

## 17. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей. Операции поэлементного извлечения из словаря и их использование.

Всего существует 2 базовые операции со словарями, учитывающие возможное отсутствие ключей:

- `dict.get()` - получение элемента словаря по ключу
- `dict.setdefault()` - получает элемент, если он есть, если нет - вставляет его в словарь

Все остальные методы вызовут exception типа `KeyError`.

Подробное описание метода `setdefault(<Ключ>[, <Значение по умолчанию>])`:

- если ключ присутствует в словаре, то метод возвращает значение, соответствующее этому ключу
- если ключ отсутствует, то вставляет новый элемент, со значением, указанным во втором параметре
- если второй параметр не указан, значением нового элемента будет `None`.

Пример:

```
[>>> example_dict = {5: 6, 8: 5, 3: 7}
[>>> example_dict.setdefault(5, "koshka")
6
[>>> example_dict
{5: 6, 8: 5, 3: 7}
[>>> example_dict.setdefault(9, "koshka")
'koshka'
[>>> example_dict
{5: 6, 8: 5, 3: 7, 9: 'koshka'}
[>>> ]
```

Подробное описание метода `get(<Ключ> [, <Значение по умолчанию>])`.

- если ключ присутствует в словаре, то метод возвращает значение, соответствующее этому ключу
- если ключ отсутствует, то возвращается значение `None` или значение, указанное во втором параметре.

Пример:

```
[>>> example_dict = {5: 6, 8: 5, 3: 7}
[>>> example_dict.get(3)
7
[>>> example_dict.get(7)
None
```

Операции многоэлементного изменения словарей.

**dict.clear()** - очищает словарь.

**dict.fromkeys(seq[, value])** - создает словарь с ключами из seq и значением value (по умолчанию None):

```
locale_list = list(range(1,4))
locale_dict = dict.fromkeys(locale_list, "value")
print(locale_dict)
#{1: 'value', 2: 'value', 3: 'value'}
```

**dict.update([other])** - обновляет словарь, добавляя пары (ключ, значение) из other.

Существующие ключи перезаписываются. Возвращает None (не новый словарь!).

```
d = {1: "one", 2: "three"}
d1 = {2: "two"}
d.update(d1)
print(d)
#{1: 'one', 2: 'two'}
```

### Операции поэлементного извлечения из словаря и их использование.

К ним относятся уже упомянутые get и setdefault, а также следующие методы:

**dict.pop(key[, default])** - удаляет ключ и возвращает значение. Если ключа нет, возвращает default (по умолчанию бросает исключение).

```
d = {"cat" : 5, 1: "one", 2: "two"}
d.pop("cat") #5
print(d)
```

**dict.popitem()** - удаляет и возвращает пару (ключ, значение), которая была последняя добавлена в словарь. Если словарь пуст, то бросает исключение KeyError.

```
d = {"cat" : 5, 1: "one", 2: "two"}
print(d.popitem()) #(2, 'two')
print(d)
```

### Дополнительно:

Обход упорядоченных ключей:

```
for k in sorted(d13.keys()):
    print('key: {}, value: {}'.format(k, d13[k]))
```

Обход значений словаря:

```
for v in d13.values():
    print('value: {}'.format(v))
```