

ОСНОВЫ Python

Работа с файлами

- чтение запись файла
- парсинг текстовых форматов


```
fp = open('filename.txt', 'w')  
fp.write('Hello world!')  
fp.close()
```

```
fp = open('filename.txt', 'r')  
fp.read()  
fp.close()
```

```
fp = open('filename.txt', 'a')  
fp.write('Hello world!')  
fp.close()
```

- Модификаторы:

“r” - чтение

“w” - запись (файл стирается, если нету - создается)

“r+” - чтение и запись

“a” - запись в конец файла (если нету - создается)


```
fp = open('filename.txt', 'r+')  
fp.write('Hello world!')  
fp.seek(0)  
fp.read()  
fp.close()
```

```
fp = open('filename.txt', 'r')  
fp.read(5)  
fp.tell()  
fp.read(2)  
fp.tell()  
fp.close()
```

```
fp = open('filename.txt', 'a+')  
fp.write('Hello world!')  
fp.close()
```

- Курсор, при открытии файла есть такое понятие как курсор. Т.е. текущее место в файле от куда будет производиться чтение или запись


```
import os
size_of_file = os.path.getsize('test1.txt')
fp = open('filename.txt', 'r+')
fp.seek(size_of_file)
fp.write("Write to end of file")
fp.seek(0)
fp.write("Write to begin of file")
fp.close()
```

- Курсор можно устанавливать в явном виде.
- При записи, если после курсора есть другие символы, то они будут затерты


```
fp = open('filename.txt', 'r+')  
fp.read() # читает все  
fp.readline() # читает одну строку  
# (ДО СИМВОЛА \n)  
fp.readlines() # читает все в список  
# разбивая по символу перевода строк  
fp.read(5) # читает количество СИМВОЛОВ  
fp.close()  
  
fp = open('filename.txt', 'r+')  
fp.write("some string") # записывает строку  
fp.writelines(['a', 'b', 'c'])  
# записывает из итератора  
fp.close()
```



```
fp = open("data.txt", 'w')
d = {'a': 1, 'b': 2, 'c': 3}
s = repr(d) # '{"b": 2, "a": 1, "c": 3}'
fp.write(s); fp.close()
fp = open("data.txt", 'r')
s1 = fp.read() # '{"b": 2, "a": 1, "c": 3}'
d1 = eval(s1) # {'b': 2, 'a': 1, 'c': 3}
fp.close()
```

- каждый тип данных имеет строковое представление `repr`, которое можно записать в файл
- вычитав такое представление из файла его можно превратить в соответствующую переменную с помощью `eval`

json формат

```
import json
data = {'a': 1, 'b': [1, 2, 3], 'c': 3}
json_str = json.dumps(data)
# '{"b": [1, 2, 3], "a": 1, "c": 3}'
fp = open("file.json", 'w')
fp.write(json_str); fp.close()
fp = open("file.json", 'r')
json_str = fp.read()
# '{"b": [1, 2, 3], "a": 1, "c": 3}'
data = json.loads(json_str)
# {'a': 1, 'b': [1, 2, 3], 'c': 3}
fp.close()
```

```
import json
data = {'a': 1, 'b': [1, 2, 3], 'c': 3}
fp = open("file.json", 'w')
json.dump(data, fp); fp.close()
fp = open("file.json", 'r')
data = json.load(fp) fp.close()
# {'a': 1, 'b': [1, 2, 3], 'c': 3}
```

- json формат позволяет
сериализировать более
сложные структуры
данных

- стандарт передачи
данных в web, его
понимает javascript

- МОЖНО дампить
данные на прямую в
файл минуя шаг
стерилизации в строку.


```
import csv
fp = open("person.csv", "w")
csv_w = csv.writer(fp)
csv_w.writerow(["Name", "Surname", "Age"])
csv_w.writerow(["Ivan", "Ivanov", "25"])
csv_w.writerow(["Pet'er", "Petrov", "45"])
fp.close()
fp = open("person.csv", "r")
csv_r = csv.reader(fp)
for row in csv_r:
    print(row)
```

- табличный формат данных, столбцы разделены запятыми
- csv сериализует данные и экранирует спецсимволы в ЭТИХ ДАННЫХ