

Яндекс



Знакомство со словарями

Старый способ – списки

```
actors = [ 'Джонни Депп', 'Эмма Уотсон', 'Билли Пайпер' ]  
print(actors[1])
```

Индексы в списках – последовательности целые числа

Энциклопедия об актёрах

Имя актёра и его биография. Можно сделать списком кортежей

```
actors = [( 'Джонни Депп', 'Джон Кристофер Депп Второй родился 9 июня
1963 года в Овенсборо, Кентукки.. ' ),
( 'Сильвестр Сталлоне', 'Сильвестр Гарденцио Сталлоне родился в Нью-
Йорке. Его отец, парикмахер Фрэнк Сталлоне — иммигрант из Сицилии,
... ' ),
( 'Эмма Уотсон', 'Эмма Шарлотта Дуерр Уотсон родилась в семье английских
адвокатов. В пять лет переехала вместе с семьей из Парижа в Англию... ' ),
# ...
]
```

Энциклопедия об актёрах

Имя актёра и его биография. Можно сделать списком кортежей

```
actors = [( 'Джонни Депп', 'Джон Кристофер Депп Второй родился 9 июня
1963 года в Овенсборо, Кентукки.. ' ),
( 'Сильвестр Сталлоне', 'Сильвестр Гарденцио Сталлоне родился в Нью-
Йорке. Его отец, парикмахер Фрэнк Сталлоне — иммигрант из Сицилии,
... ' ),
( 'Эмма Уотсон', 'Эмма Шарлотта Дуерр Уотсон родилась в семье английских
адвокатов. В пять лет переехала вместе с семьей из Парижа в Англию... ' ),
# ...
]
```

Неудобно: для поиска статьи об актёре придётся пройти по всему списку

Словари

Словарь хранит, как и список, много данных.

Но индексом («ключом») может быть строка – например, имя актёра

```
actors = {  
    'Джонни Депп': 'Джон Кристофер Депп Второй родился 9 июня 1963 года в  
Овенсборо, Кентукки..',  
    'Сильвестр Сталлоне': 'Сильвестр Гарденцио Сталлоне родился в Нью-  
Йорке. Его отец, парикмахер Фрэнк Сталлоне – иммигрант из Сицилии, ...',  
    'Эмма Уотсон': 'Эмма Шарлотта Дуерр Уотсон родилась в семье английских  
адвокатов. В пять лет переехала вместе с семьей из Парижа в Англию...',  
    # ...  
}
```

Пустой словарь

```
d = dict()  
# или так  
d = {}
```

Обращение к элементу словаря

Теперь легко найти и вывести статью о конкретном актёре

```
print(actors[ 'Эмма Уотсон' ])
```

(сравните с обращением к элементу списка)

Обращение к элементу словаря

Теперь легко найти и вывести статью о конкретном актёре

```
print(actors['Эмма Уотсон'])
```

(сравните с обращением к элементу списка)

```
first_name = 'Сильвестр'  
last_name = 'Сталлоне'  
print(actors[first_name + ' ' + last_name])
```

Обращение к элементу словаря

Теперь легко найти и вывести статью о конкретном актёре

```
print(actors['Эмма Уотсон'])
```

(сравните с обращением к элементу списка)

```
first_name = 'Сильвестр'  
last_name = 'Сталлоне'  
print(actors[first_name + ' ' + last_name])
```

Если ключа в словаре нет, возникнет ошибка

```
print(actors['Несуществующий ключ'])  
KeyError: 'Несуществующий ключ'
```

Изменение словаря

Добавляем новую запись или редактируем существующую

```
actors[ 'Брэд Питт' ] = 'Уильям Брэдли Питт, более известный как Брэд Питт  
– американский актёр и продюсер. Лауреат премии «Золотой глобус» за 1995  
год, ...';
```

Изменение словаря

Добавляем новую запись или редактируем существующую

```
actors[ 'Брэд Питт' ] = 'Уильям Брэдли Питт, более известный как Брэд Питт  
– американский актёр и продюсер. Лауреат премии «Золотой глобус» за 1995  
год, ...';
```

Удаляем запись про Джонни Деппа

```
del actors[ 'Джонни Депп' ]  
# больше в словаре нет ни ключа 'Джонни Депп', ни соответствующего ему  
значения  
  
print(actors[ 'Джонни Депп' ])  
KeyError: 'Джонни Депп'
```

Изменение словаря

Удалять элемент из словаря можно и по другому

```
actors.pop( 'Джонни Депп' )
```

Этот способ идентичен предыдущему, однако ещё и возвращает удалённое значение

```
deleted_value = actors.pop( 'Джонни Депп' )
```

Если в словаре нет ключа 'Джонни Депп', в обоих случаях ошибка `KeyError`

Изменение словаря

Чтобы ошибки о несуществующем ключе не возникало, можно в функцию `pop()` передать второй аргумент. В случае отсутствия нужного ключа `pop` вернёт этот аргумент

```
deleted_value = actors.pop('Джонни Депп', None)
```

Если ключа 'Джонни Депп' в словаре не оказалось, в переменной `deleted_value` окажется `None`, а ошибки не возникнет

Поиск в словаре

Есть ли ключ в словаре?

```
if 'Джонни Депп' in actors:  
    print('У нас есть статья про Джонни Деппа')
```

Аналогичный оператор **not in**

```
if 'Сергей Безруков' not in actors:  
    print('У нас нет статьи о Сергее Безрукове')
```

Методы словарей

Взять значение в словаре можно и с помощью метода **get**:

```
article = actors.get( 'Джонни Депп' )
```


Методы словарей

Взять значение в словаре можно и с помощью метода **get**:

```
article = actors.get( 'Джонни Депп' )
```

Кроме ключа он может принимать и второй аргумент — значение, которое вернётся, если заданного ключа нет:

```
article = actors.get( 'Джонни Депп', 'Статья о Джонни Деппе не найдена' )
```

Методы словарей

С помощью метода `keys` можно получить список всех ключей словаря:

```
actors_names = list(actors.keys())
```

Методы словарей

Есть и парный метод **values**, возвращающий все значения словаря:

```
all_articles = list(actors.values())
```

Методы словарей

С помощью метода **items** можно получить пары ключ-значение и использовать их, например, при переборе элементов словаря:

```
for key, val in d.items():
```

Яндекс