

Строки и операции над ними
Индексы и срезы

Строки в Python – упорядоченные неизменяемые последовательности символов, используемые для хранения и представления текстовой информации

Способы создания и сохранения строк

1. С помощью одинарных кавычек(апострофов) `s='hello'`
2. С помощью двойных кавычек `s="hello"`
3. С помощью тройных кавычек
`s = """Это очень длинная строка, ей нужно много места"""`
4. С помощью метода `str ()`.

```
my_num = 12345  
my_str = str(my_num)
```

Операции над строками

Операция конкатенация (сцепление) – сложение строк

```
a = 'Вот так работает'
```

```
b = ' конкатенация строк'
```

```
print (a + b)
```

Вот так работает конкатенация строк

Операции над строками

Операция повторения – умножение строк. Если *s* это строка, а *n* целое число, то будет создано *n* копий строки *s*.

```
s = 'Строка'  
print (5 * s)
```

СтрокаСтрокаСтрокаСтрокаСтрока

Нахождение длины строки

Использование функции `len()`.

Эта встроенная функция возвращает количество символов в исследуемой нами строке, определяя таким образом её длину.

Находим длину строки в Python с помощью функции `len()`

```
str = 'Hello'  
print(len(str))
```

Итогом работы функции станет следующий вывод :

5

Индекс строки в Python

Строка представляет собой упорядоченный набор символов. Это значит, что каждый символ в строке занимает свое место - **индекс**.

0	1	2	3	4	5	6	7	8	9	10
H	E	L	L	O		W	O	R	L	D
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Срез в Python

Срезы (slices) — это извлечение из данной строки одного символа или некоторого фрагмента (подстроки).

Срезы создаются с помощью квадратных скобок [] и двоеточия : .

Синтаксис среза выглядит следующим образом:

`iterable_object[start:stop:step]`

- `start` — индекс начала среза (включительно);
- `stop` — индекс окончания среза (исключительно);
- `step` — шаг (по умолчанию равен 1).

Индексация в Python начинается с нуля.

```
alphabet = 'abcdefg'
```

```
print(alphabet[:])
```

abcdefg

```
print(alphabet[::-1])
```

gfedcba

```
print(alphabet[0::2])
```

aceg

```
print(alphabet[:2])
```

ab

```
print(alphabet[:-3:-1])
```

gf

```
print(alphabet[3:4])
```

d

```
print(alphabet[4:])
```

efg

```
print(alphabet[3:5])
```

de

```
print(alphabet[-3:-5:-1])
```

ed

