

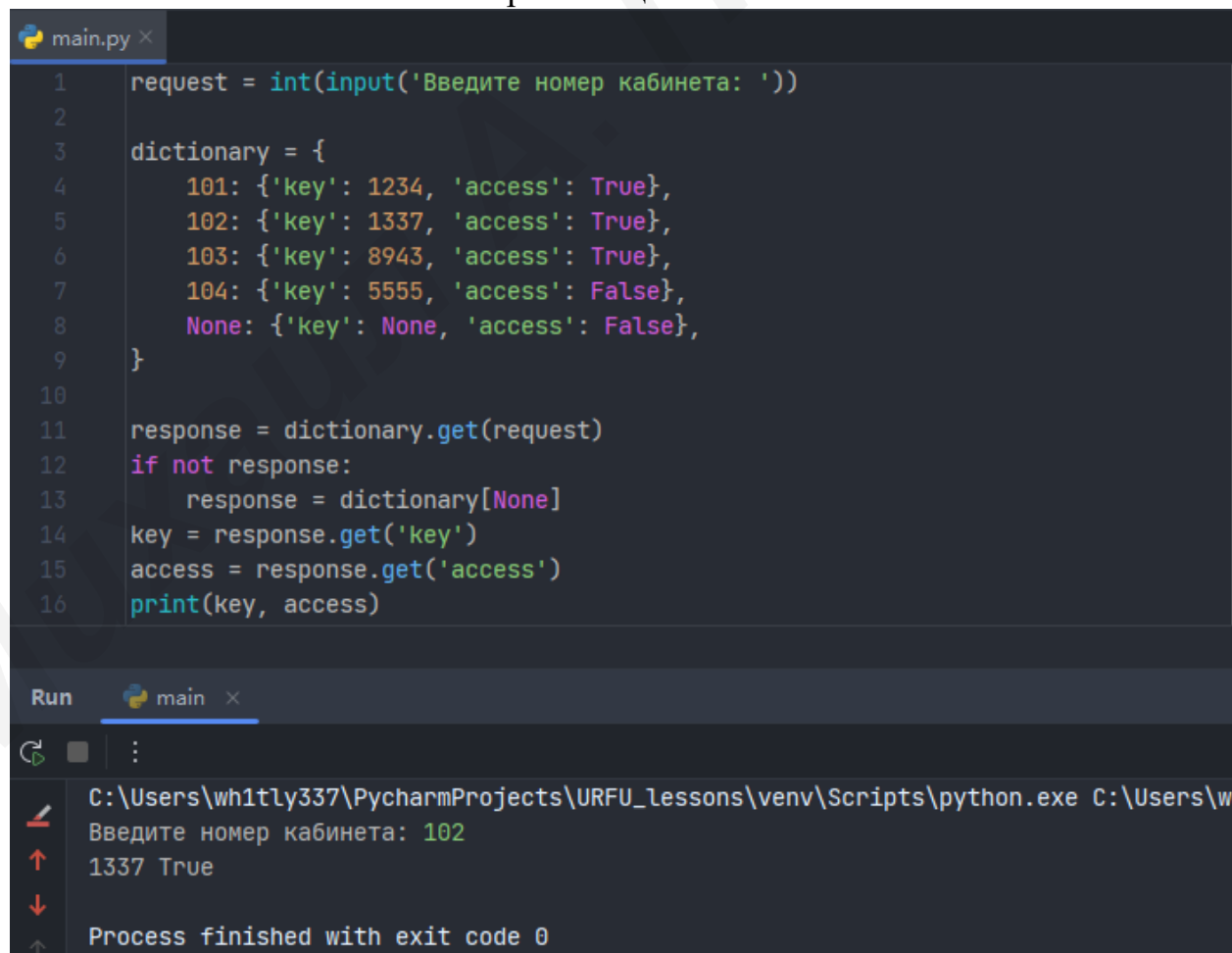
Лабораторная работа 6

ТЕМА 6. Базовые коллекции: словари, кортежи

Лабораторные задания:

- 1) В школе, где вы учились, узнали, что вы крутой программист и попросили написать программу для учителей, которая будет при вводе кабинета писать для него ключ доступа и статус, занят кабинет или нет. При написании программы необходимо использовать словарь (dict), который на вход получает номер кабинета, а выводит необходимую информацию. Если кабинета, который вы ввели нет в словаре, то в консоль в виде значения ключа нужно вывести "None" и в виде статуса вывести "False".

По большому счету написав данную программу мы с вами научились заменять иногда громоздкую конструкцию if/elif/else. Поскольку здесь функционал словаря полностью повторяет функционал условия, но при этом у использования словарей в более сложных программах есть намного больше возможностей реализации.



