Escape-последовательности

`\n` – новая строка
`\r` – возврат каретки
`\t` – табуляция
`\\` – обратный слэш

Вместе с пробелом (`' '`) они составляют пробельные символы.

```
print("a\nb\tc")
```

```
a
b	c
```

Методы для работы со строками

Так как строки неизменяемы, методы возвращают новые строки, а старые остаются без изменений.

Метод `.split()` позволяет разбить строку на подстроки по какому-нибудь символу-разделителю. Он возвращает список строк. По умолчанию разделитель – любая последовательность пробельных символов.

```
a = "1, 2, 3\t4"
a = a.split()
print(a)
```

```
['1,', '2,', '3', '4']
```

```
a = "1, 2, 3\t   4"
a = a.split()
print(a)
```

```
['1,', '2,', '3', '4']
```

```
a = "1, 2, 3"
a = a.split(", ")
print(a)
```

```
['1', '2', '3']
```

Параметр `maxsplit` позволяет указать максимальное количество разбиений.

```
a = "1, 2, 3"
a = a.split(", ", maxsplit=1)
print(a)
```

```
['1', '2, 3']
```

Если мы хотим отсчитывать разбиения справа, используем метод `.rsplit()`.

```
a = "1, 2, 3"
a = a.rsplit(", ", maxsplit=1)
print(a)
```

```
['1, 2', '3']
```

Метод `.strip()` позволяет обрезать пробельные символы по краям строки. `.lstrip()` делает это только слева, `.rstrip()` – только справа.

```
a = "   a\t"
print([a.strip(), a.rstrip(), a.lstrip()])
```

```
['a', '   a', 'a\t']
```

Метод `.join()` позволяет объединить список строк с некоторой строкой как разделителем.

```
a = ["1", "2", "3"]
print("-".join(a))
```

```
1-2-3
```

Если разделитель не нужен, укажем в качестве него пустую строку.

```
a = ["1", "2", "3"]
print(" ".join(a))
```

```
1 2 3
```

Метод `.replace()` позволяет заменить заданные подстроки в строке на что-то другое.

```
a = "1_2_3_4_5"
print(a.replace("_", "+"))
```

```
1+2+3+4+5
```

```
a = "1_2_3_4_5"
print(a.replace("_", ""))
```

```
12345
```

Метод `.startswith()` позволяет определить, начинается ли строка с заданной подстроки (`True`) или нет (`False`). `.endswith()` – то же самое для конца строки.

```
a = "this is a test"
print(a.endswith("est"))
```

```
True
```

Методы `.upper()`, `.lower()` и `.capitalize()` управляют регистром строки.

```
a = "tHiS iS a TeSt"
print(a.upper())
print(a.lower())
print(a.capitalize())
```

```
THIS IS A TEST
this is a test
This is a test
```

С полным списком строковых методов можно ознакомиться в документации.

Задание: дан текст. Уберите из него запятые и точки, приведите к нижнему регистру и создайте список всех слов, которые начинаются на букву "т". Выведите на экран все эти слова через нижнее подчёркнивание.

```
text = (
    "Был тихий серый вечер. Дул ветер, слабый и тёплый. "
    "Небо было покрыто тучами, сквозь которые едва пробивались лучи "
    "заходящего солнца."
)
# решение
text = text.lower()
text = text.replace(".", "")
text = text.replace(",", "")
words = text.split()
t_words = [w for w in words if w.startswith("т")]
print(t_words)
```

```
['тихий', 'тёплый', 'тучами']
```

Особые типы строк

"Сырые" строки (r-строки) не учитывают escape-последовательности.

```
a = "a\tb"
b = r"a\tb"
print(a)
print(b)
```

```
a	b
a\tb
```

Форматные строки (f-строки) позволяют вставлять в строку значения переменных.

```
a = 4
b = "1"
c = f"some value: {a},  another value: {b}"
print(c)
```

```
some value: 4,  another value: 1
```

С названиями переменных:

```
c = f"some value: {a=},  another value: {b=}"
print(c)
```

```
some value: a=4,  another value: b='1'
```

Если хотим "добить" нулями до определённой длины:

```
x = 11
print(f"{x:04d}")
```

```
0011
```

Если нужна определённая точность:

```
y = 11.123123123123
print(f"{y:.4f}")
```

```
11.1231
```

Полную спецификацию можно посмотреть в документации.

Многострочные строки:

```
text = """Line 1
Line 2
Line 3
Line 4"""
print(text)
```

```
Line 1
Line 2
Line 3
Line 4
```

В таких строках часто оформляют документацию (т.н. doc-строки). Обратите внимание, что это не комментарии!

Домашнее задание:

1. Возьмите какой-нибудь достаточно длинный текст на русском языке в формате .txt.
2. Для каждой буквы найдите самое длинное и самое короткое слово, которое начинается на эту букву.
3. Выведите полученную информацию на экран.