Python for AI

Daurenov AD

№6 ДӘРІС. Python тілінде файлдармен және CSV кестелік деректерімен жұмыс істеу

Дәрістің мақсаты

- Python тілінде файлдармен жұмыс істеудің негізгі принциптерін,
- CSV форматындағы кестелік деректерді оқу, жазу және өңдеу әдістерін меңгеру.
- Көлік жүйелерінің өлшеу деректерін файлдарда сақтап, оларды бағдарламалық түрде талдау жолдарын қарастыру.

Күтілетін нәтижелер:

- білім алушылар файлдардың қызметін түсіндіреді,
- open(), with, csv.reader(), csv.DictReader(), csv.writer() құралдарын қолданады,
- өлшеу деректерін талдап, нәтижелерді жаңа файлға сақтай алады.

1. Python бағдарламасында файлдар не үшін қажет?

Алдыңғы дәрістерде біз деректерді Python бағдарламасының ішінде сақтадық.
Мысалы: temperature = [88, 92, 95, 101, 106]
Бұл жағдайда өлшеу мәндері бағдарлама кодында тікелей көрсетілген.

Алайда нақты көлік жүйелерінде өлшеу деректерінің көлемі әлдеқайда үлкен болуы мүмкін.
Мысалы, электрмобильдегі батареяны басқару жүйесі (BMS) кернеу, ток және температура туралы ақпаратты тұрақты түрде өңдейді. Мұндай деректердің белгілі бір бөлігін кейінгі талдау үшін журналға немесе файлға жазуға болады.

Файл — деректерді сақтау құрылғысында сақтауға арналған атауы бар ақпараттық объект.
Файлдарды пайдалану арқылы:

- деректерді бағдарлама аяқталғаннан кейін де сақтауға;
- бұрын жазылған өлшеулерді қайта оқуға;
- бірнеше мың өлшеуді өңдеуге;
- нәтижелерді басқа бағдарламаларға беруге;
- машиналық оқыту үшін деректер жиындарын дайындауға болады.

1. Температура датчигі

Көлік жүйесіндегі температура датчигі



Өлшеу мәндері (мысал):

88°C
92°C
95°C
101°C
106°C

2. Контроллер / деректерді тіркеу жүйесі

Датчиктен алынған өлшеу мәндерін жинау



Жиналған деректер:

Температура (°C)
88
92
95
101
106
...

3. CSV файлы

Өлшеу мәндері CSV файлына сақталады

CSV

ID	Temperature
1	88
2	92
3	95
4	101
5	106
...	...

4. Python бағдарламасы

CSV файлын оқу, есептеу және классификациялау

```python
import csv

data = []
with open("temperature.csv", "r")
  as file:
    reader = csv.DictReader(file)
    for row in reader:
      temp = float(row["Temperature"])
      status = check_temperature(temp)
      data.append((temp, status))

# Есептеу және талдау
```

5. Талдау нәтижесі

Әр өлшеу мәні бағаланып, күйі анықталады

88°C	NORMAL
92°C	NORMAL
95°C	NORMAL
101°C	OVERHEATING
106°C	OVERHEATING

Жалпы талдау нәтижесі

Минимум
88°C

Максимум
106°C

Орташа мән
96.4°C

Классификация:
NORMAL: 3
OVERHEATING: 2

2. Файлдардың негізгі түрлері

Python әртүрлі файл форматтарымен жұмыс істей алады.

Файл түрі	Кеңейтімі	Қолданылуы
Мәтіндік файл	`.txt`	Қарапайым мәтіндерді сақтау
CSV файлы	`.csv`	Кестелік деректерді сақтау
JSON файлы	`.json`	Құрылымдалған деректерді сақтау
Excel файлы	`.xlsx`	Электрондық кестелер
Бинарлық файл	`.bin`	Екілік форматтағы ақпарат

Осы дәрісте мәтіндік файлдар мен CSV форматына назар аударамыз.

2.1 Мәтіндік және бинарлық файлдар

Мәтіндік файлдарда ақпарат символдар түрінде сақталады. Python оларды белгілі бір кодтаумен, мысалы UTF-8 арқылы оқиды.

Бинарлық файлдарда ақпарат байттар түрінде сақталады. Мұндай файлдар кескіндер, бағдарламалар немесе арнайы құрылғы деректері үшін қолданылуы мүмкін.

3. Python тіліндегі open() функциясы

Python файлды ашу үшін кіріктірілген open() функциясын қолданады.
Жалпы синтаксисі:
open("filename", "mode")

Мұнда:

- filename — файлдың атауы немесе орналасу жолы;
- mode — файлмен жұмыс істеу режимі.

