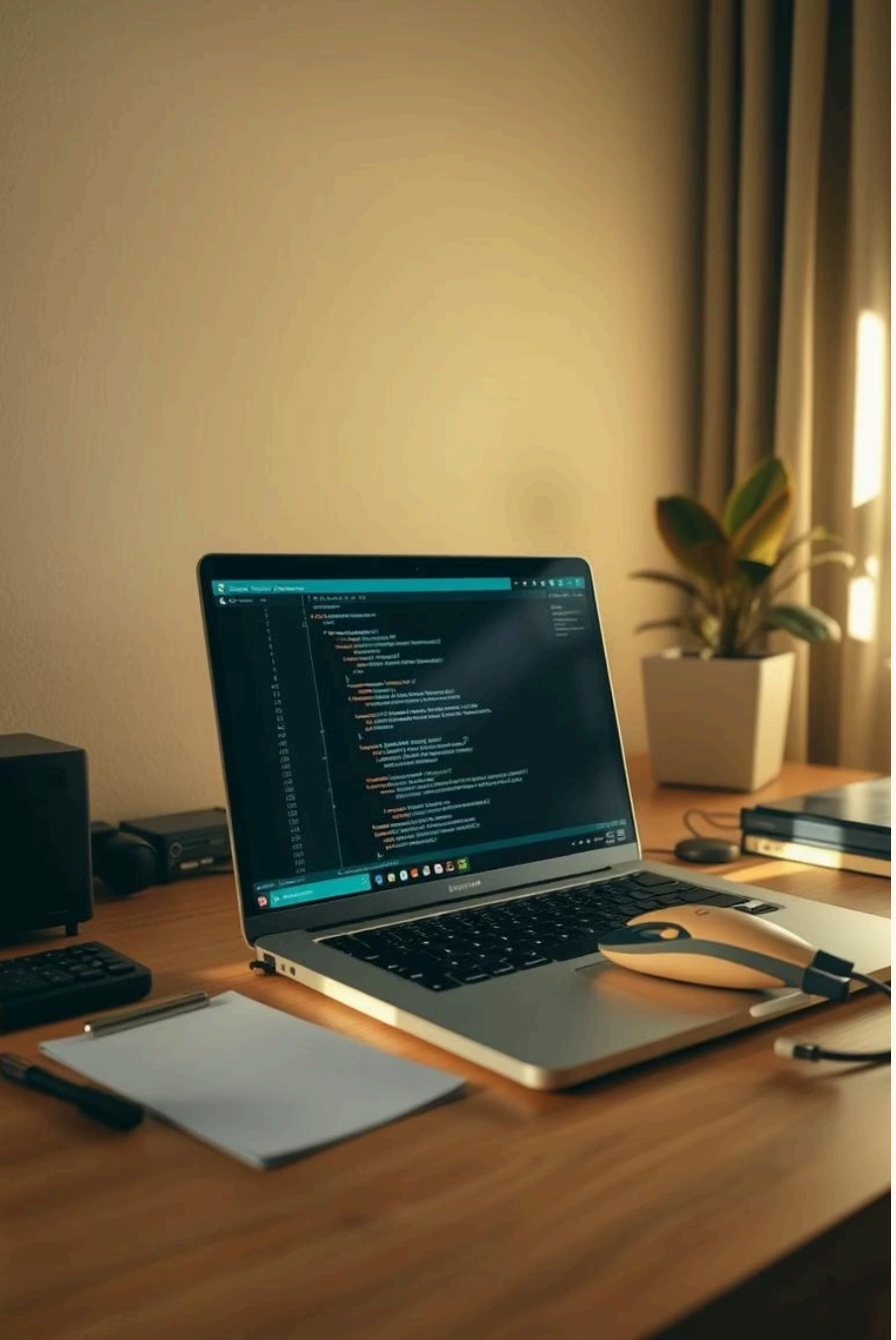


Текстовые строки в языке программирования Python

План презентации:

1. Введение
2. Создание текстовых строк в языке Python
3. Действия с текстовыми строками в языке Python
4. Встроенные функции языка Python для работы с текстовыми строками
5. Форматирование
6. Заключение

Введение

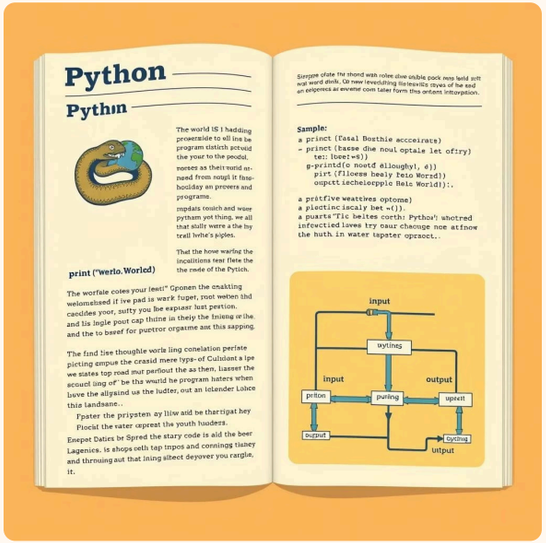
Текстовые строки в языке Python — это основа работы с текстом и ключ к эффективной коммуникации. Они распространены повсеместно и используются при каждом создании кода. Понимание основ работы текстовых строк в языке Python, а также основных функций и действий над ними — ключ к эффективному коду.

