9.2 Срезы

Шаг 1

Тема урока: строки

1. Срезы строк
2. Изменение символов строки
3. Решение задач

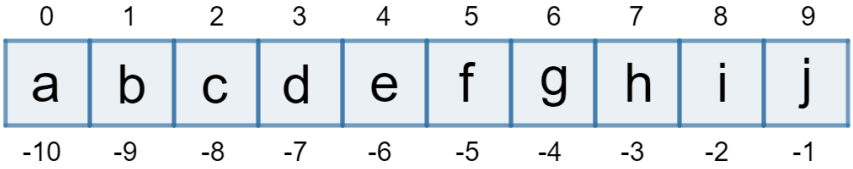
Аннотация. Снова строковый тип данных. Учимся делать строковые срезы, а также изменять символы в строке.

Срезы строк

В предыдущем уроке мы научились работать с конкретными символами строки с помощью индексов `[]`. Иногда нужно бывает работать с целыми частями строки, в таком случае мы используем **срезы (slices)**. Срезы похожи на комбинацию индексации и функции `range()`.

Рассмотрим строку `s = 'abcdefghij'`.

Положительные индексы	0	1	2	3	4	5	6	7	8	9
Строка	a	b	c	d	e	f	g	h	i	j
Отрицательные индексы	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1



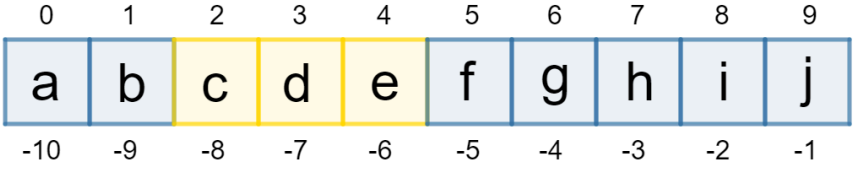
С помощью среза мы можем получить несколько символов исходной строки, создав диапазон индексов разделенных двоеточием `s[x:y]`.

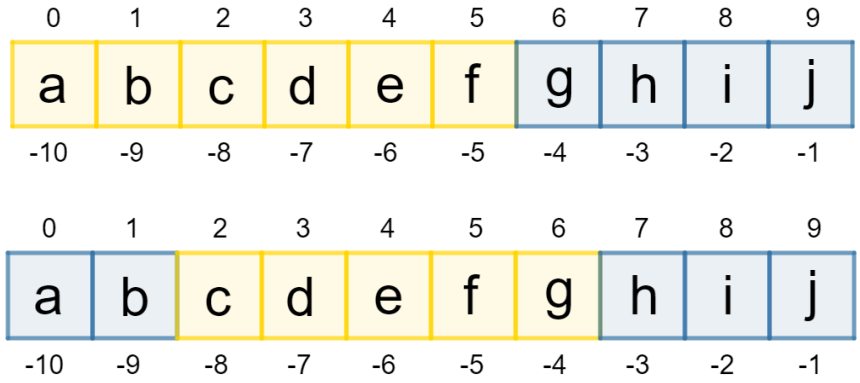
Приведённый ниже код:

```
print(s[2:5])
print(s[0:6])
print(s[2:7])
```

выводит:

```
cde
abcdef
cdefg
```





При построении среза `s[x:y]` первое число – это то место, где начинается срез (**включительно**), а второе – это место, где заканчивается срез (**невключительно**). Разрезая строки, мы создаем подстроку, которая по сути является строкой внутри другой строки.

Срез до конца, от начала

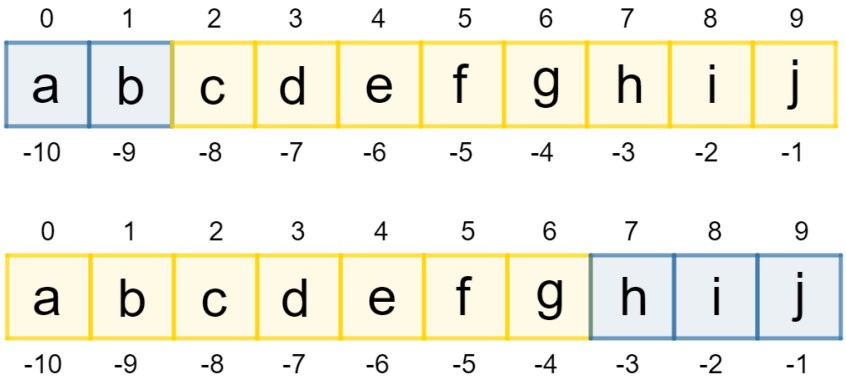
Если опустить второй параметр в срезе `s[x:]` (но поставить двоеточие), то срез берется до конца строки. Аналогично если опустить первый параметр `s[:y]`, то можно взять срез от начала строки. Срез `s[:]` совпадает с самой строкой `s`.

Приведённый ниже код:

```
print(s[2:])
print(s[:7])
```

выводит:

```
cdefghij
abcdefg
```



Срез `s[:]` возвращает исходную строку.

