



# Строки и их методы

## Занятие 1



## Цель

познакомить со строковым типом данных и научить применять функции и методы при решении задач.

## Задачи

- Познакомиться со строковым типом данных.
- Разобрать синтаксис функций `len()` и `str()`.
- Познакомиться с основными операциями со строковыми данными.
- Разобрать на примерах методы работы со строками.



# Разминка

- Какой цикл использовать если вы знаете количество повторов?
- Какой цикл использовать если вы не знаете количество повторов?
- Что позволяет сделать оператор `break`?

# Найди ошибку

```
for in range(10):  
    print("Hello")
```

```
for in range(10):  
print("Hello")
```

```
1 i = 0  
2 while i > 0  
3     print(i)
```

```
i = 0  
while i < 0  
    print(i)
```

# Что будет результатом программы:

```
i = 0
while i < 10:
    print(i*2)
    i += 2
```

```
i = 0
while i != 5:
    if i%2==0:
        print(i)
        i +=1
```

# Строки в Python

Строка – это сложный тип данных, представляющий собой последовательность символов.

```
>>> a = 'Python'  
>>> b = 'Эквилибристика'
```

П	Р	И	В	Е	Т
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1



## Функции `str()` и `len()`

`str()` — переход к строковому типу данных

`input()` — всегда возвращает строку, поэтому нет смысла использовать `str()`

```
>>> print(len('эквилибристика'))  
14
```

# Действия со строками

- Конкатенация строк

```
>>> a = 'Почему '  
>>> b = 'коровы '  
>>> c = ' не летают? '  
>>> print(a + b + c)  
Почему коровы не летают?
```

- Умножение строки на число

```
>>> a = 'Нас не догонят! '  
>>> b = a * 3  
>>> b  
'Нас не догонят! Нас не догонят! Нас не догонят! '
```

# Действия со строками

- Оператор in

```
s = 'программирование'
if 'a' in s:
    print('Введенная строка содержит символ a')
else:
    print('Введенная строка не содержит символ a')
```

- Срезы строк

```
>>> x = 'Python'
>>> x[0:3]
'Pyt'
>>> x[:3]
'Pyt'
>>> x[-4:]
'thon'
>>> x[:]
'Python'
```

```
>>> x = 'Престиж'
>>> x[0:7:2]
'Петж'
>>> x[::2]
'Петж'
>>> x[::3]
'Псж'
```

# Методы работы со строками

- 1) Преобразование строк
- 2) Классификация строк
- 3) Конвертация регистра
- 4) Поиск, подсчет и замена символов

<i>Функция или метод</i>	<i>Назначение</i>
<code>str in S</code>	Проверка на входение подстроки в строку
<code>S.find(str, [start],[end])</code>	Поиск подстроки в строке. Возвращает номер первого входения или -1
<code>S.rfind(str, [start],[end])</code>	Поиск подстроки в строке. Возвращает номер последнего входения или -1
<code>S.replace(шаблон, замена)</code>	Замена
<code>S.isdigit()</code>	Состоит ли строка из цифр
<code>S.isalpha()</code>	Состоит ли строка из букв
<code>S.isalnum()</code>	Состоит ли строка из цифр или букв
<code>S.islower()</code>	Состоит ли строка из символов в нижнем регистре
<code>S.isupper()</code>	Состоит ли строка из символов в верхнем регистре
<code>S.istitle()</code>	Начинаются ли слова в строке с заглавной буквы
<code>S.upper()</code>	Преобразование строки к верхнему регистру
<code>S.lower()</code>	Преобразование строки к нижнему регистру
<code>S.join(список)</code>	Сборка строки из списка с разделителем S
<code>S.split(символ)</code>	Разбиение строки по разделителю
<code>ord(символ)</code>	Символ в его код ASCII
<code>chr(число)</code>	Код ASCII в символ
<code>S.capitalize()</code>	Переводит первый символ строки в верхний регистр, а все остальные - в нижний
<code>S.strip([chars])</code>	Удаление пробельных символов в начале и в конце строки
<code>S.swapcase()</code>	Переводит символы нижнего регистра в верхний, а верхнего - в нижний
<code>S.title()</code>	Первую букву каждого слова переводит в верхний регистр, а все остальные - в нижний