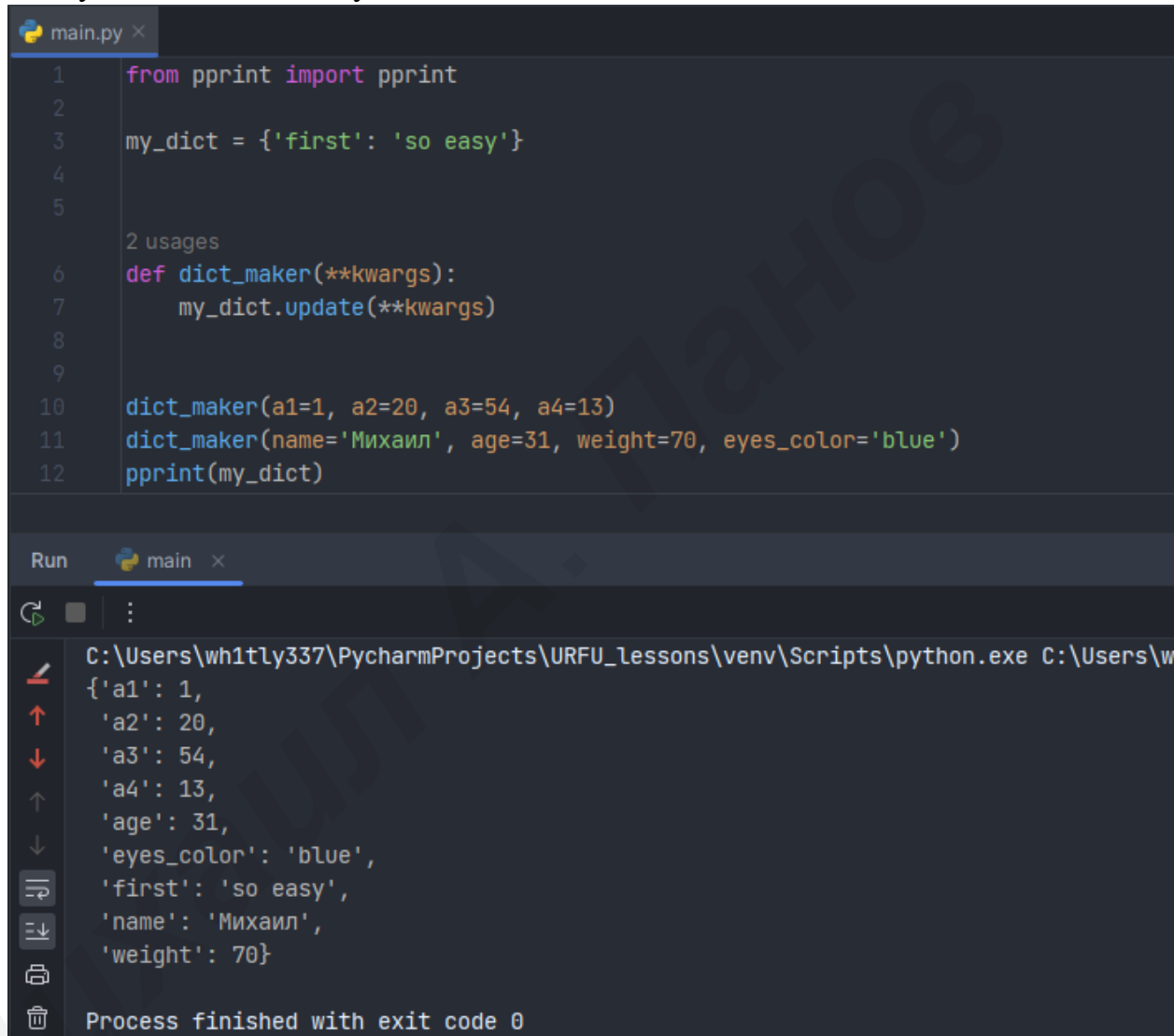
```
main.py x
1 request = int(input('Введите номер кабинета: '))
2
3 dictionary = {
4     101: {'key': 1234, 'access': True},
5     102: {'key': 1337, 'access': True},
6     103: {'key': 8943, 'access': True},
7     104: {'key': 5555, 'access': False},
8     None: {'key': None, 'access': False},
9 }
10
11 response = dictionary.get(request)
12 if not response:
13     response = dictionary[None]
14 key = response.get('key')
15 access = response.get('access')
16 print(key, access)

Run main x
C:\Users\wh1tly337\PycharmProjects\URFU_lessons\venv\Scripts\python.exe C:\Users\w
Введите номер кабинета: 102
1337 True
Process finished with exit code 0
```

- 2) Алексей решил создать самый большой словарь в мире. Для этого он придумал функцию dict_maker (**kwargs), которая принимает неограниченное количество параметров «ключ: значение» и обновляет

созданный им словарь `my_dict`, состоящий всего из одного элемента «first» со значением «so easy». Помогите Алексею создать данную функцию.

