Урок 2

Цели урока:
1. *Закрепить прошедшие темы*
2. *Изучить коллекции*

***
Задачи на закрепление

Задача 1:

Перемножить все не чётные значения в диапазоне от 0 до 9435;
```
result = 1
for i in range(9435):
    if i % 2 != 0:
        result *= i
print('Результат равен ' + str(result))
```

Задача 2:

Напишите проверку на то, является ли строка палиндромом. Палиндром — это слово или фраза, которые одинаково читаются слева направо и справа налево.
```
word = input('Введите слово: ')
if word == word[::-1]:
    print(f'{word} является палиндромом')
else:
    print(f'{word} не является палиндромом')
```

Задача 3:

Пользователь вводит числа, до тех пор пока сумма этих чисел не будет равна 0.После этого выводится сумма квадратов данных чисел
```
summa = 0
kv_summa = 0
while True:
    number = int(input('Введите число: '))
    summa += number
    kv_summa += number ** 2
    if summa == 0:
        break
print(kv_summa)
```

***
Коллекции
Коллекции - это типы данных контейнера которые можно использовать для хранения данных. Коллекции могут хранить списки, наборы, кортежи и словари. Каждый из этих типов данных имеет свои собственные характеристики.

Тип данных	Изменяемость	Индексированность	Уникальность	Создание

| Список | да | да | нет | list()
|
| Кортеж | нет | да | нет | tuple()
|
| Множество | да | нет | да | set()
|
| Словарь | да | нет | да* | dict()
|

Списки

Список может хранить n-е количество элементов разных типов(строки, числа, другие типы коллекций и т.п)

Для создания списка достаточно поместить необходимые элементы в квадратные скобки []
```
a = [1, 2, 3, 4]
a = [1, 2, 'hello', [3, 4]]
a = []     # создание пустого списка
a = list() # создание пустого списка
```
Списки - изменяемый тип данных, т.е в списке можно добавить, удалить, изменить элемент. Каждый элемент списка имеет свой индекс(порядковый номер элемента). Индексация в Python начинается с 0, т.е первый элемент списка имеет индекс 0, второй 1 и т.д

К каждому элементу списка можно обратиться по индексу:
```
print(a[0]) # вывод первого элемента в консоль
a[0] = 5    # изменение значения первого элемента
```
Перебрать список по элементам, можно с помощью циклов:
```
for elem in a:
    print(elem)
```
```
for i in range(len(a)):
    print(a[i])
```
```
i = 0
while i < len(a):
    print(a[i])
    i += 1
```
Методы списков:
* *append* - добавляет в конец списка новый элемент
```
a.append(7)  # добавит в конец списка 7
```
* *remove* - удаляет элемент списка по значению
```
a.remove('hello')  # удалит элемент из списка
```
* *pop* - удаляет элемент из списка по индексу
```
a.pop(2)  # удалит элемент под индексом 2
b = a.pop(2)  # удалит элемент и создаст новый объект с данным значением
```

* *index* - находит индекс указанного элемента
```
a.index('hello')  # найдет индекс элемента
```
* *insert* - вставит элемент в список в указанный индекс
```
a.insert(1, 'buy')  # вставит 'buy' под индексом 1
```
* *count* - находит количество повторов элемента
```
a.count('buy')  # найдет количество повторов 'buy'
````
* *clear* - очистит список
```
a.clear()  # очистит список
```
* *sort* - отсортирует список
```
a.sort()  # отсортирует список
```
* *copy* - создаст независимую копию списка
```
b = a.copy()  # создаст копию списка a
```

Задача 1:

Даны списки:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89];

b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13].

Нужно вернуть список, который состоит из элементов, общих для этих двух списков.

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
c = []
for elem in a:
    if elem in b and elem not in c:
        c.append(elem)
print(c)
```

Задача 2:

Вы принимаете от пользователя последовательность чисел, разделённых запятой. Составьте список из этих чисел

```
a = []
numbers = input('Введите числа: ').split(',')
for i in range(len(numbers)):
    numbers[i] = int(numbers[i])
print(numbers)
```
Задача 3:

Напишите программу, которая принимает два списка и добавляет в третий список все общие элементы первого и второго списков, которые в количестве более 2.

```
a = [8, 1, 1, 2, 3, 5, 8, 8, 13, 21, 34, 55, 89]
b = [1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 8, 11, 12, 13]
c = []
for i in a:
    if a.count(i) > 1 and b.count(i) > 1 and i not in c:
        c.append(i)
print(sorted(c))
```

Кортежи

Кортежи очень похожи на списки.По сути это такие же списки, только они неизменяемые.
Методы кортежей и списков совпадают, за исключением методов изменяющих кортеж
Для создания кортежа достаточно поместить необходимые элементы в круглые скобки ()

```
a = (1, 2, 3, 4)
a = (1, 2, 'hello', [3, 4])
a = (1, )     # создание кортежа из одного элемента
a = tuple() # создание пустого кортежа
```

У кортежей есть такая особенность, как распаковка.
Пример:

```
a = (1, 2)
b, c = a
```

Благодаря данной особенности мы можем поменять значение переменных местами:

```
a = 5
b = 6
a, b = b, a
```