# Практика решения задач

## Уровень А

- **Задача 1.** Составьте программу замены всех букв «а» на буквы «о».
- **Задача 2.** Составьте программу, чтобы в заданном тексте подсчитать слова, начинающиеся на «до» («дорожники долг достраивали дорогу»).
- **Задача 3.** Напишите программу, которая в заданном предложении удаляет все символы пробела. Напечатайте исходный и преобразованный тексты, а также количество удаленных символов.
- **Задача 4.** Составить программу, которая проверяет, является ли введенное слово «перевертышем», т.е. читающееся одинаково слева направо и справа налево. Например, «шалаш», «кабак».
- **Задача 5.** Напишите программу, которая в заданном предложении находит самое короткое и самое длинное слова. Напечатайте исходный текст и найденные слова.

# Практика решения задач

## Уровень В

- **Задача1.** Составьте программу замены всех букв «а» на буквы «о». Модифицируйте программу так, чтобы происходила замена буквы «а» на букву «о» и наоборот. Например, слово «баба» программа превращала бы в «абаб». Сделайте то же с учетом заглавных букв.
- **Задача2.** Предположим, что в заданном тексте необходимо подсчитать слова, начинающиеся на «до» («дорожники долг достраивали дорогу»). Выясните путем эксперимента, можно ли эту задачу решить с помощью метода find.
- **Задача 3.** Напишите программу, которая в заданном предложении перед каждым словом вставляет символ «?». Напечатайте преобразованный текст и найденные слова.
- **Задача 4.** Составить программу, которая проверяет, является ли введенное предложение «перевертышем», т.е. читающееся одинаково слева направо и справа налево (без разделителей, разумеется). Например, «аргентина манит негра», «а роза упала на лапу азора».
- **Задача 5.** Составить программу, в которой вводится двузначное число и выводится его название (вводится «36» - выводится «тридцать шесть»).

# Практика решения задач

## Уровень С

- **Задача1.** Подсчитайте, сколько пятерок потребуется, чтобы записать все числа от 1 до 999.

- **Задача2.**

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах  $v$  и  $w$  обозначают цепочки символов.

1. заменить ( $v, w$ )

2. нашлось ( $v$ )

Первая команда заменяет в строке первое слева вхождение цепочки  $v$  на цепочку  $w$ . Если цепочки  $v$  в строке нет, эта команда не изменяет строку. Вторая команда проверяет, встречается ли цепочка  $v$  в строке исполнителя Редактор.

Дана программа для исполнителя Редактор:

НАЧАЛО

ПОКА нашлось(01) ИЛИ нашлось(02) ИЛИ нашлось(03)

    заменить(01, 30)

    заменить(02, 3103)

    заменить(03, 1201)

КОНЕЦ ПОКА

КОНЕЦ

Известно, что исходная строка начиналась с нуля, а далее содержала только единицы, двойки и тройки. После выполнения данной программы получилась строка, содержащая 31 единицу, 24 двойки и 46 троек. Сколько троек было в исходной строке?

- **Задача 3.** Напишите программу, подсчитывающую количество слов во введенной фразе, если:
  - а) между словами стоит один пробел;
  - б) между словами может стоять произвольное число пробелов;
  - в) между словами стоит один произвольный разделитель (пробел, «,», «.», «!», «?», «;», «-»);
  - г) между словами стоит произвольное число разделителей.



# Домашнее задание

## Количество слов

Дана строка, состоящая из слов, разделенных ровно одним пробелом. Определите, сколько в ней слов. Используйте для решения задачи метод `count`.