



Строки

Урок №9

Сегодня

Откроем тайну типа
данных “строка”,
чтобы стать на шаг
ближе к ...



Сегодня

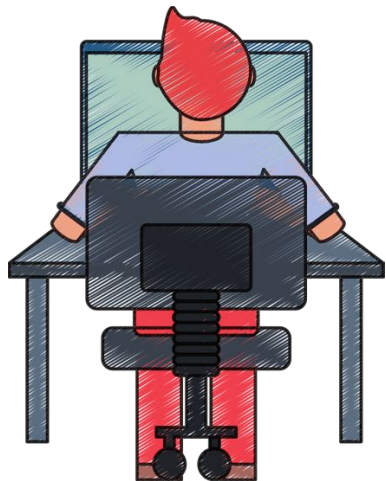
Откроем тайну типа данных “строка”, чтобы стать на шаг ближе к ...



Александру, который работает в Google и занимается обучением поисковой системы. Учит её распознавать смысл текстов.

Сегодня

Откроем тайну типа данных “строка”, чтобы стать на шаг ближе к ...



Евгению, который пишет ботов для Discord, Telegram и социальных сетей. Его боты общаются с пользователями, почти как настоящие люди.

Сегодня

Откроем тайну типа данных “строка”, чтобы стать на шаг ближе к ...



Дмитрию, который работает в подразделении Алисы в Яндексе и умеет обрабатывать самые странные и запутанные человеческие запросы.

Список дел

- ☐ Как считывать строки?
- ☐ Как строки хранятся в компьютере?
- ☐ Как их менять?
- ☐ Как их **срезать**?

Задача с прошлого года

1. `a = input()`

2. `b = input()`

3. `print(a+b)`

что этот код выводит на экран?



Тип данных “строка”

Строка - это тип данных предназначенный для хранения последовательности символов.

string - название типа данных на английском.

str(что-то) - это функция, превращающая что-то в строку.

`s = input()` - считывание строки



Действия со строкой



Создать строку:

```
s = "banana"  
s = 'банан'
```

Сложить (склеить) строки:

```
s = "bana" + "na"  
print(text1 + text2)  
text1, text2 - переменные
```

Умножить строку:

```
s = "ха" * 10
```



Действия со строкой



Узнать длину строки:

length — длина с англ.

len(строка) — функция, которая возвращает длину строки.

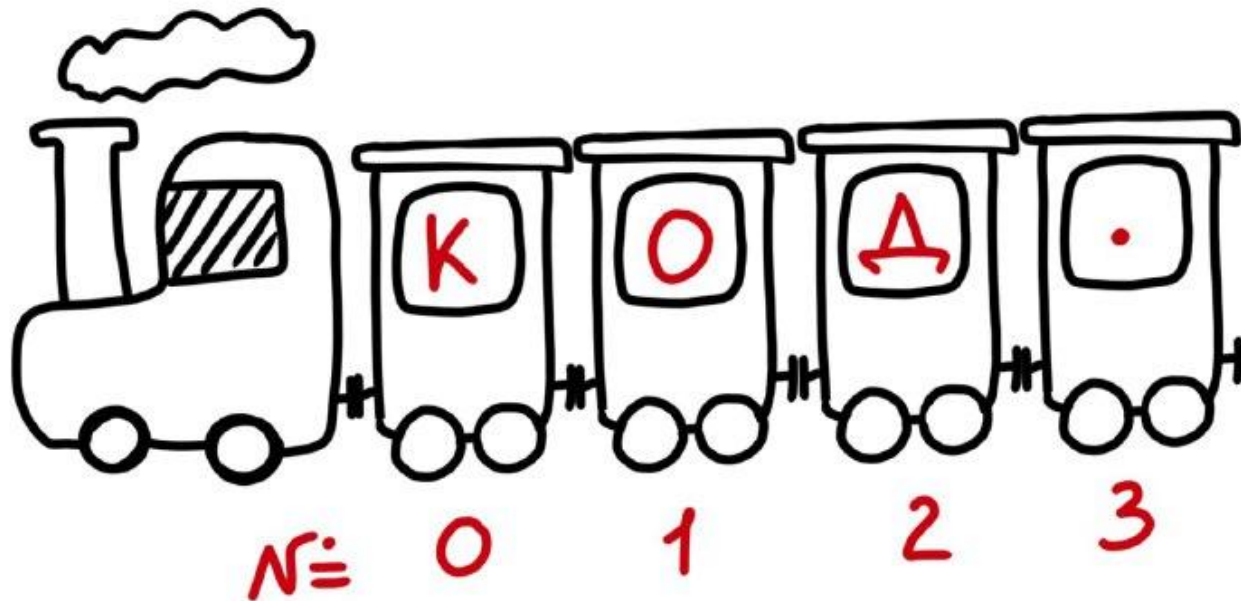
Пример:

```
text      = "люблю    домашку"  
x          = len(text)  
print(x)
```

