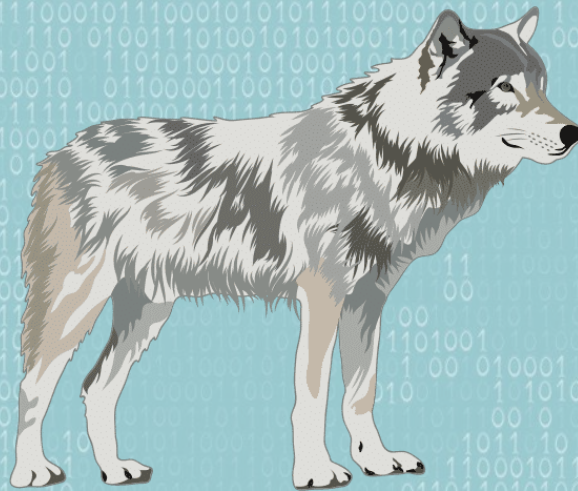




РЕБУС



`s[:-2]`



`s[:-1]`

РЕБУС ИН ІНГЛІШ



+



s[:2]

ПОВТОРЕНИЕ

`s = "ГЛИНА"`

Как вывести **НА**?

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

Как вывести **ГАЛИНА**?

```
print(s[0] + s[-1] + s[1:])
```

Что будет напечатано при `print(s[::2])`?

ГИА

ИНТЕРЕCHO...

```
dice = 49
```

```
# red_dice = dice // 10
```

```
# green_dice = dice % 10
```

```
red_dice = int(str(dice)[0])
```

```
green_dice = int(str(dice)[1])
```


СЕГОДНЯ...

- Узнаем, что такое **методы**
- Научимся обрабатывать строки с помощью методов
- Обсудим лайфхаки обработки текста

ЕСЛИ БЫ...

Если бы вы могли добавить в Python любую команду для работы со строками, что бы вы добавили?

НА САМОМ ДЕЛЕ...

На самом деле, многие из этих команд уже существуют!

МЕТОДЫ

Метод – это функция,
связанная с определенным **типом данных**

Строки в Python имеют встроенные методы
для выполнения различных операций со строками

ВЫЗОВ МЕТОДА

`string_name.method_name(args)`

название
строки

точка
(оператор
принадлежности)

название
метода

аргументы
в скобках

```
word = "абракадабра"
```

```
word.replace("абра", "ухи")
```

```
# ухикадухи
```



КОМАНДНАЯ РАБОТА

- Работа в парах
- Подготовить презентацию о методе
- Рассказать всей группе про метод



КОМАНДНАЯ РАБОТА

СОДЕРЖАНИЕ ПРЕЗЕНТАЦИИ

- Название метода и его перевод
- Какие параметры есть у метода
- Что делает метод
- Что возвращает метод
- Пример кода (вызвать метод и показать результат)

КОМАНДНАЯ РАБОТА

ПРИМЕР ПРЕЗЕНТАЦИИ

- Метод `replace` (англ. «заменить»)
- Параметры: исходная подстрока, новая подстрока
- Заменяет все исходные подстроки в строке на новые
- Возвращает исправленную строку

```
word = "абракадабра".replace("абра", "ухи")
```

```
# "ухикадухи"
```

МЕТОДЫ

- 1 Сделай все буквы в строке маленькими!
- 2 Найди индекс подстроки в строке!
- 3 Посчитай, сколько раз подстрока встречается в строке!
- 4 Убери лишние пробелы слева и справа от строки!
- 5 Скажи, строка содержит только цифры, или нет?

МЕТОДЫ

Название, параметры, возвращаемое значение метода	Пример и результат
replace(string1, string2) -> str Возвращает строку, в которой все подстроки string1 заменены на подстроку string2	<pre>word = "абракадабра" word = word.replace("абра", "ухи") # "ухикадухи"</pre>
lower() -> str Возвращает строку со всеми маленькими буквами	<pre>word = "ЭгЕгЕй!" word = name.lower() # "эгегей!!!"</pre>
find(string) -> int Возвращает индекс первого вхождения string в строку	<pre>word = "абракадабра" num = word.rfind("кад") # 4</pre>

МЕТОДЫ

Название, параметры, возвращаемое значение метода	Пример и результат
count(string) -> int Возвращает количество раз, которое подстрока string встречается в строке	<pre>word = "абракадабра" num = word.count("абра") # 2</pre>
strip() -> str Возвращает строку с убранными символами whitespace в начале и конце строки	<pre>sentence = " Дядя Петя\n" sentence = sentence.strip() # "Дядя Петя"</pre>
isnumeric() -> bool Возвращает True , если все символы в строке – это цифры, иначе False	<pre>word = "1, 2, 3" word.isnumeric() # False</pre>

МЕТОДЫ

Название, параметры, возвращаемое значение метода	Пример и результат
upper() -> str Возвращает строку со всеми заглавными буквами	<pre>word = "yPa-yPa!!!" word = name.upper() # "УРА-УРА!!!"</pre>
capitalize() -> str Возвращает строку, в которой первая буква заглавная, а все остальные маленькие	<pre>name = "гАрри пОттЕр" name = name.capitalize() # "Гарри поттер"</pre>
swapcase() -> str Возвращает строку, в которой размер каждой буквы заменен на противоположный	<pre>name = "ХаРиТоН?" name = name.swapcase() # "хАрИтОн?"</pre>

МЕТОДЫ

Название, параметры, возвращаемое значение метода	Пример и результат
isalpha() -> bool Возвращает True , если все символы в строке – это буквы, иначе False	word = “Кошка Матрёшка” word.isalpha() # False
isalnum() -> bool Возвращает True , если все символы в строке – это буквы или цифры, иначе False	word = “abc123” word.isalnum() # True
rfind(string) -> int Возвращает индекс последнего вхождения string в строку	word = “абракадабра” num = word.rfind(“абра”) # 7

СТРОКИ И ЦИКЛЫ

Какой тип циклов мог бы быть полезен
в работе со строками?

ЦИКЛ FOR

```
for i in range(5):
```

```
    print(i)    # что будет выведено на экран?
```

0

1

2

3

4

ЦИКЛ FOR

```
for ...:  
    print(i)
```

Как сделать, чтобы таким же образом
на экран выводились символы из строки?

ЦИКЛ FOR

```
word = "Ура!"
```

```
for i in word:
```

```
    print(i)
```

У

р

а

!



ЦИКЛ FOR

```
sentence = "Мама мыла раму..."
```

```
for ...:  
    print(i)
```

Как сделать, чтобы таким же образом
на экран выводились **слова** из строки?

ЦИКЛ FOR

```
sentence = "Мама мыла раму..."
```

```
for i in sentence.split():
```

```
    print(i)
```

Мама

мыла

раму...



ОПЕРАТОР IN

```
word = "строка"
```

```
print("a" in word)
```

Как вы думаете, что будет напечатано?

ОПЕРАТОР IN

word = "строка"

print("a" in word)

True



оператор
проверки наличия
подстроки в строке



ИТОГИ ЗАНЯТИЯ

- Узнали, что такое методы
- Познакомились с многими методами строк
- Выяснили, как проходиться по строкам с помощью цикла **for** и метода **split()**, научились проверять наличие подстроки в строке с помощью оператора **in**

Вопросы?

КАНАЛ TELEGRAM