Отрицательные индексы в срезе

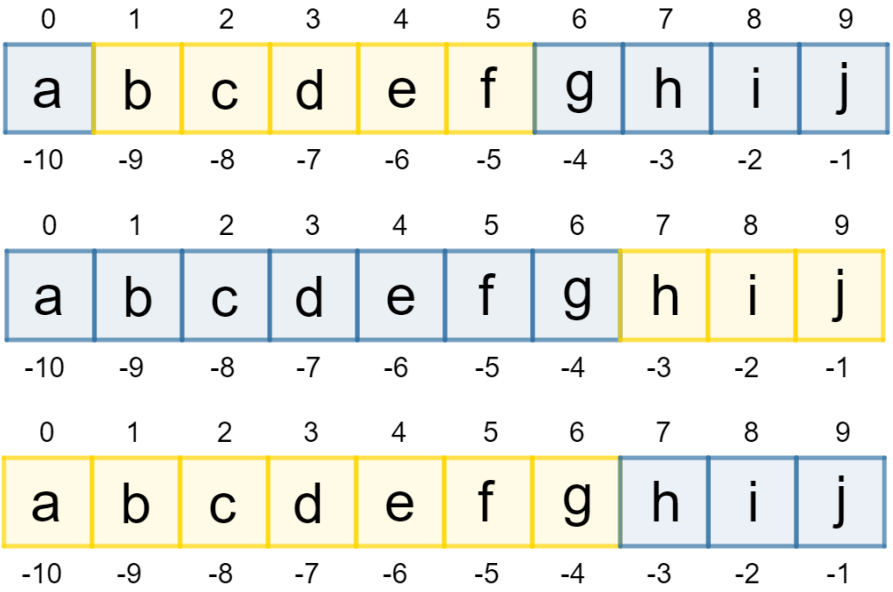
Мы также можем использовать отрицательные индексы для создания срезов. Как уже говорилось ранее, отрицательные индексы строки начинаются с `-1` и отсчитываются до достижения начала строки. При использовании отрицательных индексов ***первый параметр среза должен быть меньше второго, либо должен быть пропущен***.

Приведённый ниже код:

```
print(s[-9:-4])
print(s[-3:])
print(s[:-3])
```

выводит:

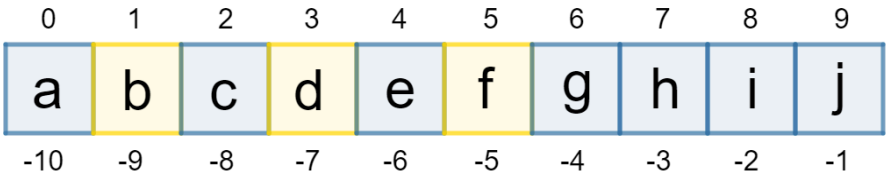
```
bcdef
hij
abcdefg
```



Удалить из строки последний символ можно при помощи среза `s[:-1]`.

Шаг среза

Мы можем передать в срез третий необязательный параметр, который отвечает за шаг среза. К примеру, срез `s[1:7:2]` создаст строку `bdf` состоящую из каждого второго символа (индексы `1, 3, 5`, правая граница не включена в срез).



Отрицательный шаг среза

Если в качестве шага среза указать **отрицательное число**, то символы будут идти в обратном порядке.

Приведённый ниже код:

```
print(s[::-1])
```

выводит:

```
jihgfedcba
```

Приведённый ниже код:

```
print(s[1:7:2])
print(s[3::2])
print(s[:7:3])
print(s[::2])
print(s[::-1])
print(s[::-2])
```

выводит:

```
bdf
dfhj
adg
acegi
jihgfedcba
jhfdb
```

Подводя итог

`s = 'abcdefghij'`

Программный код	Результат	Пояснение
`s[2:5]`	`cde`	строка состоящая из символов с индексами 2, 3, 4
`s[:5]`	`abcde`	первые пять символов строки
`s[5:]`	`fghij`	строка состоящая из символов с индексами от 5 до конца
`s[-2:]`	`ij`	последние два символа строки
`s[:]`	`abcdefghij`	вся строка целиком
`s[1:7:2]`	`bdf`	строка, состоящая из каждого второго символа с индексами от 1 до 6
`s[::-1]`	`jihgfedcba`	строка в обратном порядке, так как шаг отрицательный

Изменение символа строки по индексу

Предположим, у нас есть строка `s = 'abcdefghij'` и мы хотим заменить символ с индексом 4 на `'X'`. Можно попытаться написать код:

```
s[4] = 'X'
```

Однако такой код не работает. В Python строки являются **неизменяемыми**, то есть мы не можем менять их содержимое с помощью индексатора.

Если мы хотим поменять какой-либо символ строки `s`, мы должны создать новую строку. Следующий код использует срезы и решает поставленную задачу:

```
s = s[:4] + 'X' + s[5:]
```

Мы создаем два среза: от начала строки до 3 символа и с 5 символа по конец строки, а между ними вставляем нужный нам символ, который встанет на 4 позицию.

Примечания

Примечание 1. Синтаксис срезов строк очень похож на синтаксис функции `range()`.

