

Каждый символ строки имеет свой порядковый номер (*индекс*). Нумерация символов начинается с нуля. Теперь мы можем обратиться к заданному символу строки следующим образом:

```
>>> s = 'Я люблю писать программы!'
>>> s[0]
'Я'
>>> s[-1]
'!'
>>>
```

В квадратных скобках указывается индекс символа. Нулевой индекс – первая буква строки. А -1 индекс? Можно догадаться, что последний. Если увидите отрицательный индекс, то определить его положительный аналог можно как длина строки + отрицательный индекс. Например, для -1 это будет: `len(s) - 1`, т.е. 24.

```
>>> len(s) - 1
24
>>> s[24]
'!'
>>>
```

Какая ситуация с изменением строк в Python?

```
>>> s = 'Я люблю писать программы!'
>>> s[0] = 'J'
Traceback (most recent call last):
  File "<pyshell#41>", line 1, in <module>
    s[0] = 'J'
TypeError: 'str' object does not support item assignment
>>>
```

Попытка изменить нулевой символ в строке **s** привела к ошибке. Дело в том, что в Python строки, как и числа, являются неизменяемыми.

Работа со строковыми объектами для Python не отличается от работы с числовыми объектами:

