



Программирование на Python

Презентация занятия

Изучение возможностей и синтаксиса Python: Строки и работа с файлами.

7 занятие



инжинириум[®]

МГТУ им. Н.Э. Баумана

2019

Тема: Изучение возможностей и синтаксиса Python: Строки и работа с файлами.

Строка – базовый тип представляющий из себя **неизменяемую последовательность символов**

Одиночные кавычки. Часто встречаемый вариант записи.

```
my_str = 'а внутри "можно" поместить обычные'  
my_str = "а внутри 'можно' " + " поместить одиночные"
```

```
example_string = "Курс по Python"  
print(example_string)  
Курс по Python  
print(type(example_string))  
<class 'str'>
```

При этом отдельного символьного типа в Python нет, символ – это строка длины



Тема: Изучение возможностей и синтаксиса Python: Строки и работа с файлами.

Базовые операции

1. Арифметические операции.

```
S1 = 'Py '  
S2 = 'thon '  
S3 = S1 + S2  
print(S3)  
Python
```

```
S3 * 4  
PythonPythonPythonPython
```

2. Оператор in

```
S = 'Python '  
SubS = 'th '  
SubS in S  
True
```



Тема: Изучение возможностей и синтаксиса Python: Строки и работа с файлами.

3. Функция `len()` вычисляет длину строки, результат имеет целочисленный тип. Например, `len('Python')` выдаст 6.

4. Доступ по индексу

```
S = 'Python '
```

```
print(S [0])
```

```
'P'
```

```
print(S[-1])
```

```
'n'
```

```
S = 'Ура'
```

```
S[1] = 'x'
```

```
Traceback ( most recent call last ):
```

```
File "<pyshell #68 >", line 1, in <module >
```

```
TypeError : 'str ' objet does not support item assignment
```

```
print(S [0]+ 'x'+S[2])
```

```
'Уха'
```



Тема: Изучение возможностей и синтаксиса Python: Строки и работа с файлами.

Файлы

Открытие файла

```
f = open('filename')
```

Файлы можно открывать по-разному --- на запись, на чтение, на чтение и запись, на дозапись.

```
text_modes = ['r', 'w', 'a', 'r+']
```

```
binary_modes = ['br', 'bw', 'ba', 'br+']
```

```
f = open('filename', 'w')
```

Запись в файл

Чтобы записать в файл, применяем к соответствующему файловому объекту метод `write`, передавая ему строку.

```
f.write('The world is changed.\nI taste it in the water.\n')
```

Когда работа с файлом завершена, его нужно закрыть

```
f.close()
```



Тема: Изучение возможностей и синтаксиса Python: Строки и работа с файлами.

Итак, чтобы открыть файл на чтение и запись, нам нужно использовать r+.

```
f = open('filename', 'r+')
```

```
f.read()
```

```
'The world is changed.\nI taste it in the water.\n'
```

```
f.tell()
```

```
47
```

Для того, чтобы прочесть одну строку из файла, есть метод `readline`, а чтобы разбить

файл на строки и поместить их в список --- метод `readlines`.

```
f = open('filename', 'r+')
```

```
f.readline()
```

```
'The world is changed.\n'
```

```
f.close()
```



Тема: Изучение возможностей и синтаксиса Python: Строки и работа с файлами.

Использование контекстного менеджера

```
with open('filename') as f:  
    print(f.read())
```

```
'The world is changed.\nI taste it in the water.\n'
```

В этом случае файл закрывать не нужно