Ниже на скриншоте мы использовали встроенный модуль `pprint`, который выводит большие объемы информации более понятно для восприятия человеческим глазом. Иногда очень удобно использовать данную возможность Python.



The screenshot shows a Python IDE with a file named `main.py`. The code defines a dictionary `my_dict` and a function `dict_maker` that updates it with keyword arguments. It then calls `dict_maker` with various arguments and prints the dictionary using `pprint`. The output window shows the resulting dictionary with all key-value pairs, including the original 'first' key and the new ones like 'a1', 'a2', 'a3', 'a4', 'age', 'eyes_color', 'name', and 'weight'.

```
1 from pprint import pprint
2
3 my_dict = {'first': 'so easy'}
4
5
6 2 usages
7 def dict_maker(**kwargs):
8     my_dict.update(**kwargs)
9
10 dict_maker(a1=1, a2=20, a3=54, a4=13)
11 dict_maker(name='Михаил', age=31, weight=70, eyes_color='blue')
12 pprint(my_dict)
```

```
Run main
C:\Users\wh1tly337\PycharmProjects\URFU_lessons\venv\Scripts\python.exe C:\Users\w
{'a1': 1,
 'a2': 20,
 'a3': 54,
 'a4': 13,
 'age': 31,
 'eyes_color': 'blue',
 'first': 'so easy',
 'name': 'Михаил',
 'weight': 70}
Process finished with exit code 0
```

- 3) Для решения некоторых задач бывает необходимо разложить строку на отдельные символы. Мы знаем что это можно сделать при помощи `split()`, у которого более гибкая настройка для разделения для этого, но если нам нужно посимвольно разделить строку без всяких условий, то для этого мы можем использовать кортежи (`tuple`). Для этого напомним любую строку, которую будем делить и “обвернем” ее в `tuple` и дальше мы можем как нам угодно с ней работать, например, сделать ее списком (тогда получится полный аналог `split()`) или же работать с ним дальше, как с кортежем.

```
main.py ×
1 input_string = 'HelloWorld'
2 result = tuple(input_string)
3 print(result)
4 print(list(result))

Run main ×
C:\Users\wh1tly337\PycharmProjects\URFU_lessons\venv\Scripts\python.exe C:\Users\w
('H', 'e', 'l', 'l', 'o', 'W', 'o', 'r', 'l', 'd')
['H', 'e', 'l', 'l', 'o', 'W', 'o', 'r', 'l', 'd']
Process finished with exit code 0
```

- 4) Вовочка решил написать крутую функцию, которая будет писать имя, возраст и место работы, но при этом на вход этой функции будет поступать кортеж. Помогите Вовочке написать эту программу.

```
main.py ×
2 usages
1 def personal_info(name, age, company='unnamed'):
2     print(f"Имя: {name} Возраст: {age} Компания: {company}")
3
4
5 tom = ("Григорий", 22)
6 personal_info(*tom)
7
8 bob = ("Георгий", 41, "Yandex")
9 personal_info(*bob)

Run main ×
C:\Users\wh1tly337\PycharmProjects\URFU_lessons\venv\Scripts\python.exe C:\Users\w
Имя: Григорий Возраст: 22 Компания: unnamed
Имя: Георгий Возраст: 41 Компания: Yandex
Process finished with exit code 0
```

- 5) Для сопровождения первых лиц государства X нужен кортеж, но никто не может определиться с порядком машин, поэтому вам нужно написать функцию, которая будет сортировать кортеж, состоящий из целых чисел по возрастанию, и возвращает его. Если хотя бы один элемент не является целым числом, то функция возвращает исходный кортеж.

```
main.py x
2 usages
1 def tuple_sort(tpl):
2     for elm in tpl:
3         if not isinstance(elm, int):
4             return tpl
5     return tuple(sorted(tpl))
6
7
8 if __name__ == '__main__':
9     print(tuple_sort((5, 5, 3, 1, 9)))
10    print(tuple_sort((5, 5, 2.1, '1', 9)))

Run main x
[icon] [icon] [icon]
/usr/local/bin/python3.11 /Users/user/PycharmProjects/URFU_lessons/main.py
(1, 3, 5, 5, 9)
(5, 5, 2.1, '1', 9)
Process finished with exit code 0
```