# Решение задач на словари



## Аннотация

*В этом уроке рассматриваются методы словарей и допустимые типы ключей.*

## 1.Другие методы словарей

С различными методами словарей можно подробнее познакомиться в документации.
https://docs.python.org/3.6/library/stdtypes.html#mapping-types-dict

| Операция | Описание | Пример |
|---|---|---|
| `d.clear()` | Удаление всех элементов словаря d | `d.clear()`<br>`print(d)`<br><br>`{}` |
| `d.copy()` | Создается независимая копия словаря d | `d = {'key_1': 1, 'key_2': 2}`<br>`d1 = d.copy()`<br>`d1['key_1'] = 3`<br>`print(d)`<br>`print(d1)`<br><br>`{'key_1': 1, 'key_2': 2}`<br>`{'key_1': 3, 'key_2': 2}` |

| | | |
|---|---|---|
| `d.popitem()` | Удаляет и возвращает пару (ключ, значение) из словаря. Если словарь пуст, то будет вызвано исключение KeyError | `d = {'key_1': 1, 'key_2': 2}`<br>`print(d.popitem())`<br>`print(d)`<br><br>`('key_2', 2)`<br>`{'key_1': 1}` |
| `d.setdefault(key[, default])` | Если ключ key есть в словаре, то возвращается значение по ключу. Если такого ключа нет, то в словарь вставляется элемент с ключом key и значением default, если default не определен, то по умолчанию присваивается None | `d = {'key_1': 1, 'key_2': 2}`<br>`print(d.setdefault('key_3', 42))`<br>`print(d)`<br><br>`42`<br>`{'key_1': 1, 'key_2': 2, 'key_3': 42}` |
| `d.update([other])` | Обновить словарь парами (key, value) из other, если ключи уже существуют, то обновить их значения | `d = {'key_1': 1, 'key_2': 2}`<br>`d.update({'key_1': 1984, 'key_3': 42})`<br>`print(d)`<br><br>`{'key_1': 1984, 'key_2': 2, 'key_3': 42}` |

## 2.Допустимые типы ключей

Мы уже выяснили, что ключами в словарях могут быть строки и целые числа. Кроме этого, ключами могут быть вещественные числа и кортежи.

**Ключи в словаре**

Ключами в словаре не могут быть другие словари. В принципе в одном словаре могут быть ключи разных типов, однако обычно принято использовать однотипные ключи.

Вообще, есть строгий способ определить, может ли объект быть ключом в словаре. Для этого объект должен быть **неизменяемым**. Неизменяемые объекты не могут поменять значение в себе во время выполнения программы. Неизменяемыми в Python являются числа, строки и кортежи. Именно их обычно и используют в качестве ключей словарей.

Вот как может выглядеть словарь с ключами-кортежами. В качестве ключа используются координаты, а в качестве значения — название города.

```
cities = {
    (55.75, 37.5): 'Москва',
    (59.8, 30.3): 'Санкт-Петербург',
    (54.32, 48.39): 'Ульяновск'
}
print(cities[(55.75, 37.5)])
cities[(53.2, 50.15)] = 'Самара'
```

Возможно, нам захочется развернуть этот словарь, то есть построить такой, в котором ключами будут города, а значениями — их координаты.

```
coordinates = {}
for coordinate, city in cities.items():
    coordinates[city] = coordinate
```

Если в исходном словаре были повторяющиеся значения, некоторые из них потеряются при разворачивании словаря. Это объясняется тем, что значения в словаре могут повторяться, а вот ключи обязаны быть уникальными.

Значениями в словаре, в отличие от ключей, могут быть объекты любого типа — числа, строки, кортежи, списки и даже другие словари. Вот, например, как можно сохранить список фильмов для каждого из актеров:

```
films = {
  'Джонни Депп': [
      'Эдвард Руки-Ножницы',
      'Одинокий рейнджер',
      'Чарли и шоколадная фабрика',
      ..],
  'Эмма Уотсон': [
      'Гарри Поттер и философский камень',
      'Красавица и Чудовище',
      ..],
  # ...
}

# Вывести список фильмов, в которых
снималась Эмма Уотсон
print(films['Эмма Уотсон'])

# Проверить, снимался ли Джонни Депп в
фильме «Чарли и шоколадная фабрика»
if 'Чарли и шоколадная фабрика' in
films['Джонни Депп']:
    print('Снимался!')
```