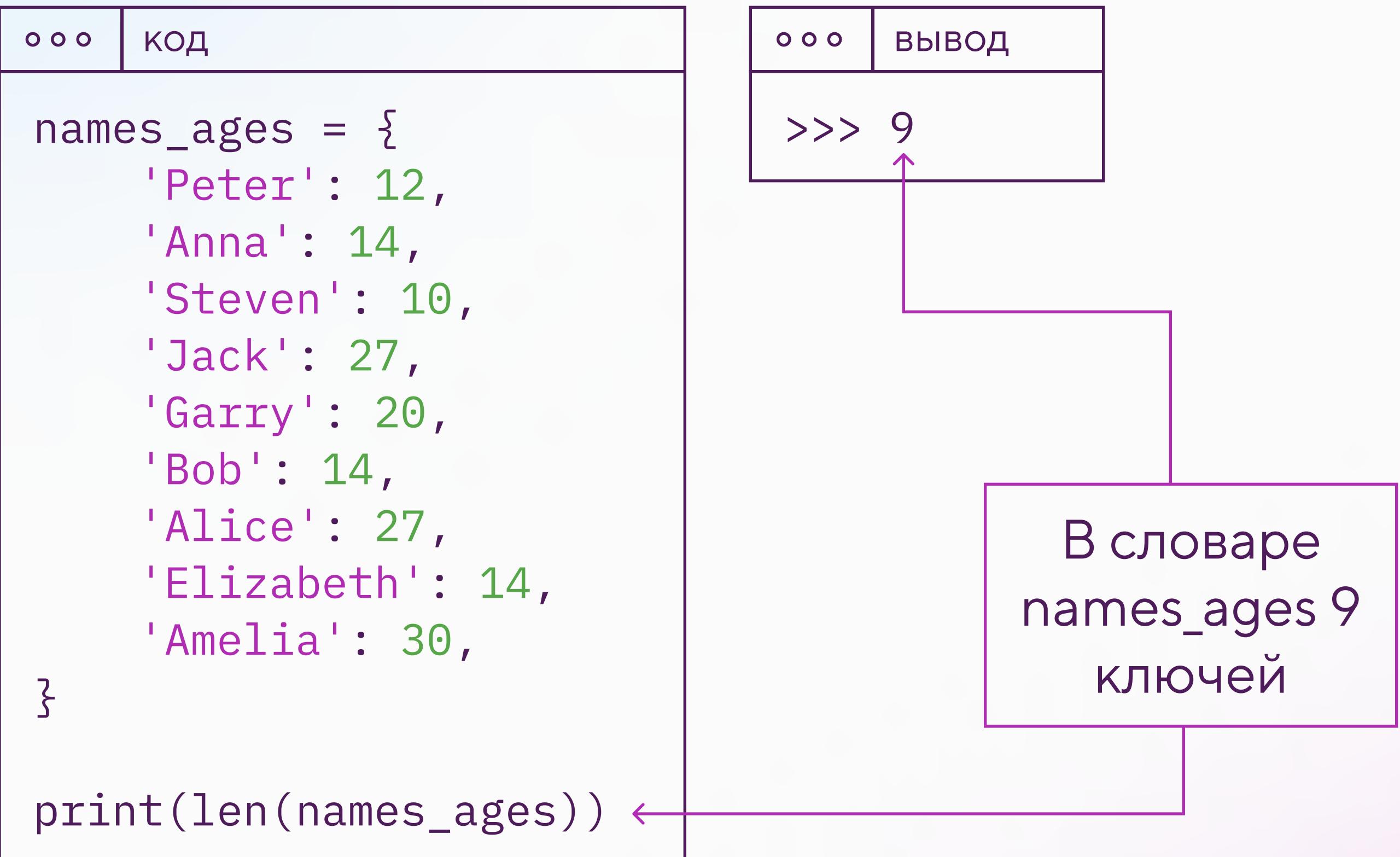
```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30,  
}  
  
for x in names_ages:  
    print(x) ↑
```

ooo | код

```
>>> Peter  
>>> Anna  
>>> Steven  
>>> Jack  
>>> Garry  
>>> Bob  
>>> Alice  
>>> Elizabeth  
>>> Amelia
```

Проходимся
по словарю
names_ages

Метод len



Метод len

ooo

код

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30,  
}  
  
try:  
    print(hash(names_ages))  
except TypeError:  
    print("dicts are unhashable")
```

ooo

вывод

```
>>> dicts are unhashable
```

Словарь

о о о КОД

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30,  
}  
  
print(names_ages['Steven'])
```

