

УДК 004.43

PYTHON ПРОГРАММАЛОО ТИЛИНДЕ ФАЙЛДАР МЕНЕН ИШТӨӨНҮН ЫКМАЛАРЫ

Пирматов Абдыманап Зияйдинович - ф.-м.и. к,
pirmatov@mail.ru

Идрисова Гулбарчын Ильясовна - окутуучу.
gulbarcynidrisova@mail.ru

Б.Осмонов атындагы ЖАМУ, Жалал-Абад ш,
Кыргыз Республикасы

Аннотация: Макалада Python программалоо тилинде файлдар менен иштөөнүн ыкмаларына токтолдук жана ар бир ыкма үчүн практикалык мисалдар карап, программаларын түзүп көрсөттүк.

Түйүндүү сөздөр: Python программалоо тили, бинардык режим, open жана close функциялары, write берилгендер, writelines удаалаштыгы, writable, read количество, readline количество, flush, fileno, truncate сан, seekable, encoding ыкмалары.

МЕТОДЫ ДЛЯ РАБОТЫ ФАЙЛАМИ В ЯЗЫКЕ ПРОГРАМИРОВАНИЯ PYTHON

Пирматов Абдыманап Зияйдинович - к.ф.-м.н.,
pirmatov@mail.ru

Идрисова Гулбарчын Ильясовна - преподаватель
gulbarcynidrisova@mail.ru

ЖАГУ имени Б.Осмонова, г.Жалал-Абад,
Кыргызская Республика

Аннотация: В этом статье мы рассматривали методы для работы в языке программирования Python и для каждого метода рассмотрен конкретный пример.

Ключевые слова: Язык программирования Python, в бинарном режиме, функции open, close, write данные, writelines последовательность, writable, read количество, readline количество, методы flush, fileno, truncate количество, seekable.

METHODS FOR WORKING FILES IN THE PYTHON PROGRAMMING LANGUAGE

Pirmatov Abdymanap Ziyaydinovich - can. of f.-m.s.,
pirmatov@mail.ru

Idrisova Gulbarchyn Iliasovna – teacher
gulbarcynidrisova@mail.ru

JASU named after B.Osmonov, Jalal-Abad city,
Kyrgyz Republic

Annotation: In this article, we looked at methods for working in the Python programming language, and for each method, a specific example is considered.

Key words: Python programming language, in binary mode, functions open, close, write data, writeline sequence, writable, read quantity, readline quantity, methods flush, fileno, truncate quantity, seekable.

Python программалоо тилинде файлдар менен иштөөдө open () функциясы менен файлдарды ачкандан баштап, алар менен андан иш жүргүзүүгө боло тургандыгын билебиз. Анда файлдар менен иштөөдөгү негизги ыкмаларды карап чыгабыз:

1) **close ()** — бул ыкма файлды жабат. Файлдарды жаап иштөө – биринчиден, "ResourceWarning: unclosed file" эскертме билдирүүсү чыгуусун алдын алат, экинчиден, файлдарды жаап иштөө программалоонун жакшы стили болуп саналат.

Python программалоо тили контексттин протокол менеджерин камтыгандыктан, бул протокол файлдарды сөзсүз түрдө жабат.

Мисал:

```
with open (r"file.txt", "w», encoding="cp1251") as f:
    f.write("Жолчо") # Жолчону файлга жазабыз
# Ушул жерде файл автоматтык түрдө жабылат
```

2) **write (<Берилгендер>)** — бул ыкма жолчону же байттардын удаалаштыгын файлга жазат. Эгерде <Берилгендер> параметри жолчо түрүндө берилсе, анда файлды тексттик режимде ачуу керек, ал эми байттардын удаалаштыгын жазуу үчүн, файлды бинардык режимде ачуу зарыл. Эч качан жолчону файлга бинардык режимде жазууга, ал эми байттардын удаалаштыгын тексттик режимде жазууга болбойт.

Файлга жазууга мисал:

```
# Текстовый режим
f = open (r"file.txt", "w", encoding="cp1251")
f.write ("Информатика \нкафедрасы") # Записываем строку в файл
f.close () # файлды жабабыз
```

Жыйынтыгын Блокнот программасы менен file.txt файлын окусак:

Информатика
кафедрасы

деген жазууну алабыз.

```
# Бинарный режим
f = open (r"file, txt», «wb")
f.write (bytes ("Информатика /нкафедрасы", "cp1251"))
f.write (bytearray ("\n2021- жыл", "cp1251"))
f.close()
```

Жыйынтыгын Блокнот программасы менен file.txt файлын окусак:

Информатика кафедрасы 2021-жыл

деген жазууну алабыз.

3) **writelines (<Удаалаштык>)** — бул ыкма файлга удаалаштыктарды жазат. Эгерде удаалаштыктардын бардык элементтери жолчолор болсо, анда файл тексттик режимде ачык болушу керек, ал эми удаалаштыктын бардык элементтери байттар болсо, анда файлды бинардык режимде ачуу керек.. Тизменин элементтерине мисал:

```
# Текстовый режим
f = open(r"file.txt", "w", encoding="cp1251")
f.writelines(["Информатика\n", "кафедрасы -\n", "2021-ж"])
f.close()
```

Жыйынтыгын Блокнот программасы менен file.txt файлын окусак:

Информатика

кафедрасы –

2021-ж

деген жазууну алабыз.

```
# Бинарный режим
f = open(r"file.txt", "wb")
arr = [bytes("Информатика\n", "cp1251"), bytes("кафедрасы", "cp1251")]
f.writelines(arr)
f.close ()
```

Жыйынтыгын Блокнот программасы менен file.txt файлын окусак:
Информатика кафедрасы
деген жазууну алабыз.

