**Строки в Python.**

В данном уроке мы разберем чем же являются строки и разберём основные операции со строками.

***Определение:***

**Строки в языке программирования Python — это объекты, которые состоят из последовательности символов.**

С точки зрения Питона, **строка** — это упорядоченная последовательность символов, которая предназначена для хранения информации в виде простого текста.

Тип данных – string.

**Как задаётся строка?**

В Python есть несколько способов для создания строки.

Первый из них - это одинарные апострофы 'текст'. Первый апостроф открывает строку, затем следует сам текст, и вторым апострофом вы закрываете строку.

print('Привет мир!')

Второй способ - это кавычки "текст". Также первыми кавычками вы открываете строку, пишете текст и закрываете строку кавычками.

print("Привет двойные кавычки!")

Третий способ необходим для создания многострочных строк. Если в IDLE вы сначала введете три одинарных апострофа, введете текст и затем нажмете Enter, то программа будет ждать от вас дальнейшего ввода. Дальше вы сможете писать текст, также переходить на новую строку, затем закрываете также одинарными апострофами и получаете строку такого вида:

print("""Привет   
многострочные  
кавычки!  
""")

\*При несоответствии количества апострофов в начале и конце строки Python выдаст ошибку.

**Операции со строками.**

1. Конкатенация (сложение).

В этой операции две заданные строки слепляют в одну.

S1 = 'spam'  
S2 = '12345'  
print(S1 + S2)

В результате получим:

spam12345

2. Дублирование строки.

В этой операции программист может вызывать строку необходимое количество раз.

print("stroka"\*4)

В результате получим:

strokastrokastrokastroka

1. Нахождение длины строки (функция len).

При необходимости узнать длину строки нужно воспользоваться функцией len.

print(len("stroka"))

В результате получим:

6

4. Доступ по индексу.

К строке можно обращаться по индексам, как к массиву, только элементами строки будут символы.

Для наглядности выведем все индексы строки “Mospo123”.

s="Mospo123"  
print(s[0])  
print(s[1])  
print(s[2])  
print(s[3])  
print(s[4])  
print(s[5])  
print(s[6])  
print(s[7])

В результате получим:

M

o

s

p

o

1

2

3

5. Извлечение среза.

В этой операции из строки выделяют определённые символы(срез). Для ff

Для наглядности из строки “'Kto mi kto mi kto mi'” выделим строку “mi”. Для этого нужно подобрать такие значения, которые будут соответствовать индексам символа “m” и “i”. Последний элемент среза не учитывается.

s = 'Kto mi kto mi kto mi'  
print(s[4:6])

В результате получим:

mi

6. Проверка вхождения.

Ещё одна интересная операция для строк – проверка вхождения символа в строку. Эта операция выводит только 2 значения – “True” и “False”.

Для наглядности проверим вхождения символа “o” в строку “Moscow”. После проверим вхождения символа “777” в строку “Moscow”.

s = 'Moskow'  
print("o" in s)

В результате получим:

True

s = 'Moskow'  
print("777" in s)

В результате получим:

False

7. Сравнение строк.

Последняя из необходимых любому программисту базовая операция со строками – сравнение. Чтобы сравнить 2 строки можно использовать операторы «==, <, >».

Этот метод лучше всего закрепляется на практике.

Ниже приведены примеры.

language = 'russian'  
print(language == 'chinese')  
print(language != 'chinese')

В результате получим:

False

True

chislo = '12345'  
print(chislo > '1234')  
print(chislo < '99')

В результате получим:

True

True

**Дополнительный материал.**

На этом уроке вы познакомились со строками и изучили основные операции, однако это не все возможные операции со строками в языке программирования питон.

Ниже представлены оставшиеся методы, используемые при работе со строками.





