

Лекция 6

Файлы

1 Открытие и закрытие файла

```
<file>.open(file, mode='r', encoding=None, newline=None, ...)
```

```
<file>.close()
```

```
# открыть файл в текущей папке
```

```
f = open("test.txt")
```

```
# указание полного пути
```

```
f = open("C:/Python/README.txt")
```

```
# не правильно
```

```
# f = open("C:\Users\mike\py101book\nnn\test.txt")
```

```
# правильно
```

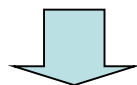
```
f = open(r"C:\Users\mike\py101book\nnn\test.txt")
```

Режимы открытия файла

Режим	Обозначение
'r'	открытие на чтение (является значением по умолчанию), указатель стоит в начале файла
'w'	открытие на запись, содержимое файла удаляется, указатель стоит в начале файла; если файла не существует, то создается новый файл
'x'	открытие на запись; если файла не существует, то создается новый файл, иначе исключение <code>FileExistsError</code>
'a'	открытие на дозапись, информация добавляется в конец файла (указатель стоит в конце файла); если файла не существует, то создается новый файл
'b'	открытие в двоичном режиме (постфикс)
't'	открытие в текстовом режиме (является значением по умолчанию) (постфикс)
'+'	открытие на чтение и запись

Режимы открытия файла 'x'

```
try:
    print('Проверка существования файла f.txt')
    f = open("f.txt", "x")
    print('Файл создан и открыть для записи')
except FileExistsError:
    print('Файл существует, сработало исключение')
```



Файла f.txt не существует:

```
Проверка существования файла f.txt
Файл создан и открыть для записи
>>>
```

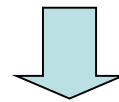
Файл f.txt существует:

```
Проверка существования файла f.txt
Файл создан и открыть для записи
>>>
```

Некоторые атрибуты файлового объекта

<code><file>.closed</code>	Возвращает True если файл был закрыт.
<code><file>.mode</code>	Возвращает режим доступа, с которым был открыт файл.
<code><file>.name</code>	Возвращает имя файла.

```
my_file = open("f.txt", "w")
print("Имя файла: ", my_file.name)
print("Файл закрыт: ", my_file.closed)
print("В каком режиме файл открыт: ", my_file.mode)
```



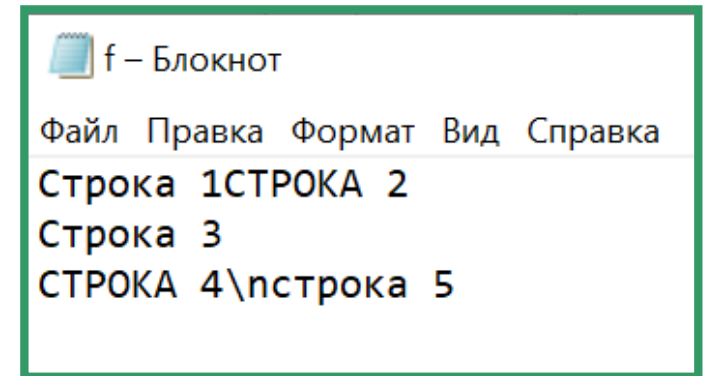
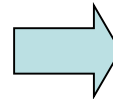
```
Имя файла:  f.txt
Файл закрыт:  False
В каком режиме файл открыт:  w
>>> |
```

2 Запись в файл

<file>.write(строка_для_записи)

```
my_file = open("f.txt", "w")
my_file.write("Строка 1")
my_file.write("СТРОКА 2\n")
my_file.write("Строка 3\n")
my_file.write(r"СТРОКА 4\n")
my_file.write("строка 5")

my_file.close()
```



```

>>> 'Байты'.encode('utf-8')
b'\xd0\x91\xd0\xb0\xd0\xb9\xd1\x82\xd1\x8b'
>>> 'Байты'.encode('utf-16')
b'\xff\xfe\x11\x040\x049\x04B\x04K\x04'
>>> bytes('Байты', encoding = 'utf-8')
b'\xd0\x91\xd0\xb0\xd0\xb9\xd1\x82\xd1\x8b'
>>> bytes('Байты', encoding = 'utf-16')
b'\xff\xfe\x11\x040\x049\x04B\x04K\x04'
>>> chr(65)
'A'
>>> bytes([65, 66, 67])
b'ABC'