о о о ВЫВОД

```
>>> 10
```

Обращаемся
к значению
по ключу

Словарь

о о о КОД

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30,  
}  
  
print(names_ages['Steven'])
```

о о о ВЫВОД

```
>>> 10
```

Получаем значение
возраста по ключу Steven
в словаре names_ages

Словарь

ooo | код

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30,  
}  
  
try:  
    print(names_ages['Joe'])  
except KeyError:  
    print('there is no key Joe in names_ages')
```

ooo | вывод

```
>>> there is no key Joe in names_ages
```

Словарь

ооо	код	ооо	вывод
	<pre>names_ages = { 'Peter': 12, 'Anna': 14, 'Steven': 10, 'Jack': 27, 'Garry': 20, 'Bob': 14, 'Alice': 27, 'Elizabeth': 14, 'Amelia': 30, } names_ages['Joe'] = 30 ← print(names_ages)</pre>		<pre>>>> {'Peter': 12, >>> 'Anna': 14, >>> 'Steven': 10, >>> 'Jack': 27, >>> 'Garry': 20, >>> 'Bob': 14, >>> 'Alice': 27, >>> 'Elizabeth': 14, >>> 'Amelia': 30, >>> 'Joe': 30}</pre>

Добавляем ключ "Joe"
со значением 30

Словарь

ооо	код	ооо	вывод
	<pre>names_ages = { 'Peter': 12, 'Anna': 14, 'Steven': 10, 'Jack': 27, 'Garry': 20, 'Bob': 14, 'Alice': 27, 'Elizabeth': 14, 'Amelia': 30, } names_ages['Joe'] = 31 print(names_ages)</pre>		<pre>>>> {'Peter': 12, >>> 'Anna': 14, >>> 'Steven': 10, >>> 'Jack': 27, >>> 'Garry': 20, >>> 'Bob': 14, >>> 'Alice': 27, >>> 'Elizabeth': 14, >>> 'Amelia': 30, >>> 'Joe': 31}</pre>

Добавляем ключ "Joe"
со значением 31

Словарь

ооо	код	ооо	вывод
	<pre>names_ages = { 'Peter': 12, 'Anna': 14, 'Steven': 10, 'Jack': 27, 'Garry': 20, 'Bob': 14, 'Alice': 27, 'Elizabeth': 14, 'Amelia': 30, 'Joe': 31 } del names_ages['Joe'] ← print(names_ages)</pre>		<pre>>>> {'Peter': 12, >>> 'Anna': 14, >>> 'Steven': 10, >>> 'Jack': 27, >>> 'Garry': 20, >>> 'Bob': 14, >>> 'Alice': 27, >>> 'Elizabeth': 14, >>> 'Amelia': 30}</pre>

Удаляем пару
с ключом Joe

Словарь

о о о

код

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30}  
  
print(list(names_ages))
```

о о о

вывод

```
>>> ['Peter',  
      'Anna',  
      'Steven',  
      'Jack',  
      'Garry',  
      'Bob',  
      'Alice',  
      'Elizabeth', 'Amelia']
```

Методы просмотра



Метод keys

ооо

код

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30}
```

```
keys = names_ages.keys()  
print(keys, type(keys))
```

ооо

вывод

```
>>> ['Peter',  
     'Anna',  
     'Steven',  
     'Jack',  
     'Garry',  
     'Bob',  
     'Alice',  
     'Elizabeth', 'Amelia']
```

Метод values

ооо

код

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30}
```

```
values = names_ages.values()  
print(values, type(values))
```

ооо

вывод

```
>>> dict_values([12, 14,  
10, 27, 20, 14, 27, 14,  
30])  
<class 'dict_values'>
```

Метод items

о о о

код

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30}  
  
items = names_ages.items()  
print(items, type(items))
```

о о о

вывод

```
>>> dict_items  
([('Peter', 12),  
 ('Anna', 14),  
 ('Steven', 10),  
 ('Jack', 27),  
 ('Garry', 20),  
 ('Bob', 14),  
 ('Alice', 27),  
 ('Elizabeth', 14),  
 ('Amelia', 30)])  
<class 'dict_items'>
```

Словарь

ооо

код

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30}
```

```
for name, age in  
names_ages.items():  
    print(name, age) ←
```

ооо

вывод

```
>>> Peter 12  
>>> Anna 14  
>>> Steven 10  
>>> Jack 27  
>>> Garry 20  
>>> Bob 14  
>>> Alice 27  
>>> Elizabeth 14  
>>> Amelia 30
```

Мы проходим по коллекции `items` и записываем
данные кортежа из коллекции в переменные `name` и `age`

Оператор in

о о о

код

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30}  
  
print('Peter' in names_ages)
```

