



Фоксфорд

Кружки

Кружок по программированию на Python

Занятие №7



Кружок по программированию на Python

Строковый тип данных

- Строки в Python
- Срезы строк в Python



Строковый тип данных

```
s1 = 'Привет'  
print(s1)  
s2 = 'Как дела? '  
print(s2)  
s3= """Абзац закончен.  
    Красная строка."""  
print(s3)
```

Апострофы ' ' или
двойные кавычки " "
обрамляют строки



Строковый тип данных

```
s1 = 'Привет'  
s2 = 'Как дела? '  
s3= """Абзац закончен.  
    Красная строка."""  
print(s1+s2+'\n'+s3)
```

Строки можно складывать. Эта операция называется конкатенацией строк – «склеиванием».



Строковый тип данных

```
s1 = 'Привет'  
print(('-' + s1 + '! \n')*2)
```

```
>>-Привет!
```

```
>>-Привет!
```

Также строки можно умножать на число.

Строковый тип данных

```
s1 = 'Привет'  
print(s1[0])  
>>П
```

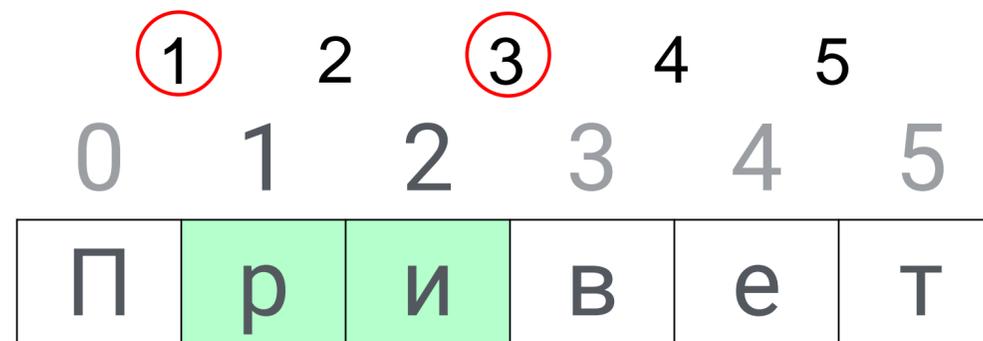
0	1	2	3	4	5
П	р	и	в	е	т

Апострофы ' ' или двойные кавычки " " обрамляют строки

Как было упомянуто ранее, строки – это итерируемые объекты.

Срезы

```
s1 = 'Привет'  
print(s1[1:3])  
>>ри
```



Апострофы ' ' или
двойные кавычки " "
обрамляют строки

Как было упомянуто ранее, строки – это итерируемые объекты.

Срезы

```
s1 = 'Привет'  
print(s1[-5:-3])  
>>ри
```

	1	2	3	4	5
0	1	2	3	4	5
П	р	и	в	е	т
-6	-5	-4	-3	-2	-1

Апострофы ' ' или
двойные кавычки " "
обрамляют строки

Как было упомянуто ранее, строки – это итерируемые объекты.

Срезы

```
s1 = 'Весна!'
print(s1[1::2])
>>ен!
```

	1	2	3	4	5
0	1	2	3	4	5
В	е	с	н	а	!
-6	-5	-4	-3	-2	-1

Апострофы `'` или
двойные кавычки `"`
обрамляют строки

Как было упомянуто ранее, строки – это итерируемые объекты.



Задача 1

- Дана строка. Разрежьте ее на две равные части (если длина строки — четная, а если длина строки нечетная, то длина первой части должна быть на один символ больше). Переставьте эти две части местами, результат запишите в новую строку и выведите на экран.
- При решении этой задачи нельзя пользоваться инструкцией `if`.



