



GeekBrains

Основы Python



GeekBrains

Урок 5

Работа с файлами

На этом уроке

1. Работа с данными: чтение и запись.
2. Менеджеры контекста.
3. Выявление ошибок при работе с файлами.
4. Режимы доступа к файлу.
5. Параметры файлового объекта.
6. Определение позиции указателя в файле.
7. Print в файл.
8. Модуль json.
9. Модуль shutil.
10. Модуль sys.

Работа с файлами: открытие, закрытие, чтение, запись

Открытие



```
f_obj = open("my_file.txt", 'r')  
f_obj = open("my_file.txt")  
f_obj = open("my_file.txt", 'w')
```

Чтение



```
content = f_obj.read()  
content = f_obj.readline()  
content = f_obj.readlines()
```

Запись



```
content = f_obj.write()  
content = f_obj.writelines()
```

Закрытие



```
f_obj.close()
```



Менеджеры контекста

```
with open("text.txt") as f_obj:
```

Заккрытие  ~~f_obj.close()~~



Выявление ошибок при работе с файлами

```
try:
    f_obj = open("text.txt")
    for line in f_obj:
        print(line)
except IOError:
    print("Произошла ошибка ввода-вывода!")
finally:
    f_obj.close()
```



Режимы доступа к файлу

Режим	Описание
r	Открыть файл на чтение (режим по умолчанию)
w	Открыть на запись. При этом удалить содержимое файла. Если файла нет, создать новый.
x	Открыть файл на запись, если файл отсутствует. Если файл есть, он не будет создан.
a	Открыть файл на дозапись. Добавить информацию в конец файла.
b	Открыть файл в двоичном формате.
t	Открыть файл в текстовом формате (режим по умолчанию)
+	Открыть файл на чтение и запись

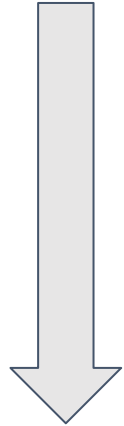
Параметры файлового объекта



Атрибут	Описание
file.closed	Возвращает значение True, если файл закрыт
file.mode	Возвращает режим доступа, по которому был открыт файл
file.name	Возвращает имя файла

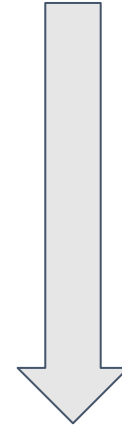
Определение позиции указателя в файле

Метод `tell()`



Определяет, в скольких
байтах от начала файла
находится указатель

Метод `seek()`



Позволяет выполнить
переход на нужную
позицию

Print в файл

```
with open("python.txt", "w") as f_obj:  
    print("Необычная работа функции print", file=f_obj)
```



Модуль os



`.remove()`

`.rename()`

`.listdir()`

`.path()`

`.path.basename()`

`.path.dirname()`

`.path.exists()`

`.path.isdir()`

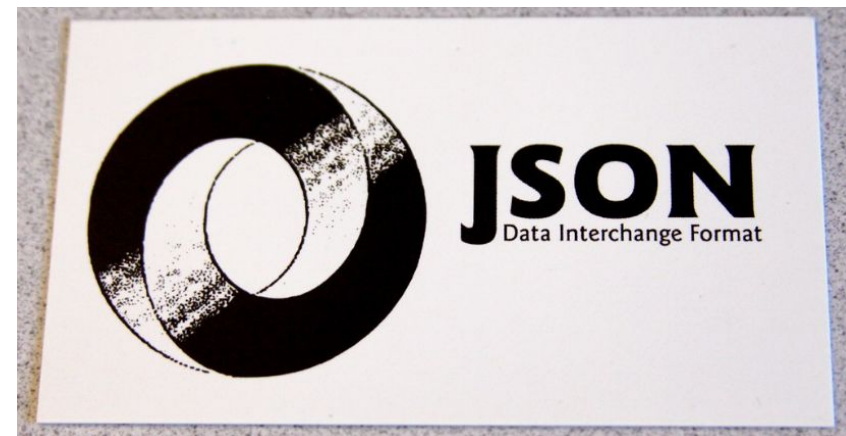
`.path.isfile()`

`.path.join()`

`.path.split()`

Модуль json. Часть 1

```
{  
  "worker": "Jon Smith",  
  "skills": ["programming", "design", "engineering"],  
  "age": 40,  
  "workplaces": [  
    {  
      "first": "IBM",  
      "time": "2010-2014"  
    },  
    {  
      "second": "Apple",  
      "time": "2014-2018"  
    }  
  ]  
}
```



Модуль json. Часть 2

Сериализация

Методы `dump()`,
`dumps()`

Python	JSON
<code>dict</code>	<code>object</code>
<code>list, tuple</code>	<code>array</code>
<code>str</code>	<code>string</code>
<code>int, long, float</code>	<code>number</code>
<code>True</code>	<code>true</code>
<code>False</code>	<code>false</code>
<code>None</code>	<code>null</code>

Модуль json. Часть 3

Десериализация

Методы `load()`,
`loads()`

JSON	Python
object	dict
array	list
string	str
number (int)	int
number (real)	float
true	True
false	False
null	None

Модуль shutil



`.copyfileobj()`

`.copyfile()`

`.copy()`

`.rmtree()`

`.move()`

Модуль sys

`sys.argv`

`sys.executable`

`sys.exit`

`sys.path`

`sys.platform`

`sys.stdin / stdout / stderr`



ИТОГИ

1. Научились работать с файлами: считывать и записывать данные.
2. Узнали о такой полезной возможности, как менеджер контекста.
3. Научились открывать файлы в различных режимах, определять параметры файлового объекта и позиции указателя.
4. Разобрали, как выявлять ошибки при работе с файловыми объектами.
5. Научились успешно применять навыки работы с модулями `shutil`, `sys`, `json`.