## ****Строка в Python****

Строка считывается со стандартного ввода функцией input(). Напомним, что для двух строк определена операция сложения (конкатенации), также определена операция умножения строки на число.

**Сложение строк:**

**S1 = 'Чат'**

**S2 = 'Бот'**

**print(S1 + S2)**

Результат: ЧатБот

**Умножение строк:**

**S1 = 'Чат'**

**print(S1\*5)**

**Результат:** ЧатЧатЧатЧатЧат

или

**S = input()**

**print(S\*5)**

**Вводим** Привет!

**Результат:** Привет!Привет!Привет!Привет!Привет!

Строка - это набор символов. При этом каждый символ имеет свой порядковый номер. Нумерация строк начинается с 0.



Например, если мы возьмём переменную и сохраним в ней эту строчку, то сможем её обрабатывать:

a = 'Привет это строка'

 Теперь, если я попрошу вывести 5-й символ, то напишу: print(a[5])  в качестве результата я получу: т .

 Но стоит учитывать, что строку нельзя менять напрямую. Т.е. команда вида a[5] = 'я' выдаст ошибку:TypeError: 'str' object does not support item assignment

 Для изменения строк существуют специальные функции, но о них чуть позже.

 *Номер элемента в строке принято называть индексом.*

Индекс строки может быть отрицательным. Тогда мы будем рассматривать строку в обратном порядке. В этой ситуации -1 это последний символ строки, -2 - предпоследний и т.д.

## ****Срез****

Срез позволяет вернуть часть строки.

Срез выглядит следующим образом: a[start:stop:step]

Где a - это наша строка, start - начало среза, stop - конец среза, step - шаг среза.

Каждый из параметров можно указывать, а можно игнорировать. Они могут принимать целочисленные значения как положительные, так и отрицательные.

**Рассмотрим несколько примеров:**

 a = 'Привет это строка' print(a[5:8])

 В данном случае мы получим срез символов от 5 до 8 (не включительно). То есть программа выведет 5, 6 и 7 символы.

 т э

 **Ещё пример:**

 a = 'Привет это строка' print(a[5:])

 В данном примере мы указали начало среза, но не указали конец. В этой ситуации будет взят срез от 5 го символа и до конца строки. И результатом будет:  т это строка

 **Пример:**

 a = 'Привет это строка' print(a[:8])

 В этом примере мы не указали начальное значение. Тогда мы получим срез с начала строки и до 8-го символа (не включая его).

  Привет э

 **Пример с шагом:**

 a = 'Привет это строка' print(a[1:10:2])

 В этом примере мы идём с 1-го (не нулевого) символа и берём каждый второй символ (шаг равен 2). Тогда мы получим следующий срез.

 рвтэо

***ВАЖНО:****чтобы развернуть строку, достаточно прописать: a[::-1]. То есть мы делаем срез от начала и до конца с шагом -1, движемся в обратном направлении.*

*И если у вас задача на палиндром, то всё что вам нужно проверить, это узнать равны ли 2 строки: a == a[::-1]*

## ****Функции и методы строк****

Мы уже рассмотрели некоторые возможности строк, в частности объединение, умножение и срезы.

Перейдём к другим функциям:

**len()** - определяет длину строки

**Пример:**

print(len('Как дела'))

**Результат:** 8

 **s.find(str, [start],[end])**  - поиск подстроки str в строке s. В качестве ответа получим номер первого вхождения строки str в строку s или -1, если этой строки нет.

**Пример:**

 print('Как дела'.find('ла'))

print('Как дела'.find('ого'))

 **Результат:**

 6

-1

 **s.index(str, [start],[end])**- поиск подстроки str в строке s. В качестве ответа получим номер первого вхождения строки str в строку s или ValueError , если этой строки нет. В данном случае, если подстроки нет, то выкинет с ошибкой, если её правильно не обработать.

**Пример:**

 print('Как дела'.index('ла'))

print('Как дела'.index('ого'))

 **Результат:**

6

ValueError

 **s.split(символ)** - разбивает строку по символу разделителю.

**Пример:**

print('Как дела'.split('а'))

**Результат:**

['К', 'к дел', '']

В результате мы разделили на 3 строки по символу а

Представьте, что вы взяли ножницы. У вас на листочке написан текст Как дела И вы разрезаете листочек по тем местам, где у вас буква а. В результате вы делаете три разреза и получаете три листочка. На первом написано К, на втором к дел и третий пустой, т.к. ножницы прошлись как раз по последней букве а.

 **s.count(str, [start],[end])**- считает количество вхождений строки str в строку s. При этом учитываются только не пересекающиеся вхождения.

**Пример:**

print('863483468'.count('8'))

print('868686868686868'.count('868'))

**Результат:**

3 - подсчитает количество 8.

4 - подсчитает количество 868, при этом они не должны пересекаться. 86868 - это всего 1 комбинация 868.

 **s.isdigit()** - проверяет состоит ли строка только из цифр

**Пример:**

print('863483468'.isdigit())

print('863483 468'.isdigit())

**Результат:**

True - в предложении только цифры.

False - в предложении есть пробел.