Задача 1

```
s = input()
l=len(s)
print((s[l//2+l%2:])+ (s[:l//2+l%2]))
```

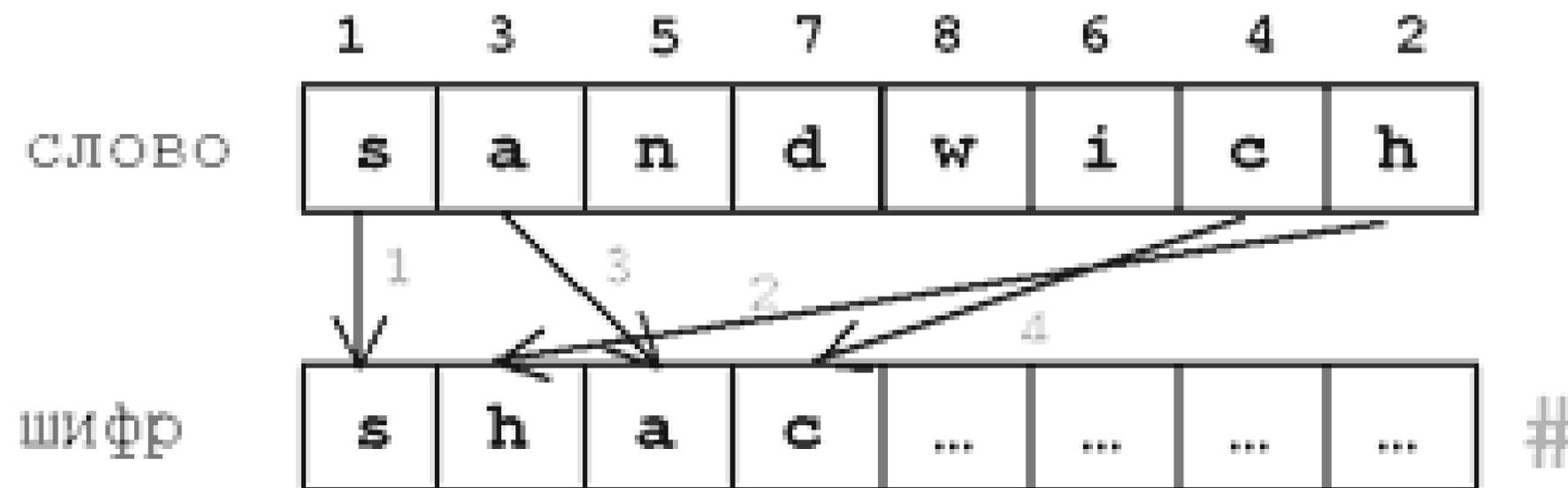


Задача 2

- Секретное агентство «Super-Secret-но» решило для шифрования переписки своих сотрудников использовать «метод бутерброда». Сначала буквы слова нумеруются: первая буква получает номер 1, последняя буква - номер 2, вторая – номер 3, предпоследняя – номер 4, потом третья ... и так для всех букв. Затем все буквы записываются в шифр в порядке своих номеров и в конец зашифрованного слова добавляется #, который нельзя использовать в сообщениях.

Задача 2

- Например, слово «sandwich» зашифруется в «shacnidw#».



На рисунке
показано
начало
шифрования
слова
«sandwich»

- К сожалению, программист «Super-Secret-но», написал только программу шифрования и уволился. И теперь агенты не могут понять, что же они написали друг другу. Помогите им.



Задача 2

```
a = input()
a = a[:len(a)-1]
b = a[0::2]
c = a[1::2]
c = c[::-1]
print(b+c)
```



Методы строк

<code>len(s)</code>	Длина строки <code>s</code>
<code>s.find(str)</code>	Поиск подстроки <code>str</code> в строке <code>s</code>
<code>s.index(str)</code>	Индекс первого вхождения подстроки <code>str</code>
<code>s.split(sep)</code>	Разделить строку <code>s</code> по разделителю <code>sep</code>
<code>s.replace(s1, s2)</code>	В строке <code>s</code> заменить <code>s1</code> на <code>s2</code>
<code>s.count(str)</code>	Количество непересекающихся вхождений <code>str</code> в <code>s</code>
<code>s.format()</code>	Форматирование строки
<code>s.isdigit()</code>	Проверка, что строка числовая
<code>s.isalpha()</code>	Проверка, что строка буквенная



Метод `format`

```
'Hello, {}!'.format('Harry')
```

```
>> Hello, Harry!
```

```
'Первые три планеты – {2}, {1}, {0}.'.format('Земля',  
'Венера', 'Меркурий')
```

```
>> Первые три планеты – Меркурий, Венера, Земля.
```

```
'Координаты соперника – {:.3f}, {:.3f}.'.format(-5/2,  
5/3)
```



Задача 3

- Дана строка, состоящая ровно из двух слов, разделенных пробелом. Переставьте эти слова местами. Результат запишите в строку и выведите получившуюся строку.
- При решении этой задачи нельзя пользоваться циклами и инструкцией `if`.



Задача 3

```
s = input()
a = (s.find(' '))
print((s[a+1:])+ (s[a:a+1])+ (s[:a]))
```

Или:

```
a, b = input().split()
print(b, a)
```



Задача 4

- Решите уравнение $ax \pm b = 0$. Вводится строка в формате
- $ax + b = 0$ или $ax - b = 0$, где вместо a — целое число, вместо b — целое неотрицательное число. Числа по модулю не превосходят 1000. Число a может быть пропущено, если оно равно 1. b всегда присутствует, x также всегда присутствует, даже если $a = 0$.
- Выведите корни уравнения, если их конечное число; 'NO', если корней нет и 'INF', если корней бесконечно много.



Задача 4

```
s = input ()[:-2]
indexx = s.find('x')
if (indexx == 0):
    a = 1
else:
    a = int(s[:indexx])
s=s[indexx+1:]
b = int(s)
```

```
if a == 0:
    if b == 0:
        print('INF')
    else:print ('NO')
else:
    print(-b / a)
```



Задача 5

- Напишите программу, которая считывает целое число в диапазоне от -1000 до 1000 и выводит текст:
- `>>179`
- The next number for the number 179 is 180.
- The previous number for the number 179 is 178.



Задача 5

```
number = int(input())  
print('The next number for the number ' + str(number) + ' is ' +  
str(number+1) + '.')  
print('The previous number for the number ' + str(number) + ' is ' +  
str(number-1) + '.')
```



Фоксфорд
Кружки

Спасибо за внимание!