

Лабораторная работа № 5

Строки

Строки в Питоне — это упорядоченная последовательность символов.

Строка может заключаться как в кавычки, так и в апострофы:

```
S = 'Dr"s'  
S = "Dr's"
```

Строка считывается со стандартного ввода функцией `input()`:

```
a=input()  
print(a)
```

ОПЕРАЦИИ СО СТРОКАМИ

Для двух строк определена операция сложения (конкатенации), также определена операция умножения строки на число:

```
a="па"  
b="рад"  
print(a+b) # парад
```

```
a="кар"  
print (a*4) # каркаркаркар
```

Работа, как с массивами (индексация начинается с 0):

```
a="парад"  
print (a[2]) # р
```

Длина строки – функция `len()`:

```
a="парад"  
print (len(a)) # 5
```

Срезы

Оператор извлечения среза из строки: `[X:Y]`

X – это индекс начала среза, а Y – его окончания

```
tday = 'morning, afternoon, night'  
tday[0:7] # 'morning'
```

Рассмотрим примеры того, как используются в Python срезы:

```
s = 'spameggs'  
s[3:5] # 'me'  
s[2:-2] # 'ameg'  
s[-4:-2] # 'eg'
```

```
s[:6] # 'spameg'
s[1:] # 'pameggs'
s[:] # 'spameggs'
```

Можно задать шаг:

```
s = 'spameggs'
s[::-1] # 'sggemaps'
s[3:5:-1] # ""
s[2::2] # 'aeg'
```

Пример:

Извлеките из строки символы с индексами кратными трем.

Решение:

Задание можно выполнить, используя цикл (сложный вариант решения):

```
1 s = 'spameggs'
2 x=3
3 l=len(s)//3
4 for i in range(l):
5     print(s[x:x+1:3]) # m g
6     x+=3
```

Для решения можно использовать просто срез:

```
1 s = 'spameggs'
2 print(s[1::3])
```

Задание Python 5_1: Извлеките из строки следующие срезы:

- первые восемь символов;
- четыре символа из центра строки;
- пять символов с конца строки.

Задание Python 5_2:

Дана строка длиной N. Вывести символы строки в обратном порядке. (Не использовать цикл).

Задание Python 5_3:

Дана строка длиной N (N — четное число). Вывести символы с четными номерами в порядке возрастания их номеров:

```
a2, a4, a6, ... an
```

Условный оператор не использовать.

Задание Python 5_4:

Дана строка длиной N. Вывести сначала символы с четными номерами (в порядке возрастания номеров), а затем — символы с нечетными номерами (также в порядке возрастания номеров):

a2, a4, a6,...a1, a3, a5...

Условный оператор не использовать.

Методы строк

Строки, как объекты Python, обладают методами (т.е. функциями, которые выполняют сами объекты).

`join(str)` — Соединение строк из последовательности `str` через разделитель, заданный строкой

```
s="hello"  
s1="-".join(s)  
s1 # 'h-e-l-l-o'
```

`s1.count(s[, i, j])` — количество вхождений подстроки `s` в строку `s1`. Результатом является число. Можно указать позицию начала поиска `i` и окончания поиска `j`:

```
s1="abrakadabra"; s1.count('ab') # 2  
s1.count('ab',1) # 1  
s1.count('ab',1,-3) # 0 , т.к. s1[1:-3]='brakada'
```

`s1.find(s[, i, j])` — определяется позиция первого (считая слева) вхождения подстроки `s` в строку `s1`. Результатом является число. `i` и `j` определяют начало и конец области поиска:

```
s1="abrakadabra"; s1.find('br') # 1
```

`s1.replace(s2,s3[, n])` — создаётся новая строка, в которой фрагмент (подстрока) `s2` исходной строки заменяется на фрагмент `s3`. Необязательный аргумент `n` указывает количество замен:

```
s1="breKeKeKeKs"; ss=s1.replace('Ke','XoXo',2)  
ss # breXoXoXoXoKeKs
```

Задание Python 5_5: Преобразовать дату в «компьютерном»

представлении (системную дату: 2018-03-26) в «русский» формат, т. е. день/месяц/год (например, 26/03/2018). Известно, что на год выделено всегда 4 цифры, а на день и месяц – всегда 2 цифры.