Создание текстовых строк в языке Python

Строка в Python создается заключением символов в парные одинарные или двойные кавычки, редко - в тройные одинарные либо двойные:

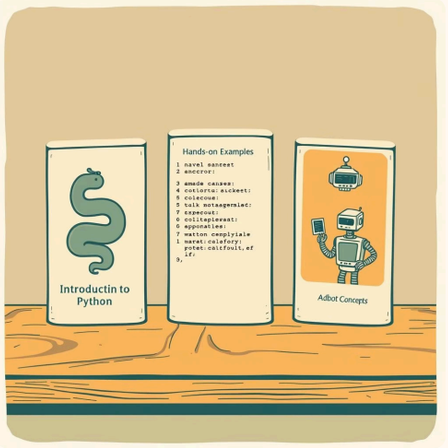
Одинарные кавычки

'Dog'



Парные кавычки

"Dog"



Допускается использование трех одинарных или трех двойных кавычек *(для создания многострочного текста , к примеру, текста песенки)*

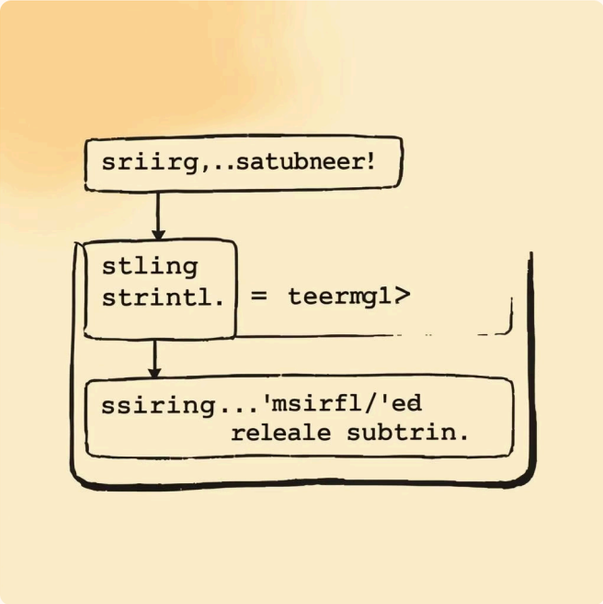
poem = '''That was a farmer had a dog,

... and that was mini Dingo,

...D-I-N-G-O,

...and Dingo was his name."""

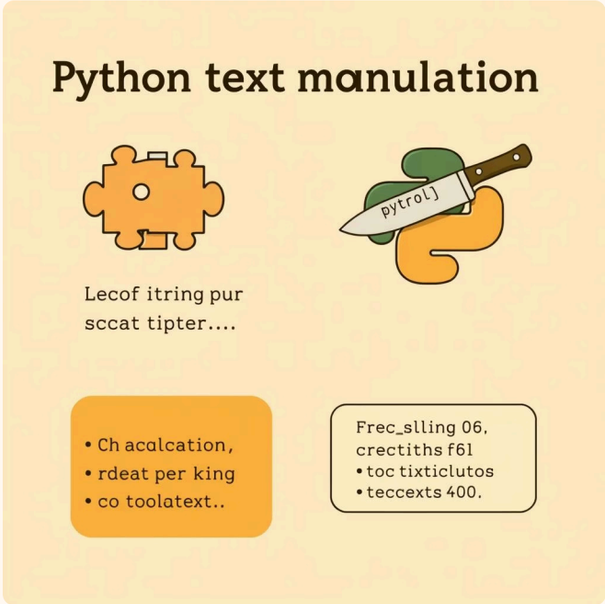
Действия с текстовыми строками в языке Python



Создание текстовую строку из иного типа данных при помощи функции str():

>>> str(91.1)

'91.1'



Объединение строк с использованием символа +

>>> 'Come on John! ' + 'No time to wait!'

'Come on John! 'No time to wait!'