Результат:

13

Запись строки в компьютере





Запись строки в компьютере





Действия со строкой



Обратиться к символу строки:

```
text = "Экспеллиармус"
```

```
print(text[2])
```

Результат:

с

Срезы строк

```
text = "milkshake"
```

```
text[2 : 5]
```

```
text[2 : ]
```

```
text[ : 5]
```

```
text[-2]
```

```
text[ : ]
```

```
text[ :: 2]
```



Срезы строк



Как работает?	Пример кода
[от : до] до не включен	<code>text[2:5]</code>
[от :] от и до конца	<code>text[2:]</code>
[: до] от начала и до (не включен)	<code>text[:5]</code>
[- индекс] можно ставить минус и считать с конца	<code>text[-2]</code>

Индексация



Строка S	З	А	Х	А	Р
Индекс	S[0]	S[1]	S[2]	S[3]	S[4]
Индекс	S[-5]	S[-4]	S[-3]	S[-2]	S[-1]

Срезы строк



Как работает?	Пример кода
[от : до : x] x - это шаг, с которым берутся символы по умолчанию = 1	<code>text[1:5:2]</code> берем каждый 2 символ с 1 по 5 (не включая)

Если вместо **от** или **до** оставить пустоту, то по умолчанию это будет “от начала” и “до конца”.

P.S. Попробуйте, что будет, если шаг отрицательный

Вечная строка



У строк в Питоне есть особенность:
они неизменяемы.

Нельзя изменить символы в строке, но можно переопределить строку целиком.

`s = "МОЛОКО"`

Нельзя сделать **`s[3] = "А"`**

Можно сделать **`s = "МОЛАКО"`**

Вечная строка

У строк в Питоне есть особенность:

они неизменяемы.

Нельзя изменить символы в строке, но можно переопределить строку целиком.

`s = "МОЛОКО"`

Можно сделать **`s = s + "батон"`**

Функция определения типа переменной

```
text = "abracadabra"  
print( type(text) )
```

```
n = 165  
print( type(n) )
```

Соревнование

Дано слово, например, word = “программист”. За 7 минут нужно собрать из букв этого слова как можно больше других слов русского языка.

Придуманные слова нужно вывести на экран и пронумеровать.

НО нельзя написать само слово.

Нужно собрать его по кусочкам из исходной строки, используя срезы.

Можно использовать сложение строк.

Пример

```
word = "программист"
```

```
print(1, word[1:4]) #рог
```

```
print(2, word[-4:-2] + word[1]) #мир
```



f-строки

Начиная с Python 3.6, f-строки — отличный новый способ форматирования строк.

Пример:

```
name = "Павел Воляев"  
mark1 = 4 #оценка по математике  
mark2 = 5 #оценка по физ-ре  
print(f"Ученик: {name}. Средняя оценка {(mark1+mark2)/2}")
```

Результат: Ученик: Гарик. Средняя оценка 3.5