Примечание:

Использовать строковые функции языка.

Функциями работы с датами и временем «заведует» в Python datetime модуль, а непосредственно для работы с датами используется объект date и его методы.

Подсказка:

```
from datetime import date
# Получаем текущую дату
d1=date.today()
# Преобразуем результат в строку
ds=str(d1)
```

Задание Python 5_6:

Ввести адрес файла и «разобрать» его на части, разделенные знаком '/'. Каждую часть вывести в отдельной строке.

Например: c:/изображения/2019/1.jpg Результат:

```
c:
изображения
2019
1.jpg
```

Задание Python 5_7: Ввести строку, в которой записана сумма натуральных чисел, например, '1+25+3'. Вычислите это выражение. Использовать строковые функции языка.

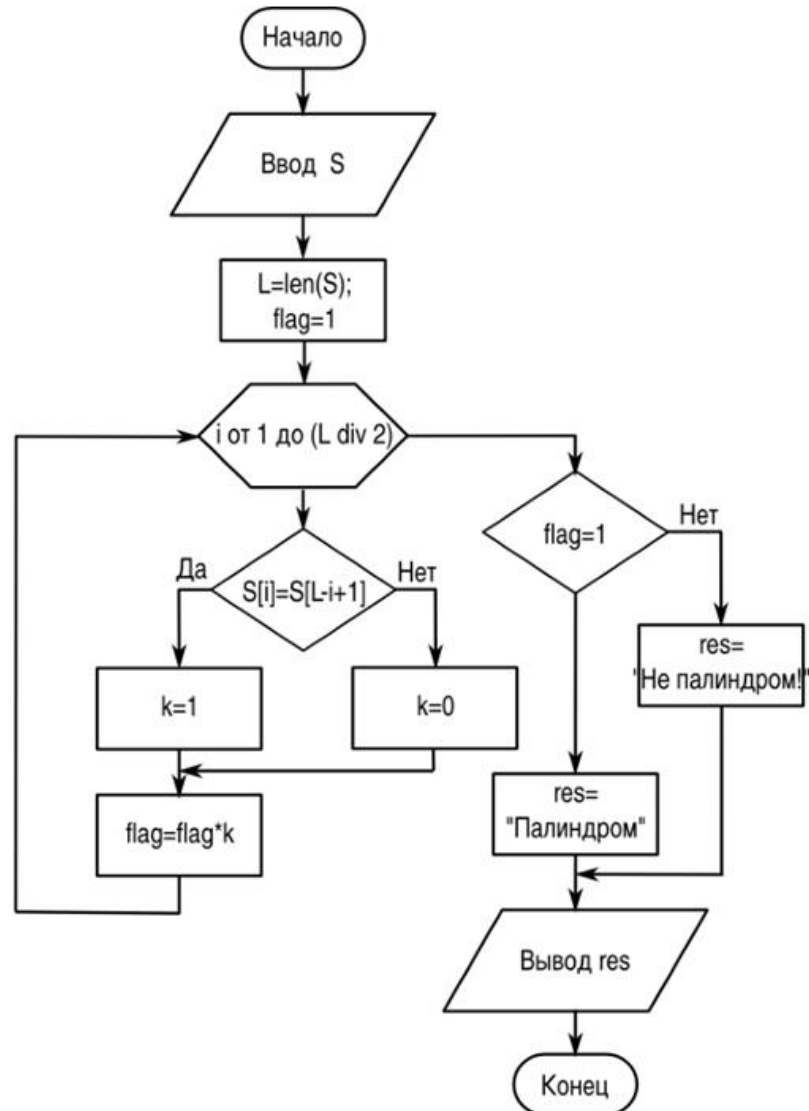
Задание Python 5_8: Определить, является ли введенная строка палиндромом («перевёртышем») типа АББА, казак и пр.