```

```

my_file = open("f.txt", "wb")

b1 = b'Text 1\n'
my_file.write(b1)

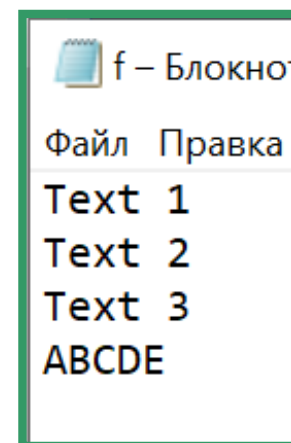
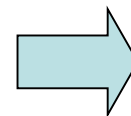
b2 = 'Text 2\n'.encode('utf-8')
my_file.write(b2)

b3 = bytes('Text 3\n', encoding = 'utf-8')
my_file.write(b3)

b4 = bytes([65, 66, 67, 68, 69])
my_file.write(b4)

my_file.close()

```



3 Чтение из файла

`<file>.read([количество_байт])`

```
my_file = open("f.txt", "w")
my_file.write("Строка 1\n")
my_file.write("СТРОКА 2\n")
my_file.write("Строка 3\n")
my_file.close()

m_f = open("f.txt", "r")
str = m_f.read()
print(str)
```



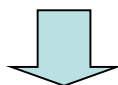
```
Строка 1
СТРОКА 2
Строка 3

>>>
```

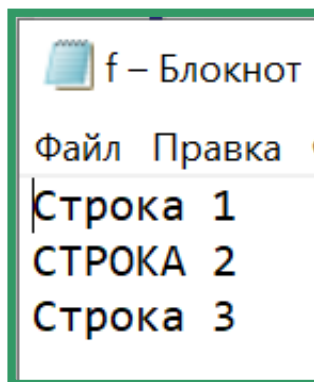


```
my_file = open("f.txt", "wb")
my_file.write("Строка 1\n".encode('utf-8'))
my_file.write("СТРОКА 2\n".encode('utf-8'))
my_file.write("Строка 3\n".encode('utf-8'))
my_file.close()

m_f = open("f.txt", "rb")
str = m_f.read()
print(str)
```



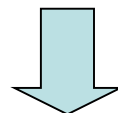
```
b'\xd0\xa1\xd1\x82\xd1\x80\xd0\xbe\xd0\xba\xd0\xb0 1\n\nd0\xa1\xd0\xa2\xd0\xa0\xd0\x9e\xd0\x9a\xd0\x90 2\n\nd0\xa1\xd1\x82\xd1\x80\xd0\xbe\xd0\xba\xd0\xb0 3\n'
>>>
```



```
my_file = open("f.txt", "wb")
my_file.write("Српoka 1\n".encode('utf-8'))
my_file.write("CTPOKA 2\n".encode('utf-8'))
my_file.write("Српoka 3\n".encode('utf-8'))
my_file.close()

m_f = open("f.txt", "rb")
str = m_f.read(2)
print(str)

str = m_f.read(5)
print(str)
m_f.close()
```



```
b'\xd0\xa1'
b'\xd1\x82\xd1\x80\xd0'
>>> |
```

tell() - сообщает в скольких байтах от начала файла находится указатель

seek(offset, [from]) - offset указывает на сколько байт перейти; аргумент *from* означает позицию, с которой начинается движение (0 - означает начало файла, 1 - нынешняя позиция, 2 - конец файла)

```
my_file = open("f.txt", "w+")
my_file.write("Строка 1\n")
my_file.write("СТРОКА 2\n")
my_file.write("Строка 3\n")

print("Указатель на позиции: ", my_file.tell())

my_file.seek(6)
print("Указатель на позиции: ", my_file.tell())
my_file.write("НОВОЕ")
print("Указатель на позиции: ", my_file.tell())

str = my_file.read()
print(str)


my_file.seek(0)
str = my_file.read()
print(str)

my_file.close()
```

Указатель на позиции: 30
Указатель на позиции: 6
Указатель на позиции: 11
ТРОКА 2
Строка 3

СтрокаНОВОЕТРОКА 2
Строка 3



 f – Блокнот

Файл Правка Формат Вид

СтрокаНОВОЕТРОКА 2
Строка 3

<file>.readline() - возвращает содержимое файла построчно; прочитав первую строку файла, операция readline при следующем запуске выведет вторую строку

<file>.readlines() - возвращает список строк, где строки представлены в виде отдельных элементов

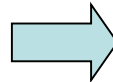
```
my_file = open("f.txt", "w+")
my_file.write("Строка 1\n")
my_file.write("СТРОКА 2\n")
my_file.write("Строка 3\n")

my_file.seek(0)
str = my_file.read()
print(str)

my_file.seek(0)
str = my_file.readline()
print(str)
str = my_file.readline()
print(str)

my_file.seek(0)
str = my_file.readlines()
print(str)

my_file.close()
```



```
Строка 1
СТРОКА 2
Строка 3

Строка 1
СТРОКА 2

['Строка 1\n', 'СТРОКА 2\n', 'Строка 3\n']
>>> |
```