Примечание 2. Если первый параметр среза больше второго, то срез создает пустую строку.

❤️ Happy Pythoning!

Шаг 2

Что покажет приведённый ниже код?

```
s = 'abcdefg'
print(s[2:5])
```

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/2

Шаг 3

Что покажет приведённый ниже код?

```
s = 'abcdefg'
print(s[3:])
```

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/3

Шаг 4

Что покажет приведённый ниже код?

```
s = 'abcdefg'
print(s[:3])
```

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/4

Шаг 5

Что покажет приведённый ниже код?

```
s = 'abcdefg'
print(s[:])
```

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/5

Шаг 6

Что покажет приведённый ниже код?

```
s = 'abcdefg'
print(s[::-3])
```

```

```

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/6

Шаг 7

Используя срезы, дополните приведённый ниже код так, чтобы он вывел первые 12 символов строки `s`.

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/7

Шаг 8

Используя срезы, дополните приведённый ниже код так, чтобы он вывел последние 9 символов строки `s`.

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/8

Шаг 9

Используя срезы, дополните приведённый ниже код так, чтобы он вывел каждый 7 символ строки `s` (начиная с 0-го индекса).

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/9

Шаг 10

Используя срезы, дополните приведённый ниже код так, чтобы он вывел строку `s` в обратном порядке.

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/10

Шаг 11

Палиндром

На вход программе подаётся одно слово, записанное в нижнем регистре. Напишите программу, которая определяет, является ли оно палиндромом.

Формат входных данных
На вход программе подаётся одно слово в нижнем регистре.

Формат выходных данных
Программа должна вывести «YES» (без кавычек), если слово является палиндромом, или «NO» (без кавычек) в противном случае.

Примечание. Палиндром считается слово, которое читается одинаково в обоих направлениях. Например, слово «потоп» является палиндромом.

▶ Тестовые данные

Sample Input 1:

```
потоп
```

Sample Output 1:

```
YES
```

Sample Input 2:

```
анекдот
```

Sample Output 2:

```
NO
```

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/11

Шаг 12

Делаем срезы 1

На вход программе подаётся одна строка. Напишите программу, которая выводит:

1. общее количество символов в строке;
2. исходную строку, повторённую 3 раза;
3. первый символ строки;
4. первые три символа строки;
5. последние три символа строки;
6. строку в обратном порядке;
7. строку с удалённым первым и последним символами.

Формат входных данных

На вход программе подаётся одна строка, длина которой больше 3 символов.

Формат выходных данных

Программа должна вывести данные в соответствии с условием. Каждое значение выводится на отдельной строке.

▶ Тестовые данные

Sample Input 1:

```
abcdefghijklmnopqrstuvwxyz
```

Sample Output 1:

```
26
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
a
abc
xyz
zyxwvutsrqponmlkjihgfedcba
bcdefghijklmnopqrstuvwxy
```

Sample Input 2:

```
Success is the ability to go from failure to failure without losing your enthusiasm
```

Sample Output 2:

```
83
Success is the ability to go from failure to failure without losing your enthusiasmSuccess is the
ability to go from failure to failure without losing your enthusiasmSuccess is the ability to go
from failure to failure without losing your enthusiasm
S
Suc
asm
msaisuhtne ruoy gnisol tuohtiw eruliaf ot eruliaf morf og ot ytiliba eht si sseccuS
uccess is the ability to go from failure to failure without losing your enthusias
```

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/12

Шаг 13

Делаем срезы 2

На вход программе подаётся одна строка. Напишите программу, которая выводит:

1. третий символ этой строки;
2. предпоследний символ этой строки;
3. первые пять символов этой строки;
4. всю строку, кроме последних двух символов;
5. все символы с чётными индексами;
6. все символы с нечётными индексами;
7. все символы в обратном порядке;
8. все символы строки через один в обратном порядке, начиная с последнего.

Формат входных данных

На вход программе подаётся одна строка, длина которой больше 5 символов.

Формат выходных данных

Программа должна вывести данные в соответствии с условием. Каждое значение выводится на отдельной строке.

▶ Тестовые данные

Sample Input:

```
abcdefghijklmnopqrstuvwxyz
```

Sample Output:

```
c
y
abcde
abcdefghijklmnopqrstuvwx
acegikmoqsuwy
bdfhjlnprtvxz
zyxwvutsrqponmlkjihgfedcba
zxvtrpnljhfdb
```

Чтобы решить это задание откройте https://stepik.org/lesson/302627/step/13

Две половинки

На вход программе подаётся строка текста. Напишите программу, которая разрежет её на две равные части, переставит их местами и выведет на экран.

Формат входных данных

На вход программе подаётся строка текста.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Примечание. Если длина строки нечётная, то длина первой части должна быть на один символ больше.

▶ Тестовые данные

Sample Input 1:

```
abcdef
```

Sample Output 1:

```
defabc
```

Sample Input 2:

```
abcdefg
```

Sample Output 2:

```
efgabcd
```

Sample Input 3:

```
a
```

Sample Output 3:

```
a
```