

Лабораторная 2. Кортежи, словари, множества, функции.

По желанию можно сразу перейти к [задачам посложнее](#), те, кто делает часть 1 может не делать часть 2, это по желанию.

ЧАСТЬ 1. Простые задания

Кортежи

1. Создайте кортеж, содержащий рандомные числа. Выведите мин, макс, сумму и среднее этих чисел.
2. Преобразуйте кортеж из первого задания в список. Выведите размеры в байтах списка и кортежа (метод `__sizeof__()`). Какой тип данных занимает меньше места?
3. Задан кортеж с вложенным списком: `tuple_1 = (11, 12, 13, [10, 20, 30])`. Выведите первый и последний элементы вложенного в кортеж списка.
4. Распакуйте кортеж из 3-го задания (должно получиться 3 переменных и словарь).

Словари `dictionary = { ключ1:значение1, ключ2:значение2, }`

5. Создайте словарь сотрудников, содержащий фамилии и зарплаты. Подумайте, что будет ключом, а что - значением.
 - Найдите значение по заданному ключу.
 - Найдите значение по несуществующему ключу, для исключения ошибки используйте метод `get()`.
 - Добавьте в словарь нового сотрудника с его зарплатой.
 - Выведите количество сотрудников в словаре.
 - Выведите список фамилий сотрудников.
 - Рассчитайте среднюю зарплату в компании.

Множества

6. Даны два заказа, выведите повторяющиеся и уникальные позиции:

```
order_1 = 'apple', 'orange', 'banana'
order_2 = 'apple', 'pear', 'orange'
```

- объедините эти заказы в один, так что бы продукты не повторялись.

Функции

7. Напишите функцию `square()`, принимающую 1 аргумент — сторону квадрата, и возвращающую 3 значения (с помощью кортежа): периметр квадрата, площадь квадрата и диагональ квадрата.
8. Напишите функцию `is_year_leap()`, принимающую 1 аргумент — год, и возвращающую `True`, если год високосный, и `False` иначе.

Желательно учесть, что последние годы любого столетия (1700-й, 1800-й, 1900-й и так далее) являются високосными, только если делятся на 400. Для проверки вызовите функцию с годами 2000 (високосный) и 2100 (не високосный).

ЧАСТЬ 2. Задания посложнее

Задание 1:

Напишите функцию, которая берет текст, подсчитывает частоту встречаемости каждого слова и выводит словарь, где ключи — слова, а значения — количество вхождений.

Задание 2:

Есть лог-файл сервера, представленный в виде списка кортежей:

```
logs = [  
    ("192.168.1.1", "200 OK", 1543),  
    ("192.168.1.2", "404 Not Found", 234),  
    ("192.168.1.1", "500 Internal Server Error", 542),  
    ("192.168.1.3", "200 OK", 876),  
    ("192.168.1.2", "200 OK", 1324),  
]
```

Напишите программу, которая:

- подсчитывает количество запросов от каждого IP;
- определяет самый частый HTTP-статус;
- вычисляет общий объем переданных данных.

Задание 3:

Даны три множества пользователей разных сервисов:

```
service_a = {"Анна", "Иван", "Мария", "Сергей", "Алексей"}  
service_b = {"Мария", "Иван", "Дмитрий", "Ольга", "Светлана"}  
service_c = {"Сергей", "Ольга", "Александр", "Иван", "Анна"}
```

Напишите программу, которая:

- определяет пользователей, зарегистрированных во всех трех сервисах;
- определяет пользователей, присутствующих только в одном из сервисов;
- определяет, у какого сервиса самая большая уникальная база пользователей.

Задание 4:

Есть список задач, хранящийся в словаре и содержащий не менее 10ти задач, для каждой из них определен дедлайн (дата сдачи):

- выведите список задач с истекшим дедлайном;
- выведите список задач до окончания которых осталось меньше 3-ёх дней;
- реализуйте добавление новой задачи с проверкой на уникальность названия задачи и валидацией даты.