 **s.isalpha()** - проверяет состоит ли строка только из букв

**Пример:**

print('afasfsafasfas'.isalpha())

print('afasfsaf asfas?'.isalpha())

**Результат:**

True - в предложении только буквы.

False - в предложении есть пробел и знак вопроса.

 **s.lower()** - переводит строку в нижний регистр

**Пример:**

print('Привет'.lower())

print('ПрИВЕТ!'.lower())

**Результат:**

привет

привет!

 **s.upper()** - переводит строку в верхний регистр.

 **Пример:**

print('Привет'.upper())

print('ПривЕТ!'.upper())

**Результат:**

ПРИВЕТ

ПРИВЕТ!

 **ord**(символ) - переводит символ в его код UTF-8.

**chr**(число) - переводит код UTF-8 в символ.

**Задача 11\_1.**

На вход подаётся строка и число. Необходимо вывести строку ровно столько раз, чему равно это число.

**Sample Input:**

Ура!

7

**Sample Output:**

Ура!Ура!Ура!Ура!Ура!Ура!Ура!

**Задача 11\_2.**

Напишите программу . На вход программе подаётся слово и строка. Нужно проверить, есть ли это слово в строке. Если такое слово есть, то посчитать сколько раз оно встречается. Если такого слова нет, то вывести **0**.

**Sample Input 1:**

ка

Привет, как у тебя дела?

**Sample Output 1:**

1

**Sample Input 2:**

нуну

Привет, как дела?

**Sample Output 2:**

0

**Sample Input 3:**

а

АаАааааАааАа

**Sample Output 3:**

8

**Задача 11\_3.**

Напишите программу, которая выводит последнюю цифру введённого числа

**Sample Input 1:**

76354671253671

**Sample Output 1:**

1

**Sample Input 2:**

433433

**Sample Output 2:**

3

**Задача 11\_4.**

Напишите программу, которая проверяет, является ли введённая строка палиндромом. Регистр не должен учитываться.

На выходе программа пишет "**ДА"** или "**НЕТ**"

**Sample Input 1:**

Совос

**Sample Output 1:**

ДА

**Sample Input 2:**

комОК

**Sample Output 2:**

ДА

**Задача 11\_5.**

С клавиатуры вводится текст и буква. Нужно посчитать, сколько раз встречается эта буква в тексте. Регистр не должен учитываться.

**Sample Input 1:**

Привет, как твои дела?

а

**Sample Output 1:**

2

**Sample Input 2:**

В лесу стояла ель, возле которой всегда бегали звери.

В

**Sample Output 2:**

4

**Списки**

Списки в Python - упорядоченные изменяемые коллекции объектов произвольных типов (почти как массив, но типы могут отличаться). [[pythonworld](https://pythonworld.ru/%22%20%5Ct%20%22_blank)]

Другими словами список - это набор значений, хранящихся в определённом порядке.

Примеры списка:

[12, 43, 5 , 23, 11 , 3213]

[12, 'hello', [32, 76, 15], 'name', 42]

## ****Создание списка****

Создать список можно несколькими способами.

1. Через создание пустого списка:

a = []

2. Через определение списка:

a = list()

3. Через создание заполненного списка:

a = [3, 4, 7, 2, 9]

4. Через умножение шаблона:

a = [0] \* 5

# [0, 0, 0, 0, 0]

a = [1, 3] \* 4

# [1, 3, 1, 3, 1, 3, 1, 3]

Есть и другие способы создания списков, например "генератор списка". Но они довольно сложные в плане понимания и выходят за рамки необходимых знаний.

## ****Индексация****

Каждый элемент внутри списка имеет свой индекс (порядковый номер). При этом индексация (нумерация) начиняется с нуля.

Мы можем обратиться к любому элементу списка, просто указав для него переменную и его порядковый номер (индекс) в квадратных скобках.

Пусть у нас задан список:

a = [4, 6, 1, 14, 2, 9, 22]

Тогда, если мы напишем: print(a[5]), то мы обратимся к **5**ому элементу из списка **a**и получим число **9.**Не забываем, что счёт начинается с 0

Таким образом:

a[0] = 4

a[1] = 6

a[2] = 1

a[3] = 14

a[4] = 2

a[5] = 9

a[6] = 22

Если попытаться обратиться к элементу, которого нет в списке, например к 10му элементу, то получим ошибку выхода за пределы списка.

print(a[10])

#Error: index out of range

Обратите внимание на эту ошибку, она одна из самых распространённых при работе со списками.

**Задача 11\_6.**

Напишите программу, которая принимает символьную строку содержащую фамилию и имя. Нужно построить новую строку, в которой записан инициал (первая буква имени с точкой и через пробел – фамилия. Например, из строки ***Семёнов Андрей*** должна получиться строка ***А.Семёнов***.

**Задача 11\_7.**

Напишите программу, которая в символьной строке заменяет:

а) все строчные латинские буквы на прописные;

б) все прописные латинские буквы на строчные;

в) все строчные русские буквы на прописные;

г) все прописные русские буквы на строчные.

**Задача 11\_8.**

Напишите программу, которая получает на вход символьную строку, состоящую из слов, разделённых пробелами, и заменяет в каждом слове первую букву на прописную.