

**MLClass. "Прикладной анализ данных".**

**Курс "Инструментарий Data Science".**

**Преподаватель ФКН НИУ ВШЭ Кашницкий Юрий**



## **Урок 2. Основы языка Python**

### **Часть 3. Строки**

**Строки выделяются одинарными или двойными кавычками**

In [ ]:

```
my_string = "This is a double-quoted string."  
my_string = 'This is a single-quoted string.'
```

In [ ]:

```
quote = "Linus Torvalds once said, 'Any program is only as good as it is use-  
ful.'"
```

In [ ]:

```
first_name = 'eric'  
  
print(first_name)  
print(first_name.title())
```

## Разные регистры

In [ ]:

```
first_name = 'eric'

print(first_name)
print(first_name.title())
print(first_name.upper())

first_name = 'Eric'
print(first_name.lower())
```

*upper()*, *lower()* и *title()* - это методы.

Синтаксис: `variable_name.action()`

*action* - это название метода, который можно применить к переменной *variable\_name*. В скобках можно указывать другие переменные (аргументы) метода.

## Конкатенация ("склеивание") строк

In [ ]:

```
first_name = 'ada'
last_name = 'lovelace'

full_name = first_name + ' ' + last_name

print(full_name.title())
```

In [ ]:

```
first_name = 'ada'
last_name = 'lovelace'
full_name = first_name + ' ' + last_name

message = full_name.title() + ' ' + "was considered the world's first computer programmer."

print(message)
```

## Пробельные символы (whitespaces)

- " " - пробел (очевидно)
- "\t" - табуляция
- "\n" - перенос строки

In [ ]:

```
print("Hello everyone!")
```

In [ ]:

```
print("\tHello everyone!")
```

In [ ]:

```
print("Hello \teveryone!")
```

In [ ]:

```
print("Hello everyone!")
```

In [ ]:

```
print("\nHello everyone!")
```

In [ ]:

```
print("Hello \neveryone!")
```

In [ ]:

```
print("\n\n\nHello everyone!")
```

### Удаление лишних пробельных символов

Часто при заполнении веб-форм появляются лишние пробелы слева и/или справа. От них можно избавиться

In [ ]:

```
name = ' eric '  
  
print(name.lstrip())  
print(name.rstrip())  
print(name.strip())
```

Так нагляднее:

In [ ]:

```
name = ' eric '  
  
print('-' + name.lstrip() + '-')  
print('-' + name.rstrip() + '-')  
print('-' + name.strip() + '-')
```

Избавляться можно не только от пробелов. Здесь '-' - аргумент методов *lstrip()*, *rstrip()* и *strip()*

In [ ]:

```
name = ',,eric,,'  
  
print('-' + name.lstrip(',') + '-')  
print('-' + name.rstrip(',') + '-')  
print('-' + name.strip(',') + '-')
```

## Методы для работы со строками

Обзор на [Pythonworld \(http://pythonworld.ru/ipy-dannyx-v-python/stroki-funkcii-i-metody-strok.html\)](http://pythonworld.ru/ipy-dannyx-v-python/stroki-funkcii-i-metody-strok.html)

Все методы строк:

In [ ]:

```
dir(str)
```

**find** - поиск подстроки в строке. Возвращает номер первого вхождения или -1

In [ ]:

```
"we bought a new house and we are happy".find("we")
```

**rfind** - поиск подстроки в строке. Возвращает номер последнего вхождения или -1

In [ ]:

```
"we bought a new house and we are happy".rfind("we")
```

**index** - поиск подстроки в строке. Возвращает номер первого вхождения или вызывает ValueError

In [ ]:

```
"we bought a new house and we are happy".index("we")
```

In [ ]:

```
"we bought a new house and we are happy".index("land")
```

**rindex** - поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает ValueError

In [ ]:

```
"we bought a new house and we are happy".rindex("we")
```

In [ ]:

```
"we bought a new house and we are happy".rindex("land")
```

**replace** - замена по шаблону

In [ ]:

```
help(str.replace)
```

In [ ]:

```
"we bought a new house".replace("house", "building")
```

In [ ]:

```
"bla bla bla".replace("bla", "wow", 1)
```

**split** - разбиение строки по разделителю

In [ ]:

```
"hello#$great#$to#$see#$you".split("#$")
```

- **isdigit** - состоит ли строка из цифр
- **isalpha** - состоит ли строка из букв
- **isalnum** - состоит ли строка из цифр или букв
- **islower** - состоит ли строка из символов в нижнем регистре
- **isupper** - состоит ли строка из символов в верхнем регистре
- **istitle** - начинаются ли слова в строке с заглавной буквы

**join** - сборка строки из списка по разделителю, применяется к разделителю

In [ ]:

```
", ".join(["Ann", "Leo", "Tiffany"])
```

**format** - форматирование строки

Пример

In [ ]:

```
# key-value pairs: client_id, (name, age, occupation)
personal_data = {1: ("Ann", 29, "Artist"),
                 2: ("Leo", 54, "Programmer"),
                 3: ("Tiffany", 43, "Shop Assistant")}
for (name, age, occupation) in personal_data.values():
    print("{0} (age {1}) is a {2}".format(name, age, occupation))
```