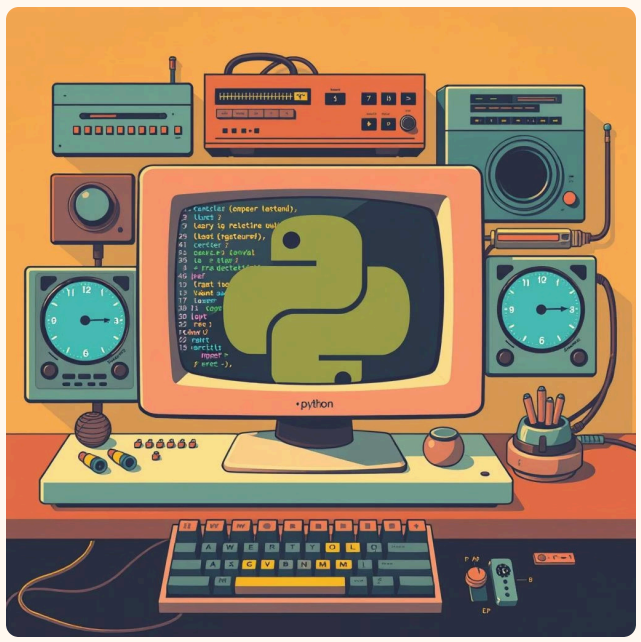
Размножение строк с помощью символа *

text = 'wow'

result = text * 3

print(result)

'wow wow wow'



Извлечение символов с помощью квадратных скобок [ ]

Для получения одного символа из строки укажите его индекс в квадратных скобках (где первый символ имеет индекс 0, последний -1).

```
>>> letters = 'abcdefghijklmnopqrstuvwxyz'
>>>
 letters[0]
'a'
>>> letters[1]
'b'
>>> letters[-1]
'z'
```

Извлечение подстроки через оператор разделения [:]

Для извлечения подстроки используется оператор [:]

```
>>> letters = 'abcdefghijklmnopqrstuvwxyz'
```

[:] извлекает всю последовательность

```
>>> letters[:]
'abcdefghijklmnopqrstuvwxyz'
```

[начало:] извлекает всю последовательность с указанной точки до конца

```
>>> letters[20:]
 'uvwxyz'
```

[:конец] извлекает всю последовательность до указанной точки

```
>>> letters[-3:]
'xyz'
```

Встроенные функции языка Python для работы с текстовыми строками

Функция len()

Возвращает длину строки

>>> len("Python")

6.

Функция split()

Разделяет строку по разделителю

>>>"a,b,c".split(",")

['a','b','c'].

Функция join()

Объединяет элементы списка в строку через разделитель

>>>join(['Hello', 'world'])

'Hello world'.

Функция replace()

Заменяет подстроки и удаляет пробелы по краям

>>>"2024-01-01".replace("-", ".")

"2024.01.01".

Функция strip()

Удаляет пробелы и управляющие символы по краям строки.

>>>clean ".strip()

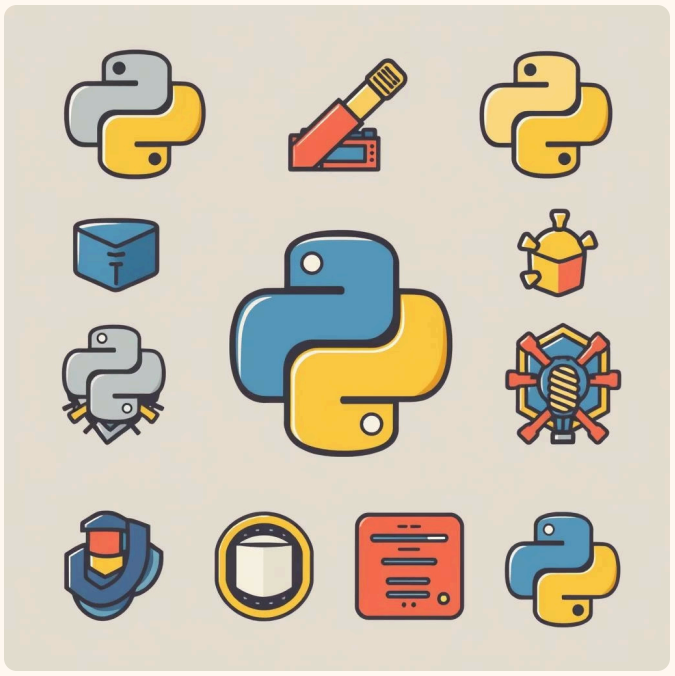
"clean".