Примечание:

если s='1234', то

$S[-1]='4'$, $s[-2]='3'$

Для решения используйте алгоритм, изображенный на блок-схеме:



САМОСТОЯТЕЛЬНАЯ РАБОТА:

1 вариант:

Функции:

Описать функцию CircleS(R) вещественного типа, находящую площадь круга радиуса R. С помощью этой функции найти площади трех кругов с заданными радиусами. Площадь круга радиуса R вычисляется по формуле $S=\pi \cdot R^2$. В качестве значения Пи использовать 3.14.

Процедуры:

Напишите процедуру, которая выводит цифры переданного ей четырехзначного числа, кратные трем.

Списки:

Создайте список, соответствующий числовому ряду, построенному по следующему принципу:

1+2+3 2+3+4 3+4+5 ... 8+9+10

т.е. результат:

[6, 9, 12, 15, 18, 21, 24, 27]

2 вариант:

Функции:

Описать функцию TriangleP(a, h), находящую периметр равнобедренного треугольника по его основанию a и высоте h, проведенной к основанию. С помощью этой функции найти периметры трех треугольников, для которых даны основания и высоты (периметр = сумме длин всех сторон). Для нахождения боковой стороны b треугольника использовать теорему Пифагора: $b^2 = (a/2)^2 + h^2$.

Процедуры:

Напишите процедуру, которая выводит произведение цифр переданного ей четырехзначного числа.

Списки:

Создайте список, соответствующий числовому ряду, построенному по следующему принципу:

1*3 2*4 3*5 ... 8*10

т.е. результат:

[3, 8, 15, 24, 35, 48, 63, 80]

Общее:

Дана строка. Вывести символы с нечетными номерами в порядке убывания их номеров (0-й символ тоже рассмотреть).

Например, для строки:

"Друзья!", результат: "язр"

"Друзья", результат: "язр"

Форматирование строк

Python включает форматирование строк. Данное понятие подразумевает подстановку какого-либо шаблона в определенное место (или в определенные позиции) текста. Подстановка происходит, что называется, «на лету».

Рассмотрим пример использования в коде на Python метода `format`:

Одна подстановка:

```
'Hello, {}'.format('Vasya') # 'Hello, Vasya!'
```

Аргументом метода является текст-подстановка, который при исполнении программы подставляется на место фигурных скобок.

Несколько подстановок:

```
'{0}{1}{0}'.format('abra', 'cad') # 'abracadabra'
```

Подстановки нумеруются, аргументы метода `format` заполняют позиции для подстановок согласно их порядковым номерам, указанным в фигурных скобках.

другой вариант форматирования с множественными подстановками:

```
'Coordinates: {latitude}, {longitude}'.format(latitude='37.24N',  
longitude='-115.81W')  
  
'Coordinates: 37.24N, -115.81W'
```

Аргументы метода `format` заполняются согласно указанным именам заполнителей.

У метода `format` есть больше возможностей, которые выходят за рамки рассмотрения темы на данном уроке.

Задание Python 5_9:

Допустим, есть какое-то объявление и несколько разных людей, которым нужно это объявление отправить. Для этого создается заготовка с содержанием объявления, внутри которого есть ряд изменяющихся параметров: имена людей и названия событий. Вывести один вариант

итогового объявления на экран. Для задания имен и названий использовать форматирование при помощи метода `format`.

Используйте шаблон объявления и ориентируйтесь по цветам:

Красным – массивы.

Коричневым – числовая переменная.

Уважаемый (ая), **Иван Иванович!**

Приглашаем Вас на **день открытых дверей**.

Дата события: **1** мая. С уважением, Василий.

** Примечание:*

Вывести пять объявлений.

Для имен создать массив (список), подстановку элементов в объявление осуществлять в цикле.

Даты меняются от 1 до 5.