Негізгі режимдер

Режим	Атауы	Қызметі
`"r"`	Read	Файлды оқу
`"w"`	Write	Файлға жазу
`"a"`	Append	Файл соңына деректер қосу
`"x"`	Exclusive creation	Тек жаңа файл құру
`"b"`	Binary	Бинарлық режим
`"t"`	Text	Мәтіндік режим

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

Бұл код data.txt файлын оқу режимінде ашады.
Егер файл табылмаса, FileNotFoundError қатесі пайда болады.
Маңызды: "w" режимі бұрыннан бар файлдың мазмұнын өшіреді. Сондықтан маңызды файлдармен жұмыс істегенде режимді дұрыс таңдау қажет.

with open() конструкциясы

Файлдармен жұмыс істеудің ұсынылатын тәсілі — with open() конструкциясын пайдалану.

```
with open("data.txt", "w", encoding="utf-8") as file:
    file.write("Hello Python")
```

Бағдарлама data.txt файлын құрып, оған мәтін жазады.
Мұндағы:

- with — файлмен жұмыс істеу блогын ұйымдастырады;
- open() — файлды ашады;
- as file — файл объектісіне file атауын береді;
- write() — мәтінді файлға жазады.

with блогы аяқталғаннан кейін файл автоматты түрде жабылады, соның ішінде блок ішінде қате пайда болған жағдайда да.

UTF-8 дегеніміз не?
UTF-8 — әртүрлі тілдердегі таңбаларды дұрыс көрсетуге және сақтауға мүмкіндік беретін мәтіндік кодтау жүйесі.
Мысалы:

- Ағылшын тілі: Hello
- Орыс тілі: Привет
- Қазақ тілі: Сәлем, Қозғалтқыш
- Арнайы таңбалар: °C, №, %

Файлды қайта оқу

```
with open("data.txt", "r", encoding="utf-8") as file:
    content = file.read()

print(content)
```

CSV форматы дегеніміз не?

CSV (Comma-Separated Values) — кестелік деректерді мәтін түрінде сақтайтын файл форматы.
CSV форматында әр жол жеке жазбаны білдіреді, ал бағандар арнайы бөлгіш таңбамен ажыратылады. Көбінесе үтір қолданылады.
CSV форматының артықшылықтары — қарапайым құрылымы, шағын көлемі және әртүрлі бағдарламалармен үйлесімділігі.
CSV файлдарында деректердің түрлері автоматты түрде сақталмайды: Python оқыған кезде оларды қажет болған жағдайда сандық түрге түрлендіру керек.

CSV және Excel форматтарын салыстыру

Сипаттама	CSV	Excel
Кеңейтімі	`.csv`	`.xlsx`
Құрылымы	Мәтіндік кесте	Электрондық жұмыс кітабы
Формулалар	Сақталмайды	Қолдайды
Диаграммалар	Сақталмайды	Қолдайды
Бірнеше парақ	Жоқ	Бар
Python арқылы өңдеу	`csv` , Pandas	`openpyxl` , Pandas
Негізгі қолданылуы	Деректер алмасу	Кестелік есептеулер мен талдау

CSV форматы (`.csv`)



temperature.csv

```
1  ID,Temperature
2  1,88
3  2,92
4  3,95
5  4,101
6  5,106
```

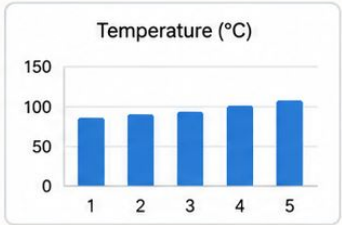
- Қарапайым мәтіндік файл
- Мәндер үтірмен (,) бөлінеді
- Форматтау, формулалар, диаграммалар жоқ
- Деректер алмасу үшін қолайлы

Excel форматы (`.xlsx`)



temperature.xlsx

	A	B
1	**ID**	**Temperature (°C)**
2	1	88
3	2	92
4	3	95
5	4	101
6	5	106



- Электрондық жұмыс кітабы
- Кестелер, формулалар, диаграммалар қолдайды
- Бірнеше парақ (sheet) болады
- Кестелік есептеулер мен талдау үшін қолайлы

Python тіліндегі `csv` кітапханасы

Python құрамында CSV файлдарымен жұмыс істеуге арналған стандартты `csv` модулі бар. Бұл модульді бөлек орнату қажет емес, өйткені ол Python-ның стандартты кітапханасына кіреді.

Орнату қажет емес!

`import csv` командасын қолдану жеткілікті.

1. csv.reader() — оқу

CSV файлды жолдар тізімі ретінде оқу.

```
import csv

with open("temperature.csv", "r",
          encoding="utf-8") as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```



temperature.csv

Нәтиже (жолдар тізімі)

```
['ID', 'Temperature']
['1', '88']
['2', '92']
['3', '95']
['4', '101']
['5', '106']
```

2. csv.DictReader() — оқу

CSV файлды сөздік (**dict**) түрінде оқу.