4) **writable ()** — бул ыкма, эгерде файлга жазууга мүмкүн болсо, **True** маанисин берет, жана башка учурда **False** — маанисин берет:

```
f = open (r"file .txt", "r")           # файлды окуу үчүн ачуу
f.writable ()
```

Жыйынтыгында:

False маанисин алабыз.

```
f = open (r"file. txt", "w")          # файлды жазуу үчүн ачуу
f.writable ()
```

Жыйынтыгында:

True маанисин алабыз.

5) **read([<Количество>])** — бул ыкма файлдан берилгендерди окуйт. Эгерде файл тексттик режимде ачык болсо, анда маанисин жолчо түрүндө окуйт, ал эми бинардык режимде болсо, анда байттардын удаалаштыгы түрүндө окуйт. Параметрде сан көрсөтүлсө, ошончо символ же байт окуйт. Эгерде файлдын акырына чейин жетсе, бул ыкма бош жолчо кайтарат. Мисал:

```
# Текстовый режим
f = open (r"file.txt", "r», encoding="cp1251")
s=f .read (8)           # 8 символ окуйбуз
print(s)
s=f .read (16)          # 6 символ окуйбуз
print(s)
Жыйынтыгы:
ика
кафедрасы
s=f .read (8)           # Файлдын акырына жетти
print(s)
Жыйынтыгы:
«»
f. close ()
```

```
# Бинарный режим
f=open(r"file.txt", "rb")
s=f.read (8) # Считываем 8 символов
print(s)
Жыйынтыгы:
b'\xc8\xed\xfa\xec\xef\xec\xef\xef\xef'
```

6) `readline` ([<Количество>]) — бул ыкма ар бир аткарганда бир эле жолчону окуйт. Эгерде файл тексттик режимде ачык болсо, анда мааниси жолчо, ал эми бинардык режимде болсо, анда мааниси байттардын удаалаштыгы болот. Файлдын акырына жеткенде бош символ чыгат. Мисал:

```
# Текстовый режим
f = open (r"file.txt», «r», encoding="cp1251")
s=f.readline()
print(s)
Жыйынтыгы:
Информатика кафедрасы
s1=f.readline()          # файл акырына жетти
print(s1)
Жыйынтыгы:
«»
f.close ()
```

```
# Бинарный режим
f=open(r"file.txt","rb")
s=f.readline(6)
print(s)
b'\xc8\xed\xfa\xee\xfo\xec'
s2=f.readline(4)
print(s2)
b'\xe0\xf2\xe8\xea'
>>s3=f.readline()
print(s3)
b'\xe0 \xea\xe0\xf4\xe5\xe4\xf0\xe0\xf1\xfb '
f.close ()
```

7) `flush` () — бул ыкма сөзсүз түрдө буфердеги берилгендерди диске жазат;

8) `fileno` () — бул ыкма файлдын дескрипторун окуйт. Анын мааниси дайыма 2 санынан чоң, себеби 0 саны `stdin` стандарттык кийрүүгө таандык, 1 — `stdout` стандарт чыгарууга таандык, ал эми 2 саны— каталар жөнүндөгү `stderr` стандартдык чыгарууга таандык.

Мисал:

```
f = open (r"file.txt", "r», encoding="cp1251")
d=f. fileno ()          # Дескриптор файла
print(d)
f. close ()
Жыйынтык:
3
```

9) `truncate`([<сан>]) — бул ыкма параметрде көрсөтүлгөн сандагы символдорду(эгерде тексттик режимде болсо) же байттарды(эгерде бинардык режимде болсо) бөлүп алат жана файлдын жаңы өлчөмүнүн маанисин кайиарат. Мисал:

```
f = open (r"file. txt”, "r+», encoding="cp1251")
s=f.read()
k=f.truncate (5)
f. close ()
```

Жыйынтык:

Информатика кафедрасы

5

10) **seekable** () — бул ыкмада файлдагы көрсөткүчү башка позицияга жылса, True маанисин берет, башка учурда False маанисин берет:

```
f = open("file.txt", "r")
```

```
t=f.seekable ()
```

```
print(t)
```

Жыйынтыгы:

True

11) **encoding**— бул ыкма файлга жазуунун же файлды окуунун алдында, жолчолорду өзгөртүүгө колдонгон кодировканын атын чыгарат. Атрибут тексттик режимде гана колдонулат. Мисал:

```
f = open("file.txt", "a", encoding="cp1251")
```

```
a=f.encoding
```

```
f.close()
```

Жыйынтыгы:

```
'cp1251'
```

Ошентип, биз Python программалоо тилинде файлдар менен иштөөдөгү ыкмаларга кыскача токтолдук жана ар бир ыкманы мисалдарда программа түзүү менен көрсөттүк.

Адабияттар:

1. Любанович Билл. Простой Python. Современный стиль программирования. — СПб.: Питер, 2016. — 480 с.: ил.
2. Доусон М. Програмируем на Python. — СПб.: Питер, 2014. — 416 с.
3. Лутц М. Изучаем Python, 4-е издание. — Пер. с англ. — СПб.: Символ-Плюс, 2011. — 1280 с.
4. Хахаев И.А. Практикум по алгоритмизации и программированию на Python. — М.: Альт Линукс, 2010. — 126 с. (Библиотека ALT Linux).
5. Интернет ресурс: <https://www.python.org/>

Программалык жабдылышы: программалар Python 3.8 платформасында текшерилди.