

Строковый тип данных в Python.

Строка – Строка – это упорядоченная последовательность символов, которая предназначена для хранения информации в виде простого текста. В Python строковые данные имеют формат **str**.

Создадим строковые переменные *s*, *S*, для того чтобы создать строку в Python используют либо одинарные кавычки, либо двойные.

ИСХОДНЫЙ КОД:

```
S='мы изучаем Пайтон'  
print(S)  
  
s="мы изучаем Пайтон"  
print(s)
```

результат:

```
===== RESTART:  
мы изучаем Пайтон  
мы изучаем Пайтон  
>>> |
```

Если нужно создать большой фрагмент текста, то используют тройные кавычки.

ИСХОДНЫЙ КОД:

```
s='''мы изучаем язык  
программирования Пайтон'''  
print(s)
```

результат:

```
===== RESTART: C:  
мы изучаем язык  
программирования Пайтон  
>>>
```

При использовании тройных кавычек, задаются многострочные тексты.

Функция **str()**.

С помощью функции **str()** можно преобразовать в строку любое число.

Основные операции со строками.

Сложение строк (конкатенация).

ИСХОДНЫЙ КОД:

```
s1='Мы изучаем '  
s2='Пайтон '  
s=s1+s2  
print(s)
```

результат:

```
=====  
===== RESTART: C:  
Мы изучаем Пайтон  
>>> |
```

Строковый тип данных, можно складывать только со строковым типом данных. Например, если к строке прибавить число, то программа выдаст ошибку `s1+7`. Чтобы произвести соединение, нужно число 7 преобразовать в строковый тип данных используя функцию **str()** или кавычки.

ИСХОДНЫЙ КОД:

```
s1='Пайтон '  
s=s1+str(3)  
print(s)
```

результат:

```
===== RESTART:  
Пайтон 3  
>>>
```

Умножение строк(дублирование строк).

ИСХОДНЫЙ КОД:

```
s1='Пайтон '  
s=s1*7  
print(s)
```

результат:

```
===== RESTART: C:\1.py :  
Пайтон Пайтон Пайтон Пайтон Пайтон Пайтон Пайтон  
>>>
```

Операция умножения является дублированием строки. Такое действие можно выполнять только целое число раз. При умножении на дробное число программа выдаст ошибку.

Длина строки len().

Функция `len()` позволяет определить длину строки т.е. количество символов в строке.

ИСХОДНЫЙ КОД:

```
s=len('Мы изучаем Пайтон '  
print(s)
```

результат:

```
-----  
===== RESTART:  
18  
>>>
```

Оператор in (проверка вхождения под строки в строку)

Исходный код:

```
a='мы' in 'мы изучаем Пайтон'  
print(a)
```

результат:

```
===== RESTART: C:\1.py  
True  
>>> |
```

Операторы сравнения (==, !=, >, <)

С помощью данных операторов можно сравнивать строки.

Код символа ord().

Получение кода ASCII по его символу. Например,

Исходный код:

```
print(ord('y'))
```

результат:

```
=====  
121  
.
```

Определение символа по его коду chr().

Получение символа по его коду ASCII. Например,

Исходный код:

```
print(chr(67))
```

результат:

```
=====  
C  
>>> |
```

Индексы.

Строка это упорядоченная последовательность. Рассмотрим на примере строки, “Мы изучаем Python”, каждый символ в данной строке имеет свой порядковый номер и нумерация начинается с 0 слева направо или справа налево начиная с -1 и т.д.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
М	ы		и	з	у	ч	а	е	м		Р	у	t	h	o	n
-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Для того чтобы выбрать определенный символ в строке, можем обратиться к нему по индексу.

Исходный код:

```
s='Мы изучаем Python'  
print(s[5])  
print(s[11])  
print(s[-3])
```

результат:

```
=====  
у  
Р  
h  
>>> |
```

Срезы строк

Существуют способы, с помощью которых, можно выделить часть строки (подстроки) из строк. Например,

Исходный код:

```
s='Мы изучаем Python'  
t=s[3:10]  
print(t)
```

результат:

```
=====  
изучаем  
>>> |
```

Мы сделали срез подстроки из строки начиная с 3 индекса до 10, причем символ стоящий под номером 10 не входит в срез.

Исходный код:

```
s='Мы изучаем Python'  
t=s[-5:-3]  
print(t)
```

результат:

```
=====  
уt  
>>>
```

Можно делать срез строки используя шаг **строка[нач:кон:шаг]**

Исходный код:

```
s='Мы изучаем Python'  
t=s[1:10:2]  
print(t)
```

результат:

```
=====  
ьиуам  
>>>
```

Сделан срез строки с 1 до 10 символ, в который вошел каждый 2 символ.

Задачи:

Задание 1. Написать программу, которая спрашивает у пользователя его имя и фамилию, а затем выводит приветствие вида «Привет, Имя Фамилия!» с использованием оператора +.

Задание 2. Написать программу, которая спрашивает у пользователя слово и повторяет его три раза через пробел с использованием оператора * \

Задание 3. Напишите программу, которая принимает строку от пользователя и выводит на экран первые 3 символа строки. Если строка короче 3 символов, выведите всю строку.

Задание 4. Напишите программу, которая принимает строку от пользователя и выводит на экран последние 3 символа строки. Если строка короче 3 символов, выведите всю строку.

Задание 5. Напишите программу, которая принимает строку от пользователя и выводит на экран символы с 2 по 5. Если строка короче 5 символов, выведите всю строку.

Задание 6. Напишите программу, которая принимает строку от пользователя и выводит на экран каждый третий символ начиная с 2-го символа. Если строка короче 2 символов, выведите всю строку.

Задание 7. Напишите программу, которая принимает строку от пользователя и выводит на экран каждый второй символ в обратном порядке. Если строка короче 2 символов, выведите всю строку.

Задание 8. Напишите программу, которая принимает строку от пользователя и выводит на экран символы с шагом 2 начиная со второго символа и до предпоследнего символа. Если строка короче 3 символов, выведите всю строку.

Задание 9. Написать программу, которая принимает на вход строку и выводит первую и последнюю букву строки, а также все символы между ними в обратном порядке.

Задание 10. Дана строка. Выведите третий символ этой строки.

Задание 11. Дана строка. Выведите предпоследний символ этой строки.

Задание 12. Дана строка. Выведите первые пять символов этой строки.

Задание 13. Дана строка. Выведите всю строку кроме последних двух символов этой строки.

Задание 14. Дана строка. Выведите все символы с четными индексами (считая, что индексация начинается с 0, поэтому символы выводятся начиная с первого).

Задание 15. Дана строка. Выведите все символы с нечетными индексами т.е. есть начиная со второго символа.

Задание 16. Дана строка. Выведите все символы в обратном порядке.

Задание 17. Дана строка. Выведите все символы строки через один в обратном порядке, начиная с последнего.