

прога 19  
сентября

**коллекции и строки**

# наборы штук в питоне

Часто в переменную запомнена не одна штука, а целый их набор

Например: последовательность пользовательских ответов на вопросы; набор упомянутых слов; продукты вместе с ценами.

Есть **типовые наборы** штук, они **называются коллекциями**

Особняком стоят **строки**: это **не коллекции**

# КОЛЛЕКЦИИ В ПИТОНЕ

Какие есть?

# Кортеж и список

[1,2,3] и (1,2,3)

В чём разница?

# Задача

Как обменять значения у двух переменных?

# Список

```
>>> guests = ['Вася', 'Татуся',  
'Пётр Яковлевич', 'мама'] #  
список строк
```

```
>>> digits = [1, 2, 3, 4, 5, 6,  
7, 8, 9] # список чисел
```

```
>>> empty_list = [] # а этот  
список абсолютно пуст
```

```
>>> mixed_list = ['Вася', 45,  
True] # в одном списке могут  
храниться элементы разных типов
```

```
>>> matrix = [[1, 2, 3], [4, 5,  
6], [7, 8, 9]]
```

# Список

Списки можно изменять

Списки НЕЛЬЗЯ положить в  
словарь и множество

Списки повсюду

Какие вы знаете операции со  
списками?



# Срезы

А помните, в `range(2, 8, 5)` можно было вот три параметра передать — начало, конец и длину шага? Вот из списка можно так вытаскивать подпоследовательности:

```
my_cool_li[START:END:ШАГ]
```

А ещё так можно из строк вытаскивать. Называется это срезами.

# Словарь (dictionary) —

неупорядоченная структура  
данных для хранения пар  
<ключ: значение>

- Добавить/обновить значение - по ключу
- Удалить - например через `.pop()`

`dict.keys()` - вернет ключи

`dict.values()` - вернет значения

`dict.items()` - вернет пары ключ:  
значение

**Считайте, что значения словаря не  
упорядочены — используйте  
`sorted()`, если важен порядок**

# Задача

Про текстовый документ  
посчитайте, сколько раз в нём  
встретилась какая буква из этих  
"abcdefg"

Пользуйтесь словарями!

# Словарь

1. `dict()`
2. `defaultdict`  
`(from collections)`
3. `Counter`  
`(from collections)`

Давайте найдём в гитхабе  
пример использования каждой  
из штук

# Множество

Принципиальное отличие множества от списка - отсутствие повторяющихся элементов.

Иногда это нужно с точки зрения формулировки, иногда с точки зрения результата.

Синтаксис - встроенная функция `set`

```
lst = [1,2,2,3,5,3]
```

```
set(lst)
```

```
{1,2,3,5}
```

# Множество

Поддерживают математику  
множеств

```
a = {1,2,3}
```

```
b = {1,3,5}
```

```
a.union(b)
```

```
> {1,2,3,5}
```

```
a.intersection(b)
```

```
> {1,3}
```

# split() и splitlines()

`split` разбивает строку по символам-разделителям (пробел, табуляция, `\n`, ...)

`splitlines` разбивает строку по символам-разделителям СТРОК (это `\n`, но не только)

```
>>> a = "pi pa"
```

```
>>> a.split()
```

```
['pi', 'pa']
```

```
>>> a = "pi\npa po"
```

```
>>> print(a)
```

```
pi
```

```
pa po
```

```
>>> a.splitlines()
```

```
['pi', 'pa po']
```

# Ещё прикольные методы строк

```
>>>  
"heLLo".upper()  
'HELLO'
```

```
>>>  
"hElLo".lower()  
'hello'
```

```
>>>  
"hElLo".isalpha()  
True
```

```
>>>  
"hElLo111".isalpha()  
False
```

```
>>>  
"hElLo".strip('o')  
'hElL'
```

```
>>>  
"hElLo".strip('ho')  
'EL'
```

```
>>>  
"123".isdigit()  
True
```

```
>>>  
"12a3".isdigit()  
False
```



# Задача

Проверьте, является ли строка палиндромом (читается в обе стороны одинаково)

# генерация и лист компрехеншонс

[ссылка](#)

# генерация и лист компрехеншонс

```
squares = []  
  
for i in range(11):  
    squares.append(i*i)
```

```
squares = [i*i for i in  
range(11)]
```

# Задача

часто в задачах такой ввод:  
сначала пользователь вводит  
число  $N$ , а потом  $N$  раз какие-то  
данные. запишите в одну строку  
считывание такого ввода.

как не надо: не  
модифицируйте то, с чем  
прямо сейчас работаете

```
li = [3, 4, 56, 7, 10, 9,  
6, 5]
```

```
for i in li:  
    if not i % 2 == 0:  
        continue  
    li.remove(i)  
  
print(li)
```

как не надо: не  
модифицируйте то, с чем  
прямо сейчас работаете

```
d = {k: k for k in  
range(10) }
```

```
for k, v in d.items():  
    if not v % 2 == 0:  
        continue  
    d.pop(k)
```

# Найдите ошибки в коде

```
product = 0
```

```
for i in range(10):
```

```
    product *= i
```

```
sum_squares = 0
```

```
for i in range(10):
```

```
    i_sq = i**2
```

```
sum_squares += i_sq
```

```
nums = 0
```

```
for num in range(10):
```

```
    num += num
```

```
dividend = float(input("enter dividend:"))
```

```
divisor = float(input("enter divisor: "))
```

```
quotient = dividend / divisor
```

```
quotient_rounded = math.round(quotient)
```

# сортировка, минимум и максимум

sort, sorted, min, max

едят ключ

Как сортировать словарь?



функции

# как выглядит функция

Что обязательно есть в функции?

первая строка:  
определение функции

ИМЯ

аргументы

типы аргументов

последняя строка:  
ретёрн

если ретёрна нет, что вернётся?  
как вернуть несколько штук?

аргументы,  
необязательные  
аргументы,  
глобальные  
переменные,  
документация

# задача

напишите красивую функцию,  
которая бьёт текст на слова

- с докстрингой
- с типами аргументов

трюк с запоминанием  
функции в  
переменную

# лямбда функции

пример: есть у нас список  
кортежей, сортируем только по  
второму элементу



**СЛОВО**

# T9: как понять, похожи ли два слова?

задача: напишите функции, которые скажут, можно ли из одного слова получить другое

1. дописав в нём куда-то букву
2. заменив лишь одну букву
3. заменив одну букву или заменив одну цифру или и то и другое
4. заменив две буквы