# Урок 15.05. Генерация словарей

Ранее мы уже говорили, что в словарях в качестве значений могут храниться любые данные, как и внутри списка. Иногда бывает так, что список хранит несколько словарей, с которыми нужно как-то работать.

Задача «Журнал посещений».

У нас есть список людей, которые посещали страницу нашего сайта. В этом списке хранятся id и name для каждого пользователя. Некоторые из них заходили на сайт несколько раз, поэтому в журнале есть одинаковые записи. Нам нужно удалить из журнала повторы или оставить только уникальные записи. Для этого мы можем применить множества:

data = [

 {'id': 10, 'name': 'Bob'},

 {'id': 11, 'name': 'Misha'},

 {'id': 12, 'name': 'Anton'},

 {'id': 10, 'name': 'Bob'},

 {'id': 11, 'name': 'Misha'},

]

uniq\_data = set(data)

print(uniq\_data)

Запустим и… Ничего не сработало. Ведь элементами множества не может быть словарь. Давайте решим задачу обычным способом.

data = [

 {'id': 10, 'name': 'Bob'},

 {'id': 11, 'name': 'Misha'},

 {'id': 12, 'name': 'Anton'},

 {'id': 10, 'name': 'Bob'},

 {'id': 11, 'name': 'Misha'},

]

uniq\_data = []

for i\_dict in data:

 data\_exists = False

 for uniq\_dict in uniq\_data:

 if uniq\_dict.get('id') == i\_dict.get('id'):

 data\_exists = True

 break

 if not data\_exists:

 uniq\_data.append(i\_dict)

print(uniq\_data)

На самом деле эта запись должна была вам напомнить работу генерации списков в list comprehansion. Давайте попробуем его применить, но только для словаря:

data = [

 {'id': 10, 'name': 'Bob'},

 {'id': 11, 'name': 'Misha'},

 {'id': 12, 'name': 'Anton'},

 {'id': 10, 'name': 'Bob'},

 {'id': 11, 'name': 'Misha'},

]

uniq\_data = []

for i\_dict in data:

 data\_exists = False

 for uniq\_dict in uniq\_data:

 if uniq\_dict.get('id') == i\_dict.get('id'):

 data\_exists = True

 break

 if not data\_exists:

 uniq\_data.append(i\_dict)

print(uniq\_data, '\n')

uniq\_data\_dict = {i\_dict['id']: i\_dict for i\_dict in data}

print(uniq\_data\_dict.values())

Таким образом, генерация словарей используется аналогичная логика как и для списков.