о о о

вывод

```
>>> True
```

МЕТОД GET

используется для поиска
значения по ключу в словаре

если значение найдено,
то оно будет возвращено

если ключ не найден, то метод вернет
значение по умолчанию, указанное
вторым аргументом

Метод get

ooo | код

```
names_ages = {  
    'Peter': 12,  
    'Anna': 14,  
    'Steven': 10,  
    'Jack': 27,  
    'Garry': 20,  
    'Bob': 14,  
    'Alice': 27,  
    'Elizabeth': 14,  
    'Amelia': 30}
```

```
print(names_ages.get("Hello", 100))
```

ooo | вывод

```
>>> 100
```

Объединение словарей

Если у двух объединяемых словарей есть совпадающие ключи, то этим ключам приписываются значения из словаря справа

ooo	словарь 1
	names_ages = { 'Peter': 12, 'Anna': 14, 'Steven': 10, 'Jack': 27, 'Garry': 20, 'Bob': 14, 'Alice': 27, 'Elizabeth': 14, 'Amelia': 30}

ooo	словарь 2
	names_ages_2 = { "William": 43, "James": 27, "Peter": 13}

Объединение словарей

Если у двух объединяемых словарей есть совпадающие ключи, то этим ключам приписываются значения из словаря справа

ooo	код
	<pre>names_ages_2 = { "William": 43, "James": 27, "Peter": 13, } names_ages = names_ages_2 print(names_ages)</pre>

ooo	вывод
	<pre>>>> {'Peter': 13, >>> 'Anna': 14, >>> 'Steven': 10, >>> 'Jack': 27, >>> 'Garry': 20, >>> 'Bob': 14, >>> 'Alice': 27, >>> 'Elizabeth': 14, >>> 'Amelia': 30 >>> 'William': 43, >>> 'James': 27}</pre>

Объединение словарей

Если у двух объединяемых словарей есть совпадающие ключи, то этим ключам приписываются значения из словаря справа

ooo	код	ooo	вывод
	<pre>names_ages_2 = { "William": 43, "James": 27, "Peter": 13, } names_ages = names_ages_2 print(names_ages)</pre>		<pre>>>> {'Peter': 13, >>> 'Anna': 14, >>> 'Steven': 10, >>> 'Jack': 27, >>> 'Garry': 20, >>> 'Bob': 14, >>> 'Alice': 27, >>> 'Elizabeth': 14, >>> 'Amelia': 30 >>> 'William': 43, >>> 'James': 27}</pre>

Используем операцию присваивания с объединением для словаря names_ages со словарём name_ages_2

Объединение словарей

Если у двух объединяемых словарей есть совпадающие ключи, то этим ключам приписываются значения из словаря справа

ooo	код
	<pre>names_ages_2 = { "William": 43, "James": 27, "Peter": 13, } names_ages = names_ages_2 print(names_ages)</pre>

ooo	вывод
	<pre>>>> {'Peter': 13, >>> 'Anna': 14, >>> 'Steven': 10, >>> 'Jack': 27, >>> 'Garry': 20, >>> 'Bob': 14, >>> 'Alice': 27, >>> 'Elizabeth': 14, >>> 'Amelia': 30 >>> 'William': 43, >>> 'James': 27}</pre>

python.org

Объединение словарей

Дан текст из слов, разделенных пробелами
Необходимо подсчитать частоту вхождения
слов в текст

ooo | код

```
text = "ice cold blue hot man ice lake ice man"

word_frequencies = {}
for word in text.split():
    if word in word_frequencies:
        word_frequencies[word] += 1
    else:
        word_frequencies[word] = 1
print(word_frequencies)
```

ooo | вывод

```
>>> {'ice': 3, 'cold': 1, 'blue': 1,
>>> 'hot': 1, 'man': 2, 'lake': 1}
```

Объединение словарей

Дан текст из слов, разделенных пробелами
Необходимо подсчитать частоту вхождения
слов в текст

ooo | код

```
text = "ice cold blue hot man ice lake ice  
man"
```

```
word_frequencies = {}  
for word in text.split():  
    word_frequencies[word] =  
        word_frequencies.get(word, 0) + 1  
print(word_frequencies)
```

ooo | вывод

```
>>> {'ice': 3, 'cold': 1, 'blue': 1,  
>>> 'hot': 1, 'man': 2, 'lake': 1}
```

СЛОВАРИ

это гибкий и мощный инструмент
в решении самых различных задач