4 Файлы CSV

Файл, который хранит данные в виде таблицы; для структурирования хранимых данных используются запятые (или другие разделители).

```
1 import csv
2
3 data = [ ["Tom", 28],["Alice", 23],["Bob", 34] ]
4
5 my_f = open('file.csv', "w+", newline = "")
6
7 # создание объекта для записи информации в файл
8 w = csv.writer(my_f, delimiter = ';')
9
10 # запись нескольких строк
11 w.writerows(data)
12 # запись одной строки
13 w.writerow( ["Alex", 40] )
14
15 # запись стандартным методом write
16 my_f.write( 'Rich;10\n' )
17 my_f.write( 'Olga;20\nRich;10\n' )
18
19 # вывод содержимого файла
20 my_f.seek(0)
21 r = csv.reader(my_f)
22 print(list(r))
23 print()
24
25 # формирования списка списков с 2 элементами
26 # и вывод на экран
27 my_f.seek(0)
28 r_new = []
29 for rr in r:
30     rr = rr[0].split(";")
31     r_new.append(rr)
32 print( r_new )
33 my_f.close()
```

```
[['Tom;28'], ['Alice;23'], ['Bob;34'], ['Alex;40'],
['Rich;10'], ['Olga;20'], ['Rich;10']]

[['Tom', '28'], ['Alice', '23'], ['Bob', '34'], ['A
lex', '40'], ['Rich', '10'], ['Olga', '20'], ['Rich
', '10']]
>>> |
```

Microsoft Excel - file

F25		
	A	B
1	Tom	28
2	Alice	23
3	Bob	34
4	Alex	40
5	Rich	10
6	Olga	20
7	Rich	10
8		
9		

file – Блокнот

Файл Правка Ф

```
Tom;28
Alice;23
Bob;34
Alex;40
Rich;10
Olga;20
Rich;10
```

5 Файлы XML

```
1 import xml.etree.ElementTree as ET
2 from xml.dom import minidom
3
4 # СОЗДАЕМ XML ФАЙЛ
5 # корневой элемент test
6 new = ET.Element('test')
7
8 # два узла
9 ET.SubElement(new, "node1", attr="str").text = "TestNode"
10 ET.SubElement(new, "node2").text = "123456789"
11
12 # третий узел - массив
13 arr = ET.SubElement(new, "array")
14 ET.SubElement(arr, "item").text = "Item1"
15 ET.SubElement(arr, "item").text = "Item2"
16 ET.SubElement(arr, "item").text = "Item3"
17
18 # декодируем в строку
19 str1 = ET.tostring(new).decode()
20 print(str1+'\n')
21
22 # формируем из строки дерево
23 xml_str = minidom.parseString(str1).toprettyxml()
24 print(xml_str)
25
26 # записываем в файл
27 f = open('test1.xml', 'w+')
28 f.write(xml_str)
29 f.close()
30
```

```
31 # ЧТЕНИЕ ДАННЫХ ИЗ ФАЙЛА
32 tree = ET.parse('test1.xml')
33
34 # получение корневого элемента
35 root = tree.getroot()
36
37 print(root.text.encode('utf-8'))
38 for elem in root:
39     print(elem.text.encode('utf-8'))
40     for subelem in elem:
41         print(subelem.text.encode('utf-8'))
42
```

XML - расширяемый
язык разметки,
предназначенный для
хранения и передачи
данных


```
<test><node1 attr="str">TestNode</node1><node2>1234
56789</node2><array><item>Item1</item><item>Item2</
item><item>Item3</item></array></test>
```

```
<?xml version="1.0" ?>
```

```
<test>
```

```
    <node1 attr="str">TestNode</node1>
```

```
    <node2>123456789</node2>
```

```
    <array>
```

```
        <item>Item1</item>
```

```
        <item>Item2</item>
```

```
        <item>Item3</item>
```

```
    </array>
```

```
</test>
```

```
b'\n\t'
```

```
b'TestNode'
```

```
b'123456789'
```

```
b'\n\t\t'
```

```
b'Item1'
```

```
b'Item2'
```

```
b'Item3'
```

```
>>>
```

test1 – Блокнот

Файл Правка Формат Вид Справка

```
<?xml version="1.0" ?>
```

```
<test>
```

```
    <node1 attr="str">TestNode</node1>
```

```
    <node2>123456789</node2>
```

```
    <array>
```

```
        <item>Item1</item>
```

```
        <item>Item2</item>
```

```
        <item>Item3</item>
```

```
    </array>
```

```
</test>
```