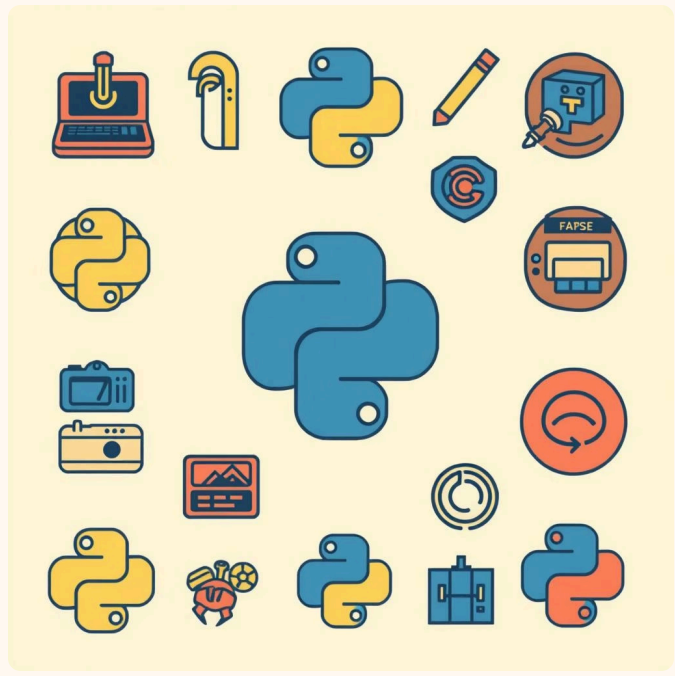
Иные функции в языке Python:

1. Проверка содержимого: isalpha(), isdigit(), islower(), isupper().
2. Изменение регистра: upper(), lower(), capitalize(), title().
3. Поиск и индексация: find(), index(), in.
4. Срезы и обращение по индексу: s[0:4], s[::-1]
5. и др.

Форматирование

на примере Python 3





Символы {} и функция format()

Новый стиль форматирования имеет вид строка.format(данные).

```
 >>> thing = 'woodchuck'

>>> '{}'.format(thing)

'woodchuck'
```

Новый стиль: f-строки

F-строки появились в версии Python.3.6. Чтобы создать f-строку, нужно сделать следующее:

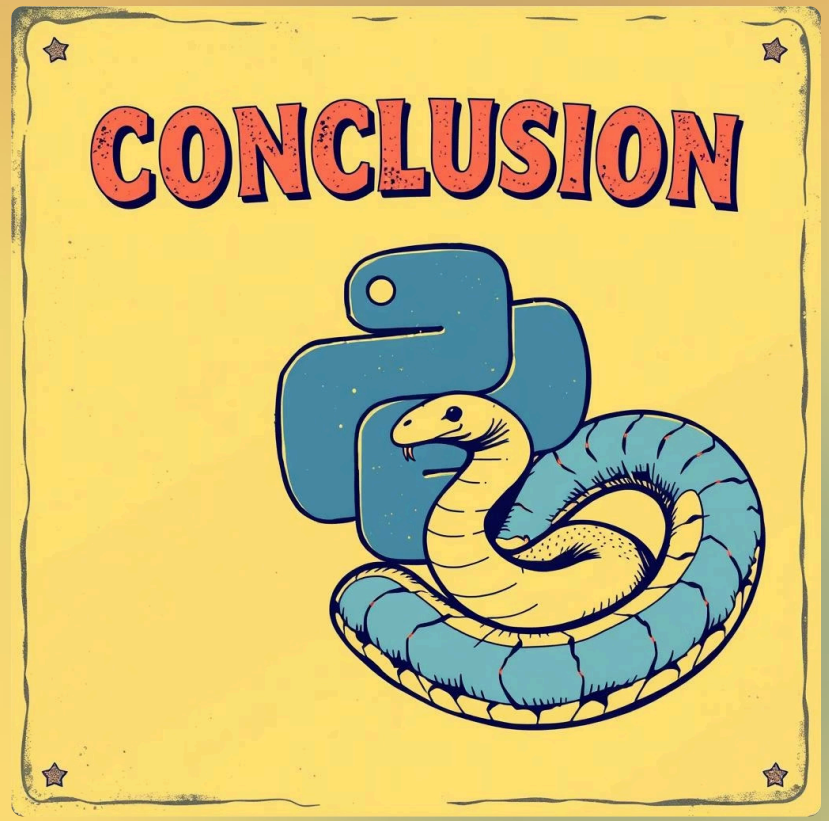
1. Ввести буквы f или F перед кавычкой;
2. Поместить имена переменной или выражения в фигурные скобки ({}), чтобы их значения попали в строку.

```
>>> thing = 'wereduck'

>>> place = 'werepond'

>>> f'The {thing} is in the {place}'

'The wereduck is in the werepond'
```



Заключение

В этой презентации рассмотрены основы работы с текстовыми строками в языке Python, включая создание, обработку и форматирование строк.

Использование встроенных функций и современных методов, таких как f-строки, позволяет писать более эффективный и читаемый код.

Презентация представляет краткий обзор возможностей языка Python в отношении текстовых строк, а также дает основу для дальнейшего более детального изучения данной темы.

Спасибо за внимание!