```
import csv

with open("temperature.csv", "r",
          encoding="utf-8") as file:
    reader = csv.DictReader(file)
    for row in reader:
        print(row)
```



temperature.csv

Нәтиже (сөздік түрі)

```
{'ID': '1', 'Temperature': '88'}
{'ID': '2', 'Temperature': '92'}
{'ID': '3', 'Temperature': '95'}
{'ID': '4', 'Temperature': '101'}
{'ID': '5', 'Temperature': '106'}
```

3. csv.writer() — жазу

Деректерді CSV файлына жазу.

```
import csv
data = [
    ['ID', 'Temperature'],
    [1, 88],
    [2, 92],
    [3, 95]
]
with open("output.csv", "w",
          encoding="utf-8", newline="") as file:
    writer = csv.writer(file)
    writer.writerows(data)
```



output.csv

ID	Temperature
1	88
2	92
3	95

4. csv.DictWriter() — жазу

Сөздіктерді CSV файлына жазу.

```
import csv
data = [
    {'ID': 1, 'Temperature': 88},
    {'ID': 2, 'Temperature': 92},
    {'ID': 3, 'Temperature': 95}
]
with open("output.csv", "w",
          encoding="utf-8", newline="") as file:
    writer = csv.DictWriter(file,
                fieldnames=['ID', 'Temperature'])
    writer.writeheader()
    writer.writerows(data)
```



output.csv

ID	Temperature
1	88
2	92
3	95

Негізгі құралдар

Құрал	Қызметі
`csv.reader()`	CSV жолдарын тізім түрінде оқу
`csv.DictReader()`	CSV жолдарын сөздік түрінде оқу
`csv.writer()`	CSV файлына жолдарды жазу
`csv.DictWriter()`	Сөздіктерді CSV файлына жазу

CSV файлдарымен жұмыс істеу кезінде кездесетін қателер

Қате	Себебі	Шешімі
`FileNotFoundError`	Файл табылмайды	Файл атауы мен жолын тексеру
`ValueError`	Мәтінді санға түрлендіру мүмкін емес	Деректердің дұрыстығын тексеру
`KeyError`	Көрсетілген баған атауы жоқ	CSV тақырыптарын тексеру
`PermissionError`	Файлға қол жеткізуге рұқсат жоқ	Файл рұқсаттарын тексеру
`ZeroDivisionError`	Бос тізімнің орташа мәні есептелді	Есептеуден бұрын тізімнің бос еместігін тексеру

Мысал: CSV файлындағы қате



temperature.csv

```
1  ID,Temperature
2  1,88
3  2,92
4  3,error
5  4,101
```

Мәтін (error), сан емес!

Неліктен қате пайда болады?

1. CSV файлының 3-жолында Temperature бағанында санның орнына мәтін (error) жазылған.
2. Python бұл мәнді `float()` көмегімен санға түрлендіруге тырысады.
3. Мағынасы сан емес болғандықтан `ValueError` қатесі пайда болады.

Python коды (қате тудырады)

```python
import csv

data = []
with open("temperature.csv", "r", encoding="utf-8") as file:
    reader = csv.reader(file)
    for row in reader:
        temp = float(row[1])   # 3-жолда қате болады
        data.append(temp)
```

Қате нәтижесі

```
Traceback (most recent call last):
  File "example.py", line 6, in <module>
    temp = float(row[1])
ValueError: could not convert string to float: 'error'
```

Қорытынды: CSV файлдарымен жұмыс істегенде деректердің дұрыстығын тексеру, қателерді өңдеу және деректерді тексеру өте маңызды.

Бақылау сұрақтары

1. Файл дегеніміз не және ол Python бағдарламасында не үшін қажет?
2. Мәтіндік және бинарлық файлдардың айырмашылығы қандай?
3. open() функциясы қандай қызмет атқарады?
4. "r", "w" және "a" режимдерінің айырмашылығы қандай?
5. with open() конструкциясын қолданудың артықшылығы неде?
6. CSV форматы дегеніміз не?
7. CSV және Excel форматтарының негізгі айырмашылықтары қандай?
8. csv.reader() және csv.DictReader() арасында қандай айырмашылық бар?
9. CSV файлынан алынған мәндерді не үшін float() түріне түрлендіреміз?
10. writer.writerow() әдісі қандай қызмет атқарады?
11. CSV файлындағы өлшеулерді Python тізіміне қалай сақтаймыз?
12. CSV деректерін талдауда функцияларды қалай қолдануға болады?
13. Өңделген деректерді жаңа CSV файлына сақтаудың қандай артықшылықтары бар?
14. Көлік жүйелерінде CSV файлдары қандай мақсатта қолданылуы мүмкін?
15. CSV деректерінің машиналық оқытумен қандай байланысы бар?